NIS-Q218-1

DMS-100

CompuCALL / Meridian SCAI

Interface Specification

CCM13

Version: 10.01 Standard March 2000

NORTEL NETWORKS™

How the world shares ideas.

DMS-100

CompuCALL / Meridian SCAI

Interface Specification

Document number: NIS-Q218-1
Version: 10.01
Document Status: Standard
Date: March 2000

Printed in the United States of America

Information is subject to change without notice. Nortel Networks reserves the right to make changes in this Specification or the equipment which it relates at any time without notice and without liability.

Publication history

Note: The version number refers to the document. The release number refers to the software.

Error Code Change History

This section lists the name changes, additions and deletions of error codes in the current software release. The left-hand column shows in alphabetical order, the names of the error codes and the release last known by the name. The corresponding new names, additions or deletions and the release are presented on the right-hand column.

Name and Release	Name and Release
already-Logged-On 03.01	link-Already-In-Use 04.01
no-Software-Resources 03.01	resource-Limitation 04.01
not-Logged-On 03.01	removed 04.01
not-Allowedn 04.01	
make-Call-In-Progress 03.01	removed 04.01
make-Call-Resources-Unavailablen 04.01	
unexpected-AcctCoden 04.01	
make-Call-Aborted 05.01	

Note applicable to dv-DN-Associate and dv-Appl-Continuity ROs only.

Parameter Name Change History

This section lists the name changes in the parameters in the current software release. The left-hand column shows the names of the parameter in Release 02.01 in alphabetical order and the corresponding new names in Release 03.01 are presented on the right-hand column.

Release 02.01	Release 03.01
CallingNumber	OrigCallingNumber
ChargeNumber	OrigChargeNumber
OrigACDDN	OrigInboundDN

Version 10.01

Updated NIS-Q218-1 for NA011, NA012 and NA013. Features updated consisted of the following:

• AF7737	Enhanced Walkaway
• AU3190	ICM Configuration
• AU3192	MWI Activation/Deactivation
• AU3193	NICM on PRI
• 59006736	Enhanced 3WC status reporting
• 59006731	MDC Residential Call Queueing
• 59006746	TAPI Extensions
• 59011953	IncACDgroups/Session raised to 100
• 59011948	CPA on Orig Inbound DN

Version 09.01

The release applicability for the NIS-Q218-1 is changed from NA010 to CCM10 to include global markets.

The coded example for dv-Agent-Status-U includes the hex dump for tuple Incallstatus.

The hex dump tag for return result and return error of dv-DN-Query includes the latest hex dump tags.

Version 08.01

The service version SCAI 12 delivers the messages for CCM10 software release. Any service version other than SCAI 12, does not deliver any new messages or changed messages. Any previous messages with different SCAI versions does work with SCAI 12.

The CCM10 release provides the following functionalities:

1 The ability to accurately identify an extended call.

a. Transferred

b. Call Forward Don’t Answer

c. Routed by Host Computer

d. Redirected

e. Threshold Routed or NACD Overflowed

2. The ability to reserve an ACD agent for a specific time and unreserve an ACD agent at the target DMS-100 switch, using the host computer.
3. The ability to notify the host computer that the previous reserved agent is unreserved by the DMS-100 switch, because of the reservation time-out timer.

The Network Intelligent Call Management feature introduces the message dv-Agent-SetAction-u message. The switch uses the dv-Agent-SetAction-U message to notify the host computer that the agent is unreserved.

The ICM Variable Wrap Reporting feature modifies the dv-Call-Released-U message to report the agent status. This includes variable wrap, release guard, not ready, or logged out status. The Postcall Status parameter reports the agent status.

The feature ICM Variable Wrap Reporting also causes the switch to send a dv-Agent-SetAction-U message when variable wrap times out and the idle agent waits.

Version 07.01

The service version SCAI 11 delivers the messages for the CCM09 software release. Any service version other than SCAI 11, does not deliver any new messages or changed messages. Any previous messages with different SCAI versions will work with SCAI 11.

Version 06.01

The service version SCAI 10 delivers the messages for the CCM08 software release. Any service version other than SCAI 10 does not deliver any new messages or changed messages. Service versions are compatible. Any previous messages with different SCAI versions work with SCAI 10.

Contents

3.0 Session management options 83

4.0 Application service options 121

List of figures

List of tables

About this specification

Applicability of this specification

This document details the use of Switch Computer Application Interface (SCAI) between the DMS-100 / SL-100 and the customer provided equipment (e.g. host computer), which allows access to CompuCALL services. Illustrations show how user applications on these computers can access these services. There is no intent in this specification to define or constrain the scope of any user application that could make use of these services.

Purpose

The purpose of the CompuCALL protocol architecture is to provide a uniform methodology for the development of applications operating between switch and computer equipment. An example of the functionality for switch-computer applications is the computerized use of services currently available from a telephone.

Audience

This document provides information for:

- Operating companies who offer services based on CompuCALL services.
- Equipment vendors who develop end-user applications using operating company services based on CompuCALL.
- Software developers who develop applications for end-users based on CompuCALL.

What you need to know

CompuCALL used in the specification refers to the DMS-100 CompuCALL interface and the SL-100 Meridian SCAI interface. References to DMS-100 system also refers to the SL-100 system.

Page Convention

Each page in the specification is identified by the page number. Chapters are numbered starting with “1.0”. The exception is the introductory Chapter.

Also the figures in this document are identified by figure number.

Introduction

The Nortel Networks trademark, CompuCALL, refers to a group of products and services which use the DMS-100 CompuCALL interface. Meridian SCAI is a group of products and services that use SL-100 Meridian SCAI interface. This document contains the specification for the CompuCALL/Meridian SCAI interface. Also, the switch computer application interface between the Nortel Networks DMS-100 / SL-100 switch, and a host computer.

CompuCALL/Meridian SCAI adheres to the Open System Interconnection (OSI) model. It carries application information over independent protocols. This specification provides information on protocol options for the transport of CompuCALL/Meridian SCAI on X.25.

Scope

The Switch Computer Application Interface (SCAI) defined by the American National Standards Institute (ANSI) T1S1 group, of which Nortel Networks is an active contributor, has a direct impact on this specification.

Switch Computer Application Interface

CompuCALL provides a data communications channel between a computer and a switch that allows an operating company to provide coordinated switch-based services to applications residing on a customer's host computer in a cost efficient manner. These switch-based services, when coupled with a computer and the association of voice and data terminals, benefits the end customer. The coordination that provided through the messaging over the link allows the implementation of specific solutions that improve end-user productivity and competitiveness. CompuCALL provides an open standard interface and allows application vendors to build appropriate user solutions for a competitive business environment.

Introduction to ICCM Integration with CompuCALL

Integrated Call Center Manager (ICCM) is an advanced call center application delivered by CompuCALL. ICCM runs on adjunct processors, not on the switch. ICCM requires CompuCALL to provide switch information control. The set of requirements placed on CompuCALL to support integration with ICCM are Basic ICCM functionality-Selective Queueing and Treatments.

Selective Queueing is the capability to route a call that comes into a queue, to a destination specified by the host computer. The host also has the capability to route the calls in the queue irrespective of its place in the queue (call plucking).

Treatments is the capability to direct specific treatments to a call located in an ACD incoming or overflow call queue. The ICCM Call Center Application has a set of requirements which are driving this development. This feature is the first phase of delivering treatment messaging capabilities.

It is important to note that although ICCM requirements must be met, this capability is available to all Call Center applications who subscribe. The development of this capability must take into account the various scenarios used and how this capability interacts with existing capabilities, both switch ACD software and ACD CompuCALL software.

Incentive

The utilization of computers in the telecommunications environment provides new services that meet end customer needs. In the switch and computer environment the alignment of the services with the switch and the computer interact. Application developers are limited to many switch and computer services provided over proprietary interfaces. This restriction and the multiple interface specializations slows the introduction of new services. Also, many of the interfaces in the switch and computer environment do not have the ability to integrate telephony services with existing data applications.

Configuration Environment

Figure 1, “Voice and Data Configurations with CompuCALL Link” on page 32 shows the relationship between the switching equipment, computer equipment, and the terminal equipment. There are several possibilities for utilizing a switch-computer interface. The various voice and data configurations in Figure 1 identify their relationship to the CompuCALL link.

Type 1
Depicts the case where the user has a Plain Old Telephone Service (POTS) voice terminal or a Meridian Business Set (MBS) and a data terminal which is connected to a host using a RS-232, LAN, or other suitable physical host connection. CompuCALL coordinatee the voice and data service to the user. This function is performed in addition to the physical control of the telephone set, by the user.

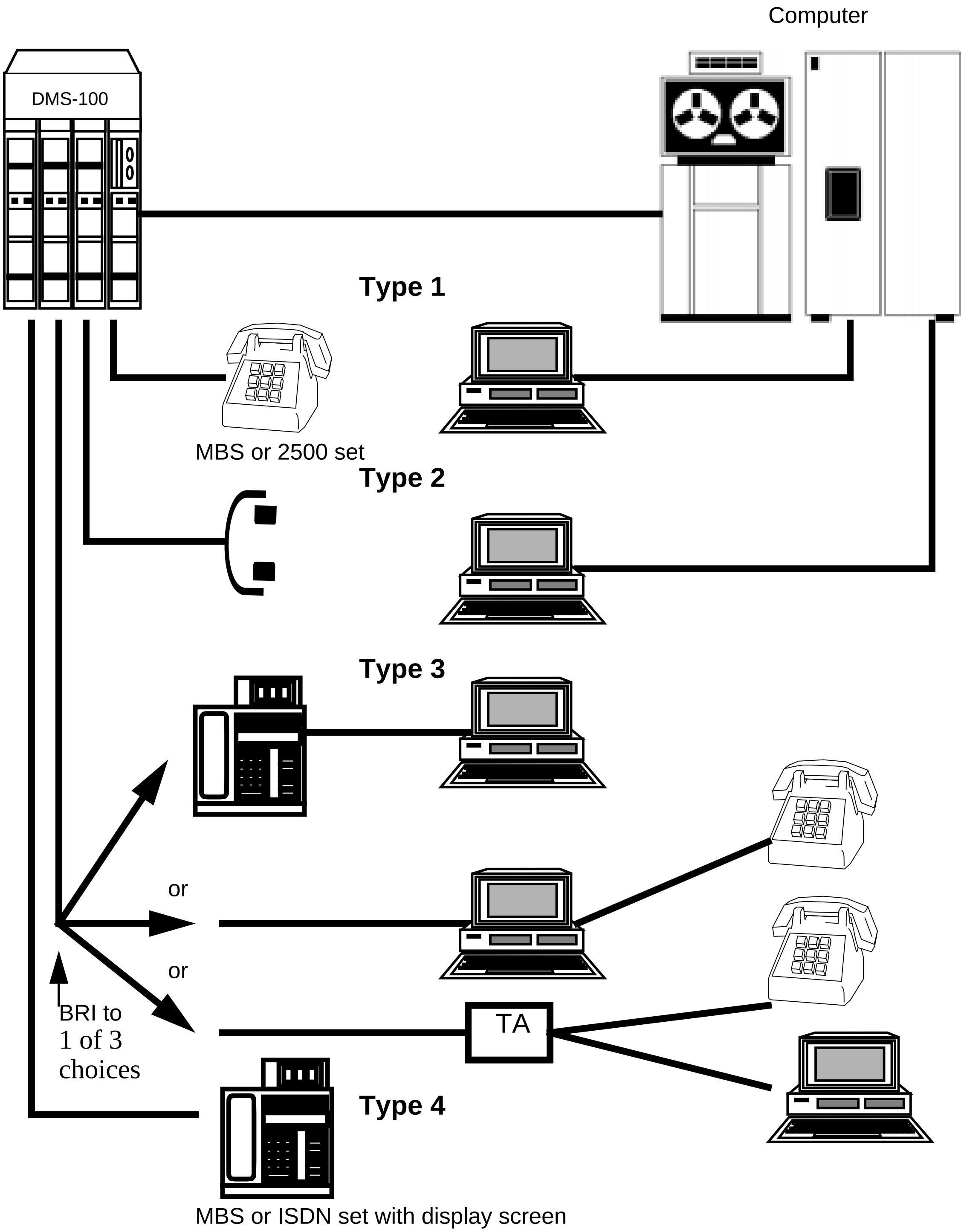
Type 2
Depicts the case where an agent or operator's voice terminal equipment consists only of a headset and will use a host-attached data terminal for all keypad functions. The data terminal attaches to the host computer through a RS-232, LAN, or other suitable physical host connection. This illustrates the

case where all the agent's call control requests go to the switch through a CompuCALL link.

Type 3
Depicts the case where an ISDN Basic Rate Interface link provides physical connectivity to both the voice and data terminals. The data connectivity can be either circuit or packet switched through the switch to the host computer. There are three connections that could fit this type. The first could have the data terminal connected to an ISDN set by a RS-232 cable. The second is for a workstation equipped with a ISDN terminal adapter (TA) internal card and a voice terminal connected to the TA. The third has an external TA with both the voice and data terminals connected to the TA. CompuCALL messages communicate between the switch and the host to coordinate the voice and data services to the user.

Type 4
Depicts a future development where an ISDN Basic Rate or an MBS is used with a screen display. In this case, the set uses CompuCALL to perform functions such as data base queries for number-to-name translation.

Figure 1 Voice and Data Configurations with CompuCALL Link



Application Example

One example of an application utilizing a switch-computer interface is in telemarketing. While most telemarketing implementations consider voice and data requirements separately, CompuCALL combines the operation of voice and data services. This is made possible through the use of a communication link between the host computer and the switch. The resulting association allows the coordination of functions provided by a computer and a switch, resulting in an overall operational improvement and enhanced flexibility of the application to the end-user.

In this example, multiple interactions are possible between

- A computer and switch to manage coordination of services.
- An agent's work station and a switch to request simultaneous voice and data call transfer, or conference to another agent.
- An agent's work station and the host computer.

In a typical application Calling Line Identification (CLID) service helps a computer retrieve a customer profile and establish a data session with the agent's data terminal. At the same time, the switch can establish the voice call. The result is a personalized service to the customer and a fast transaction. This increases the number of transactions an agent is able to handle within a given period.

The example described is one of many applications supported by data and voice coordination services using CompuCALL.

Application Services

An Application Service in CompuCALL is a grouping of options for a specific application. These options form the basis for building a CompuCALL service. The Application Service Options described are available.

> ***Note:*** CompuCALL Options are a set of CompuCALL functionalities offered by the operating company to a CompuCALL customer. From the available CompuCALL Options the customer can define with the operating company, a specific CompuCALL Application Service at subscription time.

Meridian ACD CompuCALL Options

Meridian ACD CompuCALL Options enhance the existing DMS-100 Meridian Automatic Call Distribution (MACD) by allowing customers to configure incoming and outgoing ACD call handling systems. The main application of Meridian ACD CompuCALL Options is coordinated voice and data delivery to an ACD agent for handling incoming telemarketing calls.

MDC CompuCALL Options

Meridian Digital Centrex (MDC) CompuCALL Options provide application services to non-ACD lines. A Call Center can now consist of ACD and non-ACD lines. The main application of MDC CompuCALL Options is coordinated voice and data delivery to an end-user work station for handling incoming calls.

Document Update Procedure

This document tracks change in the DMS-100 switch design. The Introduction contains the most recent release information. The Publication History, provides the history of changes to the remaining Chapters, Sections, and Appendix of the document.

Document Structure

The structure is seven chapters each containing sections. The sections contain both the underlying protocol needed to support the application interface and a description of the interface using an architectural model design. The Application Services are defined with the required procedures to map the services onto the network. The following sections describe the chapters.

Chapter 1.0 Architecture Model description

This chapter provides general information on the OSI model adhered to by CompuCALL and CompuCALL services offered by the operating company.

1.1 OSI Reference Model Description

Section 1.1 provides the relationship of CompuCALL functions to the OSI reference model. The reference model is an overview, to better understand CompuCALL and is an implementation requirement.

1.3 Generic Concept of CompuCALL

Section 1.3 defines the switch interchange points to connect a customer’s computer. It also provides the process for creating a customer’s CompuCALL service by the operating company.

1.4 Functional Architecture

Section 1.4 defines the CompuCALL application layer protocol formats. There is only use of ROSE components.

Chapter 2.0 CompuCALL Transports

The CompuCALL architecture permits a client application using a CompuCALL service to operate over a variety of transports. This chapter describes how the use of CompuCALL transports on the DMS-100 switch.

2.1X.25 Transport

This section provides a description of the X.25 services used by CompuCALL and the mechanisms used for the CompuCALL network layer protocol to work with the application layer protocol.

2.11 TCP/IP Transport

This section provides a description of the TCP/IP transport services. The transport provides TCP/IP connectivity between the captive office and the customer site through the Ethernet Interface Unit (EIU) in the switch and an Internet Service Provider.

Chapter 3.0 CompuCALL Common Application Service Options

This chapter contains application services which are common to all applications. Common application service options is used in conjunction with the application services described in Chapter 4.

3.1 CompuCALL Session Management Options

Section 1 provides a detailed description of the Session Management Service options and the mechanisms used for the CompuCALL application layer protocol to work with the network layer protocol.

3.4 CompuCALL OA&M Options

Provides a detailed description of the OA&M application service functions.

Chapter 4.0 CompuCALL Application Service Options

This chapter contains application service options. The operating companies use application service options and their customers to define CompuCALL based application services.

4.1 CompuCALL Application Services

Section 4.1 provides a detailed description of the Meridian CompuCALL Application Service Options.

4.2 CompuCALL Application Service Functions

Section 4.2 describes application service functions in detail. It describes the network/host information flow and further defines the behavior of the functions by the use of Call Walk-Through Procedure diagrams.

4.3 Interactions, Restrictions, and Limitations

In section 4.3 describes interactions between messages, features, and phone sets as well as the limitations of CompuCALL functionality.

Chapter 5.0 CompuCALL Application Service Parameter Definitions

This chapter contains application service options available in this release. These are used by telephone companies and their customers to define CompuCALL based application services.

5.1 CompuCALL Application Services

Section 5.1 provides a detailed description of the Meridian CompuCALL Application Service Options.

Chapter 6.0 CompuCALL Call Walkthrough Procedures

This chapter contains typical Meridian CompuCALL scenarios and associated messages generated by the switch and host. This is not an extensive list of Meridian CompuCALL message sequences, but a reference for the more common or exceptional scenarios.

Meridian ACD CompuCALL Options: No Call Redirection

This section presents the Meridian ACD CompuCALL scenarios.

Meridian ACD CompuCALL Options: Call Redirection

This section presents the Meridian ACD CompuCALL scenarios.

Meridian ACD CompuCALL Options: Third Party Call Control

This section presents the Meridian ACD CompuCALL scenarios.

MDC CompuCALL Options

This section shows the MDC CompuCALL scenarios.

Meridian ACD CompuCALL Options: Third Party Agent Control

This section presents Meridian ACD Agent CompuCALL scenarios.

MDC/RESCompuCALL Options

This section presents MDC/RES CompuCALL scenarios.

Chapter 7.0 CompuCALL Service Model

This Chapter describes the CompuCALL Service Model to provide a framework for the service functions and customer applications in Section 4.1, “Application service overview,” on page 121 and CompuCALL session management procedures in Section 3.1, “Session management options overview,” on page 83. The Service Model is described in terms of

- The intent of the CompuCALL options.
- Key basic concepts.
- Key service design principles governing CompuCALL service-related procedures including, where relevant, exceptions to those principles.

1.0 Architecture model description

1.1 OSI reference model description

The CompuCALL reference model shows how services can be built using CompuCALL capabilities. Please refer to Figure 2, “OSI Reference Model used by CompuCALL,” on page 38. By illustrating how capabilities are requested and provided, this structured layered description depicts the relationship of CompuCALL options to the OSI reference model. The reference model is described as a CompuCALL overview, not as an implementation requirement.

1.2 Objectives

The CompuCALL model contains the following:

- A platform for developing applications using a functional view independent of communication interfaces and protocols providing access to CompuCALL services.
- A description of how the applications communicate using the application interface and the communication layers and the services offered by each communication layer.
- A way to allow alternate communication stacks (e.g., OSI) without making the existing application obsolete or incompatible.

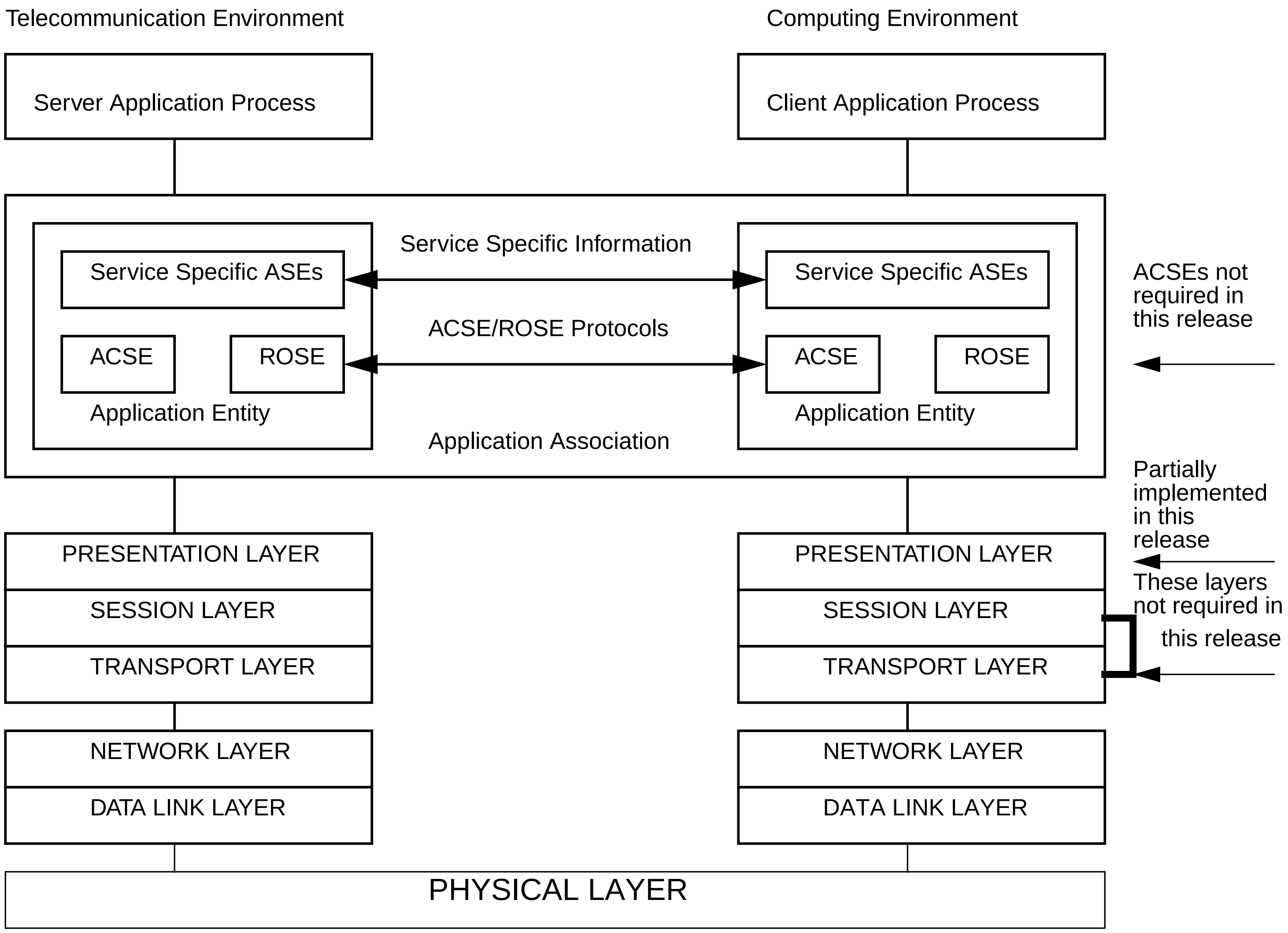
1.3 Generic concept of CompuCALL

The generic model of CompuCALL is shown in Figure 2, “OSI Reference Model used by CompuCALL” on page 38. However, not all layers transmit CompuCALL messages. Layers 4 and 5 are bypassed and layer 6 is partially implemented with CompuCALL.

CompuCALL uses Open Systems Interconnection (OSI) principles to exchange service specific information between a client application process in a computing environment and a peer-server application process in the telecommunication environment.

The CompuCALL model in Figure 2, “OSI Reference Model used by CompuCALL” on page 38, represents the exchange of service specific information between a pair of Application Processes (APs) in terms of the communication between two layer 7 Application Entities (AEs) supported by their underlying layers. The interface between each layer is called a service interface, a general OSI term referring to a set of service primitives representing information exchanged between layers. Service Primitives are a means of one-way information communication between layers. This document covers messages exchanged between layer 7 AEs. The document does not cover Service Primitives in specific implementations in the computing environment.

Figure 2 OSI Reference Model used by CompuCALL



1.3.1 Nodes

A node is a network entity such as a general purpose computer, a central office switch, or an intelligent terminal. A CompuCALL node carries or requests CompuCALL services.

1.3.2 Client and server

The CompuCALL model refers client and server. The "client" or "host" is the element in a node requesting CompuCALL services. The "server" provides the CompuCALL services. In OSI terminology, these two elements are Application Processes (AP). They process information for a particular application. There is no protocol restriction on the number of "clients" on any node or number of services a "client" may access. The appropriate server processes each request and responds as necessary. These requests are from multiple clients. The server may contain application-specific options that it can activate depending on service subscription. Once a client's service is established, these options are selected based on CompuCALL subscription parameters in DMS-100 switch.

1.3.3 Application service options, functions, and parameters

CompuCALL Application Service options provide information to build a CompuCALL service. These sets of options determine the type of server. Each option specifies a function and its parameters.

There are various Application Service options. For example, the Session Management Service LogIn option identifies the first function requested in the "Client/Server" interaction. Other options identify functions which can take effect after completing the initiating function. Session Management Service options identify LogOff functions. These functions are requested and performed last to terminate the interaction.

1.4 Functional architecture

The protocol structure of networked applications involves a number of entities working together consistently with the requirements of the application.

This section addresses application processing from the user's viewpoint. Those views are depicted in an abstract functional architecture model. This abstract model serves as a basis for the following sections on architecture and protocols. Section 1.5, "Application layer designer guide," on page 43. The network service provides the link to establish, maintain and release network connections. The functions required by CompuCALL in the network layer are described in Section 2.1, "X.25 Transport," on page 71.

Computer applications for users include computer programs such as accounting and inventory control. Other types of computer applications are switch and computer-based address telemarketing services. They include airline reservations or telephone support centers involving both voice and data services. Computer vendors, system integrators, and others create these general applications.

The applications are divided into functions which support the following applications:

- Managing a voice service for call center operation
- Coordinating voice and data services so the agent’s data terminal screen receives calling party information while agent's associated voice service is established
- Allowing services provided to the agent's voice terminal to be activated by the agent's associated data terminal.
- Screening and disposition of calls to centrex users, call center agents, or residential subscribers.

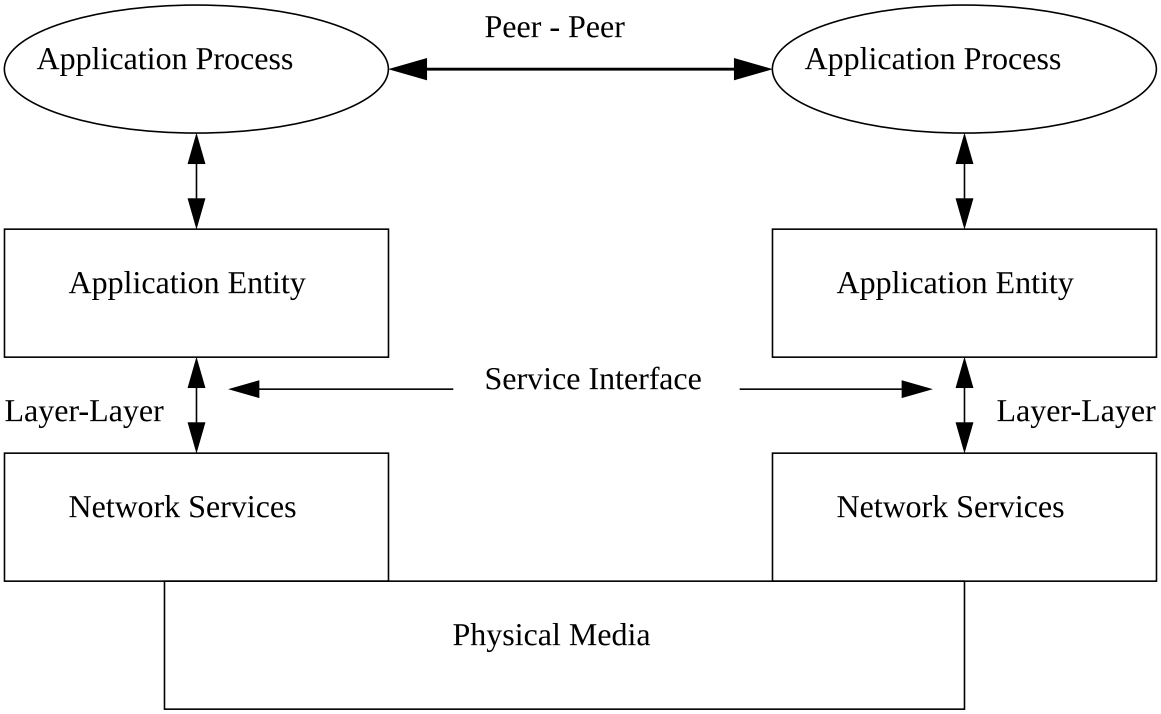
Associating these switch-based services, or options, with a computer application provides users with enhanced voice and data services.
The first step in defining CompuCALL functional architecture is to describe the application structure. In OSI terms, the application is called an Application Process (AP). The AP is an element of a real open system, as defined in CCITT X.200. It takes part in the processing of information. The Application Process sends and/or receives the information in another real open system.

There are two levels of communication in the AP. The first level addresses the context of the application on a peer level. The information is only understood and processed by peer APs. The other level addresses the layer-to-layer communications. This refers to the layers concept. A layer requests service from the layer below and provides service to the layer above. Through a service interface the AP request services through primitives from the layer below. This service interface provides a way of modeling the information (primitives) exchanged between the AP and the underlying service entity.

Figure 3 “CompuCALL model overview” on page 41 illustrates a simple view of two Application Processes and their associated underlying entities.

The Application Entity receives the services and communicates with the service provider through another service interface. In this example, applications can communicate with each other. There is no concern about details of the underlying network services or the type of protocols used. This is true assuming the underlying network service protocols support the requirements of the application.

Figure 3 CompuCALL model overview



This representation assumes each entity exists in a separate real open system. Additionally, from the overall view of the application, linking to the associated Application Entity is independent of the actual location of its physical system.

1.4.1 Application layer elements

ISO 9495 describes the application layer structure. The structure consists of a number of service elements to support applications, or APs. These service elements provide functions such as association (connections) control and remote operations. In addition, a more specific service element is designed to provide a particular set of functions to support an AP.

An AP is an element of a real open system which takes part in processing information exchanged between an AP in another real open system. The AP-to-AP communications is through an Application Entity (AE) in the application layer.

The application layer consists of one or more AEs providing services to an AP through a service interface. Each of the AEs are then addressable to the underlying service-providing entity by a presentation layer address.

The Association Control Service Element (ACSE) and the Remote Operations Service Element (ROSE) cover the application association control and remote operation functions. CCITT Recommendations X.217 and X.227 describe ACSE. CCITT Recommendations X.219 and X.229 describe ROSE.

In this architecture, ACSE and ROSE are presented as two standard ASEs. The architecture also allows an implicit application association, so previous agreements or subscription parameters can communicate without explicit association control. This releases the requirement for an explicit ACSE functionality. The DMS-100 CompuCALL service subscription parameters define that one application service is provided. CompuCALL does not require ACSE.

1.4.1.1 ASE

“ASE” is an OSI Application Layer term that applies to a set of specific services to support an AP. The ASEs are combined with the ROSE elements and modeled into an AE. This provides services to the AP through an AE service interface. DMS-100 CompuCALL service subscription parameters define one application service provided for CompuCALL.

1.4.1.2 ROSE function

The Remote Operation Service Element (ROSE) supports interactive communication between two application processes. ROSE is modeled as a request-reply interaction between an invoking and a performing AE.

The ROSE-Provider (the ROSE ASE) offers the following five services to the ROSE-User (the ASE requesting the service):

- The RO-INVOKE service enables an invoking AE to request the performing AE to do an operation.
- The RO-RESULT service enables the performing AE to return a successful reply of operation to the invoking AE.
- The RO-ERROR service enables the performing AE to return a negative reply for an unsuccessful operation to the invoking AE.
- The RO-REJECT-U service enables a ROSE-User to reject the request or reply of the other AE if it detects a problem.
- The RO-REJECT-P service informs the ROSE-User when the ROSE-Provider detects a problem in the underlying layers.

The Application Layer protocol formats in CompuCALL are defined in Section 1.11, “Application layer protocol formats,” on page 62.

1.4.2 Presentation layer

Local Concrete Syntax (LCS) is local representation. The presentation layer is partially implemented to provide translation of application data from its local representation. Transfer Syntax is the latter representation.

Different LCSs allow interconnection of systems, and allows the application layers to be independent of the Transfer Syntax to transfer data. The translation between the LCS and the Transfer Syntax follows the Abstract Syntax Notation 1 (ASN.1) encoding rules in accordance with CCITT X.209.

The Transfer Syntax and parts of the ASN.1 specification for understanding the service definitions in this specification are repeated in Section 1.9, “Transfer syntax,” on page 51 and Section 1.10, “Abstract Syntax Notation One (ASN.1),” on page 56.

1.4.3 Session layer

The session layer of the CompuCALL reference model passes messages not directly processed to the transport layer. The same is true for information coming from the transport layer.

1.4.4 Transport layer

The transport layer of the CompuCALL reference model passes unprocessed messages to and from the network layer using the primitives for network connection, disconnection, and data transfer.

1.4.5 Network layer

The network layer allows transmission messages to be encoded. The network layer establishes and terminates the connection to the network.

The network service provides the link to establish, maintain, and release network connections. Section 2.1, “X.25 Transport” on page 71 describes the functions required by CompuCALL in the network layer.

1.5 Application layer designer guide

This section provides background on the application layer service Remote Operations (RO, based on the following CCITT Blue Book (1988) Recommendations):

- X.208 Specification of Abstract Syntax Notation One (ASN.1)
- X.209 Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)
- X.219 Remote Operations: Model, Notation and Service Definition
- X.229 Remote Operations: Protocol Specification

Consult the CCITT Blue Book (1988) Recommendations as the authoritative documents.

1.6 Application layer

CompuCALL uses the principles of Open Systems Interconnection (OSI) to support the exchange of service specific information between a client application process and a peer-server application process. Please refer to Section 1.1, “OSI reference model description,” on page 37.

1.6.1 Structure

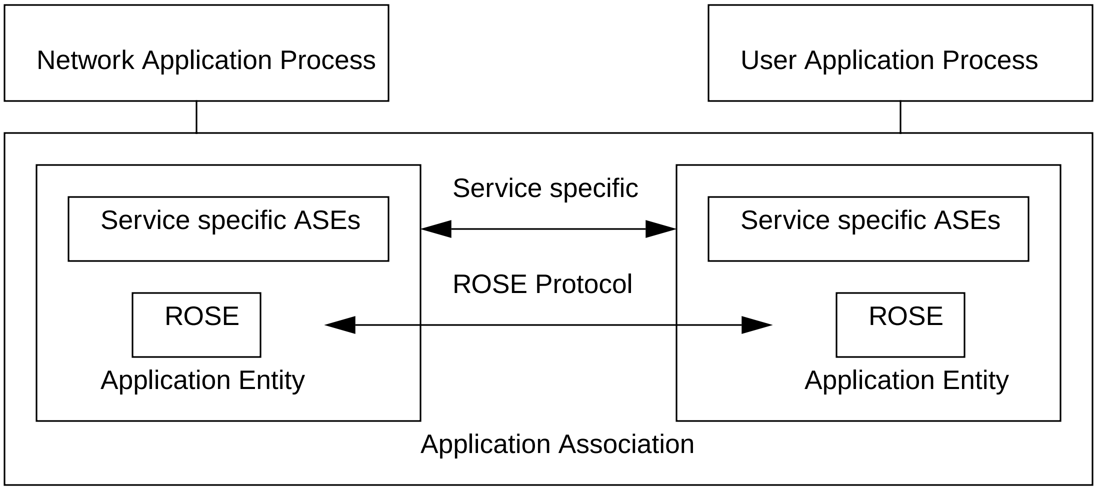
The OSI reference model represents communication between a pair of APs in terms of communication between their AEs. The functionality of an AE is factored into a number of ASEs. The ROSE supports an interactive communication between two application processes.

An application-association is a cooperative relationship between two AEs. It provides the necessary frame of reference between the AEs so they may work effectively. For the application entities to work on an application association, an application context identifies application service elements, options, and other information. Figure 4 shows the structure of the application layer.

The interaction between AEs is described in terms of their use of ASE services. The following ASN.1 macro notation defines the interaction:

> OPERATION and ERROR macros specify the service-specific use of ROSE (see Section 1.8.1, “Macro notation,” on page 48).

Figure 4 Application layer structure



Remote Operations Service Element (ROSE)

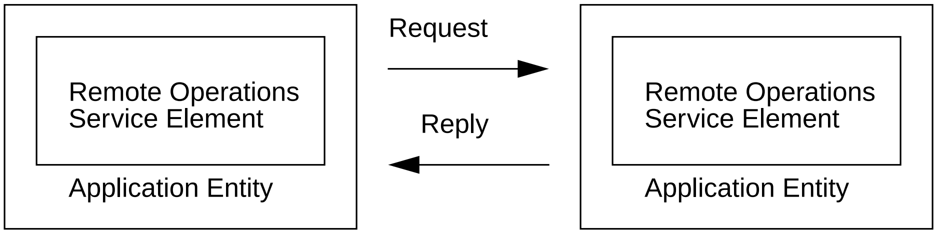
1.6.1.1 General

This section overviews the remote operations service, notation and protocol defined in X.219/X.229. Network Application and User Application are called the Switch Application and Host Application respectively in this document.

1.6.1.2 Remote operation model

The Remote Operation (RO) service supports an interactive communication between two Application Processes. This is modeled as a request and reply interaction between the two AEs. The AEs represent the functionality for providing open communication between application processes. The ROSE represents functionality which provides the open communication for a request/reply interaction (see Figure 5).

Figure 5 ROSE request/reply interaction


Remote Operations Service Element
Application Entity
Request
Reply
Remote Operations Service Element
Application Entity

One AE invokes operations, and the other AE performs operations. Operations are classified according to the expected outcome:

- If success or failure, a result reply is returned if the operation is successful. An error reply is returned if the operation is unsuccessful.
- If failure only, no reply is returned if the operation is successful. An error reply is returned if the operation is unsuccessful.
- If success only, a result reply is returned if the operation is successful. No reply is returned if the operation is unsuccessful
- Not at all, neither a result nor an error reply is returned, whether the operation was successful or not.

Also, operations are classified according to two possible operation modes:

- Synchronous, in which the invoker requires a reply from the performer before invoking another operation.
- Asynchronous, in which the invoker continues to invoke further operations without waiting for a reply.

The Operation Classes are:

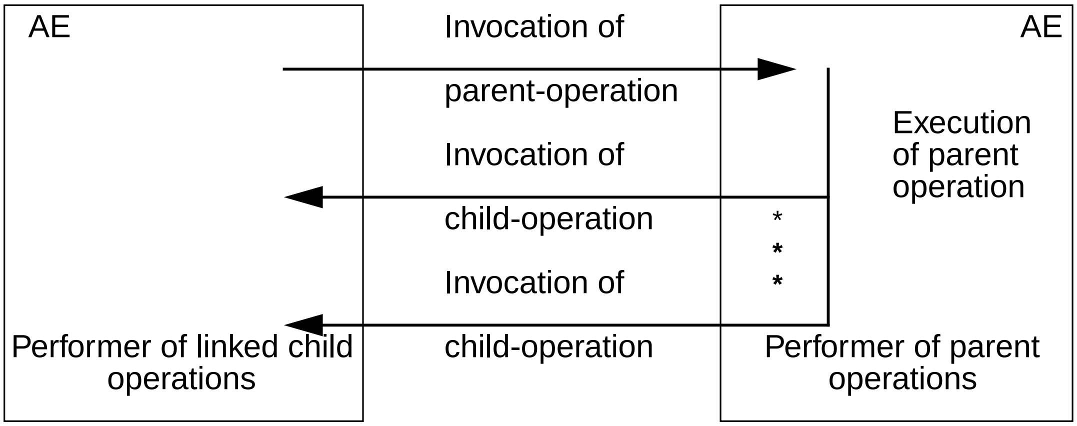
- Operation Class 1: Synchronous, reporting success or failure (result or error)
- Operation Class 2: Asynchronous, reporting success or failure (result or error)

- Operation Class 3: Asynchronous, reporting failure (error)
- Operation Class 4: Asynchronous, reporting success (result)
- Operation Class 5: Asynchronous, outcome not reported

CompuCALL uses Operation Class 1, 2 and 5. Classes 3 and 4 are not required.

It is useful to group operations into a set of linked-operations performed by one parent-operation and one or more child-operations. The performer of the parent-operation may invoke none, one, or more child-operations during the execution of the parent-operation. The invoker of the parent-operation is the performer of the child-operations. A child-operation may be a parent-operation of another set of linked-operations (Figure 6). Section 1.8.1, “Macro notation,” on page 48 defines Remote Operation Class in the form of OPERATION macro notation.

Figure 6 ROSE linked operation


AE
Invocation of
parent-operation
Invocation of
child-operation
Invocation of
child-operation
AE
Execution of parent operation
*
*
*
Performer of linked child operations
Performer of parent operations

1.6.2 Remote operation service

The Remote Operation Service Element (ROSE) supports an interactive communication between two application processes. It is modeled as a request-reply interaction between an invoking and a performing AE.

ROSE provides five services:

- RO-INVOKE lets an invoking AE request the performing AE to do an operation.
- RO-RESULT lets the performing AE return the positive reply of a successful operation to the invoking AE.
- RO-ERROR lets the performing AE return the negative reply of an unsuccessful operation to the invoking AE.
- RO-REJECT-U lets one AE reject the request or reply of the other AE if the ROSE-user detects a problem.
- RO-REJECT-P lets the ROSE-provider inform the ROSE-user of a problem.

1.6.2.1 Remote operation protocol

One AE (the invoker) uses the invocation procedure to request the other AE (the performer) to perform an operation. It supports the RO-INVOKE service. The invocation procedure uses the following components and service specific fields defined by the OPERATION macro notation (see Section 1.8.1.1, “Operation macro notation” on page 49)

- Invoke component
  - — operation value
  - — argument

One AE (the performer) uses the return-result procedure to request that the result of a successful operation transfer to the other AE (the invoker). It supports the RO-RESULT service. The return-result procedure uses the following component and service specific field defined by the OPERATION macro notation (see Section 1.8.1.1, “Operation macro notation,” on page 49))

- Return result component (result)

AE (performer) uses the return-error procedure to request transfer of error information of an unsuccessful operation to the other AE (invoker). It supports the RO-ERROR service. The return-error procedure uses the following component and service specific field defined by the ERROR macro notation (see Section 1.8.1.2, “Error macro notation,” on page 49)

- Return error component
  - — error value
  - — parameter

One AE uses the user-reject procedure to reject the request (invocation) or reply (result or error) of the other AE. It supports the RO-REJECT-U service. The user reject procedure uses the reject component to identify the invoke, return-result, or return-error problem. It has no service specific information.

The underlying protocol uses the provider reject procedure to indicate to the originating AE that it has detected a problem. It supports the RO-REJECT-P service. The provider reject procedure uses the reject component to identify the reject problem. It has no service specific information.

1.7 Application layer notation

The application context macro notation uses the following ASN.1 macro notations specified in X.219: OPERATION and ERROR macros (Section 1.8).

This macro notation defines the service specific information for the generic application layer protocols, remote operations.

Section 7.4, “CompuCALL service design principles,” on page 718 describes the ASN.1 notation for specifying data types within the macro notation. Section 1.11, “Application layer protocol formats,” on page 62 describes the transfer syntax to encode these data types in the application protocol components.

1.8 Remote operations macros

This specification uses the macro format described in X.219 to define service operations which use RO procedures. To relate these macro descriptions to actual component encoding, see the remainder of this section, which provides the notation and the encoding rules (Section 1.9) for each data type. “Facility” describes the structure of each type of RO component.

1.8.1Macro notation

The following operation and error macro notations describe the 5 classes of remote operations (Section 1.6.1.2, “Remote operation model,” on page 27). These are based on the definitions in X.219.

ARGUMENT, RESULT, and PARAMETER types in operation and error macros can be mandatory or optional. The key word OPTIONAL after the type indicates an optional type. The key word DEFAULT followed by a value and the type indicates a mandatory type with a default value.

1.8.1.1 Operation macro notation

Figure 7 Operation macro notation

```
Operationname
              OPERATION
              ARGUMENT                          ArgumentType   \ empty
              RESULT                            ResultType     \ empty
              ERRORS                            {ErrorList}    \ empty
              LINKED                            {Linked Oplist} \ empty

              ::=operation value
```

OPERATION, ARGUMENT, RESULT, ERRORS, and LINKED are key words used to structure operation macros. By using the operation value the operation macro is identified in the Invoke component.

The argument data type is operation specific and used by the Invoke component. It is defined on a per-operation basis using ASN.1.

The ResultType is the label of the result data type which is used in the Return result component and is operation specific. It is defined on a per-operation basis using ASN.1.

The “ErrorList” is a list of “errorname” labels. The error macro notation defines each label and gives them an “errorname”(see Figure 8).

The “LinkedOpList” is a list of “operationname” labels. The operation macro notation defines each label.

1.8.1.2 Error macro notation

Figure 8 Error macro notation

```
errorname     ERROR
              PARAMETER        ParameterType| empty

              ::=errorvalue
```

The keywords to structure error macros are ERROR and PARAMETER. The “errorname” is the label of the error macro. The error component identifies the label using the “errorvalue.” “ParameterType” is the label of the parameter data type, which is used in the Return error component and is error-specific. It is defined on a per-error basis using ASN.1.

1.8.2 Class 1 and class 2 operations

Class 1 and Class 2 operations always require a response whether the operation succeeds or fails. There is an optional argument. Class 1 operations are synchronous, and Class 2 operations are asynchronous.

```
operation Example 12   OPERATION
          ARGUMENT  ArgType12
          RESULT        ResultType12
          ERRORS        {error1,error2}
          ::= 9

error1    ERROR
          PARAMETER ParameterType1
          ::= 1


error2    ERROR
          ::= 2
```

In the above example, the operation name is "operationExample12" and the operation value, coded as an integer in an Invoke component, is 9. The type "ArgType12" defines the argument.

If the operation is successful, it is reported in a Return result component. The Return result component has a mandatory parameter defined by the type "ResultType12".

There are two possible error responses if the operation fails. The first, "error1", has value 1 and has one parameter, "ParameterType1". The second error, "error2", has value 2 and has no parameter. The error result is returned in a Return error component.

1.8.3 Class 5 operation

Class 5 operations do not require any response. There is a mandatory argument, ArgumentType5.

```
operation Example 51 OPERATION
                     ARGUMENT  ArgumentType5
                     ::= 17
```

In the above example, the operation name is "operationExample51" and the operation value, coded as an integer in an Invoke component, is 17. There is an argument associated with this operation, defined by the type "ArgumentType5".

1.8.4 Linked operation

The example below shows how linked operations are defined in the macro notation. The parent-operation, “parentop12”, has one child-operation, “operationExample51”.

```
parentop12     OPERATION
        ARGUMENTArgumentType12
        RESULT   ResultType12
        ERRORS   {error1,error2}
        LINKED   {operationExample51}
        ::= 12
```

1.9 Transfer syntax

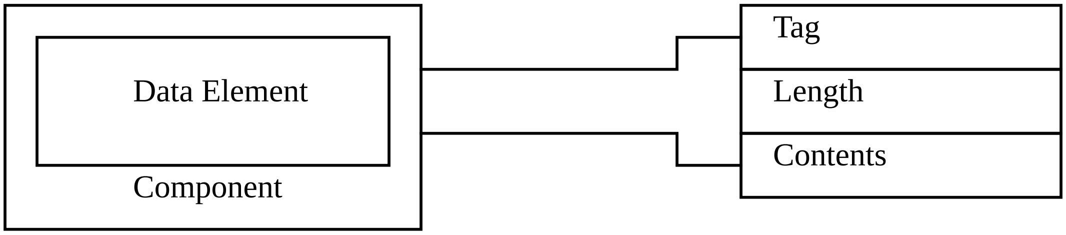
1.9.1 General description of component encoding rules

Application Layer Protocol Format uses the following description of general component encoding rules. See Section 1.11, “Application layer protocol formats,” on page 62.

1.9.1.1 General component structure

Each component consists of data elements, each of which has the same structure. A data element consists of three fields, which always appear in the following order. The tag distinguishes one type from another and governs the interpretation of the contents. The length specifies the length of the contents. The contents is the substance of the data element containing the primary information the data element conveys. Figure 9 shows an overview of a component and a data element.

Figure 9 Structure of a component and data element



a) Component b) Data Element

Each field is coded using one or more octets. Figure 10 shows octet labels. The first octet is the first transmitted. Figure 11 shows the labels of bits in an octet. Bit 1 is the least significant and the first transmitted.

Figure 10 Octet labeling scheme

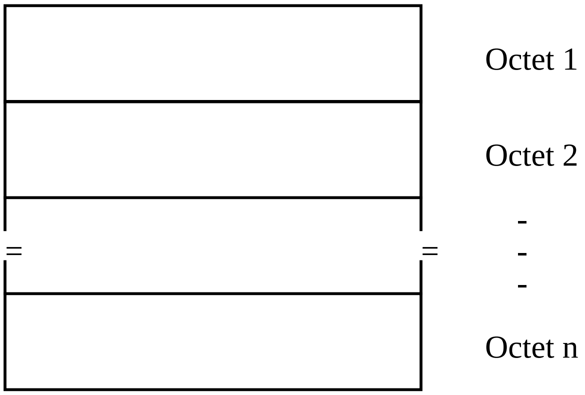


Figure 11 Bit labeling scheme

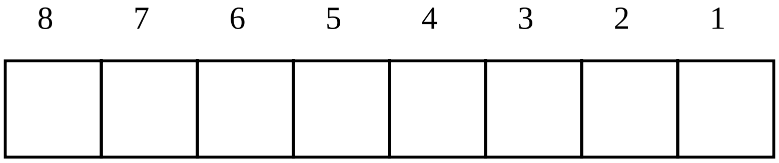
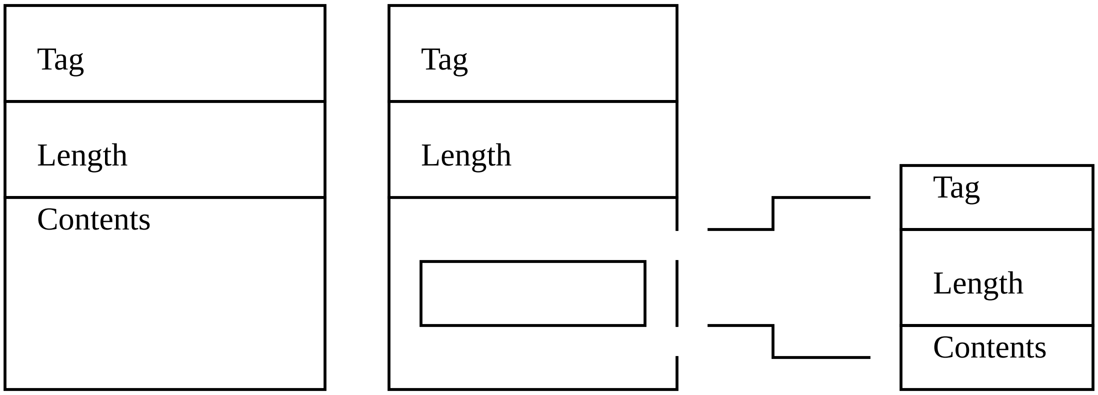


Figure 12 shows the contents of each data element as one value (primitive) or one or more data elements (constructor).

Figure 12 Types of contents



a) Primitive b) Constructor

1.9.1.2 Tag

A data element is first interpreted according to its position within the syntax of the messages. The tag distinguishes one data element from another and governs the interpretation of the contents. It is one or more octets in length. Figure 13 shows the tag format as “class”, “form” and “tag code”.

Figure 13 Format of tag

8 7	6	5 4 3 2 1
Class	Form	Tag Code (*)

The tag code may be extended to the following octets as discussed in “Tag code” on page 54.

The “Class”, “Form”, and “Tag” codes are explained below.

1.9.1.3 Tag class

All tags use the two most significant bits (8 and 7) to indicate the tag class. Figure 14 shows the code for these bits.

Figure 14 Coding of tag class

Class	Coding (bits 8 and 7)
Universal	00
Application-wide	01
Context-specific	10
Private	11

The universal class is used for tags exclusively standardized in Recommendation X.209 and are application-independent types. Universal tags may be used anywhere universal data element types are used. The universal class applies across all CCITT Recommendations, Recommendation Q.932 Facility Information Elements, CCITT Signaling System No 7 ASEs (Application Service Elements), X.400 MHS (Message Handling System), X.500 Directory Services, etc.

The application-wide class is used for data elements standardized across all applications related to a theme such as CCITT Q.932 Facility information element procedures.

The context-specific class is used for data elements specified within the context of the next higher construction and take into account the sequence of other data elements within the same construction. This class may be used for tags in a construction, and the tags may be reused in any other construction.

The private class is reserved for data elements specific to a nation, a network, or a private user, and apply across all applications.

1.9.1.4 Form of the data element

Bit 6 indicates if the data element is “primitive” or a “constructor”(Figure 15). A primitive element’s structure is atomic (one value only). A constructor element is one whose content is one or more data elements which may themselves be constructor elements. Figure 15 shows both forms of elements.

Figure 15 Coding of element form

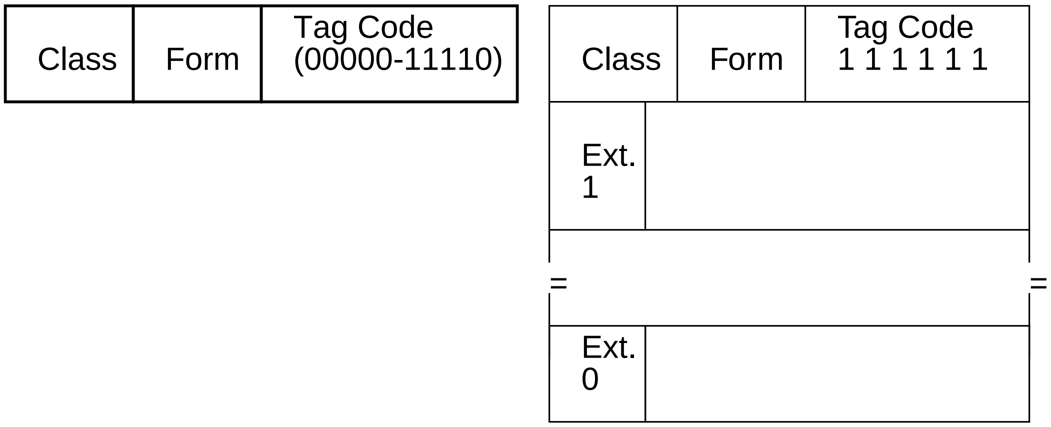
Element Form	Coding (bit 6)
Primitive	0
Constructor	1

1.9.1.5 Tag code

Bits 1 to 5 of the first octet of the tag, plus any extension octets, represent a tag code that distinguishes one element type from another of the same class. Tag codes in the range 00000 to 11110 (0 to 30 decimal) are provided in one octet.

The extension mechanism is to code bits 1 to 5 of the first octet as 11111. Bit 8 of the following octet is an extension indication. If bit 8 of the extension octet is set to 0, no further octets for this tag are used. If bit 8 is set to 1, the following octet is also used for extension of the tag code. The resultant tag consists of bits 1 to 7 of each extension octet with bit 7 of the first extension octet being most significant, and bit 1 of the last extension octet being least significant. Tag code 1F (decimal 31) is encoded as 0011111 in bits 7 to 1 of a single extension octet. Higher tag codes continue from this point using the minimum possible number of extension octets.

Figure 16 Format of the tag code



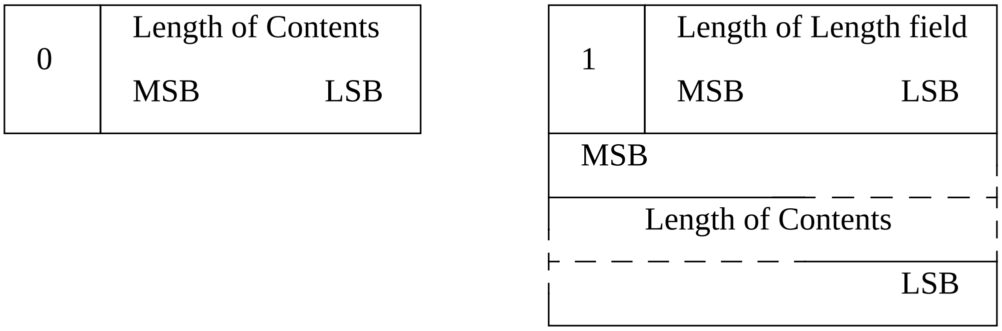
a) One Octet Format b) Extended Format

1.9.1.6 Length of the contents

The length of the contents indicates the number of octets in the contents. The length does not include the tag nor the length of the length octets. The length of the contents uses the short or long form. If the length is less than 128 octets, the short form is used. In the short form, bit 8 is coded 0, and the length is encoded as an unsigned binary number using bits 1 to 7.

If the length of the contents is greater than 127 octets, then the long form of the length of the contents is used. The long form length of the length field is from 2 to 128 octets long. Bit 8 of the first octet is coded 1, and bits 1 to 7 of the first octet encode a number as an unsigned binary whose value is one less than the total size in octets of the length field. Bit 7 of this first octet is the MSB (Most Significant Bit) and bit 1 of this first octet is the unsigned LSB (Least Significant Bit). The second octet through the last octet encode a number as an unsigned binary whose value is the actual length of the contents (in octets). Bit 8 of the second octet is the MSB and bit 1 of the last octet is the LSB. This binary number should be encoded in the fewest possible octets, with no leading octets having the value 0.

Figure 17 Format of length field



a) Short Form

b) Long Form

1.9.1.7 Contents

The contents are the substance of the data element and contain the information the data element is intended to convey. Its length is variable, but always an integral number of octets. The contents are interpreted in a type-dependent manner, that is, according to the tag value.

1.10 Abstract Syntax Notation One (ASN.1)

ASN.1 is a formal notation for specifying the data types and values used in an application layer protocol, such as Remote Operations. ASN.1 is used to describe the data types and values associated with the generic components and those associated with a specific service through the use of macros. The entirety of the ASN.1 specification is not repeated here, but only those parts that are relevant to understanding the service definitions given in this specification which have been described with ASN.1.

1.10.1 Data types

Data Types defined in the operation macros (for example, ArgumentType5 and ResultType12 in the operation examples) are defined following the main operation definition, using the following general format.

```
DataTypeLabel ::= PredefinedValue | TaggedValue
```

The left-hand side of the definition symbol “::=” is the label of the data type being defined. The right-hand side is the data type definition, which may be either a predefined value or a tagged value.

Predefined values belong to the universal class and use the tags defined in X.208. Tagged values belong to the application, context, and private classes and are defined in this specification.

The tag is composed of a class and an integer, which are enclosed in brackets "[]" (for example, [PRIVATE 1]). If the class is omitted, it defaults to context (for example [5]). The tag is not used for predefined values. The predefined values used in this specification are specified using key words, which are shown in the left-hand column of Figure 18.

Figure 18 Universal tags used in this specification

PRE-DEFINED VALUE	TAG (ASN.1)	TAG (BINARY)8 8 7 6 5 4 3 2 1	TAG(hex)
INTEGER	2UNIVERSAL	0 0 0 0 0 0 1 0	02
OCTET STRING	4UNIVERSAL	0 0 0 0 0 1 0 0	04
ENUMERATED	10UNIVERSAL	0 0 0 0 1 0 1 0	0A
SEQUENCE	16UNIVERSAL	0 0 1 1 0 0 0 0	30
SEQUENCE OF SET	17UNIVERSAL	0 0 1 1 0 0 0 1	31
SET OF IA5STRING	22UNIVERSAL	0 0 0 1 0 1 1 0	16

If the term IMPLICIT is used between the tag and the value in the type definition, the type will be a constructor or a primitive depending on the base encoding of the value. The base encoding of a constructor is tag, length, and contents. The base encoding of a primitive is contents (no tag or length). If IMPLICIT is not used, the tag is a constructor with base encoding of tag, length, and contents.

To illustrate the coding rules, for translating between ASN.1 and data element octets, examples are shown in Figure 19. The coding for the IA5String "Jones" is shown for each type. An explanation of how each type is coded follows:

Type1::=IA5String
Type2::=[APPLICATION 3] IMPLICIT Type1
Type3::=[2] Type2
Type4::=[APPLICATION 7] IMPLICIT Type3
Type5::=[2] IMPLICIT Type2

1 Type1 is a universal class primitive for the IA5String "Jones".

2 The tag for Type2 is application class, code 3. The tag and length for Type1 are not used since the value is implicit. Only the assigned tag distinguishes Type2 from Type1.

3 The tag for Type3 is context class, code 2. Since the value of Type3 is explicit, its contents are the entirety of Type2 (tag, length, contents). The explicit value results in Type3 being a constructor.

4 The tag for Type4 is application class, code 7. The tag and length of Type3 are not used since the value is implicit. The contents of Type3, and now the contents of Type4, is Type2. Since Type3 is explicit in Type2, therefore Type4 is a constructor.

5 The tag for Type5 is context class, code2. The tag and length for Type2 are not used since the value is implicit. The contents of Type2 is the IA5String “Jones”, which is now the contents of Type5. Type5 is a primitive.

Figure 19 Coding of example ASN.1 types

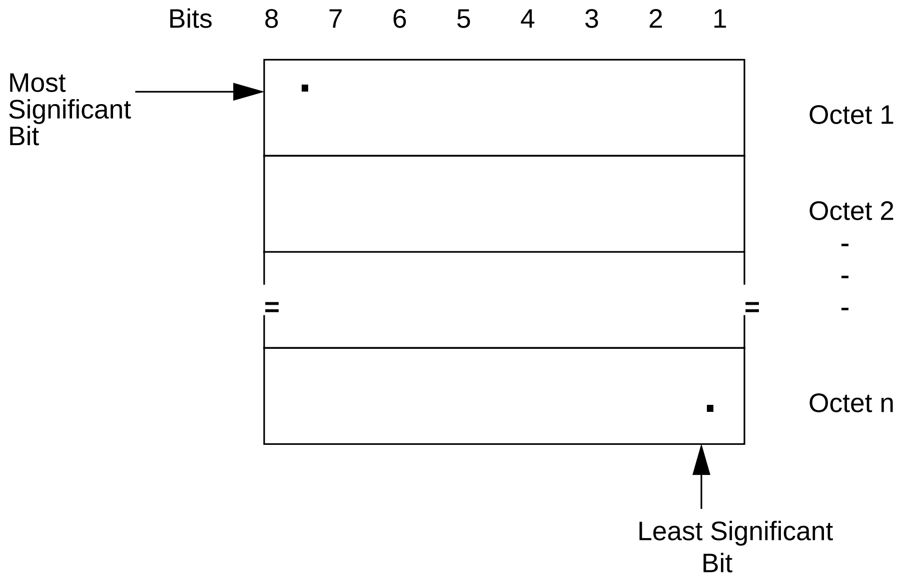
	TAG	LENGTH	CONTENTS
Type 1	16	05	4A6F6E6573
Type 2	43	05	4A6F6E6573
Type 3	A2	07	TAG=43 LEN=05 C=4A6F6E6573
Type 4	67	07	TAG=43 LEN=05 C=4A6F6FE6573
Type 5	82	05	4A6F6E6573

1.10.1.1 Integer

An INTEGER type is a primitive which has as its contents one or more octets representing a two's complement binary number equal to the integer value. The integer is interpreted, with descending bit significance, from bits 8 through 1 of octet 1, bits 8 through 1 of octet 2, and so on up to and including the last octet of the contents (see Figure 20). The bits in the first octet of the contents and bit 8 of the second octet shall not all be 1's and shall not all be 0's. This ensures that the integer value is always encoded in the smallest possible number of octets.

> ***Note:*** The value of two's complement binary number is derived by numbering the bits in the contents octets, starting with bit 1 of the last octet as bit zero, and ending the numbering with the bit 8 of the first octet. Each bit is assigned a numerical value of 2N, where N is its position in the above numbering sequence. The value of two's complement binary number is obtained by summing the numerical values assigned to each bit for those bits which are set to one. This excludes bit 8 of the first octet. If that bit is set to one then reduce this value by the numerical value assigned to bit 8.

Figure 20 Integer type content format octet string



1.10.1.2 Octet string

An OCTET STRING type is a primitive which has as its contents a string of zero or more octets. The interpretation of the octets is at the discretion of the application.

1.10.1.3 Enumerated

An ENUMERATED type is a primitive which has as its contents an integer value which is associated with one of the defined range of allowed values. The following is an example of an ENUMERATED type in ASN.1:

WeekDays = {Monday(0), Tuesday(1), Wednesday(2), Thursday(3), Friday,(4)}

When the defined type, WeekDays, is present in a component it will contain the integer associated with the selected weekday for that transaction. An integer value other than the one associated with a name is not valid. The list within the braces "{}" contains one or more items separated by commas. Each item consists of a name, followed by an integer value within parentheses "()"

1.10.1.4 Sequence

A SEQUENCE type is a constructor containing one each of a sequence type. The order of the types in the sequence must follow the order defined in the structure. The following is an example of a SEQUENCE type in ASN.1:

Stype1::= SEQUENCE{name IA5String, age INTEGER OPTIONAL}

The SEQUENCE structure consists of one or more types, separated by commas, and enclosed in braces {}. Each type in the structure can be defined where it is or elsewhere. In the above example, the type “age” is optional.

The following is coding for an Stype1 with the name “Jones” and age 30. All values are in base-16 (hexadecimal)

Figure 21 Coding example for sequences

	TAG	LENGTH	CONTENTS
SType1	30	0A	
name	16	05	4A6F6E6573
age	02	01	1E

1.10.1.5 Sequence of

A SEQUENCE OF type is a constructor which has as its contents a sequence of zero or more instances of a single type. The order of the instances of the type has significance to the application. The following is an example of a SEQUENCE OF type in ASN.1:

ReadingList ::=SEQUENCE OF books

The SEQUENCE OF structure is similar to that for SEQUENCE, but it must have exactly one type. The coding for the type “ReadingList” is similar to that for Stype1, but would have zero, one, or more instances of “Books” in the sequence.

1.10.1.6 Set

A SET type is a constructor which has as its contents one set of each type, in any order. A SET type is identical to a SEQUENCE type except for the lack of rigid order for the types within the structure. The following is an example of a SET type in ASN.1:

Stype3::=SET {name IA5String, age INTEGER OPTIONAL}

The following is the coding of an Stype3 with the name “Jones” and age 30. All values are in base-16 (hexadecimal). Note that the order of the name and age are not the same as shown in the ASN.1 notation.

Figure 22 Coding example for set

	TAG	LENGTH	CONTENTS
SType1	31	0A	
name	02	01	1E
age	16	05	4A6F6E6573

1.10.1.7 Set of

A SET OF structure is a constructor which has as its contents a set of zero or more instances of a single type. The order of the instances of the type is of no significance to the application. The following is an example of a SET OF type in ASN.1:

```
ToyBox::=SET OF toys
```

The SET OF structure is similar to that SET, but it must have exactly one type. The coding for the type “ToyBox” is similar to that for Stype1, but would have zero, one, or more instances of “Toys” in the set.

1.10.1.8 IA5 String

An IA5String type is a primitive which has as its contents a string of zero or more IA5 characters. Each octet of the contents contains one binary-coded IA5 character. IA5String types are coded in a similar fashion to OCTET STRING types.

1.10.2 Choice

The CHOICE type is a structured type, defined by referencing a fixed, unordered list of distinct types; each value of the new type is a value of one of the component types. A CHOICE type is used when one set of types must be selected. The types in the CHOICE list must be made distinguishable (usually by the tag) to allow the receiver of the type to determine which one was selected. In the example below, car and bicycle types are distinguished by their different tags.

```
Vehicle    ::=    CHOICE {car, bicycle}
car        ::=    [5] IMPLICIT IA5String
bicycle    ::=    [8] IMPLICIT IA5String
```

The CHOICE list consists of one or more types, separated by commas, and enclosed within braces “{}”. Each type can be defined in the list or elsewhere.

No universal tag exists for the CHOICE.

1.11Application layer protocol formats

This section defines the CompuCALL application layer protocol formats. Only ROSE components are used. The ROSE component conforms with the CCITT Blue Book X.229 Remote Operations Protocol Specifications. Section 1.5, “Application layer designer guide,” on page 43 describes the component.

1.12 ROSE Components

A component is a sequence of data elements each consisting of a tag, a length, and contents. There are 4 types of ROSE components, each identified by a unique component tag.

The following ROSE components are supported:

- Invoke
- Return result (RR)
- Return error (RE)
- Reject

The structure of each component type is shown in the following sections.

1.12.1 Invoke component

Figure 23 Invoke component

Invoke Component	Reference	Mandatory Indication	Octet Group
Component type tag	1.12.5	Mandatory	1
Component length (Note 1)	1.12.4		2
Invoke Identifier tag	1.12.1		3
Invoke Identifier length	1.12.4	Mandatory	4
Invoke identifier	1.12.6		5
Linked Identifier tag	1.12.6		6
Linked Identifier length	1.12.4	Optional	7
Linked Identifier	1.12.6		8
Operational Value tag	1.12.7		9
Operational Value length	1.12.4	Mandatory	10
Operational Value	1.12.7		11

Figure 23 Invoke component

Argument (Note 2)	1.12.9.7 (Note 3)	Mandatory (Note 4)	12 etc.

Note 1: The component length is coded to indicate the number of octets in the component, excluding the component type tag and component length octets.

Note 2: This is a parameter of the Invoke component type.

Note 3: The coding is service-dependent.

Note 4: For ASN.1 or ROSE this could be optional, but for CompuCALL use is mandatory.

1.12.2 Return result component

Figure 24 Return result component

Return Result Component	Reference	Mandatory Indication	Octet Group
Component type tag	1.12.5	Mandatory	1
Component length (Note 1)	1.12.4		2
Invoke Identifier tag	1.12.6		3
Invoke Identifier length	1.12.4	Mandatory	4
Invoke identifier	1.12.6		5
Sequence tag	1.12.9.7	Optional	6
Sequence length (Note 4)	1.12.4	(Note 3)	7
Operational Value tag	1.12.7		8
Operational Value length	1.12.4	Optional	9
Operational Value	1.12.7	(Note 2)	10
	(Note 6)		
Result (Note 5)	1.12.9.7	Optional	11
	(Note 6)		etc.

Note 1: The component length is coded to indicate the number of octets in the component, excluding the component type tag and the component length octets.

Note 2: If a result is included, the operation value is mandatory, and it is the first element in the sequence.

Note 3: If the Return result component does not include any result, the sequence and operational value are omitted. Section 3.1.11 shows the coding for the sequence tag.

Note 4: The sequence length is coded to indicate the number of octets in the sequence, excluding the sequence type tag and sequence length octets.

Note 5: This is a parameter of the Return result component type.

Note 6: The coding is service-dependent.

1.12.3 Return error component

Figure 25 Return error component

Return Error Component	Reference	Mandatory Indication	Octet Group
Component type tag	1.12.5	Mandatory	1
Component length (Note 1)	1.12.4		2
Invoke Identifier tag	1.12.6		3
Invoke Identifier length	1.12.4	Mandatory	4
Invoke identifier	1.12.6		5
Error Value tag	1.12.9		6
Error Value length	1.12.4	Mandatory	7
Error Value	1.12.9		8
Parameter (Note 2)	1.12.9.7 (Note 3)	Optional	9 etc.

Note 1: The component length is coded to indicate the number of octets in the component, excluding the component type tag and component length octets.

Note 2: This is a parameter of the return error component type.

Note 3: The coding is service-dependent.

1.12.4 Reject component

Figure 26 Reject component

Reject Component	Reference	Mandatory Indication	Octet Group
Component type tag	1.12.5	Mandatory	1
Component length (Note 1)	1.12.4		2
Invoke Identifier tag	1.12.6	Mandatory	3
Invoke Identifier length	1.12.4		4
Invoke identifier	1.12.6		5
Problem tag	1.12.9	Mandatory	6
Problem length	1.12.4		7
Problem	1.12.9		8

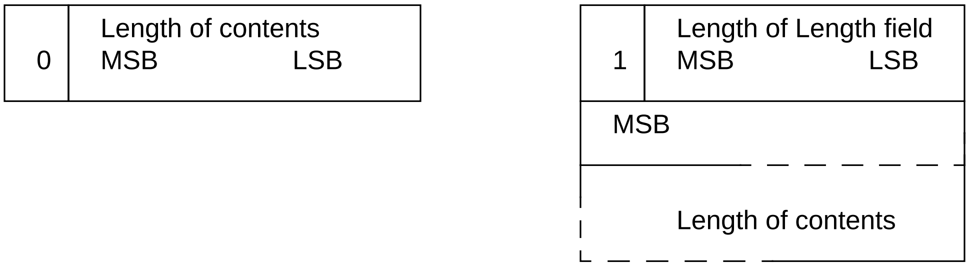
1.12.5Length of each component or data element

The length of the content uses the short or long form. If the length is less than 128 octets, the short form is used. In the short form, bit 8 is coded 0, and the length is encoded as an unsigned binary number using bits 1 to 7.

If the length of the contents is greater than 127 octets, the long form of the length of the contents is used. The long form length of the length field is from 2 to 128 octets long. Bit 8 of the first octet is coded 1, and bits 1 to 7 of the first octet encode a number as an unsigned binary whose value is one less than the total size in octets of the length field. Bit 7 of this first octet is the MSB (Most Significant Bit) and bit 1 of this first octet is the LSB (Least Significant Bit). The second octet through the last octet encode a number as an unsigned binary whose value is the actual length of the contents (in octets). Bit 8 of the second octet is the MSB and bit 1 of the last octet is the LSB. This binary number should be encoded in the fewest possible octets, with no leading octets having the value 0.

Figure 27 shows the formats of the length field described below.

Figure 27 Format of the length field



Note: CompuCALL can receive short length messages (lengths up to 127 octets) in long form formats without generating a REJECT, but it will always send short messages in short form formats.

1.12.6 Component types

Every component is specified with a component type tag. The following table contains the tags for the component types.

<u>Component type tags</u>

Bits 8 7 6 5 4 3 2 1

1 0 1 0 0 0 0 1 Invoke
1 0 1 0 0 0 1 0 Return result
1 0 1 0 0 0 1 1 Return error
1 0 1 0 0 1 0 0 Reject

1.12.7 ROSE identifiers

An invoke identifier identifies an operation invocation and is reflected in the RETURN-RESULT or RETURN-ERROR component responding to it. An invoke may refer to another Invoke through the linked identifier. When a protocol error occurs, the invoke identifier is reflected in the REJECT component, but if it is unavailable, a null is returned. Invoke and linked identifiers can be up to two octets long. The null has zero length.

The component identifiers, invoke identifier and linked identifier, are unique within a CompuCALL host application session. That is, the same identifiers can be used simultaneously in separate host application sessions without ambiguity.

<u>Component identifier tags</u>
Bits 8 7 6 5 4 3 2 1

0 0 0 0 0 0 1 0 Invoke identifier
1 0 0 0 0 0 0 0 Linked identifier

0 0 0 0 0 1 0 1 Null

1.12.7.1 Invoke identifier selection rules

CompuCALL uses the following rules for selecting the Invoke IDs:

1 The initiator of the INVOKE message assigns the Invoke ID.

2 Invoke IDs are unique per session.

3 Assignment of Invoke IDs are independent in each direction and the range (0 - 3FF or 0 - 1023 decimal) is partitioned between the DMS-100 and the host. The DMS-100 switch uses the next free Invoke ID in the range of values (512 - 1023). The host uses the next free Invoke ID in a serial loop around selection of values in the range of (0 - 511). The DMS-100 response to invoke components from the host with an invoke ID outside the expected range causes a REJECT message with a problem tag of “General Problem” and a problem reason of “Badly Structured Component.”

4 The invoke ID is encoded using 2's complement.

5 An invoke ID is released on the DMS-100 after receiving RETURN-RESULT, RETURN-ERROR or REJECT.

6 All invoke IDs are cleared after termination of the session.

7 When an ACD DN is deleted from a session, all outstanding Invoke IDs associated with the ACD DN remain. However, any subsequent responses to the outstanding Invoke IDs will cause inconsistent results (for example, any subsequent response to an outstanding dv-Call-Received-C for that ACD DN will default to the original route and no redirection would occur). Therefore, the host should not delete an ACD DN when there are outstanding Invoke IDs for that group.

1.12.8 Operations

The operation value specifies the service or operation requested. An operation value is an integer value and its meaning is specific to each service. Operation values are unique within each service discriminator.

Operation value tags

Bits 8 7 6 5 4 3 2 1

0 0 0 0 0 0 1 0 Operation value Integer

Switch computer services operation values

Bits 8 7 6 5 4 3 2 1

This one octet element identifies the operation value of messages in Chapter 3 and higher in this document.

1.12.9 Errors

Operations report errors as specified for each operation.

Error value tags

Bits 8 7 6 5 4 3 2 1	
0 0 0 0 0 0 1 0	Error value Integer

1.12.10 Problems

Protocol problems are indicated in groups. The first table specifies the tags for these groups. The other tables specify the problem values associated with each problem group.

Problem tags

Bits 8 7 6 5 4 3 2 1	
1 0 0 0 0 0 0 0	General problem
1 0 0 0 0 0 0 1	Invoke problem
1 0 0 0 0 0 1 0	Return result problem
1 0 0 0 0 0 1 1	Return error problem

Coding of general problem

Bits 8 7 6 5 4 3 2 1	
0 0 0 0 0 0 0 0	Unrecognized component
0 0 0 0 0 0 0 1	Unstopped component
0 0 0 0 0 0 1 0	Badly structured component

Coding of invoke problem
Bits 8 7 6 5 4 3 2 1

0 0 0 0 0 0 0 0 Duplicate invocation
0 0 0 0 0 0 0 1 Unrecognized operation
0 0 0 0 0 0 1 0 Unstopped argument
0 0 0 0 0 0 1 1 Resource limitation
0 0 0 0 0 1 0 0 Initiator releasing
0 0 0 0 0 1 0 1 Unrecognized linked identifier
0 0 0 0 0 1 1 0 Linked response unexpected
0 0 0 0 0 1 1 1 Unexpected child operation

Coding of return result problem
Bits 8 7 6 5 4 3 2 1

0 0 0 0 0 0 0 0 Unrecognized invocation
0 0 0 0 0 0 0 1 Result response unexpected
0 0 0 0 0 0 1 0 Unstopped result

Coding of return error problem
Bits 8 7 6 5 4 3 2 1

0 0 0 0 0 0 0 0 Unrecognized invocation
0 0 0 0 0 0 0 1 Error response unexpected
0 0 0 0 0 0 1 0 Unrecognized error
0 0 0 0 0 0 1 1 Unexpected error
0 0 0 0 0 1 0 0 Mistyped parameter

The specific problem codes for each problem class are described below.

1.12.10.1 General problems

Unrecognized component - the received component is not one of the RO defined components (i.e. the component type tag is invalid).

Mistyped component - the content of the received component does not conform to the expected content of the specific component (an Invoke component is missing an operation).

Badly structured component - the received component does not conform to the data element encoding rules (a data element is encoded with an invalid length field).

1.12.10.2 Invoke problems

Duplicate invocation - the received invoke identifier parameter is already in use and not available for reuse.

Unrecognized operation - the received is not a valid operation value or is a valid operation value but is received in a state or message that it is not allowed in.

Mistyped argument - the contents of the argument for the received Invoke component are invalid.

Resource limitation - the end receiving the operation is not able to perform the operation as a result of resource limitations.

Initiator releasing - the end receiving the operation is unwilling to perform the operation because it is about to release the connection.

Unrecognized linked identifier - there is no operation in progress with an invoke identifier equal to the specified linked identifier.

Linked response unexpected - the operation identified by the linked identifier is an operation in progress, but is not an operation for which linked operations are allowed.

Unexpected child operation - the child operation is not one the parent operation identified by the linked identifier allows.

1.12.10.3 Return result problems

Unrecognized invocation - no operation with the specified invoke identifier is in progress.

Result response unexpected - the invoked operation does not report a result, but a RETURN-RESULT.

Mistyped result - the contents of the result for the RETURN-RESULT component are invalid.

1.12.10.4 Return error problems

Unrecognized invocation - no operation with the specified invoke identifier is in progress.

Error response unexpected - the invoked operation does not report a failure, but a RETURN-ERROR was received.

Unrecognized error - the reported error is not a valid value.

Unexpected error - the reported error is a valid error value for the operation, but is not expected in response to the operation based on the contents of the invoke component argument (e.g. an error value is returned associated with an optional parameter not included in the invoke component).

Mistyped parameter - invalid contents of the parameter of the RETURN-ERROR component.

1.12.10.5 CompuCALL rules for using REJECT

REJECT in CompuCALL is kept to a minimum and only used to report protocol problems when decoding an RO message. This could occur when any condition below is encountered so the message could not be decoded:

- a tag is invalid
- a length field is invalid
- a data element marked as mandatory is missing

1.12.10.6 CompuCALL rules for using ERROR

If the message is decoded into the local concrete syntax structure (LCS) successfully, the LCS is passed to the Application Service software for further checking. At this point, each field is checked for validity. Any problems are reported in a RETURN-ERROR message. This indicates an application problem when decoding the message, or a problem with performing the requested operation. This could happen given any of the following conditions:

- if the application defined a given range for a field and the received value is outside the range
- a required data element from the application perspective is marked as optional in the syntax is missing

1.12.10.7 Parameters

Parameters included with a component (i.e. argument of ROSE Invoke, result of ROSE Return result or parameter of ROSE Return error) are defined in the individual service descriptions. They may include optional and default parameters. Parameters are one of the following:

- a sequence of parameters
- a set of parameters
- a specific parameter with its own tag
- null (absent)

If more than one parameter is required, each shall follow a sequence or set tag. Each parameter in a set or sequence is allowed to be a set or sequence.

```
Bits 8 7 6 5 4 3 2 1

    0 0 1 1 0 0 0 0 Sequence tag
    0 0 1 1 0 0 0 1 Set tag
```

2.0 CompuCALL Transports

2.1 X.25 Transport

X.25 provides layers 1-3 network services in CompuCALL. On the DMS-100 switch this is achieved by using the X.25 services of the Enhanced Multi-Protocol Controller (EMPC) card installed on the DMS I/O Controller (IOC). Detailed protocol X.25 procedures are not provided since they are beyond the scope of this specification. This chapter describes the following X.25 functions and features relevant to CompuCALL:

- virtual circuit used
- connection request
- selected CompuCALL options
- useful CompuCALL options
- multiple link configuration
- application level message dialogue
- indication of release of an established connection by the host or DMS-100
- response to an invalid host logon

2.2 Standards

CompuCALL implements level 2 and level 3 of the X.25 protocol as dictated by the International Standardization of Organization (ISO) 7776 and 8208 (See references 7 and 8 respectively in the Introduction) to support low level data communication functions. However, for level 2, CompuCALL only implements the Single Link Procedures (SLP). For level 3, the user facilities and the various default values for the packet layer parameters are in accordance with the 1984 ISO standards.

- ISO 7776 provides the ISO description of the CCITT recommendation X.25 level 2 Link Access Procedure Balanced (LAPB) interface operation as viewed by a DTE. In addition, ISO 7776 also provides the ISO description of how a DTE and a DCE (conforming to this standard) can communicate without an intervening Packet Switched Public Data Network (PSPDN).

- ISO 8208 specifies the level 3 Packet layer procedures for a DTE operating in conforming with the CCITT recommendation X.25. In addition, ISO 8208 also covers procedures necessary for a DTE and a DCE (conforming to this standard) to communicate without an intervening PSPDN.
- For modem interface specifications, see service provider specs, etc. This is beyond the scope of this document.

2.3 Type of virtual circuit used

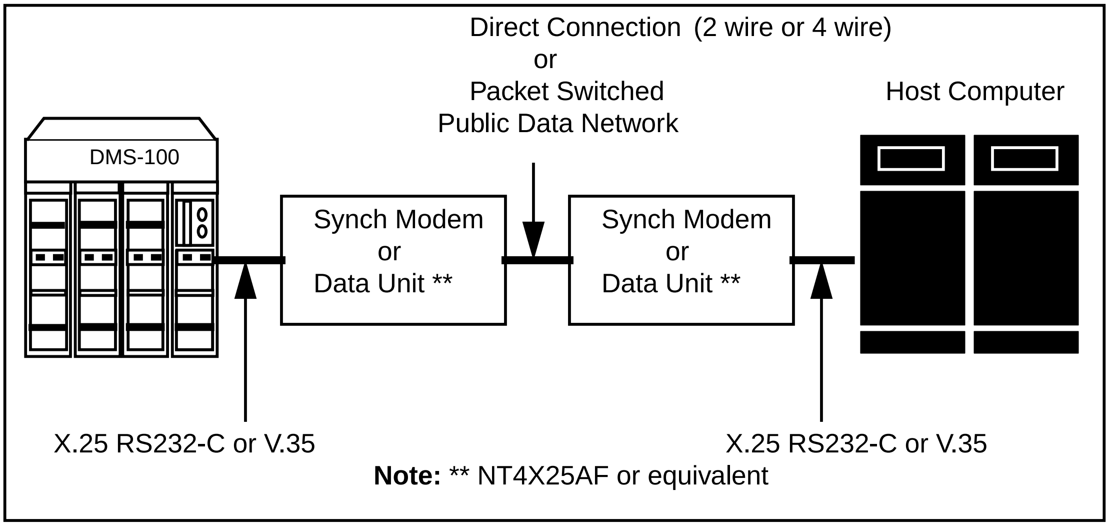
Currently CompuCALL uses the Switched Virtual Circuits (SVC) feature. An SVC is a data session that can be established by performing call setups through level 3.

2.4 Initiation of X.25 SVC

To establish a data session with the DMS-100 switch, the host must first set up a virtual circuit to the DMS-100 switch. An application session set up through layer 7 starts the process of SVC session set up. SVCs are set up with a level 3 Call_Request packet and its acceptance. Layer 7 session is subsequently set up. Any connection successes or failures are reported to layer 7. Layer 7 messages are defined in Section 3.2, “Session management,” on page 85.

This section provides all information necessary for a host to initiate and maintain an X.25 Virtual Circuit for a configuration shown in the following figure:

Figure 28 Direct or networked X.25 connection



Note: ** NT4X25AF or equivalent

2.4.1 Level 1 (physical level) characteristics

- Characteristics are supported by the selected synchronous modems or data units.

- The host must act as a DTE at the physical level (Figure 28).
- The DMS-100 switch operates as an electrical DTE, but may logically emulate a DCE. DCE emulation will be the standard protocol configuration for the X.25 link on CompuCALL at the DMS-100 end.

2.5 Selected frame and packet level options for CompuCALL

2.5.1 Level 2 (frame level) recommended characteristics

- The host acts as a DTE at the frame level either in a DTE/DTE or DTE/DCE mode with the DMS-100 switch.
- Frame Window = 7

2.5.2 Level 3 (packet level) recommended characteristics

- Packet size = 128 bytes
- Packet window = 7 (for point-to-point configuration) or 2 (for network configuration)
- Base Logical Channel Number = 1
- Number of Logical Groups = 1

For implementation detail, see the DMS-100 EMPC services from Nortel Networks.

2.6 User facility options for CompuCALL

During SVC call set up, a number of user facilities options can be specified. In this section, we identify those which may meet the requirements of a CompuCALL customer. The options below are available on 1980 X.25 or 1984 X.25. The option is a 1980 and 1984 X.25 option unless indicated otherwise between brackets.

2.6.1 Delivery confirmation

This option is generally not recommended for CompuCALL since it may reduce the network throughout. It allows the message originator to retransmit those messages in the network which were not delivered to the destination in case of network failure.The two types of X.25 error recovery procedures are the reset and restart procedure. In the reset procedure the flow control procedure on a logical channel is re-initialized to the state it was when the virtual channel was established. To reach this state, packets which may be in transit at the time of resetting are discarded, but the virtual channel connection is maintained. In this case the delivery confirmation option is useful to reduce disruption on the application-to-application session. However, in the case of the restart procedure to recover from major failures, the X.25 logical channel is cleared. Therefore, this option is not effective since CompuCALL clears all messages and restarts a new application-to-application session.

2.6.2 Extended packet sequence numbering

This is not required because the largest packet level window recommended is 7 in CompuCALL.

2.6.3 Incoming calls barred

This option prevents incoming calls on all circuits on the link. It may be used at the host side to improve security since CompuCALL X.25 connections are only originated from the host.

2.6.4 Outgoing calls barred

This option prevents outgoing calls on all circuits on the link. It is only meaningful at the switch end, but since the X.25 link at the DMS-100 switch is dedicated to CompuCALL and only receives CompuCALL X.25 calls, this option is not required.

2.6.5 Closed user group facilities

The CompuCALL customer may use closed user group basic membership for additional security.

2.6.6 End-to-end transit delay negotiation (1984)

This facility could be subscribed on the packet switched network, if provided, to improve the time response performance of messages.

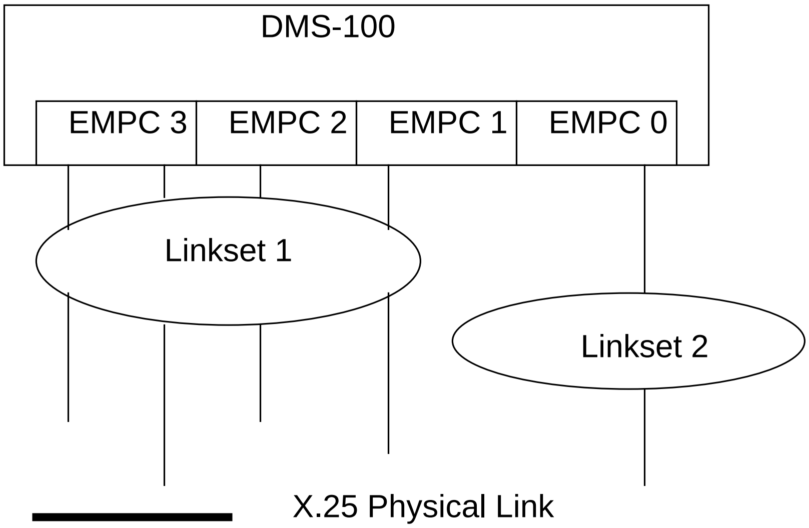
2.7 Multi-link configuration

2.7.1 CompuCALL linkset

The multiple link configuration provides resilience against network failure and increases capability throughout by allowing a CompuCALL session access to multiple network connections. The present procedure is implemented at the application layer. This is a temporary service offering to be replaced by OSI transport layer (ISO 8073/CCITT x.204) in a future release.

Multiple X.25 connections between the customer's host computer and the DMS-100 switch can be associated with the same CompuCALL session in an entity known as a linkset. A link, which belongs to a linkset, consists of a single Switched Virtual Circuit (SVC) defined by the operating company and customer. An SVC is unique to a customer and application. See Figure 29 for the allocation of links within Linksets on X.25 physical links.

Figure 29 Allocation of links on linksets on X.25 physical links



> ***Note:*** In the above diagram, linksets 1 and 2 contain 4 and 1 link(s), respectively. Link reliability and load sharing is possible for applications using Linkset 1, but not for Linkset 2. However, if a link failure occurs on Linkset 2, serviced by EMPC 0, service is discontinued for the application using this linkset (there is only one link in this Linkset).

TCP connections between the host computer and the DMS-100 switch use tcpLinksetName to identify which linkset is sent to dv-Appl-Logon. Then the session is established. For X.25 connections, tcpLinksetName is not applicable and is not checked. If tcpLinksetName is present for X.25 connections, the parameter is ignored.

The parameter supports more than one TCP connection between the host computer and the DMS-100 switch. Since TCP connection establishment identifies the endpoints consisting of IP address and port number pair, there is no association to the linkset name. Without the linkset parameter for TCP connections, it is difficult to determine which session to setup.

2.7.2 Link reliability

CompuCALL link reliability is provided if:

- More than one link belonging to a linkset is successfully established.
- Links (SVCs) associated with different X.25 physical appearances provide optimum CompuCALL link reliability.
- CompuCALL message traffic is equally distributed among all the links in a session, thus potentially increasing the message throughout the system. (See Section 2.7.3.)

2.7.3 Load sharing over the linkset

All messages originating from the host will be processed regardless of the link used. If the host requires a set of messages to be processed in a particular order, these messages should be transmitted over the same link. If different links are used, unexpected results may occur since in-sequence message delivery is not guaranteed.

In addition, the links in the session may be established or terminated during the session without affecting the DN-association. One link is required to maintain the session. The DN is associated with the session and not the individual links. The host may send the dv-DN-Associate message over any link in the linkset supporting the session.

For switch-originated messages, all links belonging to a linkset established for a given session are used. Messages are equally load-shared across all established links. The links are assigned to calls in a round-robin fashion. A link, soon after a message is transmitted, is returned to the pool of available links of the session for use. A given link transmits interspersed messages for a number of calls.

If a link fails, remaining links in the linkset carry on the message traffic of the failed link. CompuCALL service continues if there are remaining operating links in the linkset. The temporarily inoperable link, once returned to service, is returned to the pool for further use.

2.7.4 Message sequence over link

If a host requires processing messages of a particular sequence, messages must be transmitted over the same link. Similarly, if the switch requires processing messages of a particular sequence, messages are transmitted over the same link.

2.8 Interworking with layer 7

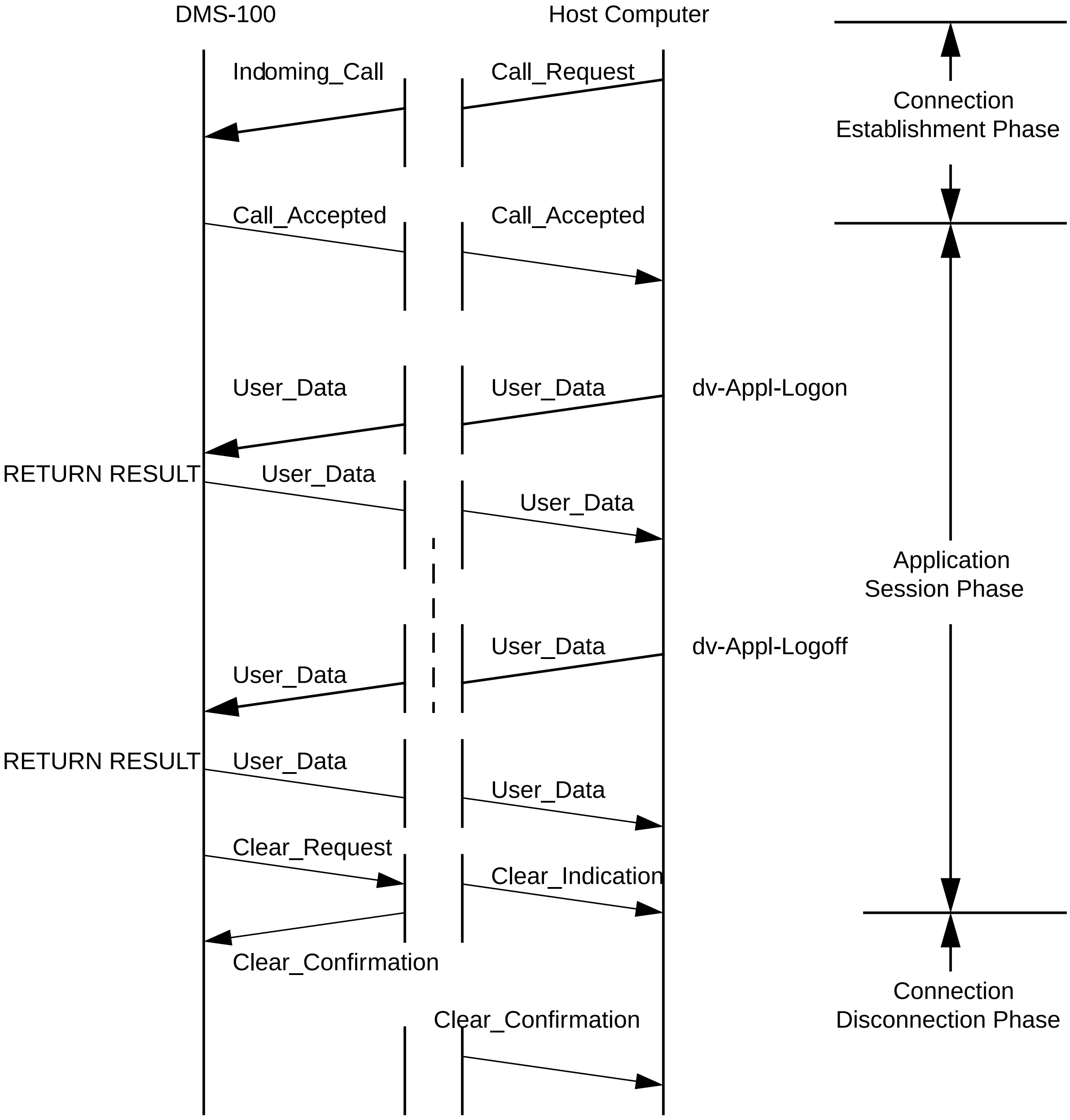
The application dialogue is performed using the ROSE Components described in Section 1.11, “Application layer protocol formats,” on page 61. An application level message is contained in a ROSE Component and is carried as a user data message over an X.25 data packet.

Normal procedures Figure 30, “X.25 and application layer interworking - normal procedure”, on page 60, shows the message sequence between the host and DMS-100 switch during a normal procedure.

- The dv-Appl-Logon message must be sent in the first Data packet following the Call_Request X.25 packet. Four octets of user data must be included in the Call_Request call from the host. The values associated with these octets are provided by the operating company as the PROTOCOL subfield. The user data must be identical to the PROTOCOL subfield and is checked before the call is accepted by the EMPC on the switch. Each value in the PROTOCOL subfield ranges from 0 to 255.

- All other application level messages (including RETURN-RESULT, RETURN-ERROR, and REJECT messages) after the dv-Appl-Logon are in X.25 data packets.
- In response to a valid dv-Appl-Logoff message from the host, the switch returns a RETURN-RESULT acknowledgment. The switch then sends a Clear_Request X.25 packet. The host should reply with a Clear_Confirmation X.25 packet.

Figure 30 X.25 and application layer interworking - normal procedure



Note: Each link in a linkset which supports an application session must be established at the X.25 level i.e., Call_Request sent for each link. In addition, each link must be established at the application level, i.e., the host sends a dv-Appl-Logon message for each link. The first successful link also establishes the application-level session. Similarly, the reverse must be performed for each link in the linkset in the termination of the session.

2.9 Abnormal procedures

- An invalid dv-Appl-Logon with a missing or invalid parameter causes a RETURN-ERROR. A dv-Appl-Logon message received with a protocol error (or any other valid or invalid application level message except dv-Appl-Logoff) in a X.25 data packet while the host is not logged on causes the switch to send a REJECT "unrecognized operation" message, not a Clear_Request packet (the existing X.25 connection is not terminated).
- A dv-Appl-Logoff message from the host application which does not have an active application level session causes a RETURN-ERROR message with reason "NotLoggedOn."
- A dv-Appl-Logon message (in a X.25 Data packet) to establish a link at the application level while the link has already been established, causes the switch to respond with a RETURN-ERROR message, not a Clear_Request packet (the existing X.25 connection is not terminated).
- A Clear_Request from the switch is used normally for taking down the SVC, i.e., terminating the X.25 connection before servicing the DMS-100 switch. The application session is taken down if the X.25 connection is the only active link in the linkset. Until normal DMS-100 service is resumed, layer 3 will not be established to receive messages.
- Procedural errors in X.25 during data transfer cause the network to reset or clear a link. In addition, if no network is involved, either the DMS-100 switch or the host may reset or clear. The reset procedure will re-initialize the flow control over the link to the previous state when the virtual channel was established. If the DMS-100 detects a link failure, subsequent messages will be routed over alternate links in the linkset, if any exist. This may cause loss of messages in transit on the network. Once an alternate link is identified, subsequent messages associated with the existing call/function are transmitted over this route even though the original link may have returned to an application-level service. Messages pertaining to the subsequent calls/functions can use the re-serviced link.
- If the DMS-100 switch receives a Clear_Indication X.25 packet during an active link connection, it confirms the packet and terminates the link at the application level. The application session is terminated only upon termination of the last link in the linkset supporting the session.
- If a link is abnormally terminated without an X.25 Clear-Request message, a switch-based failure alarm may be raised upon exceeding the alarm threshold. An alarm threshold is calculated by the number of links in the linkset which supports a given session.

2.10TCP/IP transport

TCP/IP connectivity through Ethernet Interface Unit (EIU) provides a SCAI link between the switch and the host computer for the ICCM application. The

TCP/IP link is an alternative to the X.25 MPC SCAI link. In comparison to MPC X.25, TCP/IP transport supports more flexible connectivity options.

The TCP/IP transport uses the existing TLI interface to provide connectivity between the DMS-100 switch and a business computer. Also, it makes use of Local Area Network (LAN) and an internet router.

- EIU provides the DMS-100 switch with LAN connectivity by acting as a router between the DMS-100 switch and an internet router.
- Messages originating from the business computer are routed through the internet and terminate on a router that is connected to the Ethernet LAN.
- The EIU forwards the messages to the destination node on the DMS, i.e CM.

2.10.1 Establishing a TCP connection

TCP uses a three-way handshake to establish a connection. The DMS-100 switch runs the CompuCALL server and waits for the host application to initiate the handshake. The operating company provides the IP address and the port number for the CompuCALL application.

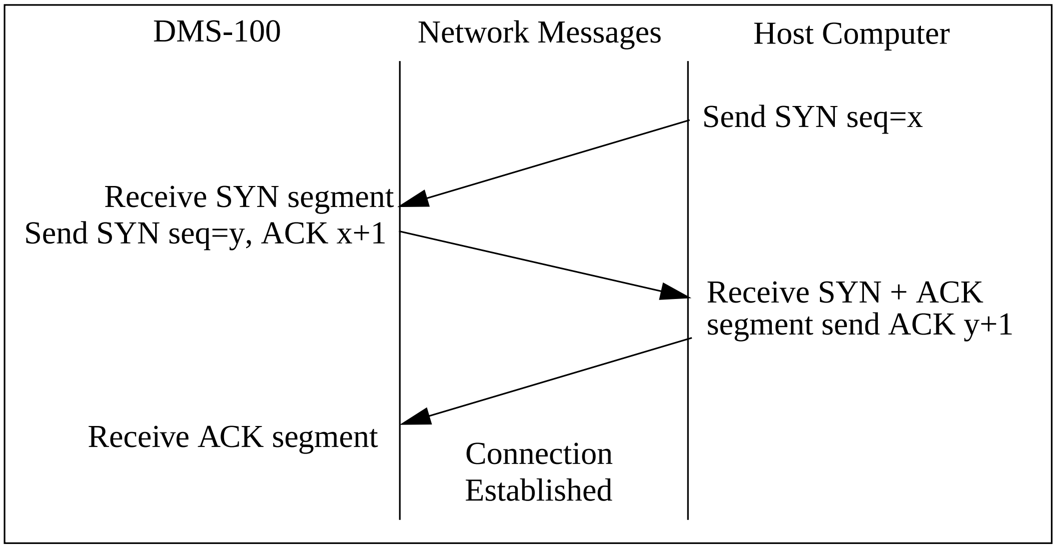
1 The host computer sends the SYN segment requesting connection to the DMS-100 switch.

2 DMS-100 switch sends an acknowledgment along with the SYN segment to the CompuCALL subscriber to indicate that it is ready to establish a session.

3 The subsciber sends an acknowledgment through the host computer.

4 The connection is fully established once the DMS-100 switch receives acknowledgment. See Figure 31, “Sequence of messages in three-way handshake,” on page 81.

Figure 31 Sequence of messages in three-way handshake



2.10.2 Terminating a TCP connection

1 The host application sends the FIN segment to close the TCP connection.

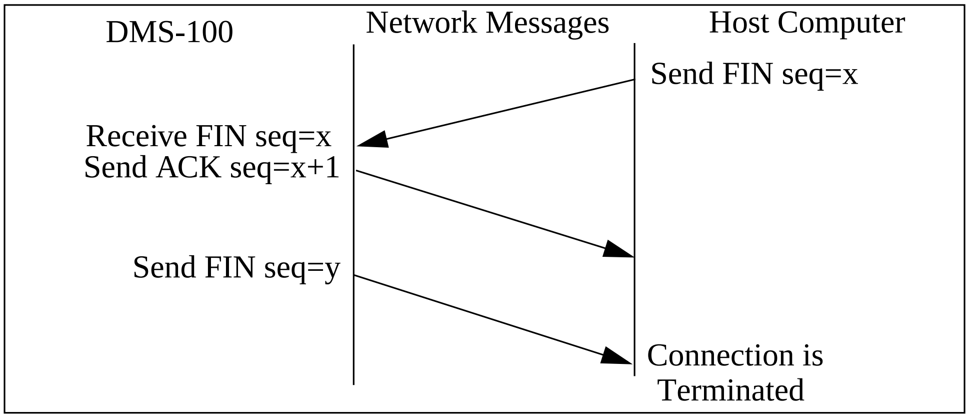
2 The DMS-100 switch acknowledges the FIN segment and informs that no more data is available.

Note: TCP refuses to accept more data from the host application and closes the connection.

3 The data continues from the DMS-100 switch to the host application. The sender closes the application.

4 DMS-100 switch sends the FIN segment. Both directions are closed and the connection is deleted.

Figure 32 Sequence of messages in terminating a TCP connection



2.10.3 Reliability

TCP has a “Keep Alive” option that the user can select to send an out of band pocket during idle conditions. This is used to detect connection failures.

2.10.4 Restrictions/limitations

The following are TCP/IP restrictions and limitations:

- The maximum number of TCP/IP connections, service nodes, and switch connections to a DMS-100 switch is 96. This is assuming no other applications are using TCP/IP connections from the CM. A maximum of 16 CompuCALL sessions are permitted using TCP/IP.
- The TCP throughout on the EIU is low when processing short messages.
- Use of the TCP stack on CM poses a significant real time impact to CM call processing. This is due to the nature of TCP processing. TCP requires error checking, acknowledgments, and retransmission of messages in case of lost messages.

3.0 Session management options

3.1 Session management options overview

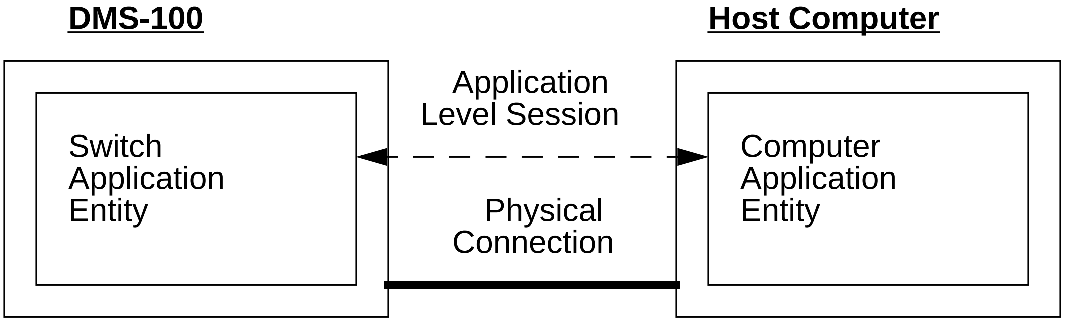
CompuCALL OA&M and CompuCALL application service options requires CompuCALL session management. For the session management service, the operating company may use all or part of the options. Details on the relationship of options with service functions, see Section 4.1, “Application service overview,” on page 121.

Session Management Application Service establishes and terminates an application session on CompuCALL. Session management includes the following capabilities:

- Host application logon establishes an application session (see Section 3.2.1).
- Host application logoff terminates an application session (see Section 3.2.3).
- Host application associateDN monitors DNs, CDNs, ACD groups and ACD agents, indicating to the switch the DNs that are associated to the session (see Section 3.2.2).

Figure 33 below shows the system configuration for session management.

Figure 33 System configuration for session management


DMS-100
Host Computer
Switch Application Entity
Application Level Session
Computer Application Entity
Physical Connection

This section also provides a description of interactions, restrictions, and limitations of session management features (see Section 3.3).

The session maintenance feature this section describes is the CONTINUITY-TEST. Once a session is established between the host and the DMS-100 switch, the host can verify the application-to-application connection with the CONTINUITY-TEST (see Section 3.5.2).

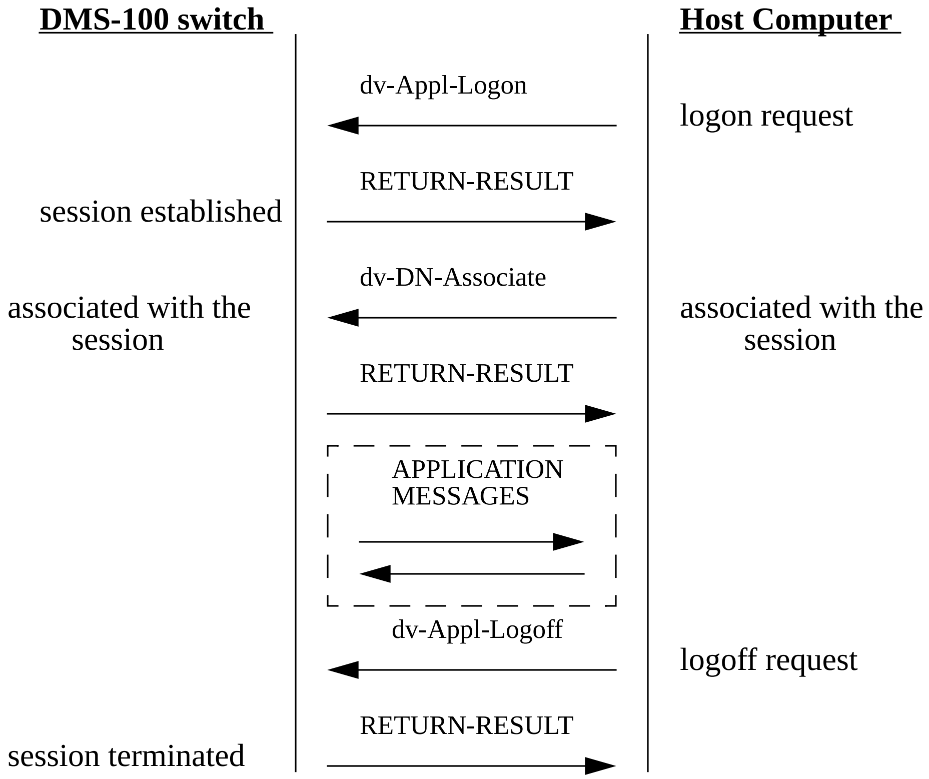
General information on remote operations is in Section 1.6 and Section 1.8 of Chapter 1.0. Following is a brief overview of remote operations:

- The Remote Operation Service Element (ROSE) protocol uses the concept of an "operation" to describe requests made by one application process to another. The party that receives an "invoke operation" request message may return a positive acknowledgment by sending a RETURN-RESULT message to the invoker. Or it may return a negative acknowledgment by sending to the invoker either a RETURN-ERROR message
that an application error occurred (e.g., invalid parameter) or a REJECT message that a protocol error occurred (e.g., unrecognized operation).
- The invoke operation message contains an "invoke identifier" which uniquely identifies the specific invoke request message. A subsequent RETURN-RESULT, RETURN-ERROR, or REJECT message contains the same invoke identifier to cross-reference the messages.
- Invoke operation messages require that either:
  - a RETURN-RESULT or RETURN-ERROR message be returned (Class 1 or 2) or,
  - no outcome be reported (Class 5).

The REJECT messages are sent when a protocol error occurs.

Figure 34 shows the RO message flow for a normal session.

Figure 34 Message flow for a normal session



3.2 Session management

During session establishment, the customer and the requested application are identified and validated for the duration of the session against information in the DMS-100 switch.

The remote operation (RO) messages associated with establishing, monitoring, and terminating the "application-to-application" level session between the host computer and the switch are:

- dv-Appl-Logon
- dv-DN-Associate
- dv-Appl-Logoff

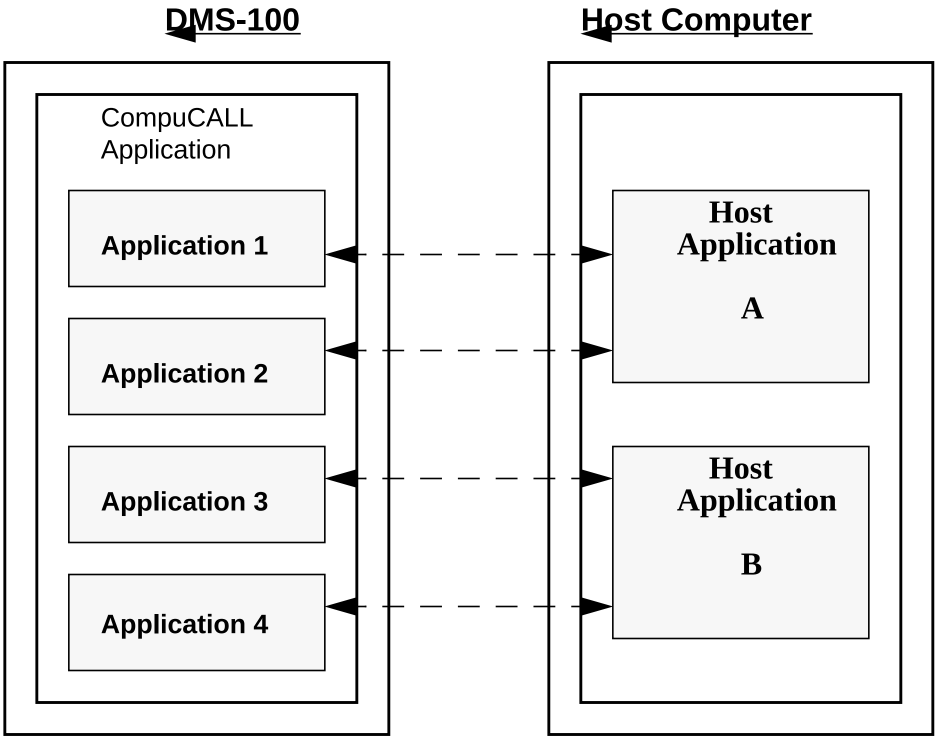
3.2.1 dv-Appl-Logon (operation value "1" hex) message

This section describes the dv-Appl-Logon message used by the host to initiate establishment of an application session over a linkset on the switch.

The host application is responsible for establishing the application-to-application level session by sending a dv-Appl-Logon message to the DMS-

100 switch. Different applications in the same physical host computer can establish separate sessions with the switch application. There can only be one host application per session, but a host computer may establish multiple sessions with the same or different applications on the DMS-100 switch. See Figure 35.

Figure 35 Relationship between host applications and sessions



Each session is carried over multiple links belonging to a linkset. Each link in a linkset is established with a separate logon procedure. The logon procedure provides basic security enforcement to ensure that a customer is not receiving information about calls to another customer. For a description of linksets, see Section 2.7, “Multi-link configuration,” on page 74.

The first dv-Appl-Logon message for a link in a linkset received by the DMS-100 switch sets up the CompuCALL session for the application. Subsequent logon attempts for other links in the linkset provide increased bandwidth for CompuCALL message traffic and also increased reliability for CompuCALL access. The subsequent dv-Appl-Logon messages for each link in the linkset must contain exactly the same parameters and values as the initial message, except the Invoke ID. After a session is established, the host application sends any other CompuCALL message (e.g., dv-DN-Associate). The host application can add other links in the linkset at the beginning or at anytime during the previous session. The other links in the established linkset provide:

- increased bandwidth of the CompuCALL session (i.e. traffic will be spread across all links)
- increased reliability of the CompuCALL session

3.2.1.1 dv-Appl-Logon parameters

The dv-Appl-Logon message includes the following parameters which must all be provided by the host. Although each parameter is defined in Appendix A of this document, they are briefly described below:

> ***Note:*** A customer session involves only one service ID, application ID, network node ID, business group ID, and password.

NetworkNodeID specifies the switch with which the host application wishes to communicate. This is the switch which the host connects to via the CompuCALL link and is defined by the operating company.

ServiceID identifies a service profile for a session. An operating company can define functions and parameters in service profiles on the DMS-100 switch. Also an operating company can assign a customer up to eight service IDs. The following are uses and functions of service IDs:

a. all are under the same password

b. can refer to the same service profile

c. an active session can use more than one

d. customer can choose any eight assigned to a linkset datafilled

ServiceVersion specifies the application level signaling version used by the host (i.e., 35 for BCS35).

BusinessGroupID identifies the customer to whom the host application belongs. This is defined by the operating company.

ApplicationID identifies the specific customer host application (e.g., insurance company customer policy information database, outstanding claims database) initiating the logon request and is specified by the host.

Password is used for security purposes and corresponds to a specific BusinessGroupID. One password datafilled for the customer is applicable to all eight service IDs.

tcpLinksetName specifies the linkset name.

3.2.1.2 dv-Appl-Logon error conditions

For each unsuccessful link logon, the switch sends a RETURN-ERROR (i.e., negative acknowledgment) message to the host indicating the appropriate error values.

> **Note:** The error reasons are bold in brackets.

1 The NetworkNodeID is incorrect {invalid_ parameter}.

2 The ServiceID determines which switch application service functions or subset does not coincide with the datafill in the DMS-100 switch in table SCAIPROF {(i.e. Service Profile) invalid_parameter}.

3 The ServiceVersion determines the application level signaling version used by the host (i.e. 35 for BCS35) is incorrect {invalid_parameter}.

> ***Note 1:*** During the session, before the DMS-100 switch sends call event messages, the dv-Appl-Logon service version is checked before including parameters at the application level.

4 The BusinessGroupID is not assigned the specific ServiceID {(i.e., Service Profile) invalid_parameter}.

5 The ApplicationID is not in the range of 1 to 32 767 {invalid_parameter}.

6 The password is invalid for the specific BusinessGroupID {invalid_parameter}.

7 The logon message is received for a link that has already logged on {link already in use}.

> ***Note 2:*** A new session is established if it is the first link in the linkset. Otherwise, the parameters are verified against the initial successful logon attempt for a link in this linkset.

8 Unavailable software resources {resource limitation}.

Only one reason is provided with the above priority. Furthermore, only the first missing or invalid parameter encountered by the switch is included in the RETURN-ERROR message.

A RETURN-ERROR indicates the session has not been established.

The switch does not retransmit if it receives a REJECT message from the host in response to either a RETURN-RESULT or RETURN-ERROR message.

If the dv-Appl-Logon message cannot be decoded by the switch, the switch will send a REJECT.

Note: Currently certain restrictions relate to the password and message generation; see Section 3.3.4, “dv-Appl-Logon restrictions,” on page 112.

3.2.1.3 X.25 Link dv-Appl-Logon: ASN.1 encoding

```
dv-Appl-Logon       OPERATION
                    ARGUMENT            LogonArgument
                    RESULT
                    ERRORS                      {link-Already-In-Use,
                                                missing-Parameter,
                                                invalid-Parameter,
                                                resource-Limitation,
                                                invalid-LinkSet-Name)
::= 1
LogonArgument ::=   SEQUENCE
{networkNodeID      [0] IMPLICIT INTEGER        OPTIONAL,
serviceID           [1] IMPLICIT INTEGER        OPTIONAL,
serviceVersion      [2] IMPLICIT INTEGER        OPTIONAL,
businessGroupID     [3] IMPLICIT INTEGER        OPTIONAL,
applicationID       [4] IMPLICIT INTEGER        OPTIONAL,
password            [5] IMPLICIT IA5String      OPTIONAL,
tcpLinksetName      [6] IMPLICIT IA5String      OPTIONAL}

link-Already-In-Use ERROR               ::=0

missing-Parameter   ERROR               :: = 1
                    PARAMETER           MissingParameter

MissingParameter  ::=SEQUENCE
{missingParameterType       [0] IMPLICIT ENUMERATED
{missingNetworkNodeID       (0),
    missingServiceID        (1),
    missingServiceVersion   (2),
    missingBusinessGroupID  (3),
    missingApplicationID    (4),
    missingPassword         (5),                OPTIONAL}
    missing TCPLinksetName(6)}                  OPTIONAL}

invalid-Parameter   ERROR                       :: = 2
                    PARAMETER                   InvalidParameter

InvalidParameter   =        SEQUENCE
    {   invalidParameterType        [0] IMPLICIT ENUMERATED
        {   invalidNetworkNodeID    (0),
        invalidServiceID            (1),
        invalidServiceVersion       (2),
        invalidBusinessGroupID      (3),
invalidApplicationID      (4),
invalidPassword           (5),                OPTIONAL)
invalidLinkset            (6)}
resource-Limitation       ERROR               :: = 3
```

3.2.1.4 TCP/IP link dv-Appl-Logon: ASN.1 encoding

```
dv-Appl-Logon       OPERATION
                    ARGUMENT    LogonArgument
                    RESULT
                    ERRORS      {  link-Already-in-use,
                                   missing-Parameter,
                                  invalid-Parameter,
                                  resource-Limitation,
                                  invalid-LinkSet-Name}
::= 1
LogonArgument               ::=     SEQUENCE
    { networkNodeID       [0] IMPLICIT INTEGER OPTIONAL,
      serviceID           [1] IMPLICIT INTEGER OPTIONAL,
      serviceVersion      [2] IMPLICIT INTEGER OPTIONAL,
      businessGroupID     [3] IMPLICIT INTEGER OPTIONAL,
      applicationID       [4] IMPLICIT INTEGER OPTIONAL,
      password            [5] IMPLICIT IA5String OPTIONAL,
      tcpLinksetName      [6] IMPLICIT IA5String OPTIONAL}
```

3.2.1.5 X.25 link dv-Appl-Logon: coded example

The host asks the switch with NetworkNodeID = 1 (decimal) to establish an application-to-application session for ServiceID = 3. It provides a password = ABCD by invoking dv-Appl-Logon [1] with InvokeID = 33 (21 Hex). The host receives RETURN-RESULT from the switch, indicating successful session establishment. If a dv-Appl-Logon is sent without a network node ID, the switch sends a RETURN-ERROR indicating that NetworkNodeID is missing.

Request from host to DMS to establish a CompuCALL session:

INVOKE DMS <===== HOST

Table 1 X.25 dv-Appl-Logon invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1D		INVOKE	This message is 29 bytes decimal.
02	01	21	InvokeID	The invoke ID is 33 decimal.
02	01	01	Operation	The operation value is dv_APPL_LOGON [1] decimal.
30	15		Argument	The logon argument is of Sequence type and is 21 bytes long decimal.
80	01	01	NetworkNodeID	The network node ID is 1.
81	01	03	ServiceID	The serviceID is 3.
82	01	07	SeviceVersion	The sevice version is SCAI12.
83	01	05	BusinessGroupID	The business group ID is 5.
84	01	01	ApplicationID	The application ID is 1.
85	04	41 42 43 44	Password	The password is ABCD.
Hex Dump=A1 1D 02 01 21 02 01 01 30 15 80 01 01 81 01 03 82 01 07 83 01 05 84 01 01 85 04 41 42 43 44				

RETURN-RESULT DMS =====> HOST

Table 2 X.25 dv-Appl-Logon return result coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	This message is 3 bytes long.
02	01	21	InvokeID	The invoke ID is 33 decimal.
Hex Dump=A2 03 02 01 21				

RETURN-ERROR DMS ======> HOST

Table 3 X.25 dv-Appl-Logon return errors coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	0B		RETURN-ERROR	This message is 11 bytes long.
02	01	21	InvokeID	The InvokeID is 33 decimal.
02	01	01	ErrorType	The error type is Missing Parameter [1].
30	03			Missing Parameter is a sequence of 3 bytes.
80	01	00	Error Parameter	The Missing Parameter is missing the NetworkNodeId, the first parameter encountered.
Hex Dump=A3 0B 02 01 21 02 01 01 30 03 80 01 00				

3.2.1.6 TCP/IP dv-Appl-Logon: Coded Example

INVOKE **DMS <===== HOST**

Table 4 TCP/IP dv-Appl-Logon invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	25		INVOKE	The message length is 37 bytes decimal.
02	01	21	InvokeID	Its Invoke ID is decimal 33.
02	01	01	Operation	Its operation value is dv-Appl-Logon [1] decima.l
30	15		Argument	Application Logon Arg is of type Sequence and is 21 bytes long decimal.
80	01	01	Network Node ID	The network node ID is a primitive integer value which maps to the NetNodeId1 in table SCAIGRP.
81	01	03	Service ID	The service ID is a primitive integer value which maps to the Service ID 3 in table SCAIPROF.
82	01	07	Service Version	The service version is a primitive integer value which maps to SCAI07.
83	01	03	Business Group ID	The business group ID is a primitive integer value which maps to “local 3” BGID in table CUSTNTWK & SCAIGRP.
84	01	01	Application ID	The Application ID is a primitive integer value which maps to 1, anything the host wishes to send within the numeric range {1.. 32 767}.
85	04	41 42 43 44	Pass Word	The password is ABCD, as datafilled in table SCAIGRP.
86	08	54 43 50 4C 49 4E 4B 31	TCP Linkset Name	The linkset name is TCPLINK1, as datafilled in table SCAICOMS and included in table SCAIGRP.
Hex dump = A1 25 02 01 21 02 01 01 30 15 80 01 01 81 01 03 82 01 07 83 01 03 84 01 01 85 04 41 42 43 44 86 08 54 43 50 4C 49 4E 4B 31				

3.2.2 dv-DN-Associate (operation value "2" hex) message

This section describes the dv-DN-Associate message the host uses to monitor DNs (MDC/RES DNs, ACD DNs, CDNs, and agent positions in an established application session.

The host application uses the dv-DN-Associate message to specify which customer's DNs to receive indications about for incoming calls. After establishing a session, the host indicates to the DMS-100 switch which DNs the application wants to associate with in the current session. Customer group DNs must match the customer's Business Group ID which the dv-Appl-Logon message establishes in the current session.

The non-ACD lines must have the ECM line option and AGTASSN option on the DMS-100 switch to associate non-ACD DNs. In all cases, the application session must subscribe to the appropriate Service Profile to allow DN association. For ACD agent association by positionID, the DMS-100 switch must datafill the AGTASSN (ACD agent association) parameter.

3.2.2.1 dv-DN-Associate parameters

The dv-DN-Associate message is sent any time during the session. It contains the following three parameters described below:

AssociatedDN identifies the DN of an ACD group or a non-ACD DN, Centrex DN, and residential DN or CDN. It is not required for ACD DN associations with defined operations.

A CDN can be viewed as an ACD DN, for which there are no agents or supervisors and one queue. A CDN is identified by a Directory Number (DN) that holds onto calls waiting for routing instructions by a host computer. The DMS-100 switch handles the call according to the host’s response. If the time expires before the DMS-100 switch receives a response, the call routes to a DEFAULT ACD DN, which the DMS-100 switch defines for the CDN.

DNoperation is provided by the host and indicates if the specified DN is to be added to, removed from, or replaced with the current set of DNs associated with the session.

a. Add

The specified DN is added to the set of DNs associated with the session. If no DN is in the message, the existing set is unaltered.

b. Delete

All previous CDNs, ACD DNs, and agentDNs are deleted from the set of DNs associated with the session. If no DN is specified, the current set is unaltered.

c. Define

   i. A dv-DN-Associate message with the DEFINE operation and no DN in the associated DN parameter associates the host computer to all the ACD DNs and CDNs within the host's customer group which are not associated with another session.

   ii. A dv-DN-Associate message with the DEFINE operation and an ACD DN in the associated DN parameter disassociates all the ACD DNs and CDNs which are associated by that host computer except for that specified ACD DN.

   iii. A dv-DN-Associate message with the DEFINE operation and a non-ACD DN in the associated DN parameter fails with an Invalid DN error message.

AssociatedAgent identifies an ACD agent position id. This is an optional parameter. The switch will perform the following validations when AssociatedAGENT is in use:

a. Verify that the service version of the current host-switch session is SCAI10 or higher.

b. Verify that the agent specified belongs to the same customer group as the current host-switch session.

c. Verify that the agent position id can only be associated if the AGTASSN option is assigned to the agent’s ACD group in table ACDGRP.

   i. If the ACD agent position id is associated, call events for the ACD agent are sent to the host. The maximum number of ACD agent positions that can be associated with a single ACD group is 1024. The dv-DN-Associate will allow associating up to 2048 agents in a session.

   ii. If a dv-DN-Associate message is sent to the switch with an ACD agent position id and DELETE in the operation parameter, call events for the ACD agent are no longer sent to the host.

If the host specifies a primary ACD DN to be associated with the current session, the host is informed by the DMS-100 switch of all calls to that ACD group. Furthermore, if the ACD group has one or more supplementary ACD DNs associated with it, all of these supplementary ACD DNs are automatically associated with the current session.

3.2.2.2 dv-DN-Associate error conditions

1 DNoperation is not included {missing DNoperation, int = 1}.

2 AssociateDN parameter is incorrect {(the DN does not belong to the customer that is logged onto the session, the ACD DN is a supplementary DN; invalid AssociatedDN, int = 2}.

Note: AssociateDN length of 0 is valid only with the DEFINE operation.

3 The AssociateDN is associated to another session {dn-already-associated, int = 3}.

4 The AssociateDN is associated to the current session {dn-already-in-set, int = 4}.

5 For DELETE operation the AssociateDN is not associated to the session {dn-not-in-set, int = 5}.

6 The maximum value of AssociateDN or AssociateAgent for a session is reached {max-association, int = 6}.

7 Unavailable software resources {no resources, int = 7}.

8 Neither the associatedDN nor the associatedAgent parameter is included in the dv-DN-Associate message {missing-DN, int = 8}.

9 The AGTASSN option is not datafilled for the agent's ACD group in table ACDGRP {not-allowed, int = 9}.

10 The associatedAgent parameter is included in a dv-DN-Associate message with the 'DEFINE' operation {not-allowed, int = 9}.

11 The associatedAgent parameter contains an invalid agent position id {invalid-agt, int = 10}.

12 For an 'ADD' operation, the associatedAgent parameter contains an agent position id which has already been associated by a different session {agent-already-associated, int = 11}.

13 For an 'ADD' operation, the associatedAgent parameter contains an agent position id which has already been associated by the current session {agent-already-in-set, int = 12}.

14 For a 'DELETE' operation, the associatedAgent parameter contains an agent position id which has not been associated by the current session {agent-not-in-set, int = 13}.

15 The ACD group is already associated to the session and the dv-DN-Associate message is received with the associatedAgent parameter {acd-group-in-set, int = 14}.

16 The ACD agent is already associated to the session and the dv-DN-Associate message is received with the associatedDN parameter which includes the ACD group primary DN {acd-agent-in-set, int = 15}.

17 The associatedDN and associatedAgent parameters are both included in the same dv-DN-Associate message {extraneous-parm, int = 16}.

If the operation is successful, the switch sends a RETURN-RESULT (positive acknowledgment) message to the host. This message has no parameters. The following operations become available.

If the operation is unsuccessful, the switch sends a RETURN-ERROR (negative acknowledgment) message to the host not allowed.

If the switch receives a dv-DN-Associate message from a host with no active application level session, it returns a REJECT message with reason “unrecognized operation."

The switch does not retransmit if it receives a REJECT message from the host in response to either a RETURN-RESULT or RETURN-ERROR message.

When an ACDDN is deleted from a session, all outstanding invoke IDs associated with the ACDDN remain. However, any subsequent responses to the outstanding invoke IDs will cause inconsistent results (for example, any subsequent response to an outstanding dv-Call-Received-C for that ACD DN will default to the original route and no redirection occurs). Therefore, the host should not delete an ACD DN when there are outstanding invoke IDs for that group.

> ***Note:*** Currently, certain restrictions relate to the dv-DN-Associate message, Section 3.3.5, “dv-DN-Associate restrictions,” on page 112.

3.2.2.3 dv-DN-Associate: ASN.1 encoding

The host sends the RO of Operation Class 2 to the DMS-100 switch.

```
dv-DN-Associate              OPERATION
                             ARGUMENT     AssociateArgument
                             RESULT
                             ERRORS     {    missing-DNOperation,
                                               invalid-DN,
                                               dn-Already-Associated,
                                               dn-Already-In-Set,
                                               dn-Not-In-Set,
                                               max-No-of-lines-Associated,
                                               no-Software-Resources,
                                               missing-Associated-DN
                                               not-Allowed
                                               invalid-Agent,
                                             agent-Already-Associated,
                                             agent-Already-In-Set,
                                             agent-Not-In-Set,
                                             acd-Group-In-Set,
                                             ACD-agent-In-Set,
                                             extraneous-Parameter}

             ::= 2

AssociateArgument            ::=     SEQUENCE
{associatedDN                [0] IMPLICIT OCTET STRING    OPTIONAL,
dnOperation          [1] IMPLICIT  ENUMERATED       OPTIONAL,
{add                         (0),                         OPTIONAL,
delete                       (1),                         OPTIONAL,
define                       (2)}                         OPTIONAL,
associatedAgent              [2]AddressType               OPTIONAL}

 AddressType::= CHOICE
  {positionID                [0]IMPLICIT  INTEGER,
    dialedDigits             [1]IMPLICIT  IA5String,
    stationNumber             [2]IMPLICIT  OCTET STRING,
    q931Address              [3]Q931AddressType}


 Q931AddressType::=            SEQUENCE
 {numberTypeNumberPlan         [0]IMPLICITOCTET STRING OPTIONAL,
  presentationScreeningIndicator [1]IMPLICITOCTET STRING OPTIONAL,
  digits                       [2]IMPLICITIA5STRING OPTIONAL}
           missing-DNOperation              ERROR::=1
           invalid-DN                       ERROR::=2
           dn-Already-Associated            ERROR::=3
```

dn-Already-In-Set	ERROR::=4
dn-Not-In-Set	ERROR::=5
max-No-of-lines-Associated	ERROR::=6
no-Software-Resources	ERROR::=7
missing-Associated-DN	ERROR::=8
not-Allowed	ERROR::=9
invalid-Agent	ERROR::=10
agent-Already-associated	ERROR::=11
agent-Already-In-Set	ERROR::=12
agent-Not-In-Set	ERROR::=13
acd-Group-In-Set	ERROR::=14
acd-Agent-In-Set	ERROR::=15
extraneous-parameter	ERROR::=16

3.2.2.4 dv-DN-Associate: coded example

The host requests the switch to associate an ACD group with a session by sending dv-DN-Associate with invokeID = 36 (24 Hex) and DNoperation = Add to add the AssociatedDN (primary ACD DN = 6137221111) to the current set of DNs associated with the session. The host receives RETURN-RESULT indicating successful operation. If the DN is already associated with the current session, the switch sends a RETURN-ERROR indicating DN is already set. If the dv-DN-Associate is received when no session is logged on, the host receives REJECT message with reason "unrecognized operation". For request from the host to the DMS-100 switch to associate an ACD agent using PositionID, see Table 5.

For request from the host to the DMS-100 switch to associate an ACD group with a session, see Table 6 (ACDDN is used).

INVOKE DMS <===== HOST

Table 5 DN associate request for agent association invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	10		INVOKE	Message length is 16 bytes.
02	01	24	InvokeID	Invoke ID is 36.
02	01	02	Operation	Operation value for dv-DN-Associate is decimal 2.
30	08		AssociateArgument	Type SEQUENCE and 8 bytes long.
81	01	00	dnOperation	The dnOperation is add.
A2	03		associateAgent	associateAgent is a 3 byte constructor.
80	01	02	positionID	The position ID is 2.
Hex Dump = A1 10 02 01 24 02 01 02 30 08 81 01 00 A2 03 80 01 02				

INVOKE DMS <===== HOST

Table 6 DN associate request for DN association invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	17		INVOKE	Message length is 23 bytes.
02	01	24	InvokeID	Invoke ID is 36.
02	01	02	Operation	Operation value for dv-DN-Associate is decimal 2.
30	0F		AssociateArgument	Type SEQUENCE and 15 bytes long.
80	0A	36 31 33 37 32 32 31 31 31 31	associatedDN	The associatedDN is 613-722-1111.
81	01	00	dnOperation	The dnOperation is add.
Hex Dump = A1 17 02 01 24 02 01 02 30 0F 80 0A 36 31 33 37 32 32 31 31 31 31 81 01 00				

RETURN-RESULT DMS =====> HOST

Table 7 DN associate request for DN association return error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	the message length is 3 bytes decimal.
02	01	24	InvokeID	its invoke ID is decimal 36.
Hex dump = A2 03 02 01 24				

RETURN-ERROR DMS =====> HOST

Table 8 DN associate request for DN association return error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	the message length is 6 bytes decimal.
02	01	24	InvokeID	its invoke ID is decimal 36.
81	01	04	Error Value	Error Value is 4 representing DN-Already-In-Set.
Hex dump = A3 06 02 01 24 02 01 04				

3.2.3 dv-Appl-Logoff (operation value "4" hex) message

This section describes the dv-Appl-Logoff message the host uses:

1 to terminate an established application

2 to close the TCP/IP connection

3 to tear down the switched X.25 virtual circuit

If the X.25 session carries over a linkset consisting of more than one link, the host must send a dv-Appl-Logoff message to terminate each link. The termination of the last link in the linkset terminates the application session.

This RO of Operation Class 1 is sent by the host to the switch to terminate one link of a linkset in an application-to-application level session.
The dv-Appl-Logoff message has no parameters.

On receiving the dv-Appl-Logoff message for a link, the switch sends a RETURN-RESULT message (acknowledgment) to the host and initiates clearing the underlying network layer. Following this, the switch no longer uses this link to deliver messages. An X.25 based session is active until all links in the linkset are terminated or taken out of service.

The host does not need to delete or disassociate the DN associations (ACD DNs, CDNs, agent positions) from the session before dv-Appl-Logoff. The switch automatically disassociates it on logoff.

Any outstanding ROs after logoff are invalid; i.e., InvokeIDs are cleared.

If the switch receives a dv-Appl-Logoff message from a host for an inactive link, it returns a RETURN-ERROR message with reason "NotLoggedOn".

A REJECT message from the host in response to the RETURN-RESULT or RETURN-ERROR message will not cause the switch to retransmit the message.

3.2.3.1 dv-Appl-Logoff: ASN.1 encoding

```
dv-Appl-Logoff          OPERATION
                        RESULT
                        ERRORS              {not-Logged-On}
::=  4

not-Logged-On           ERROR               ::= 0
```

3.2.3.2 dv-Appl-Logoff coded example:

The host application requests the switch to terminate the CompuCALL session by sending dv-Appl-Logoff (4) with InvokeID = 34 (22 Hex). The host receives RETURN-RESULT indicating successful operation.

INVOKE **DMS <===== HOST**

Table 9 dv-Appl-Logoff invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	06		INVOKE	This message is 6 bytes long decimal.
02	01	29	InvokeID	The invoke ID is 34 decimal.
02	01	04	Operation	The operation is dv-Appl-Logoff [4].
Hex Dump=A1 06 02 01 29 02 01 04				

RETURN-RESULT DMS =====> HOST

Table 10 dv-Appl-Logoff return result example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	This message is 3 bytes long decimal.
02	01	22	InvokeID	The invoke ID is 34 decimal.
30	06	29	Argument	Its argument is of type Sequence and is 6 bytes long.
Hex Dump=A2 03 02 01 22 30 06 29				

RETURN-ERROR DMS =====> HOST

Table 11 dv-Appl-Logoff return error example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	This message is 6 bytes long decimal.
02	01	22	InvokeID	The invoke ID is 34 decimal.
02	01	00	Error Value	The error value is Not-Logged-On[0].
Hex Dump:=A3 06 02 01 22 02 01 00				

REJECT DMS =====> HOST

Table 12 dv-Appl-Logoff reject example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A4	06		RETURN-REJECT	This message is 6 bytes long decimal.
02	01	22	InvokeID	The invoke ID is 34 decimal.
81	01	01	Invoke Problem	The invoke problem is unrecognized (1).
Hex Dump=A4 06 02 01 22 81 01 01				

3.3 Interactions, restrictions and limitations

3.3.1 Message interactions

This section covers interactions between these messages with other messages in the remaining section of this chapter or in Chapter 4.0 Application service options.

3.3.1.1 Between dv-DN-Associate and dv-Call-Received-C

When an ACDDN is deleted from a session, all outstanding invoke IDs associated with the ACDDN remain. Any subsequent responses to the outstanding invoke IDs will cause inconsistent results. For example, any subsequent response to an outstanding dv-Call-Received-C for that ACD DN defaults to the original route and no redirection occurs. Therefore, the host should not delete an ACD DN when there are outstanding invoke IDs for that group.

3.3.2 Interworking with X.25 Layer 3

The application dialogue is performed using the ROSE components described in Section 1.11, “Application layer protocol formats,” on page 61. An application level message is contained in a ROSE Component and is carried within an X.25 data packet.

Normal procedures

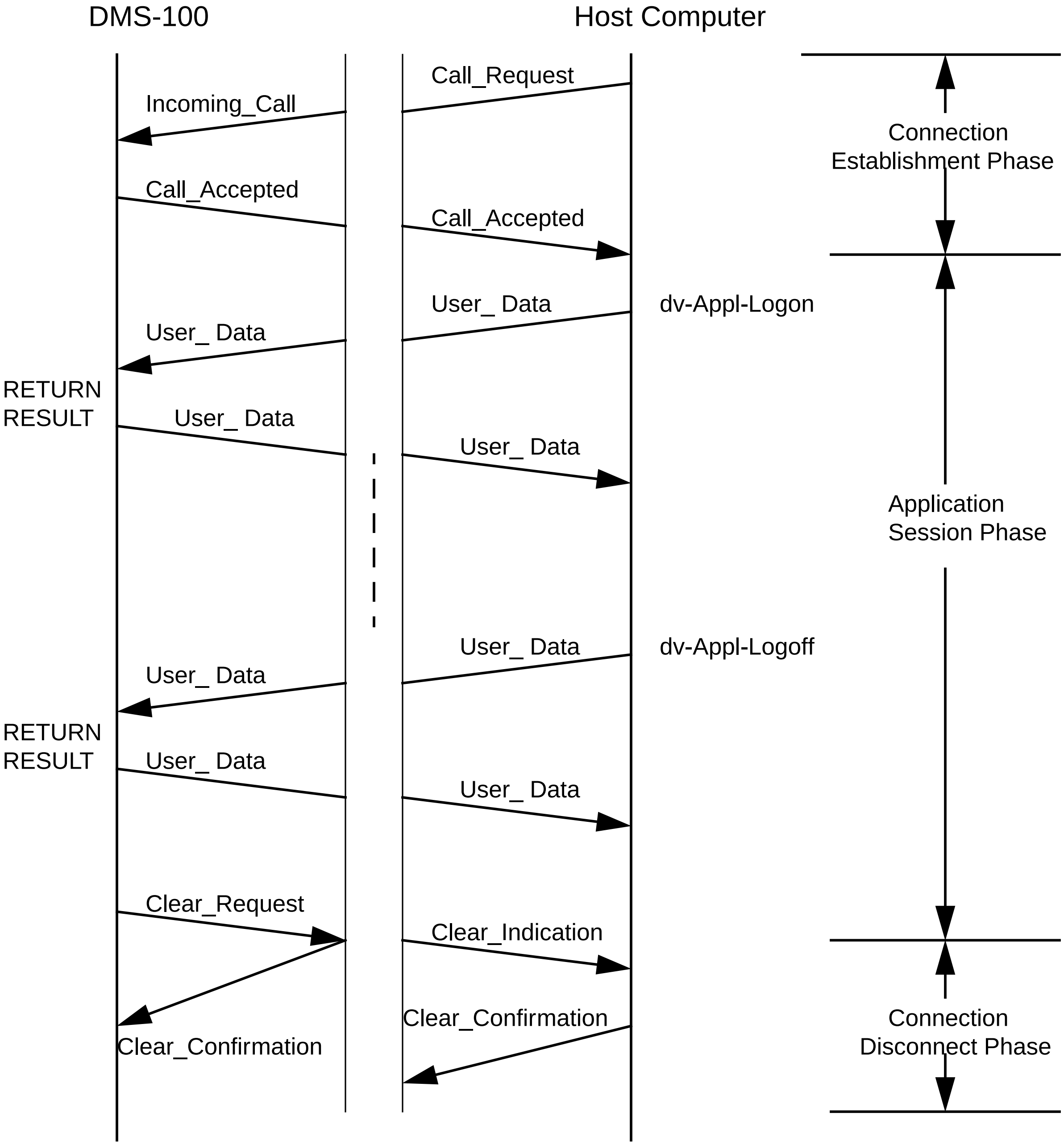
Figure 36, “X.25 and application layer interworking - normal procedure,” on page 106 shows the message sequence between the host and DMS-100 switch during a normal procedure.

- The dv-Appl-Logon message is sent in the first Data packet following the Clear_Request X.25 packet. Four octets of user data must be included in the Call_Request packet from the host. The values associated with these octets are provided by the operating company as the PROTOCOL subfield. The user data must be identical to the PROTOCOL subfield and is checked before the call is accepted by the EMPC on the switch. The DMS-100 switch datafills the PROTOCOL subfield in table SCAICOMS under the link definition field. Each value in the PROTOCOL subfield ranges from 0 to 255 (decimal).
- All other application level messages (including RETURN-RESULT, RETURN-ERROR, and REJECT messages) after the dv-Appl-Logon are contained in X.25 data packets.
- In response to a dv-Appl-Logoff message from the host, the switch returns a RETURN-RESULT acknowledgment. The switch will then send a Clear-Request X.25 packet. The host should reply with a Clear-Confirmation X.25 packet.

Abnormal procedures

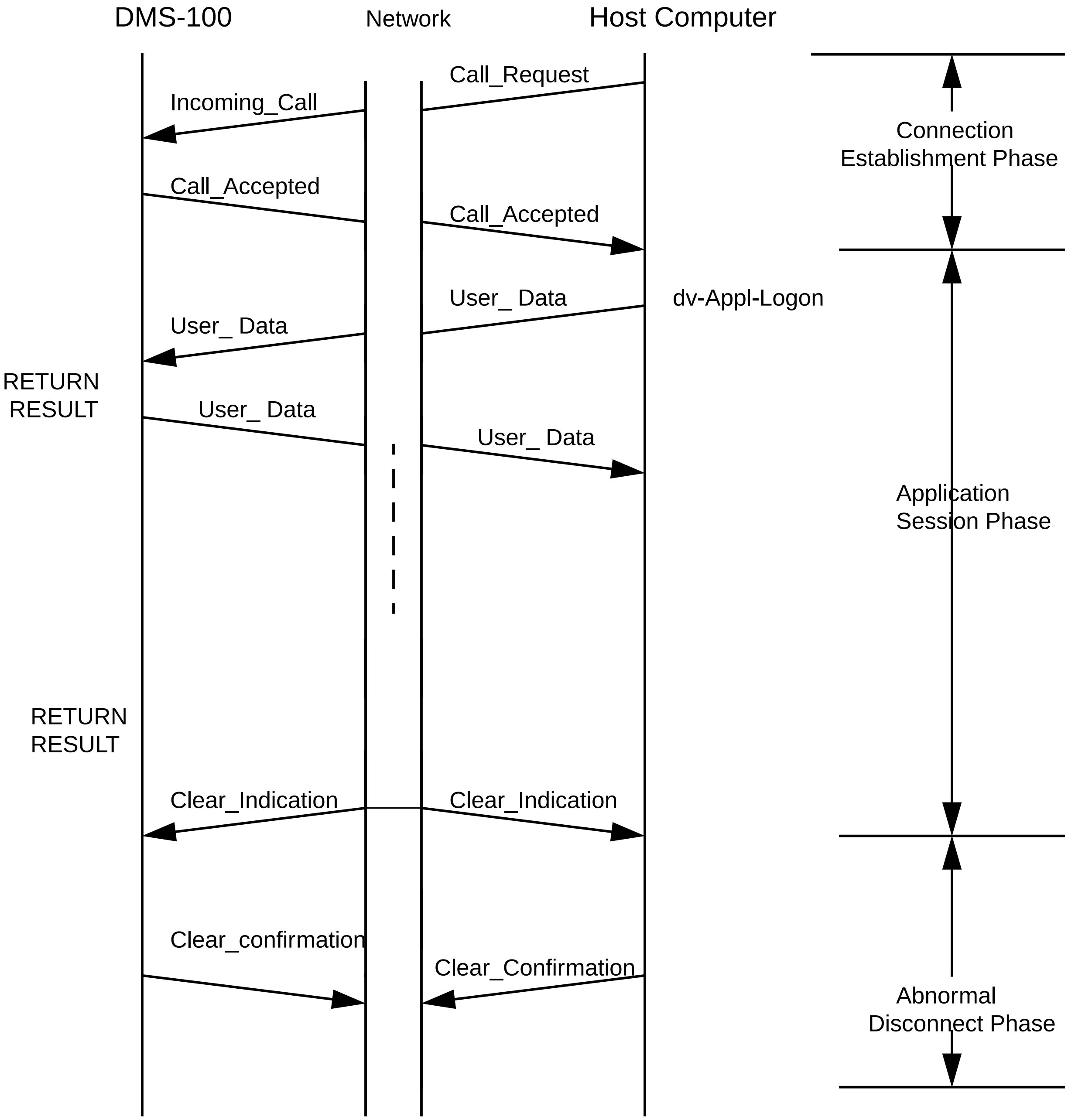
- An invalid dv-Appl-Logon with missing or invalid parameter produces a RETURN-ERROR. The DMS-100 switch sends a REJECT “unrecognized operation" message after receiving a dv-Appl-Logon message with a protocol error. The DMS-100 switch does not terminate the existing X.25 connection and does not send a Clear_Request packet.
- A dv-Appl-Logoff message from the host without an active application level session causes a RETURN-ERROR message with reason NotLoggedOn.
- A dv-Appl-Logon message (in a X.25 data packet) to establish a link at the application level while the link has already been established causes the switch to send a RETURN-ERROR message, not a Clear_Request packet (the existing X.25 connection is not terminated).
- A Clear_Request from the switch is used for taking down the SVC, i.e., terminating the X.25 connection. The host application does not receive messages until the DMS-100 switch reestablishes the layer 3 X.25 connection.
- Procedural errors in X.25 during data transfer results in resetting or clearing of a link by the network. In addition, if no network is involved, either DMS-100 switch or the host may reset or clear. The reset procedure re-initializes the flow control over the link to the state it was in when the virtual channel was established. If the DMS-100 switch detects a link failure, subsequent messages are routed over alternate links in the linkset. This results in the loss of messages in transit on the network. Once an alternate link is identified, subsequent messages associated with the existing call/function are transmitted over this route even though the original link may have returned to an application-level service. Messages pertaining to subsequent calls/functions can use the re-serviced link.
- If a Clear_Indication X.25 packet is received by the DMS-100 switch during an active link connection, it confirms the clear and terminates the link at the application level. The application session is terminated only upon termination of the last link in the linkset supporting the session.
- If a link is abnormally terminated without a dv-Appl-Logoff message, a failure alarm may be raised upon exceeding the alarm threshold. An alarm threshold is calculated by the number of links in-service in the linkset which supports a given session.

Figure 36 X.25 and application layer interworking - normal procedure



Note: Each link in a linkset which supports an application session must be established at the X.25 level, i.e., Call_Request sent for each link. In addition, each link must be established at the application level, i.e., the host sends a dv-Appl-Logon message for each link. The first successful link established also establishes the application-level session. Similarly, in the termination of the session, each link in the linkset must be disassociated.

Figure 37 X.25 network clearing the call - abnormal procedure



3.3.3 Interworking with Layer 4 (TCP)

The application dialogue is performed using the ROSE components described in Section 1.11, “Application layer protocol formats,” on page 61. An application level message is contained in a ROSE component and is carried within a TCP segment when TCP/IP transport is used.

Normal procedures Figure 38, “TCP/IP and application layer interworking: normal procedure,” on page 110, shows the message sequence between the host and DMS-100 switch during a normal procedure.

- The host initiates a Layer 4 TCP connection before starting the CompuCALL session. The connection request is made with the IP address of the DMS-100 switch where CompuCALL is subscribed. The TCP Port number 2500 and the initiation mode is Active. The operating company provides the IP address of the DMS-100 switch on subscription of CompuCALL.
- The TCP `PUSH' option sends all application messages after the connection is successful. This is to ensure that only one application level message is carried in a TCP segment. TCP does not take care of message boundaries and can send multiple messages in the same transmit segment as a sequence of bytes.
- The host computer sends a dv-Appl-Logon message as the first message after a successful TCP connection establishment, specifying the linkset name. The customer provides the linkset name upon subscription to CompuCALL. For TCP, there is no association to the linkset when the Layer 4 connection is established.
- In response to a dv-Appl-Logoff message from the host, the switch returns a RETURN-RESULT acknowledgment. It closes the Layer 4 connection.

Abnormal procedures

When TCP/IP is used as a transport for carrying application level messages, each application level session is associated with only one TCP connection. Therefore any abnormal termination of the Layer 4 connection brings down the application level session.

- An invalid dv-Appl-Logon with missing or invalid parameter produces a RETURN-ERROR. A dv-Appl-Logon message received with protocol error (or any invalid application level message except dv-Appl-Logoff) on the TCP connection while the host is not logged on causes the switch to send a REJECT "unrecognized operation" message. The existing Layer 4 TCP connection is not terminated.
- A dv-Appl-Logoff message from the host without an active application level session causes a RETURN-ERROR message with reason NotLoggedOn.

- Any subsequent dv-Appl-Logon message after the first one resulting in a successful application level session causes the switch to send RETURN-ERROR message. The existing Layer 4 connection is not terminated.
- Procedural errors in TCP/IP during data transfer may result in a clearing of the TCP connection by Layer 4 of either DMS-100 or the host computer, but not the network. This may result in the loss of messages in transit on the network or those in the Layer 4 transmit buffers.
- If the Layer 4 connection is abnormally closed by the host computer during an active application level session without sending the dv-Appl-Logoff message, the application session also gets terminated.

Figure 38 TCP/IP and application layer interworking: normal procedure

The TCP sends the FIN segment to close a Layer 4 connection.

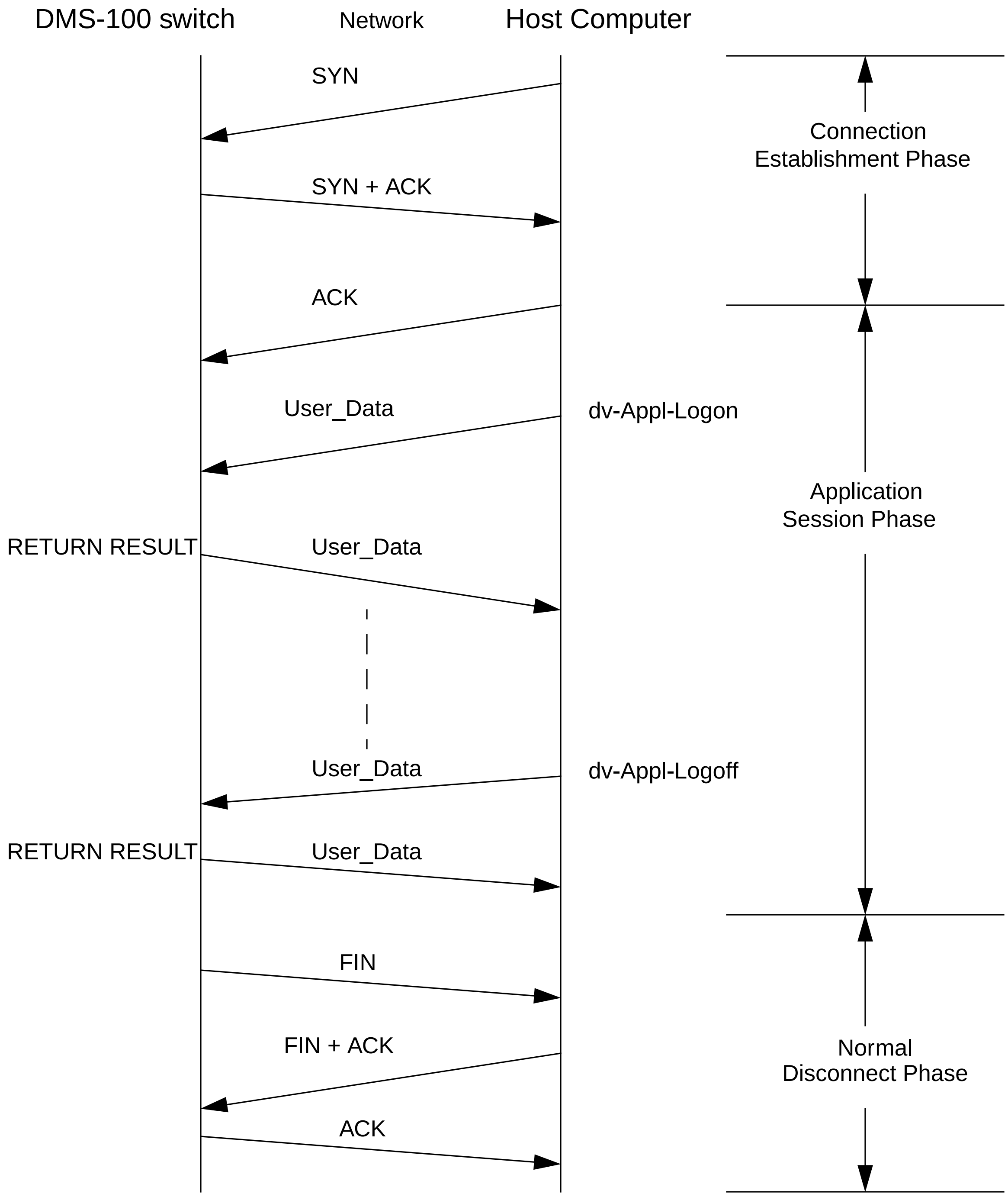
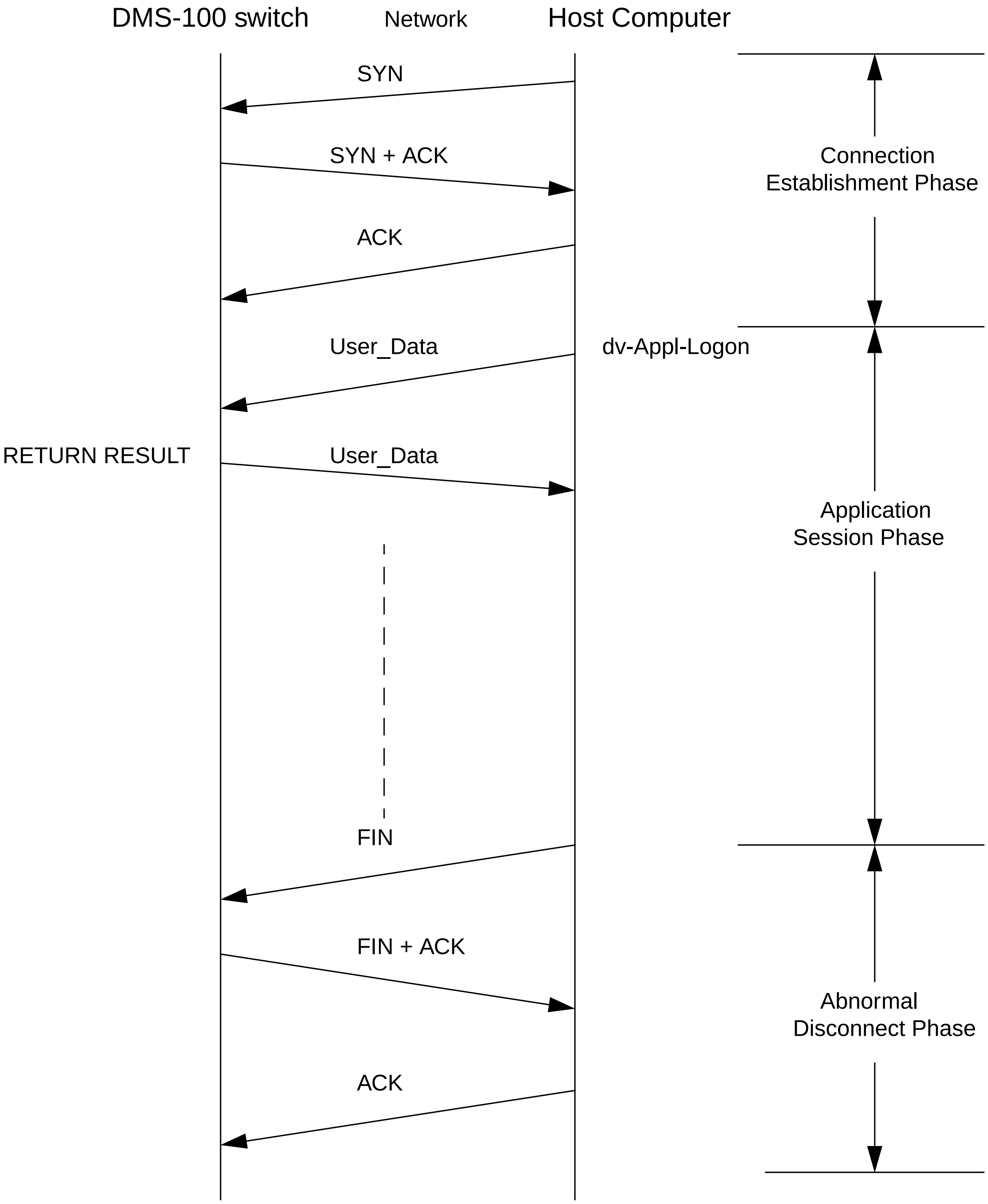


Figure 39 Clearing TCP connection - abnormal procedure

The application layer session is automatically terminated when the Layer 4 TCP connection is abnormally closed.



3.3.4 dv-Appl-Logon restrictions

3.3.4.1 Single password per customer

All host applications belonging to the same customer must use the same password in the dv-Appl-Logon message for a given switch. In other words, the logon password is associated with a customer rather than an individual host application.

3.3.4.2 Single password BusinessGroupID per customer

All host applications belonging to the same customer must use the same BusinessGroupID in the dv-Appl-Logon message for the DMS-100 switch. There are eight linksets for each BusinessGroupID.

3.3.4.3 Session establishment during active call

If the switch receives a call before the customer establishes a CompuCALL session to create the CompuCALL environment during the call, subsequent messages regarding that call may be sent. CompuCALL information may be incomplete based on the point of entry into the call for which the session was established. The incomplete information may include missing parameters and different call identifiers for parties of the same call.

3.3.4.4 Session re-establishment

Anytime a CompuCALL application session is taken down, the switch does not maintain the existing session data. The customer must establish a new session and reassociate the DNs to the new session. Therefore, it is recommended that the host delete the existing call-related information belonging to the previous session.

3.3.4.5 Link re-establishment

Anytime an SVC connection is taken down, the switch will not maintain the existing link data. The host must establish a new connection for the SVC and re-logon to the session. Therefore, it is recommended that the host application delete the existing call-related information belonging to the previous link. The DMS-100 switch can possibly lose messages during this event.

3.3.5 dv-DN-Associate restrictions

3.3.5.1 Primary ACD DNs in dv-DN-Associate message

The host can specify only Primary ACD DNs to be associated with a dv-DN-Associate session in dv-DN-Associate messages (it can’t associate individual Supplementary ACD DNs with a session).

3.3.5.2 Single DN per dv-DN-Associate message

The dv-DN-Associate message can contain only a single DN. This doesn’t preclude a host from specifying more than one DN to be associated with a session.

3.3.5.3 Define operations nonACD DN

The nonACD DN associations remain unchanged. For nonACD DN associations, the DN Operation "Define" is not allowed.

3.3.5.4 DN association during active call

If a call is received by the switch when the customer has an active session, the host uses the dv-DN-Associate message to associate the DN for that call with a given session after the call is received, subsequent messages are sent. CompuCALL information is incomplete based on the point of entry into the call for which the DN is associated. Any new calls after dv-DN-Associate has valid parameters and messages.

3.3.5.5 DN association with one session only

A given DN can be associated with only one session at any given time. This means two host applications in the same or different physical hosts (or even the same host application) with simultaneous CompuCALL application sessions established with a given switch cannot have the same DN associated with the two sessions at the same time. However, different DNs can be associated with each session.

3.3.5.6 DN validation of the DN against the MDC customer group

The customer group for BusinessGroupID must match the MDC customer group to which the DN is assigned. However, the BusinessGroupID is not validated against the customer family group which the MDC customer belongs to. Therefore, the DNs associated with a given application session must all belong to the same MDC customer group and cannot be spread across customer groups in the same customer family group.

3.3.5.7 MDC line DN-association

Only business lines with single DN appearance are supported. MDC/RES lines with CallHold (CHD) feature on the DMS-100 switch are not supported. (The CHD feature applies to only 2500 sets.)

3.3.5.8 Subscription after association

If a new ECM sub-option is subscribed for a particular line after the line has been associated, then any of the new functionality subscribed to will be immediately available for use. If call events have been subscribed to, and the host has associated the DN, then if the DN subscribes to the SCAI Message Waiting indication, then the host will automatically begin to receive the MSGWAIT indication event message for that DN as it occurs. This means that the host need not disassociate then associate the line, nor will the host need to logout then login again.

3.3.5.9 Single DN appearance

The association, SCAI call control and event messages are only applicable to a single DN appearance per line. For example, features like MADN and huntgroups in which a single DN is assigned to multiple lines are not supported by this activity.

The DV_DN_ASSOCIATE message allows the computer to inform the switch the DNs about which it expects to receive event messages. The capabilities of the DV_DN_ASSOCIATE messages support MADN SCA DNs. The procedures that implement the association establish the ICM infrastructure fro a request to associate a MADN SCA DN.

A Centrex or residential line can only be associated if the ECM option has been assigned to a line. ECM and CHD (Call Hold) options are incompatible in SERVORD. Huntgroup lines cannot be associated by a host computer. However, assigning the ECM option to huntgroup lines is not blocked in the IBNLINES and KSETLINE tables.

3.3.6 Limitations

3.3.6.1 Associated DN limit

Before NA013, for ACD group association, a maximum of 20 ACD groups can be associated to the same host application on one CompuCALL session at any time. For NA013, and later software releases, up to 100 ACD DNs can be associated to the same host application on one ICM session. The maximum number of nonACD DNs that can be associated to the same host application on the same CompuCALL session is 20 000.

3.3.6.2 Associated session limit

A switch can support a maximum of 271 CompuCALL sessions. The maximum sessions based on protocol are as follows:

- X.25 sessions = 255
- TCP sessions = 16

3.3.6.3 Linkset limitations

The maximum number of links (SVCs) per linkset is 8.

3.3.7Feature interactions

3.3.7.1Teen Service

The Teen Service feature allows assigning one primary DN to a line, then multiple secondary directory numbers (SDN) may be assigned to that same line in order to differentiate calls coming in to the one line to different people living at that residence. If calls are terminated to the SDN of the Teen Service feature, the associated call event messages will be sent for that call specifying the SDN as the terminating device ID. Only the primary DN needs to be associated (the SDNs cannot be associated).

Enhanced Teen Service is not currently supported.

3.4 OA&M Application service options

The CompuCALL OA&M service lets the operating company offer customer service for the established session on CompuCALL. This OA&M application service can be provided with any applications services in Chapter 4. It must be in conjunction with the Session Management Application Service, Section 3.2 in this chapter.

OA&M enables a host or switch initiates application-to -application level continuity test.

3.5 OA&M procedures

This section addresses the DMS-100 switch and host computer procedures for application-level signaling protocol supporting OA&M.

3.5.1 Session OA&M

The remote operation message associated with verifying the application-to-application level session between the host computer and the switch is dv-Appl-Continuity-Test.

3.5.1.1 Applications session maintenance (dv-Appl-Continuity-Test - operation value "3" hex)

This section describes the dv-Appl-Continuity-Test message initiated by either the host or the DMS-100 switch for testing the application-to-application logical connection.

Once a session is established between the host and the DMS-100 switch, the host can verify the application-to-application logical connection with the dv-Appl-Continuity-Test message. The host sends this message to the DMS-100 switch if it has not received any messages for a certain period of time. This allows early detection of a problem. The frequency with which this message is sent is a customer-defined parameter in the host. However, we recommend not sending this message more than once every two minutes to avoid performance degradation on the DMS-100 switch.

The DMS-100 switch can also send this message to the host when the switch operator wants to verify the connection. A switch operator can initiate it manually as a service provisioning and/or maintenance procedure. In addition, it can be initiated regularly on an automatic basis. The interval when this automatic test is performed is datafilled per-switch with the range 0 (no automatic tests initiated) to 720 minutes. If this interval is datafilled as n minutes (n not equal to 0), a continuity test is performed automatically every n minutes on every active application session on the switch.

Any link belonging to a Linkset could be selected to transport the dv-Appl-Continuity-Test message either in the host-to-DMS-100 switch or the DMS-100 switch to host. The response is transported over the same or another link belonging to the linkset. If the automatic test is performed by the DMS-100 switch in one interval, there is no attempt or obligation to transport a dv-Appl-Continuity-Test message over each link in a linkset.

If a switch operator manually initiates the test or the DMS-100 switch initiates the test, the following procedure is used:

1 The operator initiates the session continuity test at the maintenance and administration position (MAP) or the DMS-100 switch initiates the test on an automatic basis.

2 The switch sends the dv-Appl-Continuity-Test message.

3 If the DMS-100 switch receives a positive acknowledgment manually (RETURN-RESULT) within 10 seconds, a "test passed" indication is displayed at the MAP. The test is terminated and no further action is taken manually or automatically.

4 If the DMS-100 switch receives a RETURN REJECT or RETURN ERROR message or does not receive any response within 10 seconds:

a. The test is terminated and “test failed” appears at the MAP if initiated manually. A “test failed” indication is logged if initiated automatically. This is done if it is the fifth time to test.

b. If it is not the fifth time to test, go to step 2.

Note: Under the following conditions, the DMS-100 switch returns a RETURN REJECT message with reason "unrecognized invocation":

3.5.2 dv-Appl-Continuity-Test

The host sends this RO of Operation Class 2 to the switch, or the switch sends this to the host to verify the application-to-application logical connection (session). The response by the switch or host is the same in either direction.

3.5.2.1 dv-Appl-Continuity-Test ASN.1 encoding:

```
dv-Appl-Continuity-Test                  OPERATION
                 RESULT                  ContinuityResult
                 ERRORS                  {not-Logged-On}
::= 3

ContinuityResult         ::=      SEQUENCE
{continuityResultValue[0] IMPLICIT  INTEGER       OPTIONAL}

-- ContinuityResult parameter has the same value as the ApplicationID
```

-- ContinuityResult parameter received in DV-Appl-Logon message for that session

3.5.2.2 dv-Appl-Continuity-Test coded example:

The host verifies the application-to-application logical connection (session) by sending dv-Appl-Continuity-Test with InvokeID = 41 (29 Hex). The host receives RETURN-RESULT from the switch with the parameter ApplicationID (has the same value as the parameter ApplicationID in the dv-Appl-Logon for that session). If the dv-Appl-Continuity-Test is received when no session is logged on, the host receives REJECT with reason "unrecognized operation."

In the DMS-to-Host direction, the DMS verifies the application-to-application connection (session) by sending dv-Appl-Continuity-Test with InvokeID = 41 (29 Hex). The host sends a RETURN-RESULT to the switch with the parameter ApplicationID (has the same value as the parameter ApplicationID in the dv-Appl-Logon for that session). If the RETURN-RESULT received by the DMS contains missing parameters or invalid parameter values, the DMS will send a REJECT indicating the error.

INVOKE DMS <===== HOST

Table 13 dv-Appl-Continuity-Test invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	06		INVOKE	This message is 6 bytes long decimal.
02	01	29	InvokeID	The invoke ID is 41 decimal.
02	01	03	Operation	The operation is dv-Appl-Continuity-Test[3].
Hex Dump=A1 06 02 01 29 02 01 03				

RETURN-RESULT DMS =====> HOST

Table 14 dv-Appl-Continuity-Test return result example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	0B		RETURN RESULT	This message is 11 bytes long decimal.
02	01	29	InvokeID	The invoke ID is 41 decimal.
30	06		Argument	Its argument is of type Sequence and is 6 bytes long.
02	01	03	Operation	The operation is dv-Appl-Continuity-Test[3].
80	01	01	Continuity Result	The application ID is 1.
Hex Dump=A2 0B 02 01 29 30 06 02 01 03 80 01 01				

REJECT DMS =====> HOST

Table 15 dv-Appl-Continuity-Test return reject example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A4	06		RETURN-REJECT	This message is 6 bytes long decimal.
02	01	29	InvokeID	Tthe invoke ID is 41 decimal.
81	01	01	Invoke Problem	Tthe invoke problem is unrecognized (1).
Hex Dump=A4 06 02 01 29 81 01 01				

RETURN ERROR DMS =====> > HOST

Table 16 dv-Appl-Continuity-Test return errort example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
83	06		RETURN-ERROR	The message is 6 bytes long decimal.
02	01	29	InvokeID	The invoke ID is 41 decimal.
02	01	00	Invoke Problem	This is not logged on.
Hex Dump=83 06 02 01 29 02 01 00				

4.0 Application service options

4.1 Application service overview

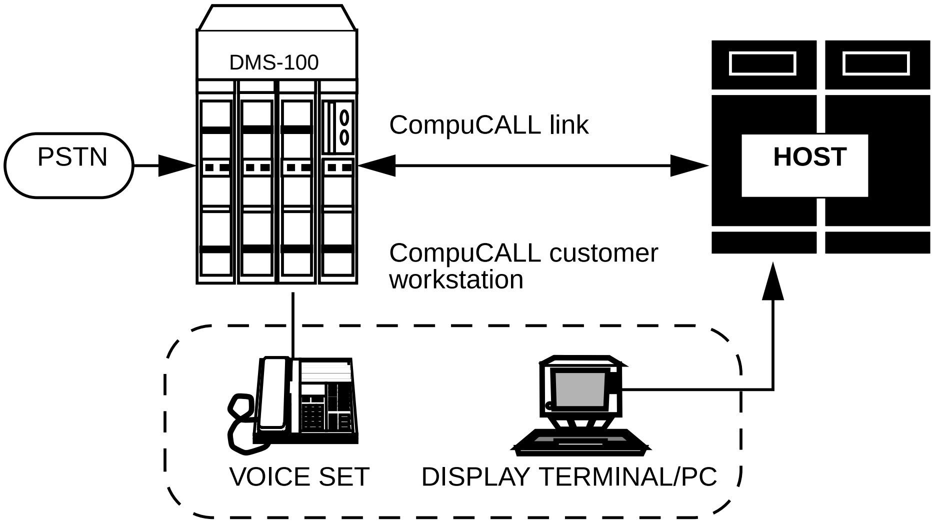
This section describes CompuCALL services from a CompuCALL customer's perspective on CompuCALL application service options. These CompuCALL application service options use coordinated voice and data and can configure the CompuCALL interface to enable a DMS-100 central office switch and an operating company client's host computer to exchange all information regarding incoming and outbound calls. Furthermore, these options let operating company clients configure customized call centers and non-call center applications. For example, you can use Integrated Voice Response (IVR) systems to configure interactive voice and data.

Section 4.4 addresses the application service functions for CompuCALL options. Section 4.5 covers ACD CompuCALL messages. Section 4.5.6.8 covers residential and Centrex CompuCALL messages. Section 4.13 covers message interactions with other CompuCALL messages and ACD feature interactions, MDC feature interactions, RES feature interactions with CompuCALL messages, and CompuCALL restrictions and limitations.

Please refer to Figure 40, "The Meridian CompuCALL application concept," on page 122.

> ***Note:*** A customer work station refers to the physical location occupied by an office worker or an ACD agent equipped with a computer terminal and a voice set.

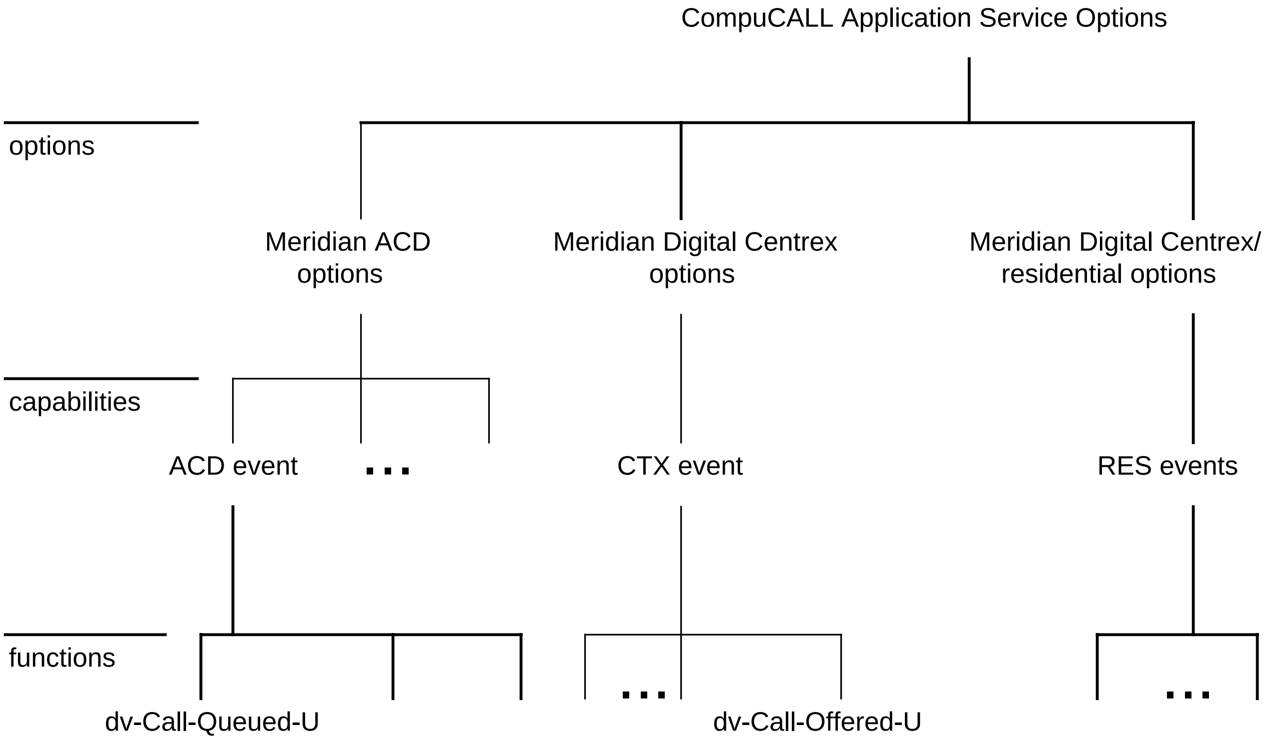
Figure 40 The Meridian CompuCALL application concept



CompuCALL services has three levels. These levels give the CompuCALL customer the ability to control or customize CompuCALL services for the customer’s environment.

1. Options is the highest or broadest level. It categorizes what type of entity CompuCALL’s service controls.
2. Capabilities is the middle level. For each supported option, there exists a set of “actions” that CompuCALL can provide for that option.
3. Functions is the lowest level. For each capability, the CompuCALL action executes a defined interface (CompuCALL message[s]).

Figure 41 CompuCALL Application Service options, capabilities, and functions



4.1.1 Application service options

The following CompuCALL service options are available:

- Meridian Automatic Call Distribution (ACD)
- Meridian Digital Centrex (CTX)
- Meridian Digital Residential (RES)
- Private Branch Exchange (PBX)

Meridian ACD CompuCALL options relate to DMS-100 Centrex lines configured with ACD features and their corresponding ACD groups. These CompuCALL options support Centrex ACD lines and the following voice sets supported by ACD: 500/2500 Meridian Digital Centrex (MDC) sets, Meridian Business Sets (MBS), and ISDN Meridian Feature Transparency (MFT) sets. Both display and nondisplay sets are supported.

Meridian Digital Centrex CompuCALL options relate to DMS-100 lines configured with Meridian Digital Centrex (MDC) business features.

Meridian Digital Residential CompuCALL options relate to DMS-100 lines confgired with MDC residential features.

Both the MDC and MDC Residential CompuCALL options support lines with single DN appearances. Business and residential lines with multiple appearance configurations such as Multiple Appearance Directory Number (MADN), hunt groups, and Universal Call Distribution (UCD) are not supported.

The MDC voice sets supported are the 500/2500 Meridian Business Sets (MBS), Custom Local Area Subscriber Services (CLASS)/Call Management Service Sets (CMS) and ISDN Meridian Feature Transparency Sets (MFT). (Note: MBS is a Nortel Networks proprietary multi-key set.). Both MDC and MDC Residential support display and nondisplay sets. Attendant consoles are not supported.

4.2 Application service capabilities

There is a set of CompuCALL application capabilities defined for each CompuCALL option. CompuCALL uses table SCAISSRV on the DMS-100 switch to determine which CompuCALL options, capabilities, and functions the customer selects.

4.2.1 ACD capabilities

The following are the CompuCALL Capabilities defined for the CompuCALL ACD option:

- Coordinated Voice and Data Delivery (CVD)
- Call routing
- Third Party Call Control
- Resource status
- Third Party Agent Control
- ICCM
- TPQC

4.2.1.1 Coordinated Voice and Data Delivery

Coordinated Voice and Data Delivery (CVD) to the CompuCALL customer work station is the major customer application of CompuCALL. This refers to combining the presentation of the voice call by the switch with the coordinated data display on a PC. The call related information might include the service desired by the caller (based on the OrigInboundDN), previously stored data regarding the caller (based on the OrigCallingNumber), and special call handling instructions such as language (also based on the OrigCallingNumber). The OrigInboundDN and OrigCallingNumber are parameters used in the Application service functions and covered in Chapter 5.0 Application service parameter definitions.

For example, in a call center environment, many ACD agents or office workers at their work stations interact with a host computer to handle incoming telemarketing calls. Presently there is no direct communication between the switch and the host computer. A worker handling calls to an insurance company might follow this procedure:

1 An incoming call is offered to the worker.

2 The worker answers and asks the caller for his home or business telephone number, and why he is calling (auto insurance or home insurance, policy changes or claims).

3 The caller verbally gives the requested information to the agent.

4 The worker enters the information provided by the caller in the host computer via a display terminal or personal computer.

5 The host processes the request and displays the relevant information (caller's auto insurance policy).

6 The worker verifies that the correct information has been displayed and proceeds with the call.

With CVD, steps 2, 3, and 4 can be eliminated in many cases. Furthermore, steps 1 and 5 can occur simultaneously. This is achieved by the switch providing call-related information such as the OrigCallingNumber to the host before a station is connected. This allows the host computer to process the call and display the appropriate information to the agent/worker at about the same time the agent/worker answers the call.

This call-related information can be provided to the host when an agent/worker has been selected to handle the call or when the agent/worker answers the call. In addition, the same information can be provided to the host application as soon as the call has been received by the switch and queued for an agent with a follow-up message sent when an agent has been selected or when an agent answers the call. This provides the host with more time to generate the screen display.

The value to the CVD customer is a significant reduction in call handling time by the agent/worker, creating operational cost savings, plus improved service quality to the caller. This saves time for the caller (and money in the case of a toll call) since it greatly reduces time for the agent/worker to access the appropriate information in the host computer. This also significantly reduces the chances of the host receiving the incorrect call-related information due to human error.

The OrigCallingNumber is not an absolutely reliable/unique piece of information for identifying the caller. Depending on the customer application, the OrigCallingNumber may or may not be useful. However, where a significant number of customers call from either their home or business phones, the OrigCallingNumber can be used by the host computer to identify the caller. This will save significant agent call handling time, even if the OrigCallingNumber is not available/usable for a large percentage of calls. Furthermore, some call center applications require the actual OrigCallingNumber even if it is not the caller's home or business number, for example, to identify where the caller is calling from. Some call center customers may want this information for traffic report generation purposes (800 service planning). If the caller is not calling from his own phone, and the call center customer needs the OrigCallingNumber, the CompuCALL options provide better availability/reliability of the OrigCallingNumber than the caller providing it verbally (calls from pay phones where the number may not be visible or the caller may have difficulty providing it).

When available, the OrigChargeNumber can also be provided by the switch to the host. The OrigChargeNumber is a parameter in a service function covered in Chapter 5.0 Application service parameter definitions. Assigned by the telephone company and in North America, it consists of a three- digit NPA (area code) plus a seven-digit billing number. It is also referred to as ANI - automatic number identification. Unlike OrigCallingNumber, the OrigChargeNumber may not be a dial number and has no indicator as to whether it refers to a single line or a group of lines (PBX, hunt group). However, it may be available for calls where OrigCallingNumber is not available (no end-to-end SS7 signaling) and therefore may be the only network-provided information to identify the caller.

When a call has been forwarded one or more times (call forwarding), the recipient agent could be notified of the first and last forwarding parties and their forwarding reasons. The answering agent could modify the initial response accordingly for the client.

4.2.1.2 Call routing

Meridian ACD CompuCALL also provides the host computer with "Call Redirection" - the ability to reroute an ACD call to another ACD group or agent/worker (primary or supplementary DN of an ACD group, specific ACD or non-ACD line on the same switch or any ACD group or non-ACD line on a different switch). Also, ACD customers may use call routing to complement call event reporting. In the previous example of ACD calls to an insurance company, call routing might be used as follows:

1. The switch informs the host computer of an incoming ACD call and provides the appropriate call-related information.
2. The host determines from the ACD DN that the call regards auto insurance.
3. If the host can locate the caller's file, it instructs the switch to proceed with connecting the caller to an agent in the ACD group associated with the

number dialed by the caller (this ACD group only handles "standard" calls, calls from existing customers where the host computer can locate the caller's file based on the OrigCallingNumber and/or OrigChargeNumber provided by the switch).

4 The switch informs the host when an agent has been selected to handle the call, and/or when the agent answers the call.

5 The host displays the caller's file (auto insurance policy) on the appropriate agent's display terminal or personal computer.

6 If the host computer cannot locate the caller's file based on the network-provided information and concludes that the caller is

a. not an existing customer

b. not calling from his home or business phone, or if no caller information is available, it instructs the switch to redirect the call to another ACD group or another line which handles "non-standard" calls. The target individual at the destination could be an ACD agent or an office worker.

7 The switch informs the host when an agent/worker has been selected to handle the call, and/or when the agent/worker answers the call.

8 The host displays the appropriate call-related information on the agent's or worker's display terminal or personal computer ("auto insurance call, OrigCallingNumber available but caller file not located").

The host’s ability to request the routing of a call to a specific ACD agent or office worker enables incoming calls to an active ACD group to be routed to specific ACD agents and workers as a result of customer follow-up, detailed inquiries, incoming call priority, etc.

4.2.1.3 Third Party Call Control

Third Party Call Control (TPCC) lets the host computer send the switch signaling commands on behalf of a specific directory number. TPCC significantly reduces agent call handling time and dialing errors creating operational cost savings. The host can also provide a value-added service, for example, the caller may not require the DN of the party to be called.

TPCC may be used for:

- Outbound Call - Preview Dialing
- Consult/Transfer Call
- Consult/Conference Call

Each of these applications is expanded below. TPCC is highly synergistic with CVD. The host can use the OrigCallingNumber in the customer's file (earlier provided by the switch to the host in CVD over the CompuCALL link for incoming calls to identify the caller) to initiate an outbound call ("callbacks"). Also, TPCC enables call transfer or conference to be requested via the host and the call-related data information can be provided simultaneously as the voice call due to CVD. TPCC calls, once established, are handled by Meridian ACD the same way as incoming calls are handled (in case of call transfer).

TPCC is valuable since agents no longer waste valuable time obtaining customer DNs and dialing outbound calls manually, thus significantly reducing call handling time and dialing errors, and creating operational cost savings.

4.2.1.4Outbound Call - Preview Dialing

Preview Dialing is a TPCC capability allowing an outbound call on behalf of the agent. Today, outbound calls are typically initiated by an agent who either has a hard-copy list of DNs, or accesses such a list from a host computer using the data terminal and then must dial each DN using the voice set. TPCC lets an agent initiate an outbound ACD call via the data terminal. The host uses the CompuCALL link to ask the switch to initiate the call. The switch then tries to place the call immediately after verifying that the request is valid. The verification may include an authorization code and/or an account code in the request. The Outbound Call feature may also be enhanced by providing a distinctive ringing tone to alert the agent of originating an outbound call. Furthermore, enhanced security may be provided when initiating outbound calls (authorization and account codes must be provided by the agent to control the agent's access to outbound calls).

For example, an ACD agent in a service company might be required to contact customers after a problem has been fixed to make sure the customer is satisfied. The agent might:

1 obtain directory numbers of clients to be contacted on a hard-copy list or on a list prepared by the host computer

2 search for the appropriate client directory entry

3 copy or record the client's DN for calling

4 contact client by manually dialing from his voice set via the secondary DN

5 waits for client to answer.

However, if an ACD agent makes an outbound call using TPCC, the preceding scenario may change as follows if the agent with an MBS set:

1 brings up a list of completed service reports on the data terminal

2 scans displayed file information and moves cursor to select the customer's service report

3 presses the "DIAL" key on data terminal to initiate the call to the customer

4 is alerted on the INCALLS key of the voice set that an outbound call is being made on his behalf

5 goes off-hook or depresses the INCALLS key and the call to the customer proceeds

4.2.1.5 Consult/Transfer Call

Consult/Transfer Call is a TPCC capability which allows the connection to a consult party (see Section 4.7) being established for consult or transfer.

Consult Call is a powerful feature which enables fast and efficient incoming or outgoing call distribution by an agent to other ACD groups or agents. For instance, an agent currently uses the telephone to call for expert assistance. To do this, the agent may rely on a hard copy list of DNs. However, the Consult Call feature allows the agent to activate the consult features from the data terminal potentially using the computer database to choose the right expert for the call. Thus, agent time efficiency is improved by operating from the data terminal from both the perspective of using one interface for both caller information and call control and the perspective of not requiring an up to date DN. Different interfaces interrupt the agent's flow of operation. If agents often use consult calls, this will save much time capability).

The following example illustrates the situation when the agent has put the original call party on hold and is consulting (with intent to conference) a third party, then the agent requests to conference all existing parties. Currently, the agent might need to do the following:

1 get a directory of other agent numbers and ACD groups either on a hard copy list or a list prepared by the host computer

2 search for the appropriate agent directory entry

3 copy the other agent's DN for calling

4 request establishment of consult call by enabling the 3WC button on the set

5 key in the destination DN manually on the voice set

6 listen to call progress tones on the set and wait for an answer to consult with the called party. Then the agent enables the "conference" function and all three parties are conferenced into the call.

7 After a conference call is established, if the agent wants to talk to only the original call party without the consult party, the agent depresses the conference function on the voice set again to drop the consult party. Then the call returns to a two-party call. If, however, the agent wants to drop out using the RLS key or the Not Ready key (on MBS) or on-hook (on

2500 set), the call is transferred and remains active between the original call party and the previous consult party.

However, if an ACD agent makes a conference call with TPCC, the scenario may be:

1. The agent brings up a directory of other agents or ACD groups on the screen.
2. The agent scans the directory and selects an entry for the consult party.
3. The agent selects the "Add/Consult " function on the data terminal.
4. The agent listens to the call progress tones on the set and waits for an answer to consult with the called party.
5. If the agent wants to conference the call after consulting, the agent enables the "Conference -Party" function on the data terminal.
6. After the conference call is established, if the agent decides to return to the original call party, then the "Drop Party" function is selected on the data terminal to drop the consult party. Or the agent may activate the "Transfer Party" function to transfer the call.

4.2.1.6 MDC/RES Third Party Call Control

Add/drop/transfer/conference party messages will only be supported for single line appearance directory numbers (DNs). Thus an origination address specifying a MADN DN, UCD DN, ACD DN, or Teen Service SDN will result in a return error indicating an invalid origination address.

The new SCAI application logon service version of SCAI07 will be adhered to for these messages. That is, if a host logs on with a service version that existed prior to our new service version, then the residential or Centrex originated add/drop/transfer/conference request will be denied as it always has been for those previous versions.

CAS (Computer Aided Signalling) Consultation/Conference/Transfer has the following capabilities, which closely mimics the existing MDC 3WC/CXR offering:

- Consult (Add) - If a residential or Centrex line is involved in an active call (two or more parties) the host may request that a consult leg be established. To do this, the party talking to the line is put on hold, and a call is placed to the destination specified by the host computer. The RES/Centrex line (also known as the SCAI controller) is now connected to the consult party but cannot talk to the party on hold.
- Transfer - Given the scenario above, the host may transfer the call. This would drop the SCAI controller from the call and create a call between the held party and the consult party. The consult leg need not have answered

the call to transfer it. Effectively, this means that the host is also able to do a blind transfer. A call may also be transferred following a conference party request; this would effectively drop the SCAI controller from the conference call.

- Drop - Given the consult leg scenario, this allows the consult leg of the call to be released. The consult leg need not be in the talking state to be dropped. Drop may only be used to release the consult party during a conference call.
- Conference -If the residential or Centrex line (the SCAI controller) is involved in a consult call via the CAS consult feature (the initial two party call is on hold, and the controller is talking to the destination specified by the host computer), the host may request that a conference call be established. All three parties will now be connected to each other, resulting in a three-party conference call. The consult leg need not have answered the call to conference.

An extended call via add party may not be controlled via the 3WC or Call Transfer set features. Once the call has been extended via add party, if the user wishes to transfer the call, conference all three parties, or drop the consult leg, the user must initiate the request via CompuCALL. If the user hits the release key or hangs-up after an add-party request and before any other actions have been taken, the consult leg will be dropped. If the user SCAI controller should release the call after a conference has been initiated, all parties will be dropped from the three-way call. If a 3WC or call transfer has been activated via the set, it cannot be controlled via CompuCALL.

SCAI 3WC/Call Transfer and set-based 3WC/Call Transfer are not interchangeable on a per call basis. This does not imply that a set controlled/initiated 3WC cannot be chained with a SCAI controlled/initiated 3WC: the two ways of creating 3WC's may be chained.

The host sends the dv-Add-Party message to the switch when an user has initiated an outgoing Centrex or residential consult-leg call from their associated data terminal. It will be the responsibility of the host to ensure that these requests are only allowed on sets requested by the owner of those sets.

Following the add party request, the host may then request a transfer party, conference party or drop party request. The add party request will set up the originator specified in the origAddress parameter as the SCAI controller of the CompuCALL initiated 3WC/Call Transfer. Once the consult leg call has been set up by a SCAI controller, only the SCAI 3WC messages may be used to transfer or conference the call.

If after the dv-Add-Party has been accepted and a consult call has been established, the user can hit the RLS key or go on-hook on a 2500 set. In both of these cases the SCAI controller will be re-rung and on answer will be reconnected to the held party of the first leg. These four capabilities (provided for Centrex and residential lines by the Remote Operations (RO): dv-Add-Party, dv-Transfer-Party, dv-Conference-Party and dv-Drop-Party) will be built on top of 3-Way-Call and Call Transfer. This means that this activity will provide the same functionality but through a different interface. The existing 3WC/Call Transfer restrictions will apply as well as any imposed by the ECM application.

In order for a host computer to utilize these messages, the host computer must be logged into a SCAI session. The user's DN that the host is doing these requests for need not be associated with the SCAI session, but it must belong to the same customer group as the SCAI session. However, if the DN is DN-associated, then the host will also receive the messages associated with the call events that it has subscribed to.

In addition, the message(s) will need to have been subscribed to by both the host SCAI session and the user (via the ECM line option).

4.2.1.7 Resource status

Meridian ACD CompuCALL allows the host computer to request the switch status and resource information of a specific ACD group. The information provides a snapshot of the ACD group's status, agent availability, CDN status and call queuing status. The benefit of resource status is that the information can provide a significant opportunity for a company to improve customer service and increase revenues. Resource status can be used with CVD, call routing, and TPCC capabilities to create powerful hosts. Benefits and examples follow:

- Call screening through IVR
  - — lets an IVR announce the estimated wait-time to the caller before transferring the client to an agent.
  - — presently, calls directly to the ACD group are treated depending on the status of the ACD group—the caller lacks control
  - — lets the call route to the IVR to receive an announcement on the wait-time so the caller can wait or call back
- Intelligent call routing
  - — The host can decide on call routing applications knowing an ACD's group state (accepting call, overflow, etc.), call queue sizes, and agent availability to handle new calls
  - — New incoming calls for an ACD group may be redirected (call routing) to another ACD group or DN when the originally called ACD group may not provide efficient service to customers.

For example, calls for an ACD group in overflow state might be redirected to another ACD group consisting of agents handling overflow calls.

4.2.1.8 Third Party Agent Control (TPAC)

TPAC lets the host computer send the switch signaling commands on behalf of a specific agent position to request agent state changes (Agent LogIn, LogOut, Ready and Not Ready). If an agent position changes state, the switch reports an event to the host. These events may also be reported to the host upon successful change in state for an agent position via an agent's manual instigation. Status changes are reported through TPAC event messages only if the ACD group is associated with a CompuCALL session with the TPAC capability. A brief description of these agent position statuses follow:

- Agent LogIn - agent brings his telephone set (agent position) into a state of availability to perform ACD functions
- Agent LogOut - agent takes his telephone set out of a state of availability to perform ACD functions
- Agent Ready - agent places his telephone set in a state to receive ACD calls
- Agent Not Ready - agent takes his telephone set out of availability to receive incoming ACD calls.

TPAC significantly reduces the agent's call handling time, thus creating operational cost savings and improved service quality. This is especially apparent when the agent performs follow-up work with a current call. The agent would activate an agent feature (Not Ready) to remove his position from the pool of available agents serving new calls. If this action were performed on an MBS set, the agent presses the Not Ready key. On a 2500 set, the agent must put the current call on hold, activate the Not Ready feature via an access code, and wait for the feature confirmation tone before returning to the call. Hence, TPAC improves service quality and common interface for different types of telephone sets for ACD service. TPAC also improves human interface over the inbound signaling (via keypad) method for agents. The host can also provide value-added services (use agent availability activity to analyze agent performance).

Event reporting of agent position status changes compliments TPAC. Logon, Logout, Ready, or Not Ready are reported to the host when an agent position enters the state, whether instigated manually or through TPAC. Agent event information provides details for applications in the operations, administration, management, and provisioning of agents. Also, call center service quality can be measured using the details in agent event reporting.

The following is a description of the Agent LogIn/Logout and Ready/Not Ready processes for ACD agents on an MBS set, a 2500 set, and through CompuCALL TPAC.

Agent LogIn

An ACD agent position on an MBS requires the Make Set Busy (MSB) key on the MBS set:

1 The agent depresses the MSB key. Its associated Liquid Crystal Display (LCD) lamp is activated.

2 The agent depresses the InCalls key (ICK). Its associated LCD lamp is activated and the agent hears a dial tone which is a signal to input the agent LogIn identification code.

3 The agent keys in the LogIn identification code.

4 The agent may be prompted to key in a password by the signal of a special tone.

5 If LogIn is successful, the agent position is placed in the Not Ready state (if available). If the Not Ready feature is not available, the agent position will automatically be placed in the Ready state and receives incoming ACD calls.

If the LogIn process was not successful, the agent hears a negative acknowledgment treatment.

An ACD agent position on a 500/2500 set requires the ACD LogIn activation code on the 500/2500 set:

1 The agent picks up the telephone receiver as if originating a call and depresses the ACD LogIn activation code.

2 The agent hears a special tone which is a signal to key in the LogIn identification code.

3 The agent may be prompted to key in a password by the signal of a second special tone.

4 If LogIn is successful, the agent hears a different tone (confirmation tone) and the agent position is placed in the Not Ready state (if available). If the Not Ready feature is not available, the agent position will automatically be placed in the Ready state and receives incoming ACD calls. This state occurs when the agent completes LogIn by hanging up the telephone receiver.

If the LogIn process was not successful, the agent hears a negative acknowledgment treatment.

If an ACD agent performs the LogIn process using TPAC, the above example might look as follows (there is no difference in the process based on the agent position (MBS or 500/2500):

1 The agent keys in the Login identification code and password (if applicable) in the LogIn template at the data terminal.

2 The agent enters the agent login request on the data terminal (presses ENTER).

3 If LogIn is successful, confirmation may be provided to the agent at the agent's data terminal. The agent position will be placed in the Not Ready state (if applicable). If the Not Ready feature is not available, the agent position will automatically be placed in the Ready state and receives incoming ACD calls immediately.

If LogIn is not successful, the agent hears negative acknowledgment at the data terminal.

For agent LogOut, an ACD agent position on an MBS requires the MSB key on the MBS set. The agent position may be in any agent state. The events follow:

1 The agent depresses the MSB key. Its associated LCD lamp is activated.

2 If LogOut is performed while an agent is active on a call, the agent position is logged out after the call is completed.

3 If LogOut is performed while an agent is idle, the agent position is logged out immediately.

> ***Note:*** If the LogOut process is not successful, the associated LCD is not activated.

An ACD agent position on a 500/2500 set requires the ACD logIn deactivation code on the 500/2500 set:

1 The agent picks up the telephone receiver as if originating a call and depresses the ACD logIn deactivation code. The agent hears a confirmation tone and the agent position is placed in the LoggedOut state.

2 If LogOut is performed while an agent is active on a call, the agent position is logged out after the call is completed (agent position is on hook).

3 If LogOut is performed while an agent is idle, the agent position is logged out immediately (agent position is on hook).

> ***Note:*** If the LogOut process fails, the agent hears a negative acknowledgment treatment.

An ACD agent performs LogOut with TPAC changes as follows (there is no difference in the process based on an MBS or 500/2500 agent position):

1 The agent enters agent logout request on data terminal (depresses ENTER).

2 If LogOut while an agent is active on a call, the agent position is logged out after the call is completed.

3 If LogOut while an agent is idle, the agent position is logged out immediately.

> ***Note:*** If the LogOut process was not successful, the agent receives a negative acknowledgment at the data terminal.

Agent Not Ready Activation (agent position state change to "not ready")

An ACD agent position on an MBS requires the Not Ready (NR) key on the MBS:

1 The agent depresses the NR key. Its associated LCD lamp is activated.

2 If the Non-Immediate Cutoff feature is available, the agent maintains the call. The agent position is placed in the Not Ready state when the call is completed. Otherwise, the call is terminated immediately and the agent position is placed in the Not Ready state.

3 The agent may be prompted to key in a walkaway code.

An ACD agent position on a 2500 set requires the ACD Not Ready (NR) Activation code on the 2500 set:

1 The agent invokes a feature activation request and keys the NR activation code.

2 If the agent is involved in a call, the call is maintained and the agent position is placed in the Not Ready state when the call is completed (agent position is on hook). (The Non-Immediate Cutoff feature-equivalent is standard for the Not Ready activation on 500/2500 sets.)

3 If the agent is not involved in a call, the agent position enters the Not Ready state immediately (agent position is on hook).

> ***Note:*** The Walkaway feature is not applicable to agents using 500/2500 positions.

ACD agent performing Not Ready activation with TPAC changes is as follows (there is no difference in the process based on an MBS or 500/2500 agent position):

1 The agent keys in the walkaway code (if required) into the Not Ready Ttemplate at the data terminal.

2 The agent enters the Agent Not Ready request on the data terminal (depresses the button) while still involved in the call.

3 If the agent is involved in a call, the call is maintained and the agent position is placed in the Not Ready state when the call is completed.

4 If the agent is not involved in a call, the agent position enters the Not Ready state immediately.

Note: Walkaway applies to agents on 500/2500 and MBS positions alike.

For the Agent Not Ready Deactivation (agent position state change to ready) an ACD agent position on an MBS requires the NR key on the MBS set:

1 The agent depresses the NR key. Its associated LCD lamp is deactivated.

2 The agent position is placed in the Ready state immediately.

An ACD agent position on a 2500 set requires the ACD NR deactivation code on the 2500 set:

1 The agent keys the NR deactivation code.

2 The agent position is placed in the Ready state immediately.

ACD agent performing Ready activation with TPAC changes as follows. There is no difference in the process based on an MBS or 500/2500 agent position.

1 The agent enters the Agent Ready template at the data terminal.

2 The agent position enters the Ready state immediately.

4.2.1.9 Integrated Call Center Manager (ICCM)

Integrated Call Center Manager (ICCM) supports dv-Set-CDN-State. A CDN is a controlled directory number, that holds calls until the host computer sends routing instructions. The host computer can change the state of the CDN to controlled, revert to default or default with dv-Set-CDN-State message. Initially all of the CDNs are in the default state.

4.2.1.10 Third Party Queue Control (TPQC)

TPQC supports the following messages:

- dv-Route-Call
- dv-Give-Treatment
- dv-Treatment-Complete-U

TPQC contains the routing message dv-Route-Call. TPQC includes the function ROUTECALL . The host computer uses dv-Route-Call to route a call in an ACD queue.

TPQC supports dv-Give-Treatment. If an ACDDN is associated to a Call Center and a call terminates at that DN, the switch sends a dv-Call-Queued-U message to the host computer. Using the information contained within that message, the host computer (call center) can request a treatment application to that call. These treatments are applied prior to the call being removed from the queue, routed to an agent, routed to another ACDDN, or receiving datafilled treatment.

The call event message dv-Treatment-Complete-U notifies the host computer when a requested AUDIO route is completed. The message is not sent unless the host computer associates the ACD DN through the dv-DN-Associate host to switch message

4.2.2 ACD application service capabilities and functions

Table 17 list the application service functions of the Meridian ACD CompuCALL options and application service capabilities.

Table 17 ACD Table SCAISSRV Capabilities and Functions

Service capabilites	Category	Function	Message
ACD Events also referred to as CVD	ACDEVENT	AGTLGDIN	dv-Agent-Logged-In-U
		AGTLGDOUT	dv-Agent-Logged-Out-U
		AGTNREADY	dv-Agent-Not-Ready-U
		AGTREADY	dv-Agent-Ready-U
		CALLANSWR	dv-Call-Answered-U
		CALLOFFR	dv-Call-Offered-U
		CALLREL	dv-Call-Released-U
		CALLQUED	dv-Call-Queued-U
		AGTSETACT	dv-Agent-SetAction-U
		LOBEVENT	dv-LOB-Event-U
		EMKEVENT	dv-EMK-U
Integrated Call Center Manager	ICCM	SETCDNST	dv-Set-CDN-State
		REASNAGT	dv-Reassign-Agent
Third Party Queue Control	TPQC	ROUTECALL	dv-Route-Call
		TRMTCOMP	dv-Treatment-Complete-U
		GIVETRMT	dv-Give-Treatment
Call Routing	ROUTING	CALLRECDC	dv-Call-Received-C
		CALLREDIR	dv-Call-Redirect
Third Party Call Control	TPCC	ANSWCALL	dv-Answer-Call
		HOLDCALL	dv-Hold-Call
		RELSCALL	dv-Release-Call
		UNHOLDCALL	dv-Unhold-Call
		CALLUNHELD	dv-Call-Unheld
		CONSULTEV	dv-Call-Consult-Originated-U
		CONFEVNT	dv-Call-Conference
		TRANSFEREV	dv-Call-Transferred-U

Service capabilites	Category	Function	Message
		ADDPTY	dv-Add-Party
		CONFPTR	dv-Conference-Party
		DROPPTY	dv-Drop-Party
		TRANPTY	dv-Transfer-Party
		MAKECALL	dv-Make-Call
		CNTRLREL	dv-Controller-Released-U
		NCTRLREL	dv-Noncontroller-Released-U
Resource Status	RESOURCE	ACDQUERY	dv-Resource-Query
		APPSTQRY	dv-Appl-Stat-Qry
Third Party Agent Control	TPAC	LOGINAGT	dv-Set-Feature
		LOGOUTACT	dv-Set-Feature
		READYAGT	dv-Set-Feature
		NREADYAGT	dv-Set-Feature
		UNRE-SERVEAGT	dv-Set-Feature
		RESERVEAGT	dv-Set-Feature
		CHGVWRAP	dv-Set-Feature
		CHGFORCE	dv-Set-Feature
Call Progress Event	CPGEVENT	DIGCOLL	dv-Call-Progress-U
		BUSY	dv-Call-Progress-U
		RINGBACK	dv-Call-Progress-U
		CONNECTED	dv-Call-Progress-U
		RAN	dv-Call-Progress-U
		MUSIC	dv-Call-Progress-U
		SILENCE	dv-Call-Progress-U

4.3 MDC and RES CompuCALL options

These options relate to lines configured with Meridian Digital Centrex business features. These options are limited to the operation of MDC in a call center.

4.3.1 MDC and RES capabilities

The following are the CompuCALL Capabilities defined for the CompuCALL MDC Option.

- Coordinated Voice and Data Delivery
- Call Control
- Three Way Call
- Call Initiation
- Message Waiting
- DN Query

Table 4 list the application service functions for MDC and RES CompuCALL options and application service capabilities.

Table 18 MDC and RES, Table SCAISSRV capabilities and functions

Service capabilities	Category	Function	Message
Centrex Event and RES Event	CTXEVENT and RESEVENT	SETOFFHK	dv-Set-Off-Hook-U
		CALLNAME	dv-Calling-Name-U
		CALLREL	dv-Call-Released-U
		CALLANSWR	dv-Call-Answered-U
		CALLOFFR	dv-Call-Offered-U
SCAI Call Control	SCAICC	ANSWCALL	dv-Answer-Call
		CALLUNHELD	dv-Call-Unheld-U
		UNHOLDCALL	dv-Unhold-Call
		HOLDCALL	dv-Hold-Call
		RELSCALL	dv-Release-Call
Call Initiative	CALLINIT	MAKECALL	dv-Make-Call
SCAI Three-way call	SCAI3WC	CONSULTEV	dv-Call-Consult-Originated
		TRANPTY	dv-Transfer-Party
		DROPPTY	dv-Drop-Party
		CONFPTY	dv-Conference-Party
		ADDPTY	dv-Add-Party
		TRANSFEREV	dv-Call-Transferred-U
		CONFEVNT	dv-Conference-Party
		CNTRLREL	dv-Controller-Released-U
		NCTRLREL	dv-Noncontroller-Released-U
SCAI Message Waiting	SCAIMWTI	MSGWAIT	dv-Message-Waiting-U
		MWTACT	dv-MWT-Activate
DN Query	DNQUERY	DNQUERY	dv-DN-Query
Call Progress Event	CPGEVENT	DIGCOLL	dv-Call-Progress-U
		BUSY	dv-Call-Progress-U
		RINGBACK	dv-Call-Progress-U
		CONNECTED	dv-Call-Progress-U

4.4 CompuCALL ACD application service functions

This section describes the CompuCALL ACD messages the switch uses to notify the host of events.

4.5 ACD Coordinated voice and data delivery

This application service capability for ACD lines includes four application Service functions:

- dv-Call-Queued-U
- dv-Call-Offered-U
- dv-Call-Answered-U
- dv-Call-Released-U

4.5.1 ACD dv-Call-Answered-U (operation value "10" hex)

This section describes the ACD dv-Call-Answered-U message the switch uses to notify the host of an incoming call offered to a CompuCALL party in an associated environment with which the host has an established session has been answered.   The CompuCALL party which answers the call is not necessarily the party to which the call has been offered (for example, in the MDC call pick-up case). The "U" means this message is unconfirmed by the host.

4.5.1.1 ACD dv-Call-Answered-U

ASN.1 Encoding

```
dv-CALL-ANSWERED-U          OPERATION
                            ARGUMENT CallAnsweredArgument
::= 16

CallAnsweredArgument        ::=     SEQUENCE
{networkCallID              [0] IMPLICIT SEQUENCE
{networkNodeID              [0] IMPLICIT INTEGER  OPTIONAL,
localCallID                 [1] IMPLICIT INTEGER OPTIONAL}

destDeviceID           [1] AddressType OPTIONAL,
acdDN                  [2] IMPLICIT OCTET STRING OPTIONAL,
origCallingNumber      [3] IMPLICIT OCTET STRING OPTIONAL,
origChargeNumber       [4] IMPLICIT OCTET STRINGOPTIONAL,
callHistoryInfo        [5] IMPLICIT SEQUENCE

 {callType             [0]  IMPLICIT  ENUMERATED
   {callTransferred    (1),
    callOverflowed     (2),
    callRedirected     (3),
    callForwarded      (4)
    callConsult          (5),
    callConference       (6),
    callReturnedToQueue (7),
    callRouted            (8)} OPTIONAL,

origInboundDN            [1] IMPLICIT DeviceAddressType
                                                    OPTIONAL,


prevApplicationID        [2] IMPLICIT INTEGER OPTIONAL,
hostCallData             [3] IMPLICIT OCTET STRING OPTIONAL}
                                             OPTIONAL,

acdGroup                 [6] AddressType   OPTIONAL,
callMode                 [7] IMPLICIT ENUMERATED
 {callWaited             (1)}    OPTIONAL,
forwardingParty          [8] ForwardingPartyType OPTIONAL,
cpadigs                  [9] IMPLICIT OCTET STRING OPTIONAL}

AddressType                     :: =     CHOICE
 {positionID             [0] IMPLICIT INTEGER,
 dialedDigits            [1] IMPLICIT IA5STRING,
 stationNumber           [2] IMPLICIT OCTET STRING,
 q931Address             [3] Q931AddressType}

Q931AddressType          :: =    SEQUENCE
 {numberTypeNumberPlan          [0] OCTET STRING OPTIONAL,
 presentationScreeningIndicator [1] OCTET STRING OPTIONAL,
 digits                         [2] IA5 STRING OPTIONAL}

ForwardingPartyType      :: =    IMPLICIT SEQUENCE
 {firstFwdNumber                [0] AddressType OPTIONAL,
 firstFwdReason                 [1] IMPLICIT ENUMERATED
 {unknown                       (0),
 userBusy                       (1),
 noReply                        (2),
 unconditional                  (3)  }   OPTIONAL,

lastFwdNumber            [2] AddressType OPTIONAL,
lastFwdReason            [3] IMPLICIT ENUMERATED
{unknown                 (0),
 userBusy                (1),
 noReply                 (2),
 unconditional           (3)  }   OPTIONAL}
```

4.5.1.2 ACD dv-Call-Answered-U: coded example 1

Switch #1 sends dv-Call-Answered-U message regarding call with LocalcallID = 2 to the host that the agent at position 9981 answered the call. The supplementary ACD DN = 7221111 and the primary DN of the ACD group = 7229999. No Return Result is expected by the switch from the host.

INVOKE DMS ===== > HOST

Table 19 dv-Call-Answered-U Invoke: coded example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	3F		INVOKE	this message is 63 bytes decimal
02	02	03 FF	InvokeID	the InvokeID is 1023
02	01	10	Operation	the operation value is dv-Call-Answered-U [16] decimal
30	36		Argument	the CallAnsweredArgument is of type sequence and is 54 bytes long decimal
A0	06		NetworkCallId	this of type constructor and is 6 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
80	01	02	LocalcallId	the LocalcallId is 2 decimal
A1	04		DeviceId	this is a constructor 4 bytes long decimal
80	02	26 FD	PositionId	the Position ID of the agent is 9981 decimal
82	0A	36 3133 37 32 32 31 31 31 31	ACDDN	the called ACDDN of the group is a supplementary DN = 6137221111
83	0C	00 83 36 31 33 37 32 32 31 31 35 32	OrigCallingNumber	the first two bytes are fixed and presently must not be read. The customer's OrigCallingNumber is 6137221152
A6	0C		ACDGRP	the ACDGroup is of type constructor and is 12 bytes long
			—continued—	

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
82	0A	36 31 33 37 32 32 39 39 39 39	StationNumber	the primary DN of the ACDGroup is 6137229999
Hex Dump = A1 3F 02 02 03 FF 02 01 10 30 36 A0 06 80 01 01 80 01 02 A1 04 80 02 26 FD 82 0A 36 31 33 37 32 32 31 31 31 31 83 0C 00 83 36 31 33 37 32 32 31 31 35 32 A6 0C 82 0A 36 31 33 37 32 32 39 39 39 39				
			—end—	

The switch sends a dv-Call-Answered-U message even if no caller information is available. The message is useful for the host to clear the workstation's screen only when the new call is answered. In the MDC case, the message is useful when a new call offered to an MDC station is put on call waiting. The dv-Call-Answered-U message for the waited call is sent only when the call is answered, and the host can display the corresponding caller's file at that time.

The contents of the dv-Call-Answered-U message are identical to those of the dv-Call-Offered-U message.

> ***Note:*** The dv-Call-Answered-U message may be preceded by a dv-Call-Offered-U message regarding the same call, i.e., with the same NetworkCallID. In the case of MDC station call pick-up, these two messages may be sent for different stations. The dv-Call-Offered-U message is sent to host for the station to which the call is offered while the dv-Call-Answered-U message is sent corresponding to the station which picks up the call.

4.5.1.3 ACD dv-Call-Answered-U: coded example 2

The OrigCallingNumber 6136217703 initiates an ACD call by dialing 6134444444. The call receives a LocalCallID of 12. An agent from the posid 7707 of the ACD group with primary DN 6136211111 answers the call. The switch sends DV-CALL-ANSWERED-U to the host. The switch does not receive a No Return Result from the host.

INVOKE DMS ===== > HOST

Table 20 dv-Call-Answered-U Invoke: coded example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	42		INVOKE	this message is 66 bytes decimal
02	02	03 FF	InvokeID	the InvokeID is 1007
02	01	10	Operation	the operation value is dv-Call-Answered-U [16] decimal
30	39		Argument	the CallAnsweredArgument is of type sequence and is 57 bytes long decimal
A0	09		NetworkCallId	this of type constructor and is 9 bytes long
80	04	03 038 D0 90	NetworkNodeId	the NetworkNodeId is 54055056 decimal
81	01	0C	LocalcallId	the LocalcallId is 12 decimal
A1	04		DeviceId	this is a constructor 4 bytes long decimal
80	02	1E 1B	PositionId	the Position ID of the agent is 7707 decimal
82	0A	36 31 33 36 32 31 32 31 31 31	ACDDN	the called ACDDN of the group is 6136212111
83	0C	00 83 36 31 33 36 32 31 37 37 30 33	OrigCallingNumber	the first two bytes are fixed and presently must not be read. The customer’s OrigCallingNumber is 6136217703
A6	0C		ACDGRP	the ACDGroup is of type constructor and is 12 bytes long
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
82	0A	36 31 33 36 32 31 32 31 31 31	StationNumber	the primary DN of the ACDGroup is 6136212111
8A	0A	36 31 33 34 34 34 34 34 34 34	cpadigs	The CPA value is 6134444444
Hex Dump = A1 42 02 02 03 EF 02 01 10 30 39 A0 09 80 04 03 38 D0 90 81 01 0C A1 04 80 02 1E 1B 82 0A 36 31 33 36 32 31 32 31 31 31 83 0C 00 83 36 31 33 36 32 31 37 37 30 33 A6 0C 82 0A 36 31 33 36 32 31 32 31 31 31 8A 0A 36 31 33 34 34 34 34 34 34 34				
			—end—	

4.5.1.4 dv-Call-Answered-U parameters

Each dv-Call-Answered-U invoke parameter description follows:

NetworkCallID contains static call-related information to uniquely identify the call in context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

DeviceID contains dynamic call-related information identifying the specific customer workstation answering the call. It is unique in the switch and is a choice parameter. The choices are PositionID and StationNumber:

PositionID identifies the ACD agent position answering the call.

ACDDN contains dynamic call-related information to indicate the primary or supplementary ACD DN associated with the call. This may not be the actual number dialed by the calling party (if an 800 service it may be the 10-digit NANP routing number translated from the dialed 800 number). This parameter applies to the ACD calls only.

OrigCallingNumber contains static call-related information to indicate the original calling number associated with a call when it enters the CompuCALL environment.

OrigChargeNumber contains static call-related information to indicate the operating company charge number (also called "ANI" - Automatic Number Identification in North America) associated with the incoming call when it enters the CompuCALL environment. A given dv-Call-

Answered-U message contains both the OrigCallingNumber and OrigChargeNumber parameters, if available.

CallHistoryInfo if the call were transferred, overflowed, or redirected from an ACD group or agent, or transferred from or forwarded by an MDC customer workstation, the dv-Call-Answered-U message includes the CallHistoryInfo parameter with one or more of the following subparameters:

Note: Each subparameter is datafilled in the switch as being provided (when appropriate) or not provided on an individual basis. In other words, the CallHistoryInfo parameter is present if the customer has subscribed to any of the following subparameters:

CallType Contains dynamic call-related information indicating if the call to the current CompuCALL party were transferred, overflowed, redirected, forwarded, or routed.

> **Note:** If a consult call, the consult party (if associated to a session) will receive the dv-Call-Answered-U with CallType transferred to indicate that the call is a consult call in the context of the call transfer feature, i.e., it could be transferred as opposed to has been transferred.

OrigInboundDN contains static call-related information identifying the first DN at which the call is directed when it enters the CompuCALL environment. It remains the same as long as the call stays in the CompuCALL environment even if it is later transferred, overflowed, redirected, or forwarded.

PrevApplicationID contains dynamic call-related information identifying the ApplicationID for the session of the associated environment to which the party extending the call belongs. (Call extension includes call transfer, overflow, redirect, and forward.)

HostCallData contains dynamic call-related information of a generic and miscellaneous nature provided by a host. If the call were redirected by a host (Section 4.6.2), this parameter may be provided by that host and include the reason for the redirection.

ACDGroup contains dynamic call-related information identifying the primary ACD DN of the ACD group in the CompuCALL environment within which the call has been answered. It lets the host correlate the dv-Call-Answered-U message with an associated DN. This parameter applies to ACD calls only. The choice is StationNumber:

StationNumber identifies the ACD group which has answered the call.

Call Mode contains dynamic call-related information identifying the modality of the call when answered by the party. Only call waited is supported. If the call were put on call waiting for the MDC station while the station is busy on an existing call, the dv-Call-Answered-U message includes the CallMode parameter to indicate that the call is waited. This parameter applies to MDC calls only.

CPADIGS contains the Called Party Address (CPA) digits as received in the setup messages for the incoming agent, displayed as a parameter in ICM messages. The first call set-up message, in the case of ISUP the Initial Address message (IAM), sent from the originating switch contains a parameter for the CPA. The terminating switch uses the CPA value to route the call. The exact address digits contained in the CPA parameter is delivered to the ICM environment. The CPADIGS field contains the dialled digits for calls to the ICM environment that arrive over a line.

In the case of overlap signalling, the originating switch sends the Subsequent Address message (SAM) following an IAM to convey additional called party address information. The CPA that arrives in the IAM and SAM, before the terminating switch sends the Address Complete Message (ACM), is delivered to the ICM environment. A CPA that arrives in the SAM after the ACM, is ignored along with the SAM. As long as the local_call_id of the call remains the same for subsequent extensions of the call, the CPA parameter in the ICM messages remains static. For different local_call_id's, the value of CPA parameter differs in an ICM environment.

ForwardingParty identifies the first forwarding number, association, last forwarding number, and reason for forwarding. In a single forwarding scenario, the information is included in the First Fwd Number and First FwdReason subparameters. This information is provided to the host only if presentation of the forwarding DNs is allowed. The parameter is a constructor including the following information:

FirstFwdNumber identifies the number from which the first forward was made in a multiple call forwarding scenario. Based on the Q.931 format, it uses the Q.931Address choice.

FirstFwdReason identifies the reason the call was first forwarded. It is an enumerated type with values unknown/not available, user busy, no reply, or unconditional.

LastFwdNumber identifies the number from which the lastforward was made in a multiple call forwarding scenario. Based on the Q.931 format, it uses the Q.931Address choice.

LastFwdReason identifies the reason the call was last forwarded (most recent). It is an enumerated type with values unknown/not available, user busy, no reply or unconditional.

No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Answered-U message. This is the reason for the "U" (unconfirmed) in the message name. If either is received by the switch, a Reject is returned with reason unrecognized invocation. Furthermore, the switch ignores any Reject message from the host in response to a dv-Call-Answered-U message.

4.5.2 ACD dv-Call-Offered-U (operation value 5)

This section describes the dv-Call-Offered-U message the switch uses to notify the host that a CompuCALL party has been selected for ACD. The party has been selected to handle an incoming call to one of the DNs belonging to the associated environment of the current switch-host session when the call is about to be offered to the CompuCALL party. The "U" indicates this message is unconfirmed, i.e., Class 5 operation which requires no outcome to be reported by the host.

4.5.2.1 ACD dv-Call-Offered-U

ASN.1 encoding

```
dv-CALL-OFFERED-U                OPERATION

                                          ARGUMENT CallOfferedArgument
    ::= 5

    CallOfferedArgument              ::=       SEQUENCE
     {networkCallID   [0] IMPLICIT SEQUENCE
     {networkNodeID            [0] IMPLICIT INTEGER OPTIONAL,
     localCallID               [1] IMPLICIT INTEGER OPTIONAL}
                                          OPTIONAL,
    acdDN              [2] IMPLICIT OCTET STRING OPTIONAL,
    origCallingNumber[3] IMPLICIT OCTET STRING OPTIONAL,
    origChargeNumber[4] IMPLICIT OCTET STRING OPTIONAL,
    callHistoryInfo     [5] IMPLICIT SEQUENCE

     {callType          [0]  IMPLICIT  ENUMERATED
         {callTransferred          (1),
          callOverflowed           (2),
          callRedirected           (3),
          callForwarded            (4)
         callConsult               (5),
         callConference            (6),
         callReturnedToQueue (7),
         callRouted                (8)} OPTIONAL,
     origInboundDN    [1] IMPLICIT DeviceAddressType OPTIONAL,
     prevApplicationID[2] IMPLICIT INTEGER OPTIONAL,
     hostCallData       [3] IMPLICIT OCTET STRING OPTIONAL}
                               OPTIONAL,

    acdGroup            [6] AddressType OPTIONAL,
    callMode            [7] IMPLICIT ENUMERATED

     {callWaited        (1)}     OPTIONAL,
    destDeviceID       [8] AddressType OPTIONAL,
    forwardingParty    [9] ForwardingPartyType OPTIONAL,
    cpadigs            [10]IMPLICIT OCTET STRING OPTIONAL}


    AddressType                      :: =      CHOICE
    {positionID         [0] IMPLICIT INTEGER,
    dialedDigits        [1] IMPLICIT IA5STRING,
    stationNumber       [2] IMPLICIT OCTET STRING,
    q931Address         [3] Q931AddressType}
```

```
Q931AddressType                     :: =    SEQUENCE
{numberTypeNumberPlan               [0] OCTET STRINGOPTIONAL,
 presentationScreeningIndicator     [1] OCTET STRINGOPTIONAL,
 digits                             [2] IA5 STRINGOPTIONAL}

ForwardingPartyType                 :: =    IMPLICIT SEQUENCE
{firstFwdNumber                     [0] AddressTypeOPTIONAL,
 firstFwdReason                     [1] IMPLICIT ENUMERATED
{unknown                            (0),
 userBusy                           (1),
 noReply                            (2),
 unconditional                      (3)} OPTIONAL,

 lastFwdNumber                      [2] AddressType OPTIONAL,
lastFwdReason                       [3] IMPLICIT ENUMERATED
{unknown                            (0),
 userBusy                           (1),
 noReply                            (2),
 unconditional                      (3)  }   OPTIONAL}
```

4.5.2.2 ACD dv-Call-Offered-U: coded example 1 (party with CompuCALL association is an ACD party)

Switch #1 receives a transferred call whose OrigInboundDN = 6137229999. The call is initiated by the OrigCallingNumber 6137221152 and is assigned a LocalCallID = 2. The call is offered to an ACD group with supplementary ACD DN = 6137221111 and primary DN of the ACD group = 6137229999. The switch sends dv-Call-Offered-U to the host when the agent at position 9981 is selected. No Return Result is required or received by the switch from the host. The transferring ACD group is not associated. As such, no PrevApplID is sent in the CallHistoryInfo of the message.

INVOKE DMS ===== > HOST

Table 21 dv-Call-Offerred-U Invoke: coded example 1

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	50		INVOKE	this message is 80 bytes decimal
02	02	03 FF	InvokeID	the InvokeID is 1023
02	01	05	Operation	the operation is dv-Call-Offered-U [5]
30	47		Argument	CallOffered argument is of sequence type and is 71 bytes long decimal
A0	06		NetworkCallId	the NetworkCallId is a constructor 6 bytes
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	01	02	LocalCallId	the LocalCallId is 2 decimal
82	0A	36 3133 37 32 32 31 31 31 31	ACDDN	the called ACDDN of the group is a supplementary DN=6137221111
83	0C	00 83 36 31 33 37 32 32 31 31 35 32	OrigCallingNumber	the first two bytes are fixed and must not be read at present. The customer's OrigCalling Number=6137221152
A5	0F		CallHistoryInfo	the CallHistoryInfo is of type constructor and is 15 bytes long
80	01	01	CallType	the call is a transferred call
81	0A	36 3133 37 32 32 39 39 39 39	OrigInboundDN	the OrigInboundDN of the call is 6137229999
A6	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long
82	0A	36 3133 37 32 32 39 39 39 39	StationNumber	the primary DN of the ACDGroup is 6137229999
A8	04		DeviceID	the DeviceID is of type constructor and is 4 bytes long
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
80	02	26 FD	Position ID	the Position ID of the selected agent is 9981 decimal
Hex Dump = A1 50 02 02 03 FF 02 01 05 30 47 A0 06 80 01 01 81 01 02 82 0A 36 31 33 37 32 32 31 31 31 31 83 0C 00 83 36 31 33 37 32 32 31 31 35 32 A5 0F 80 01 01 81 0A 36 31 33 37 32 32 39 39 39 39 A6 0C 82 0A 36 31 33 37 32 32 39 39 39 39 A8 04 80 02 26 FD				
			—end—	

The dv-Call-Offered-U message gives the host static and dynamic (call-related) information, including the station selected to handle the call so the host can display the caller's file, call handling instructions, etc. on the customer workstation data terminal at (virtually) the same time the station is connected to the caller.

A dv-Call-Offered-U message is sent by the switch even if there is no caller information available. In this case, the message is still useful since the host may used to indicate the desired service (based on the ACDDN) for the new call to the ACD agent.

4.5.2.3 ACD dv-Call-Offered-U: coded example 2

The OrigCallingNumber 6136217703 initiates an ACD call by dialing 6134444444. The call is assigned a LocalCallID = 9. The call is offered to an ACD agent with posid 7707 of the ACD group with primary DN 6136212111. The switch sends DV-CALL-OFFERED-U to the host. The switch does not receive a No Return Result from the host.

INVOKE DMS =====> HOST

Table 22 dv-Call-Offerred-U Invoke: coded example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	3E		INVOKE	this message is 66 bytes decimal
02	02	03 F9	InvokeID	the InvokeID is 1017
02	01	05	Operation	the operation is dv-Call-Offered-U [5]
30	39		Argument	CallOffered argument is of sequence type and is 57 bytes long decimal
A0	09		NetworkCallId	the NetworkCallId is a constructor 9 bytes
80	04	03 38 d0 90	NetworkNodeId	the NetworkNodeId is 54055056
81	01	09	LocalCallId	the LocalCallId is 9 decimal
82	0A	36 31 33 36 32 31 32 31 31 31	ACDDN	the called ACDDN of the group is 6136212111
83	0C	00 83 36 31 33 36 32 31 37 37 30 33	OrigCallingNumber	the first two bytes are fixed and must not be read at present. The customer's Orig-Calling Number=6136217703
A6	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long
82	0A	36 3133 36 32 31 32 31 31 31	StationNumber	the primary DN of the ACDGroup is 6136212111
A8	04		DeviceID	the DeviceID is of type constructor and is 4 bytes long
80	02	1E 1B	Position ID	the Position ID of the selected agent is 7707 decimal
8A	0A	36 31 33 34 34 34 34 34 34 34	cpadigs	The CPA value is 6134444444
Hex Dump = A1 3E 02 02 03 F9 02 01 05 30 39 A0 09 80 04 03 38 D0 90 81 01 09 82 0A 36 31 33 36 32 31 32 31 31 31 83 0C 00 83 36 31 33 36 32 31 37 37 30 33 A6 0C 82 0A 36 31 33 36 32 31 32 31 31 31 A8 04 80 02 1E 1B 8A 0A 36 31 33 34 34 34 34 34 34 34				

4.5.2.4 dv-Call-Offered-U parameters

A brief description of the parameters of the dv-Call-Offered-U invoke follows.

NetworkCallID contains static call-related information to uniquely identify the call in context of the customer's network for the maximum expected call duration. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

ACDDN contains dynamic call-related information to indicate the primary or supplementary ACD DN associated with the call. This may not be the actual number dialed by the calling party (e.g., in the case of 800 service it may be the 10-digit NANP routing number translated from the dialed 800 number). This applies to the ACD calls only.

OrigCallingNumber contains static call-related information to indicate the original Calling Number associated with a call when it enters the CompuCALL environment. For example, this may be a 10-digit North American Numbering Plan (NANP) DN associated with the calling line (or in the case of a call from a PBX, it may be the listed DN (LDN) of the PBX), or a private number.

OrigChargeNumber contains static call-related information to indicate the operating company charge number associated with the incoming call when it enters the CompuCALL environment. A dv-Call-Offered-U message contains both the OrigCallingNumber and OrigChargeNumber parameters, if available.

CallHistoryInfo if the call is transferred, overflowed or redirected from an ACD group or agent, or transferred from or forwarded by an MDC customer workstation, the dv-Call-Offered-U message includes the CallHistoryInfo parameter containing one or more of the following subparameters:

CallType contains dynamic call-related information to indicate if the call to the current CompuCALL party has been transferred, overflowed, redirected, forwarded, or routed.

OrigInboundDN contains static call-related information to identify the first DN at which the call is directed when it enters the CompuCALL environment. It remains the same as long as the call stays within the CompuCALL environment even if it is subsequently transferred, overflowed, redirected or forwarded.

> **Note:** Call extension includes call transfer, overflow, redirect, and forward.

HostCallData contains dynamic call-related information of a generic and miscellaneous nature provided by a host. If a call has been redirected by a host, this parameter may be provided by that host at that time, and might include the reason for the redirection.

ACDGroup contains dynamic call-related information identifying the primary ACD DN of the ACD group in the CompuCALL environment to which the call has been routed. It lets the host correlate the dv-Call-Offered-U message with an associated DN. This parameter applies to ACD calls only. Stationnumber is the only choice.

StationNumber identifies the ACD group to which the call is about tobe offered.

CallMode contains dynamic call-related information identifying the call's modality when it's offered to the party. Only call waited is supported. If the call was put on call waiting for the MDC station while the station is busy on an existing call, the dv-Call-Offered-U message includes the CallMode parameter to indicate the call is waited. This parameter applies only to MDC calls.

DeviceID contains dynamic call-related information to identify the customer workstation to which the call is offered. It is unique within the switch and defined as a choice parameter.

CPADIGS contains the Called Party Address (CPA) digits as received in the setup messages for the incoming agent, displayed as a parameter in ICM messages. The first call set-up message, in the case of ISUP the Initial Address message (IAM), sent from the originating switch contains a parameter for the CPA. The terminating switch uses the CPA value to route the call. The exact address digits contained in the CPA parameter is delivered to the ICM environment. The CPADIGS field contains the dialled digits for calls to the ICM environment that arrive over a line.

In the case of overlap signalling, the originating switch sends the Subsequent Address message (SAM) following an IAM to convey additional called party address information. The CPA that arrives in the IAM and SAM, before the terminating switch sends the Address Complete Message (ACM), is delivered to the ICM environment. A CPA that arrives in the SAM after the ACM, is ignored along with the SAM. As long as the local_call_id of the call remains the same for subsequent extensions of the call, the CPA parameter in the ICM messages remains static. For different local_call_id's, the value of CPA parameter differs in an ICM environment.

ForwardingParty identifies the first forwarding number and associated reason and the last forwarding number and reason for forwarding. In a single forwarding scenario, the information is included in the

FirstFwdNumber and FirstFwdReason subparameters. This information is provided to the host only if presentation of the forwarding DNs includes the following information:

FirstFwdNumber identifies the number from which the first forward was made in a multiple call forwarding scenario. It is based on the Q931 format and uses the Q931Address choice.

FirstFwdReason identifies the reason the call was forwarded the first time. It is an enumerated type with values: unknown/not available, user busy, no reply, or unconditional.

LastFwdNumber identifies the number from which the last forward was made in a multiple call forwarding scenario. Based on the Q931 format, it uses the Q931Address choice.

LastFwdReason identifies the reason the call was forwarded the last time (most recent). It is an enumerated type with values: un known/not available, user busy, no reply, or unconditional.

deviceAddressType sends a set of one or more values representing the calling device. The set includes the following subparameters:

dialedDigits contains the DN in the dial number delivery format.

stationNumber contains the directory number format.

name contains the caller’s name.

The switch sends the dv-Call-Offered-U message to the host when a customer workstation is offered an incoming call in one of the following scenarios:

1 If ACD, the first available agent has been selected to handle the call (more calls than available agents) and the call has exited the incoming call queue; or the call is received by the switch and is directly offered to the ACD agent idle the longest (i.e., more available agents than calls); or

2 If MDC, the call is received by the switch and is offered to the specific MDC station to which the call is directed.

No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Offered-U message. This is the reason for the "U" (unconfirmed) in the message name. If either is received by the switch, a reject is returned with reason “unrecognized invocation”. Furthermore, the switch ignores any reject message from the host in response to a dv-Call-Offered-U message.

4.5.3 ACD dv-Call-Queued-U (operation value 6)

This section describes the dv-Call-Queued-U message the switch uses to notify the host of a call in the incoming call queue of an ACD group of an associated environment with which the host has an established session. This message applies to the Meridian ACD CompuCALL options only.

4.5.3.1 ACD dv-Call-Queued-U

The switch sends this RO of Operation Class 5 to the host when a call to an ACD DN associated with the current session is received by the switch and placed in the incoming call queue. See Meridian CompuCALL application service functions.

ASN.1 encoding

```
dv-Call-Queued-U ARGUMENT            OPERATION

                                   ARGUMENT CallQueuedArgument
::= 6

    CallQueuedArgument                  ::=     SEQUENCE
     {networkCallID              [0] IMPLICIT SEQUENCE
     {networkNodeID              [0] IMPLICIT INTEGER OPTIONAL,
     localCallID                 [1] IMPLICIT INTEGER OPTIONAL}
                                 OPTIONAL,

    acdDN                    [1] IMPLICIT OCTET STRING OPTIONAL,
    origCallingNumber        [2] IMPLICIT OCTET STRING OPTIONAL,
    origChargeNumber         [3] IMPLICIT OCTET STRING OPTIONAL,
    callHistoryInfo          [4] IMPLICIT SEQUENCE

     {callType               [0] IMPLICIT  ENUMERATED
        {callTransferred     (1),
         callOverflowed      (2),
         callRedirected      (3),
         callForwarded       (4)
        callConsult          (5),
        callConference       (6),
        callReturnedToQueue (7),
        callRouted            (8)} OPTIONAL,

    origInboundDN            [1] IMPLICIT OCTET STRING OPTIONAL,
    prevApplicationID        [2] IMPLICIT INTEGER OPTIONAL,
    hostCallData             [3] IMPLICIT OCTET STRING OPTIONAL}
                             OPTIONAL,

    acdGroup                 [5] AddressType OPTIONAL,
    forwardingParty          [6] ForwardingPartyType OPTIONAL,
cpadigs                       [7] IMPLICIT OCTET STRING
                              OPTIONAL}


AddressType:: = CHOICE
 {positionID                  [0] IMPLICIT INTEGER,
 dialedDigits                 [1] IMPLICIT IA5STRING,
 stationNumber                [2] IMPLICIT OCTET STRING,
 q931Address                  [3] Q931AddressType}

Q931 AddressType                     :: =      SEQUENCE
 {numberTypeNumberPlan               [0] OCTET STRING OPTIONAL,
 presentationScreeningIndicator      [1] OCTET STRING OPTIONAL,
 digits                              [2] IA5 STRING OPTIONAL}


ForwardingPartyType                  :: =      IMPLICIT SEQUENCE
 {firstFwdNumber                     [0] AddressType OPTIONAL,
 firstFwdReason                      [1] IMPLICIT ENUMERATED
 {unknown                            (0),
 userBusy                            (1),
 noReply                             (2),
 unconditional                       (3)  }    OPTIONAL,
 lastFwdNumber                       [2] AddressType OPTIONAL,
 lastFwdReason                       [3] IMPLICIT ENUMERATED
 {unknown                            (0),
 userBusy                            (1),
 noReply                             (2),
 unconditional                       (3)  }    OPTIONAL},
```

4.5.3.2 ACD dv-Call-Queued-U: coded example 1

Switch #1 receives an ACD call initiated by the OrigCallingNumber 6137221152 and assigned a LocalCallID = 2, and offers it to an ACD group with supplementary ACD DN = 6137221111 and the primary DN of the ACD group = 6137229999. Since no agent is available, the call is placed in the incoming call queue and the switch sends dv-Call-Queued-U to the host. No Return Result is required or received by the switch from the host.

The call has also been forwarded several times before being forwarded to this ACD group. The first forwarding party is 5145555678. The last forwarding party, also the DN who forwarded the call to this ACD group, is 5145551121.

INVOKE DMS =====> HOST

Table 23 dv-Call-Queued-U Invoke: coded example 1

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	39		INVOKE	Message length is 77 bytes decimal
02	02	03 FF	InvokeID	the InvokeID is 1023
02	01	06	Operation	Operation value for dv-Call-Queued-U is decimal 6
30	30		Argument	CallQueued argument is of sequence type and is 68 bytes king decimal
A0	06		NetworkCallId	the NetworkCallId is a constructor 6 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	01	02	LocalCallId	the LocalCallId is 2 decimal
81	0A	36 3133 37 32 32 31 31 31 31	ACDDN	the called ACDDN of the Group is a Supplementary DN = 6137221111
82	0C	00 83 36 3133 37 32 32 31 31 35 32	OrigCallingNumber	the first 2 bytes are fixed and must not be read at present. The customer's OrigCallingNumber is 6137221152
A4	12		CallHistoryInfo	variable length
80	01	07	CallType	Call returned to Queue
81	0A	35 3134 35 35 35 35 36 37 38	OrigInboundDN	the OrigInboundDN of the call is 5145555678
82	01	06	PrevApplicationID	coded in minimum number of bytes
A5	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long
82	0A	36 3133 37 32 32 39 39 39 39	StationNumber	the StationNumber of the ACDGroup is 6137229999

Hex Dump = A1 39 02 02 03 FF 02 01 06 30 30 A0 06 80 01 01 81 01 02 81 0A 36 31 33 37 32 32 31 31 31 31 82 0C 00 83 36 31 33 37 32 32 31 31 35 32 A4 12 80 01 07 81 0A 35 31 34 35 35 35 35 36 37 38 82 01 06 A5 0C 82 0A 36 31 33 37 32 32 39 39 39 39

The switch sends the dv-Call-Queued-U message to the host when a call with an ACD DN is received by the switch and is placed in the incoming call queue of the ACD group to which that ACD DN is assigned within the Associated environment of the current CompuCALL session. The dv-Call-Queued-U message informs the host when a call has been queued so the host has more time to generate the screen display (while the call is queued). Once an agent is selected to handle the call, a dv-Call-Offered-U message is sent to the host and call-related information can be displayed for the agent.

> ***Note 1:*** If there is no incoming call queue and the call is immediately offered to an agent, a dv-Call-Queued-U message is not sent to the host.
>
> ***Note 2:*** If the service profile for that session contains the dv-Call-Queued-U and dv-Call-Received-C messages, and the Call redirection capability is active for the ACD group, then dv-Call-Received-C instead of dv-Call-Queued-U will be sent by DMS-100 for that ACD Group.

The dv-Call-Queued-U message contains the same parameters as the dv-Call-Offered-U message except the DeviceID parameter (no agent has yet been selected to handle the call) and CallMode parameter (unsupported in the Meridian ACD CompuCALL options). The dv-Call-Queued-U message is not sent for MDC lines.

4.5.3.3 ACD dv-Call-Queued-U: coded example 2

The OrigCallingNumber 6136217703 initiates an ACD call by dialing 6134444444. The call is assigned LocalCallID = 13 and offered it to an ACD group with Primary DN 6136212111. The ACD group has no available agents and the call is placed in the Incoming Call Queue. The switch sends a dv-Call-Queued-U message to the host. The switch does not require a Return Result message from the host.

INVOKE DMS ===== > HOST

Table 24 dv-Call-Queued-U Invoke: coded example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	3C		INVOKE	Message length is 60 bytes decimal
02	02	03 EA	InvokeID	the InvokeID is 1002
02	01	06	Operation	Operation value for dv-Call-Queued-U is decimal 6
30	33		Argument	CallQueued argument is of sequence type and is 51 bytes king decimal
A0	09		NetworkCallId	the NetworkCallId is a constructor 9 bytes long
80	04	03 38 D0 90	NetworkNodeId	the NetworkNodeId is 54055056
81	01	0D	LocalCallId	the LocalCallId is 13 decimal
81	0A	36 3133 36 32 31 32 31 31 31	ACDDN	the called ACDDN of the Group is 6136212111
82	0C	00 83 36 3133 36 32 31 37 37 30 33	OrigCallingNumber	the first 2 bytes are fixed and must not be read at present. The customer’s OrigCallingNumber is 6136217703
A5	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long
82	0A	36 3133 36 32 31 32 31 31 31	StationNumber	the StationNumber of the ACDGroup is 6136212111
87	0A	36 31 33 34 34 34 34 34 34 34	cpadigs	The CPA value is 6134444444
Hex Dump = A1 3C 02 02 03 EA 02 01 06 30 33 A0 09 80 04 03 38 D0 90 81 01 0D 81 0A 36 31 33 36 32 31 32 31 31 31 82 0C 00 83 36 3133 36 32 3137 37 30 33 A5 0C 82 0A 36 31 33 36 32 31 32 31 31 31 87 0A 36 31 33 34 34 34 34 34 34 34				

4.5.3.4 dv-Call-Queued-U parameters

A description of the parameters of the dv-Call-Queued-U invoke follows.

NetworkCallID contains static call-related information to uniquely identify the call within the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

ACDDN contains dynamic call-related information to indicate the primary or supplementary ACD DN associated with the call. This may not be the actual number dialed by the calling party. If an 800 service it may be the 10-digit NANP routing number translated from the dialed 800 number.

OrigCallingNumber contains static call-related information to indicate the original calling number associated with a call when it enters the CompuCALL environment.

OrigChargeNumber contains static call-related information. Indicates the operating company charge number associated with the incoming call when it enters the CompuCALL environment. A given dv-Call-Queued-U message contains both the OrigCallingNumber and OrigChargeNumber, if both are available.

CallHistoryInfo if the call is transferred, overflowed, or redirected from an ACD group or agent, or transferred from or forwarded by an MDC customer workstation, the dv-Call-Queued-U message includes the CallHistoryInfo parameter containing one or more of the following subparameters:

Note: Each subparameter is datafilled in the switch as being provided (when appropriate) or not provided on an individual basis. In other words, the CallHistoryInfo parameter is present if the customer has subscribed to any of the following subparameters:

CallType contains dynamic call-related information to indicate if the call to the current CompuCALL party was transferred, overflowed, redirected, forwarded, or routed.

Note: If a consult call, the consult party (if associated to a session) receives the dv-Call-Queued-U with CallType "transferred" indicating the call is a consult call in the context of the call transfer or conference feature, i.e., it could be transferred or conference as opposed to has been transferred or conference.

OrigInboundDN contains static call-related information to identify the first DN at which the call is directed when it enters the CompuCALL environment. It remains the same as long as the call stays in CompuCALL even if it is later transferred, overflowed, redirected, or forwarded.

PrevApplicationID contains dynamic call-related information to identify the ApplicationID for the session of the associated environment to which the party extending the call belongs. (Call extension includes call transfer, overflow, redirect, and forward.)

HostCallData contains dynamic call-related information of a generic, miscellaneous nature provided by a host. If a call was redirected by a host (Section 2.1.2.2), this parameter may be provided by that host at that time, and might include the reason for the redirection.

ReturnedtoQueue indicates that the call is returned to the same ACD queue from which it was routed. However, if the call is routed to a CDN and that call is returned to queue during an attempt to route, then the call returns to the CDN queue. When the call returns to queue it retains the original arrival time.

Note: The functionality only applies when a call is routed to an agent directly using dv-route-call.

ACDGroupContains dynamic call-related information to identify the primary ACD DN of the ACD group in the CompuCALL environment where the call was queued. It lets the host correlate the dv-Call-Queued-U message with an associated DN. Station number is the available choice.

StationNumber identifies the ACD group for which the call is queued.

CPADIGS contains the Called Party Address (CPA) digits as received in the setup messages for the incoming agent, displayed as a parameter in ICM messages. The first call set-up message, in the case of ISUP the Initial Address message (IAM), sent from the originating switch contains a parameter for the CPA. The terminating switch uses the CPA value to route the call. The exact address digits contained in the CPA parameter is delivered to the ICM environment. The CPADIGS field contains the dialled digits for calls to the ICM environment that arrive over a line.

In the case of overlap signalling, the originating switch sends the Subsequent Address message (SAM) following an IAM to convey additional called party address information. The CPA that arrives in the IAM and SAM, before the terminating switch sends the Address Complete Message (ACM), is delivered to the ICM environment. A CPA that arrives in the SAM after the ACM, is ignored along with the SAM. As long as the local_call_id of the call remains the same for subsequent extensions of the call, the CPA parameter in the ICM messages remains static. For different local_call_id's, the value of CPA parameter differs in an ICM environment.

ForwardingParty contains information on the first and the last forwarding party to identify the first forwarding number and associated reason and last forwarding number and reason for forwarding. In a s single forward ing scenario, the information is included in the FirstFwdNumber and First FwdReason subparameters. This information is provided to the host only if presentation of the forwarding DNs are allowed. The parameter is defined as a constructor including the followinginformation:

FirstFwdNumber identifies the number from which the first forward was made in a multiple call forwarding scenario. Based on the Q931 format, it uses the Q931Address choice.

FirstFwdReason identifies the reason the call was first forwarded. It is an enumerated type with values: unknown/not available, user busy, no reply, or unconditional.

LastFwdNumber identifies the number from which the lastforward was made in a multiple call forwarding scenario. Based on the Q931 format, it uses the Q931Address choice.

LastFwdReason identifies the reason the call was last forwarded (most recent). It is an enumerated type with values: unknown/not available, user busy, no reply, or unconditional.

No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Queued-U message. This is the reason for the "U" (unconfirmed) in the message name. If either is received by the switch, a Reject is returned with reason "unrecognized invocation". Furthermore, the switch ignores any Reject message from the host in response to a dv-Call-Queued-U message.

4.5.4 ACD dv-LOB-Event-U (operation value 2A)

This section describes the dv-LOB-Event-U message. The DMS-100 switch sends the message to the host after an ACD agent enters the line of business (LOB) code during an active call. The host may then extract the code and the associated time-stamp from the message to generate various management reports. There is no restriction on the number of event messages that can generate during one call unlike the MIS event message.

The agent can erase the last code by entering three stars (“***”). As a result, the DMS-100 switch does not send another dv-LOB-Event-U message until the agent enters a new code or the agent releases the call.

The table contains LOB event message generation scenarios. There is a description of the different actions an agent can take and how it affects the event message generation.

Table 25 LOB event message generation scenarios

Agent Actions/Events	DMS-100 switch actions
Enters the first LOB code for the call	buffers the code
Enters another LOB code during the same call	sends the first LOB code to the host and buffers the new one
Agent releases the call	Sends the previously buffered code in a dv-LOB-Event-U message. If there is no buffered code, the switch does not send any code even if the switch datafills a default code for the ACD group.
Cancels the last LOB code entered	Erases the code lying in the buffer. The switch only erases the last code and not any entered before.
Does not enter any code during the entire call	Does not send dv-LOB-Event-U message. This action is different from the ACD MIS that reports the default code.
Enters more that three LOB codes	Sends all codes entered to the host. This action is different from the ACD MIS that reports only the first three codes.
Enters an LOB code after the caller has hung up	Waits for 2.5 seconds before tearing the call. If agent presses the LOB key during this interval, stretches out the wrap up time and waits for the agent to finish entering the code. Then sends the code in a dv-LOB-Event-U message.
Enters an invalid LOB code	Validates all codes. Accepts only codes from 000 through 999 and the special code “***”. Sends the dv-LOB-Event-U message for the valid codes.
Presses the LOB key twice in quick succession	Restarts LOB feature on second key press. This action is different from Nortel’s PBX.

4.5.4.1 dv-LOB-Event-U parameters

All of the following parameters are optional except LOB code.

NetworkCall ID contains a CompuCALL generated call ID composed of a Network Node ID and a Local Call ID. Use the call ID already assigned to the call unless the call ID is new.

Agent Pos ID specifies the position ID of the agent.

LOB code specifies the three digit LOB code that the agent enters.

LOB time specifies the time associated with the LOB code parameter, indicating the local switch time when the agent enters the LOB.

4.5.4.2 dv-LOB-Event-U validations

The switch performs a series of checks before sending the dv-LOB-Event-U message. The switch verifies:

1 The switch-host session subscribes to the call event message.

2 The associated device address of the ACD Group DN or the individual agent's DN subscribes to the call event message.

3 The SCAI service version is at least SCAI11.

4.5.4.3 dv-LOB-Event-U: ASN.1 definitions

```
dv-LOB-Event-U                    OPERATION
                                  ARGUMENT LOBEventArgument
::= 42

LOBEventArgument:           ::= SEQUENCE
{networkCallID      [0]     IMPLICIT NetworkCallID
{agentPosID         [1]     IMPLICIT INTEGER          OPTIONAL,
{LOBCode            [2]     IMPLICIT OCTET STRING OPTIONAL,
{LOBTime            [3]     IMPLICIT SwitchTimeOfDayOPTIONAL}

SwitchTimeOfDay             ::= IMPLICIT SEQUENCE
    {hours      [0]     IMPLICIT INTEGER                (0..23),
     minutes    [1]     IMPLICIT INTEGER                (0..59),
     seconds    [2]     IMPLICIT INTEGER                (0..59)}
```

4.5.4.4 dv-LOB-Event-U: coded example

The following table shows a coded example of the dv-LOB-Event-U message.

INVOKE DMS ===== > HOST

Table 26 dv-LOB-Event-U Invoke

Hex Dump			Information	Comments
Tag	**Length**	**Content**		
A1	25		INVOKE	the message length is 37 bytes decimal
02	02	01 EA	InvokeID	the invoke ID is 1006
02	01	2A	Operation	the operation value is dv-LOB-Event-U [42] decimal
30	1C		Argument	Argument is type sequence and is 28 bytes long
A0	06		networkCAllID	networkCallID is type constructor and is 6 bytes long
80	01	00	networkNodeID	the networkNodeID is 0
81	01	3C	localCallID	the localCallID is 60 decimal
81	02	26 FD	AgentPosID	the position ID of the agent is 9981 decimal
82	03	33 35 31	LOBCode	LOBCode is an octet string, 3 bytes in length and contains 0x03, 0x05, and 0x01 representing the LOBCode value of 351
A3	09		LOBTime	LOBTime is a constructor and is 9 bytes in length
80	01	0A	hours	hours is a primitive type of length 1 and its value is 10
81	01	17	minutes	minutes is a primitive type of length 1 and it value is 23
82	01	2E	seconds	seconds is a primitive type of length 1 and it value is 46
Hex Dump=A1 25 02 02 01 EA 02 01 2A 30 1C A0 06 80 01 00 81 01 3C 81 02 26 FD 82 03 33 35 31 A3 09 80 01 0A 81 01 17 82 01 2E				

4.5.5 ACD dv-Call-Released-U (operation value 7)

This section describes the dv-Call-Released-U message the switch uses to notify the host why the call was released from a CompuCALL party belonging to an Associated environment with which the host has an established session. The "U" means this message is unconfirmed by the host. The switch sends this RO is of operation class 5 to the host indicating a call has been released.

ASN.1 Encoding

```
dv-Call-Released-U OPERATION
                            ARGUMENT                ReleaseArgument

::= 7

ReleaseArgument                         ::=     SEQUENCE
{networkCallID          [0] IMPLICIT SEQUENCE
{networkNodeID          [0] IMPLICIT INTEGER OPTIONAL,
localCallID             [1] IMPLICIT INTEGER OPTIONAL}
                                                OPTIONAL,
releaseReason           [1] IMPLICIT ENUMERATED
{callCleared            (0),
callTransferred         (1),
callOverflowed          (2),
callAbandoned           (3),
callRedirected          (4),
callForwarded           (5),
callPickedUp            (6),
partyDropped            (7),
partyDroppedNoAnswer    (8)}    OPTIONAL,

acdDN                   [2] IMPLICIT OCTET STRINGOPTIONAL,
acdGroup                [4] AddressTypeOPTIONAL,
deviceID                [5] AddressTypeOPTIONAL}

postcallStatus          [6] IMPLICIT ENUMERATED
{ null                          (0)
    varWrapStarted              (1)
    zeroInterval                (2)
    releaseGuardStarted         (3)
    notReady                    (4)
    loggedOut                   (5)
                                OPTIONAL}
AddressType             :: =    CHOICE
{positionID             [0] IMPLICIT INTEGER,
dialedDigits            [1] IMPLICIT IA5STRING,
stationNumber           [2] IMPLICIT OCTET STRING,
q931Address             [3] Q931AddressType}
```

```
Q931AddressType              :: =   SEQUENCE
{numberTypeNumberPlan               [0] OCTET STRINGOPTIONAL,
presentationScreeningIndicator      [1] OCTET STRINGOPTIONAL,
digits                              [2] IA5 STRINGOPTIONAL}
```

4.5.5.1 ACD dv-Call-Released-U: coded example

Switch #1 sends dv-Call-Released-U message regarding call with LocalCallID = 2 to the host that the agent at position 9981 released the call. The supplementary ACD DN = 7221111 and the primary DN of the ACD group = 7229999. No Return Result is expected by the switch from the host.

INVOKE DMS ===== > HOST

Table 27 dv-Call-Released-U Invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	37		INVOKE	this message is 55 bytes decimal
02	02	03 FA	InvokeID	the InvokeID is 1018
02	01	07	Operation	the operation is dv-Call-Released-U [7]
30	2E		Argument	argument is of sequence type 46 bytes decimal
A0	06		NetworkCallId	the NetworkCallId is a constructor 6 bytes
80	01	00	NetworkNodeId	the NetworkNodeId is 1
81	01	02	LocalCallID	the LocalCallID is 2 decimal
81	01	00	ReleaseReason	the ReleaseReason is CallCleared (0)
82	0A	36 31 33 36 32 31 38 35 30 30	ACDDN	the called ACDDN of the Group is a Supplementary DN = 6136218500
A4	0C		ACDGroup	the ACDGroup is of type constructor 12 bytes
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
82	0A	36 31 33 36 32 31 38 35 30 30	StationNumber	the Primary DN of the ACDGroup is 61376218500
A5	04		DeviceID	the DeviceID is of type constructor 4 bytes
80	02	05 DF	PositionID	the PositionID of the agent is 1503 decimal
86	01	01	postcallStatus	the postcallStatus is 1 byte, variable wrap started
Hex Dump = A1 37 02 02 03 FA 02 01 07 30 2E A0 06 80 01 00 81 01 02 81 01 00 82 0A 36 31 33 36 32 31 38 35 30 30 A4 0C 82 0A 36 31 33 36 32 31 38 35 30 30 A5 04 80 02 05 DF 86 01 01				
			—end—	

This message which indicates why the call was released provides "closure" regarding previous CompuCALL messages for that party on the same call. It also allows the host to close a "call record" associated with the call.

4.5.5.2 dv-Call-Released-U parameters

A brief description of the dv-Call-Released-U parameters follows:

NetworkCallID contains static call-related information to identify the call in context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

ReleaseReason indicates why a call was released from a CompuCALL party belonging to an associated environment:

a. Call Cleared (one party in a two-party CompuCALL call has disconnected after being answered)

b. Caller Transferred (the CompuCALL party has transferred the call and disconnected)

c. Call Overflowed (call was overflowed from the CompuCALL party by an ACD "overflow" feature),

d. Call Abandoned (caller in a two-party call has disconnected before a call was answered by the CompuCALL party - while in the Incoming Call Queue)

e. Call Redirected (the call was redirected from the CompuCALL party by the host using call redirection)

f. Call Forwarded (call was forwarded to another station by the MDC Call Forward Don't Answer feature)

g. Call Picked-up (call was picked up by another station after call was offered to the MDC CompuCALL party)

h. PartyDropped (CompuCALL party in an established 3-way call is dropped or released).

i. PartyDroppedNoAnswer (CompuCALL party is a consult party in a 3-way call and is dropped or released before answering).

Note: For consistency, if call transfer or 3-way call conference, the party who initiates the transfer or conference is the controller. The original call party is the party who calls the controller or the party the controller first calls to establish the original 2-way call. The consult party is the party who the controller calls after a 2-way call between the controller and original call party already exists.

ACDDN contains dynamic call-related information to indicate the primary or supplementary ACD DN associated with the call. This may not be the actual number dialed by the calling party (if 800 service it may be the 10-digit NANP routing number translated from the dialed 800 number). This parameter applies to ACD calls only.

ACDGroup contains dynamic call-related information to identify the primary ACD DN of the ACD group in the CompuCALL environment from which the call was released. It provides the information for the host to correlate the dv-Call-Released-U message with an associated DN. The choice is StationNumber:

StationNumber identifies the ACD group to which the ACD agent who released the call belongs.

DeviceID contains dynamic call-related information to identify the specific customer workstation from which the call is released. It is unique within the switch and a choice parameter. The choices are PositionID and StationNumber:

PositionID identifies the ACD agent position from which the call is released.

PostcallStatus indicates the status of the agent once the agent releases the call.

No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Released-U message. If either is received by the switch, a Reject is returned with reason "unrecognized invocation." Furthermore, the switch ignores any Reject message from the host in response to a dv-Call-Released-U message.

4.5.6 dv-EMK-U (operation value 2B hex)

This section describes the dv-EMK-U message the DMS-100 switch sends to the host application when:

- an ACD agent activates the Emergency Key feature
- an ACD agent terminates

  Note: The switch sends the dv-EMK-U message to terminate only when there are two parties on the call.
- an agent, caller, supervisor, or auxillary device exits

The ACD Emergency Key feature (EMK) provides an ACD customer with the ability to monitor and record calls which turn out to be threatening or abusive. The intent of the ACD Emergency Key feature is to conference in a supervisor and/or auxiliary device when an ACD agent presses the Emergency Key. There are two parts involved:

1 An ACD agent activities the EMK, a supervisor phone set and/or auxiliary device is ringing.

Note: An ACD agent can only activate the EMK feature while the agent is in the talking state of a normal 2-party ACD call.

2 An ACD agent establishes an EMK conference when the supervisor and/or auxiliary device answers.

If the supervisor is not available on an EMK or AnswerEMK (AEMK) call, an ACD agent presents the EMK request to an available AEMK within the supervisor's hunt group. If an ACD agent can not find an available supervisor or auxiliary device, the EMK feature terminates and the EMK activation is unsuccessful.

If the auxiliary device associated with the ACD agent's EMK is available, the auxiliary device rings and upon answer conferences into the ACD call.

If an agent associates with the AEMK key and auxiliary device (AEMK and auxiliary devices are associated through the ACD group and subgroup that the agent belongs), both will ring and conference into the call when answered. Each call rings for a maximum of 30 seconds.

4.5.6.1 EMK activation

The following table summarizes the various combinations of supervisor and auxiliary device states, EMK activation state and whether the switch sends an EMK event message.

Table 28 dv-Call-Released-U Invoke

Supervisor	Auxiliary	EMK activation	EMK event message
ringing	ringing	successful	yes
ringing	not ringing	successful	yes
not ringing	ringing	successful	yes
not ringing	not ringing	unsuccessful	no
ringing	does not exist	successful	yes
does not exist	ringing	successful	yes
does not exist	does not exist	unsuccessful	no

If the supervisor has not answered within the 30 second time-out and it belongs to a hunt group, the EMK feature hunts for another supervisor AEMK to present the call. If one is found, the host computer receives an EMK activation message.

4.5.6.2 EMK termination

An ACD agent can terminate the EMK by one of the following conditions:

> ***Note:*** For each of these conditions, the host computer receives an EMK message that the agent terminated the EMK.

- 30 second time out
  - — If the ACD agent activates the EMK, then the supervisor and/or auxiliary device rings for 30 seconds. If the AEMK or auxiliary device does not answer within the 30 seconds, the ACD agent abandons the call. The switch sends an EMK event message when each time out occurs. When there are only two parties left (the last one times out) the switch sends an EMK event message stating that the EMK event has terminated.

  Note: The 30 second time out applies to the original supervisor/auxiliary device pair. Hunt group supervisor/auxiliary devices have a 15 second time-out.

- ACD agent cancels EMK

  - — The ACD agent can attempt to cancel the EMK by re-pressing the EMK. This cancellation is successful, if neither the AEMK nor the auxiliary device is active on the call or the ACD agent answers AEMK or auxiliary device. Otherwise the ACD agent ignores the attempt to cancel. If the cancel is successful, the switch sends an EMK event message stating that the ACD agent cancelled EMK to the host computer.

- ACD agent activates Not Ready Key
  - — The ACD agent activates the ACD NotReady key and the DMS-100 switch does not datafill the ACD Non-Immediate Cutoff feature. This releases the call immediately and the EMK lamp turns off. The entire call is down. The switch sends an EMK event message stating that the ACD agent terminated the EMK event to the host computer.
  - — If the ACD agent activates the ACD NotReady key, but the ACD Non-Immediate Cutoff feature is active, the agent releases the call. As far as functionality of the EMK event message, this is the same as if the ACD agent releases the call. The switch sends the EMK event message stating that the ACD agent terminated the EMK event to the host computer.
  - — The EMK event message states the agent exited the call. Because the agent terminates the EMK feature when the agent releases the call, the EMK event message that states the agent exited the call also implies that the agent deactivated the EMK call.

4.5.6.3 Exits from the EMK call

Any of the parties involved on the EMK call can exit the call at any time. The results of this exit vary depending on which party exits and how many parties remain on the call.

The ACD EMK feature is only active while more than two parties are on the call. When the EMK call has only two parties, the ACD agent never activates the emergency key feature.

> ***Note:*** A ringing emergency line is part of the call. For example, a ringing line is a party on the call.

- Activating ACD agent releases call
  - — The activating ACD agent must remain on the call for its duration. So if the agent exits by simply releasing the call, all remaining parties exit. The host computer receives an EMK event message stating that the agent exits the EMK call. There is not a separate EMK deactivate message, because the EMK feature deactivates once the agent leaves the call.

    Note: The EMK event message that states the agent has exited the call also implies that the agent has terminated the EMK call.

- Caller releases call
  - — If the caller exits while either the supervisor or auxiliary device is still ringing, the caller removes the ringing party from the call. At this point, there can be one, two or three parties still active on the call.
  - — The host application receives an EMK event message stating that the caller has exited the EMK call.
  - — If there is only one call, the ACD agent releases the call. The host computer receives an additional EMK event message stating that the EMK event has terminated.
  - — If there are two parties left, the ACD agent and either the AEMK key or the auxiliary device, the conference call between them is still active. The EMK feature is no longer active with only two parties on the call. The host computer receives another EMK event message stating that the EMK event has terminated.
  - — If the caller exits and there are three parties still active on the call, the agent, the AEMK key and auxiliary device, the EMK feature is active because there are more than 2 parties on the call. The DMS-100 switch does not send an EMK event message until one of the remaining parties exits from the call, making the EMK feature inactive. At that time, the host computer receives an EMK event message stating that the EMK event has terminated.
- Supervisor or auxiliary device exits
  - — The DMS-100 switch sends an EMK event message when a supervisor and/or Auxiliary device exits. The message says either the supervisor or auxiliary device has exited the call. What follows depends on the number of parties remaining on the call.
  - — If three parties remain, the emergency key feature remains active and the switch does not send another EMK event message. If only two parties remain, the EMK feature is no longer active and the host computer receives an EMK event message saying that the EMK event has terminated.

4.5.6.4 dv-EMK-U parameters

The following parameters are optional except EMKInfo.

NetworkCallID a CompuCALL generated call ID composed of a network node ID and a local call ID. This is a new if it is the first instance of the call within the session.

AgtPosID identifies the position ID of the agent.

AgtDN identifies the DN of the ACD agent.

SuprvPID identifies the position ID of the supervisor.

SuprvDN identifies the AEMK DN of the supervisor. The field is zero only if the auxiliary device is available.

EMKInfo identifies if the agent activates or terminates the EMK. If the EMK terminates, identify the reason for termination.

4.5.6.5 dv-EMK-U validation

The switch validates the following CompuCALL setup before the host application can receive the dv-EMK-U message:

1. The switch establishes a CompuCALL host session. The host computer sends a successful dv-Appl-Logon request.
2. The dv-DN-Associate message associates the line which the host application is monitoring the EMK event to the session.
3. The EMK event message subscribes the switch-host session.
4. The service version is at least SCAI11.

4.5.6.6 dv-EMK-U: ASN.1 encoding

The ASN.1 notations for the messages provided by this activity are in this section. Following each notation is a sample X.209 hex encoding. The length fields are variable because the length depends on the variable content. The tag values are from the ROSE protocols.

```
dv-EMK-U                        OPERATION
                                ARGUMENT  EMKArgument
::= 43

EMKArgument       ::=     SEQUENCE
{networkCallID    [0] IMPLICIT SEQUENCE
   {networkNodeID [0] IMPLICIT INTEGER      OPTIONAL,
   localCallID    [1] IMPLICIT INTEGEROPTIONAL},

AgtPosID          [1] IMPLICIT INTEGER            OPTIONAL,
AgtDN             [2] IMPLICIT OCTET STRING       OPTION     AL,
SuprPosID         [3] IMPLICIT INTEGER            OPTIONAL,
SuprAEMKDN        [4] IMPLICIT OCTET STRING       OPTIONAL,
AuxDN             [5] IMPLICIT OCTET STRING       OPTIONAL,
EMKinfo           [6] IMPLICIT ENUMERATED
{Activated        (0),
SuprTimeOut       (1),
AuxTimeOut        (2),
AgentCancels      (3),
NotUsed           (4),
AgentExits        (5),
CallerExits       (6),
SuprExits         (7),
AuxExits          (8),
Deactivated       (9)}                    OPTIONAL}
```

4.5.6.7 dv-EMK-U successful activation: coded example

INVOKE DMS ===== > HOST

Table 29 dv-EMK-U successful activation

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	37		INVOKE	Message length is 55 bytes.
02	01	01	InvokeID	Invoke ID is 1.
02	01	2B	Operation	Operation ID for dv-EMK-U is decimal 43, hex 2B.
30	2F		EMKArgument	Type sequence and 47 bytes long.
A0	06		NetworkCallID	NetworkCallID is type constructor and is 6 bytes long.
80	01	01	networkNodeID	networkNodeID is 1.
81	01	02	localCallID	local call ID is 02.
81	04	32 33 30 30	AgtPosID	Position ID of Agent 2300
82	0A	36 31 33 36 31 32 32 33 30 30	AgtDN	DN for the agent 613-621-2300
83	04	35 37 30 30	SuprvPID	Position ID of supervisor 5700
84	0A	36 31 33 36 32 31 35 37 30 30	SuprvDN	Supervisor DN is 6136215700
86	01	00	EMKinfo	activated EMK
Hex dump = A1 37 02 01 01 02 01 2B 30 2F A0 06 80 01 01 81 01 02 81 04 32 33 30 30 82 0A 36 31 33 36 31 32 31 33 30 30 83 04 35 37 30 30 84 0A 36 31 33 36 32 31 35 37 30 30 86 01 00				

4.5.6.8 dv-EMK-U termination: coded example

Table 30 dv-EMK-U termination

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	37		INVOKE	Message length is 55 bytes.
02	01	01	InvokeID	Invoke ID is 1.
02	01	2B	Operation	Operation ID for dv-EMK-U is decimal 43, hex 2B.
30	2F		EMKArgument	Type sequence and 47 bytes long.
A0	06		NetworkCallID	NetworkCallID is type constructor and is 6 bytes long
80	01	01	networkNodeID	networkNodeID is 1
81	01	02	localCallID	local call ID is 02
81	04	32 33 30 30	AgtPosID	Position ID of Agent 2300
82	0A	36 31 33 36 31 32 32 33 30 30	AgtDN	DN for the agent 613-621-2300
83	04	35 37 30 30	SuprvPID	Position ID of supervisor 5700
84	0A	36 31 33 36 32 31 35 37 30 30	SuprvDN	Supervisor DN is 6136215700
86	01	01	EMKinfo	Supr time-out terminates EMK
Hex dump = A1 37 02 01 01 02 01 2B 30 2F A0 06 80 01 01 81 01 02 81 04 32 33 30 82 0A 36 31 33 36 31 32 33 30 30 83 04 35 37 30 30 84 0A 36 31 33 36 32 31 35 37 30 30 86 01 01				

4.6 ACD Call routing

This Application Service capability covers two Application Service functions (i.e., messages):

- dv-Call-Received-C
- dv-Call-Redirect

4.6.1ACD dv-Call-Received-C (operation value "A" hex)

The switch uses the dv-Call-Received-C message to notify the host that a call to an ACD group belonging to an associated environment with which the host has an established session has been received. This allows the host to redirect the call. The "C" in the name means this message requires a confirmation from the host. This message is valid only for the Meridian ACD CompuCALL options.The switch sends this RO is of Operation Class 2 to the host when the switch receives a call to an ACD DN associated with the current session.

4.6.1.1ACD dv-Call-Received-C ASN.1 encoding

```
dv-Call-Received-C        OPERATION
                          ARGUMENT          CallReceivedArgument
                          RESULT
                          ERRORS            {  call-Received-Error}
                          LINKED            {  dv-Call-Redirect}
                          ::= 10

CallReceivedArgument      ::=      SEQUENCE
 {  networkCallID         [0] IMPLICIT SEQUENCE
      {  networkNodeID        [0] IMPLICIT INTEGER       OPTIONAL,
         localCallID          [1] IMPLICIT INTEGER       OPTIONAL}
                                                         OPTIONAL,
    acdDN                 [1] IMPLICIT OCTET STRING      OPTIONAL,
    origCallingNumber     [2] IMPLICIT OCTET STRING      OPTIONAL,
    origChargeNumber      [3] IMPLICIT OCTET STRING      OPTIONAL,
    callHistoryInfo       [4] IMPLICIT SEQUENCE
      {  callType             [0] IMPLICIT ENUMERATED
           {  callTransferred       (1),
              callOverflowed        (2),
              callRedirected        (3),
              callForwarded         (4),
              callConsult           (5),
              callConference        (6),
              callReturnedToQueue   (7),
              callRouted            (8)}             OPTIONAL,
         origInboundDN    [1] IMPLICIT OCTET STRING      OPTIONAL,
         prevApplicationID [2] IMPLICIT INTEGER          OPTIONAL,
          hostCallData          [3] IMPLICIT OCTET STRING    OPTIONAL}
                                                             OPTIONAL,
      acdGroup                  [5] AddressType              OPTIONAL,
      forwardingParty           [6] ForwardingPartyType      OPTIONAL,
      cpadigs                   [7] IMPLICT OCTET STRING     OPTIONAL}


AddressType                     :: =     CHOICE
 {    positionID                [0] IMPLICIT INTEGER,
      dialedDigits              [1] IMPLICIT IA5STRING,
      stationNumber             [2] IMPLICIT OCTET STRING,
      q931Address               [3] Q931AddressType}

Q931AddressType                 :: =     SEQUENCE
 {    numberTypeNumberPlan               [0] OCTET STRING      OPTIONAL,
      presentationScreeningIndicator     [1] OCTET STRING      OPTIONAL,
      digits                             [2] IA5 STRING        OPTIONAL}

ForwardingPartyType             :: =     IMPLICIT SEQUENCE
 {    firstFwdNumber                     [0] AddressType       OPTIONAL,
      firstFwdReason                     [1] IMPLICIT ENUMERATED
       {  unknown                            (0),
          userBusy                           (1),
          noReply                            (2),
          unconditional                      (3)}           OPTIONAL,
      lastFwdNumber                      [2] AddressType       OPTIONAL,
      lastFwdReason                      [3] IMPLICIT ENUMERATED
       {  unknown                            (0),
          userBusy                           (1),
          noReply                            (2),
          unconditional                      (3)}              OPTIONAL}


call-Received-Error             ERROR               ::= 0
```

4.6.1.2ACD dv-Call-Received-C: coded example 1

Switch #1 receives an ACD call initiated by the OrigCallingNumber 6137221152 and assigns a LocalCallID = 2 to the call meant for ACD group with supplementary DN = 6137221111 and primary DN of the ACD group = 6137229999. The switch sends dv-Call-Received-C message to the host who responds with Return Result message. Then the switch continues to route the call to the current ACD group. If the host determines the call should be redirected, it responds with a linked dv-Call-Redirect message (see dv-Call-Redirect in Appendix A). If the host finds an error, it sends a Return Error that dv-Call-Received-C had an error.

INVOKE DMS ===== > HOST

Table 31 dv-Call-Received-C invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	38		INVOKE	this message is 56 bytes decimal
02	01	32	InvokeID	the InvokeID is 50 decimal
02	01	0A	Operation	the operation is dv-Call-Received-C[10] decimal
30	30		Argument	CallReceived argument is of sequence type and is 48 bytes long decimal
A0	06		NetworkCallId	the NetworkCallId is a constructor 6 bytes
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	01	02	LocalCallId	the LocalCallId is 2 decimal
81	0A	36 31 33 37 32 32 31 31 31 31	ACDDN	the called ACDDN is a supplementary DN = 6137221111 decimal
82	0C	00 83 36 31 33 37 32 32 31 31 35 32	OrigCallNumber	the first two bytes are fixed and presently unused. The customer's OrigCallNumber is 6137221152
A5	0C		ACDGroup	the ACDGroup is a type constructor 12 bytes
82	0A	36 31 33 37 32 32 39 39 39 39	StationNumber	the customer group's Primary ACDDN is 6137229999
Hex Dump = A1 38 02 01 32 02 01 0A 30 30 A0 06 80 01 01 81 01 02 81 0A 36 31 33 37 32 32 31 31 31 31 82 0C 00 83 36 31 33 37 32 32 31 31 35 32 A5 0C 82 0A 36 31 33 37 32 32 39 39 39 39				

RETURN-RESULT DMS <===== HOST

Table 32 dv-Call-Received-C return result

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	32	InvokeID	the InvokeID is 50 decimal
Hex Dump = A2 03 02 01 32				

RETURN-ERROR DMS <===== HOST

Table 33 dv-Call-Received-C return error

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	06		RETURN RESULT	this message is 3 bytes long decimal
02	01	32	InvokeID	the InvokeID is 50 decimal
02	01	00	Error Value	the Error Value is Call-Received-Error (0)
Hex Dump = A3 06 02 01 32 02 01 00				

The switch sends the dv-Call-Received-C message to the host when a call to an ACD group belonging to an associated environment with which the host has established a session is received by the switch. The host must send an acknowledgment to the switch within 1 to 30 seconds, defined by the customer. Two seconds is recommended.

If the service profile for the session of the associated environment contains the dv-Call-Queued-U and dv-Call-Received-C messages and the call redirection capability is active for that ACD group, dv-Call-Received-C, not dv-Call-Queued-U is sent by DMS-100 for that ACD group. In other words, either dv-Call-Received-C or dv-Call-Queued-U is sent to the host when a call is received by a given ACD group, but not both.

The dv-Call-Received-C message is sent before the call is determined to be routed to Night Service, overflowed (immediate overflow), queued in the incoming call queue or offered to an agent. The host may redirect the call to another destination (which could then redirect it if another ACD group). dv-Call-Queued-U differs from dv-Call-Received-C in that dv-Call-Queued-U is sent after the switch determines to queue the call.

Call redirection (although subscribed for the ACD group) is not allowed for:

1 Calls "overflowed" into the same group (by Ring Threshold, Not Ready or Make Set Busy).

2 Calls which overflow to another group after being queued which may be logically queued in both groups (Time Delay Overflow or Overflow From Queue).

3 The third consecutive time (calls redirected two subsequent times without intermittent transfer, overflow, or forward).

4 The same DN subsequently.

For these calls, the dv-Call-Received-C is not sent to the host. Instead, dv-Call-Queued-U message is sent if the customer subscribes to that message and if the call is queued.

The switch expects either a return Result (positive acknowledgment), a Return Error (negative acknowledgment), or a dv-Call-Redirect message in response to a dv-Call-Received-C message. If no Return Result message is received, or if a Reject or Return Error message is received, the switch logs this as a fault occurrence. Only the switch operator can only access this information.

4.6.1.3 ACD dv-Call-Received-C: coded example 2

The OrigCallingNumber 9192466001 initiates an ACD call by dialing 9197224444. The call has a LocalCallID = 4 and goes to the Primary DN of the ACD group =9194815200. The switch sends a dv-Call-Received-C message to the host which responds with either a Return Result message or a Return Error message.

INVOKE DMS ===== > HOST

Table 34 dv-Call-Received-C example 2 invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	39		INVOKE	this message is 57 bytes decimal
02	02	03 F9	InvokeID	the InvokeID is 1017 decimal
02	01	0A	Operation	the operation is dv-Call-Received-C[10] decimal
30	30		Argument	CallReceived argument is of sequence type and is 48 bytes long decimal
A0	06		NetworkCallId	the NetworkCallId is a constructor 6 bytes
80	01	00	NetworkNodeId	the NetworkNodeId is 0
81	01	04	LocalCallId	the LocalCallId is 4
81	0A	39 31 39 34 38 31 35 32 30 30	ACDDN	the called ACDDN is 9194815200
82	0C	00 83 39 31 39 32 34 36 36 30 30 31	OrigCallNumber	the first two bytes are fixed and presently unused. The customer's OrigCallNumber is 9192466001
A5	0C		ACDGroup	the ACDGroup is a type constructor 12 bytes
82	0A	39 31 39 34 38 31 35 32 30 30	StationNumber	the customer group's Primary ACDDN is 9194815200
87	0A	39 31 39 37 32 32 34 34 34 34	cpadigs	The CPA value is 9197224444
Hex Dump = A1 39 02 02 03 F9 02 01 0A 30 30 A0 06 80 01 00 81 01 04 81 0A 39 31 39 34 38 31 35 32 30 30 82 0C 00 83 39 31 39 32 34 36 36 30 30 31 A5 0C 82 0A 39 31 39 34 38 31 35 32 30 30 87 0A 36 31 33 37 32 32 34 34 34 34				

RETURN-RESULT DMS <===== HOST

Table 35 dv-Call-Received-C example 2 return result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	32	InvokeID	the InvokeID is 50 decimal
Hex Dump = A2 03 02 01 32				

RETURN-ERROR DMS <===== HOST

Table 36 dv-Call-Received-C example 2 return error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN RESULT	this message is 6 bytes long decimal
02	01	32	InvokeID	the InvokeID is 50 decimal
02	01	00	Error Value	the Error Value is Call-Received-Error (0)
Hex Dump = A3 06 02 01 32 02 01 00				

4.6.1.4 dv-Call-Received parameters

The dv-Call-Received-C message has the same parameters as the dv-Call-Queued-U message. A brief description follows.

NetworkCallID contains static call-related information identifying the call in context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

ACDDN contains dynamic call-related information indicating the primary or supplementary ACD DN associated with the call. This may not be the actual number dialed by the calling party (if 800 service it may be the 10-digit NANP routing number translated from the dialed 800 number).

OrigCallingNumber contains static call-related information indicating the original calling number associated with a call when it enters CompuCALL. For example, this may be a 10-digit North American

Numbering Plan (NANP) DN associated with the calling line (or if a call from a PBX, it may be the Listed DN (LDN) of the PBX), or a private number.

OrigChargeNumber contains static call-related information indicating the operating company charge number (also called "ANI" - Automatic Number Identification in North America) associated with the incoming call when it enters the CompuCALL environment. A given dv-Call-Received-C message will contain both the OrigCallingNumber and OrigChargeNumber parameters, if both are available.

CallHistoryInfo if the call were transferred, overflowed, or redirected from an ACD group or agent, or transferred from or forwarded by an MDC customer workstation.

Each subparameter is datafilled in the switch as being provided (when appropriate) or not provided on an individual basis. In other words, the CallHistoryInfo parameter is present if the customer subscribed to any of the following subparameters:

CallType contains dynamic call-related information indicating if the call to the current CompuCALL party were transferred, overflowed, redirected, forwarded, or routed.

Note: If a consult call, the consult party (if associated to a session) receives the dv-Call-Received-C with CallType "transferred" indicating the call is a consult call in the context of the call transfer feature, i.e., it could be not has been transferred.

OrigInboundDN contains static call-related information identifying the first DN at which the call is directed when it enters the CompuCALL environment. It remains the same as long as the call stays within the CompuCALL environment even if it is later transferred, overflowed, redirected, or forwarded.

PrevApplicationID contains dynamic call-related information identifying the ApplicationID for the session of the Associated environment to which the party extending the call belongs (call extension includes call transfer, overflow, redirect, and forward).

HostCallData contains dynamic call-related information of a generic and miscellaneous nature provided by a host. If the call were redirected by a host, this parameter may be provided by that host at that time, and may include the reason for the redirection.

ACDGroup contains dynamic call-related information identifying the primary ACD DN of the ACD group in the CompuCALL environment where the call was queued. It lets the host correlate the dv-Call-Received-C message with an associated DN. The choice is StationNumber.

StationNumber identifies the ACD group to which the incoming call is routed.

CPADIGS contains the Called Party Address (CPA) digits as received in the setup messages for the incoming agent, displayed as a parameter in ICM messages. The first call set-up message, in the case of ISUP the Initial Address message (IAM), sent from the originating switch contains a parameter for the CPA. The terminating switch uses the CPA value to route the call. The exact address digits contained in the CPA parameter is delivered to the ICM environment. The CPADIGS field contains the dialled digits for calls to the ICM environment that arrive over a line.

In the case of overlap signalling, the originating switch sends the Subsequent Address message (SAM) following an IAM to convey additional called party address information. The CPA that arrives in the IAM and SAM, before the terminating switch sends the Address Complete Message (ACM), is delivered to the ICM environment. A CPA that arrives in the SAM after the ACM, is ignored along with the SAM. As long as the local_call_id of the call remains the same for subsequent extensions of the call, the CPA parameter in the ICM messages remains static. For different local_call_id's, the value of CPA parameter differs in an ICM environment.

ForwardingParty identifies the first forwarding number and associated reason and last forwarding number and reason for forwarding. In a single forwarding scenario, the information is included in the FirstFwdNumber and First FwdReason subparameters. This information is provided to the host only if presentation of the forwarding DNs are allowed. The parameter is a constructor including the information:

FirstFwdNumber identifies the number from which the first forward was made in a multiple call forwarding scenario. Based on the Q.931 format, it uses the Q931Address choice.

FirstFwdReason identifies the reason the call was first forwarded. It is an enumerated type with values: unknown/not available, user busy, no reply or unconditional.

LastFwdNumber identifies the number from which the lastforward was made in a multiple call forwarding scenario. Based on the Q.931 format, it uses the Q931Address choice.

LastFwdReason identifies the reason the call was forwarded the last time (most recent). It is an enumerated type with values: unknown/not available, user busy, no reply or unconditional.

When the ACD host computer receives a dv-Call-Received-C message, it responds with one of the following messages:

- dv-Call-Redirect message
  - — If the host determines the call should be redirected, it responds with a linked dv-Call-Redirect message.
- Return Result
  - — If the host determines call redirection is not required, it responds with a Return Result message. The switch then continues to route the call to the current ACD group. The Return Result has no parameters.
- Return Error
  - — If the host responds with a Return Error message, the switch continues to route the call to the current ACD group.
- Reject
  - — If the host responds with a Reject message indicating a protocol error or a message received out of sequence, the switch continues to route the call to the current ACD group.
- No Response from host
  - — If the host does not respond to a dv-Call-Received-C message within a predetermined time period, the switch continues to route the call to the current ACD group.
  - — If the host does not respond within this predetermined time, the log system reports a LOG indicating the host is not responding. This log is reported each time this occurs. Only the switch operator can accessed this information.
- Response Received Out of sequence (after time-out)
  - — These messages are treated as extraneous and the switch responds to the host with a Reject message. There are no OMs or logs for this situation.

4.6.2 ACD dv-Call-Redirect (operation value "B" hex)

The host uses the dv-Call-Redirect message to redirect a call going to an ACD group belonging to an Associated environment with which the host has an established session to another destination. This message is valid for the Meridian ACD CompuCALL options only. The host sends this RO of Operation Class 2 to the switch to redirect an ACD call.

4.6.2.1 dv-Call-Redirect ASN.1 encoding

```
dv-Call-Redirect               OPERATION
                               ARGUMENT       CallRedirectArgument
                               RESULT
                               ERRORS         {  missing-Parameter,
                                                 invalid-Parameter,
                                                 not-Allowed,
                                                 dest-Unavailable}
                               ::= 11


CallRedirectArgument           ::=      SEQUENCE
 {  networkCallID              [0] IMPLICIT SEQUENCE
     {  networkNodeID              [0] IMPLICIT INTEGER        OPTIONAL,
        localCallID                [1] IMPLICIT INTEGER        OPTIONAL}
                                                               OPTIONAL,
    redirectDestination        [1] AddressType                 OPTIONAL,
    hostCallData               [2] IMPLICIT OCTET STRING       OPTIONAL}


AddressType                    :: =     CHOICE
 {  positionID                 [0] IMPLICIT INTEGER,
    dialedDigits               [1] IMPLICIT IA5STRING,
    stationNumber              [2] IMPLICIT OCTET STRING,
    q931Address                [3] Q931AddressType}

Q931AddressType                :: =     SEQUENCE
 {  numberTypeNumberPlan                [0] OCTET STRING       OPTIONAL,
    presentationScreeningIndicator      [1] OCTET STRING       OPTIONAL,
    digits                              [2] IA5 STRING         OPTIONAL}

missing-Parameter              ERROR          ::= 3
                               PARAMETER      MissingParameter
MissingParameter               ::=            SEQUENCE
 {  missingParameterType       [0] IMPLICIT ENUMERATED
     {  missingNetworkCallID            (0),
        missingRedirectDestination      (1),
        missingHostCallData             (2)}                   OPTIONAL}

                               --  only the first (i.e., one) missing parameter
                               --  encountered is reported in this message

invalid-Parameter              ERROR          ::= 4
                               PARAMETER      InvalidParameter
```

```
InvalidParameter            :=     SEQUENCE
 {  invalidParameterType    [0] IMPLICIT ENUMERATED
     {  invalidNetworkCallID       (0),
        invalidRedirectDestination(1),
        invalid HostCallData       (2)}                  OPTIONAL}

                            --  only the first (one) invalid parameter
                            --  encountered is reported in this message


not-Allowed                 ERROR             ::= 5

dest-Unavailable            ERROR             ::= 6
```

4.6.2.2 ACD dv-Call-Redirect: coded example

Determining to redirect the call, the host responds to the dv-Call-Received-C (10) message from the switch with a linked dv-Call-Redirect (11) message. This contains the DN where the call will be redirected DialedDigits = 6137221220. The switch responds with Return Result indicating only that the switch accepted the dv-Call-Redirect message, not that the call was successfully routed to the RedirectDestination. If the dv-Call-Redirect has a missing NetworkCallID parameter, the switch responds with a Return Error message that the NetworkCallID parameter is missing.

INVOKE DMS ===== > HOST

Table 37 dv-Call-Redirect invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	2E		INVOKE	this message is 46 bytes decimal
02	01	3C	InvokeID	the InvokeID is 60 decimal
80	01	32	LinkedInvokeId	the LinkedInvokeId is 50 decimal
02	01	0B	Operation	it operation value is dv-Call-Redir[11] decimal
30	23		Argument	the CallRedirectArgument is of type sequence and is 35 bytes long decimal
A0	07	01	NetworkCallId	this of type sequence 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
A1	0C		RedirectDestination	this is a constructor 12 bytes long decimal
81	0A	36 31 33 37 32 32 31 32 32 30	DialedDigits	the DialedDigits are (613)722-1220
82	0C	52 65 64 69 72 20 31 32 32 30	HostCallData	this is 10-bytes
Hex Dump = A1 2E 02 01 3C 80 01 32 02 01 0B 30 23 A0 07 01 80 01 01 81 02 05 55 A1 0C 81 0A 36 31 33 37 32 32 31 32 32 30 82 0A 52 65 64 69 72 20 31 32 32 30				

RETURN-RESULT DMS=====>HOST

Table 38 dv-Call-Redirect return result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	3C	InvokeID	the InvokeID is 60 decimal
Hex Dump = A2 03 02 01 3C				

RETURN-ERROR DMS=====>HOST

Table 39 dv-Call-Redirect return error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	3C	InvokeID	the InvokeID is 60 decimal
02	01	03	ErrorType	the ErrorType is MissingParameter[3]
30	03		ReturnErrorSequence	MissingParameter is a sequence 3 bytes
80	01	00	ErrorParameter	the MissingParameter missingNetworkCallID(0)
Hex Dump = A3 0B 02 01 3C 02 01 03 30 03 80 01 00				

The dv-Call-Redirect message is a "linked operation" RO as a response to dv-Call-Received-C. There will be a Return Result or Return Error received for "linked operations" before a Return Result or Return Error is given for the first, or parent, RO. A maximum of 2 consecutive call redirections is allowed for a given call within a switch after which the call remains at the last redirected destination. For each of the two consecutive times, the DMS-100 sends a dv-Call-Received-C message to the host. Upon receiving the second dv-Call-Redirect message from the host, the DMS-100 sends a dv-Call-Queued-U instead of a dv-Call-Received-C message to the host at the last redirected (i.e., the third) destination, assuming that dv-Call-Queued-U is in the service profile.

Call redirection to the Incalls keys of specific ACD agents is possible as well as call redirection to any dial DN. There are certain conditions under which call redirection to specific ACD agent is not allowed described under Return Error. However, if the agent is logged in but busy on a call, an attempt will be made to add the call to the ACD agent's personal agent queue only if the ACD agent has the ACDXFER feature.

4.6.2.3 dv-Call-Redirect parameters

A brief description of the parameters of the dv-Call-Redirect invoke follows.

NetworkCallID contains static call-related information uniquely identifying the call in context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

RedirectDestination specifies the destination to which the call is to be redirected. It is a choice parameter. DialedDigits is the available choice.

DialedDigits is a string of IA5-encoded digits corresponding to the dialed digits if the dialing were performed manually. The digits are translated and routed according to the business group (currently equivalent to MDC customer group) and Network Class of Service (NCOS) datafill set up for the originator of the call, in this case, the ACD group.

HostCallData is for use in some customer configurations where an ACD call is redirected by the host to another CompuCALL destination resulting in dv-Call-Queued-U or dv-Call-Received-C, dv-Call-Offered-U and/or dv-Call-Answered-U messages being sent to another host with which the first host wishes to communicate call-related information if no direct communication between the two hosts. In this case, the HostCallData parameter might indicate to the second host computer why the call has been redirected. This information is provided by the host; the switch just passes it on to the redirected destination. If the service profile for that session contains the HostCallData parameter in the dv-Call-Redirect message, it may be used for all customer ACD groups with the call redirection capability active. However, DMS-100 does not respond with Return Error if the host does not include a HostCallData parameter in dv-Call-Redirect messages related to those ACD groups.

This message must contain the same NetworkCallID as the corresponding dv-Call-Received-C and must be sent on the same session as dv-Call-Received-C to the host.

When the DMS-100 receives a dv-Call-Redirect message, it redirects the call according to the RedirectDestination parameter in the message. It also sends a dv-Call-Released-U message to the host indicating the ACD group is no longer involved in the call.

The DMS-100 responds with one of the following messages:

- Return Result
  - The switch acknowledges the dv-Call-Redirect message with a Return Result message which contains no parameters. This indicates only that the switch accepted the dv-Call-Redirect message; it does not necessarily indicate that the call has been successfully routed to the redirect destination, i.e., the call is redirected without any validation of the redirect destination, except when it is to be redirected to an individual ACD agent destination.
  - The switch must acknowledge the dv-Call-Redirect message before the host can acknowledge the dv-Call-Received-C message.
- Return Error

- — Or the switch may respond with a Return Error message indicating if the dv-Call-Redirect message has not been accepted due to:
  - – missing parameter (identified)
  - – invalid parameter (identified)
  - – request not allowed
  - – specific ACD agent destination unavailable to receive call (see Note)
- Reject
  - — A protocol violation error will be responded to with a Reject message.

This list is presented in the order of priority since only the first error condition (including the first missing/invalid parameter) encountered by the switch is included in the Return Error message.

4.6.2.4 dv-Call Redirect error conditions

The following are conditions under which redirection to individual ACD agents is not possible. In all of the cases below, the call proceeds to the default route after a Return Error message (with an Error Reason of Dest-Unavailable) is sent.

1 The call is redirected to an ACD agent who is not logged in.

2 The call is redirected to an ACD agent who is unable to receive calls due to the agent's ACD group being in night service or has CIF activated.

3 The call is redirected to an ACD agent whose ACD group is in an overflow condition (i.e., maximum call queue size or maximum wait time threshold reached).

4 The call is redirected to a non-idle ACD agent whose ACD group call transfer queue limit is reached.

5 The call is redirected to an ACD agent whose personal agent queue limit is reached.

6 The ACD agent's ACDNR feature is activated and ACDXFER is not assigned to ACD group.

7 The call is redirected to a specific agent in a different customer group.

4.7 ACD Third Party Call Control

Third Party Call Control (TPCC) lets an ACD agent with a voice set and a data terminal initiate and manage calls at a data terminal. This application service capability covers the following application service functions (messages):

- dv-Answer-Call
- dv-Hold-Call
- dv-Release-Call
- dv-Unhold-Call
- dv-Call-Unheld
- dv-Call-Consult-Originated-U
- dv-Call-Conferenced-U
- dv-Call-Transferred-U
- dv-Add-Party
- dv-Conference-Party
- dv-Drop-Party
- dv-Transfer-Party
- dv-Make-Call
- dv-Controller-Released-Call-U
- dv-Noncontroller-Released-Call-U

TPCC is valid for the Meridian ACD CompuCALL options only. The dv-Make-Call message can be used by the host to assist the agent in establishing outbound calls. The dv-Add-Party, dv-Transfer-Party, dv-Drop-Party and dv-Conference-Party messages collectively called call party messages can be used to assist the agent in establishing consult, transfer and conference calls.

The following provides further detail on the steps taken by the switch in response to the Call Party messages. Figure 42 provides a simplified state diagram.

1 When an ACD agent (controller) is involved in an active call, they may request through the host that a consult call be established. To do this, the host sends a dv-Add-Party message to the switch. The original call party is put on hold, and a call is placed to the destination (consult party) specified by the host. The controller is now connected to the consult party but cannot talk to the original call party who is now on hold.

> **Note:** After the dv-Add-Party has been accepted and a consult call has been established, the controller can always hit the RLS key (on MBS) or go on-hook (on a 2500 set). In both cases the consult party will be

released and the controller will be re-rung if being on-hook (call released with "party dropped" will be sent) and on answer will be reconnected to the held original call party.

2 Given this scenario, the host may transfer the call by sending dv-Transfer-Party to drop the controller from the call and create a call between the held original call party and the consult party. This is called a held transfer. The consult party need not have answered the call for the call to be transferred. This means the host can also do a blind transfer.

> **Note:** If transfer is not allowed due to the controller not having the MDC Call Transfer feature enabled on the DMS-100, the consult party is released and the call is reverted to the original call party. (In the existing MDC case, both the original call party and the consult party are released.)

3 Given the consult scenario above, the host may conference the call by sending dv-Conference-Party. This would establish a three-way conference call for the controller with both the original call party and the consult party. The consult party need not have answered the call for conferences. All three parties are now active on the same call and can talk to one another (Figure 45).

> **Note:** Once a conference call is initiated or established, if a dv-Drop-Party message is sent or the controller depresses the RLS key or the Not Ready key with immediate cut-off (on MBS) or goes on-hook (on a 2500 set), the consult party will be dropped and the controller will be connected to the original call party, (Figure 47). For 2500 sets, the consult party is dropped and the controller rings. Or the controller can transfer the call by sending a dv-Transfer-Party if the controller's call transfer capability is enabled on the DMS-100 (Figure 44).

4 Once the consult or conference call has been initiated or established, the host can send dv-Drop-Party to cause the consult party to be released and the call between the controller and original call party re-established. The consult party need not be in the talking state in order for it to be dropped. Drop only applies to the consult party and may not be used to release the original call party or the controller (Figure 47 and Figure 47).

Figure 42 CompuCALL call party simplified state diagram

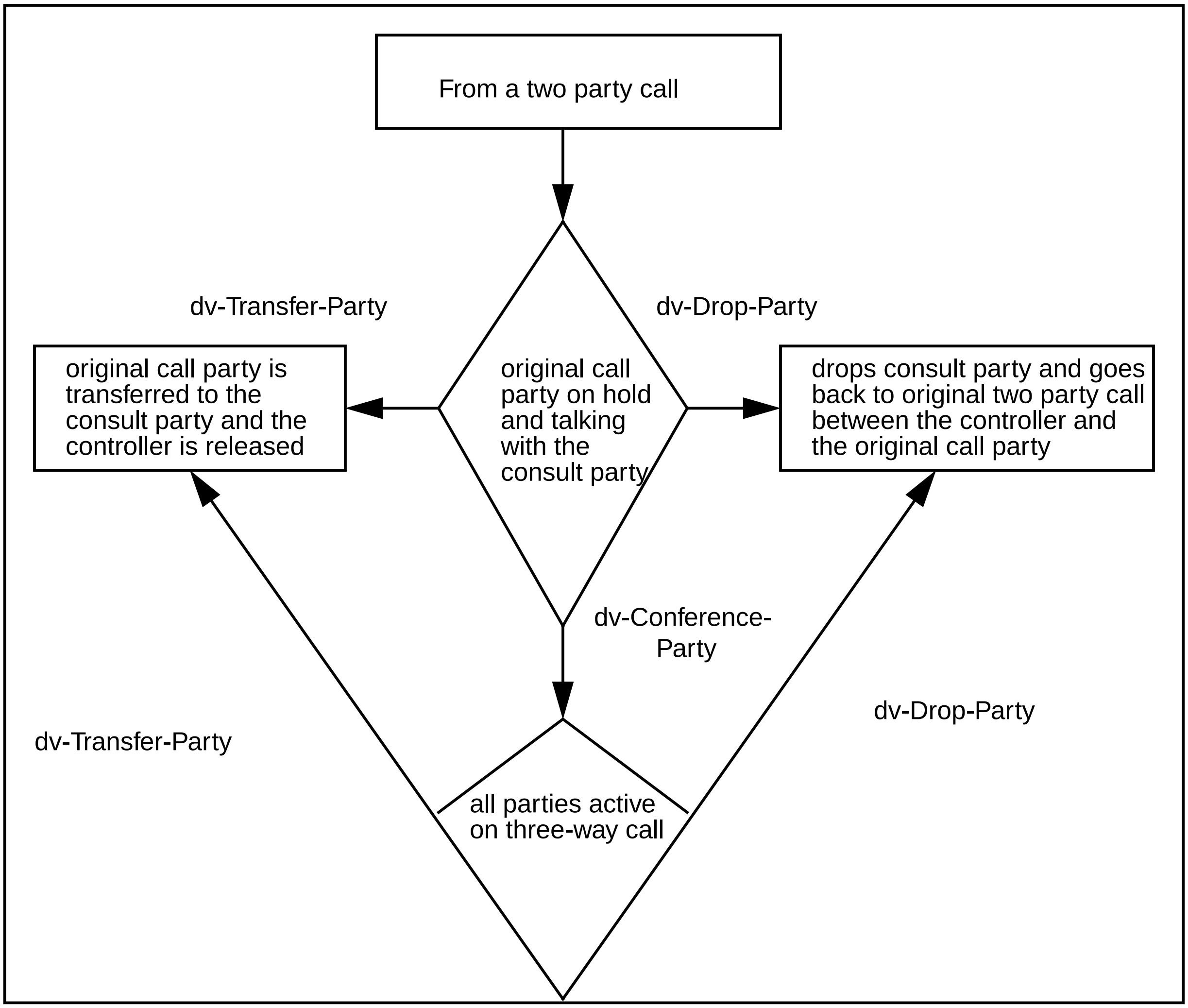

From a two party call
dv-Transfer-Party
dv-Drop-Party
original call party is transferred to the consult party and the controller is released
original call party on hold and talking with the consult party
drops consult party and goes back to original two party call between the controller and the original call party
dv-Conference-Party
dv-Drop-Party
dv-Transfer-Party
all parties active on three-way call

Figure 43 Consult (dv-Add-Party)

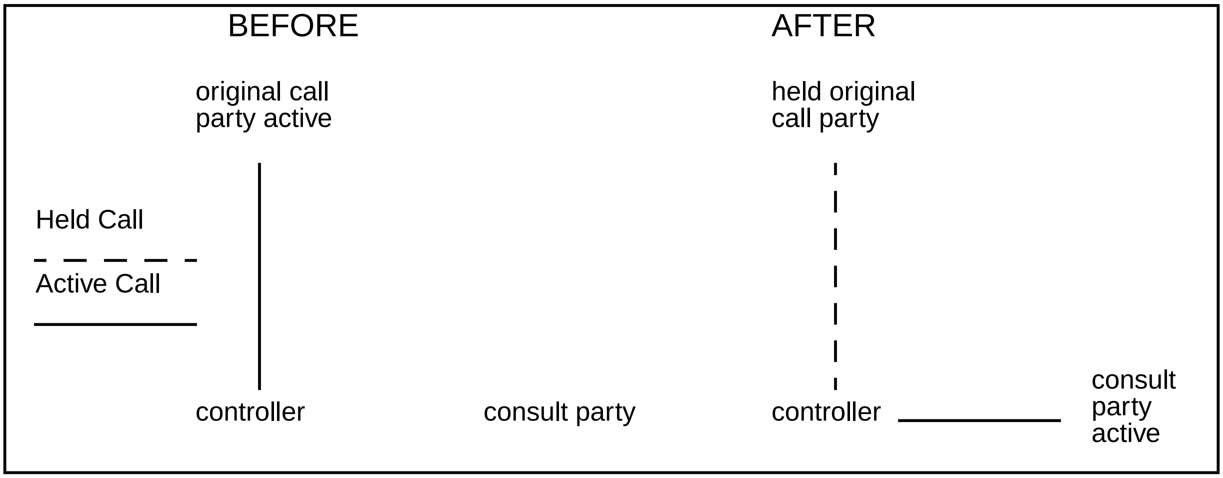


Figure 44 Transfer from a two-way call (dv-Transfer-Party)

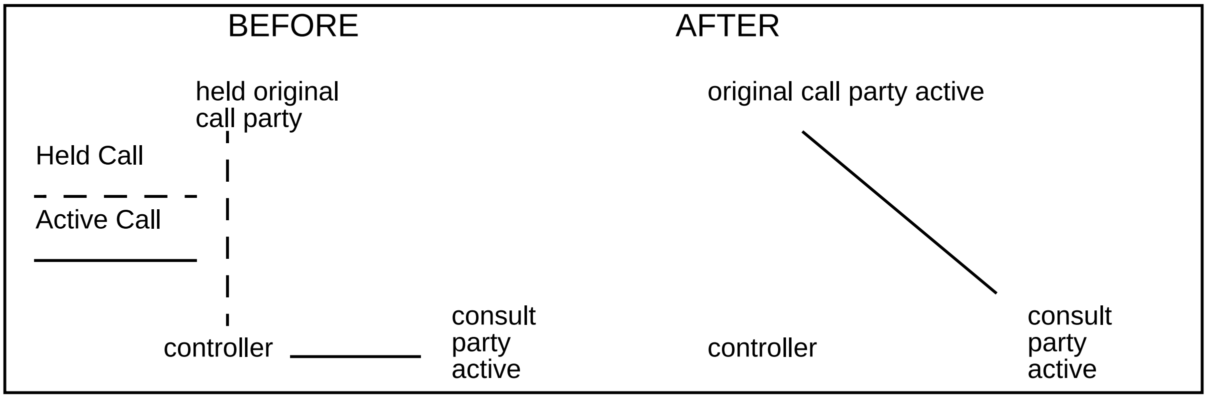


Figure 45 Transfer from a conference call (dv-Transfer-Party)

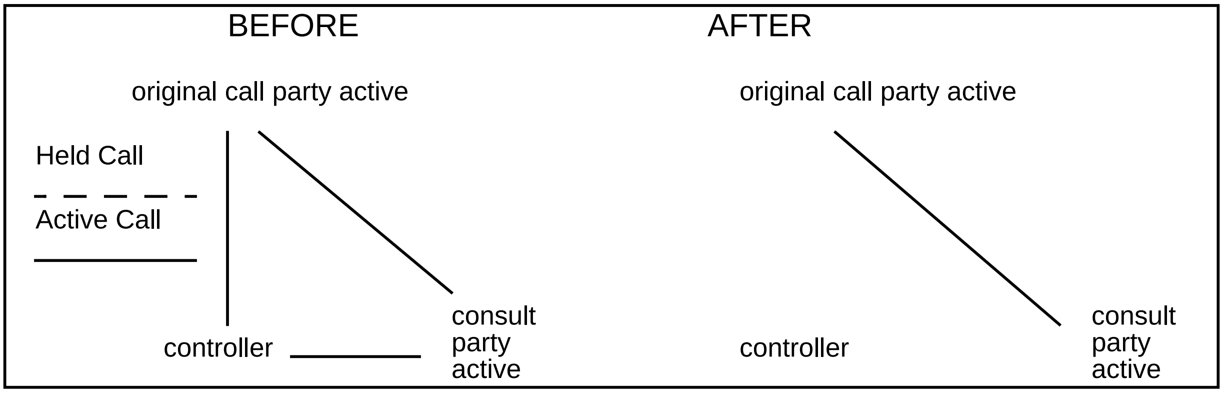


Figure 46 Conference (dv-Conference-Party)

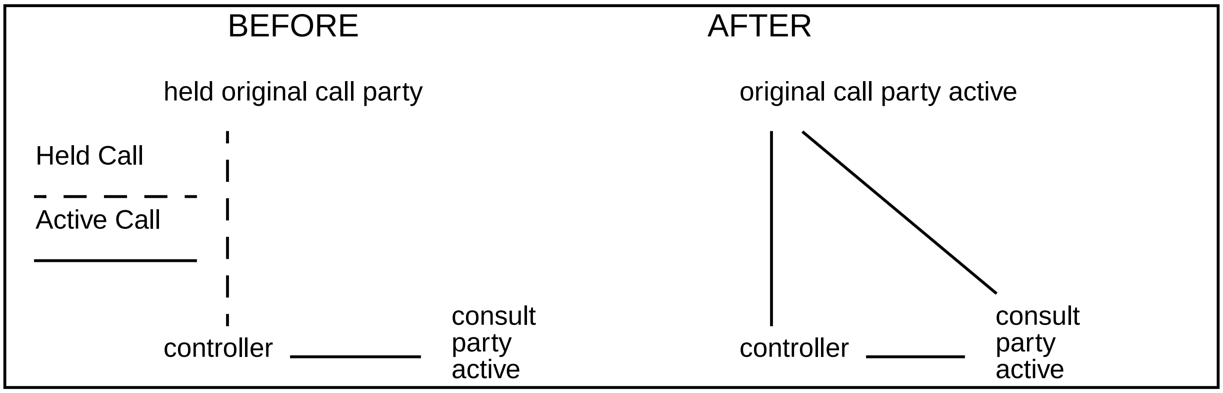


Figure 47 Drop from a two-way call (dv-Drop-Party)

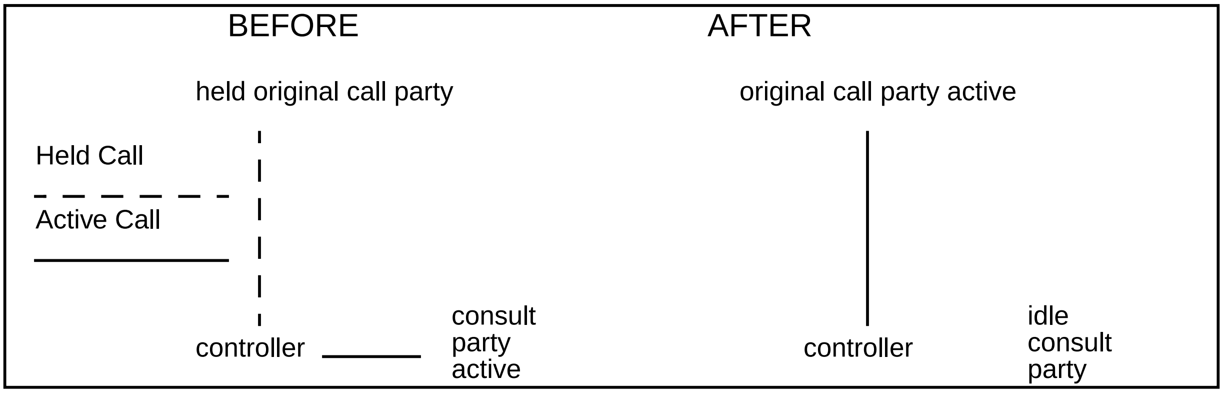
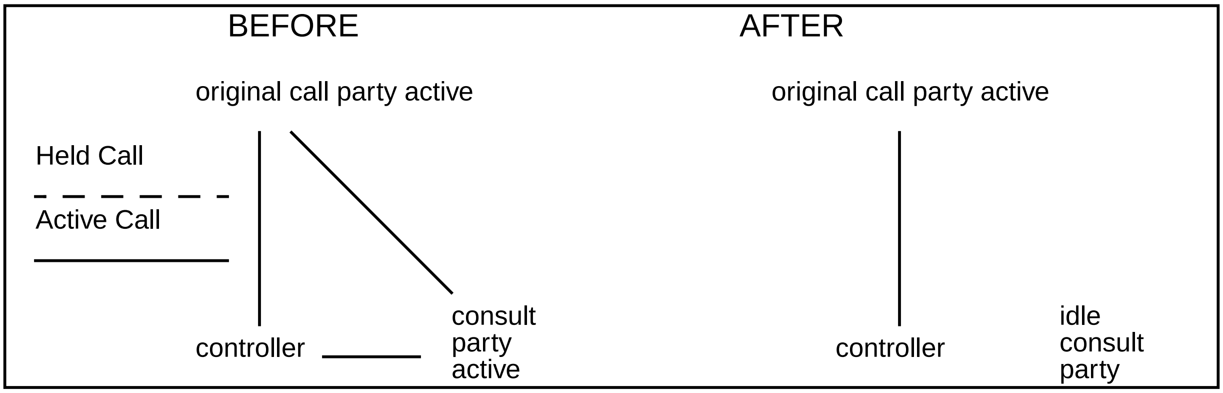


Figure 48 Drop from a conference call (dv-Drop-Party)



These 4 Call Party messages (dv-Add-Party, dv-Transfer-Party, dv-Drop-Party and dv-Conference-Party) provide the same functionality as Meridian Digital Centrex (MDC) 3-way-call (3WC), call transfer and call conference, but through a different interface. The existing MDC 3WC/call transfer/call conference feature restrictions apply as well as any imposed by the Meridian CompuCALL options application.

CompuCALL call party messages and 3WC/call transfer/call conference (set features) are very similar in functionality, but there are a few differences between the two:

1 In CompuCALL, if the controller has an MBS and the consult party exits a consult call via dv-Drop-Party, the held original call party is reconnected to the controller immediately. In normal 3WC consult scenario, the connection to the original party is on hold and the controller must hit the Incalls key to access the held original call party. If the consult party exits a conference call via dv-Drop-Party, then the controller will be automatically connected to the original call party.

2 In CompuCALL, if the controller has an MBS and goes on-hook during a consult call, the controller is re-rung for the held call to the original call party (ringback). However, if the controller has an MBS and hits the release (RLS) key during a CompuCALL consult call, the controller will be automatically connected to the original call party, i.e., no ringback. In a normal MDC 3WC scenario, the connection to the consult party is released, and the controller is automatically connected to the original call party if the controller presses the release key on an MBS, or if the controller has gone on-hook on a 2500 set. The connection to the held original call party remains on hold and the controller must hit the INCALLS key to access the original call party.

3 In CompuCALL, if the controller has an MBS in a conference call and hits the release (RLS) key or goes on-hook, the third party is dropped and the controller rings for the held call to the original call party (ringback). In a normal 3WC scenario, the connection to the controller is released and the original call party is transferred to the consult party to have a normal two-way call if the controller's call transfer capability is enabled on the DMS-100.

The ACD features like Not Ready and Make Set Busy interact with CompuCALL call party calls the same way as with normal 3WC. For a host computer to use these call party application service functions, the host must create a CompuCALL environment on the switch by establishing a session for which the service profile contains the appropriate call party application service functions. The ACD group of the agent who the host is sending these requests for need not be in the associated environment, but it must be in the CompuCALL environment. However, if the ACD group is in the associated environment, then the host will also receive the CVD messages which are contained in the service profile for that session, in particular dv-Call-Released-U following a dv-Transfer-Party. It is therefore recommend that the service profile for that CompuCALL session contain the appropriate CVD messages with the call party messages and the ACD group of the agent be in the associated environment.

dv-Add-Party, dv-Transfer-Party, dv-Drop-Party and dv-Conference-Party can only be used if the corresponding MDC three-way call (3WC), call transfer (CXR or FXR) and/or call conference features are active for the agent (or the customer group to which the agent belongs) who initiates these messages.

In software release NA012, and later releases, the DMS-100 switch sends two messages to the host regarding 3WC status. The switch sends the two messages in the following cases:

- If the result of 3WC to 2WC state transition is call transfer controlled by controller, DMS-100 will send an SCAI message (dv_Controller_Released_U) with the DN of releasing party.

- If the result of 3WC to 2WC state transition is not call transfer, DMS-100 will send an SCAI message (dv_Noncontroller_Released_U) with the DN of releasing party.

4.7.1 ACD dv-Answer-Call

The answer function is provided by a Class 2 RO, dv-Answer-Call. This capability supports functionality to answer an incoming call on behalf of an ACD agent, Centrex line or a residential line. This capability supports functionality to answer an incoming call on behalf of an ACD agent, Centrex line or a residential line.

If the party which is specified in the dv-Answer-Call message is being offered a call (ringing), then the dv-Answer-Call request will be processed by the switch to set up answering the call on behalf of the user.

A successful invocation of a dv-Answer-Call generates the CVD message dv-Call-Answered-U to be sent to the host. The dv-Call-Answered-U message must be subscribed to and the ACD group or Centrex/RES line must be dv-DN-Associated with the SCAI session in order for the host to receive an event message.

The dv-Answer-Call message is supported on MBS with a headset device or enabling the microphone and speaker on a hands-free device. On MBS the set is updated as appropriate by solidifying the lamp on when the dv-Answer-Call message is received. The dv-Answer-Call message will not be supported on 2500 sets.

4.7.1.1 Answer Call Message Parameters

A brief description of the message and parameter are given below:

dv-Answer-Call The host sends a dv-Answer-Call in order to answer a call on behalf of a specific directory number (DN). A dv-Answer-Call will contain the following mandatory parameter:

answeringParty The AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that the host is making the answer request on behalf of.

4.7.1.2 dv-Answer-Call validations

Upon receiving the dv-Answer-Call message the switch will perform the following validations:

1 Verify that the service version of the current host-switch session is SCAI08 or higher.

2 Verify that the AnswerCall message has been subscribed to by the host session via table SCAISSRV.

3 Verify that the required parameter, answeringParty is present in the message.

4 Verify that the answeringParty specified belongs to same customer group as the current host-switch session.

5 Verify that the Centrex or residential line specified in the answeringParty parameter of the dv-Answer-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.

6 Verify that there is an ACD, Centrex, or residential call being offered to the set (ringing).

7 Verify that the ACD agent, Centrex, or residential line attempting to answer the call is a MBS set with hands free capabilities.

4.7.1.3 Answer Call Responses

The switch responds to the dv-Answer-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - — If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - — When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08 a REJ is sent back to the host.
- Return-Error (RE)
  - — Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be

initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 40 Answer Call Error Conditions

Error	Description
notAllowed	The AnswerCall message is not subscribed to at the link-level. Subscription to messages is done via Table SCAIPROF and Table SCAISSRV. The Centrex or residential line specified in the answeringParty has not subscribed to the ECM sub-option SCAICC. The answeringParty parameter contains an ACD agent or Centrex line offered to a set that is MFT.
missingParameter	The required answeringParty parameter is not included in the message
invalidAnsweringParty	The DN specified in the answeringParty parameter is not in the same customer group as the host-switch session. The answeringParty parameter contains an ACD agent, Centrex, or residential call being offered to a set that is not a MBS.
unknownAnsweringParty	The DN specified in the answeringParty parameter is unknown to the switch.
invalidCallState	The answeringParty parameter contains an ACD, a Centrex, or a residential party that is not in the ringing state.

4.7.1.4 dv-Answer-call ASN.1 encoding

```
ASN.1 Encoding
    dv-Answer-Call                    OPERATION
                                      ARGUMENT
AnswerCallArgument
                        RESULT         AnswerCallResult
                        ERRORS                {invalidCallState,
                                              missingParameter
                                              invalidAnsweringParty,
                                              notAllowed,
                                              unknownAnsweringParty,
                                              resourcesUnavailable}
                ::= 23
    AnswerCallArgument ::=            SEQUENCE
    {answeringParty            [0]    AddressType   OPTIONAL}

    AddressType   ::=                 CHOICE
    {positionID                [0]IMPLICIT          INTEGER,
    dialedDigits               [1]IMPLICIT          IA5STRING,
    stationNumber              [2]IMPLICIT          OCTET STRING,
    q931Address                [3]Q931AddressType}

    Q931AddressType::=                SEQUENCE
    {numberTypeNumberPlan  [0]IMPLICITOCTET STRING OPTIONAL,
presentationScreeningIndicator[1]IMPLICITOCTET STRING OPTIONAL,
    digits                      [2]IMPLICITIA5STRING OPTIONAL}

    AnswerCallResult         ::=SEQUENCE
        {networkCallID         [0] IMPLICIT SEQUENCE
        {networkNodeID         [0]   IMPLICIT INTEGER OPTIONAL,
        localCallID            [1]   IMPLICIT INTEGER OPTIONAL}
                                              OPTIONAL}
    invalidCallState           ERROR::=0
    missingParameter           ERROR::=1
                PARAMETER             MissingParameter

        MissingParameter ::=                  SEQUENCE
        {missingParameterType         [0]IMPLICIT ENUMERATED
        {missingAnsweringParty        (1)}              OPTIONAL}
    invalidAnsweringParty      ERROR::=2
    notAllowed                 ERROR::=3
    unknownAnsweringParty      ERROR::=4
    resourcesUnavailable       ERROR::=5
```

4.7.1.5 dv-Answer-Call: coded example

This example illustrates the case where a host sends a message to answer a call. The position ID of the agent's set is 2000 and the Invoke ID is 11.

INVOKE **DMS <===== HOST**

Table 41 Answer Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0E		INVOKE	Message length is 14 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	17	Operation	Operation value for dv-Answer-Call is decimal 23.
30	06		AnswerCallArgument	Type sequence and 6 bytes long.
A0	04		answeringParty	AnsweringParty is a 4 byte constructor.
80	02	07 D0	positionID	Agent’s Position ID = 2000.
Hex Dump = A1 0E 02 01 0B 02 01 17 30 06 A0 04 80 02 07 D0				

The switch verifies the request and sends back a Return Result indicating the request will be initiated.

RETURN-RESULT DMS ====> HOST

Table 42 RETURN-RESULT for Answer Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0B	InvokeID	Invoke ID is 11.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	17	Operation	its operation is dv-Answer-Call [23]
30	09		AnswerCall Result	the Answer Call Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0B 30 0E 02 01 17 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to answer a call using dv-Answer-Call and there is not a call being offered, the switch will respond with a Return Error message that the call is in an invalid state.

RETURN-ERROR DMS ====> HOST

Table 43 Return Error for Answer Request with an Error Value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		Return Error	Message length is 6 bytes
02	01	0B	InvokeID	Invoke ID is 11.
02	01	00	ErrorType	ErrorType is Invalid-Call-State [0]
Hex Dump = A3 06 02 01 0B 02 01 00				

4.7.2 ACD dv-Hold-Call

This message supports functionality to hold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The hold function is provided by two Class 2 ROs: dv-Hold-Call, dv-Unhold-Call. A Class 5 RO is also provided: dv-Call-Unheld-U. The new messages supports functionality to hold and unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The dv-Hold-Call message is usable on “active, connected” calls, while the dv-Unhold-Call message may only be used to unhold a call held by the dv-Hold-Call message. If a user should hit their DN key on an MBS set, this will effectively “unhold” the call if it is being “held” by the CompuCALL application, upon which time switch will send a dv-Call-Unheld-U message to the host computer. If a 500/2500 (IBN and POTs) set is being “held” by the CompuCALL application, when the user flashes on a 500/2500 (IBN and POTs) set, this will “unhold” the call and a dv-Call-Unheld-U message is sent to the host computer.

After a dv-Hold-Call is sent to a 500/2500 (IBN and POTs) line and silence is heard, SCAI 3WC messages (add, drop, and transfer) messages cannot be used to originate a call.

The party specified in the dv-Hold-Call message can be a residential, Centrex, or ACD position ID. If a dv-Hold-Call message is sent to the ACD position ID, and the user hits their DN key on an MBS set, the party specified in the dv-Call-Unheld-U message is the ACD position ID.

Suppose a party specified in the dv-Hold-Call message is placed on hold in an active call. Next, another dv-Hold-Call message is sent to hold the other party on an active call, so both parties are on hold. A dv-Unhold-Call message is used to unhold one of the parties. A second dv-Unhold-Call message is used to unhold the other party and establishes a talking connection.

If at any time during the "hold" of the line, the user activates a set based "hold", the CompuCALL activated hold will remain activated until deactivated as outlined above, and the set based "hold" will hold the line until deactivated on the set. Thus the functionality of the dv-Hold-Call will not be interchangeable with the set based Hold.

Since the user may subsequently choose to continue the call from the telephone set, it is desirable to inform the user of the call hold - for example, on an MBS a flashing diamond will be displayed next to the line in use.

4.7.2.1Hold parameters

A brief description of the messages and parameters are given below:

dv-Hold-Call The host sends a dv-Hold-Call in order to hold a call on behalf of a specific DN. A dv-Hold-Call will contain the following mandatory parameter:

holdingParty The AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that the host is making the hold request on behalf of.

4.7.2.2 dv-Hold-Call validations

Upon receiving the dv-Hold-Call message the switch will perform the following validations:

1. Verify that the service version of the current host-switch session is SCAI08 or higher.
2. Verify that the HoldCall message has been subscribed to by the host session via table SCAISSRV.
3. Verify that the required parameter, holdingParty is present in the message.
4. Verify that the holdingParty specified belongs to same customer group as the current host-switch session.
5. Verify that the Centrex, or residential line specified in the holdingParty parameter of the dv-Hold-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.
6. Verify that the ACD, Centrex, or residential line specified in the holdingParty is in an active, connected call.

4.7.2.3 Hold Call Responses

The switch responds to the dv-Hold-Call and dv-Unhold-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08, a REJ is sent back to the host.
  - Since the dv-Call-Unheld-U is a Class 5 operation, the host will not respond to the message.
- Return-Error (RE)

Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 44 Hold call error conditions

Error	Description
notAllowed	The HoldCall message is not subscribed to at the link-level. The Centrex or residential line specified in the holdingParty has not subscribed to the ECM sub-option SCAICC The holdingParty parameter contains an ACD agent Centrex line offered to a set that is MFT.
missingParameter	The required holdingParty parameter is not included in the message.
invalidholdingParty	The DN specified in the holdingParty parameter is not in the same customer group as the host-switch session. The holdingParty parameter contains an ACD agent, Centrex, or residential line offered to a set that is not MBS or 500/2500.
unknownHoldingParty	The DN specified in the holdingParty parameter is unknown to the switch.
invalidCallState	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in an active connected call. {dv-Hold-Call} The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in a call held by the dv-Hold-Call. {dv-Unhold-Call}
callNotHoldableByUser	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is involved with an emergency service line.

4.7.2.4 dv-Hold-Call ASN.1 encoding

ASN.1 Encoding

```
dv-Hold-Call              OPERATION
                          ARGUMENT    HoldCallArgument
                          RESULT      HoldCallResult
                          ERRORS      {invalidCallState,
                                      missingParameter,
                                      invalidHoldingParty,
                                      notAllowed,
                                      unknownHoldingParty,
                                      callNotHoldableByUser,

        resourcesUnavailable}
                 ::= 28

HoldCallArgument::=       SEQUENCE
{holdingParty             [0] AddressType       OPTIONAL}

AddressType::=            CHOICE
{positionID               [0] IMPLICIT          INTEGER, (ONLY
CHOICE)
 dialedDigits             [1] IMPLICIT          IA5STRING,
 stationNumber            [2] IMPLICIT          OCTET STRING,
 q931Address              [3] Q931AddressType}

Q931AddressType::=        SEQUENCE
{numberTypeNumberPlan         [0]IMPLICITOCTET STRING OPTIONAL,
 presentationScreeningIndicator [1]IMPLICITOCTET STRING OPTIONAL,
 digits                        [2]IMPLICITIA5STRING OPTIONAL}

HoldCallResult              ::=   SEQUENCE
    {networkCallID          [0]   IMPLICIT SEQUENCE
            {networkNodeID      [0]   IMPLICIT INTEGER       OPTION-
    AL,
             localCallID        [1]   IMPLICIT INTEGER       OPTION-
    AL}
                                                             OPTION-
    AL}

invalidCallState          ERROR::=0
missingParameter          ERROR::=1
                          PARAMETER   MissingParameter

    MissingParameter           ::=   SEQUENCE
     {missingParameterType   [0]    IMPLICIT ENUMERATED
```

```
        {missingHoldingParty       (1)}                      OPTIONAL}

invalidHoldingParty       ERROR::=2
notAllowed                ERROR::=3
unknownHoldingParty       ERROR::=4
callNotHoldableByUser     ERROR::=5
resourcesUnavailable      ERROR::=6
```

4.7.2.5 dv-Hold-Call: coded example

This example illustrates the case where a host sends a message to hold DN 782-3640.

INVOKE DMS <===== HOST

Table 45 Hold Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	16		INVOKE	The message length is 22 bytes.
02	01	0D	InvokeID	Invoke ID is 13.
02	01	1C	Operation	Operation value for dv-Hold-Call is decimal 28.
30	0E		HoldCallArgument	Type sequence and 14 bytes long.
A0	0C		holdingParty	holdingParty is a 12 byte constructor.
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	stationNumber is 919-782-3640
Hex Dump = A1 16 02 01 0D 02 01 1C 30 0E A0 0C 82 0A 39 31 39 37 38 32 33 36 34 30				

The switch verifies the request and sends back a Return Result indicating the request will be initiated.

RETURN-RESULT DMS ====> HOST

Table 46 RETURN-RESULT for Hold Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0D	InvokeID	Invoke ID is 13.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	1C	Operation	its operation is dv-Hold-Call [28]
30	09		HoldCall Result	the HoldCall Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0D 30 0E 02 01 1C 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to hold a call using dv-Hold-Call but the holding party (residential line) has not subscribed to the ECM sub-option SCAICC. A REeturn Error of not allowed is sent to the host computer.

RETURN-ERROR DMS=====>HOST

Table 47 RETURN-ERROR for Hold Request with an error value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0D	InvokeID	Invoke ID is 13.
02	01	02	ErrorType	ErrorType is notAllowed [2]
Hex Dump = A3 06 02 01 0D 02 01 02				

4.7.3 ACD dv-Release-Call

This functionality supports functionality to release an active call on behalf of an ACD agent, Centrex line, or a residential line.

The release function is provided by a Class 2 RO, dv-Release-Call. This functionality supports functionality to release an active call on behalf of an ACD agent, Centrex line, or a residential line.

If the party which is specified in the dv-Release-Call message is on an active call (ringing, talking, dialing, hold, listening to busy or an announcement), then the dv-Release-Call request will be processed by the switch to set up releasing the call on behalf of the user.

When the party specified in the dv-Release-Call message is the terminator of the call and the terminator is receiving physical ringing then the call is not released. On the other hand, when the party specified in the dv-Release-Call is the originator of the call and the originator is receiving audible ringing the call is released.

When the party specified in the dv-Release-Call message is on hold by CompuCALL, the party on hold is released. Whereas in the same scenario on the switch, the party that is on hold can not be released by the controller until after the call is answered.

A successful invocation of a dv-Release-Call generates the CVD message dv-Call-Released-U to be sent to the host just as though the call had been released manually. The dv-Call-Released-U message must be subscribed to and the ACD group or Centrex/RES line must be dv-DN-associated with the SCAI session in order for the host to receive the CVD message. The release reason provided with the dv-Call-Released-u will be the same as if the call had been released manually. (i.e., callCleared for two party calls, callAbandoned for unanswered calls, etc.)

The dv-Release-Call message is supported on MBS and 500/2500 sets. On MBS the set is updated as appropriate by turning the lamp off when the dv-Release-Call message is received. After successfully issuing a dv-Release-Call request on a 500/2500 set, the user receives dial tone. The user is then able to initiate calls before going on hook and returning to the idle state. If the user stays off hook for a certain length of time, then the set will go to lock out treatment.

4.7.3.1 dv-Release-Call validation

Upon receiving the dv-Release-Call message the switch performs the following validations:

1 Verify that the service version of the current host-switch session is SCAI08 or higher.

2 Verify that the ReleaseCall message has been subscribed to by the host session via table SCAISSRV.

3 Verify that the required parameter, releasingParty is present in the message.

4 Verify that the optional parameter, releaseType has been subscribed to by the host session.

5 Verify that the releasingParty specified belongs to same customer group as the current host-switch session.

6 Verify that the Centrex or residential line specified in the releasingParty parameter of the dv-Release-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.

7 Verify that the ACD, Centrex, or residential line specified in the releasingParty is in a state that is considered releasable.

4.7.3.2 Release Call Message Parameters

A brief description of the message and parameters are given below:

dv-Release-Call The host sends a dv-Release-Call in order to release a call on behalf of a specific directory number (DN). A dv-Release-Call will contain the following parameters:

releasingParty The AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that a host is making the release request on behalf of.

releaseType This optional parameter contains an enumerated type which consists of: releaseThisParty, and releaseAllParties. This parameter specifies whether to release this party alone or release all parties in a call. In a normal two-party call, either type will be accepted and the call will simply be released on behalf of the releasingParty. In a three-way call conference scenario, the releaseThisParty type will release on behalf of the controller which may or may not cause the call to be transferred depending on whether the user has call transfer capabilities, while the releaseAllParties type will release all three parties from the three-way call without any type of transfer taking place.

4.7.3.3 Release call responses

The switch responds to the dv-Release-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08, a REJ is sent back to the host.
- Return-Error (RE)
  - Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 48 Release Call Error Conditions

Error	Description
notAllowed	The ReleaseCall message is not subscribed to at the link-level. The Centrex or residential line specified in the releasingParty has not subscribed to the ECM sub-option SCAICC The releasingParty parameter contains an ACD agent or Centrex line offered to a set that is MFT.
missingParameter	The required releasingParty parameter is not included in the message.
invalidParameter	A parameter that is not subscribed to is included in the message.
invalidReleasingParty	The DN specified in the releasingParty parameter is not in the same customer group as the host-switch session. The releasingParty parameter contains an ACD agent, Centrex, or residential line offered to a set that is not MBS nor 2500.

—continued—

Table 48 Release Call Error Conditions

Error	Description
unknownReleasingParty	The DN specified in the releasingParty parameter is unknown to the switch.
invalidCallState	The releasingParty parameter contains an ACD, a Centrex, or a residential party that is not in a releasable state (i.e., the releasingParty is in an idle state).
callNotReleasableByUser	The releasingParty parameter contains a ACD, a Centrex, or a residential party that is involved with an emergency service line.
—end—	

4.7.3.4 dv-Release-Call ASN.1 encoding

```
dv-Release-Call           OPERATION
                          ARGUMENT     ReleaseCallArgument
                          RESULT       ReleaseCallResult
                          ERRORS       {invalidCallState,
                                       missingParameter
                                       invalidParameter
                                       notAllowed,
                                       unknownReleasingParty,
                                       callNotReleasableByUser,

      resourcesUnavailable}

::= 24

ReleaseCallArgument::=    SEQUENCE
{releasingParty           [0] AddressType  OPTIONAL,
 releaseType              [1]  IMPLICIT ENUMERATED
    {releaseThisParty         (1) OPTIONAL,
     releaseAllParties        (2) OPTIONAL}

AddressType::=            CHOICE
{positionID               [0] IMPLICIT        INTEGER, (ONLY
CHOICE)
 dialedDigits             [1] IMPLICIT        IA5STRING,
 stationNumber            [2] IMPLICIT        OCTET STRING,
 q931Address              [3] Q931AddressType}

Q931AddressType::=        SEQUENCE
{numberTypeNumberPlan     [0] IMPLICIT        OCTET STRING
OPTIONAL,
 presentationScreeningIndicator[1]IMPLICIT    OCTET STRING
OPTIONAL,
 digits                   [2] IMPLICIT        IA5STRING
OPTIONAL}
```

```
ReleaseCallResult            ::=    SEQUENCE
     {networkCallID            [0]   IMPLICIT SEQUENCE
     {networkNodeID  [0] IMPLICIT INTEGER OPTIONAL,
     localCallID  [1] IMPLICIT INTEGER OPTIONAL}OPTIONAL,

invalidCallState             ERROR::=0
missingParameter             ERROR::=1
                             PARAMETER   MissingReleaseParm
     MissingReleaseParm          ::=    SEQUENCE
       {missingParameterType  [0]    IMPLICIT ENUMERATED
          {missingReleasingParty      (0)}                         OPTIONAL}

invalidParameter             ERROR::= 2
                             PARAMETER   InvalidReleaseParm
     InvalidReleaseParm          ::=    SEQUENCE
       {invalidParameterType   [0]    IMPLICIT ENUMERATED
          {invalidReleaseType         (0)                          OPTIONAL}
           invalidReleasingParty      (1)                          OPTIONAL}

notAllowed                   ERROR::=3
unknownReleasingParty   ERROR::=4
callNotReleasableBy UserERROR::=5

resourcesUnavailable ERROR::=6
```

4.7.3.5 dv-Release-Call: coded example

This example illustrates the case where an ACD agent is requesting that an incoming call be released. The position ID of the set is 2000 and the Invoke ID is 11.

INVOKE **DMS <===== HOST**

Table 49 Release call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0E		INVOKE	Message length is 14 bytes.
02	01	0C	InvokeID	Invoke ID is 12.
02	01	18	Operation	Operation value for dv-Release-Call is decimal 24.
30	06		ReleaseCallArgument	Type sequence and 6 bytes long.
A0	04		releasingParty	ReleasingParty is a 4 byte constructor.
80	02	07 D0	PositionID	Agent’s Position ID = 2000.
81	01	01	releaseType	releaseThisParty
Hex Dump = A1 0E 02 01 0C 02 01 18 30 06 A0 04 80 02 07 D0 81 01 01				

The switch verifies the request and sends back a Return result indicating the request will be initiated.

RETURN-RESULT DMS ====> HOST

Table 50 RETURN-RESULT for Release call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0C	InvokeID	Invoke ID is 12
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	18	Operation	its operation is dv-Release-Call [24]
30	09		ReleaseCall Result	the Release Call Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365
Hex Dump = A2 13 02 01 0C 30 0E 02 01 18 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to release a call using dv-Release-Call and selects an unknown DN, then the switch must respond with a Return Error with the error value unknown releasing party.

RETURN-ERROR DMS ====> HOST

Table 51 RETURN-ERROR for Release Request unknown Releasing Party

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0C	InvokeID	Invoke ID is 12
02	01	04	ErrorType	ErrorType is UnknownReleasingParty [4]
Hex Dump = A3 06 02 01 0C 02 01 04				

4.7.4 ACD dv-Unhold-Call

This message supports functionality to unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

ASN.1 Encoding

```
dv-Unhold-Call              OPERATION
                            ARGUMENT     UnHoldCallArgument
                            RESULT       UnHoldCallResult
                            ERRORS       {invalidCallState,
                                         missingParameter,
                                         invalidHoldingParty,
                                         notAllowed,
                                         unknownHoldingParty,

        resourcesUnavailable}
::= 29

UnHoldCallArgument::=       SEQUENCE
{unholdingParty             [0] AddressType       OPTIONAL}

AddressType::=              CHOICE
{positionID                 [0] IMPLICIT          INTEGER, (ONLY
CHOICE)
 dialedDigits               [1] IMPLICIT          IA5STRING,
 stationNumber              [2] IMPLICIT          OCTET STRING,
 q931Address                [3] Q931AddressType}

Q931AddressType::=          SEQUENCE
{numberTypeNumberPlan       [0] IMPLICIT          OCTET STRING
OPTIONAL,
 presentationScreeningIndicator[1]IMPLICIT        OCTET STRING
OPTIONAL,
 digits                     [2] IMPLICIT          IA5STRING
OPTIONAL}

UnHoldCallResult           ::=    SEQUENCE
    {networkCallID          [0]   IMPLICIT SEQUENCE
            {networkNodeID      [0]   IMPLICIT INTEGER       OPTION-
    AL,
             localCallID        [1]   IMPLICIT INTEGER       OPTION-
    AL}
                                                             OPTION-
    AL}

invalidCallState           ERROR::=0
missingParameter           ERROR::=1
                           PARAMETER   MissingParameter

    MissingParameter           ::=   SEQUENCE
     {missingParameterType  [0]   IMPLICIT ENUMERATED
        {missingHoldingParty      (1)}                     OPTIONAL}
```

```
invalidHoldingParty      ERROR::=2
notAllowed               ERROR::=3
unknownHoldingParty      ERROR::=4

resourcesUnavailableERROR::=5
```

4.7.4.1 dv-Unhold-Call: coded example

This example illustrates the case where a host sends a message to unhold a call that has been held by the dv-Hold-Call message.

INVOKE DMS <===== HOST

Table 52 UnHold Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	16		INVOKE	Message length is 22 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	1D	Operation	Operation value for dv-Unhold-Call is decimal 29.
30	0E		UnholdCallArgument	Type sequence and 14 bytes long
A0	0C		unholdingParty	unholdingParty is a 12 byte constructor
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	The station number is 919-782-3640
Hex Dump = A116 02 01 0B 02 01 1D 30 0E A0 0C 82 0A 39 31 39 37 38 32 33 36 34 30				

The switch verifies the request and sends back a Return Result indicating the request will be initiated.

RETURN-RESULT DMS ====> HOST

Table 53 RETURN-RESULT for UnHold call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0B	InvokeID	Invoke ID is 11.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	1D	Operation	its operation is dv-UnHold-Call [29]
30	09		unholdCall Result	the unholdCall Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0B 30 0E 02 01 1D 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to unhold a call using dv-Unhold-Call but forgets to include the holdingParty parameter in the message then the switch must respond with a Return Error describing why the request could not be completed.

RETURN-ERROR DMS ====> HOST

Table 54 RETURN-ERROR for UnHold request with an error value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0B	InvokeID	Invoke ID is 11.
02	01	01	ErrorType	ErrorType is missingParameter[1]
Hex Dump = A3 06 02 01 0B 02 01 01				

The hold function is provided by two Class 2 ROs: dv-Hold-Call, dv-Unhold-Call. A Class 5 RO is also provided: dv-Call-Unheld-U. The new messages supports functionality to hold and unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The dv-Hold-Call message will be usable on active, connected calls, while the dv-Unhold-Call message may only be used to unhold a call held by the dv-Hold-Call message. If a user should hit their DN key on an MBS set, this will effectively “unhold” the call if it is being “held” by the CompuCALL application, upon which time switch will send a dv-Call-Unheld-U message to the host computer. If a 500/2500 (IBN and POTs) set is being held by the CompuCALL application, when the user flashes on a 500/2500 (IBN and POTs) set, this will unhold the call and a dv-Call-Unheld-U message is sent to the host computer.

Suppose a party specified in the dv-Hold-Call message is placed on hold in an active call. Next, another dv-Hold-Call message is sent to hold the other party on an active call, so both parties are on hold. A dv-Unhold-Call message is used to unhold one of the parties. A second dv-Unhold-Call message is used to unhold the other party and establishes a talking connection.

If at any time during the hold of the line, the user activates a set based hold, the CompuCALL activated hold will remain activated until deactivated as outlined above, and the set based hold will hold the line until deactivated on the set. Thus the functionality of the dv-Hold-Call will not be interchangeable with the set based hold.

Since the user may subsequently choose to continue the call from the telephone set, it is desirable to inform the user of the call hold - for example, on a Meridian Business Set a flashing diamond will be displayed next to the line in use.

Upon receiving the dv-Unhold-Call message the switch will perform the following validations:

1 Verify that the service version of the current host-switch session is SCAI08 or higher.

2 Verify that the UnholdCall message has been subscribed to by the host session via table SCAISSRV.

3 Verify that the required parameter, holdingParty is present in the message.

4 Verify that the holdingParty specified belongs to the same customer group as the current host-switch session.

5 Verify that the Centrex or residential line specified in the holdingParty parameter of the dv-Unhold-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.

6 Verify that the ACD, Centrex, or residential line specified in the holdingParty is involved in a call held by the dv-Hold-Call message.

4.7.4.2 Unhold parameters

A brief description of the messages and parameters are given below:

dv-Unhold-Call The host sends a dv-Unhold-Call in order to unhold a call on behalf of a specific directory number (DN). A dv-Unhold-Call will contain the following mandatory parameter:

unholdingParty The AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that the host is making the unhold request on behalf of.

4.7.4.3 Unhold call responses

The switch responds to the dv-Hold-Call and dv-Unhold-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ

is sent back to the host. Also if the service version is prior to SCAI08, a REJ is sent back to the host.

— Since the dv-Call-Unheld-U is a Class 5 operation, the host will not respond to the message.

- Return-Error (RE)

— Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 55 Hold call error conditions

Error	Description
notAllowed	The HoldCall message is not subscribed to at the link-level. The Centrex or residential line specified in the holdingParty has not subscribed to the ECM sub-option SCAICC The holdingParty parameter contains an ACD agent Centrex line offered to a set that is MFT.
missingParameter	The required holdingParty parameter is not included in the message.
invalidholdingParty	The DN specified in the holdingParty parameter is not in the same customer group as the host-switch session. The holdingParty parameter contains an ACD agent, Centrex or residential line offered to a set that is not MBS or 500/2500.
unknownHoldingParty	The DN specified in the holdingParty parameter is unknown to the switch.
invalidCallState	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in an active connected call. {dv-Hold-Call}. The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in a call held by the dv-Hold-Call. {dv-Unhold-Call}
callNotHoldableByUser	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is involved with a Emergency Service line.

4.7.5 ACD dv-Call-Unheld-U

If a user should hit their DN key on a MBS set or 2500 set flashes and unhold the call (held by the CompuCALL application), the switch will send an unheld message to the host computer.

4.7.5.1 dv-Call-Unheld-U ASN.1 encoding

```
dv-Call-Unheld-U           OPERATION
                           ARGUMENT     CallUnheldArgument

::= 30

CallUnheldArgument::=      SEQUENCE
    {networkCallID          [0]   IMPLICIT SEQUENCE
    {networkNodeID          [0]   IMPLICIT INTEGER OPTIONAL,
    localCallID            [1]   IMPLICIT INTEGER OPTIONAL}
                              OPTIONAL,
    unheldParty             [1] Address TypeOPTIONAL}


AddressType::=             CHOICE
{positionID                [0] IMPLICIT          INTEGER, (ONLY
CHOICE)
 dialedDigits              [1] IMPLICIT          IA5STRING,
 stationNumber             [2] IMPLICIT          OCTET STRING,
 q931Address               [3] Q931AddressType}

Q931AddressType::=         SEQUENCE
{numberTypeNumberPlan      [0] IMPLICIT          OCTET STRING
OPTIONAL,
 presentationScreeningIndicator[1]IMPLICIT       OCTET STRING
OPTIONAL,
 digits                    [2] IMPLICIT          IA5STRING
OPTIONAL}
```

4.7.5.2 dv-Call-Unheld-U: coded example

This example illustrates when the user hits the DN key to unhold a call which was held by the dv-Hold-Call message.

INVOKE DMS ====> HOST

Table 56 Call Unheld

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1E		INVOKE	Message length is 30 bytes.
02	01	0F	InvokeID	Invoke ID is 15.
02	01	1E	Operation	Operation value for dv-Call-Unheld-U is decimal 30.
30	16		CallUnheldArgument	Type sequence and 22 bytes long.
A0	06		Network Call ID	NetworkCallId is of type constructor and is 6 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	01	02	Local Call ID	Local Call ID is 2.
A1	0C		unheldParty	unheldParty is a 12 byte constructor.
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	The DN of the phone is 919-782-3640
Hex Dump = A11E 02 01 0F 02 01 1E 30 16 A0 06 80 01 01 81 01 02 A1 0C 82 0A 39 31 39 37 38 32 33 36 34 30				

The hold function is provided by two Class 2 ROs: dv-Hold-Call, dv-Unhold-Call. A Class 5 RO is also provided: dv-Call-Unheld-U. The new messages supports functionality to hold and unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The dv-Hold-Call message will be usable on active, connected calls, while the dv-Unhold-Call message may only be used to unhold a call held by the dv-Hold-Call message. If a user should hit their DN key on an MBS set, this will effectively unhold the call if it is being held by the CompuCALL application, upon which time switch will send a dv-Call-Unheld-U message to the host computer. If a 500/2500 (IBN and POTs) set is being held by the CompuCALL application, when the user flashes on a 500/2500 (IBN and POTs) set, this will unhold the call and a dv-Call-Unheld-U message is sent to the host computer.

The party specified in the dv-Hold-Call message can be a residential, Centrex or ACD position ID. If a dv-Hold-Call message is sent to the ACD position ID, and the user hits their DN key on an MBS set, the party specified in the dv-Call-Unheld-U message is the ACD position ID.

It is not necessary to associate a line to get the Call Unheld message.

4.7.5.3 dv-Call-Unheld-U validations

The switch will perform a series of checks before sending the call event message, dv-Call-Unheld-U. The switch will verify:

1. Verify that the CompuCALL Hold message, dv-Hold-Call, placed the party on hold.
2. The Service Version is at least SCAI08
3. The switch-host session subscribes to the call event message.
4. The mandatory parameter (for the ECM application) of the message is present.
5. The DN specified is within the same customer group as the SCAI session that the host is connected to and is a valid ACD agent, residential DN, or Centrex DN.

4.7.5.4 Call-Unheld parameters

A brief description of the messages and parameters are given below:

dv-Call-Unheld-U This message is sent after the successful invocation of dv-Hold-Call and the user releases the call by hitting their DN key or flashing the 500/2500 set. A dv-Call-Unheld-U will contain the following parameters:

Network Call ID This optional parameter is a CompuCALL generated call ID composed of a network node ID (taken from table SCAIGRP) and a local call ID (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session/environment.

unheldParty The mandatory parameter is the AddressType which can be specified as a positionID or stationNumber which had been “unheld”.

Since the dv-Call-Unheld-U is a Class 5 operation, the host will not respond to the message.

4.7.6 ACD dv-Call-Consult-Originated-U

```
ASN.1 Encoding            dv-Call-Consult-Originated-U
                                   OPERATION

ARGUMENT CallConsultOrigArgument

                          ::= 31

CallConsultOrigArgument ::=SEQUENCE

{networkCallID        [0] IMPLICIT SEQUENCE

{networkNodeID        [0] IMPLICIT INTEGEROPTIONAL,

localCallID           [1] IMPLICIT INTEGEROPTIONAL}

                      OPTIONAL,

    deviceID   [1] AddressTypeOPTIONAL,

    consultDN [2] AddressTypeOPTIONAL}


AddressType::= CHOICE
 {positionID          [0]      IMPLICIT  INTEGER,
   dialedDigits       [1]      IMPLICIT  IA5String,
   stationNumber      [2]      IMPLICIT  OCTET STRING,
   q931Address        [3]      Q931AddressType}
```

4.7.6.1 dv-Call-Consult-Originated-U coded example

INVOKE **DMS ====> HOST**

Table 57 dv-Call-Consult-Originated-U

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	2E		INVOKE	the message length is 46 bytes long
02	02	03 EE	InvokeID	it invokeID is 1006
02	01	1F	Operation	its operation value is dv-Call-Consult-Originated-U [31] decimal
30	25		Argument	CallConsultOrigArgument is of type sequence and is 37 bytes long decimal
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
A1	0C		deviceID	deviceID is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 34 30 30	station Number	the DN of the phone is 613-621-2400
A2	0D		consultDN	consultDN is of type constructor and is 13 bytes long
81	0B	31 36 31 33 36 32 31 32 32 30 30	dialed digits	the DN of the phone is 1-613-621-2200
Hex Dump = A1 2E 02 02 03 EE 02 01 1F 30 25 A0 06 80 01 01 01 81 01 02 A1 0C 82 0A 36 31 33 36 32 31 32 34 30 30 A2 0D 81 0B 31 36 31 33 36 32 31 32 32 30 30				

These manual 3WC/CXR interaction events are reported via these messages to the host computer monitoring the controller in both intra-office three party and inter-office three party calls. Also, conference and transfer events are reported to the host computer monitoring the terminator of the second leg of the call, only under intra-office three party calls. No messages are reported for the originator of the first leg of the three-way call under any circumstances, except if the originator of the first leg of the call is also the controller of the three party call, in which case messages belonging to the controller is sent.

4.7.6.2 dv-Call-Consult-Originated-U validations

The switch performs a series of checks before sending the 3WC/CXR call event message. The switch will verify:

1. The switch-host session subscribes to the call event message.
2. The mandatory parameter (for the ECM application) of the message is present.
3. The associated device address specified is within the same customer group as the SCAI session that the host is connected to and is a valid residential DN, Centrex DN, or ACD DN.
4. The associated device address of the residential DN, Centrex DN, or ACD DN has subscribed to the call event message.
5. The Service Version is at least SCAI09 (The dv-Application-Logon message contains a parameter, service version, which uniquely identifies the protocol version stream of messages which will be sent during the application logged on session).

The switch sends the dv-Call-Consult-Originated-U message for the 3WC/CXR controller when the 3WC/CXR controller activates (hits the 3WC/CXR key) a consult call to a third party and all digits have been collected. A consult call means that a line is involved in an active call and requests to put the original call on hold, and originate a consult call.

The dv-Call-Consult-Originated-U message will be sent in both intra and inter switch consult leg call scenarios.

The 3WC/CXR controller must be associated in order to receive this incoming call event message. Association to a line is done from the host computer by sending a DN-associate message containing the DN of the line.

A brief description of the parameters of the dv-Consult-Originated-U is given below:

4.7.6.3 dv-Call-Consult-Originated-U parameters

The following is a list of parameters:

networkCallID CompuCALL generated ID composed of a network node ID and a local call ID. This is a new call ID if this is the first instance of the call within the session environment. Otherwise, the call ID already assigned to this call will be used.

deviceID This parameter identifies the address of the 3WC/CXR controller in station number format. The 3WC/CXR controller is the party that has 3WC/CXR assigned.

consultDN This parameter identifies the consulted party in dialed digits format. The reason dialed digits is to make the host computer aware what digits the controller dialed to originate the call.

4.7.7 ACD dv-Call-Conferenced-U

ASN.1 Encoding

```
dv-Call-Conferenced-UOPERATION
                    ARGUMENT    CallConferencedArgument
::= 32
CallConferencedArgument ::=    SEQUENCE
   {networkCallID     [0] IMPLICIT SEQUENCE
   {networkNodeID     [0] IMPLICIT INTEGEROPTIONAL,
   localCallID        [1] IMPLICIT INTEGEROPTIONAL}
                              OPTIONAL,
   controllerDN      [1] AddressTypeOPTIONAL,
   consultDN         [2] AddressTypeOPTIONAL,
   firstLegDN        [3] AddressTypeOPTIONAL}

AddressType::= CHOICE
{positionID          [0]     IMPLICIT  INTEGER,
  dialedDigits       [1]     IMPLICIT  IA5String,
  stationNumber      [2]     IMPLICIT  OCTET STRING,
  q931Address        [3]     Q931AddressType}
```

4.7.7.1 dv-Call-Conferenced-U coded example

INVOKE **DMS ====> HOST**

Table 58 dv-Call-Conferenced-U

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	3B		INVOKE	the message length is 59 bytes decimal
02	02	03 EE	InvokeID	it invokeID is 1006
02	01	20	Operation	its operation value is dv-Call-Conferenced [32] decimal
30	32		Argument	CallConferencedArgument is of type sequence and is 50 bytes long decimal
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
A1	OC		controllerDN	deviceID is of type constructor and is 12 bytes long
82	OA	36 31 33 36 32 31 32 34 30 30	stationNumber	the DN of the phone is 613-621-2400
A2	OC		consultDN	consultDN is of type constructor and is 12 bytes long
82	OA	36 31 33 36 32 31 32 32 30 30	stationNumber	the DN of the phone is 613-621-2200
A3	OC		firstLegDN	firstLegDN is of type constructor and is 12 bytes long
82	OA	36 31 33 36 32 31 32 33 30 30	stationNumber	the DN of the phone is 613-621-2300
Hex Dump = A1 3B 02 02 03 EE 02 01 20 30 32 AO 06 80 01 01 81 01 02 A1 OC 82 OA 36 31 33 36 32 31 32 34 30 30 A2 0C 82 0A 36 31 33 36 32 31 32 32 30 30 A3 0C 82 0A 36 31 33 36 32 31 32 33 30 30				

The switch sends the dv-Call-Conferenced-U message when the 3WC/CXR controller conferences a consulted party with the originator of the first leg of a call. The controller requests that a conference call is established by hitting the 3WC/CXR key and all three parties are connected to each other resulting in a three-party conference call.

The dv-Call-Conferenced-U message will be sent for the 3WC/CXR controller, if the controller is associated and this message is sent for the controller for inter-switched and intra-switched consult leg calls.

The dv-Call-Conferenced-U message will also be sent for the consulted DN, if the consult leg call is intra-switched and the consult DN in the consult leg cal is associated. The message will not contain the firstLegDN due to privacy considerations.

4.7.7.2 dv-Call-Conferenced-U parameters

A brief description of the parameters of the dv-Call-Conferenced-U is given below:

networkCallID CompuCALL generated call ID composed of a network node ID (taken from table SCAIGRP) and a local call Id (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session environment; otherwise the call ID already assigned to this call will be used.

controllerDN This parameter identifies the address of the 3WC/CXR controller in station number format. The 3WC/CXR controller is the party that has 3WC/CXR assigned.

consultDN This parameter identifies the consulted party in station number format for intra-switch consult leg call or in dialed digits format for inter-switch consult leg call.

firstLegDN This parameter identifies the party who is connected to the controller. The firstLegDN will be in station number format if the first-leg call is intra-switch or the controller is the terminating party on the first leg call. The firstLegDN will be in dialed digits format if the controller is the originator of the first leg of an inter-switch call.

Note: This parameter is only mandatory for the dv-Call-Conferenced message intended for the host of the **controller**. This parameter is not included in the message intended for the host of the consult party due to privacy considerations.

4.7.8 ACD dv-Call-Transferred-U ASN.1 encoding

ASN.1 Encoding

```
dv-Call-Transferred-UOPERATION
                        ARGUMENT     CallTransferredArgument
:= 33
 CallTransferredArgument :=     SEQUENCE
    {networkCallID              [0] IMPLICIT SEQUENCE
    {networkNodeID             [0] IMPLICIT INTEGEROPTIONAL,
    localCallID                [1] IMPLICIT INTEGEROPTIONAL}
                               OPTIONAL,

    deviceID          [1] AddressTypeOPTIONAL,

    otherPtyDN        [2] AddressTypeOPTIONAL}

AddressType:= CHOICE
{positionID           [0]     IMPLICIT  INTEGER,
   dialedDigits       [1]     IMPLICIT  IA5String,
   stationNumber      [2]     IMPLICIT  OCTET STRING,
   q931Address        [3]     Q931AddressType}
```

4.7.8.1 dv-Call-Transferred-U coded example

INVOKE DMS ====> HOST

Table 59 dv-Call-Transferred-U Invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	2D		INVOKE	the message length is 45 bytes decimal
02	02	03 EE	InvokeID	its Invoke ID is 1006
02	01	21	Operation	its operation value is dv-Call-Transferred-U [33] decimal
30	24		Argument	CallTransferredArg is of type sequence and is 36 bytes long decimal
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
A1	0C		deviceID	deviceID is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 32 30 30	stationNumber	the DN of the phone is 613-621-2200
A2	0C		.... otherPtyDN	otherPtyDN is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 33 30 30	stationNumber	the DN of the phone is 613-621-2300
Hex dump = A1 2D02 02 03 EE 02 01 21 30 24 A0 06 80 01 01 81 01 02 A1 0C 82 0A 36 31 33 36 32 31 32 32 30 30 A2 0C 82 0A 36 31 33 36 32 31 32 33 30 30				

The switch sends the dv-Call-Transferred-U message for the consult party when a controller transfers an active call and if the controller is in the same switch. This event will be sent when the controller is dropped and the call is transferred.

The terminator of the consult leg must be associated in order to receive this incoming call event message. Association to a line is done from the host computer by sending a DN-associate message containing the DN of the line.

4.7.8.2 dv-Call-Transferred-U parameters

A brief description of the parameters of the dv-Call-Transferred-U is given below:

networkCallID CompuCALL generated call ID composed of a network node ID (taken from table SCAIGRP) and a local call Id (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session environment; otherwise the call ID already assigned to this call will be used.

deviceID This parameter identifies the address of the party to which the call was transferred. The deviceID will be in station number format if the consult party is a residential or Centrex line. The deviceID will be in position ID format if the consult party is an ACD agent.

otherPtyDN This parameter identifies the party who was the originator of the first leg of the call and was eventually transferred, in station number format.

Note: This parameter will not be included in the message if the controller transfers the consult party to a conference call. (The display of the consult party's set will show conference /call transfer rather than the first leg party DN.)

4.7.9 ACD dv-Add-Party (operation value "C" hex)

This RO is a request to set up a consult call. The type of consult call to be set up is defined by the AddPartyType RO parameter. This message is valid for Meridian ACD CompuCALL options only. The host sends this RO of Operation Class 2 to the switch when a consult party is to be added to a call.

4.7.9.1dv-Add-Party ASN.1 encoding

ASN.1 Encoding

```
dv-Add-Party                          OPERATION
ARGUMENT                              AddPartyArgument
RESULT                                AddPartyResult
ERRORS              {         invalid-Call-State,
                                    missing-Parameter,
                                    invalid-Parameter
                                    not-Allowed
                                    agent-not-Logged-in
                                    resources-Unavailable}
::= 12

AddPartyArgument            ::=     SEQUENCE
        {   addPartyType            [0] IMPLICIT ENUMERATED
        {   addConsultParty         (0)     OPTIONAL,
            addConsultforConf       (1)     OPTIONAL}
                                            OPTIONAL,
            origAddress             [2] AddressTypeOPTIONAL,
            destAddress             [3] AddressTypeOPTIONAL}
            calledAbtNo             [4] AddressTypeOPTIONAL}
AddressType                 :: =    CHOICE
        {   positionID              [0] IMPLICIT INTEGER,
            dialedDigits            [1] IMPLICIT IA5STRING,
            stationNumber           [2] IMPLICIT OCTET STRING,
            q931Address             [3] Q931AddressType}

Q931AddressType             :: =    SEQUENCE
    {   numberTypeNumberPlan        [0] OCTET STRINGOPTIONAL,
        presentationScreeningIndicator[1] OCTET STRINGOPTIONAL,
        digits                      [2] IA5 STRINGOPTIONAL}


AddPartyResult              ::=  SEQUENCE
        {networkCallID              [0] IMPLICIT SEQUENCE
        {   networkNodeID           [0] IMPLICIT INTEGER OPTIONAL,
            localCallID             [1] IMPLICIT INTEGER OPTIONAL}
                                                        OPTIONAL}

invalid-Call-State                              ERROR::= 0
missing-Parameter           ERROR               ERROR::= 1
                            PARAMETER           MissingParameter

MissingParameter            ::=     SEQUENCE
        {   missingParameterType            [0] IMPLICIT ENUMERATED
        {   missingAddPartyType             (0),
            missingOrigAddress              (1),
            missingDestAddress              (2)}    OPTIONAL}

--  only the first (i.e., one) missing parameter
--  encountered is reported in this message
```

```
invalid-Parameter  ERROR                      ERROR::= 2
                   PARAMETER                  InvalidParameter


InvalidParameter         ::=      SEQUENCE
        {   invalidParameterType          [0] IMPLICIT ENUMERATED
        {   invalidAddPartyType           (0),
            invalidOrigAddress            (2),
            invalidDestAddress            (3)}     OPTIONAL}

--   only the first (i.e., one) invalid parameter
--   encountered is reported in this message

not-Allowed              ERROR         ::= 3
agent-Not-Logged-In      ERROR         ::= 5
resources-Unavailable    ERROR         ::= 6
```

4.7.9.2 ACD dv-Add-Party: coded example

The host requests the switch to set up a consult call with no intent to conference with the second party on behalf of the ACD agent whose PositionID = 33 decimal (21 Hex) to a consult party (third party) whose destination Address (DialedDigits) = 26050 by sending dv-Add-Party with invokeID = 3. As a result, the second party is put on hold. The switch verifies the request and sends back Return Result that the consult call will be initiated. If the switch receives the dv-Add-Party message with a missing AddPartyType, it sends a Return Error message that the AddPartyType parameter is missing. Similarly, if it receives the message when the agent is not logged in, it sends a Return Error indicating the agent is not logged in.

INVOKE DMS ===== > HOST

Table 60 dv-Add-Party Invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	19		INVOKE	this message is 25 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	0C	Operation	it operation value is dv-Add-Party[12] decimal
30	11		Argument	the AddPartyArgument is of type sequence and is 17 bytes long decimal
80	01	00	AddPartyType	the AddPartyType is AddConsultParty
A2	03		OrigAddress	the OrigAddress is a constructor
80	01	21	PositionId	the agent’s PositionId is 33 decimal
A3	07		DestAddress	this is a constructor
81	05	32 36 30 35 30	DialedDigits	the DialedDigits are 26050
Hex Dump = A1 19 02 01 03 02 01 0C 30 11 80 01 00 A2 03 80 01 21 A3 07 81 05 32 36 30 35 30				

RETURN-RESULT DMS ===== > HOST

Table 61 dv-Add-Party Return-Result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
30	0E		ReturnErrorSequence	this is a sequence 14 bytes long decimal
02	01	0C	Operation	its operation is dv-Add-Party[12] decimal
30	09		AddPartyResult	the AddPartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this of type sequence 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 03 30 0E 02 01 0C 30 09 A0 07 80 01 01 81 02 05 55				

RETURN-ERROR DMS ===== > HOST

Table 62 dv-Add-Party Return-Error example 1

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	30	InvokeID	the InvokeID is 3 decimal
02	01	01	ErrorType	the ErrorType is MissingParameter[1]
30	03		ReturnErrorSequence	MissingParameter is a sequence 3 bytes
80	01	00	ErrorParameter	the MissingParameter missing AddPartyType(0)
Hex Dump = A3 0B 02 01 30 02 01 01 30 03 80 01 00				

RETURN-ERROR DMS ===== > HOST

Table 63 dv-Add-Party Return-Error example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	05	ErrorValue	the ErrorValue is Agent-Not-Logged-In[5]
Hex Dump = A3 06 02 01 03 02 01 05				

The host sends dv-Add-Party message to initiate a consult call on behalf of the controller. The controller does not have to be in the associated environment with which the host has an established session.

4.7.9.3 ACD dv-Add-Party with display called about number: coded example

The host requests the switch to set up a consult call with no intent to conference with the second party on behalf of the ACD agent with PositionID 21 Hex. The party to consult with has adestination address of 855-2222. The host also sends the Called About number (506-308-8957), received from the VRU, in the DV-ADD-PARTY message. The switch verifies the request and if valid, responds with a call ID in the RETURN-RESULT.

With the Logon version less than 11 and the Called About number subscribed to in table SCAISSRV for that profile, the TPCC dv-add-pty message responds to the host with a RETURN-REJECT message indicating UNRECOGNIZED_PARAMETER.

INVOKE DMS ===== > HOST

Table 64 dv-Add-Party with display called about number Invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	29		INVOKE	this message is 41 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	0C	Operation	the operation value is dv-Add-Party[12] decimal
30	21		Argument	the AddPartyArgument is of type sequence and is 33 bytes long decimal
80	01	00	AddPartyType	the AddPartyType is AddConsultParty
A2	03		OrigAddress	the OrigAddress is a constructor 12 bytes long
82	01	21	PositionId	the agent’s PositionId is 33 decimal
A3	09		DestAddress	this is a constructor 9 bytes long decimal
81	07	38 35 35 32 32 32 32	DialedDigits	the DialedDigits are 26050
A4	0C		CalledAbtNo	this is a constructor 12 bytes long decimal
81	07	35 30 36 33 30 38 38 39 35 37	DialedDigits	the DialedDigits are 26050
Hex Dump = A1 29 02 01 03 02 01 0C 30 21 80 01 00 A2 03 82 0A 21 A3 09 81 07 38 35 35 32 32 32 32 A4 0C 81 0A 35 30 36 33 30 38 38 39 35 37				

RETURN-REJECT DMS ===== > HOST

Table 65 dv-Add-Party with display called about number return reject

Hex Dump			Information Element	Comments
Tag	Length	Content		
A4	06		RETURN-REJECT	this message is 6 bytes decimal
02	01	10	InvokeID	the InvokeID is 16 decimal
812	01	08	Invoke problem	the invoke problem is Unrecognized_Parameter [8]
Hex Dump = A4 06 02 01 10 81 01 08				

4.7.9.4 dv-Add-Party parameters

A brief description of the parameters follows:

AddPartyType indicates the type of conference or consult call to be set up. The values are AddConsultParty and AddConsultforConf. Both values request to put the existing call on hold and initiate a call to a consult party. The former value specifies that this call may not be subsequently conferenced, and therefore, conference resources are not needed. The latter value, however, requests resources to be reserved for conferences.

OrigAddress identifies the originator of the consult call to be set up. It is a choice parameter. The only choice for OrigAddress is PositionID. This parameter uniquely identifies an agent position (not the actual agent) for the customer in a switch.

DestAddress identifies the terminator (other-end) of the call to be set up. It is also a choice parameter. The only choice for DestAddress is DialedDigits. This is a string of IA5-encoded digits that correspond to the dialed digits, had the dialing been performed manually. The digits are translated and routed according to the business group (currently equivalent to DMS customer group) and Network Class of Service (NCOS) datafill set up for the originator of the call.

CalledAbtNo identifies the called about number which the user enters into the voice response unit (VRU). This parameter is a string of IA5-encoded digits

Note: All these parameters must be included by the host in the dv-Add-Party message.

Upon receiving a dv-Add-Party message, the switch ensures that it is a valid request. If translations are valid, the switch returns a Return Result message to the host indicating that the consult call request is valid and that it will be initiated. The Return Result message always contains the NetworkCallID of the call. At this point, there will be no more CompuCALL signaling done with this request. Should anything cause the request to fail (no software resources), inbound signaling will be given to the agent.

4.7.9.5 dv-Add-Party error conditions

Any error conditions cause the call to remain in its current state and a Return Error message sent to the host with the error cause (indicated below in parentheses). Possible error conditions are:

1 The message is not subscribed to (Not-Allowed).

2 Any of the required parameters missing (Missing-Parameter with the missing parameter included).

3 The position ID of the OrigAddress does not correspond to an ACD agent. As well, the ACD agent must belong to the same customer group as the CompuCALL session over which dv-Add-Party is sent. (Invalid-Parameter with value "InvalidOrigAddress");

4 The destination address contains something other than dial digits (Invalid-Parameter with value "InvalidDestAddress").

5 Agent not logged in (Agent-not-logged-in).

6 The call did not reach the state capable of adding a consult party (Invalid-Call-State).

7 The AddPartyType is not a valid type (Invalid-Parameter with the value "InvalidAddPartyType").

8 Conference resources unavailable for adding consult party for conference purpose (resource unavailable).

This list is presented in order of priority since only the first error condition (including the missing/invalid parameter) encountered by the switch is included in the Return Error message.

> ***Note:*** The digits dialed must be the same digits an ACD agent would have dialed to reach the same destination and will be translated the same way as they would be had the ACD agent dialed the digits. Valid translations are anything that translate to a route. Entities that do not qualify as a route are:
>
> a. translations that require more digits
>
> b. translations that require a database query (e.g., TCAP)
>
> c. translations that correspond to a feature access code

The Return Error message may also be returned by the switch application if the consult call setup fails. Note that a Return Result message does not guarantee success of the dv-Add-Party consult call but indicating that the initial verification has passed. Call set-up attempt failure might still occur due to system resource unavailability (for example).

The SCAI interface does not consider an invalid called about number received from the host as an error condition. Instead, the error message “INVALID CLDABT #” displays on the ACD agent’s set that receives the transfer.

4.7.10 ACD dv-Conference-Party (operation value "F" hex)

This RO is a request sent by the host to the switch to set up a conference call. It involves the controller, original call party, and the consult party on the same call. The host send this RO of operation class 2 to the Meridian ACD CompuCALL application when a consult party is to be added to a call. This message is valid for Meridian ACD CompuCALL options only.

4.7.10.1 dv-Conference-Party ASN.1 encoding

```
dv-Conference-Party                         OPERATION
ARGUMENT                                    ConferencePartyArgument
RESULT                                      ConferencePartyResult
    ERRORS                          {       invalid-Call-State,
                                            missing-Parameter,
                                            invalid-Parameter
                                            not-Allowed
                                            agent-Not-Logged-on
                                            resources-Unavailable}
                            ::= 15

ConferencePartyArgument     ::=     SEQUENCE
    {origAddress            [2] AddressTypeOPTIONAL}

    AddressType             :: =    CHOICE
    {   positionID                  [0] IMPLICIT INTEGER,
        dialedDigits                [1] IMPLICIT IA5STRING,
        stationNumber               [2] IMPLICIT OCTET STRING,
    q931Address                     [3] Q931AddressType}

    Q931AddressType         :: =    SEQUENCE
    {numberTypeNumberPlan           [0] OCTET STRINGOPTIONAL,
    presentationScreeningIndicator  [1] OCTET STRINGOPTIONAL,
    digits                          [2] IA5 STRINGOPTIONAL}


    ConferencePartyResult   ::=     SEQUENCE
    {   networkCallID               [0]  IMPLICIT SEQUENCE
    {   networkNodeID               [0] IMPLICIT INTEGER OPTIONAL,
        localCallID                 [1] IMPLICIT INTEGER OPTIONAL}
    OPTIONAL}

    invalid-Call-State      ERROR::= 0

    missing-Parameter       ERROR::= 1
    PARAMETER MissingParameter

    MissingParameter        ::=     SEQUENCE
    {   missingParameterType        [0] IMPLICIT ENUMERATED
    {   missingOrigAddress                  (1)}    OPTIONAL}

    invalid-Parameter       ERROR::= 2
    PARAMETERInvalidParameter

    InvalidParameter        ::=     SEQUENCE
    {   invalidParameterType        [0] IMPLICIT ENUMERATED
{    invalidOrigAddress(2)}OPTIONAL}

not-Allowed              ERROR::= 3

agent-Not-Logged-In      ERROR::= 5

resources-Unavailable    ERROR::= 6
```

4.7.10.2 ACD dv-Conference-Party: coded example

The host requests the switch to set up a conference call for the ACD agent whose PositionID = 33 decimal (21 Hex). The switch verifies the request and returns a Return Result indicating that a conference will be initiated. If the switch receives the dv-Conference-Party message with a missing OrigAddress, it sends a Return Error message that the OrigAddress parameter is missing. Similarly, if it receives the message when the agent is not logged in, it sends a Return Error indicating this.

INVOKE DMS < ====== HOST

Table 66 dv-Conference-Party Invoke

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0D		INVOKE	this message is 13 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	0F	Operation	it operation value is dv-Conference-Party[15] decimal
30	05		Argument	the ConferencePartyArgument is of type sequence and is 5 bytes long decimal
A2	03		OrigAddress	the OrigAddress is a constructor
80	01	21	PositionId	the agent’s PositionId is 33 decimal
Hex Dump = A1 0D 02 01 03 02 01 0F 30 05 A2 03 80 01 21				

RETURN-RESULT DMS ===== > HOST

Table 67 dv-Conference-Party Return Result

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
30	0E		ReturnResult sequence	this is a sequence 14 bytes long decimal
02	01	0E	Operation	its operation is dv-Conference-Party [15] decimal
30	09		ConferenceParty	the ConferencePartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this is of type sequence and is 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A213 02 01 03 30 0E 02 01 0E 30 09 A0 07 80 01 01 81 02 05 55				

RETURN-ERROR DMS ===== > HOST

Table 68 dv-Conference-Party Return Error example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	01	ErrorType	the ErrorType is MissingParamter[1]
30	03		ReturnErrorSequence	this is a sequence 3 bytes long decimal
80	01	01	ErrorParameter	the MissingParameter is MissingOrigAddress (1)
Hex Dump = A3 0B 02 01 03 02 01 01 30 03 80 01 01				

RETURN-ERROR DMS ===== > HOST

Table 69 dv-Conference-Party Return Error example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	05	ErrorValue	the ErrorValue is Agent-Not-Logged-In[5]
Hex Dump = A3 06 02 01 03 02 01 05				

The host sends dv-Conference-Party message to initiate a conference call on behalf of the controller. The controller needs not be in the associated environment with which the host has an established session. A brief description of the parameter of the dv-Conference-Party invoke follows.

OrigAddress identifies the controller. It is a choice parameter. The choice for OrigAddress is PositionID. This parameter uniquely identifies an agent position (not the actual agent) for the customer in a switch.

The above parameter must be included in the dv-Conference-Party message. Upon receiving a dv-Conference-Party message, the switch ensures that it is a valid request.

If none of the errors below occurs, the switch will return a Return Result message to the host indicating that the dv-Conference-Party is valid and will be initiated. The Return Result message will always contain the NetworkCallID. At this point, there will be no more CompuCALL signaling done in connection with this request. Should anything cause the request to fail, inbound signaling will be given to the agent.

4.7.10.3 dv-Conference-Party error conditions

Any error conditions will result in the call remaining in its current state and a Return Error message being returned to the host with an appropriate error cause value (shown below in parentheses). Possible error conditions are:

1 The message is not subscribed to (Not-Allowed).

2 The PositionID of the OrigAddress does not correspond to an ACD agent who is the controller (first party) of a call. As well, the ACD agent must belong to the same customer group as the CompuCALL session over which dv-Conference-Party is sent. (Invalid-Parameter with value "InvalidOrigAddress").

3 Agent not logged in (Agent-not-logged-in).

4. The call did not reach the state capable of conferencing the parties (Invalid-Call-State).
5. Any of the required parameters missing (Missing-Parameter with the missing parameter included)
6. Conference resources unavailable (Resource-Unavailable).

This list is presented in the order of priority since only the first error condition (including the first missing/invalid parameter) encountered by the switch is included in the Return Error message.

The Return Error message may also be returned by the application if the initial setup fails. Note that a Return Result message does not guarantee success of the dv-Conference-Party consult but indicating that the initial verification has passed. Failure might still occur due to system resource unavailability, for example.

4.7.11 ACD dv-Drop-Party (operation value "E" hex)

This RO is a request by the host to the switch to drop a party involved in a consult call. The type of drop to be done is defined by the DropPartyType parameter. This message is valid for Meridian ACD CompuCALL options only. The host sends this RO of Operation Class 2 to the switch when a consult party of a call is to be released and the controller reconnected to the held party.

4.7.11.1 dv-drop-party ASN.1 encoding

```
dv-Drop-Party               OPERATION
                            ARGUMENT        DropPartyArgument
                            RESULT          DropPartyResult
                            ERRORS          {  invalid-Call-State,
                                               missing-Parameter,
                                               invalid-Parameter
                                               not-Allowed
                                               agent-not-Logged-in}
::= 14

DropPartyArgument           ::=  SEQUENCE
 {  dropPartyType           [0] IMPLICIT ENUMERATED
      {  dropConsultParty        (0)}                   OPTIONAL,
    origAddress             [2] AddressType             OPTIONAL}

AddressType                 :: =    CHOICE
 {  positionID              [0] IMPLICIT INTEGER,
    dialedDigits            [1] IMPLICIT IA5STRING,
    stationNumber           [2] IMPLICIT OCTET STRING,
    q931Address             [3] Q931AddressType}
Q931AddressType             :: =    SEQUENCE
 {  numberTypeNumberPlan            [0] OCTET STRING    OPTIONAL,
    presentationScreeningIndicator  [1] OCTET STRING       OPTIONAL,
    digits                          [2] IA5 STRING         OPTIONAL}

DropPartyResult            ::=      SEQUENCE
 {  networkCallID          [0] IMPLICIT SEQUENCE
      {  networkNodeID         [0] IMPLICIT INTEGER        OPTIONAL,
         localCallID           [1] IMPLICIT INTEGER        OPTIONAL},
                                                           OPTIONAL}

invalid-Call-State         ERROR              ::= 0

missing-Parameter          ERROR              ::= 1
                           PARAMETER          MissingParameter

MissingParameter           ::=      SEQUENCE
 {  missingParameterType   [0] IMPLICIT ENUMERATED
      {  missingDropPartyType  (0),
         missingOrigAddress    (1} OPTIONAL}

                           -- only the first (i.e., one) invalid parameter
                           -- encountered is reported in this message

invalid-Parameter          ERROR              ::= 2
                           PARAMETER          InvalidParameter

InvalidParameter           ::=      SEQUENCE
 {  invalidParameterType   [0] IMPLICIT ENUMERATED
      {  invalidDropPartyType  (0),
         invalidOrigAddress    (2)}                        OPTIONAL}

                           --  only the first (i.e., one) invalid parameter
                           --  encountered is reported in this message

not-Allowed                ERROR              ::= 3

agent-Not-Logged-In        ERROR              ::= 5
```

4.7.11.2 ACD dv-Drop-Party: coded example

After the consult call is set up, the host may drop the consult party in the call and reestablish it between the controller and the held party (second party). The host requests to drop the consult party for the controller whose PositionID = 33 by sending dv-Drop-Party with invokeID = 8. After verifying the request, the switch sends Return Result that the drop request will be initiated. If an error occurs (e.g., missing parameter), the switch sends Return Error with error description.

INVOKE DMS < ====== HOST

Table 70 dv-Drop-Party Invoke

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	10		INVOKE	this message is 16 bytes decimal
02	01	08	InvokeID	the InvokeID is 8 decimal
02	01	0E	Operation	it operation value is dv-Drop-Party[14] decimal
30	08		Argument	the DropPartyArgument is DropConsultLeg
80	01	00	DropPartyType	the DropPartyType is of type sequence and is 8 bytes long decimal
A2	03		OrigAddress	the OrigAddress is AddressType (CHOICE)
80	01	21	PositionID	the agent’s positionID is 33 decimal
Hex Dump = A1 10 02 01 08 02 01 0E 30 08 80 01 00 A2 03 80 01 21				

RETURN-RESULT DMS ===== > HOST

Table 71 dv-Drop-Party Return-Result

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	08	InvokeID	the InvokeID is 8 decimal
30	0E		ReturnResultSequence	this is a sequence 14 bytes long decimal
02	01	0E	Operation	its operation is dv-Drop-Party[14] decimal
30	09		DropPartyResult	the TransferPartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this is of type sequence and is 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 08 30 0E 02 01 0E 30 09 A0 07 80 01 01 81 02 05 55				

RETURN-ERROR DMS ===== > HOST

Table 72 dv-Drop-Party Return-Error example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	08	InvokeID	the InvokeID is 8 decimal
02	01	01	ErrorType	the ErrorType is MissingParamter[1]
30	03		ReturnErrorSequence	this is a sequence 3 bytes long decimal
80	01	00	ErrorParameter	the MissingParameter is MissingDropPartyType (0)
Hex Dump = A3 0B 02 01 08 02 01 01 30 03 80 01 00				

RETURN-ERROR DMS ===== > HOST

Table 73 dv-Drop-Party Return-Error example 2

Hex Dump			Information Element	Comments
Tag	Len	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	08	InvokeID	the InvokeID is 8 decimal
02	01	05	ErrorValue	the ErrorValue is Agent-Not-Logged-In[5]
Hex Dump = A3 06 02 01 08 02 01 05				

4.7.11.3 dv-Drop-Party parameters

The host sends dv-Drop-Party message to drop the consult party on behalf of the controller. The controller need not be in the associated environment with which the host has an established session. A brief description of the parameters of the dv-Drop-Party invoke follows.

DropPartyType indicates who is to be dropped and how. The only valid DropPartyType is DropConsultParty, a request to drop the consult party and reestablish the connection between the held original call party and controller.

OrigAddress identifies the controller. It is a choice parameter. The only choice for OrigAddress is PositionID. This parameter uniquely identifies an agent position (not the actual agent) for the customer in a given switch.

Note: Both parameters must be included in the dv-Drop-Party message.

Upon receiving a dv-Drop-Party message, the switch ensures that it is a valid request. If none of the errors below occurs, then the switch will return a Return Result to the host indicating that the dv-Drop-Party request is valid and will be initiated. The Return Result will always contain the Network CallID of the active call. At this point, there will be no more CompuCALL signaling done with this request. Should anything cause the request to fail, inbound signaling will be given to the agent.

4.7.11.4 dv-Drop-Party error conditions

Any error conditions result in the call remaining in its current state and a Return Error being returned to the host with an appropriate error cause value (shown below in parentheses). Possible error conditions are:

1 The message is not subscribed to (Not-Allowed).

2 The Position ID of the OrigAddress does not correspond to an ACD agent who is the controller of the call. As well, the ACD agent must belong to the same customer group as the CompuCALL session over which dv-Drop-Party is sent (invalid parameter with InvalidOrigAddress).

3 Agent not logged in (Agent-not-logged-in).

4 The call did not reach the state (through the RO dv-Add-Party) capable of dropping a consult party (Invalid-Call-State).

5 Any of the required parameters missing (Missing-Parameter with the missing parameter included).

6 The DropPartyType is not "dropConsultParty" (Invalid-Parameter with value "InvalidDropPartyType).

Note: This list is presented in the order of priority since only the first error condition (including the first missing/invalid parameter) encountered by the switch is included in the Return Error message.

4.7.12 ACD dv-Transfer-Party (operation value "D" hex)

This RO is a request by the host to the DMS-100 to perform a call transfer from the held original call party to the consult party. The host sends this RO of Operation Class 2 to the switch to transfer a call on behalf of an ACD agent. This message is valid for Meridian ACD CompuCALL options only.

4.7.12.1 dv-Transfer-Party ASN.1 encoding

```
dv-Transfer-Party          OPERATION
                           ARGUMENT          TransferPartyArgument
                           RESULT            TransferPartyResult
                           ERRORS            {  invalid-Call-State,
                                                missing-Parameter,
                                                invalid-Parameter
                                                not-Allowed
                                                agent-not-Logged-in}
::= 13

TransferPartyArgument      ::=  SEQUENCE
 {  origAddress            [2] AddressType                 OPTIONAL}

AddressType                :: =     CHOICE
 {  positionID             [0] IMPLICIT INTEGER,
    dialedDigits           [1] IMPLICIT IA5STRING,
    stationNumber          [2] IMPLICIT OCTET STRING,
    q931Address            [3] Q931AddressType}

Q931AddressType            :: =     SEQUENCE
 {  numberTypeNumberPlan            [0] OCTET STRING       OPTIONAL,
    presentationScreeningIndicator  [1] OCTET STRING       OPTIONAL,
    digits                          [2] IA5 STRING         OPTIONAL}
```

```
TransferPartyResult          ::=  SEQUENCE
 {   networkCallID           [0]  IMPLICIT SEQUENCE
     {   networkNodeID           [0] IMPLICIT INTEGER        OPTIONAL,
         localCallID             [1] IMPLICIT INTEGER        OPTIONAL}
                                                             OPTIONAL}


invalid-Call-State           ERROR               ::= 0

missing-Parameter            ERROR               ::= 1
                             PARAMETER           MissingParameter

MissingParameter             ::=      SEQUENCE
 {   missingParameterType    [0] IMPLICIT ENUMERATED
     {   missingOrigAddress      [1]}                        OPTIONAL}

invalid-Parameter            ERROR               ::= 2
                             PARAMETER           InvalidParameter

InvalidParameter             ::=      SEQUENCE
 {   invalidParameterType    [0] IMPLICIT ENUMERATED
     {   invalidOrigAddress      (2)}                        OPTIONAL}

not-Allowed                  ERROR               ::= 3

agent-Not-Logged-In          ERROR               ::= 5
```

4.7.12.2 ACD dv-Transfer-Party: coded example

After the consult call is set up, the host may transfer the call. The host requests to transfer the call being held to the consult party, dropping the controller from the call, by sending dv-Transfer-Party with invokeID = 6. After verifying the request, the switch sends Return Result that the transfer request will be initiated. If the switch receives the dv-Transfer-Party message with a missing OrigAddress, it sends a Return Error message that the OrigAddress parameter is missing. Similarly, if it receives the message when the agent is not logged in, it sends a Return Error that the agent is not logged in.

INVOKE DMS < ====== HOST

Table 74 dv-Transfer-Party Invoke

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	0D		INVOKE	this message is 13 bytes decimal
02	01	06	InvokeID	the InvokeID is 6 decimal
02	01	0D	Operation	it operation value is dv-Transfer-Party[13] decimal
30	05		Argument	the TransferPartyArgument is of type sequence and is 5 bytes long decimal
A2	03		OrigAddress	the OrigAddress is AddressType (CHOICE)
80	01	21	PositionId	the agent’s PositionId is 33 decimal
Hex Dump = A1 0D 02 01 06 02 01 0D 30 05 A2 03 80 01 21				

RETURN-RESULT DMS ===== > HOST

Table 75 dv-Transfer-Party Return-Result

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	06	InvokeID	the InvokeID is 6 decimal
30	0E		ReturnErrorSequence	this is a sequence 14 bytes long decimal
02	01	0D	Operation	its operation is dv-Transfer-Party[13] decimal
30	09		TransferPartyResult	the TransferPartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this is of type sequence and is 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 06 30 0E 02 01 0D 30 09 A0 07 80 01 01 81 02 05 55				

RETURN-ERROR DMS ===== > HOST

Table 76 dv-Transfer-Party Return-Error example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	06	InvokeID	the InvokeID is 6 decimal
02	01	01	ErrorType	the ErrorType is MissingParamter[1]
30	0E		ReturnErrorSequence	this is a sequence 3 bytes long decimal
80	01	02	ErrorParameter	the MissingParameter is MissingOrigAddress (2)
Hex Dump = A3 0B 02 01 06 02 01 01 30 0E 80 01 02				

RETURN-ERROR DMS ===== > HOST

Table 77 dv-Transfer-Party Return-Error example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	06	InvokeID	the InvokeID is 6 decimal
02	01	05	ErrorValue	the ErrorValue is Agent-Not-Logged-In[5]
Hex Dump = A3 06 02 01 06 02 01 05				

4.7.12.3 dv-Transfer-Party parameters

The host sends dv-Transfer-Party message to initiate transfer of the held original call party on behalf of a specific agent position. The agent position needs not be in the Associated environment with which the host has an established session. A brief description of the dv-Transfer-Party invoke parameter follows.

OrigAddress is a choice parameter identifying the controller. The only choice for OrigAddress is PositionID which uniquely identifies an agent position (not the actual agent) for the customer in a switch.

This parameter must be included in the dv-Transfer-Party message. Upon receiving a dv-Transfer-Party message, the switch ensures that it is a valid request. If none of the following errors occurs, the switch returns a Return Result message to the host and initiates the transfer. The Return Result message always contains the NetworkCallID. At this point, there will be no more CompuCALL signaling done with this request. Should anything cause the request to fail, inbound signaling will be given to the agent.

4.7.12.4 dv-Transfer-Party error conditions

Any error conditions result in the call remaining in its current state and a Return Error message being returned to the host with an appropriate error cause value (shown below in parentheses). Possible error conditions are:

1. The message is not subscribed to (Not-Allowed);
2. The PositionID of the OrigAddress does not correspond to an ACD agent who is the controller of the call. As well, the ACD agent must belong to the same customer group as the CompuCALL session over which dv-Transfer-Party is sent (Invalid-Parameter with value "invalidOrigAddress");
3. Agent not logged in (Agent-not-logged-in).

4 The call did not reach the state (through the dv-Add-Party message) capable of being transferred (Invalid-Call-State).

5 Any of the required parameters missing (Missing-Parameter with the missing parameter included).

> ***Note:*** This list is presented in order of priority since only the first error condition (including the first missing/invalid parameter) encountered by the switch is included in the Return Error message.

If transfer is not allowed, the result will depend on the reason:

a) If the Service Profile for the session does not contain a dv-Transfer-Party message, then the DMS-100 sends a Return Error and the controller remains connected to the consult party. The controller can then use the "drop party" function on the data terminal, or use the RLS key (on MBS) or go on-hook (on 2500 set) to revert to the original call party.

b) If the controller does not have the MDC call transfer capability enabled on the DMS-100 switch, the DMS-100 switch sends a Return Result and releases the consult party. The call is reverted to the original call party. It is therefore important that the equivalent MDC capability be active for the controller.

4.7.13 ACD dv-Make-Call (operation value 9)

The host uses the dv-Make-Call message to instruct the switch to set up an outbound call. This message is valid for Meridian ACD, residential, and Centrex CompuCALL options. The host send this RO of Operation Class 2 to the switch when an agent initiates a call at the data terminal.

4.7.13.1 dv-Make-Call ASN.1 encoding

```
dv-Make-Call        OPERATION
                    ARGUMENT         MakeCallArgument
                    RESULT           MakeCallResult
                    ERRORS           { not-Allowed,
                                       not-Idle,
                                       agent-Not-Logged-In,
                                       orig-Time-Out,
                                       makeCallAborted,
                                       makeCall-Mismatched-State,
                                       makeCall-Resources-Unavailable,
                                       missingOrigAddress,
                                       missingDestAddress,
                                       missingMakeCallType,
                                       invalidOrigAddress,
                                       invalidDestAddress,
                                       invalidMakeCallType,
                                       invalidAuthCode,
                                                  invalidAcctCode,
                                                  authOptNotSubscribed,
                                                  acctOptNotSubscribed
                                                  unexpectedAcctCode}
                                ::= 9

MakeCallArgument                ::=     SEQUENCE
 {  origAddress                 [0] AddressType                OPTIONAL,
    destAddress                 [1] AddressType                OPTIONAL,
    applicationData             [2] IMPLICIT SEQUENCE
     {  makeCallType                [0] IMPLICIT ENUMERATED
         {  callingAgentOnline            (0),
            callingAgentReserved          (1),
            callingAgentNotReserved       (2), callingAgentBuzzBase      (3),

                                callingAgentBeepHset         (4)}OPTIONAL,
        authCodeDigits              [1] IMPLICIT IA5String     OPTIONAL,
        acctCodeDigits              [2] IMPLICIT IA5String     OPTIONAL}
                                                               OPTIONAL}
AddressType                     :: =    CHOICE
 {  positionID                  [0] IMPLICIT INTEGER,
    dialedDigits                [1] IMPLICIT IA5STRING,
    stationNumber               [2] IMPLICIT OCTET STRING,
    q931Address                 [3] Q931AddressType}

Q931AddressType                 :: =    SEQUENCE
 {  numberTypeNumberPlan                [0] OCTET STRING       OPTIONAL,
    presentationScreeningIndicator      [1] OCTET STRING       OPTIONAL,
    digits                              [2] IA5 STRING         OPTIONAL}


MakeCallResult                  ::=     SEQUENCE
 {  networkCallID               [0] IMPLICIT SEQUENCE
     {  networkNodeID               [0] IMPLICIT INTEGER       OPTIONAL,
        localCallID                 [1] IMPLICIT INTEGER       OPTIONAL}
                                                               OPTIONAL}
                not-AllowedERROR::= 1

                not-IdleERROR::= 2

                agent-Not-Logged-In ERROR::= 3

                orig-Time-OutERROR::= 4

                makeCall-Resources-
                Unavailable                     ERROR::= 5
                missingOrigAddress              ERROR::= 6
```

missingDestAddress	ERROR::= 7
missingMakeCalltype	ERROR::= 8
invalidOrigAddress	ERROR::= 9
invalidDestAddress	ERROR::= 10
invalidMakeCalltype	ERROR::= 11
invalidAuthCode	ERROR::= 12
invalidAcctCode	ERROR::= 13
authOptNotSubscribed	ERROR::= 14
acctOptNotSubscribed	ERROR::= 15
makeCallAborted	ERROR::= 16
makeCallMismatchedState	ERROR::= 17
unexpectedAcctCode	ERROR::= 18

4.7.13.2 ACD dv-Make-Call: coded example 1

The host asks the DMS-100 to place an outgoing call on behalf of the agent whose positionID = 1131 and who is on-line to the destination address (DialedDigits) = 26050. The AuthCode and AcctCode digits are 1234 and 567 respectively. The switch verifies the request and sends Return Result to the host when the agent acknowledges the dv-Make-Call request (agent answers the alerting on the Incalls key or answers MDC set). If a dv-Make-Call is sent while one is in progress, a Return Error is sent that the agent is not idle.

INVOKE DMS < ====== HOST

Table 78 dv-Make-Call Invoke coded example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	27		INVOKE	this message is 39 bytes decimal
02	01	28	InvokeID	the InvokeID is 40 decimal
02	01	09	Operation	the operation is dv-Make-Call [9]
30	1F		Argument	MakeCall argument is of sequence type and is 31 bytes long decimal
A0	04		OrigAddress	the OrigAddress is a constructor 6 bytes
80	02	04 6B	PositionId	the agent's PositionId is 1131
A1	07		DestAddress	the DestAddress is a construction of 7 bytes long
A2	0E		ApplicationData	ApplicationData is a constructor 14 bytes long
80	01	00	MakeCallType	the MakeCallType is CallingAgentOnline
81	04	31 32 33 34	AuthCodeDigits	the AuthCodeDigits are 1234
82	03	35 36 37	AcctCodeDigits	the AcctCodeDigits are 567
Hex Dump = A1 27 02 01 28 02 01 09 30 1F A0 04 80 02 04 6B A1 07 A2 0E 80 01 00 81 04 31 32 33 34 82 03 35 36 37				

RETURN-RESULT DMS ===== > HOST

Table 79 dv-Make-Call Return-Result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	12		RETURN RESULT	this message is 18 bytes long decimal
02	01	28	InvokeID	the InvokeID is 40 decimal
30	0D		ReturnResult Sequence	this is a sequence 13 bytes long decimal
02	01	09	Operation	the operation is dv-Make-Call[9]
30	08		MakeCallResult	the MakeCallResult is of type sequence and is 8 bytes long
A0	06		NetworkCallID	the NetworkCallID is of type sequence and is 6 bytes long
80	01	01	NetworkNodeID	the NetworkNodeID component is 1
81	01	02	LocalCallID	the LocalCallID component is 2 decimal
Hex Dump = A2 12 02 01 28 30 0D 02 01 09 30 08 A0 06 80 01 01 81 01 02				

RETURN-ERROR DMS ===== > HOST

Table 80 dv-Make-Call Return-Error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	28	InvokeID	the InvokeID is 40 decimal
02	01	02	ErrorValue	the ErrorValue is Not-Idle[2].
Hex Dump = A3 06 02 01 28 02 01 02				

4.7.13.3 ACD dv-Make-Call: coded example 2

INVOKE **DMS < ====== HOST**

Table 81 dv-Make-Call Invoke coded example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	2A		INVOKE	the message length is 42 bytes decimal
02	01	28	InvokeID	its Invoke ID is decimal 40
02	01	09	Operation	its operation value is [9] decimal
30	22		Argument	MakeCallArg is of type sequence and is 34 bytes long decimal
A0	0C		origAddress	the origination address is of type constructor and is 12 bytes long
82	0A	35 30 36 38 35 38 31 32 31 32	stationNumber	the station number is 506-858-1212
A1	0D		destAddress	the destination address is of type constructor and is 13 bytes long decimal
81	0B	31 36 31 33 35 35 35 31 32 31 32	dialedDigits	the dial DN is 1-613-555-1212
A2	03		applicationData	Application Data is a constructor and is 3 bytes long.
80	01	03	makeCallType	makeCallType is a primitive enumerated value, in this case = CallingAgentBuzzBase.
Hex dump = A1 2A 02 01 28 02 01 09 30 22 A0 0C 82 0A 35 30 36 38 35 38 31 32 31 32 A1 0D 81 0B 31 36 31 33 35 35 35 31 32 31 32 A2 03 80 01 03				

RETURN-RESULT DMS ===== > HOST

Table 82 dv-Make-Call example 2 Return-Result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	12		RETURN RESULT	the message length is 18 bytes decimal
02	01	28	InvokeID	its Invoke ID is decimal 40
30	0D		Return Result Sequence	this sequence is 13 bytes long decimal
02	01	09	Operation	its operation is dv-Make-Call [9]
30	08		MakeCall Result	the MakeCallResult is of type sequence and is 8 bytes long
A0	06		Network Call ID	NetworkCallId is of type sequence and is 6 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	01	38	Local Call ID	Local Call ID is 56.
Hex dump = A2 12 02 01 28 30 0D 02 01 09 30 08 A0 06 80 01 01 81 01 38				

RETURN-ERROR DMS ===== > HOST

Table 83 dv-Make-Call example 2 Return-Error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	the message length is 6 bytes decimal
02	01	28	InvokeID	its Invoke ID is decimal 40
02	01	04	Error Value	Error Value is 4 representing Orig-Time out occurred (user did not respond to alerting by answering their set).
Hex dump = A3 06 02 01 28 02 01 04				

The host sends the dv-Make-Call message to the switch to initiate an outbound call on behalf of a specific agent position. The agent position does not need to be in the Aassociated environment with which the host has an established session. When the host sends a makeCall without associating the original agent, subsequent messages for this call will have different callIDs.

4.7.13.4 dv-Make-Call parameters

A brief description of the parameters of the dv-Make-Call invoke follows

OrigAddress is a choice parameter identifying the originator of the call to be set-up.The only choice for OrigAddress is PositionID. This parameter uniquely identifies an agent position (not the actual agent) for the customer in a switch.

DestAddress is a choice parameter identifying the terminator (other-end) of the call to be set-up. The only choice for DestAddress is DialedDigits. This is a string of IA5-encoded digits corresponding to the dialed digits if the dialing were manual. The digits are translated and routed according to the business group (equal to DMS customer group) and Network Class of Service (NCOS) datafill set up for the originator of the call.

ApplicationData contains any application-specific subparameters the host wants to pass to the DMS. For the dv-Make-Call message, the ApplicationData parameter contains one or more of the following subparameters.

MakeCallType indicates the necessary agent state for the outbound call to proceed There are five makeCallType: CallingAgentOnline, CallingAgentReserved, CallingAgentNotReserved, CallingAgentBuzzBase, and CallingAgentBeepHset. CallingAgentOnline means that the agent (i.e., operator) must be on-line (logically on-hook, available to interact with the destination party) before the call proceeds. CallingAgentReserved specifies that the MakeCall request is permitted only if ACDNR on the agent set is activated. When CallingAgentNotReserved is specified, the ACDNR feature on the agent set must NOT be activated for the MakeCall request to be permitted. These types relate to distinctive ringing with dv-Make-Call (see below). When a dv-Make-Call is originated with the MakeCallType parameter as Calling_Agent_Buzz_Base, a buzz will be applied to the base of the telephone set and the setup of the call will proceed to route the call to the destination. When a dv-Make-Call is originated with the MakeCallType parameter as Calling_Agent_Beep_Hset, a tone will be applied to the head set and the setup of the call will proceed to route the call to the destination.

AuthCodeDigits subparameter may be provided at the beginning of a call as the user identification. It contains the authorization code digits to be verified before the agent accesses the appropriate NCOS other than the current default NCOS. Furthermore, the authorization code digits may be further categorized into authorization code types verified by the switch for additional security. Section 4.14.5 provides the interaction of dv-Make-Call with the authorization and account code features.

AcctCodeDigits subparameter is for internal cost allocation purposes only. It allows the agent or the host to enter a cost-accounting or client number into the station message detail recording (SMDR) record for charge back purposes. If the combined authorization and account code option is chosen, AcctCodeDigits must be included in AuthCodeDigits.

If any subparameter is in the service profile for that session, it must also be in the dv-Make-Call message for it to be valid. The service profile to which the customer subscribes must always contain the MakeCallType subparameter. However, AuthCodeDigits and AcctCodeDigits are optional.

4.7.13.5 dv-Make-Call Return-Result parameters

The dv-Make-Call Return Result includes the following parameter:

NetworkCallID identifies the call in the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

Various distinctive ringing tones are provided for Meridian ACD CompuCALL customers so agents can differentiate outbound calls initiated via the CompuCALL interface from normal inbound calls.

Customers can use distinctive ringing on their sets for inbound and/or outbound calls. Distinctive ringing does not apply when call forcing is in effect. In this case, or when the agent wants to receive only outbound calls, MakeCallType, "CallingAgentReserved" may be used which essentially "reserves" the agent for that outbound call. Or"CallingAgentNotReserved" may be used which provides more flexibility to the agent to accept both inbound and outbound calls.

The Set Off-Hook message will not be sent for either MakeCall or add party calls since the information would be redundant (the network call ID is currently sent as a return result from either of these messages, which indicates the call was successfully originated).

4.7.13.6 dv-Make-Call validations

On receiving a dv-Make-Call message, the switch:

1 The parameters OrigAddress, DestAddress, and ApplicationData are in the message.

2 The PositionID is valid for the customer group for the active Meridian ACD CompuCALL session.

3 The DialedDigits will translate successfully according to the datafill for the originator in the DMS.

4 The ApplicationData parameter contains a MakeCallType of the appropriate value. For example, the agent must be on-line when the MakeCallType is "CallingAgentOnLine" before the call proceeds.

5 The authorization code in the MakeCall request is valid for the agent to access the NCOS for that call if the authorization code is in the MakeCall request.

6 The authcode can be used for that originating station.

7 The acctcode length is valid if the acctcode is in the MakeCall request.

8 Checks the agent’s capability to make outbound call by verifying that

a. The customer group to which the DN belongs can initiate outbound calls.

b. The set is logged in as an ACD agent.

c. Ensuring DN is in the proper state as specified in the MakeCallType sub-parameter (Note: for the CallingAgentBuzzBase or CallingAgentBeepHset, the DN on behalf of which the dv-Make-Call message is sent must be idle but the main hand set must be offline. Offline means the DN is available to interact with the destination party).

If both the above checks succeed, the additional checks for the new MakeCallType sub-parameters are made as follows:

1 Verifies that the originating agent or DN(as specified by the OrigAddress field) is a key set and not a 2500/500 type set.

2 Verifies that the originating agent or line is also a ACD/CTX line.

dv-Make-Call requires the agent set to be in the state as specified in the request, i.e., logically on-hook and idle or logically on-hook with ACDNR feature activated for CallingAgentOnLine; or logically on-hook with ACDNR feature activated for CallingAgentReserved; or logically on-hook with ACDNR feature not activated for CallingAgentNotReserved. If call forcing is assigned to an ACD set, it is possible to issue a dv-Make-Call request with MakeCallType "CallingAgentReserved" to reserve agent for that outbound call while the set is physically off hook, but the line is logically in the idle state. (Note: call forcing does not apply to 500/2500 sets).

If the request to initiate the call is valid, the agent will receive an alerting signal (physical ringing or a tone). This signal can be a distinctive ringing tone as specified by the customer to inform the agent of an outbound call if Call Forcing is not in effect. Also, the agent must "accept" the outgoing call (by going off-hook, for example). In this case after the agent accepts the call, the switch sends a Return Result (positive acknowledgment) message to the host with the parameter NetworkCallID. Then the switch proceeds with the call set-up attempt. If call forcing is in effect, the agent will be presented with the outbound call after a short tone burst.

4.7.13.7 dv-Make-Call error conditions

If the request to initiate the call is unsuccessful, or the ring threshold time expires before the agent accepts the outgoing call, the switch sends a Return Error (negative acknowledgment) message to the host indicating failure due to:

1 Customer is not allowed to make outbound calls (Not-Allowed).

2 ACDNR on agent set does not match MakeCallType (MakeCall-Mismatched-State).

3 Invalid Origination or Destination Address (InvalidOrigAddress or InvalidDestAddress).

4 Agent not available to receive outbound call (i.e., busy on incoming/outgoing call or not logged in) (Not-Idle or Agent-Not-Logged-In).

5 MakeCall resources unavailable.

6 Missing parameter (identified).

7 Invalid MakeCallType.

8 Invalid AuthCode/AcctCode.

9 The AuthCodeDigits subparameter is not subscribed.

10 The AcctCodeDigits subparameter is not subscribed.

11 Agent time out (failed to respond within period specified by ring threshold field of table ACD group).

12 The AcctCodeDigits subparameter is already provided in a combined form in the AuthCodeDigits subparameter upon the subscription to the combined authorization/account code option.

Except for agent time out, this list is presented in the order of priority since only the first error condition (including the first missing/invalid parameter) encountered by the switch is included in the Return Error message.

The switch does not retransmit if it receives a Reject message in response to a Return Result or Return Error message.

4.7.14 ACD dv-Controller-Released-U (Operation Value 48)

If a 3WC to a 2WC state transition results in a call transfer controlled by a controller, the DMS-100 switch sends a dv-Controller-Released-U message to the host that contains the DN of the releasing party.

4.7.14.1dv-Controller-Released-U ASN.1 Encoding

```
 dv_CONTROLLER_RELEASED_U                       OPERATION

 ARGUMENT                                        Ctrl_releasedargument

 := 48

Ctrl_releasedargument:=               SEQUENCE

{NetworkCallID              [0]      IMPLICIT    SEQUENCE

      {networkNodeID        [0]        IMPLICIT    INTEGER     OPTIONAL,

       localCallID          [1]       IMPLICIT         INTEGER     OPTIONAL}

                                                                OPTIONAL,

 Device_id                [1]        AddressType                         OPTIONAL

 Released_Party_info              [2]       ReleasedPartyInfoType       OPTIONAL}

AddressType::= CHOICE
 {positionID[0]                        IMPLICIT     INTEGER,
  dialedDigits[1]                      IMPLICIT     IA5 STRING,
  stationNumber [2]                    IMPLICIT     OCTET   STRING,
  q.931Address[3]                      Q.931AddressType}

Q.931AddressType::= SEQUENCE
 {numberTypeNumberPlan     [0]       IMPLICIT   OCTET    STRING    OPTIONAL,
  presentationScreeningIndicator[1]  IMPLICIT   OCTET    STRING    OPTIONAL,
  digits [2]                          IMPLICIT      IA5 STRING      OPTIONAL}
```

```
ReleasedPartyInfoType::=CHOICE

{   POSID           [0]                 UNSIGNEDINT,

    DEVICEID        [1]                 DEVICEADDRESSTYPE}

DEVICEADDRESSTYPE ::= SET OF

    { dialedDigits    [1]     IMPLICIT    IA5 String          OPTIONAL,

     stationNumber    [2]     IMPLICIT    OCTET STRING        OPTIONAL,

    name              [4]     IMPLICIT    IA5 STRING          OPTIONAL}
```

4.7.14.2 ACD dv-Controller-Released-U Coded Example

All three parties in a 3WC are on the same switch. The controller, party A, is an ACD agent with position ID 922. The first leg of the call was initiated by party B. Party B is ECM associated at station number 919-722-1005. The controller exits the 3WC and the switch sends the dv-Controller-Released-U message to the originating party of the 3WC.

INVOKE DMS < ====== HOST

Table 84 dv-Control-Released-U Invoke

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	25		INVOKE	this message is 37 bytes decimal
02	02	03FF	InvokeID	the InvokeID is 1023 decimal
02	01	30	Operation	the operation is dv-Controller-Released-U [48]
30	1C		Ctrl_releaseargument	Ctrl_releaseargument is of sequence type and is 28 bytes long
A0	06		NetworkCallID	the NetworkCallID is a constructor 6 bytes decimal
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	01	02	LocalCallID	the LocalCallID is 2 decimal
A1	0C		DeviceID	the DeviceID is a construction of 12 bytes decimal long
82	0A	39 31 39 37 32 32 31 30 30 35	Station number	the Station number of the destination device (919-722-1005)
A2	04		Released_Party_info	the Released_Party_info is a constructor 4 bytes long
80	02	039A	POSID	the Position ID of the ACD agent is 922 decimal
Hex Dump = A1 25 02 02 03 FF 02 01 30 30 1C A0 06 80 01 01 81 01 02 A1 0C 82 0A 39 31 39 37 32 32 31 30 30 35 A2 04 80 02 03 9A				

4.7.14.3 dv-Controller-Released-U parameters

The dv-Controller-Released-U message includes the following parameter:

NetworkCallID an optional parameter that identifies the call in the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

deviceID a required parameter that identifies the address of the party intended to receive the new message.

Released-Party-Info a required parameter identifies the released party by the position ID, station number or dialed digits, and the releasing party’s name, if possible. Only the controller receives the Released-Party-Info parameter. The non-controller does not receive this parameter in the message. The releasing party’s name appears in the message only when the releasing party is a RES/MDC line without presentation of name restrictions.

4.7.15 ACD dv-nonController-Released-U (Operation Value 49)

If a 3WC to a 2WC state transition does not result from a call transfer the DMS-100 switch sends a dv-nonController-Released-U message to the host that contains the DN of the releasing party. The DN of the releasing party can consist of one of the following:

- Position ID
- station number
- dialed digits

4.7.15.1dv-nonController-Released-U ASN.1 Encoding

```
 dv_NONCONTROLLER_RELEASED_U              OPERATION

 ARGUMENT                                        nonctrl_releasedargument

 := 49

nonctrl_releasedargument :=              SEQUENCE

{NetworkCallID                  [0]             IMPLICIT     SEQUENCE

       {networkNodeID          [0]             IMPLICIT    INTEGER     OPTIONAL,

        localCallID            [1]             IMPLICIT    INTEGER     OPTIONAL}

                                                                      OPTIONAL,

Device_id                 [1]          AddressType                     OPTIONAL,
 Released_Party_Info       [2]        ReleasedPartyInfoType            OPTIONAL ,

  Reason                        [3]     IMPLICIT        ENUMERATED   OPTIONAL }
```

```
AddressType ::= CHOICE
 {positionID[0]                    IMPLICIT    INTEGER,
  dialedDigits[1]                  IMPLICIT    IA5 STRING,
  stationNumber [2]                IMPLICIT    OCTET   STRING,
  q.931Address[3]                  Q.931AddressType}

Q.931AddressType::= SEQUENCE
 {numberTypeNumberPlan     [0]     IMPLICIT   OCTET    STRING    OPTIONAL,
  presentationScreeningIndicator[1]  IMPLICIT   OCTET    STRING    OPTIONAL,
  digits [2]                       IMPLICIT     IA5 STRING       OPTIONAL}

ReleasedPartyInfoType::=CHOICE

{   POSID            [0]                UNSIGNEDINT,

    DEVICEID      [1]                   DEVICEADDRESSTYPE}

DEVICEADDRESSTYPE ::= SET OF

    { dialedDigits     [1]      IMPLICIT     IA5 String             OPTIONAL,

     stationNumber   [2]        IMPLICIT    OCTET STRING       OPTIONAL,

    name                [4]     IMPLICIT     IA5 STRING          OPTIONAL}


Reason::= IMPLICIT ENUMERATED

{ first_leg_party_released  [0]

second_leg_party_released  [1] }
```

4.7.15.2 ACD dv-nonController-Released-U Coded Example

All three parties in a 3WC are on the same switch. The controller, party A, is an ACD agent with position ID 4196. Party B (613-770-5543) calls party A to set up the first leg of the call. The controller places party B on hold and calls party C (613-621-4442). Party B releases the call while on hold. The controller and party C both receive the dv-nonController-Released-U message.

INVOKE DMS < ====== HOST (message sent to the controller)

Table 85 dv-nonController-Released-U Invoke controller

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	39		INVOKE	this message is 57 bytes decimal
02	02	03FF	InvokeID	the InvokeID is 1023 decimal
02	01	31	Operation	the operation is dv-nonController-Released-U [49]
30	30		nonctrl_releaseargument	nonctrl_releaseargument is of sequence type and is 48 bytes decimal long
A0	06		NetworkCallID	the NetworkCallID is a constructor 6 bytes decimal
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	01	03	LocalCallID	the LocalCallID is 3 decimal
A1	04		DeviceID	the DeviceID is a construction of 4 bytes decimal long
80	02	1064	POSID	the positionId of the agent is 4196
A2	1D		Released_Party_info	the Released_Party_info is a constructor 29 bytes long
A1	1B		DeviceID	the DeviceID is a construction of 27 bytes decimal long
82	0A	36 31 33 37 37 30 35 35 34 33	Station number	the Station number of the released party device (613-770-5543)
84	0D	53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53	name	the name of the released party is SHANANA JONES of length 13
83	01	00	Reason	the reason for receiving the message is 0, which indicates the call was released by the first leg party
Hex Dump = A1 39 02 02 03 FF 02 01 31 30 30 A0 06 80 01 01 81 01 03 A1 04 80 02 10 64 A2 1D A1 1B 82 0A 36 31 33 37 37 30 35 35 34 33 84 0D 53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53 83 01 00				

INVOKE DMS < ====== HOST (message sent to the conferenced party)

Table 86 dv-nonController-Released-U Invoke conferenced party

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	24		INVOKE	this message is 36 bytes decimal
02	02	03FF	InvokeID	the InvokeID is 1023 decimal
02	01	31	Operation	the operation is dv-nonController-Released-U [49]
30	1B		nonctrl_releaseargument	nonctrl_releaseargument is of sequence type and is 27 bytes decimal long
A0	06		NetworkCallID	the NetworkCallID is a constructor 6 bytes decimal
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	01	03	LocalCallID	the LocalCallID is 3 decimal
A1	0C		DeviceID	the DeviceID is a construction of 12 bytes decimal long
82	0A	36 31 33 36 32 31 34 34 34 32	Station number	the Station number of the released party device (613-621-4442)
A2	03		Released_Party_info	the Released_Party_info is a constructor 26 bytes long
83	01	00	Reason	the reason for receiving the message is 0, which indicates the call was released by the first leg party
Hex Dump = A1 24 02 02 03FF 02 01 31 30 1B A0 06 80 01 01 81 01 03 A1 0C 82 0A 36 31 33 36 32 31 34 34 34 32 A2 03 83 01 00				

4.7.15.3 dv-nonController-Released-U parameters

The dv-nonController-Released-U message includes the following parameter:

NetworkCallID an optional parameter that identifies the call in the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

deviceID a required parameter that identifies the address of the party intended to receive the new message.

Released-Party-Info a required parameter identifies the released party by the position ID, station number or dialed digits, and the releasing party's name, if possible. Only the controller receives the Released-Party-Info parameter. The non-controller does not receive this parameter in the message. The releasing party's name appears in the message only when the releasing party is a RES/MDC line without presentation of name restrictions

Reason a required parameter that identifies the party that released the call. A value of 0 indicates the noncontroller of the first leg of the call performed the release. A value of 1 indicates the noncontroller of the second leg of the call performed the release.

4.8 ACD Third Party Agent Control

Third Party Agent Control (TPAC) allows an ACD agent with a voice set and a data terminal to initiate agent position status changes at a data terminal. The RO used in the host-to-switch direction is dv-Set-Feature. The function associated with this RO is to change a specific agent position status. These include: Agent LogIn, Agent LogOut, Agent Ready, Agent Not Ready, Agent Reserved, Agent Not Reserved, Change Call Forcing, and Change Variable Wrapup. Reports of agent status changes are sent to the host associated with ACD DNs active with a Meridian ACD CompuCALL session. The ROs in the switch-to-host direction are: dv-Agent-Logged-In-U, dv-Agent-Logged-Out-U, dv-Agent-Ready-U, and dv-Agent-Not-Ready-U.

The agent states requested by a host or reported by the switch include the following states: Agent Log In, Agent Log Out, Agent Not Ready, and Agent Ready.

Reports of agent status changes are sent when the state change occurs, not when the switch receives a request.

The ASN.1 notation and a coded example for these messages are in Appendix A of this Volume. Appendix B provides more details on the parameters and their structures. These messages are valid for Meridian ACD CompuCALL options only. This Application Service capability covers the following Application Service functions:

- dv-Set-Feature
- dv-Agent-SetAction-U
- dv-Agent-Logged-In-U
- dv-Agent-Logged-Out-U
- dv-Agent-Ready-U
- dv-Agent-Not-Ready-U

4.8.1 ACD dv-Set-Feature (operation value "11" hex)

The host uses the RO of Operation Class 2 dv-Set-Feature message to ask the switch to logIn/logOut an ACD agent position or to place the agent position in the Ready/Not Ready state. These requests are initiated by the agent at a data terminal associated with the host. This message is valid for Meridian ACD CompuCALL options only.

4.8.1.1dv-Set-Feature ASN.1 encoding

```
dv-Set-Feature                    OPERATION
    ARGUMENT                      SetFeatureArgument
        RESULT
        ERRORS                    {not-Allowed,
                                  missing-Parameter,
                                  invalid-Parameter,
                                  invalid-Parameter-Content,
                                  invalid-Line-Configuration,
                                  agent-Already-Logged-In,
                                  login-ID-In-Use-Elsewhere,
                                  position-Already-Logged-In,
                                  invalid-Set-State,
                                  password-Mismatch,
                                  resources-Unavailable,
                                  agent-Not-Logged-In,
                                  agent-Logout-Pending,
                                  invalid-AgtPos-State,
                                  agent-Presently-Ready,
                                  agent-Presently-Not-Ready,
                                  supervisor-Override
                                  agent-presently-reserved,
                                  agent-presently-unreserved,
                                  wrapup-presently-inactive,
                                  forcing-presently-inactive}
                    ::= 17

SetFeatureArgument   ::=SEQUENCE
{origAddress        [0] AddressTypeOPTIONAL,
set-Feature-Info    [1] SetFeatureInformationOPTIONAL,
privateData         [2] SetFeaturePrivateDataOPTIONAL}

AddressType         :: =     CHOICE
{positionID         [0] IMPLICIT INTEGER,
dialedDigits        [1] IMPLICIT IA5STRING,
stationNumber       [2] IMPLICIT OCTET STRING,
q931Address         [3] Q931AddressType}

Q931AddressType:: =          SEQUENCE
```

```
{numberTypeNumberPlan          [0] OCTET STRINGOPTIONAL,
presentationScreeningIndicator [1] OCTET STRINGOPTIONAL,
digits                         [2] IA5StringOPTIONAL}

SetFeatureInformation:: = CHOICE
    {agentFeature              [0] SetAgentFeature}

SetAgentFeature:: = CHOICE
    {logIn             [0] IMPLICIT LogInInfoType,
    logOut             [1] IMPLICIT NULL,
    ready              [2] IMPLICIT NULL,
    notReady           [3] IMPLICIT NULL,
    reserve            [4] IMPLICIT NULL,
    unreserve          [5] IMPLICIT NULL,
    CHGVWRAP           [6] IMPLICIT CHGVWRAPInfoType,
    CHGFORCE           [7] IMPLICIT ChngforcingInfoType}

LogInInfoType:: = SEQUENCE
    {loginID           [0] IMPLICIT IA5StringOPTIONAL,
    password           [1] IMPLICIT IA5StringOPTIONAL}

CHGVWRAPInfoType::= SEQUENCE
{WrapTimer             [0] IMPLICIT INTERGER OPTIONAL}

CHGFORCEInfoType::= SEQUENCE
{CHGFORCE_Type         [0] IMPLICIT ENUMERATED OPTIONAL}

CHGFORCE_Type::= ENUMERATED
Base                   (0),
Hset                   (1),
None                   (2) }



SetFeaturePrivateData:: = CHOICE
    {agentPrivateData  [0] AgentPrivateDataType}

AgentPrivateDataType:: = CHOICE
    {logInPrivateData  [0] IMPLICIT NULL,
    logOutPrivateData  [1] IMPLICIT NULL,
    readyPrivateData   [2] IMPLICIT NULL,
    notReadyPrivateData [3] IMPLICIT NotReadyPrivateDataType,
    reservePrivateData [4] IMPLICIT ReservePrivateDataType,
    unreservedPrivateData [5] IMPLICIT NULL,
    CHGVWWRAPPrivateData [6] IMPLICIT NULL,
    CHGFORCEPrivateData [7] IMPLICIT NULL}
```

```
NotReadyPrivateDataType:: = SEQUENCE
        {WalkawayReason     [0] IMPLICIT IA5StringOPTIONAL}
Reserve PrivateDataType ::= SEQUENCE
        {Reserv_timer           [0] IMPLICIT INTERGER OPTIONAL}
    not-Allowed                         ERROR::0

    missing-Parameter                   ERROR::1
            PARAMETERFeatureParm

    invalid-Parameter                   ERROR::2
            PARAMETERFeatureParm

    invalid-Parameter-Content           ERROR::3
            PARAMETERFeatureParm

    invalid-Line-Configuration          ERROR::4

    agent-Already-Logged-In             ERROR::5

    login-ID-In-Use-Elsewhere           ERROR::6

    position-Already-Logged-In          ERROR::7

    invalid-Set-State                   ERROR::8

    password-Mismatch                   ERROR::9

    resources-Unavailable               ERROR::10

    agent-Not-Logged-In                 ERROR::11

    agent-Logout-Pending                ERROR::12

    invalid-AgtPos-State                ERROR::13

    agent-Presently-Ready               ERROR::14

    agent-Presently-Not-Ready           ERROR::15

    supervisor-Override                 ERROR::16
    agent-presently-reserved            ERROR::=17
    agent-presently-unreserved          ERROR::=18
    wrapup-presently-inactive           ERROR::=19
    forcing-presently-inactive          ERROR::=20

    FeatureParm        :: =     SEQUENCE
        {SetFeatureParm [0] IMPLICIT SetFeatureParmTypeOPTIONAL,
        LogInParm[1] IMPLICIT LogInParmTypeOPTIONAL,
        LogOutParm[2] IMPLICIT NULLOPTIONAL,
        ReadyParm[3] IMPLICIT NULLOPTIONAL,
         NotReadyParm[4] IMPLICIT NotReadyParmTypeOPTIONAL,
        reserve [5] IMPLICIT ReserveParmType OPTIONAL,
        unreserve [6] IMPLICIT NULL OPTIONAL
        CHGVWRAP [7] IMPLICIT CHGVWRAPParmType OPTIONAL
        CHGFORCE [8] IMPLICIT ChngforcingParmType OPTIONAL}

    SetFeatureParmType:: = ENUMERATED
        {origAddress  (0),
        setFeatureInfo(1),
        privateData     (2)}

    LogInParmType:: = ENUMERATED
        {loginID(0),
        password(1)}

    NotReadyParmType:: = ENUMERATED
        {walkawayReason(0)}

    ReserveParmType ::= ENUMERATED
        {Reserv_Timer          (0)}

    CHGVWRAPParmType ::= ENUMERATED
        {WrapTimer             (0)}

    ChngforcingParmType ::= ENUMERATED
        {Frce_type             (0)}
```

4.8.1.2 ACD LogIn Agent Functionality: coded example

The host requests the switch to log in an ACD agent whose Login ID is 2345 at Position ID 0012 with a password of 5432 by sending a dv-Set-Feature message with Invoke ID of 3. The switch verifies the request and sends a Return Result that the request will be initiated.

If the switch receives the dv-Set-Feature message with a missing login ID, it sends a Return Error message so indicating. If it receives the message when the agent is already logged in, it sends a Return Error message indicating that.

INVOKE DMS < ====== HOST

Table 87 dv-Set-Feature login agent

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1F	03	INVOKE	this message is 31 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	11	Operation	the operation value is dv-Set-Feature [17] decimal
30	17		SetFeatureArgument	the SetFeatureArgument is of type sequence and is 23 bytes long decimal
A0	03		OrigAddress	OrigAddress is a constructor
80	01	0C	PositionID	the PositionID is 12
A1	10		SetFeatureInfo	the SetFeatureInfo is of type constructor and is 16 bytes long decimal
A0	0E		AgentFeature	the AgentFeature component
A0	0C		LogIn	the LogIn is of type constructor and is 12 bytes long decimal
80	04	32 33 34 35	LogInID	the LogInID is 2345
81	04	35 34 33 32	Password	the Password is 5432
Hex Dump = A1 1F 02 01 03 02 01 11 30 17 A0 03 80 01 0C A1 10 A0 0E A0 0C 80 04 32 33 34 35 81 04 35 34 33 32				

INVOKE DMS < ====== HOST

Table 88 dv-Set-Feature for reserve agent request

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	1B		INVOKE	the message length is 27 bytes decimal
02	01	05	InvokeID	its Invoke ID is 5
02	01	11	Operation	its operation value is dv-Set-Feature [17] decimal
30	14		SetFeatureArgument	SetFeatureArgument is of type sequence and is 20 bytes long
A0	04		OrigAddress	OrigAddress is a constructor
80	02	13 9D	PositionID	the agent's PositionID is 5021 decimal
A1	04		Set-Feature-Info	SetFeatureInformation is a constructor
A0	02		AgentFeature	Set AgentFeature is a constructor
84	00		Reserve	Reserve is of NULL type
A2	05		Set_feature_private_data	SetFeature Private Data is a constructor
A4	03		ReservePrivateData	reservePrivateData is a constructor
80	01	0F	Reserv_Timer	the reserve timer is of type integer and value is 15 seconds
Hex Dump = A1 1B 02 01 05 02 01 11 30 14 A0 04 80 02 13 9D A1 04 A0 02 84 00 A2 05 A4 03 80 01 0F				

RETURN-RESULT DMS ===== > HOST

Table 89 dv-Set-Feature for login agent request

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
Hex Dump = A2 03 02 01 03				

RETURN-ERROR DMS ====> HOST

Table 90 dv-Set-Feature for login agent request w/Error Value (Missing/Invalid Parameter)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	01	ErrorType	the ErrorType is MissingParameter[1]
30	03		MissingFeatureParm	MissingFeatureParm is a sequence 3 bytes
81	01	00	MissingLogInParm	MissingLogInParm is MissingLoginId (0)
Hex Dump = A3 0B 02 01 03 02 01 01 30 03 81 01 00				

RETURN-ERROR DMS =====> HOST

Table 91 dv-Set-Feature for login agent request w/Error Value (general format)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	05	ErrorType	this is Agent-Already-Logged-In [5]
Hex Dump = A3 06 02 01 03 02 01 05				

4.8.1.3 ACD LogOut Agent Functionality: coded example

The host requests the switch to log out an ACD agent logged in at position ID 0012 by sending a dv-Set-Feature message with Invoke ID of 3. The switch verifies the request and sends a Return Result that the request will be initiated.

INVOKE DMS < ===== HOST

Table 92 dv-Set-Feature logout agent

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	13		INVOKE	this message is 19 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	11	Operation	it operation value is dv-Set-Feature [17] decimal
30	0B		SetFeatureArgument	the SetFeatureArgument is of type sequence and is 11 bytes long decimal
A0	03		OrigAddress	the OrigAddress is of type constructor
80	01	0C	PositionID	the PositionID is 12
A1	04		SetFeatureInfo	the SetFeatureInfo is of type constructor
A0	02		AgentFeature	the AgentFeature component
81	00		LogOut	the Logout is of type NULL
Hex Dump = A1 13 02 01 03 02 01 11 30 0B A0 03 80 01 0C A1 04 A0 02 81 00				

RETURN-RESULT DMS =====> HOST

Table 93 dv-Set-Feature for a logout agent request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	05	InvokeID	the InvokeID is 5decimal
Hex Dump =A2 03 02 01 05				

RETURN-ERROR DMS=====> HOST

Table 94 dv-Set-Feature for a logout agent with error value (missing parameter)

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	08		RETURN-ERROR	the message length is 08 bytes
02	01	03	InvokeID	its InvokeID is 3
02	01	11	ErrorType	the ErrorType is MissingParameter [17]
30	01	00	InvalidTimerContent	Error is InvalidTimerContent is NULL
Hex Dump = A3 08 02 01 03 02 01 11 30 01 00				

4.8.1.4 ACD Ready Agent Functionality: coded example

The host asks the switch to change the availability status of an ACD agent to Ready so the agent can receive ACD calls. The agent is logged in at position ID 0012. The host sends a dv-Set-Feature message with Invoke ID of 5. After verifying the request, the switch sends a Return Result that the request will be initiated.

INVOKE DMS < ===== HOST

Table 95 dv-Set-Feature ready agent

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	13		INVOKE	this message is 19 bytes decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
02	01	11	Operation	it operation value is dv-Set-Feature [17] decimal
30	0B		SetFeatureArgument	the SetFeatureArgument is of type sequence and is 11 bytes long decimal
A0	03		OrigAddress	the OrigAddress is of type constructor
80	01	0C	PositionID	the PositionID is 12
A1	04		SetFeatureInfo	the SetFeatureInfo is of type constructor and is 4 bytes long decimal
A0	02		AgentFeature	the AgentFeature is of type constructor
82	00		Ready	Ready is of type NULL
Hex Dump = A1 13 02 01 05 02 01 11 30 0B A0 03 80 01 0C A1 04 A0 02 82 00				

RETURN-RESULT DMS =====> HOST

Table 96 dv-Set-Feature for a ready agent request

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
Hex Dump = A2 03 02 01 05				

4.8.1.5 ACD Not-Ready Agent Functionality: coded example

The host asks the switch to change the status an ACD agent to Not Ready, thus removing the agent's availability to receive ACD calls. The agent is logged in at Position ID 0012. The host sends a dv-Set-Feature message with Invoke ID of 5. After verifying the request, the switch sends a Return Result that the request will be initiated.

INVOKE DMS ===== > HOST

Table 97 dv-Set-Feature not ready agent

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1E		INVOKE	this message is 30 bytes decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
02	01	11	Operation	it operation value is dv-Set-Feature [17] decimal
30	16		SetFeatureArgument	the SetFeatureArgument is of type sequence and is 22 bytes long decimal
A0	03		OrigAddress	the OrigAddress is of type constructor
80	01	0C	PositionID	the PositionID is 12
A1	04		SetFeatureInfo	the SetFeatureInfo is of type constructor and is 4 bytes long decimal
A0	02		AgentFeature	the AgentFeature is of type constructor
83	00		NotReady	NotReady is of type NULL
A2	09		PrivateData	PrivateData is of type constructor
A0	07		AgentPrivateData	AgentPrivateData is of type constructor
A3	05		NotReady-Private Data	NotReadyPrivateData is of type constructor
80	03	35 35 35	WalkawayReason	the WalkawayReason is 555
Hex Dump = A1 1E 02 01 05 02 01 11 30 16 A0 03 80 01 0C A1 04 A0 02 83 00 A2 09 A0 07 A3 05 80 03 35 35 35				

RETURN-RESULT DMS =====> HOST

Table 98 dv-Set-Feature for a not ready agent request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
Hex Dump = A2 03 02 01 05				

The host sends a dv-Set-Feature message to perform one of the following on behalf of the agent:

- Instruct the switch to "log in" a specific ACD agent position.
- Instruct the switch to "log out" a specific ACD agent position.
- Request a specific ACD agent position be made available to receive incoming ACD calls. The agent is put in the ready state.
- Request a specific ACD agent position to be taken out of its present state and put in the not ready state. Once the agent is put in the not ready state, the agent will not receive incoming ACD calls.

4.8.1.6 ACD Change Variable Wrap-Up: coded example

The host requests the switch to modify the ACD agent position at Position ID 5021 by sending a dv-Set-Feature message with Invoke ID of 5, and wrapup timer of 20 seconds. The switch verifies the request and sends a Return Result that the request will be initiated.

If the switch receives the dv-Set-Feature message with an invalid timer content, it sends a Return Error message so indicating.

INVOKE DMS <===== HOST

Table 99 change variable wrapup

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	17		INVOKE	this message is 23 bytes decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
02	01	11	Operation	the operation value is dv-Set-Feature [17] decimal
30	0F		SetFeatureArgument	the SetFeatureArgument is of type sequence and is 15 bytes long decimal
A0	04		OrigAddress	the OrigAddress is of type constructor
80	02	13 9D	PositionID	the PositionID is 5021 decimal
A1	07		SetFeatureInfo	the SetFeatureInfo is of type constructor and is 07 bytes long decimal
A0	05		AgentFeature	the AgentFeature is of type constructor
A6	03		CHGVWRAP	the CHGVWRAP is of CHGVWRAPinfo type
80	01	14	WrapTimer	the WrapTimer is of type integer and is set to 20
Hex Dump = A1 17 02 01 05 02 01 11 30 0F A0 04 80 02 13 9D A1 07 A0 05 A6 03 80 01 14				

INVOKE DMS <===== HOST

Table 100 change variable wrapup for CHGVWRAP request with station number (for group)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	19		INVOKE	the message length is 25 bytes decimal
02	01	05	InvokeID	its Invoke ID is 5
02	01	11	Operation	its operation value is dv-Set-Feature [17] decimal
30	14		SetFeatureArgument	SetFeatureArgument is of type sequence and is 17 bytes long
A0	06		OrigAddress	OrigAddress is a constructor of length 60
82	04	23 E9 E2 68	Station Number	the group's directory number is 9192465000 decimall
A1	07		Set-Feature-Info	SetFeatureInformation is a constructor
A0	05		AgentFeature	Set AgentFeature is a constructor
A4	03		CHGVWRAP	CHGVWRAP is of CHGVWRAPinfoType type
80	01	14	WrapTimer	the wrap timer is of type integer and value is 20 seconds
Hex Dump = A1 19 02 01 05 02 01 11 30 11 A0 06 82 04 23 e9 e2 68 A1 07 A0 05 A6 03 80 01 14				

RETURN-RESULT DMS ====> HOST

Table 101 change variable wrapup for a CHGVWRAP request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
Hex Dump = A2 03 02 01 05				

RETURN-ERROR DMS ====> HOST

Table 102 change variable wrapup for CHGVWRAP request with error value (invalid parameter)

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	09		RETURN-ERROR	this message is 9 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	11	ErrorType	the ErrorType is InvalidParameter [17]
30	01	00	InvalidCHGVWRAP	error is InvalidWrapTimer is greater than 600 decimal
Hex Dump = A3 09 02 01 03 02 01 11 30 01 00				

4.8.1.7 ACD Change Call Forcing: coded example

The host requests the switch to modify the ACD agent position logged in with loginid 5021 by sending a dv-Set-Feature message with Invoke ID of 5, and forcing parameter 0 which represents a headset. The switch verifies the request and sends a Return Result that the request will be initiated.

If the switch receives the dv-Set-Feature message with an invalid force type content, it sends a Return Error message so indicating.

INVOKE DMS <===== HOST

Table 103 change call forcing

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	15		INVOKE	this message is 21 bytes decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
02	01	11	Operation	the operation value is dv-Set-Feature [17] decimal
30	13		SetFeatureArgument	the SetFeatureArgument is of type sequence and is 19 bytes long decimal
A0	04		OrigAddress	the OrigAddress is of type constructor
80	02	13 98	PositionID	the PositionID is 5021 decimal
A1	07		SetFeatureInfo	the SetFeatureInfo is of type constructor
A0	05		AgentFeature	the AgentFeature is of type constructor
A7	03		CHGFORCE	the CHGFORCE is of ChngForcinginfo-Type type
80	01	00	Frce_Type	the Frce_Typer is of type Forcing Type with a value of 0 that implies Base
Hex Dump = A1 15 02 01 05 02 01 11 30 13 A0 04 80 02 13 98 A1 07 A0 05 A7 03 80 01 00				

INVOKE DMS <===== HOST

Table 104 CHGFORCE request with station number (for group)

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	19		INVOKE	the message length is 25 bytes decimal
02	01	05	InvokeID	its Invoke ID is 5
02	01	11	Operation	its operation value is dv-Set-Feature [17] decimal
30	11		SetFeatureArgument	SetFeatureArgument is of type sequence and is 17 bytes long
A0	06		OrigAddress	OrigAddress is a constructor
82	04	23 e9 e2 68	Station Number	the group’s directory number is 9192465000 decimall
A1	07		Set-Feature-Info	SetFeatureInformation is a constructor
A0	05		AgentFeature	Set AgentFeature is a constructor
A7	03		CHGFORCE	CHGFORCE is of ChngForcinginfoType type
80	01	00	Frce_Type	the CHGFORCE_Type is of type ForcingType with a value of 0 that implies Base
Hex Dump = A1 19 02 01 05 02 01 11 30 11 A0 06 82 04 23 e9 e2 68 A1 07 A0 05 A7 03 80 01 00				

RETURN-RESULT DMS ====> HOST

Table 105 CHGFORCE request

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	03		RETURN RESULT	this message is 3 bytes long decimal
02	01	05	InvokeID	the InvokeID is 5 decimal
Hex Dump = A2 03 02 01 05				

RETURN-ERROR DMS ====> HOST

Table 106 CHGFORCE request w/error value (invalid parameter)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 8 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	11	ErrorType	the ErrorType is InvalidParameter [20]
Hex Dump = A3 09 02 01 03 02 01 11 30 01 00				

4.8.1.8 dv-Set-Feature parameters

The agent positions need not be in the associated environment with which the host has an established session. A brief description of the parameters of the dv-Set-Feature invoke follows:

OrigAddress is a choice parameter identifying the ACD agent position initiating the request. The only choice for OrigAddress is position ID which uniquely identifies an agent position (not the actual agent) for the customer in a switch.

Set-Feature-Info is a choice parameter identifying the function of the ACD agent request. The AgentFeature is the only choice.

AgentFeature identifies the agent state change requested. It is a Choice parameter with the following choices: LogIn, LogOut, Ready and NotReady, Reserve, Unreserve, CHGVWRAP, and CHGFORCE. The LogIn agent function requires a LoginID and password (optional).

LogIn indicates the switch has requested the LogIn function on behalf of the agent. The following subparameters are then required:

LoginID identifies the ACD agent (person/human representative) to be logged in to the position associated with the OrigAddress (PositionID).

Password contains the password required for agent login security. This requirement is optional in the DMS Meridian ACD applications.

PrivateData contains private data specific to the requested agent function. The NotReady agent function has an ACD group-based option to accept a walkaway reason code. The Reserve agent function specifies the reserv_Timer parameter.

WalkawayReason indicates the reason the agent wants to be taken out of the current state into a not ready state.

reserv_Timer allows the host to specify the time needed to reserve the agent. If the parameter and reserve function are not used at the same time, the agent reserves for 15-seconds. The agent can reserve a maximum of 30-seconds.

FrceType indicates the call forcing tone destination as 0 (base), 1 (headset), or 2 (none).

WrapTimer indicates the variable wrapup timer as a range of 0 through 600 seconds.

4.8.1.9 dv-Set-Feature validations

On receiving a dv-Set-Feature message, the switch:

1 validates the message by verifying that:

a. the switch-host session subscribes to the dv-Set-Feature RO as well as to one of its functionalities (LogIn, LogOut, Ready, NotReady. reserve, unreserve, CHGFORCE, or CHGVWRAP).

b. the mandatory parameters are subscribed to and are present in the message.

c. the required subparameters for the respective functionality are subscribed to and are in the message.

2 validates the requested function

a. LogIn function

For a specific agent position, the following is verified:

i. the PositionID specified is valid for the specific customer group for the active Meridian ACD CompuCALL session

ii. the LoginID parameter is valid

iii. the LoginID specified is not in use by another agent position within the same customer group

iv. the password parameter is valid (if subscribed to and present in the message)

v. the specified ACD agent position is not logged in

vi. the specified agent position is not involved in a call. The ACD line must be in idle state.

vii. the given password (if required) is a valid one for the LoginId

viii. the LoginID specified is valid for the specific customer group for the active Meridian ACD CompuCALL session

ix. if the agent position is an Meridian Business Set, it is verified that the Make Set Busy (MSB) feature is available.

Note: The ACD agent position is put in the Not Ready state automatically upon login, if the Not Ready agent feature is available. Otherwise, the agent position is put in the Ready state immediately after Login.

b. LogOut function

For a specific agent position, the following is verified:

i. the PositionID specified is valid for the specific customer group for the active Meridian ACD CompuCALL session

ii. the specified ACD agent position is logged in. (The ACD agent position is considered logged in only once the entire LogIn process has successfully completed.)

iii. the specified ACD agent position is not in a transient state due to a feature interaction

iv. the specified ACD agent position is not in a pending logged out state.

c. Ready function

i. For a specific agent position, the following is verified:

ii. the PositionID specified is valid for the specific customer group for the active Meridian ACD CompuCALL session

iii. the specified ACD agent position is either idle or active on a call

iv. the specified ACD agent position is logged in

v. the specified ACD agent position has the ACD Not Ready (ACDNR) feature available

vi. the specified ACD agent position is not in the process of executing the Walkaway function (applicable to MBS set only)

vii. the specified ACD agent position is not in a transient state due to a feature interaction

viii.the specified ACD agent position is in one of the following states: Not Ready, pending Not Ready, or Variable Wrap-up state.

d. Not Ready function

For a specific agent position, the following is verified:

i. the positionID specified is valid for the specific customer group for the active Meridian ACD CompuCALL session

ii. the WalkawayReason parameter (if it is subscribed to and included in the message) is valid

iii. the specified ACD agent position is logged in

iv. the specified ACD agent position has the ACD Not Ready (ACDNR) feature available

v. the specified ACD agent position is not in the process of executing the Walkaway function (applicable to MBS set only)

vi. the specified ACD agent position is not in a transient state due to a feature interaction

vii. the specified ACD agent position is in one of the following states: Ready or Variable Wrap-up state.

Note: The equivalent of the Non-Immediate Cut-off (NONIMCUT) feature functionality is provided upon Not Ready activation through the application service function, dv-Set-Feature (NotReady).

4.8.1.10 dv-Set-Feature error conditions

The switch can respond to the dv-Set-Feature message with one of the following messages: Return Result (RR), Return Error (RE) or Reject (REJ). The section below expands on the conditions for each of the possible responses.

A switch response of Return Result (RR) indicates the dv-Set-Feature message has been successfully verified. An RR sent to the host with no additional information (no parameters) to indicate that the request is valid and will be initiated.

A switch response of Return Error (RE) indicates the dv-Set-Feature message has not been successfully verified. An RE is sent to the host with an appropriate error cause value. The ACD agent position remains in the same state prior to the request. The possible error conditions are described below with respect to the RO and the functions (LogIn, LogOut, Ready, Not Ready).

Note: Only the first error encountered will be reported in the RE response.

Errors specific to the dv-Set-Feature RO:

1 The dv-Set-Feature message is not subscribed for the Meridian ACD CompuCALL session (Not-Allowed)

2 Missing Parameter (identified)

3 Invalid parameter (identified)

4 Invalid Origination Address (Invalid-Parameter-Content with value OrigAddress)

Errors specific to the LogIn Agent functionality:

1 the logIn agent functionality is not subscribed for the Meridian ACD CompuCALL session (Not-Allowed)

2 invalid LoginID (Invalid-Parameter-Content with value LoginID)

3 invalid password (Invalid-Parameter-Content with value password)

4 LogIn restrictions or invalid agent position (Invalid-Line-Configuration)

5 agent position is already logged on with the given LoginID (Agent-Already-Logged-in)

6 LoginID is already in use with another PositionID (Login-ID-In-Use-Elsewhere)

7 the agent position is already logged in under a different LoginID (Position-Already-Logged-In)

8 the agent position is in invalid line state (Invalid-Set-State)

9 invalid password (Password-Mismatch)

10 the agent position does not have the Make Set Busy feature available (MBS sets only) (Invalid-Line-Configuration)

11 resources unavailable (Resources Unavailable)

Errors specific to the LogOut Agent functionality:

1 the LogOut agent functionality is not subscribed to for the Meridian ACD CompuCALL session (Not-Allowed)

2 the agent position is not logged on (Agent-Not-Logged-In)

3 the agent position is in a pending logged out state (Agent-Logout-Pending)

4 the agent position is in a transient state i.e., feature interaction (Invalid-AgtPos-State)

Errors specific to the Ready Agent functionality:

1 the Ready Agent functionality is not subscribed to for the Meridian ACD CompuCALL session (Not-Allowed)

2 the agent position is not logged on (Agent-Not-Logged-In)

3 the agent position is in a pending logged out state (Agent-Logout-Pending)

4 the agent position is involved with another process or in a transient state i.e., feature interaction e.g., agent entering a walkaway reason code (Invalid-AgtPos-State)

5 the agent position's line state is invalid (Invalid-Set-State)

6 the agent position does not have the ACD Not Ready agent feature available (Invalid-Line-Configuration)

7 the agent position is currently in the Ready state (Agent-Presently-Ready)

Errors specific to the NotReady Agent functionality

1 the NotReady Agent functionality is not subscribed to for the Meridian ACD CompuCALL session (Not-Allowed)

2 the agent position is not logged on (Agent-Not-Logged-In)

3 the agent position is in a pending logged out state (Agent-Logout-Pending)

4 the agent position is involved with another process or in a transient state i.e., feature interaction e.g., agent entering a walkaway reason code (Invalid-AgtPos-State)

5 the agent position's line state is invalid (Invalid-Set-State)

6 the agent position does not have the ACD Not Ready agent feature available (Invalid-Line-Configuration)

7 the walkaway reason entered is invalid (Invalid-Parameter-Content with value WalkawayReason)

8 the agent position is currently in the NotReady state (Agent-Presently-Not-Ready)

9 the agent's supervisor has enforced the Forced Agent Availability (FAA) on the agent position, overriding the NotReady activation (Supervisor-Override)

Errors specific to the Change Call Forcing functionality

1 the ACD group does not have the forcing option assigned when the CHGFORCE option is selected for an ACD group or agent (forcing Presently Inactive)

Errors specific to the Change Variable Wrapup functionality

1 the ACD group does not have the variable wrapup option assigned when the CHGVWRAP option is selected for an ACD group or agent (wrap Presently Inactive)

A switch response of Reject (REJ) indicates the dv-Set-Feature message cannot be understood or interpreted by the switch. An REJ is sent to the host with no additional information (no parameters).

4.8.2 dv-Agent-SetAction-U (operation value “2C” hex)

The agent is unreserve when the agent_reservation timer expires. The switch uses the dv-Agent-SetAction-U message to notify the host computer about unreserving the agent.

4.8.2.1 dv-Agent-SetAction-U parameters

The dv-Agent-SetAction-U message includes the following parameters:

AgtPosID contains the Position ID of the ACD agent. If the agent does not have a position ID then the switch sends the message to the host computer.

NetwStAct indicates to the host computer that the switch unreserved the agent.

IntervalStatus indicates the interval status

ENHWAC indicates the walkaway reason enterted via the enhanced walkaway functionality.

4.8.2.2 dv-Agent-SetAction-U error conditions

The following error conditions apply to dv-Agent-SetAction-U message

1 The service version is not SCAI 12 or more.

Note: Previous service versions will pass dv-Call-Released-u, but without the postcall status parameter of the message.

2 The CompuCALL session does not subscribe to the ACDEVENT category and the AGTSETACT function.

3 The switch-to-host session does not subscribe to the message.

4.8.2.3 dv-Agent-SetAction-U: ASN.1 encoding

```
    dv-Agent-SetAction-U        OPERATION
                                ARGUMENT      SetAction Argument
::= 45

SetAction Argument              :: = SEQUENCE

{   agtPosID                    [0] IMPLICIT INTEGER         MANDATORY,

    NtwSetAct                   [1] IMPLICIT ENUMERATED
    {     unreserved                  (0)                    }
                                                             OPTIONAL,

    IntervalStatus              [2] IMPLICIT ENUMERATED
    {     intervalEnded               (0)                    }
                                                             OPTIONAL
                                                                     }

EnhWalkaway                   [3] IMPLICIT IA5STRING

                                                           OPTIONAL

                                                                     }

            Q.931AddressType:: =SEQUENCE
           {    numberTypeNumberPlan[0] OCTET STRINGOPTIONAL,
                presentationScreeningIndicator[1] OCTET STRINGOPTIONAL,
                digits                  [2] IA5 STRING       OPTIONAL}
```

4.8.2.4 dv-Agent-SetAction-U: coded example

The following is a coded example of the dv-Agent-SetAction-U message, when the timer expiers. An agent with posid 5021 is reserved, but is not offered a call untill the timer expires. Hence the agent is now unreserved and host is informed about unreservation. The invoke ID is 3. This example has the message delivered when the SCAI version is above SCAI 12 and the parameter is subscribed through table control.

INVOKE DMS ====> HOST

Table 107 dv-Agent-SetAction-U coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0F		INVOKE	the message length is 18 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	2D	Operation	its operation value is dv-Agent-SetAction-U [44] decimal
30	07		Argument	SetAction Argument is of type choice and is 10 bytes long decimal
80	02	13 9D	AgtPosID	the posid of the agent 5021
81	01	00	. . . . NtwSetAct	Indicates that the agent is unreserved.
82	01	00	IntervalStatus	the status is intervalEnded
83	03	34 35 36	...EnhWalkaway	indicates that the new walkaway reason entered is 456.
Hex dump = A1 11 02 01 03 02 01 2C 30 09 80 02 13 9D 83 03 34 35 36				

4.8.3 ACD dv-Agent-Logged-In-U (operation value "12" hex)

The switch sends the RO of Operation Class 5 dv-Agent-Logged-In-U message to the host to indicate completion of the requested agent logIn function.

ASN.1 Encoding

```
                dv-Agent-Logged-In-UOPERATION
                ARGUMENTAgtLoggedInArg
    ::= 18

    AgtLoggedInArg          ::=     SEQUENCE
    {   acdGroup        [0] AddressTypeOPTIONAL,
        posID           [1]  AddressTypeOPTIONAL,
        loginID         [2] IMPLICIT IA5STRINGOPTIONAL}

    AddressType             :: =    CHOICE
    {   positionID      [0] IMPLICIT INTEGER,
        dialedDigits    [1] IMPLICIT IA5STRING,
        stationNumber[2] IMPLICIT OCTET STRING,
    q931Address         [3] Q931AddressType}

    Q931AddressType         :: =    SEQUENCE
    {numberTypeNumberPlan           [0] OCTET STRING OPTIONAL,
    presentationScreeningIndicator  [1] OCTET STRINGOPTIONAL,
    digits                          [2] IA5StringOPTIONAL}
```

4.8.3.1 ACD dv-Agent-Logged-In-U: coded example

The switch has completed the LogIn process for an ACD agent position with a position ID 0002 logged in with a login ID 2345. The ACD group DN is 613-555-2248. Hence, the switch sends a dv-Agent-Logged-In-U message to the host with an Invoke ID of 3.

INVOKE DMS < ===== HOST

Table 108 dv-Agent-Logged-In-U coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	21		INVOKE	this message is 33 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	12	Operation	it operation value is dv-Agent-Logged-In-U [18] decimal
30	19		AgtLoggedInArg	AgtLoggedInArg is of type sequence and is 25 bytes long decimal
A0	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long decimal
82	0A	36 31 33 35 35 35 32 32 34 38	StationNumber	the PrimaryDN of the ACDGroup is (613)555-2248
A1	03		PosID	the agent's PosID is of type constructor and is 3 bytes long decimal
81	01	02	PositionId	the agent's PositionId is 2 decimal
82	04	32 33 34 35	LoginID	the LoginId is 2345
Hex Dump = A1 21 02 01 03 02 01 12 30 19 A0 0C 82 0A 36 31 33 35 35 35 32 32 34 38 A1 03 81 01 02 82 04 32 33 34 35				

4.8.3.2 dv-Agent-Logged-In-U parameters

This message contains the following parameters:

ACDGroup identifies the primary ACD DN of the ACD group in the CompuCALL environment to which an ACD agent position belongs. The choice is StationNumber.

PosID identifies the specific customer workstation to which the request is made. The only choice is PositionID.

LoginID contains the identity of the ACD agent used to associate the agent (person/human representative) with the agent position.

4.8.4 ACD dv-Agent-Logged-Out-U (operation value "14" hex)

The switch sends the this RO of Operation Class 5 dv-Agent-Logged-Out-U message to the host to indicate completion of the requested agent LogOut function.

```
ASN.1 Encoding
    dv-Agent-Logged-Out-U OPERATION
        ARGUMENTAgtLoggedOutArg
    ::= 20

    AgtLoggedOutArg::=SEQUENCE
    {   acdGroup[0]  AddressType          OPTIONAL,
        posID[1]  AddressTypeOPTIONAL}

    AddressType:: =CHOICE
    {   positionID[0] IMPLICIT INTEGER,
        dialedDigits[1] IMPLICIT IA5STRING,
        stationNumber[2] IMPLICIT OCTET STRING,
    q931Address[3] Q931AddressType}

    Q931AddressType:: = SEQUENCE
    {numberTypeNumberPlan[0] OCTET STRING          OPTIONAL,
    presentationScreeningIndicator [1] OCTET STRING           OPTIONAL,
    digits   [2] IA5StringOPTIONAL}
```

4.8.4.1 dv-Agent-Logged-In-U parameters

This message contains the following parameters:

ACDGroup identifies the primary ACD DN of the ACD group in the CompuCALL environment to which an ACD agent position belongs. The choice is StationNumber.

PosID identifies the specific customer workstation to which the request is made. The only choice in is PositionID.

4.8.4.2 ACD dv-Agent-Logged-Out-U: coded example

The switch has completed the LogOut process for an ACD agent position with a position ID 0002. The ACD group DN is 613-555-2248. Hence, the switch sends a dv-Agent-Logged-Out-U message to the host with an Invoke ID of 3.

INVOKE DMS ===== > HOST

Table 109 dv-Agent-Logged-Out-U coded example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	1B		INVOKE	this message is 27 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	14	Operation	it operation value is dv-Agent-Logged-Out-U [20] decimal
30	13		AgtLoggedOutArg	AgtLoggedOutArg is of type sequence and is 19 bytes long decimal
A0	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long decimal
82	0A	36 31 33 35 35 35 32 32 34 38	StationNumber	the PrimaryDN of the ACDGroup is (613)555-2248
A1	03		PosID	the agent’s PosID is of type constructor and is 3 bytes long decimal
81	01	02	PositionId	the agent’s PositionId is 2 decimal
Hex Dump = A1 1B 02 01 03 02 01 14 30 13 A0 0C 82 0A 36 31 33 35 35 35 32 32 34 38 A1 03 80 01 02				

4.8.5 ACD dv-Agent-Ready-U (operation value "15" hex)

The switch sends the RO of Operation Class 5 dv-Agent-Ready-U message to the host to indicate completion of the requested Agent Ready function. Hence, the ACD agent position will have entered the Ready state and would be available to receive incoming ACD calls. If the Ready/Not Ready functional feature is not available to an agent position for manual initiation, the switch still sends this message to the host to indicate the agent state transition.

4.8.5.1dv-Agent-Ready-U ASN.1 encoding

```
dv-Agent-Ready-U            OPERATION
                            ARGUMENT          AgtReadyArg
::= 21

AgtReadyArg                  ::=      SEQUENCE
     {  posID                [0]  AddressType                    OPTIONAL,
        privateData          [1]  IMPLICIT  ReadyPrivateData     OPTIONAL}

ReadyPrivateData             ::=            SEQUENCE
     {  acdGroup             [0]  AddressType                    OPTIONAL}

AddressType                  :: =           CHOICE
 {  positionID               [0] IMPLICIT INTEGER,
    dialedDigits             [1] IMPLICIT IA5STRING,
    stationNumber            [2] IMPLICIT OCTET STRING,
    q931Address              [3] Q931AddressType}

Q931AddressType              :: =           SEQUENCE
{numberTypeNumberPlan        [0] OCTET STRING                    OPTIONAL,
  presentationScreeningIndicator [1] OCTET STRING                OPTIONAL,
  digits                     [2] IA5String                       OPTIONAL}
```

4.8.5.2 ACD dv-Agent-Ready-U: coded example

The switch has completed the Ready Agent process for an ACD agent position with a position ID of 0002. The ACD group DN is 613-555-2248. Hence, the switch sends a dv-Agent-Ready-U message to the host with an invoke ID of 3.

INVOKE DMS ===== > HOST

Table 110 dv-Agent-Ready-U coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1D		INVOKE	this message is 29 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	15	Operation	it operation value is dv-Agent-Ready-U [21] decimal
30	15		AgtReadyArg	AgtReadyArg is of type sequence and is 21 bytes long decimal
A0	03		PosID	the agent's PosID is of type constructor and is 3 bytes long decimal
80	01	02	PositionId	the agent's PositionId is 2 decimal
A1	0C		PrivateData	the PrivateData is of type constructor
A0	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long decimal
82	0A	36 31 33 35 35 35 32 32 34 38	StationNumber	the PrimaryDN of the ACDGroup is (613)555-2248
Hex Dump = A1 1D 02 01 03 02 01 15 30 15 A0 03 80 01 02 A1 0C A0 0C 82 0A 36 31 33 35 35 35 32 32 34 38				

4.8.5.3 dv-Agent-Ready-U parameters

This message contains the following parameters:

PosID identifies the specific customer workstation to which the request is made. The choice is PositionID.

PrivateData contains private data specific to the requested agent function. The Ready agent includes ReadyPrivateData information consisting of ACD group information.

ACDGroup identifies the Primary ACD DN of the ACD group in the CompuCALL environment to which an ACD agent position belongs. The choice is StationNumber.

4.8.6 ACD dv-Agent-Not-Ready-U (operation value "16" hex)

The switch sends the RO of Operation Class 5 dv-Agent-Not-Ready-U message to the host to indicate completion of the requested Agent Not Ready. Hence, the ACD agent position will have entered the a Not Ready state and is no longer available to receive incoming ACD calls. In the case of the Not Ready functional feature being requested while the agent is involved in a call, this message is sent after the call has been released by the agent. It is at this point in call that the agent enters the Not Ready state.

ASN.1 Encoding

```
dv-Agent-Not-Ready-UOPERATION
    ARGUMENTAgtNotReadyArg
::= 21

AgtNotReadyArg::=SEQUENCE
{   posID[0]  AddressTypeOPTIONAL,
    privateData[1]  IMPLICIT  NotReadyPrivateData
                    OPTIONAL}

NotReadyPrivateData::=SEQUENCE
{   acdGroup[0]  AddressType OPTIONAL,
    walkawayReason[1]  IMPLICIT IA5STRING OPTIONAL}

AddressType:: =CHOICE
{   positionID[0] IMPLICIT INTEGER,
    dialedDigits[1] IMPLICIT IA5STRING,
    stationNumber[2] IMPLICIT OCTET STRING,
q931Address[3] Q931AddressType}

Q931AddressType:: = SEQUENCE
{numberTypeNumberPlan [0] OCTET STRINGOPTIONAL,
presentationScreeningIndicator [1] OCTET STRING OPTIONAL,
digits   [2] IA5StringOPTIONAL}
```

4.8.6.1 ACD dv-Agent-Not-Ready-U: coded example

The switch has completed the NotReady agent process for an ACD agent position with a Position ID 0002 with walkaway reason 123. The ACD group DN is 613-555-2248. Hence, the switch sends a dv-Agent-Ready-U message to the host with an Invoke ID of 3.

INVOKE DMS ===== > HOST

Table 111 dv-Agent-Not-Ready-U coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	22		INVOKE	this message is 34 bytes decimal
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	16	Operation	it operation value is dv-Agent-Ready-U [22] decimal
30	1A		AgtReadyArg	AgtReadyArg is of type sequence and is 26 bytes long decimal
A0	03		PosID	the agent’s PosID is of type constructor and is 3 bytes long decimal
80	01	02	PositionId	the agent’s PositionId is 2 decimal
A1	13		PrivateData	the PrivateData is of type constructor and is 19 bytes long decimal
A0	0C		ACDGroup	the ACDGroup is of type constructor and is 12 bytes long decimal
82	0A	36 31 33 35 35 35 32 32 34 38	StationNumber	the PrimaryDN of the ACDGroup is (613)555-2248
81	03	31 32 33	WalkawayReason	the WalkawayReason is 123
Hex Dump = A1 22 02 01 03 02 01 16 30 1A A0 03 80 01 02 A1 13 A0 0C 82 0A 36 31 33 35 35 35 32 32 34 38 81 03 31 32 33				

4.8.6.2 dv-Agent-Not-Ready-U parameters

This message contains the following parameters:

PosID identifies the specific customer workstation to which the request is made. The only choice is PositionID.

PrivateData contains private data specific to the requested agent function. The Not Ready agent function includes NotReadyPrivateData information consisting of ACD group and the walkaway reason.

ACDGroup identifies the primary ACD DN of the ACD group in the CompuCALL environment to which an ACD agent position belongs. The choice is StationNumber.

WalkawayReason indicates the reason the agent position is taken out of the current state and into a not ready state.

4.9 ACD resource status

4.9.1 ACD dv-Resource-Query (operation value "13" hex)

The host uses the RO of Operation Class 2 dv-Resource-Query message to ask the switch for information on a specific nodal ACD group. The switch responds with information on ACD group status, ACD agent status summary, ACD queue parameters defined in the DMS and ACD queue status. This message is valid for Meridian ACD CompuCALL options only.

```
ASN.1 Encoding
    dv-Resource-Query OPERATION
                                  ARGUMENT      ResourceQueryArgument,
         ERRORS          {not-Allowed,
                                                  missing-Parameter,
                                                  invalid-Parameter,
     data-Temporarily-Unavailable}
::=19

ResourceQueryArgument::=SEQUENCE
{queryAddress[1] AddressType                              OPTIONAL}

    AddressType:: =  CHOICE
    {positionID[0] IMPLICIT INTEGER,
     dialedDigits[1] IMPLICIT IA5STRING,
     stationNumber[2] IMPLICIT OCTET STRING
     q931Address[3] Q931AddressType}

    Q931AddressType::=  SEQUENCE
    {numberTypeNumberPlan              [0] OCTET STRING
OPTIONAL,
    presentationScreeningIndicator      [1] OCTET STRING
OPTIONAL,
    digits                             [2] IA5String
OPTIONAL}

ResourceQueryResult::= CHOICE
{ACDDNQuery[0] ACDDNResult     OPTIONAL}

ACDDNResult::= IMPLICIT SEQUENCE
{   time[0] SwitchTimeOfDayOPTIONAL,
    queryAddress[1] AddressTypeOPTIONAL,
    grpPrimDN[2] AddressTypeOPTIONAL,
    grpStat[3] IMPLICIT ENUMERATED
    {acceptingCalls(0),
    overflow(1),
    nightService(2),
     controlledInterFlow(3),
     statusReroute(4)}OPTIONAL,
queryInfo[4] QueryTypesOPTIONAL}


SwitchTimeOfDay::= IMPLICIT SEQUENCE
{    hours[0]IMPLICIT INTEGER  (0..23) OPTIONAL,
     minutes[1]IMPLICIT INTEGER  (0..59)OPTIONAL,
     seconds[2]IMPLICIT INTEGER  (0..59)OPTIONAL}

QueryTypes::=IMPLICIT SEQUENCE
{    grpInfo[0] GrpInfoTypeOPTIONAL,
     qInfo[1] QInfoTypeOPTIONAL,
     agtInfo[2] AgtInfoTypeOPTIONAL}

GrpInfoType::=IMPLICIT SEQUENCE
{    maxPQSize[0] IMPLICIT INTEGER  OPTIONAL,
     maxCTQSize[1] IMPLICIT INTEGER  OPTIONAL}

QInfoType::=IMPLICIT SEQUENCE
{    acdDNPrio[0] IMPLICIT INTEGER   (0..3)OPTIONAL,
     pQSize[1] IMPLICIT INTEGER OPTIONAL,
     cTQSize[2] IMPLICIT INTEGER OPTIONAL,
     outQSize[3] IMPLICIT INTEGER  OPTIONAL,
     queueInfo[4] IMPLICIT SEQUENCE OF
     QueuesByPriorityOPTIONAL}

QueuesByPriority::=IMPLICIT SEQUENCE
{    priority[0] IMPLICIT INTEGER   (0..3)OPTIONAL,
     pQatThisPrio[1]IMPLICIT INTEGER  OPTIONAL}

AgtInfoType::=IMPLICIT SEQUENCE
{    agtsLgd[0] IMPLICIT INTEGER OPTIONAL,
     agtsBsy[1] IMPLICIT INTEGER OPTIONAL,
     agtsIdle[2] IMPLICIT INTEGER OPTIONAL,
     agtsNR[3] IMPLICIT INTEGER OPTIONAL,
     agtsCTQ[4] IMPLICIT INTEGER OPTIONAL}

not-Allowed                ERROR    ::=  0

missing-Parameter      ERROR           ::=  1
                 PARAMETER                  MissingParameter

missing-Parameter  IMPLICIT SEQUENCE
{MissingParameterType::=  [0]  IMPLICIT ENUMERATED
     {missingACDDN[0]}OPTIONAL}
```

```
invalid-Parameter              ERROR         ::=  2
                    PARAMETER                    InvalidParameterinvalid-
ParameterIMPLICIT SEQUENCE
{InvalidParameterType::=  [0]        IMPLICIT ENUMERATED
     {invalidACDDN[0]}OPTIONAL}
data-Temporarily-Unavailable  ERROR             ::=  3
```

4.9.1.1 ACD dv-Resource-Query: coded example

The host asks the DMS for information on an ACD group with ACD DN = 613 555 2248 (ten digits). The switch verifies the request and sends a Return Result with the information to the host. If the request can't be fulfilled, the switch sends a Return Error indicating the error.

INVOKE DMS ===== > HOST

Table 112 dv-Resource-Query invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	16		INVOKE	this message is 22 bytes decimal
02	01	01	InvokeID	the InvokeID is 1 decimal
02	01	13	Operation	it operation value is dv-Resource-Query [19] decimal
30	0E		Argument	the ResourceQueryArgument is of type Choice and is 14 bytes long decimal
A1	0C		QueryAddress	the ACDDN for which information is requested
82	0A	36 31 33 35 35 35 32 32 34 38	StationNumber	the ACDDN is (613)555-2248
Hex Dump = A1 16 02 01 01 02 01 13 30 0E A1 0C 82 0A 36 31 33 35 35 35 32 32 34 38				

RETURN-RESULT DMS =====> HOST

Table 113 dv-Resource-Query return result coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	79		RETURN RESULT	this message is 121 bytes long decimal
02	01	01	InvokeID	the InvokeID is 1 decimal
30	74		ReturnResultSequence	this is a sequence 116 bytes long decimal
02	01	13	Operation	its operation is dv-Resource-Query[19] decimal
30	6F		ResourceQueryResult	the ResourceQueryResult is of type Choice 111 bytes long decimal
A0	09		ACDDNQuery	this is of type ACDDNResult and is 109 bytes long
A0	6D		Time	Time is of type SwitchTimeOfDay
80	01	17	Hours	the time is 2300 hours
81	01	31	Minutes	49 minutes
81	01	07	Seconds	and 7 seconds
A1	0C		QueryAddress	QueryAddress is a constructor
82	0A	36 31 33 35 35 35 32 32 34 38	StationNumber	the ACDDN is (613)555-2248
A2	0C		GrpPrimDN	the Primary DN of the ACD group
82	0A	36 31 33 35 35 35 32 32 32 32	StationNumber	the Primary DN is (613)555-2222
83	01	00	GrpStat	the ACD group status is "Accepting Calls"
A4	41		QueryInfo	QueryInfo is of type QueryType
A0	06		GrpInfo	GrpInfo is of type GrpInfoType
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
80	02	00 9D	MaxPQSize	maximum incoming call queue size assigned for ACD group is 157 slots decimal
81	01	20	MaxCTQSize	maximum call transfer queue size assigned for ACD group is 32 slots decimal
A1	26		QInfo	QInfo is of type QInfoType
80	01	01	ACDDNPrio	the priority of the ACDDN is 1 decimal
81	01	1B	PQSize	the incoming call queue is 27 decimal
82	01	05	CTQSize	the call transfer queue is 5 decimal
83	01	02	OutQSize	the overflow out queue is 2 decimal
A4	18		QueueInfo	QueueInfo is sequence of QueuesByPriority
80	01	00	Priority	the priority of this queue is 0 decimal
81	01	03	PQatThisPriority	this queue is 3 decimal
80	01	01	Priority	the priority of this queue is 1 decimal
81	01	00	PQatThisPriority	this queue is 0 decimal
80	01	02	Priority	the priority of this queue is 2 decimal
81	01	05	PQatThisPriority	this queue is 5 decimal
80	01	03	Priority	the priority of this queue is 3 decimal
81	01	0C	PQatThisPriority	this queue is 12 decimal
A2	0F		AgtInfo	AgtInfo is of type AgtInfoType
80	01	08	AgtsLgd	8 (decimal) Agents are Logged in
81	01	03	AgtsBsy	83(decimal) Agents are active on calls
82	01	00	AgtsIdle	0 (decimal) Agents are available to service calls
83	01	05	AgtsNR	5 (decimal) Agents are in NotReady state
84	01	01	AgtsCTQ	1 (decimal) Agent with transferred calls is waiting in queue
			—continued—	

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
Hex Dump = A2 79 02 01 01 30 74 02 01 13 30 6F A0 6D A0 09 80 01 23 81 01 49 82 01 07 A1 0C 82 0A 36 31 33 35 35 35 32 32 34 38 A2 0C 82 0A 36 31 33 35 35 35 32 32 32 32 83 01 00 A4 41 A0 06 80 02 00 9D 81 01 20 A1 26 80 01 01 81 01 1B 82 01 05 83 01 02 A4 18 80 01 00 81 01 03 80 01 01 81 01 00 80 01 02 81 01 05 80 01 03 81 01 0C A2 0F 80 01 08 81 01 03 82 01 00 83 01 05 84 01 01				
			—end—	

RETURN-ERROR DMS =====> HOST

Table 114 dv-Resource-Query return error coded example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	01	InvokeID	the InvokeID is 1 decimal
02	01	05	ErrorValue	the ErrorValue is MissingParameter [1]
30	03		ReturnErrorSequence	MissingParameter is a sequence 3 bytes
80	01	00	ErrorParameter	MissingParameter is MissingACDDN(0)
Hex Dump = A3 0B 02 01 01 02 01 01 30 03 80 01 00				

4.9.1.2 dv-Resource-Query validations

On receiving a dv-Resource-Query message, the switch validates the message received by verifying

1 The ACDDN is a primary or supplementary ACDDN assigned in the switch.

2 The ACDDN is valid for the active Meridian ACD CompuCALL session.

3 The host session has subscribed to the service function.

If the request for the information is valid, the swith sends a Return Result with information for the ACD group.

4.9.1.3 dv-Resource-Query parameters

Time contains the switch time in a twenty-four hour format of hour, minutes, and seconds.

QueryAddress contains the primary or supplementary ACD DN for which the host is requesting information. The choice is StationNumber. The value of this parameter is the same as that in the dv-Resource-Query invoke message.

StationNumber identifies the ACD DN for which information is requested

GrpPrimDN contains the primary ACD DN of the ACD group associated with the ACDDN for which information is requested.

StationNumber identifies the primary ACD DN for which information is requested.

GrpStat specifies the current operational status of the ACD group. Possible ACD group states follow. The ACD group state may change instantaneously with the flow of incoming call traffic and call handling events.

Accepting Calls the ACD group is in service. Incoming calls are being serviced or queued.

Overflow the ACD group is servicing calls queued in the incoming-call queue. New calls may be rerouted to a customer-defined destination (another ACD group), an announcement, or a non-ACD DN.

Night Service the ACD group isn't servicing calls. The calls maybe rerouted to a night service route, if assigned, or receive an announcement. Night service takes effect when all agents have logged off or have been placed in this state by the supervisor (on the ACD supervisor set).

Controlled InterFlow the ACD group is only servicing calls in the incoming-call queue. New calls are not queued for this ACD group but are rerouted to a controlled interflow route. The ACD group is placed in this state by activation of the Controlled Interflow feature by the ACD supervisor.

Status Reroute the ACD group is not servicing or queuing calls. New calls are rerouted to the customer-specified overflow route. Calls in the Incoming-call queue for this ACD group before the group entering this state remain in queue until answered or treated for exceeding the customer-specified threshold. An ACD group enters this state due to an equipment fault affecting the agents' telephone lines of this ACD group.

GrpInfo contains assigned values for the ACD group regarding the maximum incoming call queue size and call transfer queue size. These subparameters are briefly described below:

MaxPQSize indicates the customer-defined value for the maximum number of calls which may be queued in the incoming-call queue.

MaxCTQSize indicates the customer-defined value for the maximum number of calls which may be in the call-transfer queue.

QInfo contains current queue status information for this ACD group. Below is a list of subparameters:

ACDDNPrio specifies the priority associated with the ACDDN for which information was requested.

PQSize specifies the number of queue slots occupied by calls in the ACD group's incoming-call queue. This value also includes calls queued in the call transfer and overflow out queues.

CTQSize specifies the number of queue slots occupied by calls in the ACD group's call transfer call queue (CTQ).

OutQSize specifies the number of queue slots occupied by calls in the ACD group's overflow out queue.

QueueInfo gives queue information on the number of calls queued in the priority queues. The information is in the priority of the queue and the number of call queued at this priority. The subparameters follow:

Priority specifies the priority of a priority queue.

PQatThisPrio specifies the number of queue slots occupied by calls in the Priority queue.

AgtInfoGives information for all the agents assigned for the ACD group associated with the ACD DN specified. The subparameters follow:

AgtsLgd specifies the number of agents currently logged in.

AgtsBsy specifies the number of agents currently active on calls.

AgtsIdle specifies the number of agents currently waiting for calls.

That this value should be zero if queue size for the incoming-call queue (PQSize above) is non-zero.

AgtsNR specifies the number of agents in NotReady state. These agents may be involved with extra work due to handling of the call (wrap-up) or have temporarily left their station (walkaway).

AgtsCTQ specifies the number of agents who currently have calls queued for them in the call transfer queue. Agents' service calls from the call transfer queue before servicing calls in the incoming call queue for the ACD group.

4.9.1.4 dv-Resource-Query error conditions

If an error occurs when the host sends a dv-Resource-Query message, the switch responds with a Return Error message to the host with an error cause value (shown below in parentheses). Possible error conditions are:

1 The function message is not subscribed to (Not Allowed).

2 Any of the required parameters are missing.

3 Any of the required parameters are invalid.

4 The information is temporarily unavailable thus the request cannot be fulfilled (Data-Temporarily-Unavailable).

Note: This list is presented in order of priority since only the first error condition (including the first missing/invalid parameter) encountered by the switch is included in the Return Error message.

The switch does not retransmit if it receives a Reject message in response to a Return Result or Return Error message.

Upon a protocol violation error the switch will send a Reject message to the host.

4.9.2 dv-Appl-Stat-Qry (operation value “27” hex)

The host uses the RO of Operation Class 2 dv-Appl-Stat-Qry message to request from the switch status of the devices associated to the session. Associated devices are ACD agents and CDNs. The host sends the message after a link to the host computer fails.

Once the switch receives the message, it decodes the dv-Appl-Stat-Qry message and responds with the following messages:

- Switch report agent status provides ACD agent information
- Switch report CDN status provides the status of CDNs

4.9.2.1 dv-Appl-Stat-Qry parameters

One of the following parameters is mandatory for this message:

DN contains the primary ACD DN or the CDN or PositionID of an agent for which the host application is requesting status.

Query Filter (Allagents | AllCDNs | AllDevices) is a filter that allows the host to select the required information. The following three parameters can be selected through this filter:

AllAgents the switch reports the status of all the ACD agents associated to the session using positionIDs.

AllCDNs the switch reports the status of all the CDNs associated to the session.

AllDevices the switch reports the status of all the CDNs and agents associated by their PositionIDs.

4.9.2.2 dv-Appl-Stat-Qry error conditions

If the dv_Appl_Stat_Qry does not pass validation then the switch sends a Return Error service message, dv_Appl_Stat_Qry_RE.

Note: The error reasons are bold in brackets.

1 The service version is at least SCAI11.

2 The CompuCALL session subscribes to the application status query message in table SCAISSRV. Also one or more parameters of the status query parameters are subscribed in table SCAISSRV for the response messages {not_allowed}.

3 There is at least one parameter in the message and not more than one query parameter (either DN or query filter) specified in the message {invalid_message}.

4 The address is a valid DN (only if specified) and is associated to the current session {invalid_DN}.

5 The query filter parameter specified is one of Allagents, AllCDNs, or AllDevices {invalid_filter}.

6 The address is a valid POSID (only if specified) and is associated to the session either using its PositionID or the ACD group to which it belongs {invalid_DN}.

7 There is at least one agent or CDN associated to the session ({for the status to be reported for QueryFilter parameters noassociated_device}).

8 There is at least one agent belonging to the associated DN when primDN of an ACD group is specified as the query parameter {noassociated_device}.

9 The specified DN or POSID belongs to the customer group associated to the current session {unknown_DN}.

4.9.2.3 dv-Appl-Stat-Qry return result

The switch sends a dv_Appl_Stat_Qry_RR to the host confirming that it is a valid query. This Return Result contains no parameters.

4.9.2.4 dv-Appl-Stat-Qry return reject

In the case of invalid service version or corruption in data or incorrect argument, the host computer sends a Return Reject message indicating that the message is incorrectly decoded.

4.9.2.5 dv-Appl-Stat-Qry message: ASN.1 encoding

```
dv_Appl_Stat_Qry                    OPERATION
                                    ARGUMENT QueryMessageArg
                                    RESULT
                                    ERRORS QueryMessageErrors
::= 39
QueryMessageArg ::= SEQUENCE
{DN                    [0]     AddressType          OPTIONAL,
QueryFilter            [1] IMPLICIT QueryFilterTypeOPTIONAL}

AddressType   ::= CHOICE
{positionID    [0] IMPLICIT         INTEGER,
dialedDigits   [1] IMPLICIT         IA5String,
stationNumber [2]      IMPLICIT     OCTET STRING,
q931Address    [3] Q931AddressType}

QueryFilterType::=     BITSTRING
    {Allagents, AllCDNs, AllDevices}

QueryMessageErrors     ::= ENUMERATED
{notAllowed            (0),
 unknownDN             (1),
 invalidDN             (2),
 invalidFilter         (3),
 invalidMessage        (4),
 noassociateddevice    (5),
QryalreadyInProgress (6)}
```

4.9.2.6 dv-Appl-Stat-Qry: coded example

dv-Appl-Stat-Qry message sent by the host application to the switch enquiring the status of a DN whose station number is (504)886-1180. The invoke ID is 3.

INVOKE DMS < ===== HOST

Table 115 dv-Appl-Stat-Qry invoke coded example

Hex Dump			Information	Comments
Tag	Length	Content		
A1	16		INVOKE	the message length is 22 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	27	Operation	its operation value is 39 decimal
30	0E		Argument	QueryMessageArg is of type sequence and 14 bytes long decimal
A0	0C		DN	the DN is type constructor and is 12 bytes long
82	0A	35 30 34 38 38 36 31 31 38 30	stationNumber	the DN of the phone is 504-886-1180
Hex Dump = A1 16 02 01 03 02 01 27 30 0E A0 0C 82 0A 35 30 34 38 38 36 31 31 38 30				

RETURN-RESULT DMS ====> HOST

Table 116 dv-Appl-Stat-Qry return result coded example

Hex Dump			Information	Comments
Tag	Length	Content		
A3	06		RETURN RESULT	the message length is 6 bytes decimal
02	01	03	Invoke ID	its Invoke ID is 3
02	01	00	Error Value	the message is not allowed since it is not subscribed to
Hex Dump = A3 06 02 01 03 02 01 00				

RETURN-ERROR DMS ====> HOST

Table 117 dv-Appl-Stat-Qry return error coded example

Hex Dump			Information	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	the message length is 6 bytes decimal
02	01	03	Invoke ID	its Invoke ID is 3
02	01	00	Error Value	the message is not allowed since it is not subscribed to
Hex Dump = A3 06 02 01 03 02 01 00				

4.9.2.7 Examples of query filter

The following figure shows all the different combinations of the bit stream of the QueryFilter, the hex value of the bit stream and the meaning of the QueryFilter. Bit 1 represents the Allagents parameter, bit 2 represents the AllCDNs parameter. If bit 1 is set to value 1 then the host is requesting the status of all associated agents and if bit 2 is set to value 1, then the host is requesting the status of all associated CDNs. If both bit 1 and bit 2 is set to 1, it implies that the host is requesting the status of all the associated devices. The host receives a return error with reason invalid filter if anything other than the following hex values are in the QueryFilter: 01, 02, 03.

Figure 49 Bitstream for query filter

Bits		Hex	Query Filter
0 0 0 0	0 0 0 1	01	Allagents
0 0 0 0	0 0 1 0	02	AllCDNs
0 0 0 0	0 0 1 1	03	AllDevices
8 7 6 5	4 3 2 1		

4.9.3 dv-Agent-Status-U message

The dv-Agent-Status-U message maps to a ROSE Class 5 Operation. The switch sends this message under three conditions based on the interpretation of the Status Query Message:

- The DN parameter the host specifies is an ACD DN.
- The DN parameter the host specifies is the POSID of an associated agent.
- The host specifies Allagents or AllDevices parameter in the query filter.

A dv-Agent-Status-U message reports the status of an ACD agent for the ACD incalls key and the secondary DN key. Then the switch reports:

1 POSID

2 Status

a. Agent status (busy, idle, not ready, logged out, not logged in) If the status is not ready, either the three-digit walkaway code or an indication of the variable wrap-up time is provided. If the walkaway code is not entered by the agent, then the switch reports the default walkaway code 000.

b. Call status (idle, ringing, active, line_not_in_condition) for both the ACD incalls key and the first located secondary DN key.

4.9.3.1 dv-Agent-Status-U parameters

FLAG indicates if upload is in progress or all done.

POSID specifies the Position-ID of the ACD agent.

STATUS includes Agent Status as well as Call Status.

Idle indicates that the agent is ready to receive calls.

Busy indicates that the agent is attending the call.

Logged out indicates that the agent is logging out of the session.

Not Logged in indicates that the agent is not logged in to the session.

Not Ready indicates that the agent is not ready to receive the incoming calls.

4.9.3.2 dv-Agent-Status-U message: ASN.1 encoding

```
dv_Agent_Status_U                  OPERATION
                                   ARGUMENT  StatusInfoArg
:: = 40

StatusInfoArg ::=  SEQUENCE
{FLAG                       [0]            IMPLICIT FlagType,
 POSID                      [1]            AddressType,
STATUS                      [2]            IMPLICIT StatusType }

AddressType   ::=          CHOICE
{positionID                [0] IMPLICIT          INTEGER,
dialedDigits               [1] IMPLICIT          IA5String,
stationNumber              [2] IMPLICIT          OCTETSTRING,
q.931Address               [3] Q.931AddressType}

CCL - StatusType::=   SEQUENCE
{AgentStatus   [0]  AgentStatusType OPTIONAL,
 CallStatus      [1] IMPLICIT CallStatusType   OPTIONAL }

AgentStatusType ::=  CHOICE
{busy            [0]              IMPLICIT NULL,
 idle            [1]              IMPLICIT NULL,
 notready        [2]              notReadyArgType,
 loggedout       [3]              IMPLICIT NULL,
notloggedin      [4]             IMPLICIT NULL}

CallStatusType  ::=      SEQUENCE
{Incallstatus [0]        IMPLICIT, CallstateType
 SDNstatus   [1]        IMPLICIT CallstateType  }

CallstateType   ::=     ENUMERATED
{idle            (0),
 ringing         (1),
 active                   (2),
 line_not_in_condition (3)}


notReadyArgType  ::=   CHOICE
{walkawayreason   [0]   IMPLICIT   IA5STRING   OPTIONAL,
 varwrapuptime     [1]  IMPLICIT NULL  OPTIONAL }

FlagType        ::=     ENUMERATED
{uploadinprogress     (0),
 lastcdnstat          (1),
 alldone              (2)}
```

4.9.3.3 dv-Agent-Status-U: coded example

In response to the query requested by the host application about the agent status of the ACD DN (504)886-1180, the dv_Agent_Status_U message is sent by the switch reporting the status information of the agents associated to this ACD group. The following is a typical coded example of this message from switch to host reporting the status of a single agent whose POSID is 1039. The status of the agent is not ready with the walkaway reason as 123. The call status is idle for incalls key and active for the first located SDN key. The flag parameter is ‘uploadinprogress' which implies that there is more agent status information to follow.

INVOKE SWITCH ====> HOST

Table 118 dv-Agent-Status-U invoke coded example

Hex Dump			Information	Comments
Tag	Length	Content		
A1	24		INVOKE	the message length is 36 bytes decimal
02	01	05	InvokeID	it invoke ID is 5 decimal
02	01	28	Operation	it operation value is 40 decimal
30	1C		Argument	StatusInfoArg is type sequence and is 28 bytes long decimal
80	01	00	Flag	FlagType is of 1 byte long decimal and indicates upload in progress
A1	04		POSID	the POSID is type constructor and is 4 bytes decimal long
80	02	04 0F	positionID	the agent positionID is 1039 decimal
A2	11		STATUS	the Status Type is a constructor of length 17 bytes decimal
A0	07		Agent status	Agent status is a constructor 7 bytes long decimal
A2	05		not ready	the agent status is not ready
80	03	31 32 33	walkaway reason	three digit walkaway code 123
A1	06		Call status	it is a constructor length 6 bytes
80	01	00	Incallstatus	the incall status is idle
81	01	02	SDN status	the status of the SDN is active
Hex Dump = A1 24 02 01 05 02 01 28 30 1C 80 01 00 A1 04 80 02 04 0F A2 11 A0 07 A2 05 80 03 31 32 33 A1 06 80 01 00 81 01 02				

4.9.4 dv-CDN-Status-U message

A dv-CDN-Status-U message maps to a ROSE Class 5 Operation. The switch sends this message under the following conditions:

- The DN number the host specifies is a CDN.
- The host selects AllCDNs or AllDevices as the QueryFilter parameter.

The switch reports:

1 CDN number

2 Status

   a. Controlled

   b. Default

   c. Revert to default

4.9.4.1 dv-CDN-Status-U parameters

FLAG This parameter indicates if upload is in progress or the status of the last CDN is being reported or all done.

CDN This parameter specifies the CDN for which the status is to be reported.

STATUS This parameter specifies the status of the CDN.

4.9.4.2 dv-CDN-Status-U message: ASN.1 encoding

```
dv_CDN_Status_U             OPERATION
                            ARGUMENT          cdnInfoArg
::= 41

cdnInfoArg    ::=     SEQUENCE
{FLAG          [0]             IMPLICIT FlagType,
 CDN           [1]            AddressType,
 STATUS        [2]            IMPLICIT cdnStatType}

AddressType   ::=             CHOICE
{positionID    [0] IMPLICIT          INTEGER,
dialedDigits   [1] IMPLICIT          IA5String,
stationNumber [2]     IMPLICIT       OCTET STRING,
q.931Address    [3] Q.931AddressType}

cdnStatType    ::=      ENUMERATED
{default         (0),
 controlled      (1),
 reverttodefault  (2)}

FlagType       ::=ENUMERATED
{uploadinprogress    (0),
 lastcdnstat            (1),
 alldone                (2)}
```

4.9.4.3 dv-CDN-Status-U: coded example

In response to the query requested by the host application about the status of the associated CDN number (504)886-0811, the dv_CDN_Status_U message is sent by the switch reporting the status of the CDN as controlled. The flag parameter is set as alldone indicating that there are no other messages to follow.

INVOKE SWITCH ====> HOST

Table 119 dv-CDN-Status-U invoke coded example

Hex Dump			Information	Comments
Tag	**Length**	**Content**		
A1	1E		INVOKE	the message length is 30 bytes decimal
02	01	08	InvokeID	it InvokeID is 8 decimal
02	01	29	OPERATION	the operation value of the message is 41 decimal
30	19		Argument	the cdninfoArg is 25 bytes long decimal
80	01	03	FLAG	FlagType is of 1 byte long decimal and it indicates alldone
A1	0C		CDN	the CDN is type constructor and is 12 bytes long
82	0A	35 30 34 38 38 36 30 38 31 31	stationNumber	the station number is 504-886-0811
82	03		STATUS	it is a primitive 1 byte long
81	01	01	controlled	the state of the CDN is controlled
Hex Dump = A1 1E 02 01 08 02 01 29 30 19 80 01 03 A1 0C 82 0A 35 30 34 38 38 36 30 38 31 31 82 03 81 01 01				

4.10 Third Party Queue Control

The Application Service capability covers three application service functions:

- dv-Route-Call
- dv-Give-Treatment
- dv-Treatment-Complete-U

4.10.1 dv-Route-Call

The host computer uses this message to route a call in an incoming or overflow ACD queue. In CompuCALL, dv-Route-Call is a call control message. Therefore, do not send an association message to associate the ACD group DN, ACD agent position ID and CDN prior to this message.

The host computer has to provide the following information:

- NetworkCallID
- ACDDN
- destination

Destination can be any one of the following:

1 A DN within or outside of the switch

2 An ACD group DN, CDN within or outside of the switch

3 AN ACD agent’s position ID within the switch

For a CDN, the destination parameter is used to represent a default destination as well as a destination that is not a default. For details on the format of this parameter see Section 4.10.1.5, “dv-Route-Call: ASN.1 definition,” on page 341.

The switch does the following when it receives a dv-Route-Call message:

- checks the parameters for validity
- stops the timer
- datafills Time t if it is still running
- removes the call from the ACD queue
- removes any treatment from the call
- checks if the agent is idle
- routes the call to the destination
- sends a Return Error or Return Result
- gives treatment to a call if the destination is busy

- sends a Return Error if the destination and agent is busy

  Note: A timer to check response time is not there for current ACD groups. It is present when there is a CDN option for the ACD group.

4.10.1.1 dv-Route Call Parameters

All of the parameters are mandatory for the message to be successful.

NetworkCallID identifies the call. This consists of two subparameters:

NetworkNodeID is 1-2 bytes long

LocalCallID is 1–4 bytes long

ACDDN indentifies the incoming call's queue. Thirty-two digits is the maximum definition.

Destination identifies the destination (internal or external to switch, default DN) to terminate. This is a maximum of 32 digits. Destination is a choice parameter.

— only the first missing parameter encountered is reported in this message

— only the first invalid parameter encountered is reported in this message

4.10.1.2 Return Result

If the requested route call was a valid request the switch will respond by sending the host a return result with the following parameter:

NetworkCallID identifies the call within the context of the customer's network for the maximum duration of the call. It provides a cross-reference for DMS-100 switch or messages the host generates that relate to the same call.

4.10.1.3 Return Error

If the requested route call is invalid the switch responds by sending the host a return error with the following parameter:

NetworkCallID identifies the type of error that occurs to cause the route call request to fail.

4.10.1.4 dv-Route-Call error conditions

The switch performs a series of checks on receiving this message. If the request to route a call fails, the switch sends a Return Error message to the host indicating the failure. The information in braces indicates the error message parameter which the switch sends if the validation fails.

1 The service version the host uses to log on to a session is SCAI10 or higher {invoke_problem:= unrecognized_op}.

2 The link in table SCAIPROF the host uses to log on has the category TPQC {not allowed}.

3 The host session subscribes to the dv-Route-Call message using table SCAISSRV ([optional subscription to TPQC message {not allowed}]).

4 The parameters NetworkCallID, ACDDN and destination are present and valid {missing-parameter including which is missing}.

5 The parameter NetworkCallID identifies the call and the parameter ACDDN identifies the call in the queue {invalid parameter = invalid-Network CallId}.

6 If the destination is default, the ACDDN is a CDN and the default ACD group is datafilled in table ACDGRP {invalid parameter = invalid-destination}.

7 The destination translates successfully according to the datafill for DMS-100 switch {invalid parameter = invalid-destination}.

8 If the destination is an agent position ID, the agent (destination) the position IDs is idle {agent-busy}.

Note: Only the first error condition (including the missing/invalid parameter) the switch encounters is included in the Return Error message

Error Handling:
An error occurs if any of the parameters (NetworkCallID, ACDDN, destination) are not valid or missing. In this situation the call remains in the queue. The switch sends an error message with appropriate error reason to the host. If the destination is an agent position ID failure occurs when:

1 agent is Not Ready

2 agent is Busy (This can occur during pre-termination and termination)

When failure occurs, during:

1 Pre-termination, the switch sends an error message to host and the call remains in queue

2 Termination and post termination, the switch returns the call to the ACD queue and creates the event dv-Call-Queued-U with a call history reason return to queue and sends it to the host computer. The call receives ringback treatment while it is being presented to an agent and continues to receive ringback treatment when it is returned to queue. This prevents the caller from getting discontinuous treatment.

4.10.1.5 dv-Route-Call: ASN.1 definition

The ASN.1 notations for the messages provided by this activity are shown in this section. Following each notation is a sample X.209 hex encoding. The length fields are variable because the length depends on the variable content. The tag values are taken from the ROSE protocols.

```
dv-Route-Call  OPERATION
                              ARGUMENT          RouteCallArgument
                              RESULT            RouteCallResult
                              ERRORS            {missing-Parameter
                                                 invalid-Parameter,
                                                 not-allowed,
                                                 agent-busy}
::= 34
RouteCallArgument   ::=       SEQUENCE
    {networkCallID    [0] IMPLICIT SEQUENCE
    networkNodeID     [0] IMPLICIT INTEGER      OPTIONAL,
    localCallID        [1] IMPLICIT INTEGER     OPTIONAL}

acdDN          [1] IMPLICIT OCTET STRING        OPTIONAL,
destination    [2] DestinationInfo              OPTIONAL}

DestinationInfo :: =   CHOICE
{   default            [0] IMPLICIT NULL,
    destAddress        [1] AddressType }

AddressType   :: =   CHOICE
{   positionID         [0] IMPLICIT INTEGER,
    dialedDigits       [1] IMPLICIT IA5STRING,
    stationNumber      [2] IMPLICIT OCTET STRING
    q931Address        [3] Q931AddressType}

Q931AddressType      ::= SEQUENCE
{   numberTypeNumberPlan              [0] OCTET STRING  OPTIONAL,
    presentationScreeningIndicator    [1] OCTETSTRING   OPTIONAL,
    digits                            [2] IA5 STRING    OPTIONAL}

RouteCallResult ::=   SEQUENCE
    {networkCallID     [0] IMPLICIT INTEGER             OPTIONAL,
     localCallID       [1] IMPLICIT INTEGER             OPTIONAL}

missing-Parameter   ERROR ::= 0
                    PARAMETER    MissingParameter

 MissingParameter::=   SEQUENCE
{ missingParameterType               [0]  IMPLICIT  ENUMERATED
   {missingNetworkCallID             (0),
    missingACDDN                 (1),
    missingDestination           (2)}    OPTIONAL}


invalid-Parameter    ERROR ::= 1
                    PARAMETER  InvalidParameter

InvalidParameter    ::=   SEQUENCE
{ invalidParameterType           [0] IMPLICIT  ENUMERATED
  { invalidNetworkCallID         (0),
    invalidACDDN                 (1),
    invalidDestination           (2)}              OPTIONAL}
not-allowed    ERROR        ::= 2
agent-busy     ERROR        ::= 3
```

4.10.1.6dv-Route-Call: coded example

This example illustrates the case where a host sends a message to route a call. Invoke ID is 11.

INVOKE DMS <===== HOST

Table 120 dv-Route-Call (destination specified as a default DN)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	20		INVOKE	Message length is 32 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	22	Operation	Operation value for dv-Route-Call is decimal 34.
30	18		RouteCallArgument	Type sequence and 24 bytes long.
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
81	0A	36 31 33 36 32 31 32 34 30 30	...........acdDN	the called ACDDN of the group is DN = 6136212400
A2	02		..........destination	Information about the destination is of constructor type and is 2bytes long
80	00		..........default	default destination is NULL type
Hex Dump = A1 20 02 01 0B 02 01 22 30 18 A0 06 80 01 01 81 01 02 81 0A 36 31 33 36 32 31 32 34 30 30 A2 02 80 00				

INVOKE DMS <===== HOST

Table 121 dv-Route-Call (destination specified as an agent)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	23		INVOKE	Message length is 35 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	22	Operation	Operation value for dv-Route-Call is decimal 34.
30	1B		RouteCallArgument	Type sequence and 27 bytes long.
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
81	0A	36 31 33 36 32 31 32 34 30 30	...........acdDN	the called ACDDN of the group is DN = 6136212400
A2	05		..........destination	Information about the destination is of constructor type and is 5 bytes long
A1	03		..........destAddress	The destination Address of constructor type and is 3 bytes long
80	01	2F	..........PositionID	the position ID component is decimal 47
Hex Dump = A1 23 02 01 0B 02 01 22 30 1B A0 06 80 01 01 81 01 02 81 0A 36 31 33 36 32 31 32 34 30 30 A2 05 A1 03 80 01 2F				

Table 122 RETURN-RESULT for Route Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0D	InvokeID	Invoke ID is 13.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	22	Operation	its operation is dv-Route-Call [34]
30	09		RouteCallResult	the RouteCall Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0D 30 0E 02 01 22 30 09 A0 07 80 01 01 81 02 05 55				

Table 123 RETURN-ERROR for Route Call with an error value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0D	InvokeID	Invoke ID is 13.
02	01	02	ErrorType	ErrorType is notAllowed [2]
Hex Dump = A3 06 02 01 0D 02 01 02				

4.10.2 dv-Give-Treatment message

dv-Give-Treatment supports the function to apply treatment to a call in an ACD incoming or overflow call queue.

If calls in the ACD incoming and overflow call queues are receiving treatment, the switch verifies the request prior to stopping the treatment and applies the new request.

The dv-Give-Treatment message supports calls in an acd incoming or overflow call queue. An ACD incoming call queue is associated with an ACD DN and an ACD group. The overflow call queue holds the overflow of calls from the incoming call queue.

When a call terminates in an ACD DN incoming call queue and the host application associates to that ACD DN using the dv-DN-Associate message, the host computer receives a dv-Call-Queued-U message if there are no agents available and the customer does not subscribe to the CompuCALL ACD redirection capability. If a customer subscribes to the CompuCALL ACD redirection capability, the switch sends a dv-Call-Received message to the host.

This message provides the host computer with the information it needs to send a dv-Give-Treatment message back to that DN so the caller on the other end can receive either AUDIO, RAN, Music, Ringback, Silence, Busy, Fastbusy or Disconnect. The host sends the dv-Give-Treatment messages after receiving a dv-Call-Queued-U message.

The RAN and Music request are only applied if the route parameters specify an audio sequence consistent with the Audio Treatment type specified in the request. RAN request receive Audio treatment and Music request receive Music treatment.

Busy, Fastbusy, and Disconnect requests remove the call from the incoming or overflow call queue prior to the application of the treatment. RAN, Music, Ringback, and Silence allow the call to remain in the queue while receiving the treatment, thus being able to receive additional treatments.

4.10.2.1 dv-Give-Treatment Message parameters

dv-Give-Treatment contains the following mandatory parameters:

NetworkCallID contains the information to locate the within the queue. It contains the NetNodeID and LocalCallID. The ECM LocalCallID is not the same as the DMS CallID.

ACDDN contains the information to locate the queue that holds the call. It is defined as an octet string, but is verified for the contents of 10 digits.

treatmentType determines the treatment to apply to the call. The treatment type can be either a tones type or an audio type. The tones type consists of Ringback, Silence, Busy, Fastbusy, or Disconnect. The Audio type consists of RAN or MUSIC. If the audio type RAN or MUSIC is sent, an additional ROUTE parameter must also be sent.

A route entry of 512, defaults the AUDIO to the datafilled route in table ACDGRP. This is the ACD group that associates with the ACDDN parameter.

4.10.2.2 dv-Give-Treatment Return Result

If the switch validates an incoming request, the host receives a Return Result to indicate that the request is valid and initiated.

4.10.2.3 dv-Give-Treatment Return Reject

Any error condition as a result of protocol problems encountered when decoding an RO message.

4.10.2.4 dv-Give-Treatment error conditions

Any error conditions the switch detects result in a Return Error message to the host. The Return Error message contains an error cause value to inform the host why the request is not initiated. The error conditions are bold in braces.

1 The service version of the current host-switch session is SCAI10 or higher {invoke_problem}.

2 The link in table SCAIPROF the host uses to log on has the category TPQC {not_allowed}.

3 The host session subscribes to the dv-Give-Treatment message using table SCAISSRV {not_allowed}.

4 The parameters; NetworkCallID, ACDDN and treatmentType are missing in the message {missing_parameter}.

5 The NetworkCallID exists in the ACDDN incoming or overflow call queue {invalid_parameter}.

6 The AUDIO option is present if there is a request for RAN or Music {no_Audio_Option}.

7 A call is currently receiving audio treatment, ringing, or silence a treatment request is applied. All treatments which result in the call remaining in the ACD incoming call queues are interruptible {invalid_FeatureInteraction}.

8 The AUDIO request does not include the ROUTE parameter {missingParameter and missingTreatmentToGive}.

9 The host request a sequence that does not end in the appropriate keyword {InconsistentRoute}.

10 The host request a sequence on an ACD group that has an entry in table ACDRTE. The request does not include a predefined audio sequence {RouteNotAllowed}.

4.10.2.5 dv-Give-Treatment: ASN.1 encoding

The ASN.1 notations for the call control message provided by this activity are show in this section. Following each notation is a sample X.209 hex encoding. The length fields are variable because the length depends on the variable content. The tag values are taken from the ROSE protocols.

Operation Class 2: Asynchronous, reporting success or failure (result or error).

```
dv-Give-Treatment                   OPERATION
                                    ARGUMENT  TreatmentArgument
                                    RESULT    GiveTreatmentResult
                         ERRORS     {notAllowed,
                                    missingParameter,
                                    invalidParameter,
                                    noAudioOption,
                                    invalidFeatureInteraction
                                    audioRouteDoesNotExist,
                                    inconsistent_route,
                                    route_not_allowed}
::= 35
TreatmentArgument::=SEQUENCE
{   networkCallID           [0] IMPLICIT SEQUENCE
        {networkNodeID      [0] IMPLICIT INTEGER     OPTIONAL,
         localCallID        [1] IMPLICIT INTEGER     OPTIONAL},
    ACDDN           [1] IMPLICIT OCTET STRING        OPTIONAL,
    treatmentToGive [2] TreatmentType                OPTIONAL}

TreatmentType        ::=                CHOICE
{   Announcements           [0] IMPLICIT AnnouncementType,
    Tones                   [1] IMPLICIT Tones
{Ringback (0), Silence (1), Busy (2), Fastbusy (3), Disconnect (4)}

AnnouncementType     ::=                SEQUENCE
{AnnouncementName           [0] IMPLICIT 1A5STRING   OPTIONAL,
AnnouncementRoute           [1] IMPLICIT INTEGER     OPTIONAL}

GiveTreatmentResultt::=                 SEQUENCE
    {networkCallID  [0] IMPLICIT INTEGER             OPTIONAL,
     localCallID    [1] IMPLICIT INTEGER             OPTIONAL}

notAllowed          ERROR::=0
missingParameter    ERROR::=1
            PARAMETER      MissingParameter

MissingParameter    ::=             SEQUENCE
    {missingParameterType           [0] IMPLICIT ENUMERATED
        {missingNetworkCallID       (0)     ,
              missing ACDDN              (1),
              missingTreatmentToGive     (2)                 OPTIONAL}

invalidParameter        ERROR::=2
                 PARAMETER    InvalidParameter
      InvalidParameter  ::=                      SEQUENCE
      {invalidParameterType            [0] IMPLICIT ENUMERATED
            {invalidNetworkCallID      (0),
             invalidACDDN              (1),
             invalidTreatmentToGive    (2)
noAudioOption                 ERROR::=3
invalidFeatureInteraction     ERROR::=4
audioRouteDoesNotExist        ERROR::=5
inconsistent- route           ERROR::=6
route-not-allowed             ERROR::=7
```

4.10.2.6 dv-Give-Treatment: coded example

INVOKE DMS <===== HOST

Table 124 dv-Give-Treatment invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	28		INVOKE	Message length is 33 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	23	Operation	Operation value for dv-Give Treatment is decimal 35.
30	19		TreatmentArgument	Type sequence and is 25 bytes long.
A0	06		networkCallID	NetworkCallID is of type constructor and is 6 bytes long.
80	01	01	.....networkNodeID	The networkNodeID is 1.
81	01	02	..... localCallID	The localCallID is 2.
81	0A	36 31 33 36 32 31 32 34 30 30	ACDDN	The called ACDDN is 613-621-2400.
A2	0A		...treatmentToGive	The treatmentToGive is a constructor and is 3 bytes long.
81	01	00	Tones	The tones is 0 for Ringback
Hex Dump = A1 21 02 01 0B 02 01 23 30 19 A0 06 80 01 01 81 01 02 81 0A 36 31 33 36 32 31 32 34 30 30 A2 0A 81 01 00				

INVOKE DMS<====HOST

Table 125 GIVE-TREATMENT for AUDIO request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	28		INVOKE	Message length is 33 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	23	Operation	Operation value for dv-Give Treatment is decimal 35.
30	19		TreatmentArgument	Type sequence and is 25 bytes long.
A0	06		networkCallID	NetworkCallID is of type constructor and is 6 bytes long.
80	01	01	.....networkNodeID	The networkNodeID is 1.
81	01	02	..... localCallID	The localCallID is 2.
81	0A	36 31 33 36 32 31 32 34 30 30	ACDDN	The called ACDDN is 613-621-2400.
A2	0A		...treatmentToGive	The treatmentToGive is a constructor and is 3 bytes long.
80	05	41 55 44 49 4F	AnnouncementName	The announcement name is AUDIO.
81	01	14	AnnouncementRoute	The announcement route is 20.
Hex Dump = A1 21 02 01 0B 02 01 23 30 19 A0 06 80 01 01 81 01 02 81 0A 36 31 33 36 32 31 32 34 30 30 A2 0A 80 05 41 55 44 49 4F 81 01 14				

RETURN-RESULT DMS==>HOST

Table 126 RETURN-RESULT for Give Treatment Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	12		RETURN RESULT	The message length is 18 bytes.
02	01	0B	.. InvokeID	InvokeID is 11.
30	03		.. Return Result Sequence	This sequence is 13 bytes long.
02	01	23	... Operation	Operation value for dv-Give-Treatment is decimal 35.
30	08		...... GiveTreatmentResult	the givetreatmentresult is of type sequence and is 8 bytes long.
A0	06		.......Network Call ID	NetworkCallid is of type constructor and is 6 bytes long.
80	01	01	.... NetworkNodeID	Network Node ID is 1.
81	01	02	.... LocalCallID	Local Call ID is 2.
Hex Dump = A2 12 02 01 0B 30 03 02 01 23 30 08 A0 06 80 01 01 81 01 02				

RETURN-ERROR DMS==>HOST

Table 127 RETURN-ERROR Give Treatment with an error value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	The message length is 6 bytes.
02	01	0B	.. InvokeID	InvokeID is 11
02	01	04	.. ErrorType	ErrorType is InvalidFeatureInteraction.
Hex Dump = A3 06 02 01 0B 02 01 04				

4.10.3 dv-Treatment-Complete-U

This capability notifies the host application when a treatment is complete.

> ***Note:*** RAN and Music capabilities are limited in CCM08 release. This release focuses on providing the ability for the host application to request a RAN, Music, or AUDIO treatment with routing instructions.

RAN and Music are the only audio treatment requests, that a dv-Treatment-Complete-U message sends. This lets the host application know that the audio route is complete and they should send another audio request. The completed route is included in the completed message. The route specifies which announcement is complete.

RAN and Music requests require a specific route and are considered complete when the AUDIO sequence in that route is complete.

Since the RAN and Music requests result in the same audio being played, a host application that wants Music to interrupt RAN has to achieve this through switch datafill not dv-Give-Treatment messages.

4.10.3.1 dv-Treatment-Complete-U message parameters

The switch sends a dv-Treatment-Complete to the host application when an audio treatment is complete. The only treatments which can complete are those defined as finite in table AUDIO. The RAN and Music requests specify which route in table AUDIO the caller should hear. The message will contain the following mandatory parameters:

NetworkCallID A netnodeID (0 to 32767) and a localcallID (0 to 4,294,967,295) defines a network call ID.

ACDDN is 32 Bytes long.

TreatmentCompleted Audios is the only valid treatment included in the dv-Treatment-Complete message. The audio route is also included with the AUDIO key word to keep from confusing the host when the host may request multiple RANs or Musics to a call.

4.10.3.2 Ringback and Silence capabilities

If a host application requests that the audio route for the ACD group be made of announcements and music and the music has a finite time defined, the audio route could potentially complete before the call is answered. The host computer receives a dv-Treatment-Complete-U message to indicate the audio is complete. The host computer can decide to request Ringback or Silence on the call. Later, if the call is still not answered, the host computer can decide to send the call back to the audio treatment.

> ***Note:*** This also can be achieved through a datafilled sequence in table AUDIO. Although this sequence could not be changed from call to call and may not result in a dv-Treatment-Complete-U message being sent as described previously.

Another reason for a host computer to request Ringback and Silence is to request treatment on a call when a call first enters an ACD incoming call queue. In the current ACD CompuCALL environment, a caller can receive ringback when entering the queue if the AUDIO option is assigned with the RANTH (recorded announcement threshold) set to some number greater than 0 and less than 60 and Ringing set to Y. If the RANTH field is set to 0 the caller immediately receives a recorded announcement.

Even though the AUDIO option must be datafilled to process RAN and Music treatment requests, it does not need to be datafilled in order to receive Ringback or Silence. The host computer can request Ringback or Silence as the first treatment.

For CCM08, if an ACD incoming or overflow call queue is following switch datafill instructions and if the AUDIO option is datafilled, then the first treatment is defined under the AUDIO option. A host computer can send a treatment message to change that treatment, but the datafilled treatment is already applied. If the AUDIO option is not present, then the caller hears silence until the host computer sends a treatment request to the switch. RAN and Music requests are invalid if the AUDIO option is not present for the ACD group associated with the call.

If the datafill in table ACDGRP indicates that the incoming or overflow call queue should follow instructions the host computer sends then the AUDIO option and the associated RANTH and RINGING fields are ignored. Instead, the host computer is required to send a treatment request of Silence or Ringback if desired as a first treatment. The AUDIO option is only needed for the RAN, Music, AUDIO and default route requests, ignoring the RANTH and RINGING fields.

4.10.3.3 AUDIO capabilities

The customer sends AUDIO treatment when they do not want to check the sequence for keywords prior to applying treatment. Audio sequences can end in the keywords: ANN, MUSIC, SILENCE, RINGBACK, and REPEAT.

You can enter the ANN keyword multiple times in the sequence as long as you enter one of the keywords MUSIC, SILENCE or RINGBACK between the ANN keywords. MUSIC is defined as finite if it is between 2 ANN keywords and infinite if it is the last keyword in the sequence. MUSIC, SILENCE, and RINGBACK are delays and are inserted in the sequence to break between announcements.

For the customer who only requires music, the sequence only contains the keyword MUSIC with 0 entered in the TIME prompt. 0 is the only valid entry when MUSIC is the last keyword in the sequence. Keyword REPEAT is not valid when you enter MUSIC in the sequence.

Host computer dv-Give-Treatment messages which contain a request for RAN or MUSIC will not be able to play a sequence with REPEAT, SILENCE, or RINGBACK as the final keyword in the sequence.

Messages which contain a request for AUDIO allow any sequence to play.

4.10.3.4 Busy, fastbusy, disconnect capabilities

The host computer sends treatment requests for busy and fastbusy to force a call that is located in an ACD incoming or overflow call queue to be removed from the queue.

The DMS-100 switch does not consider a disconnect to be a treatment, but rather a clearing of the call. The switch clears calls when it can not correct abnormal conditions.

Once the call receives a busy, fastbusy, or disconnect treatment request, any current treatment the call receives is stopped (including audio treatments) and the call is removed from the queue and routed to busy, fastbusy, or disconnect.

If the current call treatment is an audio RAN treatment and the host computer interrupts with a Busy, Fastbusy, or Disconnect treatment request, no dv-Treatment-Complete-U message is sent. The AUDIO RAN treatment is interrupted and not completed.

The caller must hang-up to end the treatment, busy, and fastbusy. Disconnect ends the call for the caller.

4.10.3.5 dv-Treatment-Complete-U: ASN.1 encoding

```
dv-Treatment-Complete-U     operation
                            ARGUMENT  TreatmentCompleteArgument
::=36
TreatmentCompleteArgument::=       SEQUENCE
    {networkCallID          [0] IMPLICIT SEQUENCE
        {networkNodeID      [0] IMPLICIT INTEGER        OPTIONAL,
         localCallID        [1] IMPLICIT INTEGER        OPTIONAL},

 ACDDN              [1] IMPLICIT OCTET STRING   OPTIONAL,
     treatmentType  [2] AnnouncementType        OPTIONAL}

AnnouncementType::=         SEQUENCE
{   AnnouncementName[0] IMPLICIT 1A5STRING      OPTIONAL,
    AnnouncementRoute[1] IMPLICIT INTEGER       OPTIONAL}
```

4.10.3.6 dv-Treatment-Complete-U: coded example

INVOKE DMS==>HOST

Table 128 Treatment complete message

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	29		INVOKE	Message length is 41 bytes.
02	02	0201	InvokeID	Invoke ID is 513.
02	01	24	Operation	Operation value for dv-Treatment-Complete is decimal 36.
30	20		TreatmentCompleteArgument	Type sequence and is 32 bytes long.
A0	06		networkCallID	NetworkCallID is of type constructor and is 6 bytes long.
80	01	01	.......networkNodeID	The networkNodeID is 1.
81	01	02	.... localCallID	The localCallID is 2.
81	0A	36 31 33 36 32 31 32 34 30 30	... ACDDN	The called ACDDN 613-621-2400.
A2	0A		... treatmentType	The treatment is a constructor and is 10 bytes long.
80	05	41 55 44 49 4F	.... AnnouncementName	The announcement name is AUDIO
81	01	00	.... AnnouncementRoute	The announcement route is 0.
Hex Dump = A1 29 02 02 02 01 02 01 24 30 20 A0 06 80 01 01 81 01 02 81 0A 36 31 33 36 32 31 32 34 30 30 A2 0A 80 05 41 55 44 49 4F 81 01 00				

4.11 Integrated Call Center Manager (ICCM)

The application service capability supports dv-Set-CDN-State and dv-Reassign-Agent messages.

4.11.1 dv-Set-CDN-State Message

The dv-Set-CDN-State has an RO of Operation Class 2.

A directory number (DN) identifies a CDN that holds calls until receiving a host computer’s routing instructions. The host computer can route calls to any addressable local or external party. Also the host computer can route calls to agent positions, ACD DNs, and other CDNs. The call can receive call treatments while waiting for routing instructions from the host computer. For each incoming call to a CDN, the switch informs the host computer about the call’s arrival, starts a timer, and waits for the host computer’s routing instructions. The switch handles the call according to the host’s response. If the time expires before the host computer receives a response, the switch routes the call to a default ACD DN which the switch defines for the CDN.

The host uses to change the state of a CDN using dv-Set-CDN-State message. The CDN has the following states:

- DEFAULT - incoming calls are routed to the default ACD DN
- CONTROLLED - incoming calls are handled by the host computer
- REVERT TO DEFAULT - all incoming calls and the existing calls in the CDN queues are routed to the default ACD DN

Initially, all of the CDNs are in the DEFAULT state. The host computer must send a dv-Set-CDN-State message in order to set the state of a CDN to controlled.

Once a party calls a CDN that has a state of controlled, the party waits in a queue for the host computers routing instruction. Then the host receives a dv-Call-Queued-U message. The switch allows a datafilled delay for the host computer (e.g., ICCM) to respond to a new call message. The datafilled delay is ICCM_RESPTM. If the host computer does not respond within the time frame the host receives a dv-Call-Released-U message. Then the default ACD DN services the call.

After an originating party calls a CDN and the host computer routes the call and the terminator answers, the host no longer controls the originating party.

After associating a CDN and if the state of the CDN is controlled, the following call events are reported to the host computer:

- dv-Call-Queued-U notifies the host computer that a call is placed in the incoming call queue of a CDN ACD group.

- dv-Call-Released-U notifies the host computer when the DMS-100 switch or host releases an outgoing or incoming call (gone on-hook, or hit a release key). Also notifies the host when a Route-Call message takes out the call in a CDN queue due to re-route instructions or to CDN timer expiration.

CDNs use the dv-Resource-Query message. In order to use the dv-Resource-Query message for a CDN, the QueryAddress parameter must contain the CDN.

4.11.1.1dv-Set-CDN-State parameters

At the SCAI communication protocol level all of these parameters are optional. While at the ECM application level, CDN and state are mandatory parameters.

The Set CDN state message which the customer subscribes to always requires sending the CDN and state parameters.

CDN identifies the address of the CDN, which is in station number format.

state identifies the state of the CDN, which is default, controlled or revert to default.

4.11.1.2 dv-Set-CDN-State Return Error

setCDNStateError identifies the type of error that causes the set CDN state request to fail.

4.11.1.3 dv-Set-CDN-State error conditions

If the request to set the CDN state is unsuccessful the switch sends a Return-Error (e.g., negative acknowledgment) message to the host indicating the failure is due to one of the following reasons:

> ***Note:*** The information in braces indicates the error message parameter.

1 The DN specified in the CDN parameter is not datafilled as a CDN {invalid parameter: invalid CDN}.

2 The required CDN parameter is not included in the message {missing parameter: missing CDN}.

3 The Set CDN State message is not subscribed to at the link-level. Subscription to messages is done via table SCAIPROF and table SCAISSRV {notAllowed}.

4 The required state parameter are not included in the message {missingParameter: missing State}.

5 The DN specified in the CDN parameter is not in the same customer group as the host-switch session {invalid parameter}.

6 The DN specified in the CDN parameter is unknown to the switch {unknownCDN}.

7 The length of the DN specifies in the CDN parameter is not valid {invalid parameter: invalid CDN}.

When the switch can not decode or interpret the incoming request, the switch sends a Return Reject message back to the host and no call-related actions take place. For example, if a mandatory parameter is not included in message, then switch sends a Return Reject back to the host. Also if the service version is prior to SCAI10 a the switch sends a Return Reject back to the host.

4.11.1.4 dv-Set-CDN-State ASN.1 definition

```
dv-Set-CDN-State                OPERATION
                                ARGUMENT SetCdnStateArgument
                                RESULT
                                ERRORS{missing-parameter,
                                       invalid-CDN,
                                       notAllowed,
                                       unknownCDN,
                                       resourcesUnavailable}
::= 37
SetCdnStateArgument         ::= SEQUENCE
{CDN              [0] AddressType                   OPTIONAL,
state             [1] IMPLICIT ENUMERATED OPTIONAL
{default          (1)           OPTIONAL,
controlled        (2)           OPTIONAL,
revertToDefault   (3)           OPTIONAL

AddressType                 ::= CHOICE
 {positionID                [0] IMPLICIT  INTEGER,
   dialedDigits             [1] IMPLICIT  IA5String,
   stationNumber            [2] IMPLICIT  OCTET STRING,
   q931Address              [3] Q931AddressType}

Q931AddressType             ::= SEQUENCE
{numberTypeNumberPlan  [0]IMPLICITOCTET STRING OPTIONAL,
 presentationScreeningIndicator [1]IMPLICITOCTET STRING OPTIONAL,
 digits                     [2] IMPLICITIA5STRING OPTIONAL}

missingParameter ERROR::=1
    PARAMETERMissingParameter

MissingParameter            ::= SEQUENCE
{missingParameterType       [0] IMPLICIT ENUMERATED
{missingCDN              (0)}    OPTIONAL}
missingState             (1)}    OPTIONAL}

invalidParameterERROR::=2
    PARAMETERInvalidParameter

InvalidParameter         ::=     SEQUENCE
{invalidParameterType    [0]     IMPLICIT ENUMERATED
{invalidCDN              (0)}    OPTIONAL}
invalidState             (1)}    OPTIONAL}

notAllowed         ERROR::=3
unknownCDN         ERROR::=4
```

4.11.1.5 dv-Set-CDN-State: coded example

INVOKE DMS <===== HOST

Table 129 Set CDN State Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	19		INVOKE	Message length is 25 bytes.
02	01	0D	InvokeID	Invoke ID is 13.
02	01	25	Operation	Operation value for dv-Set-CDN-State is decimal 37.
30	11		SetCDNState Argument	Type sequence and 17 bytes long.
A0	0C		CDN	CDN is a 12 byte constructor.
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	The station number is 919-782-3640.
81	01	02	State	The state of the CDN is set to controlled.
Hex Dump = A1 19 02 01 0D 02 01 25 30 11 A0 0C 82 0A 39 31 39 37 38 32 33 36 34 30 81 01 02				

RETURN-RESULT DMS =====> HOST

Table 130 Set CDN State return result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	the message length is 3 bytes decimal
02	01	0D	InvokeID	its Invoke ID is decimal 13
Hex dump = A2 03 02 01 0D				

4.11.2 dv-Reassign-Agent Message

This activity provides an ICM message to reassign the supervisor of an ACD agent position. The result of this activity moves an ACD agent from one subgroup to another subgroup. When the host computer issues a dv-Reassign-Agent message to change the agent subgroup, after validation, the request results in datafill modification of the corresponding tables.

For mapped feature keys, reassign the following keys when reassigning an agent to a new subgroup and supervisor:

- CLSUP - the call supervisor key on the agent set
- AAK - the answer agent key on the supervisor set
- EMK - the emergency key on the agent set
- AEMK- the answer emergency key on the supervisor set

The following applies to agent reassignment:

- An agent position that has moved to a new subgroup with a new supervisor that presses the CLSUP key rings the new supervisor with a different incalls (DIFFINC) of Y. The agent position rings the old supervisor with a DIFFINC of N.
- An agent position that has moved to a new subgroup with a new supervisor that presses the EMK key rings the new supervisor with a DIFFINC of Y. The agent position rings the old supervisor with a DIFFINC of N.
- A supervisor still monitors an agent position after the position has moved to a new subgroup, if that position had an agent status lamp (ASL) in the original subgroup.

4.11.2.1dv-Reassign-Agent parameters

The host sends the dv-Reassign-Agent message to the DMS-100 switch to request a supervisor chance for the specified agent. The message requires the following two parameters:

AgentPosid identifies the agent posid for assignment to a new supervisor.

NewSuperPosid identifies the supervisor Posid for the agent reassignment

4.11.2.2 Validations of dv-Reassign-Agent

The switch performs a series of checks before processing the Reassign Agent message. The switch verifies:

1 The Service Version is at least SCA113 (The dv-Application-Logon message contains a parameter, service version, which uniquely identifies the protocol version stream of messages which will be sent during the application logged on session).

2 The ICM session subscribes to the ICCM category and the new REASNAGT function sends the parameters in it.

3 The RO contains at least one of the following parameter:

- AgentPosid
- NewSupervisorPosid

4 The AgentPosid parameter is a valid ACD agent.

5 The NewSupervisorPosid is a valid ACD supervisor.

6 The agent is logged in.

4.11.2.3 dv-Reassign-Agent error conditions

If the request to set the CDN state is unsuccessful the switch sends a Return-Error (e.g., negative acknowledgment) message to the host indicating the failure is due to one of the following reasons:

Note: The information in braces indicates the error message parameter.

1 {invalid parameter: invalid Supervisor PosID}

- the agent and supervisor posid are not associated with the same session
- reassignment of an agent with options CLSUP and EMK and DIFFINC=N
- the agent posid belongs to an ecm session but the supervisor posid does not

2 {invalid parameter: invalid Agent PosID}

- the agent posid and the ecm session do not belong to the same customer group

3 {Agent_not_logged_in}

  - the agent posid is not logged into the DMS-100 switch during reassignment

4 {Max_agents_exceeded}

  - a customer group contains the maximum allowed number of agents and cannot accept the reassignment

4.11.2.4 dv-Reassign-Agent ASN.1 definition

```
dv-Reassign-Agent              OPERATION
                               ARGUMENT ReassignAgentArgument
                               RESULT
                               ERRORS{missing-parameter,
                                      invalid_parameter,
                                      not_allowed,
                                      max_agents_exceeded,
                                      agent_not_logged_in}
::= 46
ReassignAgentArgument     ::=  SEQUENCE

 {agentPosID              [1]  IMPLICIT  INTEGER OPTIONAL,
  newSuperPosID}          [2]  IMPLICIT  INTEGER OPTIONAL}


missingParameter          ERROR::=1
                          PARAMETER       Reasn_Error_Parm



invalidParameter          ERROR::=2
                          PARAMETER       Reasn_Error_Parm



notAllowed                ERROR::=3

max_agents_exceeded       ERROR::=4

agent_not_logged_in       ERROR::=5
```

4.11.2.5 dv-Reassign_Agent: coded example

The following dv_Reassign_agent message coded example shows an agent with posid 5021 assigned to a new Supervisor with Posid 5011. The Invoke ID is 3. This example has the message delivered when the scai version is above scai13 and the parameter is subscribed through table control.

INVOKE DMS <===== HOST

Table 131 dv-Reassign-Agent invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0D		INVOKE	Message length is14 bytes decimal .
02	01	0D	InvokeID	Invoke ID is 3.
02	01	25	Operation	Operation value for dv-Reassign-Agent is 46 decimal .
30	08		ReassignAgent Argument	Type sequence and decimal 18 bytes long.
80	02		agentPosID	agent PosID of 5011
81	01	02	newSuperPosID	supervisor PosID of 5021
Hex Dump = A1 10 02 01 03 02 01 2D 30 08 80 02 13 93 81 02 13 9D				

RETURN-RESULT DMS ====> HOST

Table 132 dv-Reassign-Agent return result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	03		RETURN RESULT	the message length is 3 bytes decimal
02	01	05	InvokeID	its Invoke ID is decimal 15
Hex dump = A2 03 02 01 05				

RETURN-ERROR DMS=====>HOST

Table 133 dv-Reassign-Agent invoke coded example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	08		RETURN-ERROR	this message is 8 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	11	ErrorValue	the ErrorValue is MissingParameter[45]
02	01	47	Missing Parameter	the error is Missing parameter 67 decimal (missing Newsuperposid)
Hex Dump = A3 08 02 01 03 02 01 11 30 02 01 47				

4.12 ACD TAPI Extensions

ICM uses a dv-Call-Progress-U message to comply with Telephony Applications Programming Interface (TAPI) Class 5 event messaging. In NA12, the DMS-100 switch sends a dv-Call-Progress-U message to the customer's host computer to report on a call's progress. The message reports on orginating call states. A subset of the call states can apply to either ACD or MDC/RES lines. The list that follows contains the ACD dv-Call-Progress-U call notification states sent from the switch:.

- digit collection
- busy
- ringback
- connected
- recorded announcement from an ACD group
- music from an ACD group
- silence from an ACD group

The dv-Appl-Logon message contains a parameter entitled Service Version, which uniquely identifies the version of messages sent during a session. The dv-Call-Progress-U messages require Service Version SCAI14, or higher, for delivery from the switch to the host.

4.12.1 ACD dv-Call-Progress-U (operation value “2F”)

The DMS-100 switch sends the Class 5 operation asynchronous or unsolicited message to the host to report different states of a call's progress. The dv-Call-Progress-U message is valid for Meridian ACD CompuCALL and MDC/RES options.

4.12.1.1 ACD dv-Call-Progress-U parameters

At the SCAI communication protocol level all of these parameters are optional. While at the ECM application level, CDN and state are mandatory parameters.

The Set CDN state message which the customer subscribes to always requires sending the CDN and state parameters.

NetworkCallID ICM generated call ID composed of a Network Node ID (taken from table SCAIGRP) and a Local Call Id (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session environment; otherwise the call ID already assigned to this call will be used.

OrigDevInfo This parameter identifies the address of the party which originated the call. If the call originated from a Residential or Centrex line, the DeviceId in station number format and the name of the originator will be provided. If the call is originated by an ACD agent, the ACD group and position ID will be provided.

TermDevInfo This parameter identifies the address of the terminating party. For line-to-line calls, if the call is not terminating to an ACD group, the DeviceID in station number format and the name of the terminator will be provided. If the call does terminate to an ACD group and an agent is being alerted, the primary ACD group and the position ID will be provided. If no ACD agents are available and the call is placed in queue, the primary ACD group will be provided.

DialedDigits identifies the digits dialed.

ACD DN identifies the ACD DN dialed by the caller. It may be the primary ACD DN or a supplementary ACD DN.

ForwardInfo Information about forwarding of call. Includes the firstFwdNumber, firstFwdReason, lastFwdNumber, and lastFwdReason.

CallType identifies the type of call, if not a “normal” call.

FirstLegDevInfo identifies the address of the first leg of the call if the originator is already active in a call. If the first leg is a Residential or Centrex line, the DeviceId in station number format and the name will be provided. If the first leg is an ACD agent, the ACD group and position ID will be provided.

4.12.1.2 Validations of dv-Call-Progress-U

The following setup and conditions must be met in order for the host application to receive dv-Call-Progress-U messages:

1. An ICM switch-host session is established. A successful dv-Appl-Logon request was sent by the host computer.
2. The line which the host application is monitoring must be associated to the session via dv-DN-Associate.
3. The switch-host session is subscribed to the Call Progress event messages
4. The SCAI Service Version is at least SCAI14.

4.12.1.3 dv-Call-Progress-U ASN.1 encoding

```
dv-Call-Progress-U    OPERATION
                      ARGUMENT CallProgressArgument
::= 47

CallProgressArgument::= SEQUENCE
```

```
    {   networkCallID[0] IMPLICIT SEQUENCE
        {   networkNodeID    [0] IMPLICIT INTEGEROPTIONAL,
             localCallID       [1] IMPLICIT INTEGEROPTIONAL}
                                          OPTIONAL,
        OrigDevInfo   [1] DeviceInfoType    OPTIONAL,
        CallProgressParm[2] CallProgressParmTypeOPTIONAL,
        OrigCallStatus[3]      OrigCallStatusTypeOPTIONAL}

DeviceInfoType        ::= SEQUENCE
    {   acdGroup       [0] IMPLICIT OCTET STRINGOPTIONAL,
        acdPositionID [1] IMPLICIT INTEGEROPTIONAL,
        DevAddrName[2] DeviceAddressTypeOPTIONAL}

CallProgressParmType::= CHOICE
    {   DigitCollectionParm[0] AddressTypeOPTIONAL,
        FarEndAnsweredParm[1] TerminatorInfoTypeOPTIONAL,
        FarEndBusyParm[2] AddressTypeOPTIONAL,
        FarEndRingingParm[3] TerminatorInfoTypeOPTIONAL,
        RANParm       [4] TerminatorInfoTypeOPTIONAL,
        MusicParm     [5] TerminatorInfoTypeOPTIONAL,
        SilenceParm   [6] TerminatorInfoTypeOPTIONAL}

OrigCallStatusType    ::= SEQUENCE
    {   CallType        [0] IMPLICIT ENUMERATED
        {   callTransferred     (1),
            callOverflowed     (2),
            callRedirected      (3),
            callForwarded       (4),
            callConsult          (5),
            callConference      (6),
            callRouted           (8)}     OPTIONAL,
        FirstLegDevInfo[1] DeviceInfoTypeOPTIONAL}

TerminatorInfoType    ::= SEQUENCE
    {   TermDevInfo  [0] DeviceInfoTypeOPTIONAL,
        acdDN           [1] IMPLICIT OCTET STRINGOPTIONAL,
        forwardingParty[2] ForwardingPartyTypeOPTIONAL}

DeviceAddressType    :: =SET OF
    {   dialedDigits    [1]  IMPLICIT IA5String,OPTIONAL,
        stationNumber[2]  IMPLICIT OCTET STRINGOPTIONAL,
        name              [4]  IMPLICIT IA5StringOPTIONAL}

AddressType            ::= CHOICE
    {   positionID      [0] IMPLICIT INTEGER,
        dialedDigits    [1] IMPLICIT IA5 STRING,
        stationNumber[2] IMPLICIT OCTET STRING,
        q.931Address  [3] Q.931 AddressType}

Q.931 AddressType    ::= SEQUENCE
    {   numberTypeNumberPlan[0] OCTET STRINGOPTIONAL,
        presentationScreeningIndicator[1] OCTET STRINGOPTIONAL,
        digits             [2] IA5 STRING        OPTIONAL}

ForwardingPartyType :: = IMPLICIT SEQUENCE
    {   firstFwdNumber[0]  AddressTypeOPTIONAL,
        firstFwdReason[1]  IMPLICIT ENUMERATED
        {   unknown            (0),
            userBusy           (1),
            noReply            (2),
            unconditional      (3)  }   OPTIONAL,
        lastFwdNumber[2]  AddressTypeOPTIONAL,
        lastFwdReason[3]  IMPLICIT ENUMERATED
        {   unknown            (0),
            userBusy           (1),
            noReply            (2),
            unconditional      (3)  }   OPTIONAL}
```

4.12.1.4 Digit collection: coded example

The following dv_Call_Progress_U example shows the digits dialed call progress event. A subscriber at DN 613-621-2000 dials 621-3000. If the the caller had been an ACD agent, the ACD group and position ID replaces the stationNumber and name information elements. Intra-switch calls that are routed to a feature do not report the digit collection event, except those calls routed to an ACD group or agent

INVOKE DMS =====> HOST

Table 134 Digit Collection invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	37		INVOKE	this message is 55 bytes decimal
02	01	04	InvokeID	the InvokeID is 4 decimal
02	01	2F	Operation	the operation value for dv-Call-Progress-U is 47 decimal.
30	2F		Argument	the CallProgressArgument is of type sequence and is 47 bytes long decimal
A0	06		NetworkCallId	the NetworkCallID is of type constructor and is 6 bytes long
80	01	00	NetworkNodeId	the NetworkNodeId is 0
81	01	3C	LocalcallId	the LocalcallId is 60 decimal
A1	18		OrigDevInfo	the OrigDevInfo is a constructor 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
82	0A	36 3133 36 32 31 32 30 30 30	stationNumber	the calling DN is 6136212000
84	08	4A 4F 45 20 50 54 59 41	name	the caller's name (Joe Ptya) is 8 bytes long
A2	0B		CallProgressParm	the CallProgressParm is of type constructor and is 11 bytes long decimal
A0	09		DigitCollectionParm	the DigitCollectionParm is of type constructor and is 9 bytes long
81	07	36 32 31 33 30 30 30	dialedDigits	the digits dialed by the caller are 6213000
Hex Dump = A1 37 02 01 04 02 01 2F 30 2F A0 06 80 01 00 81 01 3C A1 18 A2 16 82 0A 36 31 33 36 32 31 32 30 30 30 84 08 4A 4F 45 20 50 54 59 41 A2 0B A0 09 81 07 36 32 31 33 30 30 30				

4.12.1.5 Busy: coded example

The following dv_Call_Progress_U example shows the busy call progress event. A subscriber at DN 613-621-2000 dials 621-3000, which produces a terminator busy message. If the the caller had been an ACD agent, the ACD group and position ID replaces the stationNumber and name information elements. Intra-switch calls that are routed to a feature do not report the busy event, except those calls routed to an ACD group or agent

INVOKE DMS =====> HOST

Table 135 Far End Busy invoke coded example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	37		INVOKE	this message is 55 bytes decimal
02	01	04	InvokeID	the InvokeID is 4 decimal
02	01	2F	Operation	the operation value for dv-Call-Progress-U is 47 decimal.
30	2F		Argument	the CallProgressArgument is of type sequence and is 47 bytes long decimal
A0	06		NetworkCallId	the NetworkCallID is of type constructor and is 6 bytes long
80	01	00	NetworkNodeId	the NetworkNodeId is 0
81	01	3C	LocalcallId	the LocalcallId is 60 decimal
A1	18		OrigDevInfo	the OrigDevInfo is a constructor 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
82	0A	36 3133 36 32 31 32 30 30 30	stationNumber	the calling DN is 6136212000
84	08	4A 4F 45 20 50 54 59 41	name	the caller's name (Joe Ptya) is 8 bytes long
A2	0B		CallProgressParm	the CallProgressParm is of type constructor and is 11 bytes long decimal
A2	09		FarEndBusyParm	the FarEndBusyParm is of type constructor and is 11 bytes long
81	07	36 32 31 33 30 30 30	dialedDigits	the digits dialed by the caller are 6213000
Hex Dump = A1 37 02 01 04 02 01 2F 30 2F A0 06 80 01 00 81 01 3C A1 18 A2 16 82 0A 36 31 33 36 32 31 32 30 30 30 84 08 4A 4F 45 20 50 54 59 41 A2 0B A2 09 81 07 36 32 31 33 30 30 30				

4.12.1.6 Ringback: coded example

The following dv_Call_Progress_U example shows the ringback call progress event. A subscriber at DN 613-621-2000 dials 621-3000. In this example, DN 613-621-2000 is busy which causes the call to forward to DN 613-621-4000 and the originating subscriber to receive ringback tone. which produces a terminator busy message. Inter-switch (trunk) calls do not provide the ringback event message. Intra-switch calls that are routed to a feature do not report the ringback event, except those calls routed to an ACD group or agent.

If the the caller had been an ACD agent, the ACD group and position ID replaces the stationNumber and name information elements for the OrigDevInfo fields. If the call terminated to an ACD group and an agent was alerted, the primary ACD group and the position ID would appear in the TermDevInfo fields. If no ACD agents were available and the call was placed in queue, the primary ACD group appears.

An ACD DN information element appears if the intended terminator was an ACD agent. The field contains either a primary or supplementary ACD DN. If the the first leg of the call had been an ACD agent, the ACD group and position ID replaces the stationNumber and name information elements for the OrigDevInfo fields.

INVOKE DMS =====> HOST

Table 136 Far End Ringing invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	63		INVOKE	this message is 99 bytes decimal
02	01	04	InvokeID	the InvokeID is 4 decimal
02	01	2F	Operation	the operation value for dv-Call-Progress-U is 47 decimal.
30	5B		Argument	the CallProgressArgument is of type sequence and is 91 bytes long decimal
A0	06		NetworkCallId	the NetworkCallID is of type constructor and is 6 bytes long
80	01	00	NetworkNodeId	the NetworkNodeId is 0
81	01	3C	LocalcallId	the LocalcallId is 60 decimal
A1	18		OrigDevInfo	the OrigDevInfo is a constructor 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
82	0A	36 3133 36 32 31 32 30 30 30	stationNumber	the calling DN is 6136212000
84	08	4A 4F 45 20 50 54 59 41	name	the caller's name (Joe PtyA) is 8 bytes long
A2	37		CallProgressParm	the CallProgressParm is of type constructor and is 55 bytes long decimal
A2	35		FarEndRingingParm	the FarEndRingingParm is of type constructor and is 53 bytes long
A0	18		TermDevInfo	the TermDevInfo is a constructor 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
82	0A	36 3133 36 32 31 34 30 30 30	stationNumber	the terminating (forwarded to) DN is 6136214000
84	08	4A 4F 45 20 50 54 59 43	name	the terminating party's name (Joe PtyC) is 8 bytes long
A2	19		forwardingParty	the forwardingParty is of type constructor and is 25 bytes long decimal
A0	14		firstFwdNumber	the firstFwdNumber is of type constructor and is 20 bytes long decimal
A0	12		q.931Address	the q.931Address is a constructor 18 bytes long decimal
80	01	21	numberTypeNumber-Plan	the numberTypeNumberPlan is a value of 21 hex
81	01	80	presentationScreen-ingIndicator	the presentationScreeningIndicator is a value of 80 hex
82	0A	36 31 33 36 32 31 33 30 30 30	digits	the digits dialed by the caller are 6136213000
81	01	01	firstFwdReason	the firstFwdReason is userBusy(1)
Hex Dump = A1 63 02 01 04 02 01 2F 30 5B A0 06 80 01 00 81 01 3C A1 18 A2 16 82 0A 36 31 33 36 32 31 32 30 30 30 84 08 4A 4F 45 20 50 54 59 41 A2 37 A3 35 A0 18 A2 16 82 0A 36 31 33 36 32 31 34 30 30 30 84 08 4A 4F 45 20 50 54 59 43 A2 19 A0 14 A3 12 80 01 21 81 01 80 82 0A 36 31 33 36 32 31 33 30 30 30 81 01 01				
			—end—	

4.12.1.7 Connected: coded example

The following dv_Call_Progress_U example shows the connected call progress event. A subscriber at DN 613-621-2000 dials 621-3000. In this example, the originating subscriber makes a consult call to another subscriber at DN 613-621-4000.

If the the caller had been an ACD agent, the ACD group and position ID replaces the stationNumber and name information elements for the OrigDevInfo fields. If the call terminated to an ACD group and answered by an agent, the primary ACD group and the position ID appear in the TermDevInfo fields.

An ACD DN information element appears if the intended terminator was an ACD agent. The field contains either a primary or supplementary ACD DN. If the first leg of the call had been an ACD agent, the ACD group and position ID replaces the stationNumber and name information elements for the OrigDevInfo fields.

INVOKE DMS =====> HOST

Table 137 Connected event invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	67		INVOKE	this message is 103 bytes decimal
02	01	04	InvokeID	the InvokeID is 4 decimal
02	01	2F	Operation	the operation value for dv-Call-Progress-U is 47 decimal.
30	5F		Argument	the CallProgressArgument is of type sequence and is 95 bytes long decimal
A0	06		NetworkCallId	the NetworkCallID is of type constructor and is 6 bytes long
80	01	00	NetworkNodeId	the NetworkNodeId is 0
81	01	3C	LocalcallId	the LocalcallId is 60 decimal
A1	18		OrigDevInfo	the OrigDevInfo is a constructor 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
82	0A	36 3133 36 32 31 32 30 30 30	stationNumber	the calling DN is 6136212000
84	08	4A 4F 45 20 50 54 59 41	name	the caller’s name (Joe PtyA) is 8 bytes long
A2	1C		CallProgressParm	the CallProgressParm is of type constructor and is 28 bytes long decimal
A1	1A		FarEndAnsweredParm	the FarEndAnsweredParm is of type constructor and is 26 bytes long
A0	18		TermDevInfo	the TermDevInfo is a constructor 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
82	0A	36 3133 36 32 31 34 30 30 30	stationNumber	the consult DN is 6136214000
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
84	08	4A 4F 45 20 50 54 59 43	name	the consult party’s name (Joe PtyC) is 8 bytes long
A3	1D		OrigCallStatus	the OrigCallStatus is of type constructor and is 29 bytes long decimal
80	01		CallType	the CallType is a callConsult (5)
A1	18		firstLegDevInfo	the firstLegDevInfo is of type constructor and is 24 bytes long decimal
A2	16		DevAddrName	the DevAddrName is a constructor 22 bytes long decimal
82	0A	36 31 33 36 32 31 33 30 30 30	stationNumber	the stationNumber dialed by the caller are 6136213000
84	08	4A 4F 45 20 50 54 59 42	name	the terminating party’s name (Joe PtyB) is 8 bytes long
Hex Dump = A1 67 02 01 04 02 01 2F 30 5F A0 06 80 01 00 81 01 3C A1 18 A2 16 82 0A 36 31 33 36 32 31 32 30 30 30 84 08 4A 4F 45 20 50 54 59 41 A2 1C A1 1A A0 18 A2 16 82 0A 36 31 33 36 32 31 34 30 30 30 84 08 4A 4F 45 20 50 54 59 43 A3 1D 80 01 05 A1 18 A2 16 82 0A 36 31 33 36 32 31 33 30 30 30 84 08 4A 4F 45 20 50 54 59 42				
			—end—	

4.13 CompuCALL MDC/RES application service functions

This section describes the CompuCALL messages the switch uses to notify the host of events.

4.13.1CompuCALL MDC/RES coordinated voice and data delivery

This section describes the MDC/RES call events messages. This application service capability for MDC/RES lines includes four application service functions:

- dv-Call-Offered-U
- dv-Call-Answered-U
- dv-Call-Released-U
- dv-Set-Offhook-U

4.13.2 MDC/RES dv-Call-Offered-U (operation value 5)

The switch sends this RO of Operation Class 5 to the host when an incoming CompuCALL call is about to be offered to a station. This message is sent when an incoming call is offered to an associated DN.

```
ASN.1 Encoding
        dv-Call-Offered-U           OPERATION
                                    ARGUMENT            CallOfferedArgument

                        ::= 5

        CallOfferedArgument         ::=     SEQUENCE
         {  networkCallID           [0] IMPLICIT SEQUENCE
             {  networkNodeID           [0] IMPLICIT INTEGER        OPTIONAL,
                localCallID             [1] IMPLICIT INTEGER        OPTIONAL}
                                                                    OPTIONAL,

                    ,
                        origDeviceID [3]  DeviceAddressTypeOPTIONAL,
                        origChargeNumber[4]  IMPLICIT OCTET STRINGOPTIONAL,
                        callHistoryInfo [5]  IMPLICIT SEQUENCE
                                {callType           [0]  IMPLICIT  ENUMERATED
                                    {callTransferred    (1),
                                     callOverflowed     (2),
                                     callRedirected     (3),
                                     callForwarded      (4),

                            callConsult(5),

                            callConference(6)}}OPTIONAL,
                                origInboundDN      [1] IMPLICIT DeviceAddressType OPTION-
                        AL,
          prevApplicationID [2]  IMPLICIT INTEGER OPTIONAL,
          hostCallData      [3]  IMPLICIT OCTET STRINGOPTION-
AL}
                                                     OPTIONAL,
        callMode            [7]  IMPLICIT ENUMERATED
          {callWaited       (1)                    OPTIONAL,

 teenService(2)OPTIONAL}
     destDeviceID      [8]   AddressType          OPTIONAL,
     forwardingParty   [9]   ForwardingPartyTypeOPTIONAL}

DeviceAddressType     :: =   SET OF
     {  dialedDigits            [1]  IMPLICIT IA5String,OPTIONAL,
        stationNumber           [2]  IMPLICIT OCTET STRING,OPTION-
AL,
        name                    [4]  IMPLICIT IA5StringOPTIONAL}

AddressType           :: =   CHOICE
     {  positionID      [0]  IMPLICIT INTEGER,
        dialedDigits    [1]  IMPLICIT IA5String,
        stationNumber   [2]  IMPLICIT OCTET STRING,
        q931Address     [3]  Q931AddressType}

ForwardingPartyType::=       IMPLICIT SEQUENCE
     {  firstFwdNumber  [0]  AddressType,
        firstFwdReason  [1]  IMPLICIT ENUMERATED
          {unknown           (0),
            userBusy         (1),
            noReply          (2),
            unconditional    (3)}                   OPTIONAL,
        lastFwdNumber   [2]  AddressType,
        lastFwdReason   [3]  IMPLICIT ENUMERATED
{unknown(0),
            userBusy         (1),
            noReply          (2),
            unconditional    (3)}                   OPTIONAL,
```

4.13.2.1 MDC/RES dv-Call-Offered-U: coded example 1

Following is a coded example of the dv-Call-Offered-U message. A call is originated by a residential phone that has an origination device ID of 858-1234 and the caller’s name is blocked. The call is offered to a phone that has a destination device ID of 504 858-4321. The invoke ID is 3. This example has the calling name parameter subscribed to through datafill.

INVOKE **DMS =====> HOST**

Table 138 dv-Call-Offered-U invoke coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	35		INVOKE	the message length is 53 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	05	Operation	its operation value is dv-Call-Offered-U [5] decimal
30	2D		Argument	CallOfferedArgument is of type sequence and is 48 bytes long decimal
A0	06		networkCallID	the networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	03	localCallID	the localCallID is 3
A3	15		origDeviceID	the origDeviceID is of type constructor and is bytes 21 long
81	07	38 35 38 31 32 33 34	dialedDigits	the dialable number is 858-1234
82	0A	35 30 34 38 35 38 31 32 33 34	stationNumber	the DN of the phone is 504-858-1234
A8	0C		destDeviceID	the destDeviceID is a constructor and is 12 bytes long
82	0A	35 30 34 38 35 38 34 33 32 31	stationNumber	the DN of the phone is 504-858-4321
Hex dump = A1 35 02 01 03 02 01 05 30 2D A0 06 80 01 01 81 01 03 A3 15 81 07 38 35 38 31 32 33 34 82 0A 35 30 34 38 35 38 31 32 33 34 A8 0C 82 0A 35 30 34 38 35 38 34 33 32 31				

4.13.2.2 MDC/RES dv-Call-Offered-U: coded example 2

Following is a coded example of the dv-Call-Offered-U message. A call is originated by a residential phone with an origination device ID of 858-1234 and the caller's name is SHANANA JONES. The call is offered to a phone with a station number of (504) 858-4321. The invoke ID is 3. This example has the calling dialable number parameter subscribed to through table control.

INVOKE DMS <===== HOST

Table 139 dv-Call-Offered-U example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	44		INVOKE	the message length is 68 bytes decimal
02	01	03	..InvokeID	its InvokeID is 3
02	01	05	..Operation	its operation value is dv-Call-Offered-U [5] decimal
30	3C		..Argument	CallOfferedArgument is of type sequence and is 60 bytes long decimal
A0	06		..networkCallID	the networkCallID is of type constructor and is 6 bytes long
80	01	01	.....networkNodeID	the networkNodeID is 1
81	01	03	...... localCallID	the localCallID is 3
A3	24		.......origDeviceID	the origDeviceID is of type constructor and is 36 bytes long
81	07	38 35 38 31 32 33 34	...... dialedDigits	the dial number is 858-1234
82	0A	35 30 34 38 35 38 31 32 33 34	......stationNumber	the DN of the phone is 504-858-1234
84	0D	53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53	......name	the customer's name is SHANANA JONES of length 13
			—continued—	

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A8	0C		....destDeviceID	the destDeviceID is a constructor length of 12
82	0A	35 30 34 38 35 38 34 33 32 31	......stationNumber	the DN of the phone is 504-858-4321
Hex dump = A1 44 02 01 03 02 01 05 30 3C A0 06 80 01 01 81 01 03 A3 24 81 07 38 35 38 31 32 33 34 82 0A 35 30 34 38 35 38 31 32 33 34 84 0D 53 48 41 4E 41 4E 41 20 4A 4F 45 53 A8 0C 82 0A 35 30 34 38 35 38 34 33 32 31				
			—end—	

4.13.2.3 MDC/RES dv-Call-Offered-U: coded example 3

Following is a coded example of the dv-Call-Offered-U message. A call is originated by a residential phone that has an origination device ID of 858-1234. The terminating line is forwarded to 858-5252. The invoke ID is 3. This example has the calling number parameter subscribed to through table control.

INVOKE DMS =====> HOST

Table 140 dv-Call-Offered-U example 3

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	54		INVOKE	the message length is 84 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	05	Operation	its operation value is dv-Call-Offered-U [5] decimal
30	4C		Argument	CallOfferedArgument is of type sequence and is 76 bytes long decimal
A0	06		networkCallID	the networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	03	localCallID	the localCallID is 3
A3	24		origDeviceID	the origDeviceID is of type constructor and is bytes 36 long
81	07	38 35 38 3132 33 34	dialedDigits	the dial number is 858-1234
82	0A	35 30 34 38 35 38 31 32 33 34	stationNumber	the DN of the phone is 504-858-1234
84	0D	53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53	name	the customer’s name is SHANANA JONES of length 13
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
A5	0E		callHistoryInfo	callHistoryInfo is a constructor of length 14
80	01	04	callType	the call is forwarded
A1	09		origInboundDN	origInBoundDN is a constructor of length 9
81	07	38 35 38 34 33 32 31	dialedDigits	the DN is 858-4321
A8	0C		destDeviceID	the destDeviceID is a constructor of length 12
82	0A	35 30 34 38 35 38 35 32 35 32	stationNumber	the DN of the phone is 504-858-5252
Hex dump = A1 54 02 01 03 02 01 05 30 47 A0 06 80 01 01 81 01 03 A3 24 81 07 38 35 38 31 32 33 34 82 0A 35 30 34 38 35 38 31 32 33 34 84 0D 53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53 A5 0E 80 01 04 A1 09 81 07 38 35 38 34 33 32 31 A8 OC 82 0A 35 30 34 38 35 38 35 32 35 32				
			—end—	

4.13.2.4 MDC/RES dv-Call-Offered-U: coded example 4

Following is a coded example of the dv-Call-Offered-U message. A call is originated by a residential phone that has an origination device ID of 621-1501. The terminating line, 621-1502 is CFBand forwards to voice mail 621-1504. The invoke ID is 8.

INVOKE DMS =====> HOST

Table 141 dv-Call-Offered-U example 4

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	69		INVOKE	the message length is 105 bytes decimal
02	01	08	InvokeID	its Invoke ID is 8
02	01	05	Operation	its operation value is dv-Call-Offered-U [5] decimal
30	61		Argument	CallOfferedArgument is of type sequence and is 97 bytes long decimal
A0	06		networkCallID	the networkCallID is of type constructor and is 6 bytes long
80	01	00	networkNodeID	the networkNodeID is 0
81	01	22	localCallID	the localCallID is 34
A3	16		origDeviceID	the origDeviceID is of type constructor and is bytes 22 long
81	08	39 36 32 3131 35 30 31	dialedDigits	the dial number is 9-621-1501
82	0A	36 31 33 36 32 31 31 35 30 31	stationNumber	the calling party DN is 504-858-1234
A5	1E		callHistoryInfo	callHistoryInfo is a constructor of length 30
80	01	04	callType	the call is forwarded
A1	16		origInboundDN	origInBoundDN is a constructor of length 22
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
81	08	39 36 32 31 31 35 30 32	dialedDigits	the DN for the called party is 9-621-1502
82	0A	36 31 33 36 32 31 31 35 30 32	stationNumber	the called party’s station number is 613-621-1502
82	01	01	prevApplicationeID	the previous application ID is 1
A8	0C		destDeviceID	the destDeviceID is a constructor of length 12
82	0A	36 31 33 36 32 31 31 35 30 34	stationNumber	the voice mail port’s number is 613-621-1504
A9	11		forwardingParty	the forwardingParty is a constructor of length 17 bytes
A0	0F		firstFwdNumber	the firstFwdNumber is a constructor of length 15 bytes
82	0A	36 31 33 36 32 31 31 35 30 32	stationNumber	the first forwarded number is 613-621-1502 (the call in this example directly forwards to voice mail)
81	01	01	firstFwdReason	the forwarded reason is user busy
Hex dump = A1 69 02 01 08 02 01 05 30 61 A0 06 80 01 00 81 01 22 A3 16 81 08 39 36 32 31 31 35 30 31 82 0A 36 31 33 36 32 31 31 35 30 31 A5 1E 80 01 04 A1 16 81 08 39 36 32 31 31 35 30 32 82 0A 36 31 33 36 32 31 31 35 30 32 A8 OC 82 0A 36 31 33 36 35 38 35 32 35 32 32 31 31 35 30 34 A9 11 A0 0F 82 OA 36 31 33 36 32 31 31 35 30 32 81 01 01				
			—end—	

4.13.3 dv-Call-Offered-U parameters

A brief description of the parameters of the dv-Call-Offered-U invoke follows. If the set is ringing, dv-Call-Offered contains the following optional parameters:

NetworkCallID contains static call-related information to uniquely identify the call in the context of the customer's network for the maximum expected call duration. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

This is a CompuCALL-generated call ID composed of a network node ID (from table SCAIGRP) and a local call ID (unique call ID in the session). This is a new call id if this is the first instance of the call within this session/environment; otherwise the call ID already assigned to this call is used.

OrigDeviceID contains static call-related information to indicate the original calling number associated with a call when it enters the CompuCALL environment. For example, this may be a 10-digit North American Numbering Plan (NANP) DN associated with the calling line (or if a call from a PBX, it may be the listed DN (LDN) of the PBX) or a private number.

Provided in the new deviceAddressType format for terminating calls. The deviceAddressType lets us send a set of one or more values representing the calling device. The set includes the following parameters:

dialedDigits contains the DN in dial number delivery format

stationNumber contains directory number format

name contains the caller’s name

OrigChargeNumber The number being billed for this call. Contains static call-related information to indicate the operating company charge number (also called "ANI" - Automatic Number Identification in North America) associated with the incoming call when it enters the CompuCALL environment. A dv-Call-Offered-U message contains both the OrigCallingNumber and OrigChargeNumber parameters, if available.

CallHistoryInfo If the call were transferred, overflowed, or redirected from an ACD group or agent, or transferred from, or forwarded by an MDC customer workstation, the dv-Call-Offered-U message includes the CallHistoryInfo parameter with one or more of the following subparameters.

Each subparameter is individually datafilled in the switch as provided (if appropriate) or not provided. In other words, CallHistoryInfo parameter is present if the customer subscribed to any of the following subparameters:

CallType contains dynamic call-related information indicating if the call to the current CompuCALL party were transferred, overflowed (by ACD), redirected (by ACD), forwarded, consult, or conference

In a three-way call consult scenario, the call type will be consult.

OrigInboundDN contains static call-related information to identify the first DN at which the call is directed when it enters the CompuCALL environment. It remains the same as long as the call stays within the

CompuCALL environment even if it is subsequently transferred, overflowed, redirected, or forwarded.

In new deviceAddressType (if a transferred call, this is the DN of the party who transferred the call). Provided in the new device address type format for terminating calls.The device AddressType lets us send a set of one or more values representing the calling device. The set includes these parameters:

dialedDigits contains the DN in dial number delivery format

stationNumber contains directory number format

name contains the caller’s name

PrevApplicationID contains dynamic call-related information to identify the ApplicationID for the session of the associated environment to which the party extending the call belongs -must be on the same switch). (Note: Call extension includes call transfer, overflow, redirect, and forward).

HostCallData contains dynamic call-related information of a generic and miscellaneous nature provided by a host. If a host redirects a call, the host may provide this parameter at that time and may include the reason for redirection.

CallMode contains dynamic call-related information identifying the call’s modality when it’s offered to the party. If the call was put on Call Waiting for the MDC station while the station is busy in an existing call the dv-Call-Offered-U message includes the CallMode parameter to indicate the call is waited. This parameter applies only to MDC calls. The call mode now includes Teen Service.

callWaited is delivered if the call is waiting

teenService is delivered if the terminating Teen Service feature is active

destDeviceID the address of the terminator being alerted for this call in the station number format. Contains dynamic call-related information to identify the customer workstation to which the call is offered. It is unique in the switch and a choice parameter. The choice is StationNumber:

StationNumber identifies the specific MDC line to which the call is offered.

deviceAddressType is a new type that will enable us to send a set of one or more values representing the calling device. The set will include the following parameters:

dialedDigits which contains the DN in the dial number delivery format

stationNumber which contains the directory number format

name which contains the caller's name. When the call is local, the calling name is retrieved by accessing data via datafill. If the calling name is datafilled then the callingname will be provided in the call event parameter. When the call goes over an ISUP trunk or ISDN trunk, the calling name can be retrieved by accessing data. Note, that when the call goes over a BTUP trunk or ATUP trunk the calling name will not be retrieved. Also, name delivery using TCAP query to the service control point (SCP) database is not supported.

ForwardingParty identifies the first forwarding number and associated reason and last forwarding number and reason for forwarding. In a single forwarding scenario, the information is included in the FirstFwdNumber and FirstFwdReason subparameters. This information is provided to the host only if presentation of the forwarding DNs includes the following information:

FirstFwdNumber Identifies the number from which the first forward was made in a multiple call forwarding scenario. It is based on the Q931 format and uses the Q931Address choice.

FirstFwdReason Identifies the reason the call was forwarded the first time. It is an enumerated type with values: unknown/not available, user busy, no reply, or unconditional.

LastFwdNumber Identifies the number from which the last forward was made in a multiple call forwarding scenario. Based on the Q931 format, it uses the Q931Address choice.

LastFwdReason Identifies the reason the call was forwarded the last time (most recent). It is an enumerated type with values: unknown/not available, user busy, no reply, or unconditional.

The switch sends the dv-Call-Offered-U message to the host when a customer workstation is offered an incoming call in one of the following scenarios:

- If MDC, the call is received by the switch and is offered to the specific MDC station to which the call is directed.
- No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Offered-U message. This is the reason for the "U" (unconfirmed) in the message name. If either is received by the switch, a Reject is returned with reason “unrecognized invocation."

Furthermore, the switch ignores any reject message from the host in response to a dv-Call-Offered-U message.

4.13.4 MDC/RES dv-Call-CallingName-U

This message enhances CompuCALL calling name and number delivery service for residential or centrex lines. This is possible when the ECM option is enabled, the line subscribes to the delivery option, or the feature group exist for the office without the line subscription.

4.13.4.1 dv-Call-CallingName-U: ASN.1 encoding

The ASN.1 notations for the new/changed messages for residential lines provided by this activity are shown in this section. Following each notation is a sample X.209 hex encoding. The length fields are variable because the length depends on the variable contents. The tag values are taken from the Remote Operation Service Element (ROSE) protocol.

```
dv-Call-CallingName-U          OPERATION
                                ARGUMENT   CallNameArgument
::= 30

CallNameArgument          ::=    SEQUENCE
    {   networkCallID      [0] IMPLICIT SEQUENCE
            {networkNodeID    [0] IMPLICIT INTEGER OPTIONAL,
              localCallID        [1] IMPLICIT INTEGEROPTIONAL}
                           OPTIONAL,
        destDeviceID       [1] AddressTypeOPTIONAL,
          OrigDeviceID     [3] DeviceAddressTypeOPTIONAL,


    DeviceAddressType     :: =   SET OF
        {dialedDigits              [1] IMPLICIT IA5String,OPTIONAL,
          stationNumber            [2] IMPLICIT OCTET STRING,OPTION-
    AL,
          name                     [4] IMPLICIT IA5StringOPTIONAL}

    AddressType           :: =   CHOICE
        {  positionID      [0] IMPLICIT INTEGER,
           dialedDigits    [1] IMPLICIT IA5String,
           stationNumber   [2] IMPLICIT OCTET STRING,
           q931Address     [3] Q931AddressType}
```

4.13.4.2 Call name: coded example

Following is a coded example of the dv-Call-CallingName message. A call is originated by a RES/MDC line that has an origination device ID of 858-1234 and the caller’s name is SHANANA JONES. The call is offered to a line that has a station number of (504) 858-4321. The invoke ID is 3. This example has the calling name delivered when the mode is QUERY and the parameter is subscribed through table control.

INVOKE DMS ====> HOST

Table 142 dv-Call-CallingName-U

Hex Dump			Information Element	Comments
Tag	Len	Content		
A1	44		INVOKE	the message length is 68 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	26	Operation	its operation value is dv-Call-CallingName-U [38] decimal
30	3C		Argument	CallNameArgument is of type sequence and is 60 bytes long decimal
A0	06		networkCAllID	the networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	03	localCallID	the localCallID is 3
A3	0C		destDeviceID	the destDeviceID is a constructor length 12
82	0A	35 30 34 38 35 38 34 33 32 31	stationNumber	the DN of the phone is 504-858-4321
A8	24		OrigDeviceID	the OrigDeviceID is of type constructor and is bytes 36 long
81	07	38 35 38 31 32 33 34	dialedDigits	the dial number is 858-1234
82	0A	35 30 34 38 35 38 31 32 33 34	stationNumber	the DN of the phone is 504-858-1234
84	0D	53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53	name	the customer's name is SHANANA JONES of length 13
Hex Dump = A1 44 02 01 03 02 01 26 30 0C A0 06 80 01 01 81 01 03 A3 0C 82 0A 35 30 34 38 35 38 34 33 32 31 A8 24 81 07 38 35 38 31 32 33 34 82 0A 35 30 34 38 35 38 31 32 33 34 84 0D 53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53				

4.13.4.3 CallNamepParameters

The dv-Call-CallingName-U and dv-Call-Offered-U messages are sent only when the Name Display/Delivery mode is using QUERY mechanism. Since dv-Call-CallingName-U is sent after dv-Call-Offered-U the following parameters are required:

NetworkCallID contains static call-related information to uniquely identify the call in the context of the customer's network for the maximum expected call duration. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

OrigDeviceID includes the name the calling party.

Calling Number is the number (DN) of the caller.

Calling Name is the name of the caller.

Diallable DN is the number you dial to call back the caller.

4.13.4.4 Callname message

The dv-Call-CallingName-U message is sent in addition to the dv-Call-Offered-U message only under situations where the Name Display/Delivery mode is using QUERY mechanism.

4.13.4.5 Validations of dv-Call-CallingName-U

The switch performs a series of checks before sending the Callname message. The switch verifies:

1. The Service Version is at least SCAI10 (The dv-Application-Logon message contains a parameter, service version, which uniquely identifies the protocol version stream of messages which will be sent during the application logged on session).
2. The terminating line has ECM option.
3. The terminating line is associated to a CompuCALL session and the session to which the line belongs to, subscribes to the new dv-Call-CallingName-U message.
4. The new message is not sent if the Calling name is not available when the query time-out occurs.

4.13.4.6 Calling name and number delivery

To deliver the Calling Name and Number Delivery, dv-Call-Offered-U, and dv-Call-Answered-U messages are modified to change the interpretation of the OrigDeviceID parameter, which contains callingID for all call scenarios.

When a call is terminated on a line that is associated to a SCAI session with service version SCAI10, the OrigDeviceID parameter includes the name of the calling party and not the originating party. The OrigDeviceID of type DeviceAddressType is one or more of the following parameters:

- dialedDigits, IA5String, representing our dialable number delivery format
- stationNumber, Octet String, representing out 10-digit number
- name, IA5String, representing the caller's name

4.13.4.7 ECM option and line, or feature group option

CompuCALL calling name and number delivery service for residential or centrex lines is offered when ECM option is enabled and the line subscribes to the delivery option. Also the feature group option exist for the office without the line level subscription.

If either of the following conditions are met:

- The ECM line option does not require validation of CNAMD(CND for number) option for the line or feature group(FTRGRP).
- The ECM line option requires the CNAMD(CND for number) option for the line or feature group CNAMD(CND for number) option be verified and the validations pass.

This activity enhances the functionality to provide the calling name and number for IBN Type (500/2500) MDC/RES lines.

The MDC/RES lines with display sets validate before the calling name is delivered (calling number is provided if available without any restrictions for display sets) either of the following:

- The ECM line option does not require validation of NAMEDISP option for the customer group option.
- The ECM line option requires the validation of NAMEDISP option of the customer group.

4.13.4.8 Inter-switch calls using QUERY mode for name display

A gated procedure is used to determine if the terminating line has ECM option, is associated to a CompuCALL session, and subscribes to the new dv-Call-CallingName-U message. If the terminating set is either a display set or validates for CompuCALL, the calling name is requested or the TCAP query is launched to a centralized database. The callingname is delivered to the centrex line.

For the RES lines, if the terminating set is either a display set or validates for CompuCALL, the callingname is requested or the TCAP query is launched to a centralized database. The callingname is delivered to the RES line.

The dv-Call-Offered-U message contains the name if it is available for intra-switch calls and inter-switch calls with Name Display/Delivery mode using SETUP.

4.13.5 MDC/RES dv-Call-Answered-U (operation value "10" hex)

This section describes the dv-Call-Answered-U message the switch uses to notify the host of an incoming call offered to a CompuCALL party in an associated environment with which the host has an established session has been answered.   The CompuCALL party which answers the call is not necessarily the party to which the call has been offered (for example, in the MDC call pick-up case.) The "U" means this message is unconfirmed by the host.

Call Answered ASN.1 Encoding:

```
    dv-Call-Answered-U          OPERATION
                                ARGUMENT          CallAnsweredArgument
    ::= 16

    CallAnsweredArgument        ::= SEQUENCE
     {networkCallID            [0]               IMPLICIT  SEQUENCE
       {networkNodeID           [0]  IMPLICIT INTEGER             OPTIONAL,
         localCallID            [1]  IMPLICIT INTEGER             OPTIONAL}

                  destDeviceID[1]  AddressType                        OPTIONAL,
            acdDN[2]  IMPLICIT OCTET STRINGOPTIONAL,
                          origDeviceID       [3]  DeviceAddressType   OPTIONAL,
                          origChargeNumber[4]  IMPLICIT OCTET STRING      OPTION-
                      AL,
                          callHistoryInfo     [5]  IMPLICIT SEQUENCE
                              {callType             [0]  IMPLICIT  ENUMERATED
                                  {callTransferred    (1),
                                   callOverflowed     (2),
                                   callRedirected     (3),
                                   callForwarded      (4)

                           callConsult(5),

                           callConference(6)}}               OPTIONAL,
                              origInboundDN      [1]  IMPLICITDeviceAddressType OPTION-
                      AL,
                              prevApplicationID  [2]  IMPLICIT INTEGER OPTIONAL,
                              hostCallData        [3]  IMPLICIT OCTET STRING OPTION-
                      AL}
                                        OPTIONAL,
                          acdGroup            [6]  AddressType  OPTIONAL,
                          callMode            [7]  IMPLICIT ENUMERATED
                              {callWaited          (1)  OPTIONAL,
 teenService(2)  OPTIONAL}
     forwardingParty     [8]  ForwardingPartyType OPTIONAL}

DeviceAddressType     :: =    SET OF
     {  dialedDigits              [1]  IMPLICIT IA5String,       OPTION-
AL,
        stationNumber             [2]  IMPLICIT OCTET STRING,OPTION-
AL,
        name                      [4]  IMPLICIT IA5String OPTIONAL}

AddressType              :: =    CHOICE
     {  positionID         [0]  IMPLICIT INTEGER,
        dialedDigits       [1]  IMPLICIT IA5String,
        stationNumber   [2]  IMPLICIT OCTET STRING,
        q931Address      [3]  Q931AddressType}
ForwadingPartyType:: =    IMPLICIT SEQUENCE

{firstFwdNumber[0]  AddressType

  firstFwdReason [1]  IMPLICIT ENUMERATED
     {unknown(0),
     {userBusy (1),
      noReply(2),
     unconditional(3)}   OPTIONAL,
lastFwdNumber [2]  AddressType  OPTIONAL
lastFwdReason   [3]  IMPLICIT ENUMERATED
     {unknown(0),
       userBusy(1),
       noReply(2),
       unconditional(3)  OPTIONAL}
```

4.13.5.1 MDC/RES call answered: coded example 1

Following is a coded example of the dv-Call-Answered-U. The residential phone that answered has an destination device ID of (504) 858-4321. The originator’s dialed digits are provided but the caller’s name is blocked. This example has the calling dialable number parameter subscribed to through table control. The invoke ID is 4.

INVOKE DMS ====> HOST

Table 143 dv-Call-Answered-U

Hex Dump			Information Element	Comments
Tag	Len	Content		
A1	21		INVOKE	the message length is 33 bytes decimal
02	01	04	. . InvokeID	its Invoke ID is 4
02	01	10	. . Operation	its operation value is dv-Call-Answered-U [16] decimal
30	1C		. . Argument	CallAnsweredArgument is of type sequence and is 28 bytes long decimal
A1	0C		. . . . destDeviceID	the destDeviceID is of type constructor and is 12 bytes long
82	0A	35 30 34 38 35 38 34 33 32 31	. . . . stationNumber	the DN of the phone is 504-858-4321
A3	09		. . . . origDeviceID	origDeviceID is a constructor and is 9 bytes long
81	07	38 35 38 31 32 33 34	. . . . dialedDigits	the digits dialed are 858-1234
Hex dump = A1 21 02 01 04 02 01 10 30 1C A1 0C 82 0A 35 30 34 38 35 38 34 33 32 31 A3 09 81 07 38 35 38 31 32 33 34				

4.13.5.2 MDC/RES call answered: coded example 2

Following is a coded example of the dv-call-Answered-U message. The residential phone that answered has a destination device ID of (504) 858-5252. The originator’s name and dialed digits are provided. This example has the calling dialable number parameter, the origInboundDN, and the calling name parameter subscribed to through table control. The invoke ID is 4.

INVOKE DMS =====> HOST

Table 144 dv-Call-Answered-U example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	40		INVOKE	the message length is 64 bytes decimal
02	01	04	InvokeID	its Invoke ID is 4
02	01	10	Operation	its operation value is dv-Call-Answered-U [16] decimal
30	37		Argument	CallAnsweredArgument is of type sequence and is 55 bytes long decimal
A1	0C		destDeviceID	the destDeviceID is of type constructor and is 12 bytes long
82	0A	35 30 34 38 35 38 35 32 35 32	stationNumber	the DN of the phone is 504-858-5252
A3	18		origDeviceID	origDeviceID is a constructor and is 24 bytes long
81	07	38 35 38 31 32 33 34	dialedDigits	the digits dialed are 858-1234
84	0C	53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53	name	the customer's name is SHANANA JONES of length 13
A5	0E		callHistoryInfo	callHistoryInfo is a constructor of length 14
80	01	04	CallType	the call is forwarded
A1	09		origInboundDN	origInboundDN is a constructor of length 9
81	07	38 35 38 34 33 32 31	dialedDigits	the dialed digits are 858-4321
Hex dump = A1 40 02 01 04 02 01 10 30 37 A1 0C 82 0A 35 30 34 38 35 38 35 32 35 32 A3 18 81 07 38 35 38 31 32 33 34 84 0C 53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53 A5 0E 80 01 04 A1 09 81 07 38 35 38 34 33 32 31				

The switch sends a dv-Call-Answered-U message even if no caller information is available. The message is useful for the host to clear the workstation 's screen only when the new call is answered. In the MDC case, the message is useful when a new call offered to an MDC station is put on call waiting. The dv-Call-Answered-U message for the waited call is sent only when the call is answered, and the host can display the corresponding caller's file at that time.

The contents of the dv-Call-Answered-U message are identical to those of the dv-Call-Offered-U message.

> ***Note:*** The dv-Call-Answered-U message may be preceded by a dv-Call-Offered-U message regarding the same call, i.e., with the same NetworkCallID. In the case of MDC station call pick-up, these two messages may be sent for different stations. The dv-Call-Offered-U message is sent to host for the station to which the call is offered while the dv-Call-Answered-U message is sent corresponding to the station which picks up the call.

4.13.5.3 MDC/RES call answered: coded example 3

Following is a coded example of the dv-call-Answered-U message. The VMS station that answered has a Centrex DN of 613-621-1504l. The originator has a DN of 613-621-1501 and the call forwards from DN 613-621-1502. The invoke ID is 9.

INVOKE DMS =====> HOST

Table 145 dv-Call-Answered-U example 3

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	69		INVOKE	the message length is 105 bytes decimal
02	01	09	InvokeID	its Invoke ID is 9
02	01	10	Operation	its operation value is dv-Call-Answered-U [16] decimal
30	61		Argument	CallAnsweredArgument is of type sequence and is 97 bytes long decimal
A0	06		networkCallID	the networkCallID is of type constructor and is 6 bytes long
80	01	00	networkNodeID	the networkNodeID is 0
81	01	22	localCallID	the localCallID is 34
			—continued—	

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0C		destDeviceID	the destDeviceID is a constructor of length 12
82	0A	36 31 33 36 36 32 31 31 35 30 34	stationNumber	the voice mail port’s number is 613-621-1504
A3	16		origDeviceID	the origDeviceID is of type constructor and is bytes 22 long
81	08	39 36 32 3131 35 30 31	dialedDigits	the dial number is 9-621-1501
82	0A	36 31 33 36 32 31 31 35 30 31	stationNumber	the calling party DN is 613-621-1501
A5	1E		callHistoryInfo	callHistoryInfo is a constructor of length 30
80	01	04	callType	the call is forwarded
A1	16		origInboundDN	origInBoundDN is a constructor of length 22
81	08	39 36 32 31 31 35 30 32	dialedDigits	the DN for the called party is 9-621-1502
82	0A	36 31 33 36 32 31 31 35 30 32	stationNumber	the called party’s station number is 613-621-1502
82	01	01	prevApplicationeID	the previous application ID is 1
A9	11		forwardingParty	the forwardingParty is a constructor of length 17 bytes
A0	0F		firstFwdNumber	the firstFwdNumber is a constructor of length 15 bytes
82	0A	36 31 33 36 32 31 31 35 30 32	stationNumber	the first forwarded number is 613-621-1502 (the call in this example directly forwards to voice mail)
81	01	01	firstFwdReason	the forwarded reason is user busy
			—continued—	

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
Hex dump = A1 69 02 01 09 02 01 10 30 61 A0 06 80 01 00 81 01 22 A1 0C 82 0A 36 31 33 36 32 31 31 35 30 34 A3 16 81 08 39 36 32 31 31 35 30 31 82 0A 36 31 33 36 32 31 31 35 30 31 A5 1E 80 01 04 A1 16 81 08 39 36 32 31 31 35 30 32 82 0A 36 31 33 36 32 31 31 35 30 32 82 01 01 A8 11 A0 0F 82 OA 36 31 33 36 32 31 31 35 30 32 81 01 01				
			—end—	

4.13.5.4 dv-Call-Answered parameters

dv-Call-Answered contains the following optional parameters:

NetworkCallID CompuCALL-generated call ID composed of a network node ID (from table SCAIGRP) and a local call ID (unique call ID in the session). This is a new call id if this is the first instance of the call within this session/environment; otherwise the call ID already assigned to this call will be used. same call ID as the set off hook and call offered messages.

Contains static call-related information to uniquely identify the call in context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

destDeviceID the address of the terminator who answered this call in station number format for terminated calls -- same as in the call offered message. Contains dynamic call-related information identifying the specific customer workstation answering the call. It is unique in the switch and is a choice parameter. The choice is StationNumber.

StationNumber identifies the MDC (non-ACD) station which answered the call.

Originator's Device ID provided in the deviceAddressType format for terminating calls -- same as in the call offered message.

OrigChargeNumber the number being billed for this call. Contains static call-related information to indicate the operating company charge number (also called "ANI" - Automatic Number Identification in North America) associated with the incoming call when it enters the CompuCALL environment. A given dv-Call-Answered-U message contains both the OrigCallingNumber and OrigChargeNumber parameters, if available.

CallHistoryInfo if the call were transferred, overflowed, or redirected from an ACD group or agent, or transferred from or forwarded by an MDC customer workstation, the dv-Call-Answered-U message includes the CallHistoryInfo parameter.

Each subparameter is datafilled in the switch as provided (when appropriate) or not provided on an individual basis. In other words, the CallHistoryInfo parameter is present if the customer has subscribed to any of the following subparameters:

CallType contains dynamic call-related information indicating if the call to the current CompuCALL party were transferred, overflowed, redirected, forwarded, consult, or conference.

OrigInboundDN contains static call-related information identifying the first DN at which the call is directed when it enters the CompuCALL environment. It remains the same as long as the call stays in the CompuCALL environment even if it is later transferred, overflowed, redirected, or forwarded.

PrevApplicationID contains dynamic call-related information identifying the ApplicationID for the session of the associated environment to which the party extending the call belongs (call extension includes call transfer, overflow, redirect, and forward).

HostCallData contains dynamic call-related information of a generic and miscellaneous nature provided by a host. If the call were redirected by a host (Section 4.6.2), this parameter may be provided by that host and include the reason for the redirection.

Call Mode now includes Teen Service. Same format as in the Call Offered message. CallWaited is delivered if the call is waiting. TeenService is delivered if the terminating Teen Service feature is active. CallWaitOnTeenService which is delivered if a user terminates on a Teen Service SDN that has CWT.

ForwardingParty identifies the first forwarding number and associated reason and last forwarding number and reason for forwarding. In a single forwarding scenario, the information is included in the FirstFwd Number and First FwdReason subparameters. This information is provided to the host only if presentation of the forwarding DNs is allowed. The parameter is a constructor including the following information:

FirstFwdNumber identifies the number from which the first forward was made in a multiple call forwarding scenario. Based on the Q931 format, it uses the Q931Address choice.

FirstFwdReason identifies the reason the call was first forwarded. It is an enumerated type with values: unknown/not available, user busy, no reply, or unconditional.

LastFwdNumber identifies the number from which the lastforward was made in a multiple call forwarding scenario. Based on the Q931 format, it uses the Q931Address choice.

LastFwdReason identifies the reason the call was last forwarded (most recent). It is an enumerated type with values unknown/not available, user busy, no reply, or unconditional.

deviceAddressType lets us send a set of one or more values repre senting the calling device. The set includes the following parameters:

dialedDigits contains the DN in dial number delivery format

StationNumber contains directory number format name contains the caller’s name. When the call is local, the calling name is retrieved by accessing data via datafill. If the calling name is datafilled then the calling name will be provided in the call event parameter. When the call goes over an ISUP trunk or ISDN trunk, the calling name can be retrieved by accessing data. Note, that when the call goes over a BTUP trunk or ATUP trunk the calling name will not be retrieved. Also, name delivery using TCAP query to the service control point (SCP) database is not supported.

No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Answered-U message. This is the reason for the "U" (unconfirmed) in the message name. If either is received by the switch, a Reject is returned with reason "unrecognized invocation." Furthermore, the switch ignores any Reject message from the host in response to a dv-Call-Answered-U message.

4.13.6 MDC/RES dv-Call-Released-U (operation value 7)

This section describes the dv-Call-Released-U message the switch uses to notify the host why the call was released from a CompuCALL party belonging to an Associated environment with which the host has an established session. The "U" means this message is unconfirmed by the host.

```
ASN.1 encoding
    dv-Call-Released-U              OPERATION
                                   ARGUMENT                  ReleaseArgument

    ::= 7

    ReleaseArgument                ::=            SEQUENCE
                             {  networkCallID   [0]  IMPLICIT SEQUENCE
                                 {networkNodeID    [0] IMPLICIT INTEGER OPTIONAL,
                                   localCallID        [1] IMPLICIT INTEGER OPTIONAL}
                                                                          OPTIONAL,
                             releaseReason       [1] IMPLICIT ENUMERATED
                                 { callCleared                  (0),
                                   callTransferred              (1),
                                   callOverflowed               (2),
                                   callAbandoned                (3),
                                   callRedirected               (4),
                                   callForwarded                (5),
                                   callPickedUp                 (6),
                                   partyDropped                 (7),
                                   partyDroppedNoAnswer  (8)}                OPTIONAL,
                         acdDN            [2] IMPLICIT OCTET STRINGOPTIONAL,
                             acdGroup              [4] AddressType  OPTIONAL,
                             deviceID              [5] AddressType  OPTIONAL}

                         AddressType               :: =    CHOICE
                             {  positionID         [0]  IMPLICIT INTEGER,
                                dialedDigits       [1]  IMPLICIT IA5String,
                                stationNumber   [2]  IMPLICIT OCTET STRING,
                                q931Address      [3]  Q931AddressType,}
```

This message. which indicates why the call was released, provides closure regarding previous CompuCALL messages for that party on the same call. It also allows the host to close a call record associated with the call.

4.13.6.1 dv-Call-Released-U parameters

This message is sent when either party involved in either an outgoing or incoming call has released the call (gone on-hook, or hit a release key); or if the call has been abnormally released. A dv-Call-Released will contain the following optional parameters:

NetworkCallID same as Call ID as was delivered with the set off hook, call offered and call answered messages. This contains static call-related information to identify the call in context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages that relate to the same call.

ReleaseReason indicates why a call was released from a CompuCALL party belonging to an associated environment. The release reason can be one of the following:

a. Call Cleared (one party in a two-party CompuCALL call has disconnected after being answered)

b. Caller Transferred (the CompuCALL party has transferred the call and disconnected)

c. Call Abandoned (caller in a two-party call has disconnected before a call was answered by the CompuCALL party - while in the Incoming Call Queue)

d. Call Forwarded (call was forwarded to another station by the MDC Call Forward Don't Answer feature, see Note)

e. Call Picked-up (call was picked up by another station after call was offered to the MDC CompuCALL party)

f. PartyDropped (CompuCALL party in an established 3-way call is dropped or released).

g. PartyDroppedNoAnswer (CompuCALL party is a consult party in a 3-way call and is dropped or released before answering).

To be consistent, if call transfer or 3-way call conference, the party who initiates the transfer or conference is the controller. The original call party is the party who calls the controller or the party the controller first calls to establish the original 2-way call. The consult party is the party who the controller calls after a 2-way call between the controller and original call party already exists.

DeviceID contains dynamic call-related information to identify the specific customer workstation from which the call is released. It is unique within the switch and a choice parameter. The choice is StationNumber.

StationNumber identifies the MDC line from which the call is released.

No Return Result (positive acknowledgment) message or Return Error (negative acknowledgment) message is expected by the switch from the host in response to a dv-Call-Released-U message. This is the reason for the "U" (unconfirmed) in the message name. If either is received by the switch, a

Reject is returned with reason "unrecognized invocation." Furthermore, the switch ignores any Reject message from the host in response to a dv-Call-Released-U message.

4.13.7 dv-Set-Offhook-U (operation value 5)

The switch sends the Class 5 operation dv-Set-Offhook message to notify the host that an associated line has gone off-hook and been allocated for a call. The "U" means this message is unconfirmed by the host.

This message is sent when an associated line has gone off-hook and is allocated for dialing -- at this point the line could be activating a feature or originating a call.

ASN.1 Encoding

```
    dv-Set-OffHook-U            OPERATION
                                ARGUMENT            SetOffHookArgument
    ::= 25

    SetOffHookArgument          ::=     SEQUENCE

    { networkCallID             [0]  IMPLICIT  SEQUENCE
        {networkNodeID            [0]  IMPLICIT INTEGER            OPTIONAL,
          localCallID             [1]  IMPLICIT INTEGER            OPTIONAL}
                origAddress         [1]  AddressTypeOPTIONAL}

            AddressType::=          CHOICE
             {positionID            [0]    IMPLICIT  INTEGER,
               dialedDigits         [1]    IMPLICIT  IA5 String,
               stationNumber        [2]    IMPLICIT  OCTET STRING,
               q931Address          [3]    Q931AddressType}
```

4.13.7.1 dv-Set-OffHook parameters

dv-Set-OffHook contains the following optional parameters:

Network Call ID CompuCALL-generated call ID composed of a network node ID (from table SCAIGRP) and a local call ID (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session/environment.

Device ID The directory number (DN) in station number format of the address going off hook.

The set off-hook message will not be sent for either MakeCall or Add Party calls since the information would be redundant (the network call ID is currently sent as a return result from either of these messages, which indicates the call was successfully originated).

4.13.7.2dv-Set-OffHook-U: coded example 1

Following is a coded example of the dv-Set-OffHook-U message. The residential phone has a origination address of 858-1234 and has gone off-hook. The invoke ID is 2

INVOKE DMS =====> HOST

Table 146 dv-Set-OffHook-U example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1E		INVOKE	the message length is 53 bytes decimal
02	01	02	InvokeID	its Invoke ID is 3
02	01	19	Operation	its operation value is dv-Call-Offered-U [5] decimal
30	16		Argument	CallOfferedArgument is of type sequence and is 48 bytes long decimal
A0	06		networkCallID	the networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 3
A3	0C		origDeviceID	the origDeviceID is of type constructor and is bytes 21 long
82	0A	35 30 34 38 35 38 31 32 33 34	stationNumber	the DN of the phone is 504-858-1234
Hex dump = A1 1E 02 01 02 02 01 19 30 16 A0 06 80 01 01 81 01 02 A1 0C 82 0A 35 30 34 38 35 38 31 32 33 34				

4.14 CompuCALL MDC/RES call control

4.14.1CompuCALL MDC/RES dv-Answer-Call

This capability supports functionality to answer an incoming call on behalf of an ACD agent, Centrex line, or a residential line.

ASN.1 Encoding

```
dv-Answer-Call              OPERATION
                            ARGUMENT  AnswerCallArgument
                            RESULT    AnswerCallResult
                            ERRORS    {invalidCallState,
                                      missingParameter
                                      invalidAnsweringParty,
                                      notAllowed,
                                      unknownAnsweringParty,
                                      resourcesUnavailable}
::= 23

AnswerCallArgument::=       SEQUENCE
{answeringParty     [0]     AddressType  OPTIONAL}

AddressType::=      CHOICE
{positionID         [0]IMPLICIT          INTEGER,
dialedDigits        [1]IMPLICIT          IA5STRING,
stationNumber       [2]IMPLICIT          OCTET STRING,
q931Address         [3]Q931AddressType}

Q931AddressType::=          SEQUENCE
{numberTypeNumberPlan        [0]IMPLICITOCTET STRING OPTIONAL,
presentationScreeningIndicator[1]IMPLICITOCTET STRING OPTIONAL,
digits                       [2]IMPLICITIA5     STRING OPTIONAL}

AnswerCallResult::=         SEQUENCE
    {networkCallID          [0]  IMPLICIT SEQUENCE
        {networkNodeID         [0]   IMPLICIT INTEGER OPTIONAL,
        localCallID            [1]   IMPLICIT INTEGER OPTIONAL}
                                                      OPTIONAL}

invalidCallState    ERROR::=0
missingParameter    ERROR::=1
                    PARAMETER            MissingParameter

    MissingParameter::=  SEQUENCE
    {missingParameterType        [0]IMPLICIT ENUMERATED
        {missingAnsweringParty(1)}                  OPTIONAL}
```

```
invalidAnsweringParty      ERROR::=2
notAllowed                 ERROR::=3
unknownAnsweringParty      ERROR::=4
resourcesUnavailable       ERROR::=5
```

4.14.1.1 dv-Answer-Call: coded example

This example illustrates the case where a host sends a message to answer a call. The position ID of the agent's set is 2000 and the invoke ID is 11.

INVOKE DMS <===== HOST

Table 147 Answer call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0E		INVOKE	Message length is 14 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	17	Operation	Operation value for dv-Answer-Call is decimal 23.
30	06		AnswerCallArgument	Type sequence and 6 bytes long.
A0	04		answeringParty	AnsweringParty is a 4 byte constructor.
80	02	07 D0	positionID	Agent’s Position ID = 2000.
Hex Dump = A1 0E 02 01 0B 02 01 17 30 06 A0 04 80 02 07 D0				

The switch verifies the request and sends back a Return Result indicating the request will be initiated.

RETURN-RESULT DMS =====> HOST

Table 148 Answer call request return result

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0B	InvokeID	Invoke ID is 11.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	17	Operation	its operation is dv-Answer-Call [23]
30	09		AnswerCall Result	the Answer Call Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0B 30 0E 02 01 17 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to answer a call using dv-Answer-Call and there is not a call being offered, the switch will respond with a Return Error message that the call is in an invalid state.

RETURN-ERROR DMS =====> HOST

Table 149 Answer call request return error

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0B	InvokeID	Invoke ID is 11
02	01	00	ErrorType	ErrorType is Invalid-Call-State [0]
Hex Dump = A3 06 02 01 0B 02 01 00				

4.14.1.2 Answer call functionality

The answer function is provided by a Class 2 RO, dv-Answer-Call. This capability supports functionality to answer an incoming call on behalf of an ACD agent, Centrex line, or a residential line.

If the party which is specified in the dv-Answer-Call message is being offered a call (ringing), then the dv-Answer-Call request will be processed by the switch to set up answering the call on behalf of the user.

A successful invocation of a dv-Answer-Call generates the CVD message dv-Call-Answered-U to be sent to the host. The dv-Call-Answered-U message must be subscribed to and the ACD group or Centrex/RES line must be dv-DN-Associated with the SCAI session in order for the host to receive an event message.

The dv-Answer-Call message is supported on Meridian Business Sets (MBS) with a headset device or enabling the microphone and speaker on a hands-free device. On MBS the set is updated as appropriate by solidifying the lamp on when the dv-Answer-Call message is received. The dv-Answer-Call message will not be supported on 2500 sets.

4.14.1.3 dv-Answer-Call validations

Upon receiving the dv-Answer-Call message the switch will perform the following validations:

1 Verify that the service version of the current host-switch session is SCAI08 or higher.

2 Verify that the AnswerCall message has been subscribed to by the host session via table SCAISSRV.

3 Verify that the required parameter, answeringParty is present in the message.

4 Verify that the answeringParty specified belongs to same customer group as the current host-switch session.

5 Verify that the Centrex or residential line specified in the answeringParty parameter of the dv-Answer-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.

6 Verify that there is an ACD, Centrex, or residential call being offered to the set (ringing).

7 Verify that the ACD agent, Centrex, or residential line attempting to answer the call is a MBS set with handsfree capabilities.

4.14.1.4 Answer call message parameters

A brief description of the message and parameter are given below:

dv-Answer-Call the host sends a dv-Answer-Call in order to answer a call on behalf of a specific directory number (DN). A dv-Answer-Call will contain the following mandatory parameter:

answeringParty the AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that the host is making the answer request.

4.14.1.5 Answer call responses

The switch responds to the dv-Answer-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08 a REJ is sent back to the host.
- Return-Error (RE)
  - Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 150 Answer call error conditions

Error	Description
notAllowed	The AnswerCall message is not subscribed to at the link-level. Subscription to messages is done via table SCAIPROF and table SCAISSRV. The Centrex or residential line specified in the answeringParty has not subscribed to the ECM sub-option SCAICC. The answeringParty parameter contains an ACD agent or Centrex line offered to a set that is MFT.
missingParameter	The required answeringParty parameter is not included in the message
—continued—	

Table 150 Answer call error conditions

Error	Description
invalidAnsweringParty	The DN specified in the answeringParty parameter is not in the same customer group as the host-switch session. The answeringParty parameter contains an ACD agent, Centrex, or residential call being offered to a set that is not a MBS.
unknownAnsweringParty	The DN specified in the answeringParty parameter is unknown to the switch.
invalidCallState	The answeringParty parameter contains an ACD, a Centrex, or a residential party that is not in the ringing state.
—end—	

4.14.2 CompuCALL MDC/RES dv-Hold-Call

This message supports functionality to hold an active call on behalf of an ACD agent, Centrex line, or a residential line.

```
ASN.1 Encoding
dv-Hold-Call                      OPERATION
    ARGUMENT                      HoldCallArgument
    RESULT                        HoldCallResult
    ERRORS                        {invalidCallState,
                                  missingParameter,
                                  invalidHoldingParty,
                                  notAllowed,
                                  unknownHoldingParty,
                                  callNotHoldableByUser,
                                  resourcesUnavailable}
::= 28

HoldCallArgument    ::=           SEQUENCE
{holdingParty[0]AddressTypeOPTIONAL}

AddressType::= CHOICE
{positionID[0]IMPLICITINTEGER, (ONLY CHOICE)
 dialedDigits[1]IMPLICITIA5STRING,
 stationNumber[2]IMPLICITOCTET STRING,
 q931Address[3]Q931AddressType}

Q931AddressType::= SEQUENCE
{numberTypeNumberPlan [0]IMPLICITOCTET STRING OPTIONAL,
 presentationScreeningIndicator[1]IMPLICITOCTET STRING OPTIONAL,
 digits[2]IMPLICITIA5STRING OPTIONAL}
```

```
HoldCallResult::=SEQUENCE
{networkCallID   [0]   IMPLICIT SEQUENCE
     {networkNodeID   [0]   IMPLICIT INTEGER OPTIONAL,
       localCallID   [1]   IMPLICIT INTEGER OPTIONAL}
                    OPTIONAL}

invalidCallState ERROR::=0
missingParameter ERROR::=1
     PARAMETERMissingParameter

MissingParameter::=SEQUENCE
  {missingParameterType[0]IMPLICIT ENUMERATED
     {missingHoldingParty(1)}OPTIONAL}

invalidHoldingPartyERROR::=2
notAllowedERROR::=3
unknownHoldingPartyERROR::=4
callNotHoldableByUserERROR::=5
resourcesUnavailableERROR::=6
```

4.14.2.1 dv-Hold-Call: coded example

This example illustrates the case where a host sends a message to hold DN 782-3640.

INVOKE DMS <===== HOST

Figure 50 Hold Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	16		INVOKE	Message length is 22 bytes
02	01	0D	InvokeID	Invoke ID is 13
02	01	1C	Operation	Operation value for dv-Hold-Call is decimal 28.
30	0E		HoldCallArgument	Type sequence and 14 bytes long.
A0	0C		holdingParty	holdingParty is a 12 byte constructor.
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	stationNumber is 919-782-3640
Hex Dump = A1 16 02 01 0D 02 01 1C 30 0E A0 0C 82 0A 39 31 39 37 38 32 33 36 34 30				

The switch verifies the request and sends back a Return Result indicating the request will be initiated.

RETURN-RESULT DMS =====> HOST

Table 151 RETURN-RESULT for hold call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0D	InvokeID	Invoke ID is 13.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	1C	Operation	its operation is dv-Hold-Call [28]
30	09		HoldCall Result	the HoldCall Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 03 02 01 0D 30 0E 02 01 1C 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to hold a call using dv-Hold-Call but the holding party (residential line) has not subscribed to the ECM sub-option SCAICC. A Return Error of not allowed is sent to the host computer.

RETURN-ERROR DMS =====> HOST

Figure 51 RETURN-ERROR for hold request with an error value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0D	InvokeID	Invoke ID is 13.
02	01	02	ErrorType	ErrorType is notAllowed [2]
Hex Dump = A3 06 02 01 0D 02 01 02				

The hold function is provided by two Class 2 ROs: dv-Hold-Call, dv-Unhold-Call. A Class 5 RO is also provided: dv-Call-Unheld-U. The new messages supports functionality to hold and unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The dv-Hold-Call message is usable on active, connected calls, while the dv-Unhold-Call message may only be used to unhold a call held by the dv-Hold-Call message. If a user should hit their DN key on an MBS set, this will effectively unhold the call if it is being held by the CompuCALL application, upon which time switch will send a dv-Call-Unheld-U message to the host computer. If a 500/2500 (IBN and POTs) set is being held by the CompuCALL application, when the user flashes on a 500/2500 (IBN and POTs) set, this will unhold the call and a dv-Call-Unheld-U message is sent to the host computer.

After a dv-Hold-Call is sent to a 500/2500 (IBN and POTs) line and silence is heard, SCAI 3WC messages (Add, Drop, and Transfer) messages cannot be used to originate a call.

The party specified in the dv-Hold-Call message can be a residential, Centrex or ACD position ID. If a dv-Hold-Call message is sent to the ACD position ID, and the user hits their DN key on an MBS set, the party specified in the dv-Call-Unheld-U message is the ACD position ID.

Suppose a party specified in the dv-Hold-Call message is placed on hold in an active call. Next, another dv-Hold-Call message is sent to hold the other party on an active call, so both parties are on hold. A dv-Unhold-Call message is used to unhold one of the parties. A second dv-Unhold-Call message is used to unhold the other party and establishes a talking connection.

If at any time during the hold of the line, the user activates a set based hold, the CompuCALL activated hold will remain activated until deactivated as outlined above, and the set based hold will hold the line until deactivated on the set. Thus the functionality of the dv-Hold-Call will not be interchangeable with the set based hold.

Since the user may subsequently choose to continue the call from the telephone set, it is desirable to inform the user of the call hold - for example, on a Meridian Business Set a flashing diamond will be displayed next to the line in use.

4.14.2.2 dv-Hold-Call validations

Upon receiving the dv-Hold-Call message the switch will perform the following validations:

1. Verify that the service version of the current host-switch session is SCAI08 or higher.
2. Verify that the HoldCall message has been subscribed to by the host session via table SCAISSRV.
3. Verify that the required parameter, holdingParty is present in the message.

4 Verify that the holdingParty specified belongs to same customer group as the current host-switch session.

5 Verify that the Centrex or residential line specified in the holdingParty parameter of the dv-Hold-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.

6 Verify that the ACD, Centrex, or residential line specified in the holdingParty is in an active, connected call.

4.14.2.3 Hold parameters

A brief description of the messages and parameters are given below:

dv-Hold-Call the host sends a dv-Hold-Call in order to hold a call on behalf of a specific DN. A dv-Hold-Call will contain the following mandatory parameter:

holdingParty the AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that the host is making the hold request.

4.14.2.4 Hold call responses

The switch responds to the dv-Hold-Call and dv-Unhold-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08, a REJ is sent back to the host.
  - Since the dv-Call-Unheld-U is a Class 5 operation, the host will not respond to the message.
- Return-Error (RE)

Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 152 Hold call error conditions

Error	Description
notAllowed	The HoldCall message is not subscribed to at the link-level. The Centrex or residential line specified in the holdingParty has not subscribed to the ECM sub-option SCAICC The holdingParty parameter contains an ACD agent Centrex line offered to a set that is MFT.
missingParameter	The required holdingParty parameter is not included in the message.
invalidholdingParty	The DN specified in the holdingParty parameter is not in the same customer group as the host-switch session. The holdingParty parameter contains an ACD agent, Centrex or residential line offered to a set that is not MBS or 500/2500.
unknownHoldingParty	The DN specified in the holdingParty parameter is unknown to the switch.
invalidCallState	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in an active connected call. {dv-Hold-Call}. The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in a call held by the dv-Hold-Call. {dv-Unhold-Call}
callNotHoldableByUser	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is involved with a Emergency Service line.

4.14.3 dv-Release-Call message

This functionality supports functionality to release an active call on behalf of an ACD agent, Centrex line, or a residential line.

ASN.1 Encoding

```
dv-Release-Call             OPERATION
                            ARGUMENT    ReleaseCallArgument
                            RESULT      ReleaseCallResult
                            ERRORS      {invalidCallState,
                                        missingParameter
                                        invalidParameter
                                        notAllowed,
                                        unknownReleasingParty,
                                        callNotReleasableByUser,

        resourcesUnavailable}

::= 24

ReleaseCallArgument::=      SEQUENCE
{releasingParty             [0] AddressType  OPTIONAL,
 releaseType                [1]  IMPLICIT ENUMERATED
     {releaseThisParty          (1) OPTIONAL,
      releaseAllParties         (2) OPTIONAL}

AddressType::=              CHOICE
{positionID                 [0] IMPLICIT        INTEGER, (ONLY
CHOICE)
 dialedDigits               [1] IMPLICIT        IA5STRING,
 stationNumber              [2] IMPLICIT        OCTET STRING,
 q931Address                [3] Q931AddressType}

Q931AddressType::=          SEQUENCE
{numberTypeNumberPlan       [0] IMPLICIT        OCTET STRING
OPTIONAL,
 presentationScreeningIndicator[1]IMPLICIT      OCTET STRING
OPTIONAL,
 digits                     [2] IMPLICIT        IA5STRING
OPTIONAL}


ReleaseCallResult          ::=    SEQUENCE
    {networkCallID          [0]   IMPLICIT SEQUENCE
    {networkNodeID  [0]   IMPLICIT INTEGER OPTIONAL,
     localCallID  [1]   IMPLICIT INTEGER OPTIONAL}
                                     OPTIONAL,

invalidCallState             ERROR::=0
missingParameter             ERROR::=1
                             PARAMETER   MissingReleaseParm
     MissingReleaseParm         ::=    SEQUENCE
       {missingParameterType  [0]    IMPLICIT ENUMERATED
          {missingReleasingParty      (0)}                           OPTIONAL}

invalidParameter             ERROR::= 2
                             PARAMETER   InvalidReleaseParm
     InvalidReleaseParm         ::=    SEQUENCE
       {invalidParameterType   [0]    IMPLICIT ENUMERATED
          {invalidReleaseType         (0)                            OPTIONAL}
            invalidReleasingParty     (1)                            OPTIONAL}

notAllowed                   ERROR::=3
unknownReleasingParty   ERROR::=4
callNotReleasableBy UserERROR::=5

resourcesUnavailableERROR::=6
```

4.14.3.1 dv-Release-Call: coded example

This example illustrates the case where an ACD agent is requesting that an incoming call be released. The position ID of the set is 2000 and the invoke ID is 11.

INVOKE DMS <===== HOST

Table 153 Release call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	0E		INVOKE	Message length is 14 bytes.
02	01	0C	InvokeID	Invoke ID is 12.
02	01	18	Operation	Operation value for dv-Release-Call is decimal 24.
30	06		ReleaseCallArgument	Type sequence and 6 bytes long.
A0	04		releasingParty	ReleasingParty is a 4 byte constructor.
80	02	07 D0	PositionID	Agent’s Position ID = 2000.
81	01	01	releaseType	releaseThisParty
Hex Dump = A1 0E 02 01 0C 02 01 18 30 06 A0 04 80 02 07 D0 81 01 01				

4.14.3.2 dv-Release-Call: coded example

This example illustrates the case where an ACD agent is requesting that an incoming call be released. The position ID of the set is 2000 and the invoke ID is 11.

INVOKE DMS <===== HOST

Table 154 RETURN-RESULT for Release Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0C	InvokeID	Invoke ID is 12.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	18	Operation	its operation is dv-Release-Call [24]
30	09		ReleaseCall Result	the Release Call Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0C 30 0E 02 01 18 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to release a call using dv-Release-Call and selects an unknown DN, then the switch must respond with a Return Error with the error value unknown releasing party.

RETURN-ERROR DMS =====> HOST

Table 155 RETURN-ERROR for Release Request with unknownReleasingParty

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0C	InvokeID	Invoke ID is 12.
02	01	04	ErrorType	ErrorType is UnknownReleasingParty [4]
Hex Dump = A3 06 02 01 0C 02 01 04				

The release function is provided by a Class 2 RO, dv-Release-Call. This functionality supports functionality to release an active call on behalf of an ACD agent, Centrex line, or a residential line.

If the party which is specified in the dv-Release-Call message is on an active call (ringing, talking, dialing, hold, listening to busy, or an announcement), then the dv-Release-Call request will be processed by the switch to set up releasing the call on behalf of the user.

When the party specified in the dv-Release-Call message is the terminator of the call and the terminator is receiving physical ringing then the call is not released. On the other hand, when the party specified in the dv-Release-Call is the originator of the call and the originator is receiving audible ringing the call is released.

When the party specified in the dv-Release-Call message is on hold by CompuCALL, the party on hold is released. Whereas in the same scenario on the switch, the party that is on hold can not be released by the controller until after the call is answered.

A successful invocation of a dv-Release-Call generates the CVD message dv-Call-Released-U to be sent to the host just as though the call had been released manually. The dv-Call-Released-U message must be subscribed to and the ACD group or Centrex/RES line must be dv-DN-associated with the SCAI session in order for the host to receive the CVD message. The release reason provided with the dv-Call-Released-u will be the same as if the call had been released manually. (i.e., call_cleared for two party calls, call_abandoned for unanswered calls, etc.)

The dv-Release-Call message is supported on MBS and 500/2500 sets. On MBS the set is updated as appropriate by turning the lamp off when the dv-Release-Call message is received. After successfully issuing a dv-Release-Call request on a 500/2500 set, the user receives dial tone. The user is then able to initiate calls before going on hook and returning to the idle state. If the user stays off hook for a certain length of time, then the set will go to lock out treatment.

4.14.3.3 dv-Release-Call validations

Upon receiving the dv-Release-Call message the switch performs the following validations:

1 Verify that the service version of the current host-switch session is SCAI08 or higher.

2 Verify that the ReleaseCall message has been subscribed to by the host session via table SCAISSRV.

3 Verify that the required parameter, releasingParty is present in the message.

4 Verify that the optional parameter, releaseType has been subscribed to by the host session.

5 Verify that the releasingParty specified belongs to same customer group as the current host-switch session.

6 Verify that the Centrex or residential line specified in the releasingParty parameter of the dv-Release-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI Call Control) sub-option of the ECM option must have been selected.

7 Verify that the ACD, Centrex, or residential line specified in the releasingParty is in a state that is considered releasable.

4.14.3.4 Release call message parameters

A brief description of the message and parameters are given below:

dv-Release-Call the host sends a dv-Release-Call in order to release a call on behalf of a specific directory number (DN). A dv-Release-Call will contain the following parameters:

releasingParty the AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that a host is making the release request on behalf of.

releaseTypes is an optional parameter contains an enumerated type which consists of: releaseThisParty, and releaseAllParties. This parameter specifies whether to release this party alone or release all parties in a call.

In a normal two party call, either type will be accepted and the call will simply be released on behalf of the releasingParty. In a three-way call conference scenario, the releaseThisParty type will release on behalf of the controller which may or may not cause the call to be transferred depending on whether the user has call transfer capabilities, while the releaseAllParties type will release all three parties from the three-way call without any type of transfer taking place.

4.14.3.5 Release call responses

The switch responds to the dv-Release-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08, a REJ is sent back to the host.
- Return-Error (RE)
  - Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 156 Release Call error conditions

Error	Description
notAllowed	The ReleaseCall message is not subscribed to at the link-level. The Centrex or residential line specified in the releasingParty has not subscribed to the ECM sub-option SCAICC The releasingParty parameter contains an ACD agent or Centrex line offered to a set that is MFT.
missingParameter	The required releasingParty parameter is not included in the message.

—continued—

Table 156 Release Call error conditions

Error	Description
invalidParameter	A parameter that is not subscribed to is included in the message.
invalidReleasingParty	The DN specified in the releasingParty parameter is not in the same customer group as the host-switch session. The releasingParty parameter contains an ACD agent, Centrex, or residential line offered to a set that is not MBS nor 2500.
unknownReleasingParty	The DN specified in the releasingParty parameter is unknown to the switch.
invalidCallState	The releasingParty parameter contains an ACD, a Centrex, or a residential party that is not in a releasable state (i.e., the releasingParty is in an idle state).
callNotReleasableByUser	The releasingParty parameter contains a ACD, a Centrex, or a residential party that is involved with an emergency service line.
—end—	

4.14.4 CompuCALL MDC/RES dv-Unhold-Call

This message supports functionality to unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

ASN.1 Encoding

```
dv-Unhold-Call             OPERATION
                           ARGUMENT    UnHoldCallArgument
                           RESULT      UnHoldCallResult
                           ERRORS      {invalidCallState,
                                       missingParameter,
                                       invalidHoldingParty,
                                       notAllowed,
                                       unknownHoldingParty,

                                      resourcesUnavailable}
::= 29

UnHoldCallArgument::=      SEQUENCE
{unholdingParty            [0] AddressType       OPTIONAL}

AddressType::=             CHOICE
{positionID                [0] IMPLICIT          INTEGER, (ONLY
CHOICE)
 dialedDigits              [1] IMPLICIT          IA5STRING,
 stationNumber             [2] IMPLICIT          OCTET STRING,
 q931Address               [3] Q931AddressType}

Q931AddressType::=         SEQUENCE
{numberTypeNumberPlan      [0] IMPLICIT          OCTET STRING
OPTIONAL,
 presentationScreeningIndicator[1]IMPLICIT       OCTET STRING
OPTIONAL,
 digits                    [2] IMPLICIT          IA5STRING
OPTIONAL}

UnHoldCallResult           ::=    SEQUENCE
   {networkCallID           [0]   IMPLICIT SEQUENCE
   {networkNodeID   [0]   IMPLICIT INTEGER OPTIONAL,
   localCallID   [1]   IMPLICIT INTEGER OPTIONAL}
                                OPTIONAL}

invalidCallState          ERROR::=0
missingParameter          ERROR::=1
                          PARAMETER   MissingParameter

   MissingParameter          ::=   SEQUENCE
    {missingParameterType  [0]   IMPLICIT ENUMERATED
      {missingHoldingParty       (1)}                  OPTIONAL}
```

invalidHoldingParty	ERROR::=2
notAllowed	ERROR::=3
unknownHoldingParty	ERROR::=4

resourcesUnavailable ERROR::=5

4.14.4.1 dv-Unhold-Call: coded example

This example illustrates the case where a host sends a message to unhold a call that has been held by the dv-Hold-Call message.

INVOKE DMS <===== HOST

Table 157 UnHold Call Request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	16		INVOKE	Message length is 22 bytes.
02	01	0B	InvokeID	Invoke ID is 11.
02	01	1D	Operation	Operation value for dv-Unhold-Call is decimal 29.
30	0E		UnholdCallArgument	Type sequence and 14 bytes long.
A0	0C		unholdingParty	unholdingParty is a 12 byte constructor.
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	The station number is 919-782-3640
Hex Dump = A1 16 02 01 0B 02 01 1D 30 0E A0 0C 82 0A 39 31 39 37 38 32 33 36 34 30				

The switch verifies the request and sends back a Return Result indicating the request will be initiated.

RETURN-RESULT DMS =====> HOST

Table 158 RETURN-RESULT for UnHold call request

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	The message length is 19 bytes
02	01	0B	InvokeID	Invoke ID is 11.
30	0E		Return Result Sequence	this sequence is 14 bytes long decimal
02	01	1D	Operation	its operation is dv-UnHold-Call [29]
30	09		unholdCall Result	the unholdCall Result is of type sequence and is 9 bytes long
A0	07		Network Call ID	NetworkCallId is of type constructor and is 7 bytes long
80	01	01	NetworkNodeID	Network Node ID is 1
81	02	05 55	Local Call ID	Local Call ID is 1365.
Hex Dump = A2 13 02 01 0B 30 0E 02 01 1D 30 09 A0 07 80 01 01 81 02 05 55				

If the host attempts to unhold a call using dv-Unhold-Call but forgets to include the holdingParty parameter in the message then the switch must respond with a Return Error describing why the request could not be completed.

RETURN-ERROR DMS =====> HOST

Table 159 RETURN-ERROR for UnHold Request with an error value

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	Message length is 6 bytes
02	01	0B	InvokeID	Invoke ID is 11
02	01	01	ErrorType	ErrorType is missingParameter[1]
Hex Dump = A3 06 02 01 0B 02 01 01				

The hold function is provided by two Class 2 ROs: dv-Hold-Call, dv-Unhold-Call. A Class 5 RO is also provided: dv-Call-Unheld-U. The new messages supports functionality to hold and unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The dv-Hold-Call message will be usable on active, connected calls, while the dv-Unhold-Call message may only be used to unhold a call held by the dv-Hold-Call message. If a user should hit their DN key on an MBS set, this will effectively unhold the call if it is being held by the CompuCALL application, upon which time switch will send a dv-Call-Unheld-U message to the host computer. If a 500/2500 (IBN and POTs) set is being held by the CompuCALL application, when the user flashes on a 500/2500 (IBN and POTs) set, this will unhold the call and a dv-Call-Unheld-U message is sent to the host computer.

Suppose a party specified in the dv-Hold-Call message is placed on hold in an active call. Next, another dv-Hold-Call message is sent to hold the other party on an active call, so both parties are on hold. A dv-Unhold-Call message is used to unhold one of the parties. A second dv-Unhold-Call message is used to unhold the other party and establishes a talking connection.

If at any time during the hold of the line, the user activates a set based hold, the CompuCALL activated hold will remain activated until deactivated as outlined above, and the set based hold will hold the line until deactivated on the set. Thus the functionality of the dv-Hold-Call will not be interchangeable with the set based Hold.

Since the user may subsequently choose to continue the call from the telephone set, it is desirable to inform the user of the call hold. For example, on a Meridian Business Set a flashing diamond will be displayed next to the line in use.

Upon receiving the dv-Unhold-Call message the switch will perform the following validations:

1. Verify that the service version of the current host-switch session is SCAI08 or higher.
2. Verify that the UnholdCall message has been subscribed to by the host session via table SCAISSRV.
3. Verify that the required parameter, holdingParty is present in the message.
4. Verify that the holdingParty specified belongs to same customer group as the current host-switch session.
5. Verify that the Centrex or residential line specified in the holdingParty parameter of the dv-Unhold-Call has subscribed to the ECM line sub-options. The SCAICC (SCAI call control) sub-option of the ECM option must have been selected.
6. Verify that the ACD, Centrex, or residential line specified in the holdingParty is involved in a call held by the dv-Hold-Call message.

4.14.4.2 Unhold parameters

A brief description of the messages and parameters are given below:

dv-Unhold-Call the host sends a dv-Unhold-Call in order to unhold a call on behalf of a specific directory number (DN). A dv-Unhold-Call will contain the following mandatory parameter:

unholdingParty the AddressType is the only mandatory parameter which can be specified as a positionID or stationNumber and identifies the user that the host is making the unhold request on behalf of.

4.14.4.3 Unhold call responses

The switch responds to the dv-Hold-Call and dv-Unhold-Call message in one of three ways. The possible return messages are a Return Result (RR), Return Reject (REJ), or a Return Error (RE).

- Return-Result (RR)
  - If the incoming request is successfully validated by the switch, a RR is sent back to the host to indicate that the request is valid and will be initiated.
- Return-Reject (REJ)
  - When the incoming request cannot be decoded or interpreted by the switch, a REJ message is sent back to the host and no call-related actions will take place. For example, if a mandatory parameter (the entire sequence is not included) is not included in message, then a REJ is sent back to the host. Also if the service version is prior to SCAI08, a REJ is sent back to the host.
  - Since the dv-Call-Unheld-U is a Class 5 operation, the host will not respond to the message.
- Return-Error (RE)
  - Any error conditions detected by the switch result in a RE message being returned to the host. The RE message contains an appropriate error cause value to inform the host why the request could not be

initiated. The following is a list of error conditions and the error cause value sent with the RE:

Table 160 Hold call error conditions

Error	Description
notAllowed	The HoldCall message is not subscribed to at the link-level. The Centrex or residential line specified in the holdingParty has not subscribed to the ECM sub-option SCAICC The holdingParty parameter contains an ACD agent Centrex line offered to a set that is MFT.
missingParameter	The required holdingParty parameter is not included in the message.
invalidholdingParty	The DN specified in the holdingParty parameter is not in the same customer group as the host-switch session. The holdingParty parameter contains an ACD agent, Centrex, or residential line offered to a set that is not MBS or 500/2500.
unknownHoldingParty	The DN specified in the holdingParty parameter is unknown to the switch.
invalidCallState	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in an active connected call {dv-Hold-Call}. The holdingParty parameter contains an ACD, a Centrex, or a residential party that is not involved in a call held by the dv-Hold-Call {dv-Unhold-Call}.
callNotHoldableByUser	The holdingParty parameter contains an ACD, a Centrex, or a residential party that is involved with a emergency service line.

4.14.5 CompuCALL MDC/RES dv-Call-Unheld-U

If a user should hit their DN key on a MBS set or 2500 set flashes and unhold the call (held by the CompuCALL application), the switch will send an unheld message to the host computer.

ASN.1 Encoding

```
dv-Call-Unheld-U               OPERATION
                               ARGUMENT     CallUnheldArgument

::= 30

CallUnheldArgument::=          SEQUENCE
    {networkCallID          [0]   IMPLICIT SEQUENCE
    {networkNodeID  [0]   IMPLICIT INTEGER OPTIONAL,
    localCallID  [1]   IMPLICIT INTEGER OPTIONAL}
                                 OPTIONAL,
    unheldParty [1] Address Type OPTIONAL}


AddressType::=                 CHOICE
{positionID                    [0] IMPLICIT          INTEGER, (ONLY
CHOICE)
 dialedDigits                  [1] IMPLICIT          IA5STRING,
 stationNumber                 [2] IMPLICIT          OCTET STRING,
 q931Address                   [3] Q931AddressType}

Q931AddressType::=             SEQUENCE
{numberTypeNumberPlan          [0] IMPLICIT          OCTET STRING
OPTIONAL,
 presentationScreeningIndicator[1]IMPLICIT           OCTET STRING
OPTIONAL,
 digits                        [2] IMPLICIT          IA5STRING
OPTIONAL}
```

4.14.5.1 dv-Call-Unheld-U: coded example

This example illustrates when the user hits the DN key to unhold a call which was held by the dv-Hold-Call message.

DMS =====> HOST

Table 161 Call Unheld

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1E		INVOKE	Message length is 30 bytes.
02	01	0F	InvokeID	Invoke ID is 15.
02	01	1E	Operation	Operation value for dv-Call-Unheld-U is decimal 30.
30	16		CallUnheldArgument	Type sequence and 22 bytes long.
A0	06		Network Call ID	NetworkCallId is of type constructor and is 6 bytes long
80	01	01	NetworkNodeID	Network node ID is 1
81	01	02	Local Call ID	Local call ID is 2.
A1	0C		unheldParty	unheldParty is a 12 byte constructor.
82	0A	39 31 39 37 38 32 33 36 34 30	stationNumber	The DN of the phone is 919-782-3640
Hex Dump = A1 1E 02 01 0F 02 01 1E 30 16 A0 06 80 01 01 81 01 02 A1 0C 82 0A 39 31 39 37 38 32 33 36 34 30				

The hold function is provided by two Class 2 ROs: dv-Hold-Call, dv-Unhold-Call. A Class 5 RO is also provided: dv-Call-Unheld-U. The new messages supports functionality to hold and unhold an active call on behalf of an ACD agent, Centrex line, or a residential line.

The dv-Hold-Call message will be usable on active, connected calls, while the dv-Unhold-Call message may only be used to unhold a call held by the dv-Hold-Call message. If a user should hit their DN key on an MBS set, this will effectively unhold the call if it is being held by the CompuCALL application, upon which time switch will send a dv-Call-Unheld-U message to the host computer. If a 500/2500 (IBN and POTs) set is being held by the CompuCALL application, when the user flashes on a 500/2500 (IBN and POTs) set, this will unhold the call and a dv-Call-Unheld-U message is sent to the host computer.

The party specified in the dv-Hold-Call message can be a residential, Centrex, or ACD position ID. If a dv-Hold-Call message is sent to the ACD position ID, and the user hits their DN key on an MBS set, the party specified in the dv-Call-Unheld-U message is the ACD position ID.

It is not necessary to associate a line to get the call unheld message.

The switch will perform a series of checks before sending the call event message, dv-Call-Unheld-U. The switch will verify:

1 Verify that the CompuCALL hold message, dv-Hold-Call, placed the party on hold.

2 The service version is at least SCAI08

3 The switch-host session subscribes to the call event message.

4 The mandatory parameter (for the ECM application) of the message is present.

5 The DN specified is within the same customer group as the SCAI session that the host is connected to and is a valid ACD agent, residential DN or Centrex DN.

4.14.5.2 Call-Unheld parameters

A brief description of the messages and parameters are given below:

dv-Call-Unheld-U message is sent after the successful invocation of dv-Hold-Call and the user releases the call by hitting their DN key or flashing the 500/2500 set. A dv-Call-Unheld-U will contain the following parameters:

Network Call ID is an optional parameter and a CompuCALL generated call ID composed of a network node ID (taken from table SCAIGRP) and a local call ID (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session/environment.

unheldParty the mandatory parameter is the AddressType which can be specified as a positionID or stationNumber which had been unheld.

Since the dv-Call-Unheld-U is a Class 5 operation, the host will not respond to the message.

4.15 CompuCALL MDC/RES call initiation

4.15.1 CompuCALL MDC/RES dv-Make-Call (operation value 9)

The host sends the dv-Make-Call message to the switch when an user has initiated an outgoing Centrex or residential call from their associated data terminal. The Centrex/residential lines do not need to be in the associated environment with the host as an established session. When the host sends a Make_call without associating the original agent, subsequent messages for this call have different callIDs. It is the host's responsibility to ensure that MakeCall requests are only allowed on sets requested by the owner of those sets.

The parameters which are struck out will not be used for the residential/Centrex CompuCALL messages, but for ACD only, and the parameters which are bold and italicized are existing parameters which are validated differently for these messages.

```
MakeCall ASN.1 Encoding
                dv-Make-Call                            OPERATION
                ARGUMENT                                MakeCallArgument
                RESULT                                  MakeCallResult
                ERRORS              {not-Allowed, not-Idle,
                                       agent-Not-Logged In,
                                     orig-Time-Out, makeCallAborted,
                                     makeCall-Mismatched-State,
                                       makeCall-Resources-Unavailable,
                                     missingOrigAddress,
                                       missingDestAddress,
                                     missingMakeCallType,
                                     invalidOrigAddress,
                                       invalidDestAddress,
                                     invalidMakeCallType, invalidAuthCode,
                                       invalidAcctCode,
                                     authOptNotSubscribed,
                                     acctOptNotSubscribed
                                       unexpectedAcctCode}
   ::= 9
   MakeCallArgument ::=SEQUENCE
   {   origAddress          [0]  AddressType               OPTIONAL,
       destAddress          [1]  AddressType               OPTIONAL,
       applicationData      [2]  IMPLICIT SEQUENCE
         {makeCallType[0]  IMPLICIT ENUMERATED
              {callingAgentOnline         (0),
           callingAgentReserved      (1),
           callingAgentNotReserved  (2)}               OPTIONAL,
          authCodeDigits[1]  IMPLICIT IA5String          OPTIONAL,
            acctCodeDigits[2]  IMPLICIT IA5String         OPTIONAL}
                                                  OPTIONAL}

AddressType   :: =      CHOICE
      { positionID[0]  IMPLICIT INTEGER,
      dialedDigits[1]  IMPLICIT IA5String,
      stationNumber[2]  IMPLICIT OCTET STRING,
      q931Address[3]  Q931AddressType,}

MakeCallResult::=      SEQUENCE
    { networkCallID[0]  IMPLICIT SEQUENCE
    { networkNodeID[0] IMPLICIT INTEGER OPTIONAL,
       localCallID[1] IMPLICIT INTEGER OPTIONAL}
                    OPTIONAL}

    not-Allowed                    ERROR::= 1
    not-Idle                       ERROR::= 2
    agent-Not_Logged_In            ERROR::= 3
    orig-Time-Out                  ERROR::= 4
    makeCall-Resources-UnavailableERROR::= 5
    missingOrigAddress             ERROR::= 6
    missingDestAddress             ERROR::= 7
    missingMakeCalltype            ERROR::= 8
    invalidOrigAddress             ERROR::= 9
    invalidDestAddress             ERROR::= 10
    invalidMakeCalltype            ERROR::= 11
    invalidAuthCode                ERROR::= 12
    invalidAcctCode                ERROR::= 13
    authOptNotSubscribed           ERROR::= 14
    acctOptNotSubscribed           ERROR::= 15
    makeCallAborted                ERROR::= 16
    makeCallMismatchedStateERROR::= 17
    unexpectedAcctCode             ERROR::= 18
```

4.15.1.1MakeCall: coded example

INVOKE DMS ===== > HOST

Table 162 dv-Make-Call invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	2A		INVOKE	this message length is 42 bytes decimal
02	01	28	InvokeID	the InvokeID is 40 decimal
02	01	09	Operation	the operation is dv-Make-Call [9] decimal
30	22		Argument	MakeCall argument is of sequence type and is 34 bytes long decimal
A0	0C		OrigAddress	the OrigAddress is a constructor 12 bytes long
82	0A	35 30 36 35 38 31 32 31 32	station Number	the station number is 5068581212
A1	0D		destAddress	the destination address is of type construction and is 13 bytes long decimal
81	0B	31 36 31 33 35 35 35 31 32 31 32	DialedDigits	the dial DN is 1-613-555-1212
A2	03		ApplicationData	ApplicationData is a constructor 3 bytes long
80	01	00	makeCallType	makeCallType is a primitive enumerated value, in this case = calling agent on-line.
Hex Dump = A1 2A 02 01 28 02 01 09 30 22 A0 0C 82 0A 35 30 36 38 35 38 31 32 31 32 A1 0D 81 0B 31 36 31 33 35 35 35 31 32 31 32 A2 03 80 01 00				

RETURN-RESULT DMS =====> HOST

Table 163 dv-Make-Call return result example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	12		RETURN RESULT	this message is 18 bytes long decimal
02	01	28	InvokeID	the InvokeID is 40 decimal
30	0D		ReturnResult Sequence	this is a sequence 13 bytes long decimal
02	01	09	Operation	the operation is dv-Make-Call [9]
30	08		MakeCallResult	the MakeCallResult is of type sequence and is 8 bytes long
A0	06		NetworkCallID	the NetworkCallID is of type sequence and is 6 bytes long
80	01	01	NetworkNodeID	the NetworkNodeID component is 1
81	01	38	LocalCallID	the LocalCallID component is 56
Hex Dump = A2 12 02 01 28 30 0D 02 01 09 30 08 A0 06 80 01 01 81 01 38				

RETURN-ERROR DMS =====> HOST

Table 164 dv-Make-Call return error example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	28	InvokeID	the InvokeID is 40 decimal
02	01	04	ErrorValue	the ErrorValue is 4 representing Orig-Time out occurred (user did not respond to alerting by answering their set).
Hex Dump = A3 06 02 01 28 02 01 04				

The host requests the DMS-100 switch to place an outgoing call on behalf of the agent (meaning user) whose stationNumber = (506) 858-1212 and who will be on-line to the destination address (DialedDigits) = 1-613-555-1212. After verifying the request, the switch sends Return Result to the host when the agent acknowledges the dv-Make-Call request (user answers the alerting on the set). As shown in "1.4.3.1 make call sample session" on page 39, another request fails because the set to originate from did not respond to the alerting which causes the switch to send a Return-Error.

The dv-Make-Call message sent by the host will specify the directory number (DN) in station number format to originate the call from, the destination address in a dial number format to terminate the call to, a MakeCall type of calling agent on-line (meaning the user must be on-line--available to interact with the destination party--before the call can proceed), and optionally an authorization and/or account code digits

4.15.1.2 dv-Make-Call validations

On receiving a dv-Make-Call message the switch performs the following validations for Centrex or residential DN origination address requests:

1 Verify that the MakeCall message has been subscribed to by the host session via the MakeCall function in table SCAISSRV.

2 Validate the message received by ensuring that the parameters OrigAddress, DestAddress, and makeCallType are present in the message.

3 If authorization code digits have been sent, validate that the sub-parameter has been subscribed to by the host session via the MakeCall function in table SCAISSRV and that one or more digits has been included as the authorization code.

4 If account code digits have been sent, validate that the sub-parameter has been subscribed to by the host session via the MakeCall function in table SCAISSRV.

5 Verify that the StationNumber (origAddress) specified is valid for the specific customer and service version which is associated with the active ECM session.

6 Verify that MakeCall has been subscribed to via the ECM line sub-options for the StationNumber (origAddress) specified. The MakeCall sub-option of the ECM option must have been selected.

7 Verify that the DialedDigits (destAddress) provided will translate successfully according to the datafill for the originator in the DMS.

8 Verify that the makeCallType specified is Calling Agent On-line.

9 If the authorization code has been sent, check that the maximum number of authorization code digits has not been exceeded.

10 If the account code has been sent, check to see if the auth/acct codes should have been sent as combined (this is in datafill for auth/acct codes). If they should have been combined, sending an account code separately is treated as an error.

11 Check that the line of the origination address specified is idle (no verification of the line state of the destination address will occur).

If the request to initiate the call is valid, then the user's set will receive an alerting signal (physical ringing or distinctive ringing if subscribed to). After the user accepts the call by answering the ringing, the switch sends a Return-Result (positive acknowledgment) message to the host. This message contains the parameter NetworkCallID. The switch will then proceed with the set-up of the call and ring the destination.

When the call has ended at any time, if the originator or terminator's DN has been dv-DN-Associated to a host session, a dv-Call-Released-U message will be sent with the same Network Call ID specified in the return result of the MakeCall message. If the originator and terminator are on different switches and the terminator is associated to a CompuCALL session, the Network Call ID will not be the same for the call event messages relating to the MakeCall originated call on another switch. Currently, CompuCALL is not implemented to support network exchange of the CompuCALL information like Network Call ID.

CompuCALL takes no action if it receives a Reject message in response to a Return-Result or Return-Error message. That is, the call, if successful in initiating, will continue without interruption.

4.15.1.3 MakeCall message parameters

The host sends the dv-Make-Call message to the switch to initiate an outbound call on behalf of a specific DN or ACD position ID. The DNs need not be in the associated environment with which the host application has an established session (that is the DN need not have been dv-DN-Associated to the session attempting to initiate the MakeCall). A brief description of the parameters of the dv-Make-Call invoke follows:

Table 165 MakeCall parameter descriptions

Parameter Name	M/O[a]	Description
origAddress	M	This parameter identifies the originator of the call to be set up. It is defined as a choice parameter, which may be either an ACD agent position ID or the station number of the device to originate the call from for Centrex and RES lines (in which case it is an implicit octet string).
		—continued—

Table 165 MakeCall parameter descriptions

Parameter Name	M/O[a]	Description
destAddress	M	This parameter identifies the terminator of the call to be set up. It is also defined as a choice parameter, currently it is the dialed digits of the device to terminate to. This is a string of IA5-encoded digits that correspond to the exact number the originator would have dialed, had the dialing been performed manually.
applicationData	M	This parameter contains the application specific subparameters the host wishes to pass the DMS. Currently for MakeCall, one or more of the following three sub-parameters may be included with the message:
makeCallType	M	For ACD, this parameter can be specified as CallingAgentOnline, CallingAgentReserved, CallingAgentNotReserved, CallingAgentBuzzBase, or CallingAgentBeepHset. For MDC lines this parameter can be CallingAgentOnline, CallingAgentBuzzBase, or CallingAgentBeepHset. For residential lines this parameter can be CallingAgentOnline.
authCodeDigits	O	This subparameter may be provided at the beginning of a call as the user identification. It contains the authorization code digits which will be verified before allowing the agent to access the appropriate NCOS other than the current default NCOS. Furthermore, the authorization code digits may be further categorized into authorization code types which are verified by the switch to provide additional security.
acctCodeDigits	O	This subparameter is for internal cost allocation purposes only. It allows the user or the host to enter a cost-accounting or client number into the station message detail recording (SMDR) record for charge back purposes. If the combined authorization and account code option is chosen the AcctCodeDigits must be included in the AuthCodeDigits.
		—end—

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

At the SCAI communication protocol level all of these parameters are optional. But at the ECM application level, the origAddress, destAddress, and makeCallType are mandatory parameters.

The makeCall message which the customer subscribes to will always require sending the origAddress, destAddress, and MakeCallType. The authCodeDigits and acctCodeDigitis subparameters are optionally selected in the service profile of the MakeCall message.

4.15.1.4 MakeCall error conditions

If the request to initiate the call is unsuccessful, or the user's alerting times out before accepting the outgoing call, the switch sends a Return-Error (e.g., negative acknowledgment) message to the host indicating the failure is due to one of the following reasons (the information in curly braces indicates the error message parameter which would be sent if the validation fails):

1 Customer is not allowed to initiate a make call. {not allowed}

2 End-user (the person on whose behalf the MakeCall is being originated: this may be an ACD agent, Centrex user, or a residential line) not available to receive outbound calls, e.g., the set's DN specified is busy on another call--not idle. {not idle}

3 MakeCall invoke already in progress for the user. {not idle}

4 For Centrex, the makeCallType specified is not callingAgentOnline, CallingAgentBuzzBase, nor callingAgentBeepHset..{invalid MakeCall type}

5 For residential lines, the makeCallType specified is not callingAgentOnline. {invalid MakeCall type}

6 missing parameter (either origAddress, destAddress, or makeCallType). {missing-parameter including which is missing}

7 The service version is a previous version to SCAI09. {invalid parameter: origination address}

8 invalid parameter (either origAddress, destAddress, makeCallType, authorization or account code). {invalid-parameter including which is invalid}

9 An authorization or account code has been sent when the optional parameter has not been subscribed to. {autOptNotSubscribed or acctOptNotSubscribed}

10 The account code was sent when it should have been included with the authorization code. {unexpectedAcctCode}

11 Insufficient resources to initiate the call. {MakeCall resources unavailable}

12 MakeCall is aborted by user releasing the phone while alerting. {MakeCall Aborted}

13 The digits provided in the destination address field contains something other than dial digits or do not translate to a valid destination {invalid destination address}. The digits dialed must be the same digits the user would have dialed if they wanted to reach the same destination. The digits will be translated the same way as they would be had the user dialed the digits. Valid translations are anything that will translate to a route. Things that do not qualify as a route are:

a. translations that require more digits

b. translations that require a database query (e.g., TCAP)

c. translations that correspond to a feature access code

14 Alerting time out (failed to respond within period specified by RNG_TMEOUT_NO_OF_SECS in table OFCENG). {orig time-out}

This list is presented in the order of priority since only the first error condition (including the missing/invalid parameter) encountered by the switch is included in the return error message.

4.15.1.5 MakeCall return result

If the requested MakeCall was a valid request and the user picked up their phone in response to the alerting, the switch will respond by sending the host a return result with the following parameter: MakeCall return error

Table 166 MakeCall return result parameter

Parameter Name	M/O[a]	Description
networkCallID	M	This parameter uniquely identifies the call within the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch or host generated messages that relate to the same call (e.g., the dv-Call-Released message).

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

If the requested MakeCall was an invalid request or the user did not respond to the alerting, the switch will respond by sending the host a return error with the following parameter: CompuCALL MDC/RES SCAI 3WC

Table 167 MakeCall return error parameter

Parameter Name	M/O	Description
makeCallError	M	This parameter uniquely identifies the type of error that occurred to cause the MakeCall request to fail, e.g., missing or invalid parameter, or originator alerting timed out.

4.15.1.6 CompuCALL MDC/RES dv-Add-Party (operation value "C" hex)

The SCAI conferencing/transfer messages are now available for residential and Centrex lines. The following messages: add party, transfer party, conference party and drop party allow accepting an origination address of type station number -- representing either a Centrex or residential line.

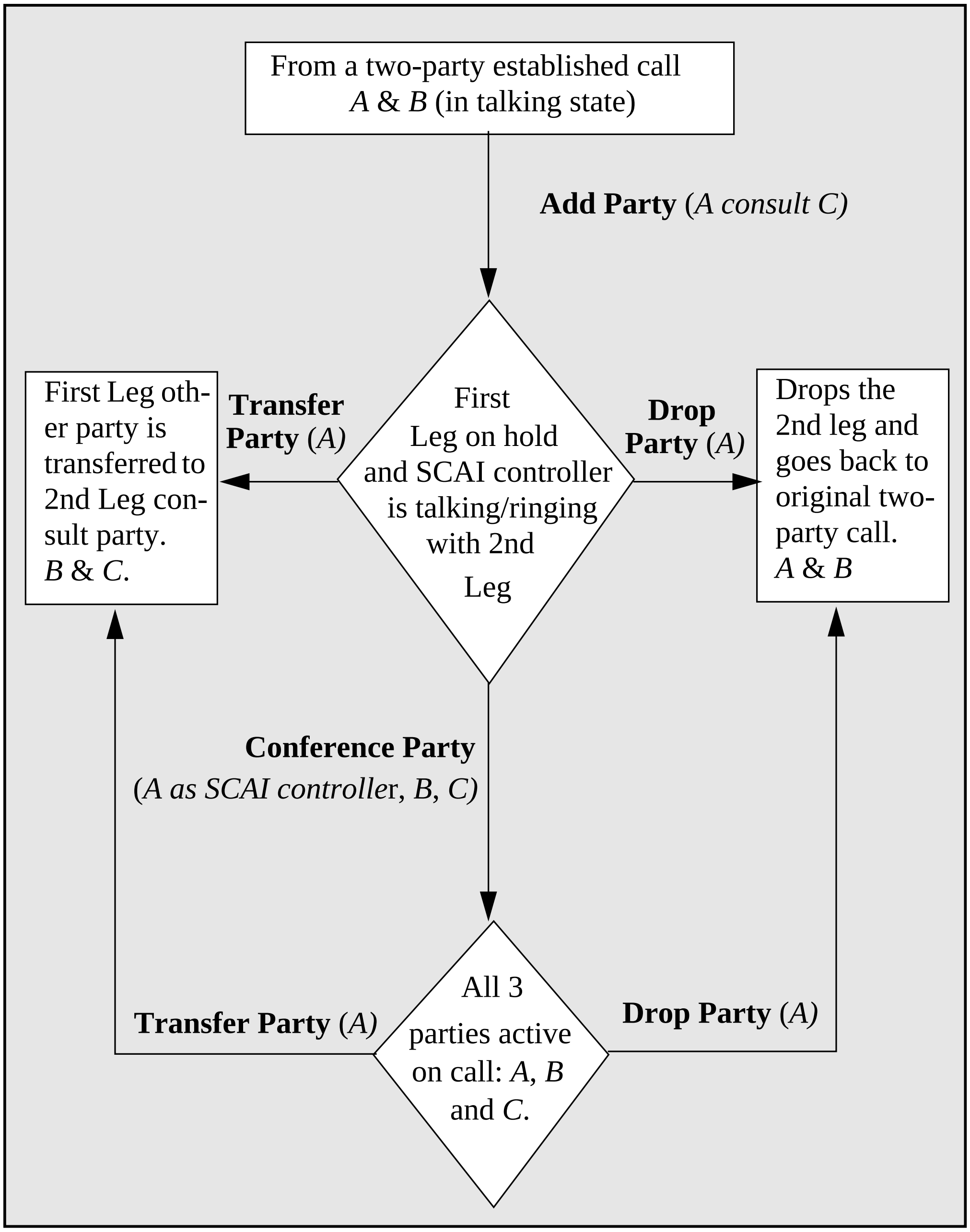
Add/drop/transfer/conference party messages are only supported for single line appearance DNs. Thus an origination address specifying a MADN DN, UCD DN, ACD DN, or Teen Service SDN will cause a return error indicating an invalid origination address.

The new SCAI application logon service version of SCAI07 is used for these messages. Thus, if a host logs on with a service version existing before our new service version, the residential or Centrex originated add/drop/transfer/conference request will be denied as it always has been for previous versions.

4.15.2 SCAI consult/conferencing/transfer message flow

The following diagram depicts the SCAI capabilities of add, consult, drop and transfer party and how they interact.

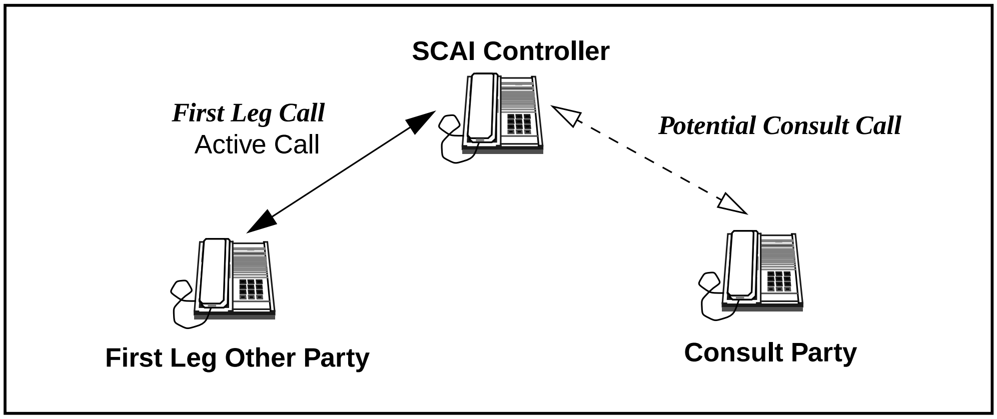
Figure 52 SCAI 3WC message flows


From a two-party established call
A & *B* (in talking state)
Add Party (*A consult C)*
First
Leg on hold
and SCAI controller
is talking/ringing
with 2nd
Leg
Transfer
Party (*A)*
First Leg other party is transferred to 2nd Leg consult party.
B & *C*.
Drop
Party (*A)*
Drops the 2nd leg and goes back to original two-party call.
A & *B*
Conference Party
(*A as SCAI controll*er, *B*, *C)*
All 3
parties active
on call: *A*, *B*
and *C*.
Transfer Party (*A)*
Drop Party (*A)*

An add party message will allow the host to set up a Consult Call from the party specified as the origination address; this party is referred to the SCAI controller of the call. The active call that is being extended may involve one or more parties (e.g., the SCAI Controller is talking on another two-party call or the SCAI Controller was connected to a conference call by another controller[1]), and is referred to as the first leg call. The other party involved in the first leg call may be any party and is referred to as the first leg other party.

The destination address specified for the consult call may be any valid DN but not a feature access code; this destination is called the consult party.

Figure 53 SCAI consult parties



The dv-Add-Party message sent by the host will specify the DN in station number format to originate the consult-leg call from, the destination address is in a dial number format to terminate the call to, an optional AddPartyType of either add consult party or add consult for conference may be specified.

The optional Add Party Type causes the switch to do one of two things:

Add consult party puts the first leg other party on hold and initiates a call to a consult party. This implies that this call may or may not be conferenced; therefore, no conference resources are reserved. If at some future point in the consult call conferencing is required, a SCAI conference can be initiated; however, there is no guarantee that conference circuits will be available to satisfy the request.

Add consult for conference will first attempt to reserve conferencing resources for the call, and if there are no conference circuits that can be reserved a “Resources Unavailable” error message will result. Otherwise the first leg other party will be put on hold and a call will be initiated to the consult party.

Once the dv-Add-Party message has been received by the switch, the switch will perform some validation on the message to ensure that it is a valid request.

The switch will perform a series of checks to ensure that the request is valid. The switch will verify:

[1] An add party may only be initiated once per SCAI controller. This limitation essentially means that only a three-way call may be set up via CompuCALL. In order to extend the number of parties on the call, one of the other parties must initiate the add-on which will set up a 3WC chaining scenario. CompuCALL will enforce the 3WC chaining limits set up for the switch.

1 The switch-host session subscribes to the dv-Add-Party message.

2 The mandatory parameters (for the ECM application) of the message are present in the message.

3 Verify the service version.

4 The origination address specified is within the same customer group as the SCAI session that the host is connected to and is a valid residential DN, Centrex DN, or an ACD agent position ID.

5 When a residential or Centrex DN is specified as the origination address, verify the line has been subscribed to this service.

6 The destination address contains dialed digits.

7 The End-user is in the appropriate state to initiate consultation calls, e.g., the set's DN specified is not idle and is talking on a stable two-party call and is not already controller of a consult or three-way call (SCAI or set-based).

8 When an addPartyType is addConsultforConf, then verify that conference resources are available.

9 The digits provided in the destination address field translate to a valid destination.

If the request to initiate the call is valid, then the first leg call will be put on hold and the consult call will be initiated to the consult party. After the destination address digits have been successfully translated and the initial setup passes, the switch sends a Return-Result (positive acknowledgment) message to the host. This message contains the network callid of the call. The switch will then proceed with the set-up of the consult call and ring the destination.

At this point, there will be no more SCAI signaling done in connection with this request. Should anything cause the request to fail (e.g., no software resources), inbound signaling will be given to the user.

When the call has ended at any time, if the originator or terminator's DN has been dv-DN-Associated to a host session, a dv-Call-Released-U message will be sent with the same network callid specified in the return result of the Add Party message. Also, the network callid of the first leg call will be the same network callid sent as part of the add party Return Result.

If the originator and terminator are on different switches and the terminator is associated to a CompuCALL session, the network callid will not be the same for the call event messages relating to the add party originated call on another switch. Currently, CompuCALL is not implemented to support network exchange of the CompuCALL information like network callid.

CompuCALL takes no action if it receives a Reject message in response to a Return-Result or Return-Error message.

4.15.2.1 Add party message parameters

The host sends the dv-Add-Party message to the switch to initiate an outbound consult-leg call on behalf of a specific DN. The DNs need not be in the associated environment with which the host application has an established session (that is the DN need not have been dv-DN-Associated to the session attempting to initiate the Add-Party consult-leg call). A brief description of the parameters of the dv-Add-Party invoke follows:

Table 168 Add Party Parameter Descriptions

Parameter Name	M/O[a]	Description
addPartyType	M	Enumerated type representing the ability to add the party strictly as a consult party for transferring calls, or as a consult party for either transferring calls or conferencing both calls together.
origAddress	M	This parameter identifies the originator of the call to be set up. It is defined as a choice parameter, which may be either an ACD agent position ID or the station number of the device to originate the call from for Centrex and RES lines (in which case it is an implicit octet string). This is the SCAI controller.
destAddress	M	This parameter identifies the terminator of the call to be set up. It is also defined as a choice parameter, currently it is the dialed digits of the device to terminate to. This is a string of IA5-encoded digits that correspond to the exact number that the originator would have dialed, had the dialing been performed manually.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

All of these parameters are optional at the SCAI communications protocol level. At the ECM application level the origAddress, destAddress, and addPartyType are mandatory parameters.

The set off-hook message will not be sent for either MakeCall or add party calls since the information would be redundant (the network call ID is currently sent as a return result from either of these messages, which indicates the call was successfully originated).

4.15.2.2 Add party error conditions

If the request to initiate the consult call was unsuccessful, the switch sends a Return-Error (e.g., negative acknowledgment) message to the host indicating the failure is due to one of the following:

1 Customer is not allowed to initiate an add party consult call because the link is not subscribed to the service. {not allowed}

2 A mandatory parameter is missing, either the destination or origination address or the add party type. {missing parameter and which parameter is missing}

3 The invoke contains a parameter which is not subscribed to. {invalid parameter and which parameter is invalid}

4 The service version is a previous version to SCAI07. {invalid parameter: origination address}

5 The origination address specified is not within the same customer group as the SCAI session that the host is connected to or is not a valid residential DN, Centrex DN, or an ACD agent position ID. {invalid parameter: origination address}

6 When a residential or Centrex DN, if the line has not been subscribed to this service via the ECM line sub-options, then the add party will not be allowed. {not allowed}

7 The destination address contains something other than dialable digits. {invalid parameter: destination address}

8 User not in the appropriate state to initiate consultation calls, e.g., the set's DN specified is idle or not yet talking on a stable two-party call or is already controller of a consult or three-way call (SCAI or set-based). {invalid Call State}

9 Add party invoke already in progress for agent. {invalid call state}

10 Insufficient resources to initiate the call. {resources unavailable}

11 Another feature is active which denies allowing the add party to progress (e.g., attendant console) {not allowed}

12 The digits provided in the destination address field contains something other than dialable digits or do not translate to a valid destination {invalid parameter: destination address}. The digits dialed must be the same digits that the user would have dialed if they wanted to reach the same destination and will be translated the same way as they would be had the user dialed the digits. Valid translations are anything that will translate to a route. Things that do not qualify as a route are:

a. translations that require more digits

b. translations that require a database query (e.g., TCAP)

c. translations that correspond to a feature access code

13 The resources were not available (e.g., if add party type is "add consult for conference", and no conference circuits are available). {resources unavailable}

Note: This list is presented in the order of priority since only the first error condition (including the missing/invalid parameter) encountered by the switch is included in the return error message.

4.15.2.3 Add party return result

If the requested add party was a valid request and the resources existed to comply with the request, the switch will respond by sending the host a return result with the following parameter:

Parameter Name	M/O[a]	Description
networkCallID	M	This parameter uniquely identifies the call within the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch or host generated messages that relate to the same call (e.g., the dv-Call-Released message).

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The network callid will be the same for all stages of the consult, conference and transfer that may occur via SCAI messaging. If the origination or destination address is associated via dv-DN-Associate, then the network callid created for the first leg call will be the same network callid used for these SCAI messages return-result.

4.15.2.4 Add party return error

If the requested add party was an invalid request or the resources did not exist to comply with the request, the switch will respond by sending the host a return error with the following parameter:

Parameter Name	M/O[a]	Description
error	M	This parameter uniquely identifies the type of error that occurred to cause the request to fail, e.g., missing or invalid parameter, or invalid call state.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The types of errors that may be encountered are detailed in the previous sections describing each of these messages.

```
                Add Party ASN.1 Encoding
                    dv-Add-PartyOPERATION
                    ARGUMENTAddPartyArgument
                    RESULTAddPartyResult
                    ERRORS{invalid-Call-State,
                        missing-Parameter,
                        invalid-Parameter
                        not-Allowed
                        agent-not-Logged-In,
                        resources-Unavailable}
                    ::= 12

                    AddPartyArgument::=  SEQUENCE
                    {   addPartyType[0] IMPLICIT ENUMERATED
                    {   addConsultParty(0) OPTIONAL,
                        addConsultforConf(1) OPTIONAL}
                                OPTIONAL,
                    origAddress[2] AddressTypeOPTIONAL,
                    destAddress[3] AddressTypeOPTIONAL}

                    AddressType:: =  CHOICE
                    {   positionID[0] IMPLICIT INTEGER,
                        dialedDigits[1] IMPLICIT IA5STRING,
                        stationNumber[2] IMPLICIT OCTET STRING,
                        q931Address[3] Q931AddressType}
Q931AddressType                :: =     SEQUENCE
 {  numberTypeNumberPlan                [0] OCTET STRING         OPTIONAL,
    presentationScreeningIndicator      [1] OCTET STRING         OPTIONAL,
    digits                              [2] IA5 STRING           OPTIONAL}


AddPartyResult                 ::=  SEQUENCE
 {networkCallID[0] IMPLICIT SEQUENCE
      {   networkNodeID            [0] IMPLICIT INTEGER         OPTIONAL,
          localCallID              [1] IMPLICIT INTEGER         OPTIONAL}
                                                                 OPTIONAL}

invalid-Call-State             ERROR                 ::= 0
```

```
missing-Parameter          ERROR              ::= 1
                           PARAMETER          MissingParameter

MissingParameter           ::=     SEQUENCE
 {  missingParameterType   [0] IMPLICIT ENUMERATED
     {  missingAddPartyType    (0),
        missingOrigAddress     (1),
        missingDestAddress     (2)}                    OPTIONAL}

                           --  only the first (i.e., one) missing parameter
--                         encountered is reported in this message


invalid-Parameter          ERROR              ::= 2
                           PARAMETER          InvalidParameter


InvalidParameter           ::=     SEQUENCE
 {  invalidParameterType   [0] IMPLICIT ENUMERATED
     {  invalidAddPartyType    (0),
        invalidOrigAddress     (2),
        invalidDestAddress     (3)}                    OPTIONAL}

                           --  only the first (i.e., one) invalid parameter
                           --  encountered is reported in this message

not-Allowed                 ERROR          ::= 3
agent-Not-Logged-In        ERROR            ::= 5
resources-Unavailable      ERROR           ::= 6
```

4.15.2.5 Add party: coded example

A host requests the switch to set up a consult call with an intent to conference (so specify add party type of addConsultforConf to cause DMS to reserve a conference port) with the second party on behalf of the residential line whose number is (506-858-1212). The party to consult with is at a local number (855-2248). The request is valid and the switch will respond with a call ID (if the DN was already associated, the call ID of the first leg call is used).

INVOKE DMS ===== > HOST

Table 169 add party invoke example

Hex Dump			**Information Element**	**Comments**
Tag	**Length**	**Content**		
A1	24		INVOKE	this message is 36 bytes decimal
02	01	1D	InvokeID	the InvokeID is 29 decimal
02	01	0C	Operation	it operation value is dv-Add-Party [12] decimal
30	1C		Argument	the AddPartyArgument is of type sequence and is 28 bytes long decimal
80	01	01	AddPartyType	The add party type is add consult for Conference.
A2	0C		OrigAddress	the OrigAddress is a constructor and is 12 bytes long
80	01	21	PositionID	the agent’s PositionID is 33 decimal
A3	09		DestAddress	he destination address is of type constructor and is 9 bytes long decimal
81	07	38 35 35 32 32 34 38	DialedDigits	the dialable DN is 855-2248
Hex Dump = A1 19 02 01 03 02 01 0C 30 11 80 01 00 A2 03 80 01 21 A3 07				

RETURN-RESULT DMS =====> HOST

Table 170 add party return result example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	1D	InvokeID	the InvokeID is 29 decimal
30	0E		ReturnErrorSequence	this is a sequence 14 bytes long decimal
02	01	0C	Operation	its operation is dv-Add-Party [12] decimal
30	09		AddPartyResult	the AddPartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this of type sequence 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 03 30 0E 02 01 0C 30 09 A0 07 80 01 01 81 02 05 55				

The following return error would be sent if an add party type of consult for conference was specified, and no conference ports were available.

RETURN-ERROR DMS =====> HOST

Table 171 add party return error example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	30	InvokeID	the InvokeID is 3 decimal
02	01	01	ErrorType	the ErrorType is MissingParameter[1]
30	03		ReturnErrorSequence	MissingParameter is a sequence 3 bytes
80	01	00	ErrorParameter	the MissingParameter missing AddPartyType(0)
Hex Dump = A3 0B 02 01 03 02 01 01 30 03 80 01 00				

RETURN-ERROR DMS =====> HOST

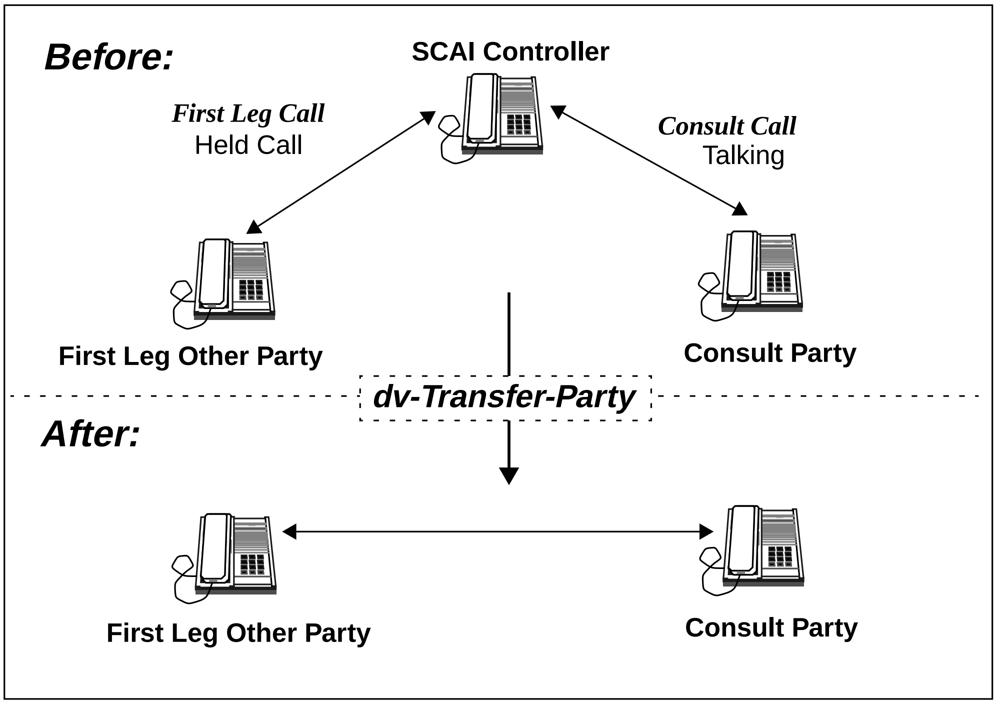
Table 172 add party return error example 2

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	1D	InvokeID	the InvokeID is 29 decimal
02	01	06	Error Value	Error Value is 6 representing Resources Unavailable.
Hex Dump = A3 06 02 01 1D 02 01 06				

4.15.2.6 MDC/RES dv-Transfer-Party (operation value "D" hex)

A transfer party message will allow transferring a call from a SCAI controller to a consult party, connecting the first leg other party with the consult party. This may be done either after the consult call has been set up or after the conference call has been set up. In either case the call need not be answered on the consult leg.

Figure 54 SCAI transfer following consultation



The dv-Transfer-Party message sent by the host will specify the DN in station number format to transfer the call from.

The switch will perform a series of checks to ensure that the request is valid. The switch will verify:

- The switch-host session subscribes to the dv-Transfer-Party message.
- The mandatory parameter (for the ECM application) of the message is present in the message.
- Verify the service version.
- The origination address specified is within the same customer group as the SCAI session that the host is connected to and is a valid residential DN, Centrex DN, or an ACD agent position ID.
- When a residential or Centrex DN is specified as the origination address, verify the line has been subscribed to this service.
- The user is in the appropriate state to initiate a transfer, e.g., the set’s DN specified is not idle and is the SCAI controller of a consult call or a conference call.
- The resources exist to transfer the call.

If the request to initiate the transfer is valid, then the SCAI controller will be released (idled) and the first leg other party will be connected to the consult party. After validation is completed, the switch sends a Return-Result (e.g., positive acknowledgment) message to the host. This message contains the network callid of the call.

At this point, there will be no more SCAI signaling done in connection with this request. Should anything cause the request to fail (e.g., no software resources), inbound signaling will be given to the user.

When the call has been transferred, if the originator’s DN has been dv-DN-Associated to a host session, a dv-Call-Released-U message will be sent with the same network callid specified in the return result of the add party message and a release reason of call transferred. Also, the network callid of the transferred call (transfer party return result) will be the same network callid sent as part of the add party return result.

If the originator and terminator are on different switches and the terminator is associated to a CompuCALL session, the network callid will not be the same for the call event messages relating to the add/transfer party requests on another switch. Currently, CompuCALL is not implemented to support network exchange of the CompuCALL information like network callid.

CompuCALL takes no action if it receives a Reject message in response to a Return-Result or Return-Error message.

4.15.3 Transfer party message parameters

This message is received when the host is requesting to transfer a consulted party to the other party involved in the first leg call. A brief description of the parameters of the dv-Transfer-Party invoke follows:

Table 1 Transfer Party Parameter Descriptions

Parameter Name	M/O[a]	Description
origAddress	M	This parameter identifies the originator which is the SCAI controller of the call to be transferred. It is defined as a choice parameter, which may be either an ACD agent position ID or the station number of the device to transfer the call from for Centrex and RES lines (in which case it is an implicit octet string). This is the SCAI controller.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

At the SCAI communications protocol level this parameter is optional. At the ECM application level, this origAddress parameter is mandatory.

4.15.3.1 Transfer party return result

If the requested transfer party was a valid request and the resources existed to comply with the request, the switch will respond by sending the host a return result with the following parameter:

Parameter Name	M/O[a]	Description
networkCallID	M	This parameter uniquely identifies the call within the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch or host generated messages that relate to the same call (e.g., the dv-Call-Released message).

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The network callid will be the same for all stages of the consult, conference and transfer that may occur via SCAI messaging. If the origination or destination address is associated via dv-DN-Associate, then the network callid created for the first leg call will be the same network callid used for these SCAI messages return-result.

4.15.3.2 Transfer party return error

If the requested transfer party was an invalid request or the resources did not exist to comply with the request, the switch will respond by sending the host a return error with the following parameter:

Parameter Name	M/O[a]	Description
error	M	This parameter uniquely identifies the type of error that occurred to cause the request to fail, e.g., missing or invalid parameter, or invalid call state.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The types of errors that may be encountered are detailed in the previous sections describing each of these messages.

4.15.3.3 Transfer party error conditions

If the request to initiate the transfer was unsuccessful, the switch sends a Return-Error (e.g., negative acknowledgment) message to the host indicating the failure is due to:

1 Customer is not allowed to initiate a transfer party request because the link has not subscribed to the service. {not allowed}

2 A mandatory parameter is missing, the origination address. {missing parameter and specify the origination address}

3 The service version is a previous version to SCAI07. {invalid parameter: origination address}

4 The origination address specified is not within the same customer group as the SCAI session that the host is connected to or is not a valid residential DN, Centrex DN, or an ACD agent position ID. {invalid parameter: origination address}

5 When a residential or Centrex DN, if the line has not been subscribed to this service via the ECM line sub-options, then the transfer party will not be allowed. {not allowed}

6 End-user not in the appropriate state to initiate a transfer party, e.g., the set’s DN specified is idle or is not the controller of a SCAI consult or three-way call. {invalid Call State}

7 Transfer Party invoke already in progress for agent. {invalid Call State}

8 Insufficient resources to initiate the transfer. {resources unavailable}

Note: This list is presented in the order of priority since only the first error condition (including the missing/invalid parameter) encountered by the switch is included in the return error message.

4.15.3.4 dv-Transfer-Party ASN.1 encoding

```
dv-Transfer-Party           OPERATION
                            ARGUMENT          TransferPartyArgument
                            RESULT            TransferPartyResult
                            ERRORS            {  invalid-Call-State,
                                                 missing-Parameter,
                                                 invalid-Parameter
                                                 not-Allowed
                                                 agent-not-Logged-in}
::= 13

TransferPartyArgument       ::=  SEQUENCE
 {   origAddress            [2] AddressType                  OPTIONAL}

AddressType                 :: =      CHOICE
 {   positionID             [0] IMPLICIT INTEGER,
     dialedDigits           [1] IMPLICIT IA5STRING,
     stationNumber          [2] IMPLICIT OCTET STRING,
     q931Address            [3] Q931AddressType}
Q931AddressType             :: =      SEQUENCE
 {   numberTypeNumberPlan             [0] OCTET STRING       OPTIONAL,
     presentationScreeningIndicator   [1] OCTET STRING       OPTIONAL,
     digits                           [2] IA5 STRING         OPTIONAL}


TransferPartyResult         ::=  SEQUENCE
 {   networkCallID          [0]  IMPLICIT SEQUENCE
       {   networkNodeID        [0] IMPLICIT INTEGER         OPTIONAL,
           localCallID          [1] IMPLICIT INTEGER         OPTIONAL}
                                                             OPTIONAL}
invalid-Call-State          ERROR             ::= 0

missing-Parameter           ERROR             ::= 1
                            PARAMETER         MissingParameter

MissingParameter            ::=       SEQUENCE
 {   missingParameterType   [0] IMPLICIT ENUMERATED
       {   missingOrigAddress   [1]}                         OPTIONAL}

invalid-Parameter           ERROR             ::= 2
                            PARAMETER         InvalidParameter
InvalidParameter            ::=       SEQUENCE
 {   invalidParameterType   [0] IMPLICIT ENUMERATED
       {   invalidOrigAddress   (2)}                         OPTIONAL}
not-Allowed                 ERROR             ::= 3

agent-Not-Logged-In         ERROR             ::= 5
```

4.15.3.5 dv-Transfer-Party: coded example

After the consult call is set up, the host may transfer the call. The host requests to transfer the call being held to the consult party, dropping the controller from the call, by sending dv-Transfer-Party with invokeID = 6. After verifying the request, the switch sends Return-Result that the transfer request will be initiated. If the switch receives the dv-Transfer-Party message with a missing OrigAddress, it sends a Return-Error message that the OrigAddress parameter is missing. Similarly, if it receives the message when the agent is not logged in, it sends a Return-Error that the agent is not logged in

INVOKE DMS =====> HOST

Table 173 dv-Transfer-Party invoke example

Hex Dump			**Information Element**	**Comments**
Tag	**Length**	**Content**		
A1	0D		INVOKE	this message is 13 bytes decimal
02	01	06	InvokeID	the InvokeID is 6 decimal
02	01	0D	Operation	it operation value is dv-Transfer-Party [13] decimal
30	05		Argument	the TransferPartyArgument is of type sequence and is 5 bytes long decimal
A2	03		OrigAddress	the OrigAddress is AddressType (CHOICE)
80	01	21	PositionId	the agent's PositionId is 33 decimal
Hex Dump = A1 0D 02 01 06 02 01 0D 30 05 A2 03 80 01 21				

RETURN-RESULT DMS =====> HOST

Table 174 dv-Transfer-Party return result example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	06	InvokeID	the InvokeID is 6 decimal
30	0E		ReturnErrorSequence	this is a sequence 14 bytes long decimal
02	01	0D	Operation	its operation is dv-Transfer-Party [13] decimal
30	09		TransferPartyResult	the TransferPartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this is of type sequence and is 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 06 30 0E 02 01 0D 30 09 A0 07 80 01 01 81 02 05 55				

The following return error would be sent if a transfer party was requested on a 3WC set initiated conference call.

RETURN-ERROR DMS =====> HOST

Table 175 dv-Transfer-Party return error example 1

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	06	InvokeID	the InvokeID is 6 decimal
02	01	01	ErrorType	the ErrorType is MissingParamter[1]
30	03		ReturnErrorSequence	this is a sequence 3 bytes long decimal
80	01	02	ErrorParameter	the MissingParameter is MissingOrigAddress (2)
Hex Dump = A3 0B 02 01 06 02 01 01 30 03 80 01 02				

RETURN-ERROR DMS =====> HOST

Table 176 dv-Transfer-Party return error example 2

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	06	InvokeID	the InvokeID is 6 decimal
02	01	05	ErrorValue	the ErrorValue is Agent-Not-Logged-In[5]
Hex Dump = A3 06 02 01 06 02 01 05				

4.15.3.6MDC/RES dv-Conference-Party (operation value "F" hex)

Conference Party ASN.1 Encoding

```
dv-Conference-PartyOPERATION
          ARGUMENT      ConferencePartyArgument
          RESULT        ConferencePartyResult
          ERRORS        {invalid-Call-State, missing-Parameter,
                          invalid-Parameter, not-Allowed,
                          agent-not-Logged-In,
                          resources-Unavailable}
::= 15
ConferencePartyArgument ::=SEQUENCE
   {origAddress      [2]  AddressType                   OPTIONAL}
   AddressType       ::=      CHOICE
     {positionID     [0]  IMPLICIT INTEGER,
      dialedDigits   [1]  IMPLICIT IA5String,
      stationNumber[2]  IMPLICIT OCTET STRING,
      q931Address    [3]  Q931AddressType}
   ConferencePartyResult::=SEQUENCE
     {networkCallID[0]  IMPLICIT SEQUENCE
      {networkNodeID[0]  IMPLICIT INTEGER  OPTIONAL,
          localCallID[1]  IMPLICIT INTEGER    OPTIONAL}
                                              OPTIONAL}
   invalid-Call-State  ERROR              ::= 0
   missing-ParameterERROR                ::= 1
                       PARAMETER          MissingParameter
   MissingParameter ::=       SEQUENCE
    {missingParameterType[0]IMPLICIT ENUMERATED
       {missingOrigAddress(1)}                OPTIONAL}
   invalid-Parameter  ERROR               ::= 2
                       PARAMETER          InvalidParameter
   InvalidParameter   ::=      SEQUENCE
    {invalidParameterType[0]IMPLICIT ENUMERATED
      {invalidOrigAddress   (2)}              OPTIONAL}
   not-Allowed        ERROR               ::= 3
   agent-Not-Logged-InERROR               ::= 5
   resources-UnavailableERROR             ::= 6
```

4.15.3.7 MDC/RES dv-Conference-Party: coded example

A host has requested the switch to set up a consult call via add party setting up (506) 858-1212 as a SCAI controller, following which the host has requested the DMS to conference all three parties. The following encoding would be sent from the host to the DMS to cause the consult leg call to be conferenced with the first leg call. The switch will respond with the same call ID sent as a result of the add party result.

INVOKE DMS <===== HOST

Table 177 dv-Conference-Party invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	16		INVOKE	this message is 32 bytes decimal
02	01	1E	InvokeID	the InvokeID is 30 decimal
02	01	0F	Operation	it operation value is dv-Conference-Party [15] decimal
30	0E		Argument	the ConferencePartyArgument is of type sequence and is 14 bytes long decimal
A2	0C		OrigAddress	the OrigAddress is a constructor and is 12 bytes long
82	0A	35 30 36 38 35 38 31 32 31 32	stationNumber	the station number 506-858-1212
Hex Dump = A1 16 02 01 1E 02 01 0F 30 0E A2 0C 82 0A 35 30 36 38 35 38 31 32 31 32				

RETURN-RESULT DMS =====> HOST

Table 178 dv-Conference-Party return result example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	1E	InvokeID	the InvokeID is 30 decimal
30	0E		ReturnResult Sequence	this is a sequence 14 bytes long decimal
02	01	0F	Operation	its operation is dv-Conference-Party [15] decimal
30	09		ConferenceParty Result	the ConferencePartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this is of type sequence and is 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 1E 30 0E 02 01 0F 30 09 A0 07 80 01 01 81 02 05 55				

The following return error would be sent if conference party was requested on a 3WC set initiated consult leg call.

RETURN-ERROR DMS =====> HOST

Table 179 dv-Conference-Party return error example 1

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	06		RETURN-ERROR	this message is 6 bytes long decimal
02	01	1E	InvokeID	the InvokeID is 30 decimal
02	01	00	Error Value	the error value is 0 representing invalid call state
Hex Dump = A3 06 02 01 1E 02 01 00				

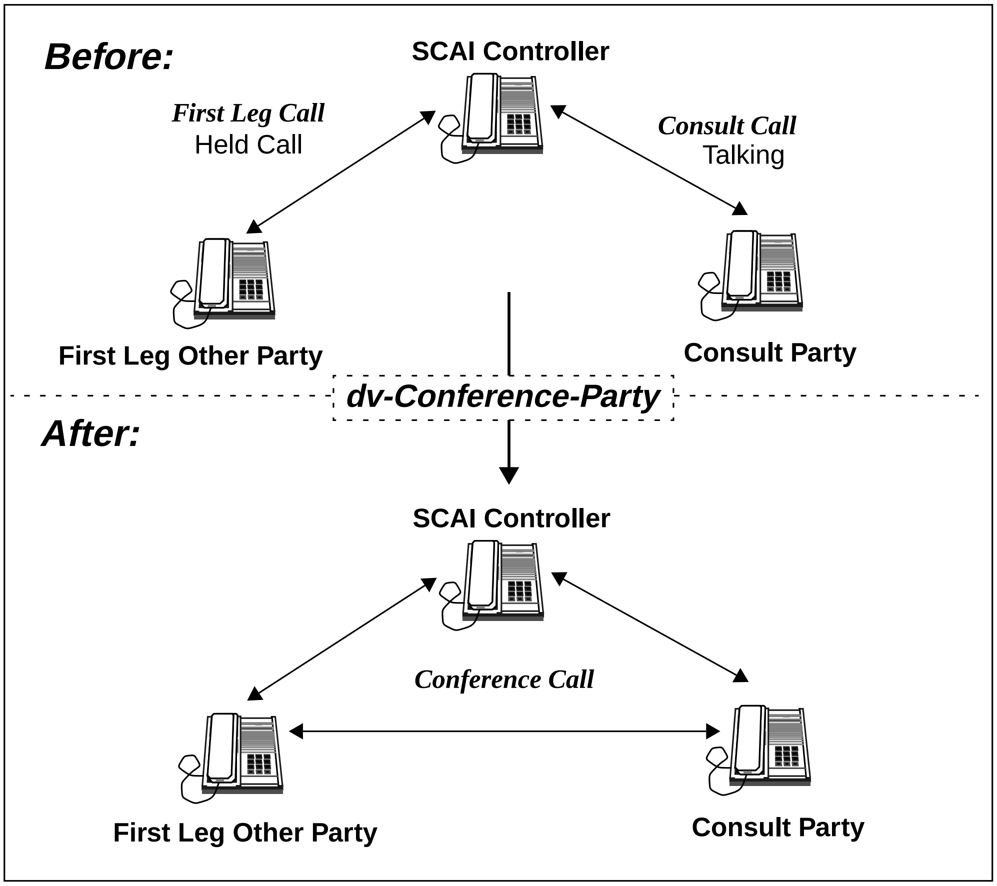
RETURN-ERROR DMS =====> HOST

Table 180 dv-Conference-Party return error example 2

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	03	InvokeID	the InvokeID is 3 decimal
02	01	05	ErrorValue	the ErrorValue is Agent-Not-Logged-In[5]
Hex Dump = A3 06 02 01 03 02 01 05				

A conference party message will allow conferencing a call connecting a SCAI controller, a consult party, and first leg other party. This may be done after the consult call has been set up. The call need not be answered on the consult leg prior to requesting a conference.

Figure 55 SCAI conference following consultation



The dv-Conference-Party message sent by the host will specify the DN in station number format to conference the call from.

The switch will perform a series of checks to ensure that the request valid. The switch will verify:

1 The switch-host session subscribes to the dv-Conference-Party message.

2 The mandatory parameter (for the ECM application) of the message is present in the message.

3 Verify the service version.

4 The origination address specified is within the same customer group as the SCAI session that the host is connected to and is a valid residential DN, Centrex DN, or an ACD agent position ID.

5 When a residential or Centrex DN is specified as the origination address, verify the line has been subscribed to this service.

6 The End-user is in the appropriate state to initiate a conference, e.g., the set's DN specified is not idle and is the SCAI controller of a consult call.

7 The resources exist to conference the call.

If the request to initiate the conference is valid, then the switch will reserve conference ports (if not already done during the Add-Party phase), and set up the three way call connecting all three parties to the appropriate conference circuits. After validation is completed, the switch sends a Return-Result (e.g., positive acknowledgment) message to the host. This message contains the network callid of the call.

A Return Result does not guarantee success of the SCAI conference, but the initial verification has passed. Failure might still occur due to system resource unavailability, for example.

At this point, there will be no more SCAI signaling done in connection with this request. Should anything cause the request to fail (e.g., no software resources), inbound signaling will be given to the user.

When the call has ended at any time, if the originator's DN has been dv-DN-Associated to a host session, a dv-Call-Released-U message will be sent with the same network callid specified in the return result of the add party message. Also, the network callid of the conferenced call (conference party return result) will be the same network callid sent as part of the add party return result.

If the originator and terminator are on different switches and the terminator is associated to a CompuCALL session, the network callid will not be the same for the call event messages relating to the add/conference party requests on another switch. Currently, CompuCALL is not implemented to support network exchange of the CompuCALL information like network callid.

CompuCALL takes no action if it receives a Reject message in response to a Return-Result or Return-Error message.

4.15.4 Conference party message parameters

This message is received when the host is requesting to conference a consulted party with the other party involved in the first leg call and the SCAI controller.

A brief description of the parameters of the dv-Conference-Party invoke is given below: .

Table 181 Conference party parameter descriptions

Parameter Name	M/O[a]	Description
origAddress	M	This parameter identifies the originator which is the SCAI controller of the call to be conferenced. It is defined as a choice parameter, which may be either an ACD agent position ID or the station number of the device to conference the call on for Centrex and RES lines (in which case it is an implicit octet string). This is the SCAI controller.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

At the SCAI communications protocol level this parameter is optional. At the ECM application level, this origAddress parameter is mandatory.

4.15.4.1 Conference party return result

If the requested conference party was a valid request and the resources existed to comply with the request, the switch will respond by sending the host a return result with the following parameter:

Parameter Name	M/O[a]	Description
networkCallID	M	This parameter uniquely identifies the call within the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch or host generated messages that relate to the same call (e.g., the dv-Call-Released message).

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The network callid will be the same for all stages of the consult, conference and transfer that may occur via SCAI messaging. If the origination or destination address is associated via dv-DN-Associate, then the network callid created for the first leg call will be the same network callid used for these SCAI messages return-result.

4.15.5 Conference party return error

If the requested conference party was an invalid request or the resources did not exist to comply with the request, the switch will respond by sending the host a return error with the following parameter:

Parameter Name	M/O[a]	Description
error	M	This parameter uniquely identifies the type of error that occurred to cause the request to fail, e.g., missing or invalid parameter, or invalid call state.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The types of errors that may be encountered are detailed in the previous sections describing each of these messages.

Conference party error conditions

If the request to initiate the conference was unsuccessful, the switch sends a Return-Error (negative acknowledgment) message to the host indicating the failure is due to:

1 Customer is not allowed to initiate a conference party request because the link has not subscribed to the service. {not allowed}

2 A mandatory parameter is missing, the origination address. {missing parameter and specify the origination address}

3 The service version is a previous version to SCAI07. {invalid parameter: origination address}

4 The origination address specified is not within the same customer group as the SCAI session that the host is connected to or is not a valid residential DN, Centrex DN, or an ACD agent osition ID. {invalid parameter: origination address}

5 When a residential or Centrex DN, if the line has not been subscribed to this service via the ECM line suboptions, the conference party will not be allowed. {not allowed}

6 User not in the appropriate state to initiate a conference party, e.g., the set’s DN specified is idle or is not the controller of a SCAI consult call. {invalid Call State}

7 Conference party invoke already in progress for agent. {invalid call state}

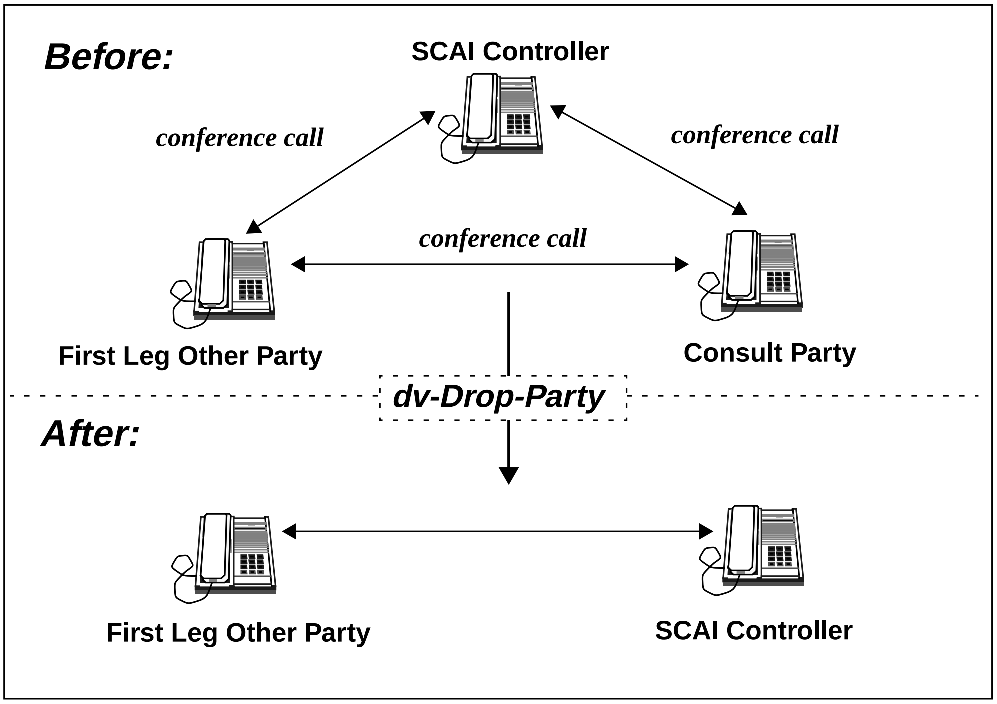
8 Insufficient resources to initiate the conference. {resources unavailable}

This list is presented in the order of priority since only the first error condition (including the missing/invalid parameter) encountered by the switch is included in the Return Error message.

4.15.5.1 MDC/RES dv-Drop-Party (operation value "E" hex)

A drop party message allows releasing a consult party connected to a SCAI controller, reconnecting the SCAI controller and the first leg other party. This may be done after the consult or conference call has been set up. The call need not be answered on the consult leg before requesting a drop consult party.

Figure 56 SCAI drop party following conference



The dv-Drop-Party message sent by the host will specify the DN in station number format to drop the consult party from.

ASN.1 Encoding

```
    dv-Drop-PartyOPERATION

        ARGUMENTDropPartyArgument
        RESULTDropPartyResult
        ERRORS{invalid-Call-State,

            missing-Parameter,
            invalid-Parameter
            not-Allowed
            agent-not-Logged-In}
    ::= 14

    DropPartyArgument::=  SEQUENCE
```

```
                  {   dropPartyType[0] IMPLICIT ENUMERATED
                  {   dropConsultParty(0)}OPTIONAL,
                  origAddress[2] AddressTypeOPTIONAL}

                  AddressType:: =  CHOICE
                  {   positionID[0] IMPLICIT INTEGER,
                      dialedDigits[1] IMPLICIT IA5STRING,
                      stationNumber[2] IMPLICIT OCTET STRING,
                  q931Address[3] Q931AddressType}


                  DropPartyResult ::=SEQUENCE
                  {   networkCallID[0] IMPLICIT SEQUENCE
                  {   networkNodeID[0] IMPLICIT INTEGER OPTIONAL,
                      localCallID[1] IMPLICIT INTEGER OPTIONAL},
                  OPTIONAL}

                  invalid-Call-State ERROR::= 0

                  missing-Parameter ERROR::= 1
                      PARAMETERMissingParameter

                  MissingParameter ::= SEQUENCE
                  {   missingParameterType [0] IMPLICIT ENUMERATED
                  {   missingDropPartyType(0),
                      missingOrigAddress(1}OPTIONAL}

                  -- only the first (i.e., one) invalid parameter
                  -- encountered is reported in this message



                  invalid-ParameterERROR::= 2

                      PARAMETER InvalidParameter

InvalidParameter              ::=     SEQUENCE
 {  invalidParameterType      [0] IMPLICIT ENUMERATED
     {  invalidDropPartyType      (0),
        invalidOrigAddress        (2)}                     OPTIONAL}

                              --  only the first (i.e., one) invalid parameter
                              --  encountered is reported in this message

not-Allowed                   ERROR               ::= 3

agent-Not-Logged-In           ERROR               ::= 5
```

4.15.5.2 MDC/RES dv-Drop-Party: coded example

A host has requested the switch to set up a consult call via add party setting up (506) 858-1212 as a SCAI controller, following which the host has requested the DMS to conference all three parties. The following encoding would be sent from the host to the DMS to cause the consult party to be dropped from 858-1212 and be connected to the other party in the first leg call. The switch will respond with the same call ID sent as a result of the add party result.

INVOKE DMS <===== HOST

Table 182 dv-Drop-Party invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	12		INVOKE	this message is 18 bytes decimal
02	01	1F	InvokeID	the InvokeID is 31 decimal
02	01	0E	Operation	it operation value is dv-Drop-Party [14] decimal
30	11		Argument	the DropPartyArgument is DropConsultLeg
80	01	00	DropPartyType	the drop party type is a primitive enumerated value representing drop consult party
A2	0C		OrigAddress	the origination address is of type constructor and is 12 bytes long
82	0A	35 30 36 38 35 38 31 32 31 32	stationNumber	the station number is 506-858-1212
Hex Dump = A1 12 02 01 1F 02 01 0E 30 11 80 01 00 A2 0C 82 0A 35 30 36 38 35 38 31 32 31 32				

RETURN-RESULT DMS =====> HOST

Table 183 dv-Drop-Party return result example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	13		RETURN RESULT	this message is 19 bytes long decimal
02	01	1F	InvokeID	the InvokeID is 31 decimal
30	0E		ReturnResultSequence	this is a sequence 14 bytes long decimal
02	01	0E	Operation	its operation is dv-Drop-Party [14] decimal
30	09		DropPartyResult	the TransferPartyResult is of type sequence 9 bytes long decimal
A0	07		NetworkCallId	this is of type sequence and is 7 bytes long
80	01	01	NetworkNodeId	the NetworkNodeId is 1
81	02	05 55	LocalCallId	the LocalCallId is 1365 decimal
Hex Dump = A2 13 02 01 1F 30 0E 02 01 0E 30 09 A0 07 80 01 01 81 02 05 55				

The following return error would be sent if drop party was requested on a 3WC set initiated conference call.

RETURN-ERROR DMS =====> HOST

Table 184 dv-Drop-Party return error example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A3	06		RETURN-ERROR	this message is 6 bytes long
02	01	1F	InvokeID	the InvokeID is 8 decimal
02	01	00	ErrorType	Error value is 0 representing invalid call state.
Hex Dump = A3 06 02 01 1F 02 01 00				

4.15.5.3 dv-Drop-Party validations

The switch will perform a series of checks to ensure that the request valid. The switch will verify:

1. The switch-host session subscribes to the dv-Drop-Party message.
2. The mandatory parameters (for the ECM application) of the message are present in the message.
3. The service version.
4. The origination address specified is within the same customer group as the SCAI session that the host is connected to and is a valid residential DN, Centrex DN, or an ACD agent Position ID.
5. When a residential or Centrex DN is specified as the origination address, verify the line has been subscribed to this service.
6. The user is in the appropriate state to initiate a drop party from, e.g., the set's DN specified is not idle and is the SCAI controller of a consult or conference call.
7. The resources exist to drop the consult party.

If the request to initiate the drop is valid, then the switch will release the consult party, and reconnect the SCAI controller with the first leg other party. After validation is completed, the switch sends a Return-Result (e.g., positive acknowledgment) message to the host. This message contains the network callid of the call.

At this point, there will be no more SCAI signaling done in connection with this request. Should anything cause the request to fail (e.g., no software resources), inbound signaling will be given to the user.

When the drop party has successfully completed, if the originator's DN has been dv-DN-Associated to a host session, a dv-Call-Released-U message will be sent with the same network callid specified in the return result of the add party message with release reason of either party dropped or party dropped no answer. Also, the network callid of the reconnected call (drop party return result) will be the same network callid sent as part of the add party return result.

If the originator and terminator are on different switches and the terminator is associated to a CompuCALL session, the network callid will not be the same for the call event messages relating to the add/drop party requests on another switch. Currently, CompuCALL is not implemented to support network exchange of the CompuCALL information like network callid. This does not in any way imply that the user will be unable to consult/transfer/conference across switches.

CompuCALL takes no action if it receives a reject message in response to a Return-Result or Return-Error message.

4.15.6 Drop party message parameters

This message is received when the host is requesting to drop the consulted party of the consult leg call or of the conference call. A brief description of the parameters of the dv-Drop-Party invoke follows:

Table 185 Drop party parameter descriptions

Parameter Name	M/O[a]	Description
dropPartyType	M	This parameter indicates who is to be dropped and how. The only valid drop party type is dropConsultParty. This is a request to drop the consult party and keep the connection between the first leg other party and the SCAI controller.
origAddress	M	This parameter identifies the originator which is the SCAI controller to be dropped. It is defined as a choice parameter, which may be either an ACD agent position ID or the station number of the device to drop the consult call from for Centrex and RES lines (in which case it is an implicit octet string). This is the SCAI controller.

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

All these parameters are optional at the SCAI communications protocol level. At the ECM application level the origAddress and dropPartyType are mandatory parameters.

4.15.7 Drop party return result

If the requested drop party was a valid request and the resources existed to comply with the request, the switch will respond by sending the host a return result with the following parameter:

Parameter Name	M/O[a]	Description
networkCallID	M	This parameter uniquely identifies the call within the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch or host generated messages that relate to the same call (e.g., the dv-Call-Released message).

[a] M/O--M represents a mandatory parameter, while O represents an optional parameter.

The network callid will be the same for all stages of the consult, conference and transfer that may occur via SCAI messaging. If the origination or destination address is associated via dv-DN-Associate, then the network callid created for the first leg call will be the same network callid used for these SCAI messages return-result.

4.15.8 Drop party return error

If the requested drop party was an invalid request or the resources did not exist to comply with the request, the switch will respond by sending the host a return error. The types of errors that may be encountered are detailed in the previous sections describing each of these messages.

4.15.8.1Drop party error conditions

If the request to drop the consult party was unsuccessful, the switch sends a Return-Error (e.g., negative acknowledgment) message to the host indicating the failure is due to:

1 Customer is not allowed to initiate a drop party request because the link has not subscribed to the service. {not allowed}

2 A mandatory parameter is missing, the origination address or the drop party type. {missing parameter and which parameter is missing}

3 The service version is a previous version to SCAI07. {invalid parameter: origination address}

> ***Note:*** The origination address specified is not within the same customer group as the SCAI session that the host is connected to or is not a valid residential DN, Centrex DN, or an ACD agent Position ID. {invalid parameter: origination address}

4 When a residential or Centrex DN, if the line has not been subscribed to this service via the ECM line sub-options, then the drop party will not be allowed. {not allowed}

5 End-user not in the appropriate state to initiate a drop party, e.g., the set’s DN specified is idle or is not the controller of a SCAI consult or conference call. {invalid Call State}

6 Drop Party invoke already in progress for agent. {invalid Call State}

7 Insufficient resources to initiate the drop party. {resources unavailable}

> ***Note:*** This list is presented in the order of priority since only the first error condition (including the missing/invalid parameter) encountered by the switch is included in the return error message.

4.15.9 MDC/RES dv-Call-Consult-Originated-U

ASN.1 Encoding

```
        dv-Call-Consult-Originated-U  OPERATION
                                      ARGUMENT CallConsultOrigArgument
        ::= 31
        CallConsultOrigArgument ::=     SEQUENCE
            { networkCallID         [0] IMPLICIT SEQUENCE
                {  networkNodeID        [0] IMPLICIT INTEGER            OPTION-
            AL,
                   localCallID          [1] IMPLICIT INTEGER            OPTION-
            AL}
                                                                        OPTION-
            AL,

            deviceID[1] AddressTypeOPTIONAL,

            consultDN[2] AddressTypeOPTIONAL}



        AddressType::= CHOICE
         {positionID            [0]     IMPLICIT  INTEGER,
           dialedDigits         [1]     IMPLICIT  IA5String,
           stationNumber        [2]     IMPLICIT  OCTET STRING,
           q931Address          [3]     Q931AddressType}
```

4.15.9.1 dv-Call-Consult-Originated-U: coded example

INVOKE DMS =====> HOST

Table 186 dv-Call-Consult-Originated-U

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	2E		INVOKE	the message length is 46 bytes decimal
02	02	03 EE	InvokeID	its InvokeID is 1006
02	01	1F	Operation	it operation value is dv-Call-Consult-Originated-U [31] decimal
30	25		Argument	CallConsultOrigArgument is of type sequence and is 37 bytes long decimal
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
A1	0C		deviceID	deviceID is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 34 30 30	stationNumber	the DN of the phone is 613-621-2400
A2	0D		consultDN	consultDN is of type constructor and is 13 bytes long
81	0B	31 36 31 33 36 32 31 32 32 30 30	dialed digits	the DN of the phone is 1-613-621-2200
Hex Dump = A1 2E 02 02 03 EE 02 01 1F 30 25 A0 06 80 01 01 81 01 02 A1 0C 82 0A 36 31 33 36 32 31 32 34 30 30 A2 0D 8 0B 31 36 31 33 36 32 31 32 32 30 30				

These manual 3WC/CXR interaction events are reported via these messages to the host computer monitoring the controller in both intra-office three party and inter-office three party calls. Also, conference and transfer events are reported to the host computer monitoring the terminator of the second leg of the call, only under intra-office three party calls. No messages are reported for the originator of the first leg of the three-way call under any circumstances, except if the originator of the first leg of the call is also the controller of the three party call, in which case messages belonging to the controller is sent.

The switch performs a series of checks before sending the 3WC/CXR call event message. The switch will verify:

1 The switch-host session subscribes to the call event message.

2 The mandatory parameter (for the ECM application) of the message is present

3 The associated device address specified is within the same customer group as the SCAI session that the host is connected to and is a valid residential DN, Centrex DN, or ACD DN.

4 The associated device address of the residential DN, Centrex DN, or ACD DN has subscribed to the call event message.

5 The service version is at least SCAI09 (The dv-Application-Logon message contains a parameter, service version, which uniquely identifies the protocol version stream of messages which will be sent during the application logged on session).

The switch sends the dv-Call-Consult-Originated-U message for the 3WC/CXR controller when the 3WC/CXR controller activates (hits the 3WC/CXR key) a consult call to a third party and all digits have been collected. A consult call means that a line is involved in an active call and requests to put the original call on hold, and originate a consult call.

The dv-Call-Consult-Originated-U message will be sent in both intra and inter switch consult leg call scenarios.

The 3WC/CXR controller must be associated in order to receive this incoming call event message. Association to a line is done from the host computer by sending a DN-Associate message containing the DN of the line.

A brief description of the parameters of the dv-Consult-Originated-U is given below:

Table 187 Call Consult originated parameter descriptions

Parameter Name	M/O[a]	Description
networkCallID	O	CompuCALL generated call ID composed of a network node ID (taken from table scaigrp) and a local call Id (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session environment; otherwise the call ID already assigned to this call will be used.
deviceID	M	This parameter identifies the address of the 3WC/CXR controller in station number format. The 3WC/CXR controller is the party that has 3WC/CXR assigned.
consultDN	M	This parameter identifies the consulted party in dialed digits format. The reason dialed digits is to make the host computer aware what digits the controller dialed to originate the call.

[a] M/O--M represents a mandatory parameter at the application level. It will be provided in the message. While O represents an optional parameter at the application level that can be subscribed to in table SCAISSRV.

4.15.10MDC/RES dv-Call-Conferenced-U

4.15.10.1 dv-Call-Conferenced-U ASN.1 encoding

```
dv-Call-Conferenced-UOPERATION
                    ARGUMENT    CallConferencedArgument
::= 32
CallConferencedArgument ::=    SEQUENCE
   { networkCallID        [0] IMPLICIT SEQUENCE
       {  networkNodeID      [0] IMPLICIT INTEGER          OPTION-
   AL,
          localCallID        [1] IMPLICIT INTEGER          OPTION-
   AL}
                                                           OPTION-
   AL,

   controllerDN[1] AddressTypeOPTIONAL,

   consultDN[2] AddressTypeOPTIONAL,

   firstLegDN[3] AddressTypeOPTIONAL}

AddressType::= CHOICE
{positionID           [0]     IMPLICIT  INTEGER,
  dialedDigits        [1]     IMPLICIT  IA5String,
stationNumber        [2]      IMPLICIT  OCTET STRING,
q931Address          [3]      Q931AddressType}
```

dv-Call-Conferenced-U: coded example

INVOKE DMS =====> HOST

Table 188 dv-Call-Conferenced-U

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	3B		INVOKE	the message length is 59 bytes long
02	02	03 EE	InvokeID	its InvokeID is 1006
02	01	20	Operation	its operation value is dv-Call-Conferenced [32] decimal
30	32		Argument	CallConferencedArgument is of type sequence and is 50 bytes long decimal
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
A1	0C		controllerDN	deviceID is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 34 30 30	stationNumber	the DN of the phone is 613-621-2400
A2	0C		consultDN	consultDN is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 32 30 30	stationNumber	the DN of the phone is 613-621-2200
A3	0C		firstLegDN	firstLegDN is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 33 30 30	stationNumber	the DN of the phone is 613-621-2300
Hex Dump = A1 3B 02 02 03 EE 02 01 20 30 32 80 01 01 81 01 02 A1 0C 82 0A 36 31 33 36 32 31 32 34 30 A2 0C 82 0A 36 31 33 36 32 31 32 32 30 30 A3 0C 82 0A 36 31 33 36 32 31 32 33 30 30				

The switch sends the dv-Call-Conferenced-U message when the 3WC/CXR controller conferences a consulted party with the originator of the first leg of a call. The controller requests that a conference call is established by hitting the 3WC/CXR key and all three parties are connected to each other resulting in a three-party conference call.

The dv-Call-Conferenced-U message will be sent for the 3WC/CXR controller, if the controller is associated and this message is sent for the controller for inter-switched and intra-switched consult leg calls.

The dv-Call-Conferenced-U message will also be sent for the consulted DN, if the consult leg call is intra-switched and the consult DN in the consult leg cal is associated. The message will not contain the firstLegDN due to privacy considerations.

A brief description of the parameters of the dv-Call-Conferenced-U is given below:

Table 189 Call conferenced parameter descriptions

Parameter Name	M/O[a]	Description
networkCallID	O	CompuCALL generated call ID composed of a network node ID (taken from table SCAIGRP) and a local call Id (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session environment; otherwise the call ID already assigned to this call will be used.
controllerDN	M	This parameter identifies the address of the 3WC/CXR controller in station number format. The 3WC/CXR controller is the party that has 3WC/CXR assigned.
consultDN	M	This parameter identifies the consulted party in station number format for intra-switch consult leg call or in dialed digits format for inter-switch consult leg call.
firstLegDN	M	This parameter identifies the party who is connected to the controller. The firstLegDN will be in station number format if the first-leg call is intra-switch or the controller is the terminating party on the first leg call. The firstLegDN will be in dialed digits format if the controller is the originator of the first leg of an inter-switch call. **NOTE:** This parameter is only mandatory for the dv-Call-Conferenced message intended for the host of the **controller**. This parameter is not included in the message intended for the host of the consult party due to privacy considerations.

[a] M/O--M represents a mandatory parameter at the application level. It will be provided in the message. While O represents an optional parameter at the application level that can be subscribed to in table SCAISSRV.

4.15.11 MDC/RES dv-Call-Transferred-U

ASN.1 Encoding

```
dv-Call-Transferred-UOPERATION

    ARGUMENT                          CallTransferredArgument

:= 33

 CallTransferredArgument:=SEQUENCE

{   networkCallID                 [0] IMPLICIT SEQUENCE

    {   networkNodeID         [0] IMPLICIT INTEGEROPTIONAL,

        localCallID           [1] IMPLICIT INTEGEROPTIONAL}

                                      OPTIONAL,

    deviceID[1] AddressType           OPTIONAL,

    otherPtyDN[2] AddressType         OPTIONAL}

AddressType:= CHOICE
{positionID             [0]     IMPLICIT  INTEGER,
   dialedDigits         [1]     IMPLICIT  IA5String,
   stationNumber        [2]     IMPLICIT  OCTET STRING,
   q931Address          [3]     Q931AddressType}
```

4.15.11.1 dv-Call-Transferred-U: coded example

INVOKE DMS ====> HOST

Table 190 dv-Call-Transferred-U invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	2D		INVOKE	the message length is 45 bytes decimal
02	02	03 EE	InvokeID	its Invoke ID is 1006
02	01	21	Operation	its operation value is dv-call-transferred-U [33] decimal
30	24		Argument	CallTransferredArg is of type sequence and is 36 bytes long decimal
A0	06		networkCallID	networkCallID is of type constructor and is 6 bytes long
80	01	01	networkNodeID	the networkNodeID is 1
81	01	02	localCallID	the localCallID is 2
A1	0C		deviceID	deviceID is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 32 30 30	stationNumber	the DN of the phone is 613-621-2200
A2	0C		.... otherPtyDN	otherPtyDN is of type constructor and is 12 bytes long
82	0A	36 31 33 36 32 31 32 33 30 30	stationNumber	the DN of the phone is 613- 621-2300
Hex dump = A1 2D02 02 03 EE 02 01 21 30 24 A0 06 80 01 01 81 01 02 A1 0C 82 0A 36 31 33 36 32 31 32 32 30 30 A2 0C 82 0A 36 31 33 36 32 31 32 33 30 30				

The switch sends the dv-Call-Transferred-U message for the consult party when a controller transfers an active call and if the controller is in the same switch. This event will be sent when the controller is dropped and the call is transferred.

The terminator of the consult leg must be associated in order to receive this incoming call event message. Association to a line is done from the host computer by sending a DN-Associate message containing the DN of the line.

A brief description of the parameters of the dv-Call-Transferred-U is given below:

Table 191 Call transferred parameter descriptions

Parameter Name	M/O[a]	Description
networkCallID	O	CompuCALL generated call ID composed of a network node ID (taken from table SCAIGRP) and a local call Id (unique call ID within the session). This will be a new call id if this is the first instance of the call within this session environment; otherwise the call ID already assigned to this call will be used.
deviceID	M	This parameter identifies the address of the party to which the call was transferred. The deviceID will be in station number format if the consult party is a residential or Centrex line. The deviceID will be in position ID format if the consult party is an ACD agent.
otherPtyDN	M	This parameter identifies the party who was the originator of the first leg of the call and was eventually transferred, in station number format. **NOTE:** This parameter will not be included in the message if the controller transfers the consult party to a conference call. (The display of the consult party's set will show conference /call transfer rather than the fIrst leg party DN.)

[a] M/O--M represents a mandatory parameter at the application level. It will be provided in the message. While O represents an optional parameter at the application level that can be subscribed to in table SCAISSRV.

4.15.12 dv-Controller-Released-U (Operation Value 48)

If a 3WC to a 2WC state transition results in a call transfer controlled by a controller, the DMS-100 switch sends a dv-Controller-Released-U message to the host that contains the DN of the releasing party.

4.15.12.1dv-Controller-Released-U ASN.1 Encoding

```
 dv_CONTROLLER_RELEASED_U                  OPERATION

 ARGUMENT                                   Ctrl_releasedargument

 := 48

Ctrl_releasedargument:=                SEQUENCE

{NetworkCallID              [0]      IMPLICIT     SEQUENCE
```

```
        {networkNodeID        [0]       IMPLICIT   INTEGER     OPTIONAL,

         localCallID          [1]       IMPLICIT        INTEGER     OPTIONAL}

                                                              OPTIONAL,

  Device_id              [1]       AddressType                    OPTIONAL

  Released_Party_info            [2]       ReleasedPartyInfoType       OPTIONAL}

 AddressType::= CHOICE
  {positionID[0]                        IMPLICIT     INTEGER,
   dialedDigits[1]                      IMPLICIT     IA5 STRING,
   stationNumber [2]                    IMPLICIT     OCTET   STRING,
   q.931Address[3]                      Q.931AddressType}

Q.931AddressType::= SEQUENCE
 {numberTypeNumberPlan     [0]      IMPLICIT   OCTET    STRING     OPTIONAL,
  presentationScreeningIndicator[1]  IMPLICIT   OCTET    STRING     OPTIONAL,
  digits [2]                          IMPLICIT      IA5 STRING       OPTIONAL}

ReleasedPartyInfoType::=CHOICE

{   POSID            [0]                UNSIGNEDINT,

    DEVICEID      [1]                   DEVICEADDRESSTYPE}

DEVICEADDRESSTYPE ::= SET OF

    { dialedDigits     [1]      IMPLICIT     IA5 String              OPTIONAL,

     stationNumber   [2]       IMPLICIT    OCTET STRING       OPTIONAL,

    name                [4]       IMPLICIT     IA5 STRING          OPTIONAL}
```

4.15.12.2 dv-Controller-Released-U Coded Example

All three parties in a 3WC are on the same switch. The controller, party A, is an ACD agent with position ID 922. The first leg of the call was initiated by party B. Party B is ECM associated at station number 919-722-1005. The controller exits the 3WC and the switch sends the dv-Controller-Released-U message to the originating party of the 3WC.

INVOKE DMS <===== HOST

Table 192 dv-Controller-Released-U invoke example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	25		INVOKE	this message is 37 bytes decimal
02	02	03FF	InvokeID	the InvokeID is 1023 decimal
02	01	30	Operation	the operation is dv-Controller-Released-U [48]
30	1C		Ctrl_releaseargument	Ctrl_releaseargument is of sequence type and is 28 bytes long
A0	06		NetworkCallID	the NetworkCallID is a constructor 6 bytes decimal
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	01	02	LocalCallID	the LocalCallID is 2 decimal
A1	0C		DeviceID	the DeviceID is a construction of 12 bytes decimal long
82	0A	39 31 39 37 32 32 31 30 30 35	Station number	the Station number of the destination device (919-722-1005)
A2	04		Released_Party_info	the Released_Party_info is a constructor 4 bytes long
80	02	039A	POSID	the Position ID of the ACD agent is 922 decimal
Hex Dump = A1 25 02 02 03 FF 02 01 30 30 1C A0 06 80 01 01 81 01 02 A1 0C 82 0A 39 31 39 37 32 32 31 30 30 35 A2 04 80 02 03 9A				

4.15.12.3 dv-Controller-Released-U parameters

The dv-Controller-Released-U message includes the following parameter:

NetworkCallID an optional parameter that identifies the call in the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

deviceID a required parameter that identifies the address of the party intended to receive the new message.

Released-Party-Info a required parameter identifies the released party by the position ID, station number or dialed digits, and the releasing party’s name, if possible. Only the controller receives the Released-Party-Info parameter. The non-controller does not receive this parameter in the message. The releasing party’s name appears in the message only when the releasing party is a RES/MDC line without presentation of name restrictions.

4.15.13 dv-nonController-Released-U (Operation Value 49)

If a 3WC to a 2WC state transition does not result from a call transfer the DMS-100 switch sends a dv-nonController-Released-U message to the host that contains the DN of the releasing party. The DN of the releasing party can consist of one of the following:

- Position ID
- station number
- dialed digits

dv-nonController-Released-U ASN.1 Encoding

```
 dv_NONCONTROLLER_RELEASED_U           OPERATION

 ARGUMENT                                     nonctrl_releasedargument

 := 49

nonctrl_releasedargument :=          SEQUENCE

{NetworkCallID              [0]          IMPLICIT   SEQUENCE

      {networkNodeID        [0]          IMPLICIT   INTEGER    OPTIONAL,

       localCallID          [1]          IMPLICIT   INTEGER    OPTIONAL}

                                                               OPTIONAL,

Device_id              [1]        AddressType                  OPTIONAL,
 Released_Party_Info       [2]       ReleasedPartyInfoType     OPTIONAL ,

  Reason                      [3]     IMPLICIT      ENUMERATED  OPTIONAL }
```

```
AddressType ::= CHOICE
  {positionID[0]                   IMPLICIT    INTEGER,
   dialedDigits[1]                 IMPLICIT    IA5 STRING,
   stationNumber [2]               IMPLICIT    OCTET  STRING,
   q.931Address[3]                 Q.931AddressType}

Q.931AddressType::= SEQUENCE
 {numberTypeNumberPlan    [0]      IMPLICIT   OCTET    STRING    OPTIONAL,
  presentationScreeningIndicator[1]  IMPLICIT   OCTET    STRING    OPTIONAL,
  digits [2]                       IMPLICIT      IA5 STRING       OPTIONAL}

ReleasedPartyInfoType::=CHOICE

{   POSID            [0]                UNSIGNEDINT,

    DEVICEID      [1]                   DEVICEADDRESSTYPE}

DEVICEADDRESSTYPE ::= SET OF

    { dialedDigits      [1]       IMPLICIT     IA5 String              OPTIONAL,

     stationNumber   [2]        IMPLICIT    OCTET STRING        OPTIONAL,

    name                  [4]        IMPLICIT     IA5 STRING           OPTIONAL}


Reason::= IMPLICIT ENUMERATED

{ first_leg_party_released  [0]

second_leg_party_released  [1] }
```

4.15.13.1 dv-nonController-Released-U Coded Example

All three parties in a 3WC are on the same switch. The controller, party A, is an ACD agent with position ID 4196. Party B (613-770-5543) calls party A to set up the first leg of the call. The controller places party B on hold and calls party C (613-621-4442). Party B releases the call while on hold. The controller and party C both receive the dv-nonController-Released-U message.

INVOKE DMS <===== HOST

Table 193 dv-nonController-Released-U (message sent to the controller)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	39		INVOKE	this message is 57 bytes decimal
02	02	03FF	InvokeID	the InvokeID is 1023 decimal
02	01	31	Operation	the operation is dv-nonController-Released-U [49]
30	30		nonctrl_releaseargument	nonctrl_releaseargument is of sequence type and is 48 bytes decimal long
A0	06		NetworkCallID	the NetworkCallID is a constructor 6 bytes decimal
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	01	03	LocalCallID	the LocalCallID is 3 decimal
A1	04		DeviceID	the DeviceID is a construction of 4 bytes decimal long
80	02	1064	POSID	the positionId of the agent is 4196
A2	1D		Released_Party_info	the Released_Party_info is a constructor 29 bytes long
A1	1B		DeviceID	the DeviceID is a construction of 27 bytes decimal long
82	0A	36 31 33 37 37 30 35 35 34 33	Station number	the Station number of the released party device (613-770-5543)
84	0D	53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53	name	the name of the released party is SHANANA JONES of length 13
83	01	00	Reason	the reason for receiving the message is 0, which indicates the call was released by the first leg party
Hex Dump = A1 39 02 02 03 FF 02 01 31 30 30 A0 06 80 01 01 81 01 03 A1 04 80 02 10 64 A2 1D A1 1B 82 0A 36 31 33 37 37 30 35 35 34 33 84 0D 53 48 41 4E 41 4E 41 20 4A 4F 4E 45 53 83 01 00				

INVOKE DMS <===== HOST

Table 194 dv-nonController-Released-U (message sent to conferenced party)

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	24		INVOKE	this message is 36 bytes decimal
02	02	03FF	InvokeID	the InvokeID is 1023 decimal
02	01	31	Operation	the operation is dv-nonController-Released-U [49]
30	1B		nonctrl_releaseargument	nonctrl_releaseargument is of sequence type and is 27 bytes decimal long
A0	06		NetworkCallID	the NetworkCallID is a constructor 6 bytes decimal
80	01	01	NetworkNodeId	the NetworkNodeId is 1 decimal
81	01	03	LocalCallID	the LocalCallID is 3 decimal
A1	0C		DeviceID	the DeviceID is a construction of 12 bytes decimal long
82	0A	36 31 33 36 32 31 34 34 34 32	Station number	the Station number of the released party device (613-621-4442)
A2	03		Released_Party_info	the Released_Party_info is a constructor 26 bytes long
83	01	00	Reason	the reason for receiving the message is 0, which indicates the call was released by the first leg party
Hex Dump = A1 24 02 02 03FF 02 01 31 30 1B A0 06 80 01 01 81 01 03 A1 0C 82 0A 36 31 33 36 32 31 34 34 34 32 A2 03 83 01 00				

4.15.13.2 dv-nonController-Released-U parameters

The dv-nonController-Released-U message includes the following parameter:

NetworkCallID an optional parameter that identifies the call in the context of the customer's network for the maximum expected duration of the call. It provides a cross-reference for other switch- or host-generated messages relating to the same call.

deviceID a required parameter that identifies the address of the party intended to receive the new message.

Released-Party-Info a required parameter identifies the released party by the position ID, station number or dialed digits, and the releasing party's name, if possible. Only the controller receives the Released-Party-Info parameter. The non-controller does not receive this parameter in the message. The releasing party's name appears in the message only when the releasing party is a RES/MDC line without presentation of name restrictions

Reason a required parameter that identifies the party that released the call. A value of 0 indicates the noncontroller of the first leg of the call performed the release. A value of 1 indicates the noncontroller of the second leg of the call performed the release.

4.16 CompuCALL MDC/RES message waiting notification

The association event message for MDC/RES lines includes one new function: dv-Message-Waiting-U

4.16.1 MDC/RES dv-Message-Waiting-U (operation value 5)

This section describes the MDC/RES dv-Message-Waiting-U message the switch uses to notify the host of a message waiting activation or deactivation event. dv-Message-Waiting-U provides a new association event message to indicate voice mail message waiting or executive voice mail message waiting activation/deactivation. This SCAI message waiting indication event for residential lines maps to a ROSE Class 5 operation and is defined as dv-Message-Waiting-U.

ASN.1 Encoding

```
    dv-Message-Waiting-U OPERATION


ARGUMENT            MessageWaitingArgument

        ::= 26


    MessageWaitingArg   ::=SEQUENCE

{deviceID[0]AddressType                          OPTIONAL,
   msgWaitInfo[1]MsgWaitInformation                  OPTIONAL}

AddressType::= CHOICE
 {positionID[0]IMPLICIT  INTEGER,
   dialedDigits[1]IMPLICIT  IA5 String,
   stationNumber [2]IMPLICIT  OCTET STRING,
   q931Address[3]Q931AddressType}

MsgWaitInformation::=  SEQUENCE
  {msgWaiting[0]IMPLICIT  MessageWaitingType   OPTIONAL,
    indicationReason[1]IMPLICIT MsgWaitIndicationType OPTIONAL}

MessageWaitingType::= ENUMERATED
 {   MsgWait(0),
     ExecMsgWait(1)}

MsgWaitIndicationType::ENUMERATED
 {activation(0),
    deactivation(1)}
```

4.16.1.1 MDC/RES dv-Message-Waiting-U indication: coded example

Following is a coded example of the dv-Message-Waiting-U message where the residential phone has a station number of (506) 858-1234. The phone’s DN has voice mail message waiting indication (MWT), and someone just finished checking his voice mail. The Invoke ID is 2.

INVOKE DMS ====> HOST

Table 195 dv-Message-Waiting-U example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	1E		INVOKE	the message length is 30 bytes decimal
02	01	02	InvokeID	its Invoke ID is 2
02	01	1A	Operation	its operation value is dv-Message-Waiting-U [26] decimal
30	16		Argument	MessageWaitingArg is of type sequence and is 22 bytes long decimal
A0	0C		deviceID	the deviceID is of type constructor and is 12 bytes long
82	0A	35 30 34 38 35 38 31 32 33 34	stationNumber	the DN of the phone is 504-858-1234
A1	06		msgWaitInfo	msgWaitInfo is a constructor and is 6 bytes long
80	01	00	msgWaiting	the message waiting is MsgWait and is 1 byte long decimal
81	01	00	indicationReason	the indicationReason is activation
Hex dump = A1 1E 02 01 02 02 01 1A 30 16 A0 0C 82 0A 35 30 34 38 35 38 31 32 33 34 A1 06 80 01 00 81 01 00				

The SCAI message waiting indication message is only supported for lines with the message waiting or executive message waiting options assigned. Currently, voice mail is the only message waiting type supported.

4.16.1.2 dv-Message-Waiting-U parameters

A brief description of the parameters of the dv-Message-Waiting-U invoke follows:

DeviceID contains the address of the device id associated for this event in station number format.

MessageWaitInfo consists of the following optional parameters:

MessageWaitType specifies the type of message waiting. Currently message waiting and executive message waiting are supported.

MessageWait identifies if the message waiting indication will be activated or deactivated.

For the host to receive incoming messages, the line must be associated. The DN-Associate message will enable the switch to send the message waiting Indication event message to the host concerning the DN.

In the scenario where a DN is associated to a session which is subscribed to message waiting indication as well as other call event messages, and the DN does not have the message waiting line option subscribed, the message waiting indication message will not be sent for that associated DN but the other call event messages will be enabled. If the associated DN does not have message waiting, and message waiting indication event message is the only event message subscribed to the CompuCALL session, the DN association will fail with a return error of not-allowed.

4.16.1.3 dv-Message-Waiting-U validations

Validation of message waiting indication message

The message waiting indication message is only sent from the Message Waiting feature software. When the Message Waiting feature is about to update the lamp on a set or apply a special message waiting tone to a 500 /2500 set, a series of checks will be performed before sending the message waiting indication message.

1 The DN is associated.

2 The switch-host session subscribes to the message waiting indication message.

3 The residential line has subscribed to the message waiting indication message.

4 The service version is at least SCAI07.

4.16.2 MDC/RES dv-MWT-Activate (operation value 45)

This section describes the MDC/RES dv-MWT-Activate message the switch uses to allow the host to control message waiting indication (MWI) activation and deactivation. This SCAI message waiting indication event for residential lines maps to a ROSE Class 2 operation and is defined as dv-MWT-Activate.

ASN.1 Encoding

```
dv-MWT-ActivateOPERATION
                ARGUMENT     MWTActivateArgument
                RESULT
                ERRORS       {  not-Allowed
                                missing-Parameter
                                invalid-Parameter
                                missing-Option
                                mismatch-MCOS
                                fail-To-Allocate-Res
                                nms-Invalid
                                nms-Bad-Address
                                nms-Blocked
                                nms-Unavailable
                                nms-Missing
                                nms-Timeout }
::= 45

MWTActivateArgument    ::=   SEQUENCE
{ destDeviceID         [0] AddressType             OPTIONAL,
  VMSDeviceID          [1] AddressType             OPTIONAL,
  indication           [2] MsgWaitIndicationType   OPTIONAL  }
```

Both AddressType and MsgWaitIndicationType are existing types shown here for completeness.

```
AddressType         ::=   CHOICE
{ positionID        [0] IMPLICIT  INTEGER,
  dialedDigits      [1] IMPLICIT  IA5String,
  stationNumber     [2] IMPLICIT  OCTET STRING,
  q.931Address      [3] Q.931AddressType  }

MsgActIndicationType  ::  ENUMERATED
{ deactivation          (0),
  activation            (1)  }

not-Allowed            ERROR      ::=   0
missing-Parameter      ERROR      ::=   1
                       PARAMETER  MissingParameter

MissingParameter       ::=   SEQUENCE
{ missingParameterType [0] IMPLICIT ENUMERATED
  {  missingDestDeviceID   (0),
     missingVMSDeviceID    (1),
     missingIndication     (2)}   OPTIONAL}

                                  -- only the first missing parameter encountered
                                  -- is reported in this message.

invalid-Parameter      ERROR      ::=   2
                       PARAMETER  InvalidParameter

InvalidParameter       ::=   SEQUENCE
```

```
{invalidParameterType      [0] IMPLICIT ENUMERATED
  {  invalidDestDeviceID       (0),
     invalidVMSDeviceID        (1),
     invalidIndication         (2)}     OPTIONAL}

                                        -- only the first invalid parameter encountered
                                        -- is reported in this message.

missing-Option             ERROR    ::=    3
mismatch-MCOS              ERROR    ::=    4
fail-To-Allocate-Res       ERROR    ::=    5
nms-Invalid                ERROR    ::=    6
nms-Bad-Address            ERROR    ::=    7
nms-Blocked                ERROR    ::=    8
nms-Unavailable            ERROR    ::=    9
nms-Missing                ERROR    ::=    10
nms-Timeout                ERROR    ::=    11
```

4.16.2.1 MDC/RES dv-MWT-Activate: coded example

Following is a coded example of the dv-MWT-Activate message where the host requests MWI activation for a destination device with a station number of (818) 722-1611. The voice mail port station number is (555)-987-1234. The Invoke ID is 5.

INVOKE DMS ====> HOST

Table 196 dv-MWT-Activate invoke example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A1	27		INVOKE	the message length is 39 bytes decimal
02	01	05	InvokeID	its Invoke ID is 5
02	01	2D	Operation	its operation value is dv-MWT-Activate [45] decimal
30	1F		Argument	MWTActivateArg is of type sequence and is 31 bytes long decimal
A0	0C		destDeviceID	the destDeviceID is of type constructor and is 12 bytes long
82	0A	36 31 38 37 32 32 31 36 31 31	stationNumber	the DN of the station destined for MWI activation or deactivation is 618-722-1611
A1	0C		VMSDeviceID	the VMStDeviceID is of type constructor and is 12 bytes long
82	0A	35 35 35 39 38 37 31 32 33 34	stationNumber	the DN of the voice mail port is 555-987-1234
82	01	01	indicationReason	the indicationReason is MWI activation
Hex dump = 27 02 01 05 02 01 2D 30 1F A0 0C 82 0A 36 31 38 37 32 32 31 36 31 31 A1 0C 82 0A 35 35 35 39 38 37 31 32 33 34 82 01 01				

RETURN-RESULT DMS ====> HOST

Table 197 dv-MWT-Activate return result example

Hex Dump			Information Element	Comments
Tag	**Length**	**Content**		
A2	03		RETURN RESULT	the message length is 3 bytes decimal
02	01	05	InvokeID	its Invoke ID is decimal 5
Hex dump = A2 03 02 01 05				

RETURN-ERROR DMS=====>HOST

Table 198 dv-MWT-Activate return error example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	0B		RETURN-ERROR	this message is 11 bytes long
02	01	05	InvokeID	the InvokeID is 5 decimal
02	01	01	Error value	the invoke message is missing a parameter
30	03		MissingParameter	MissingParameter is a sequence 3 bytes long
80	01	00	MissingParameter Type	the destDeviceID parameter is missing
Hex Dump = A3 0B 02 01 05 02 01 01 30 03 80 01 00				

4.16.2.2 dv-MWT-Activation parameters

A brief description of the parameters of the dv-MWT-Activation invoke follows:

destDeviceID identifies the destination party for MWI activation or deactivation.

VMSDeviceID identifies the voice mail port.

Indication specifies the MWI as either off (0) or on (1).

4.16.2.3 dv-MWT-Activation validations

Validation of message waiting activation/deactivation message

The switch performs a series of checks before sending the message waiting activation/deactivation message.

1 Verify the service version of the current host-switch session is SCAI13 or higher.

2 Verify the current host-switch session subscribes to the dv-MWT-Activate message.

3 For local DNs, verify the specified destDeviceID belongs to the customer group for the host ICM session. Customer group is not validated when using NMS.

4 Verify the specified destDeviceID has MWT capability and the CLASSP messaging class of service (MCOS).

5 If the destDeviceID is for a subscriber on a different switch, verify that NMS is available.

6 Verify the VMSDeviceID has the ECM line or customer group option.

7 Verify the VMSDeviceID is associated to the current host-switch session.

4.16.3 CompuCALL MDC/RES dv-DN-Query (operation value 27)

This section describes the new SCAI dv-DN-Query message the host sends to the switch requesting information about a residential or Centrex DN. This SCAI DN Query message maps to a ROSE Class 2 operation and is defined as dv-DN-Query. The dv-DN-Query message is only allowed on lines within the CompuCALL application environment. The host can select what information should be sent back in the DN query result: if the line is associated, what the line state is (idle, talking, ringing, etc.), and if a message is waiting.

ASN.1 Encoding:

```
dv-DN-QueryOPERATION
     ARGUMENTDNQueryArg
     RESULT   DNQueryResultArg
     ERRORSDNQueryErrors
::= 27

DNQueryArg::=SEQUENCE

{deviceID                [0]AddressType    OPTIONAL,
   DNQueryFilter         [1]DNQueryFilterType   OPTIONAL}

AddressType::= CHOICE
 {positionID[0]IMPLICIT  INTEGER,
   dialedDigits[1]IMPLICIT  IA5String,
   stationNumber [2]IMPLICIT  OCTET STRING,
   q931Address[3]Q931AddressType}


DNQueryFilterType::=BITSTRING

  {associationInfo, lineStateInfo, msgWaitInfo}

DNQueryResultArg::=  SEQUENCE
  {associationInfo[0]  IMPLICIT  AssociationQueryInfo OPTIONAL,
     lineStateInfo    [1]  IMPLICIT  CallStateQueryInfoOPTIONAL,
     msgWaitInfo    [2]  IMPLICIT  MessageWaitingQueryInfoOPTIONAL}

AssociationQueryInfo::=ENUMERATED
 {associatedToThisSession(0),
    associatedToAnotherSession(1),
    notAssociated(2),
    notAllowed(3)}

LineStateQueryInfo::ENUMERATED
 {   idle(0),
     origination(1),
     dialing(2),
     talking(3),
     ringing(4),
     held(5),
     treatment(6),
     lockedOut(7),
     maintenance(8)}
```

```
MessageWaitingQueryInfo    ::= ENUMERATED
  {MsgWaitingActivated(0),
     MsgWaitingDeactivated(1),
     ExecMsgWaitingActivated(2),
     ExecMsgWaitingDeactivated(3),
notAllowed(4)}

DNQueryErrors   ::= ENUMERATED
  {  notAllowed(0),
     unknownDN(1),
     invalidDN(2)
     invalidFilter(3),
     missingDN(4)}
```

4.16.3.1 dv-DN-Query: coded example

Following is a coded example of the dv-DN-Query-U message where the residential phone has a station number of (504) 858-1234. The host is requesting to query about the DN association information and line state information of (504) 858-1234. The invoke ID is 3.

INVOKE DMS <===== HOST

Table 199 dv-DN-Query coded example

Hex Dump			Information Element	Comments
Tag	Length	Content		
A1	19		INVOKE	the message length is 25 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	1B	Operation	its operation value is dv-DN-Query-U [27] decimal
30	11		Argument	DNQueryArg is of type sequence and is 17 bytes long decimal
A0	0C		deviceID	the deviceID is of type constructor and is 12 bytes long
82	0A	35 30 34 38 35 38 31 32 33 34	stationNumber	the DN of the phone is 504-858-1234
81	01	03	DNQueryFilter	DNQueryFilter is a bitstring requesting to query about the DN association and line state
Hex dump = A1 19 02 01 03 02 01 1B 30 11 A0 0C 82 0A 35 30 34 38 35 38 31 32 33 34 81 01 03				

4.16.3.2Examples of dv-DN-Query filter

Figure 57 shows all of the different combinations and permutations of the bit stream of the DN query filter, the hex value of the bit stream, and the meaning of the DN query filter. Bit 1 represents the association information, bit 2 represents the line state information, and bit 3 represents the message waiting information.

If bit 1 is set to 1, the host is querying DN association information. If bit 2 is set to 1, the host is querying line state information. If bit 3 is set to the 1, the host is querying message waiting information.

A return error with reason invalid filter is sent to the host if anything other than the following hex values are in the DN query filter: 01, 02, 03, 04, 05, 06, 07.

Figure 57 RETURN-ERROR DMS ====> HOST

Bits **HexDN Query Filter**

0	0	00	0	o	01	01	Association Info
0	0	00	0	o	11	03	Association Info, Line State Info
0	0	00	0	1	01	05	Association Info, MsgWaiting Info
0	0	00	0	1	11	07	Association Info, Line State Info, Msg Waiting Info
0	0	00	0	0	10	02	Line State Info
0	0	00	0	1	00	04	MsgWaiting Info
0	0	00	0	1	10	06	Line State Info, Msg Waiting Info
8	7	65	4	3	21		

Table 200 Return result for dv-DN-Query

Hex Dump			Information Element	Comments
Tag	Length	Content		
A2	0E		RETURN RESULT	the message length is 14 bytes decimal
02	01	03	InvokeID	its Invoke ID is 3
02	01	1B	Operation	its operation value is dv-DN-Query-U [27] decimal
30	06		Result Argument	DnQueryResultArg is a constructor and is 6 bytes long
80	01	00	DNAssociationInfo	this line is associated with this session
81	01	00	LineStateInfo	the state of the line is idle
Hex dump = A2 0E 02 01 03 02 01 1B 30 06 80 01 00 81 01 00				

Table 201 Return error for dv-DN-Query

Hex Dump			Information Element	Comments
Tag	Length	Content		
A3	06		RETURN ERROR	the message length is 06 bytes decimal
02	01	03	InvokeID	its invoke ID is 3
02	01	00	Error Value	The message is not allowed since it is not subscribed to
Hex dump = A3 06 02 01 03 02 01 00				

If the host queries the DN association info, the switch responds with one of the following parameters:

- associatedToThisSession: the DN is associated to the current session
- associatedToAnotherSession:the DN is associated to another session
- notAssociated: the DN is not associated at all
- notAllowed: the DN is not allowed to be associated with this session (for example, if the DN did not have the ECM option assigned to it or not in the same customer group)

If the host queries the current call state of the DN, the switch returns one of the following parameter values:

- idle
- originating
- dialing
- talking
- ringing
- treatment
- locked out--only sent if a 500/2500 set is left off hook
- maintenance

If the host queries the current state of the message waiting or executive message waiting indicator, the switch returns one of the following as a parameter value:

- MsgWaitingActivated: message waiting is activated
- Msg Waiting Deactivated: message waiting is deactivated
- ExecMsgWaitingActivated: executive message waiting is activated
- ExecMsgWaitingDeactivated: executive message waiting is deactivated
- notAllowed: message waiting or executive message waiting is not allowed

The host does not have to send the DN Associate message to query information about the DN.

4.16.3.3 dv-DN-Query parameters

A brief description of the parameters of the dv-DN-Query invoke follows:

Device ID is a mandatory parameter providing the address of the DN about which the host is querying information.

DN query filter is an optional parameter providing a filter so the host can select the optional information parameters to return in the DN query result. This filter doesn't guarantee subscription to each parameter. But it does guarantee that what is subscribed to may be filtered out on a per-message basis.

If the filter is not sent, all the optional information parameters will be sent in the DN query result. The following three parameters may be selected via this filter: DN association info, line state info, and message waiting info.

4.16.3.4 Validation of DN query message

Once the switch receives a dv-DN-Query, several validations are performed. The switch will verify the following:

1 Only station number of the device ID is accepted.

2 The switch-host session subscribes to the DN-Query message.

3 The address of the DN is specified within the same customer group as the SCAI session that the host is connected.

4 The address is a valid residential DN or Centrex DN.

5 At least one of the ECM sub-options is subscribed to at the line level.

6 The service version is at least SCAI07.

4.16.3.5 DN query return-result

If the validation of the DN-Query message is valid, then the switch will send a Return Result service with information concerning the line. The host can select to query one or more of the optional parameters. One or more of the following parameters is provided by the Return Result:

DN association info is one of the following: associated to this session, associated to another session, not associated, or not allowed.

Line state info is one of the following: idle, origination, dialing, talking, ringing, treatment, held, or lockedOut.

Message waiting info is one of the following: message waiting activated, message waiting deactivated, executive message waiting activated, executive message waiting deactivated, or not allowed.

4.16.3.6DN query return error

If the dv-DN-Query does not pass validation then the switch sends a Return Error service message. A Return Error with one of the following parameters is sent to the host:

DN query errorssequence of parameters:

1 not allowed is sent to the host if the message has not been subscribed to; and if any of the ECM sub-options have not been subscribed to

2 unknown DN is sent to the host if the DN specified is unknown to the switch

3 invalid DN is sent to the host if the specified DN does not belong to the associatedCentrex Customer group

4 invalid Filter is sent to the host if the specified filter is encoded incorrectly

5 missing DN is sent to the host if the DN is not included in the invoke request message.

The SCAI message waiting indication message will only be supported for lines that have at least one ECM option.

4.17 CompuCALL MDC/RES TAPI Extensions

ICM uses a dv-Call-Progress-U message to comply with Telephony Applications Programming Interface (TAPI) Class 5 event messaging. In NA12, the DMS-100 switch sends a dv-Call-Progress-U message to the customer's host computer to report on a call's progress. The message reports on originating call states. A subset of the call states can apply to either ACD or MDC/RES lines. The list that follows contains the MDC/RES dv-Call-Progress-U call notification states sent from the switch:

- digit collection
- busy
- ringback
- connected

The dv-Appl-Logon message contains a parameter entitled Service Version, which uniquely identifies the version of messages sent during a session. The dv-Call-Progress-U messages require Service Version SCAI14, or higher, for delivery from the switch to the host.

Refer to the ACD TAPI Extensions section for dv_Call_Progress_U details including the ANS.1 definition and coded examples.

Interactions, restrictions and limitations

4.17.1 Message interactions

4.17.1.1 Interaction between ACD dv-DN-Associate and ACD dv-Call-Received-C

The host should not disassociate an ACD DN when there are outstanding dv-Call-Received-C Invokes for that group. If it does, any subsequent response to the outstanding dv-Call-Received-C message for that ACD DN will default to the original route as if no call redirection was allowed for that originating ACD DN.

4.17.1.2 Interaction between ACD dv-Call-Redirect & ACD dv-Call-Received-C

As described in and Section 4.6.2, the dv-Call-Redirect is a linked operation of dv-Call-Received-C.

4.17.1.3 dv-Call-Redirect, dv-Route-Call, and dv-Give-Trtmnt interactions

dv-Call-Redirect, dv-Route-Call, and dv-Give-Trtmnt are host-initiated messages. For SCAI 12 and above using the CCS7 OPC as the unique NetNodeID requires the customer to fill-up the NetNodeID for host-initiated messages.

4.17.1.4 Relationship between ACD dv-Call-Received-C and ACD dv-Call-Queued-U

If the host established a session using a service ID whose service profile contains the dv-Call-Queued-U and dv-Call-Received-C messages and if an ACD group with the call redirection capability active is DN associated with that session, dv-Call-Received-C and not dv-Call-Queued-U will be sent by DMS-100 to the host for ACD calls to that ACD group. In other words, based on the messages in the service profile, either dv-Call-Received-C or dv-Call-Queued-U is sent to the host when a call is received by a given ACD group (depending on if it has the call redirection capability), but not both.

4.17.1.5 Interaction between ACD third party messages and ACD dv-Call-Released-U

An ACD group does not need to be DN associated for an agent in that group to use dv-Drop-Party. However, as a result of this message, the host will not receive a dv-Call-Released-U unless the ACD group to which the agent belongs is DN associated to a session and the service profile for that session contains the dv-Call-Released-U message.

A call transfer request from the agent via a dv-Transfer-Party message also requires the equivalent MDC capability (call transfer capability) active for that agent for the transfer request to be successful. If the MDC call transfer capability is not available or active for the agent who is requesting call transfer by sending dv-Transfer-Party, the consult party (the third party) will be released and the controller (the first party) will be ringing for the held party (the second party). In addition, no dv-Call-Released-U message will be sent unless the ACD group of the agent is also DN associated to a session and the service profile for that session contains the dv-Call-Released-U message. In the above case when the agent is in a conference call, whether the third party has answered or not, a dv-Transfer-Party request creates the same consequence as above (the third party will be dropped). No dv-Call-Released-U will be sent unless the ACD group of the agent is also DN associated to a session and the service profile for that session contains dv-Call-Released-U. It is therefore recommended as a good practice for the host that when the CompuCALL call party messages are used in a session, the agent's ACD group be DN associated to the same session if at all possible.

4.17.1.6 Interaction between ACD dv-Add-Party and ACD dv-Conference-Party

When a dv-Add-Party message is initiated, the host can specify AddConsultParty or AddConsultforConf in the OperationType. Both operation types will allow a third party to be added as a consult call and can potentially be conferenced in later when dv-Conference-Party is sent. The difference exists in when the conference resources are requested to be reserved. If AddConsultforConf is specified in the dv-Add-Party message, conference resources are reserved immediately (if available). If AddConsultParty is used, the switch does not reserve any conference resources until dv-Conference-Party is received and, if unavailable at that time, dv-Conference-Party will fail.

4.17.1.7 Interactions between ACD dv-Call-Released-U and other ACD Meridian CompuCALL messages

- If a host receives a dv-Call-Released-U message from the switch with ReleaseReason "Call Cleared" or "Call Abandoned", it should consider that no more messages relating to the same call (with the same Network CallID) will be sent by the switch and take the appropriate action (close the call record); the agent may still be doing follow-up work on the call so the data session may be active until cleared by the agent.
- If the host receives a dv-Call-Released-U message from the switch with ReleaseReason indicating "Call Transferred", "CallOverflowed", "CallRedirected", "CallForwarded", "CallPickedUp", "PartyDropped" or "PartyDroppedNoAnswer", it should still treat the call as being active (i.e., a subsequent event message with the same NetworkCallID may be received.) It falls entirely on the host to define a time-out after which the host can close the call record.
- A dv-Call-Released-U message may be sent before a call is established, e.g., after dv-Call-Offered-U but before dv-Call-Answered-U if the agent does not answer (i.e., picked up by another MDC station after dv-Call-Offered-U has been sent to the host but before call is answered, or after the

call has been offered to the first MDC station, the call is forwarded by the Call Forward Don't Answer feature).

- dv-Call-Released-U is not sent unless it has been preceded by dv-Call-Received-C, dv-Call-Queued-U, dv-Call-Offered-U, dv-Call-Answered-U or dv-Make-Call, provided the ACD group for which the messages are sent belongs to an associated environment and the service profile for that session contains the messages. The only exception is the case where an MDC CompuCALL station initiates a call using the voice set, then only dv-Call-Released-U is sent without any preceding CompuCALL messages for that call.
- For Call Redirection, the dv-Call-Released-U for the first group will be sent before the dv-Call-Received-C or dv-Call-Queued-U for the second group.
- If an overflow from the queue (overflow of queued calls or networked ACD time delay overflow), the dv-Call-Released-U for the first group is sent after the dv-Call-Queued-U but before the dv-Call-Offered-U of the second group. This sequence applies even if the call ends up being handled by the original group. Since call redirection is not allowed for these calls, a dv-Call-Received-C message will never be sent.
- If call forward and call pick-up, dv-Call-Released-U is sent for the station to which the call is initially offered before dv-Call-Offered-U or dv-Call-Answered-U is sent for the party who receives the forwarded call or picks up the call.

4.17.1.8 Interactions between ACD dv-Call-Answered-U and other ACD Meridian CompuCALL messages

dv-Call-Answered-U is always preceded by a dv-Call-Offered-U, dv-Call-Queued-U or dv-Call-Received-C message to the same host except with call pick-up. When station A picks up the call offered to MDC station B, only dv-Call-Answered-U is sent for A when A answers the call, and the only dv-Call-Offered-U message has been sent for station B before the call is picked-up.

Stations A and B must be in the same customer group in the same switch for call pick-up to be valid.

4.17.1.9 Interactions between ACD dv-Set-Feature (Ready/Not Ready function) and ACD agent event reporting messages

A toggle between the Not Ready and Ready states, manually or via the CompuCALL dv-Set-Feature (Ready/Not Ready function), while the agent is in an active call, will not result in any event messages reported to the host. However, once the agent has released the call, any agent state change will be reported to the host with the appropriate event message.

4.17.1.10 Interactions between ACD dv-Set-Feature (Reserv Agt) and legacy NACD reserve agent

If an NACD reserves an agent, the dv-Set-Feature (Reserv Agt) can not reserve the same agent. The result is a Return Error. If a dv-Set-Feature (Reserv Agt) reserves an agent, then NACD can not reserve the same agent. The result is a Return-Error.

4.17.1.11Interactions between Reserved agent time-out and legacy NACD time-out

When the legacy NACD software reserves an agent, the switch does not inform the host of the unreservation of an agent once the timer expires. When the dv-Set-Feature message reserves an agent, the switch informs the host of the unreservation of an agent once the timer expires. This is done through the dv-Agent-SetAction-U message.

4.17.1.12 Interactions between ACD Agent Event Reporting messages and ACD Variable Wrap-up feature

The ACD feature Variable Wrap-up (VARWRAP) places an ACD agent who has just released a call temporarily in the Not Ready state. The corresponding Not Ready and Ready events are not reported to the host with event messages.

4.17.1.13 ACD DN-Query on 500/2500 sets

When the dv-DN-Query message is sent from the host to the switch to request information about a Centrex line with ACD on a 500/2500 set, a Return Error of not allowed is sent to the host.

4.17.1.14 Answer call interaction

The user may only answer incoming calls via the dv-Answer-Call message, with the exception of allowing a user to send a dv-Answer-Call message to answer a dv-Make-Call alerting.

4.17.1.15 Make call interaction

During the alerting stage of a dv-Make-Call request, the user may send a dv-Answer-Call message in response to the alerting of the line. This will set up the processing of the dv-Make-Call so it can proceed to actually setting up the outbound call.

4.17.1.16 Call waiting interaction with answer call

During an incoming call offered via Call Waiting, the user may select to answer the call by sending the dv-Answer-Call message. This will connect the user to the incoming waited call and put the first call on hold, just as would be expected when a user flashes to accept a waiting call. In this scenario, 500/2500 sets with the call waiting option can answer the call waited call by sending the dv-Answer-Call message.

4.17.1.17 Hold call interaction with types of hold

The dv-Hold-Call message is a CompuCALL based hold. This hold will not be interchangeable with the following types of hold: MBS hold, permanent hold (HLD), call hold (CHD), Autohold, and Kset music on hold (KSMOH).

4.17.1.18 CompuCALL 3WC/CXR and 3WC call events

The 3WC call events (Call-Consult-Originated-U, Call-Conferenced-U, and Call-Transferred-U) will be sent if the 3WC/CXR operations are done by CompuCALL using dv-Add-Party, dv-Conference-Party, and dv-Transfer-Party operations.

4.17.1.19 ACD position ID and 3WC/CXR event messages

The controller DN and the consult DN parameters will be in position Id format when the 3WC call event messages (Call-Consult-Originated-U, Cal-Conferenced-U, and Call-Transferred-U) are sent on behalf of the ACD DN.

4.17.1.20 Calls through tandem switches

The information is available at the target switch, even though the BCS software release on the tandem switch is less than NA010. The NICM information is in the network transport parameter, which is ignored by intermediate switches. The chance for rejection decreases because of the interaction. The same applies to a tandem switch that is not a DMS switch.

4.17.1.21 ACD feature interactions

This section specifies the interaction of the Meridian ACD CompuCALL service-related messages with various DMS-100 Meridian ACD features which result in ACD calls being transferred by an ACD agent using either set features or CompuCALL messages via the host computer, overflowed, or redirected before being answered by an agent.

4.17.1.22 Introduction

This section provides information on the main Meridian ACD specific features as a background to understanding the feature interactions in the subsequent sections.

- Basic ACD queueing
  - — An ACD group may be defined to have one primary ACD DN and up to 16 supplementary DNs to which it receives calls. These DNs are valid directory numbers in the customer's numbering plan. Each DN is associated with a call priority which ranges from zero (the highest) to three (the lowest). The priority of a call determines the call-handling order.

- ACD incoming call queue
  - — Incoming calls which cannot be served immediately are placed in the incoming call queue in order of arrival and priority determined by the

ACD primary or supplementary DN dialed and whether calls are received from a trunk or a line on the switch. As agents become available, calls are removed from the incoming call queue in order of arrival priority, where higher priority calls are served first. Lower priority calls may wait a long time. The ACD queueing feature, priority promotion, can increase a call's priority when a wait threshold is reached.

- ACD call transfer queue
  - The ACD call transfer queue is a maximum of 42 queue slots available to an ACD group for queueing calls which are transferred from one agent to another (using the ACD call transfer to incalls feature) and for calls which are redirected to a specific agent (see Redirection below). There is no priority associated with this queue. When the agents become available, they are immediately selected to take the next call queued for them.
- ACD overflow out queue
  - Calls entering the overflow out queue are offered to the network for call handling on a first-come, first-served basis. These calls would have entered these queues when the customer-defined thresholds are exceeded.
- ACD transfer
  - ACD agents can use the MDC call transfer capability or send a dv-Transfer-Party message via the host to transfer a call with or without consult party answering and with or without conferencing to:
    - a primary or supplementary ACD DN the same as or different from the agent's
    - a specific agent, using the agent's secondary DN
    - a specific agent, using the DN associated with the agent's ACD line (normally not directly accessible); this uses the ACD Transfer to In-Calls Key feature
    - a non-ACD line (e.g., a Centrex line)
    - an ACD or non-ACD DN on another switch

  Note: That in any of the above cases, the agent to whom the call is transferred may be a supervisor.
- ACD overflow

  Note: The following features result in ACD calls being overflowed before being answered by an agent.

- Automatic overflow (threshold overflow) - overflows incoming calls for a given ACD group to any destination specified by the customer (normally another ACD group, but can also be to a treatment, e.g., busy, ringing, announcement - or to a non-ACD DN), if the number of calls in the ACD group's incoming call queue equals a customer-specified maximum, or if the first call in the incoming call queue has waited longer than the customer-specified maximum waiting time.
- Enhanced Overflow - when the incoming call queue for a given ACD group has reached a maximum threshold (specified by the customer), this feature looks ahead, in a customer-specified sequential order at each of up to four other ACD groups within the same switch. An incoming call is overflowed to the first ACD group that is able to accept it, i.e., to route it to an available agent or put it in queue; if all are unavailable, the call is overflowed to the overflow destination specified by the customer for the Threshold Overflow feature (above).
- Immediate overflow- This isn’t a feature. The term indicates the type of overflow where a call to an ACD group is overflowed before it is queued for that group.
- Overflow of Queued Calls - this feature enhances the Enhanced Overflow feature by providing for overflowing of calls that have been queued for an excessive amount of time, as specified by the customer; in the case of a call which is overflowed to another ACD group in the same switch, the call is logically queued for both groups until an agent in one of the groups is available to handle the call.
- Ring Threshold - when an ACD call is offered to an agent and no agents are available in the group, the call is queued in the overflow out queue and given ring treatment. Or the call is placed at the front of the incoming call queue of the ACD group if all agent are busy. If the call is answered by the agent before the time expires, the timer is cancelled by ICCM. If the call is not answered and the caller is not abandoned when the time expires, the agent is logged out or put into not ready by ICCM.
- or is placed at the front of the incoming call queue of the same ACD group if all agents are busy.
- Not Ready - if the agent presses the Not-Ready key while an ACD call is being offered, the call is offered to the agent in the same ACD group who has been idle longest, or is placed at the front of the incoming call queue of the same ACD group if all agents are busy.
- Night Service or Forced Night Service - overflows all incoming calls for a given ACD group to any destination specified by the customer (typically an announcement, an ACD group at another location, or a Night Service DN).

  - When the ACD group is considered to be in Night Service (calls in the incoming call queue may be re-queued for another ACD group with Forced Night Service).
  - Supervisor Control of Night Service - overflows all incoming calls for a given ACD group to the Night Service destination, prior to the onset of Night Service (to prevent calls from being in the incoming call queue when the ACD group goes into Night Service and therefore from remaining unanswered).
  - Controlled Interflow (CIF) - overflows all incoming calls for a given ACD group to any destination specified by the customer when the supervisor activates the CIF feature. The CIF feature can be activated at any time by the supervisor. (Calls already in the incoming call queue are presented to the agents as normal).
  - Status Reroute - an ACD group in this state and is not servicing or queuing calls. New calls are rerouted to customer-specified overflow route. The calls currently in the Incoming call queue for this ACD group, prior to the group entering into this state, remain in queue until they are either answered, or treated for exceeding the customer-specified threshold conditions. An ACD group would enter the status reroute state due to an equipment fault that effects the agents' telephone lines of this ACD group.
- ACD Redirection
  - The Meridian ACD CompuCALL allows a host to redirect an ACD call to any specific ACD lines within the same customer group on the same switch, or another ACD group or non-ACD lines on any switch.

4.17.1.23 ACD feature interactions with CompuCALL call event messages

An ACD call which is not a CompuCALL Call remains outside the CompuCALL environment if it is transferred, overflowed, or redirected to:

- a primary or supplementary ACD DN on the same switch not belonging to a CompuCALL environment
- a secondary DN of a specific ACD agent/supervisor on the same switch not belonging to a CompuCALL environment
- any specific ACD line (i.e., Incalls Key) or non-ACD line (e.g., a Centrex line) on the same switch not belonging to a CompuCALL environment
- any ACD group or non-ACD line on a different switch

If an ACD CompuCALL call is conferenced with, or transferred, overflowed, or redirected by any of the features to a primary or supplementary ACD DN or MDC DN belonging to a CompuCALL environment on the same switch, the static call-related information (e.g., NetworkCallID, OrigCallingNumber, OrigInboundDN) is retained and the dynamic call-related information (e.g., ACDDN, CallType) is updated. If the destination party is in an associated environment, the appropriate messages (based on the service profile) containing the static and dynamic call-related information is sent to the host.

If a call is transferred, forwarded, or overflowed before entering the CompuCALL environment, the normal messages associated with a new incoming call are sent. Note, however, that the OrigCallingNumber parameter will be:

- In the case of a transfer, the primary ACD DN of the ACD group to which the transferring agent belongs or the DN of the worker's line (except consoles).
- In the case of a time-delay overflow (e.g., from the incoming call queue) or immediate overflow, the normal OrigCallingNumber would be sent. In the case of a call forwarded into the CompuCALL environment (even if the call has been forwarded more than once outside the environment), the DN of the original calling party; or if an intermediate transfer occurs outside the CompuCALL environment, the OrigCallingNumber parameter becomes this transferring party (except consoles).

The static call-related information in dv-Call-Queued-U, dv-Call-Received-C, dv-Call-Offered-U and dv-Call-Answered-U messages for a transferred, overflowed, or conferenced ACD CompuCALL call allows the host to, for example:

- Determine the original ACD DN for a transferred, overflowed, or redirected call. For example, this is more likely to indicate the service desired by the caller in the case of a call overflowed to another ACD group.
- Save processing and worker time, if the first dv-Call-Queued-U or dv-Call-Received-C and/or dv-Call-Offered-U and/or dv-Call-Answered-U message was used to retrieve caller-related information (e.g., an insurance policy).
- Transfer the existing call record (if one is created for each call) to the second worker, rather than creating a new one.
- Provide a shared screen capability in the case where the two agents are involved with the call at the same time (i.e., transfer with consultation or conferencing).

When calls are transferred, overflowed or redirected between two different host applications in the same CompuCALL environment, the host applications must coordinate NetworkNodeID and ApplicationID values.

If a host was involved with the CompuCALL Call before it was transferred, overflowed, redirected, or conferenced, and if the second host can communicate with the first host, the NetworkCallID and PrevApplicationID parameters can be used to coordinate the transfer of call-related information from the first host to the second. However, if the second host cannot communicate with the first host application, or in the case where no host has been involved with the call (immediate overflow), the CallType, OrigInboundDN and HostCallData parameters provide the second host with additional call-related information.

4.17.1.24 ACD call events call type feature interactions with 3WC

If a consult call, the consult party (if associated to a session) will receive the dv-Call-Offered-U with CallType transferred to indicate that the call is a consult call in the context of the call transfer or conference feature, i.e., it could be transferred or conferenced as opposed to has been transferred or conferenced.

4.17.1.25 ACD feature interactions with dv-Call-Received-C message

The dv-Call-Received-C message will be sent before the call is determined whether to be routed to Night Service, overflowed (immediate overflow), queued in the incoming call queue or offered to an agent.

Call redirection (although subscribed for the ACD group) is not allowed for:

1 calls overflowed back into the same group (i.e., by ring threshold, not ready or make set busy)

2 calls which overflow to another group after being queued and which may be logically queued in both groups (i.e., time delay overflow or overflow from queue)

3 the third consecutive time (i.e., calls which have been redirected two subsequent times without intermittent transfer, overflow, or forward)

4 the same DN subsequently

For these calls, the dv-Call-Received-C will not be sent to the host. Instead, Call-Queued-U message will be sent if the customer subscribes to that message and if the call is queued.

4.17.1.26 ACD feature interactions with dv-LOB-Event-U message

This section describes the interaction of the LOB feature with other features. After activating and before completing the LOB feature, an agent can assist another feature:

- If you press the LOB key and the agent presses the LOB key again, the LOB feature restarts.

- If the agent presses the RELEASE key, the digit collection and call ends.
- If the agent presses the ACDNR key, the digit collection and call end, and the agent is in the not ready state.
- If the agent presses the MSB key, the agent's set becomes busy, but the digit collection continues.
- If the agent presses the EMK key, LOB feature aborts and the EMK feature starts.
- If the agent presses another DN or invokes a DN-like feature, the LOB digit collection ends, the INCALLS call is on hold, and the agent connects to the other line and either answers or receives a Dalton.

The agent will ignore all other feature keys.

4.17.1.27 ACD feature interactions with the dv-Call-Released-U message

The dv-Call-Released-U message is intended to facilitate CVD for transferred, overflowed, redirected, or conferenced calls:

- dv-Call-Released-U ReleaseReason Call Overflowed covers ring threshold, not ready and make set busy cases.

  Note 1: In the case of a consult or conference call, a dv-Call-Released-U message with ReleaseReason indicating "PartyDropped" is sent when the original calling party or consult party drops out or is released. In the case that the consult party is dropped or released, this message sent for the consult party will have the same NetworkCallID as that of the active call between the original call party and the controller. More messages relating to this active call with the same NetworkCallID can be sent.

- When the controller in ACD group A is in a two-way call with original call party and consults with consult party in ACD group B, and

1 if the controller drops the consult call before the consult party answers, a dv-Call-Released-U will be sent for the consult party with ReleaseReason "PartyDroppedNoAnswer" (see Figure 58)

2 if the consult party drops out or is released after answering, a dv-Call-Released-U will be sent for the consult party with ReleaseReason "PartyDropped" (see Figure Figure 59)

3 if the controller completes the transfer and hangs up (i.e., blind transfer) and the original call party hangs up before the consult party answers, i.e., while ringing, a dv-Call-Released-U message will be sent for the consult party with ReleaseReason "CallAbandoned" (see Figure 59)

4 if the original calling party drops out or is released after the consult party answers, a dv-Call-Released-U will be sent for the original call party with ReleaseReason "PartyDropped" (see Figure Figure 61)

Note 2: The ReleaseReason is independent of whether the affecting party is or is not a CompuCALL party and it is also independent of whether the voice set or the computer was used.

4.17.1.28 ACD feature interactions with dv-EMK-U message

The Emergency Key Event Message feature follows all the interactions that the ACD Emergency Key feature does. The only features that the ACD Emergency Key feature interacts with are:

1 Make Set Busy (MSB) feature

a. Activating MSB while active on an emergency call leaves the current call undisturbed and places the agent position in the "Make Set Busy" state.

b. If the Supervisor's position is in "Make Set Busy", ignore this and the supervisor receives the call.

2 ACD Not Ready (ACDNR) feature

a. The ACD Agent activates the ACD NotReady key and the ACD Non-Immediate Cutoff feature is not datafilled. This releases the call immediately and the EMK lamp turns off. The entire call is down.

b. If the ACD Agent activates the ACD NotReady key, but the ACD Non-Immediate Cutoff feature is active, the agent releases the call when the agent has finished with the call.

Figure 58 Drop Consult (before answering)

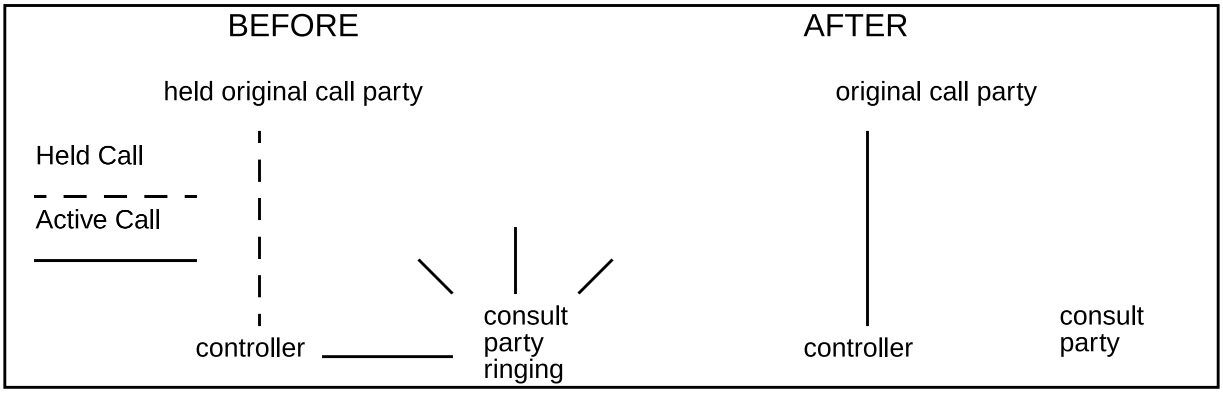


Figure 59 Drop Consult (after answering)

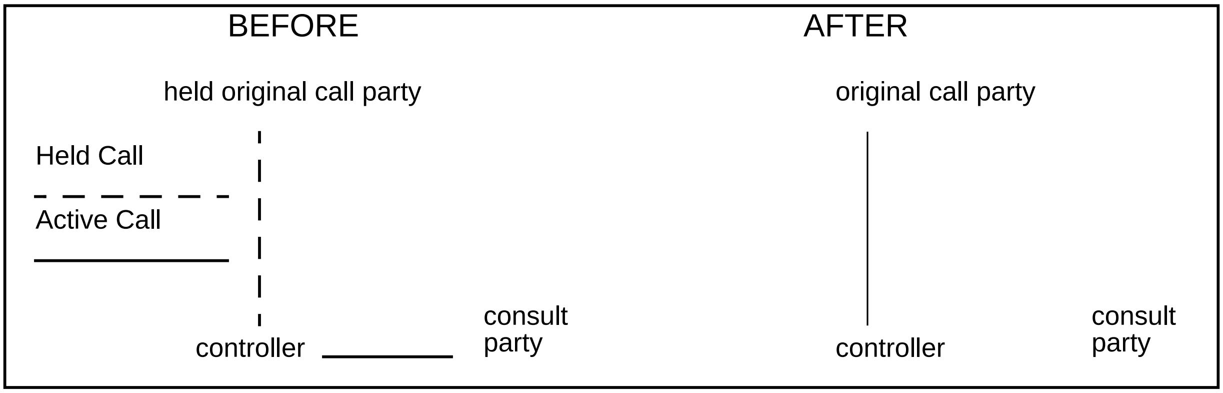


Figure 60 Blind Transfer (before answering)

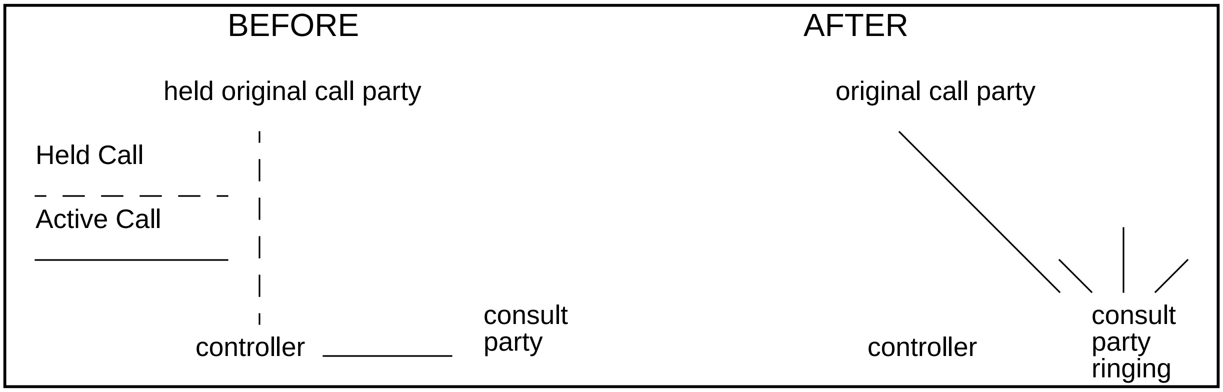
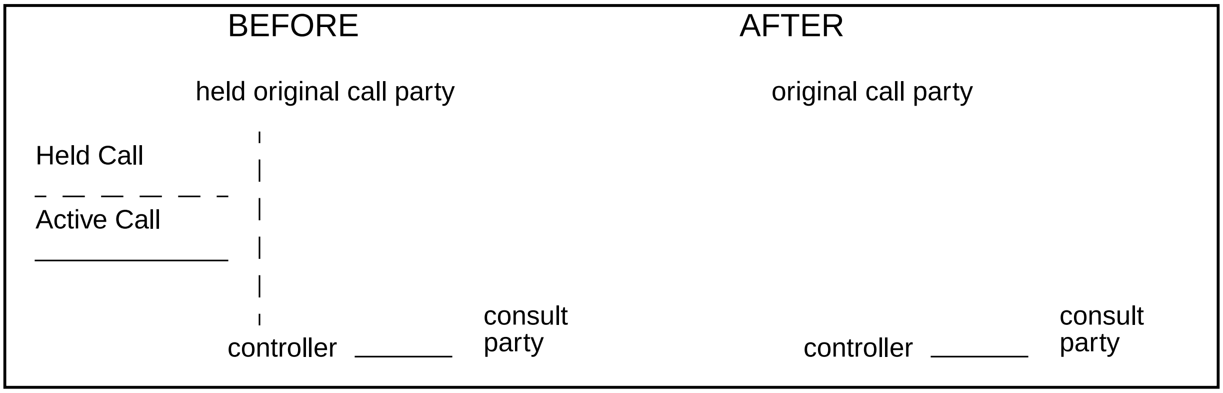


Figure 61 Consult (after answering)



4.17.1.29 ACD Interaction of 3WC with call party messages

A host can use the dv-Add-Party, dv-Transfer-Party, dv-Drop-Party and dv-Conference-Party messages only if the corresponding agent or the Customer Group the agent belongs to has been assigned the MDC Three-Way Call (3WC) or Call Transfer (CXR or FXR) features on DMS-100.

4.17.1.30 ACD MakeCall with distinctive ringing

Distinctive ringing is provided for Meridian ACD CompuCALL customers so allow agents can differentiate normal inbound calls from outbound calls initiated via the CompuCALL interface. Customers can use distinctive ringing on their sets for inbound and/or outbound calls. Distinctive ringing does not apply when call forcing is in effect. In this case, or when the agent wishes to receive only outbound calls, MakeCallType, "CallingAgentReserved" may be used which essentially "reserves" the agent for that outbound call. Alternatively, "CallingAgentNotReserved" may be used which provides more flexibility to the agent to accept both inbound and outbound calls.

4.17.1.31 ACD interaction with dv-Resource-Query message

The GrpStat parameter indicates the ACD groups status from a nodal ACD standpoint, not Network ACD (NACD) standpoint should it belong to such a configuration. For example, the NACD group may indicate a status of overflow when the maximum queue length and maximum wait time thresholds have been exceeded. However, the GrpStat may indicate the ACD group to be Accepting Calls since agents could be handling calls. The QInfo parameter does not contain information on the ACD group's overflow in queues as these counts would have already been included in the QInfo of the original ACD group which would have queued the calls.

In the event of a network failure where NACD information was not received by the corresponding ACD nodes, spurious results may occur with the resource status information on an ACD group such as GrpStat providing outdated information.

4.17.1.32 ACD Rerouting/overflow interaction with dv-Resource-Query message

Due to the complexity of the DMS ACD system, it possible to observe spurious results in response to a dv-Resource-Query message. For example, it is possible for the thresholds (e.g., maximum wait time) to be repeatedly exceeded upon the following circumstances:

- The maximum queue size has been reached for the incoming call queue.
- The call at the head of the incoming call queue exceeds the maximum wait time.
- The call at the head of the incoming call queue has been offered to an agent, but is not answered.
- The call ringing on an agent gets re-unequaled at the head of the incoming call queue.

4.17.1.33 ACD overflow interaction with CompuCALL call event messages

An ACD call in a CompuCALL environment which overflows to the same or other ACD group also in a CompuCALL environment, maintains the value of the ACD DN parameter in subsequent CompuCALL messages after the overflow. The messages which include the ACD DN parameter include, dv-Call-Queued, dv-Call-Offered, dv-Call-Answered, and dv-Call-Released.

4.17.1.34 ACD call on hold interaction with CompuCALL dv-Set-Feature (Not Ready function) message

The ACD Not Ready Agent feature is not allowed for agents holding calls on an MBS position InCalls key. The CompuCALL Not Ready function removes this restriction. Hence, ACD agent positions can’t activate the agent Not Ready feature using the CompuCALL dv-Set-Feature (Not Ready function) message while a call is on hold.

4.17.1.35 ACD call with attendant console interaction with CompuCALL dv-Set-Feature (Not Ready function) message

The ACD Not Ready agent feature is not allowed for agents connected to a DMS Attendant Console. The CompuCALL Not Ready function removes this restriction. Hence, ACD agent positions can’t activate the agent Not Ready feature using the CompuCALL dv-Set-Feature (Not Ready function) message while connected to an attendant console.

4.17.1.36 ACD call transfer/call park recalls interaction with CompuCALL dv-Set-Feature (Not Ready function) message

When an ACD agent receives a recall of a transferred or parked call, the result of an Agent Not Ready feature activation using the CompuCALL dv-Set-Feature (Not Ready function) message is consistent with that of the manual activation. The Not Ready agent request is not allowed. If the switch receives a dv-Set-Feature (Not Ready function) message in this scenario, a Return Error (with value Invalid-AgtPos-State) will be returned to the host.

4.17.1.37 ACD (MBS) Display and CompuCALL dv-Set-Feature message

The display of MBS sets will not be updated for the activation of Agent LogIn (i.e., the LogIn ID + password) or the Agent Not Ready (i.e., walkaway reason code) feature activations if initiated using the CompuCALL dv-Set-Feature message, as is the case for manual activation.

4.17.1.38 ACD VARWRAP and CompuCALL dv-Make-Call message

When the switch receives a dv-Make-Call message on behalf of an agent position in a variable wraps (variable) state, it sends a Return Error (mismatched state) to the host. MakeCall was designed not to kick off variable wraps when an ACD call ends. This is extended to release guard timing since it is an extension of variable wraps. Thus, release guard time and variable wraps do not exist for CompuCALL MakeCall calls.

4.17.1.39 ACD call forcing and dv-Answer-Call message

The Answer message will not be allowed if call forcing is active on the call.

4.17.1.40 ACD agents and dv-Release-Call message

Since 500/2500 sets have only one line for both ACD and non-ACD calls, checking will be in effect so that ACD calls, when the address to be released is specified as an ACD agent position ID, can be released via dv-Release-Call and non-ACD calls that are Centrex or residential can be released.

4.17.1.41 ACD features and dv-Release-Call message

These features are the ACD features that interact with dv-Release-Call, if a feature is not mentioned dv-Release-Call will provide the same functionality as manually releasing.

- ACD Observe Agent (OBS)
  - If an ACD supervisor is on an ACD call or a dv-Make-Call initiated call and presses the OBS key, the original Incalls Key call is put into an autohold state. If the supervisor manually released, the OBS call would be released and the supervisor could hit the Incalls Key to retrieve the original call. In the same scenario, dv-Release-Call releases the held Incalls Key and does not affect the OBS call.
  - ACD Call Agent (CAG)
  - The CAG key will act just like OBS and any other key that activates autohold. While a supervisor is active on a CAG call, a manual release releases the CAG call. In the same scenario, a dv-Release-Call message releases the original call on the Incalls Key.

- ACD Emergency Feature (EMK and AEMK)
  - If an ACD supervisor is on an ACD call and another agent presses the EMK and the supervisor answers the AEMK, the supervisor's original call will be put on autohold. If the supervisor sends a dv-Release-Call message the supervisor's original call will be released and the AEMK call will not be affected.
  - If an ACD agent is on an ACD call and presses the EMK key, an immediate conference is formed when the supervisor answers the AEMK. A dv-Release-Call message in this scenario will be treated the same as when the agent is the controller of a 3WC. The call will not be affected and a Return Error of InvalidCallState will be returned to the host.
- ACD Call Supervisor (CLSUP) and Answer Agent (AAK)
  - The ACD agent CLSUP key and ACD supervisor AAK both have the effect of autoloading a dv-Make-Call or incoming ACD call. Therefore, Incalls Key calls that are on autohold as a result of activating the AAK or CLSUP keys are releasable using dv-Release-Call. A manual release would release the CLSUP or AAK call.
- ACD Forced Agent Availability (FAA)
  - If an ACD supervisor set is active on either an ACD call or a dv-Make-Call, and the supervisor hits the FAA key to force an agent to the Ready state, the Incalls Key call will be put on autohold. Therefore a dv-Release-Call message releases the held Incalls Key call and does not affect the FAA feature. A manual release would abort the digit collection and release the FAA key.
- ACD Login Activate
  - If an ACD agent is in the middle of logging in on either an MBS or 2500 set, a manual release would abort the login attempt and the agent would remain not logged in. A dv-Release-Call sent in the middle of logging in does not abort the login attempt, the agent remains in the same state and can continue logging in.
- ACD Not Ready
  - If a dv-Release-Call is sent to an ACD agent. The agent will not go into the Not Ready state.

4.17.1.42 ACD forcing and CompuCALL Call origination from incalls (ACD DN) key.

If ACD Forcing is datafilled for the ACD group or the agent position, the dv-Make-Call invoke with any valid MakeCallType parameter, on behalf of the agent position from the Incalls Key will automatically answer the alerting and the tone or buzz will be based on the ACD Forcing datafill. Therefore ACD Forcing takes precedence over the MakeCallType choice.

4.17.1.43 ACDXFER and 3WC call event messages

The 3WC call event messages (Call-Consult-Originated-U and Call-Conferenced-U) will be sent to the ACD agent if the call is transferred to Incalls DN using ACDXFER call and only if the DN of the ACD group of the ACD agent is associated in a CompuCALL environment.

4.17.1.44 ACD and 3WC/CXR call event messages

If the intra-switch consult leg call is to an ACD group, the dv-Call-Conferenced-U message for the consulted DN will be sent only when the call is offered to an agent. The dv-Call-Conferenced-U message for the consulted DN(i.e., the ACD group) will not be sent when the call is queued, but will be sent for the controller DN. If the intra-switched conferenced call is queued to an ACD group, the dv-Call-Conferenced-U message of the controller DN will contain the ConsultDN in the dialed digits format and not the station number format.

4.17.1.45 Sending dv-Set-Feature from the host

Activation of ACDNR can take place from the host using the dv-Set-Feature message. If this occurs, the call continues until a party releases the call. Also use of the ACDNR key without nonimmediate cutoff option causes the call to continue. The postcall status remains in not ready status because the messages dv-Call-Released-U and dv-Agent-Not-Ready-U occur in succession.

The agent can be logged out from the host using the dv-Set-Feature message. The message timing is still the same. As a result, the postcall status is set to logged out for the dv-Call-Released-U message.

4.17.1.46 Use of secondary DN

If the agent’s set has a secondary DN key datafilled and the group does not have the not ready on SDN option, then pressing the secondary DN during a call has no effect on the call other than putting it on hold. The agent can then choose who to talk to by alternating hits to the incalls key and SDN.

If the agent’s set has a secondary DN key datafilled and the group does have not ready on SDN option, then pressing the secondary DN during a call causes the agent to go into not ready once the call to the agent is released. The use of a secondary DN sends a succession of dv-Call-Released-U and dv-Agent-Not-Ready-U messages. Also the use of a secondary DN requires the postcall status set to not ready. The agent is made ready again by pressing ACDNR, then the switch informs the host by using a dv-Agent-Ready-U message.

If the SDN has the feature Call Supervisor, the not ready on SDN option is null for that agent.

The agent can depress a secondary DN key during the variable wrap interval. If the the group does not have not ready on SDN option then variable wrap proceeds normally. When the interval expires, the agent can receive calls on the ACDDN. If this happens while the agent is still on the secondary DN, the phone has a distinct ring. If the group has the not ready on SDN option, pressing the SDN key ends the interval. The host receives a dv-Agent-Not-Ready-U message. This is the same as the host pressing ACDNR.

4.17.1.47 Use of ACD call park

If the ACD group has option ACD Call Park (ACDCPK), and the agent has a key datafilled with ACDCPK, pressing the key during a call releases the call on the agent's side. The switch sends a dv-Call-Released-U message to the host and the variable wrap begins. For the host, this situation is identical to the agent hitting the release key. As a result, a postcall status set to variable wrap is necessary.

4.17.1.48 Heavy display traffic

During busy call traffic, if the variable wrap time is datafilled to zero, the time is set to one second. This ensures that the call forcing tone the agent receives is not missed. The postcall status is set to release guard started.

4.17.2 MDC feature interactions

This section specifies the interaction of the MDC CompuCALL service-related messages with various DMS-100 MDC features which result in MDC calls being transferred by an MDC station, or forwarded to or picked-up by another station if a call is not answered by the MDC station to which call is offered.

4.17.2.1 Introduction

This section provides information on MDC specific features as a background to understanding the feature interactions in the subsequent sections.

- MDC Transfer
  - — MDC stations can use the MDC call transfer capability to transfer a call, with or without consult and with or without conferencing to any DN (including ACD and non-ACD) on the same or different switch.
- MDC Forward
  - — An MDC CompuCALL call can be forwarded to any DN on the same or different switch by the MDC Call Forward feature which includes the following types of forwarding:
    - – Call Forward Don't Answer (CFD) - Calls are forwarded upon exceeding a customer-specified ringing time limit when offered to the called station. The destination where the calls are forwarded is specified by the user of the DN receiving the call.

  - Call Forward Busy (CFB) - Calls are forwarded when offered to the called station for which a connection cannot be made. e.g., talking state, make-set-busy state, equipment failure. The destination where the calls are forwarded is specified by the user of the DN receiving the call.
  - Call Forward Universal/Intergroup/Fixed (CFU/CFI/CFF) - Calls are forwarded unconditionally. The CFU is a programmable option which forwards all calls unconditionally. CFI is a programmable option which forwards all intergroup calls unconditionally. CFF is a predefined option which forwards all calls unconditionally. The destination where the calls are forwarded is specified by the user of the DN forwarding the call.

- MDC Pick-Up
  - When an MDC CompuCALL call is offered to a station, any other station in the same customer group and call pick-up group on the same switch can initiate call pick-up.
- MDC Call-Waiting
  - When an MDC CompuCALL call is offered to a station which is busy on an existing call, it will be offered as a waited call.
- MDC DN Suppression Levels
  - A DN associated with a line can be suppressed for display or network availability by one or more of the following controls:
  - (Central) Office Level Suppression - lets a CO suppress presentation of DNs assigned within the office on a per-office basis. This is defined by the Toledo and overrides all other suppression indicators.
  - Group Level Suppression - lets blocks of DNs be suppressed e.g., Centrex lines. This is defined by the Toledo and their customers and overrides other suppression indicators below.
  - DN Level Suppression - lets a subscriber have a default DN suppression status on a per-line basis. This is defined by the Toledo, their customers and subscribers.
  - Call Level Suppression - lets subscriber control DN suppression status on a per-call basis. This is defined by the Toledo as a subscriber option and overrides the DN and group levels of suppression. This level of suppression is not applicable to ForwardingParty DNs since the forwarding stations cannot initiate this feature.

4.17.2.2 MDC feature interactions with dv-Call-Offered-U and dv-Call-Answered-U messages

An MDC call which is not a CompuCALL call remains outside the CompuCALL environment if it is transferred or forwarded (i.e., Call Forward Don't Answer) to:

- Any DN in the same customer group on the same switch if it does not belong to a CompuCALL environment.
- Any DN on a different switch.

The MDC CompuCALL stations can use the transfer, forward, and pick-up MDC features. The only call forward capability for which CompuCALL messages are sent for the forwarding party is Call Forward Don't Answer (CFD).

In the cases of call forward universal (CFU) or call forward busy (CFB), the call is forwarded without being offered to the forwarding party (see notes below) and therefore no dv-Call-Offered-U message is sent for the forwarding party. Unless the call was previously forwarded by CFD, transferred, overflowed, or redirected, no CallHistoryInfo with CallType "CallForwarded" information is provided.

In NA11, the IC Message Waiting Activation/Deactivation feature sends the forwarding party parameter in dv-Call-Offered-U and dv-Call-Answered-U messages to all CFD, CFU and CFB type calls.

"Forwarding party" refers to the party who forwards an incoming call to another party. "Forwarded party" is the party whose call is forwarded by the forwarding party to the "forwarded-to party", i.e., the "forwarded-to party" is the party to whom the call is forwarded.

If an MDC CompuCALL call is transferred or forwarded to a primary or Supplementary ACDDN or to another MDC station belonging to a CompuCALL environment on the same switch, the static call-related information (NetworkCallID, OrigCallingNumber, OrigInboundDN) is retained and the dynamic call-related information (DeviceID, CallType) is updated. If the destination party is in an associated environment, the appropriate messages (based on the service profile) containing the static and dynamic call-related information is sent to the host.

The static call-related information in dv-Call-Offered-U and dv-Call-Answered-U messages for a transferred or forwarded MDC CompuCALL call allows the host to, for example:

- Determine the original inbound DN for a transferred or forwarded MDC call; for example, this is more likely to indicate the service desired by the caller in the case of a call forwarded to another station when the station to which the call is offered does not answer.

- Save processing and personnel time, if the first dv-Call-Offered-U and/or dv-Call-Answered-U message was used to retrieve caller-related information (e.g., an insurance policy)
- Transfer the existing call record (if one is created for each call) to the second station, rather than creating a new one.
- Provide a shared screen capability in the case where the two stations are involved with the call at the same time (i.e., transfer with consultation or conferencing).

  Note: When calls are transferred or forwarded between two CompuCALL environments, the customers must coordinate NetworkNodeID and ApplicationID values.

If a host were involved with an MDC CompuCALL call before it was transferred or forwarded and if the second host can communicate with the first host, then the NetworkCallID and PrevApplicationID parameters can be used to coordinate the transfer of call-related information from the first host to the second. However, if the second host cannot communicate with the first host application, or in the case where no host has been involved with the call, then the CallType and OrigInboundDN provide the second host with additional call-related information.

If an MDC CompuCALL call arrives at a MDC CompuCALL party and is call waited, the CallMode indicates "CallWaited."

The Teen Service feature allows members of the same household to have separate phone numbers. A primary DN and up to six secondary DNs (SDN) can be assigned to a single line. Different ringing patterns identify the number called. If a call is terminating on the SDN, then the destination device ID parameter of the call offered message will contain the SDN. In addition, if the SDN answers the call, the destination Device ID parameter of the call answered message will contain the SDN. Also, if the SDN releases the call, the destination device ID parameter of the call release message will contain the SDN.

In three party scenarios, the SDN will not be delivered in the call history parameters. Enhanced SDN (ESDN) allows the members of the same household to originate a call from the SDN and is not currently supported.

Three-way call interaction with Teen Service:

- If a call terminates on the SDN that has 3WC, if the 3WC controller adds a second leg to the call, the SDN is delivered in the device ID parameter of the call offered message. In the call history parameter of the call offered message, the primary DN will be delivered.

- Similarly, the SDN is delivered in the device ID parameter of the call answered message and the primary DN is delivered in the call history parameter of the call answered message.

Call Forward Don't Answer Interaction with Teen Service:

- When a call terminates on a teenservice SDN whose type is P, if the primary DN has CFDA or CFBL option and later is forwarded to another DN which is also ECM associated, the device ID parameter of the call offered and call answered message will contain the SDN.
- The call history parameter of the call offered and call answered message will show the primary DN and appropriate reason. CompuCALL messages are consistent with display functionality.

Call Waiting Interaction with Teen Service:

An incoming SDN call to a busy line with the CWT feature should receive audible ring. The line with CWT should receive a Teen Service CWT tone that is repeated only once, 10 seconds later if the incoming call has not been acknowledged. The call mode in the call answered and call offered message will be call wait on Teen Service.

4.17.2.3 MDC feature interactions with the dv-Call-Released-U message

The dv-Call-Released-U message facilitates CVD for transferred calls:

- If an MDC CompuCALL Call arrives at a MDC CompuCALL party and is picked up by an MDC Party in the same customer group, the ReleaseReason in the dv-Call-Released-U for the MDC CompuCALL party will indicate "CallPickedUp."
- If a call is forwarded by an MDC CompuCALL party, the ReleaseReason in the dv-Call-Released-U for the MDC CompuCALL party will indicate "CallForwarded". Note that the dv-Call-Released-U message in this case is only sent for Call Forward Don't Answer (CFD).
- If an MDC CompuCALL party participates in a 3WC/call transfer, depending on whether the participant is the controller, original call party or consult party, the following ReleaseReason in the dv-Call-Released-U are sent.
- When the controller in ACD group A is in a two-way call with the original calling party and consults with consult (third) party in ACD group B
  - — If the controller drops the consult call before the consult party answers, a dv-Call-Released-U will be sent for the consult party with ReleaseReason "PartyDroppedNoAnswer.”

- If the consult party drops out or is released after answering, a dv-Call-Released-U will be sent for the consult party with ReleaseReason "PartyDropped.”
- If the controller completes the transfer and hangs up (i.e., blind transfer), and the original call party hangs up before the consult party answers, i.e., while ringing, a dv-Call-Released-U message will be sent for the consult party with ReleaseReason "CallAbandoned.”
- If the original calling party drops out or is released after the consult party answers, a dv-Call-Released-U will be sent for the original call party with ReleaseReason "PartyDropped.”

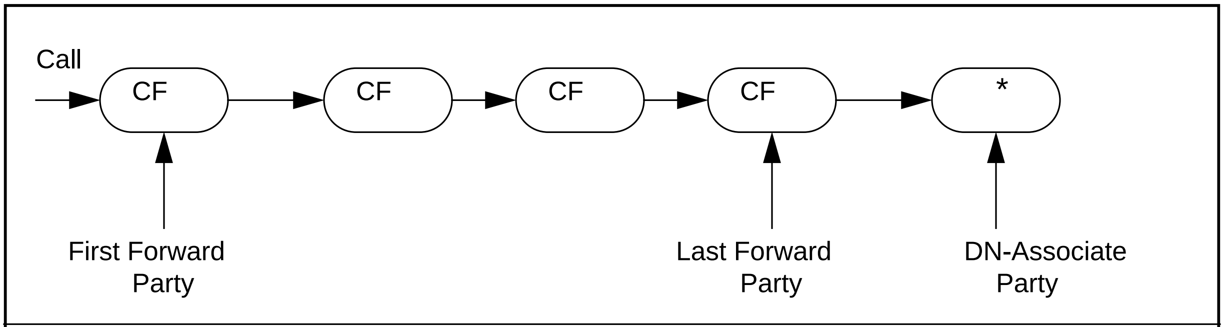
4.17.2.4 MDC call forward interactions with CompuCALL messages

The Call History information in the dv-Call-Queued-U, dv-Call-Offered-U, dv-Call-Answered-U and dv-Call-Received-C messages is lost for calls that are forwarded to another switch (i.e., over SS7/PRI network) and only consist of the information pertaining to the call after the forwarding took place. The forwarding party parameters of these messages are provided for calls that are forwarded prior to forwarding to another switch as well as any subsequent forwards (multi-forward situation).

4.17.2.5 MDC call transfer interactions with CompuCALL messages

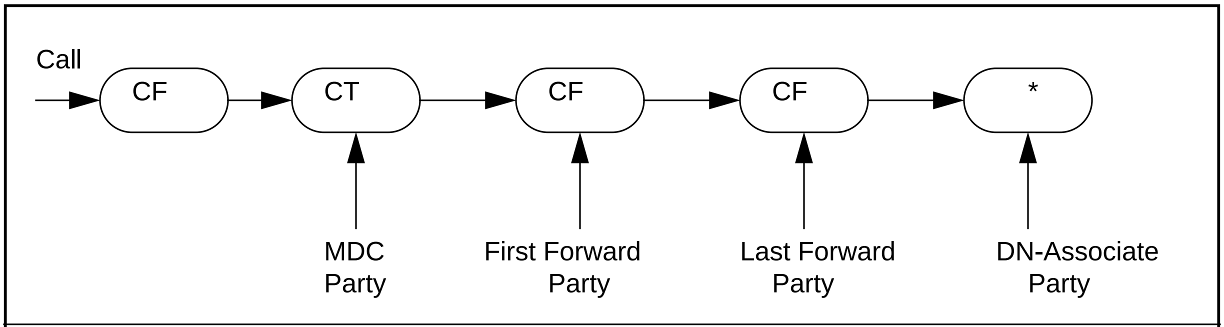
If a call transfer by an MDC line occurs in a chain of call events (Figure 63), the first forwarding party information in Figure 62 is replaced by the first forwarding party after the call transfer event occurs. If the call transfer occurs after the last forwarding party (Figure 64), no forwarding party information is provided.

Figure 62 Chain of call forward events



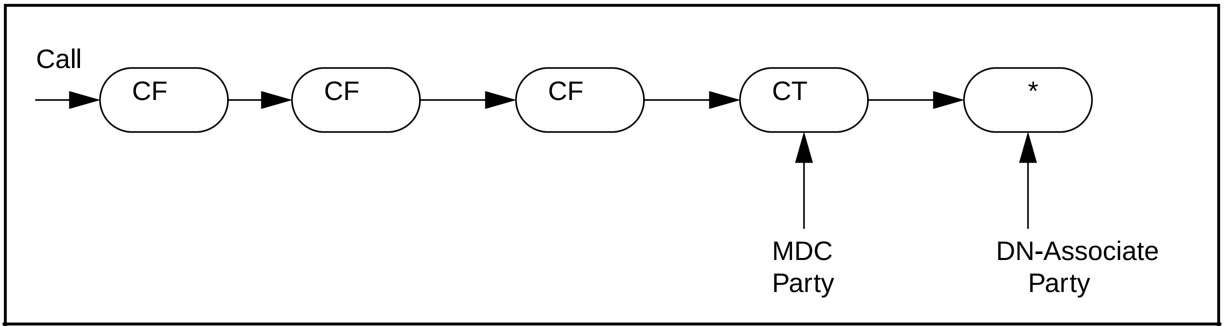
Note: CF = Call Forward

Figure 63 Chain of call forward events after transfer by MDC party



Note: CF = Call Forward, CT = Call Transfer

Figure 64 Chain of call forward events before transfer by MDC party



Note: CF = Call Forward, CT =Call Transfer

4.17.2.6 MDC call forward & DN suppression interactions with CompuCALL messages

In the same switch, forwarding party DN availability is determined by DN suppression levels subscribed by the forwarding party. If interswitch call forwarding, forwarding party DN availability is determined by associated presentation indicators.

4.17.2.7 MDC call forward and CompuCALL

The forwarding number and forwarding reason (in the following ROs: dv-Call-Queued-U, dv-Call-Offered-U, dv-Call-Answered-U and dv-Call-Received-C) are not sent to the host if the forwarding number is not provided by the network even though the forwarding reason is provided.

However, if the forwarding number is provided but not the forwarding reason, the forwarding number is set with the appropriate DN, original called number or forwarding number. The forwarding reason is set to "unknown".

4.17.2.8 MDC 3WC chaining limits

The same 3WC chaining limits which apply to set generated 3WCs or ACD SCAI generated 3WCs will apply to Centrex or residential SCAI generated 3WCs.

4.17.2.9 Private number/caller

Calling name delivery blocking (CNAB) and calling number delivery blocking (CNDB) standards are adhered to. For instance, in the call offered message if the terminating line has CNAB then the destination device ID will not deliver the name.

4.17.2.10 Call log

Feature call log provides a switch-based incoming callers list (ICL) to CLASSPLUS customers. This feature is accessed by dialing an activation code and then viewed using softkeys on a Sesame customer premise equipment (CPE). This service can log the caller's name and/or number, time and date, the number of times the caller has called, whether the call was unanswered, forwarded, or busy, and whether this call has been viewed by the subscriber.

In some scenarios, the Call Log feature requires a off hook signal which forces a Set Off-Hook event message to be generated. Along with the corresponding call released message. The following is a list of scenarios that generate a set off-hook event message and call released for call log:

- When deleting the call log feature via SERVORD with the message waiting notification type of CLASS Message Waiting (CMWI), a set off-hook message and a call released messages are generated on the CompuCALL link. The released reason is call cleared.
- When changing the message waiting notification type of the Call Log feature from CMWI to stutter dial tone (STD) via SERVORD, a set off-hook message and a call released message are generated on the CompuCALL link. The released reason is call cleared.

Make a call to a set with Call Log feature that has the message waiting notification type of CMWT and do not answer the call. Before viewing the ICL a set off-hook message and call released message are generated on the

CompuCALL link. After the user views the ICL, 2 sets of set off-hook/ call releases messages are sent. One released reason is call abandoned and the other released reason is call cleared.

4.17.2.11 MDC call screen

The Selective Call Rejection (SCRJ) feature allows a subscriber to reject calls arriving from a list of previously identified DNs. If a caller terminates on an associated residential line that is in the caller's SCRJ list, no call event messages will be sent to the host concerning the associated residential line. The Selective Call Acceptance (SCA) feature allows a subscriber to accept calls arriving from a list of previously defined DNs. If a caller terminates on an associated residential line that is not in the caller's SCA list, then no call event messages will be sent to the host concerning the associated residential or Centrex line.

4.17.2.12 Residential Call Hold (RCHD)

Residential Call Hold (RCHD) will allow a residential line to place a call on hold. No event messages will be generated when the call is placed on hold or when the call is re-established.

4.17.2.13 Suspended Service (SUS)

The Suspended Service (SUS) feature allows all service to be denied to a single line and multi-line telephone sets with SUS assigned. The subscriber is blocked from originating or receiving any calls. If a residential or Centrex line has the SUS option then the host will not receive any call event messages concerning the suspended line.

4.17.2.14 Message waiting event and message waiting options

The SCAI message waiting indication message will only be supported for lines that have the message waiting or executive message waiting options assigned to the line.

4.17.2.15 DN query message and options

The host is only allowed to query lines in the same customer groups that have at least one ECM option.

4.17.2.16 3WC and answer call interactions

dv-Answer-Call is available in a limited capacity to users in multiple call scenarios initiated by 3WC, CXR, ACDXFER, QCK, and CompuCALL initiated 3WC. Many different scenarios and interactions occur with different conference features as well as different sets. These interactions are beyond the scope of the initial dv-Answer-Call offering.

For simplification, the 3WC abbreviations of the first party being party B, the controller being party A, and the third (consult) party being C are used.

A user is considered a controller from the point of flashing or hitting the 3WC key until the call collapses back to a two party call. A user is the controller in both consult and conference scenarios.

The following restrictions apply to users in multiple party scenarios:

- If party B is the dv-Answer-Call user, then the user will be allowed to Answer using dv-Answer-Call.
- If the user is party A in the call (controller), then that party will not be allowed to answer in either consult or conference scenarios using dv-Answer-Call, and a Return Error of invalidCallState will be returned to the host.
- If the user is Party C, then that party will be allowed to answer using dv-Answer-Call.

The above restrictions apply not only to the 3WC feature, but all multiple party features including CXR, ACDXFER, QCK, and CompuCALL dv-Add-Party and dv-Conference-Party features.

4.17.2.17 Teen Service and answer call interactions

The Teen Service feature allows members of the same household to have separate phone numbers. A primary DN and up to six secondary DNs (SDN) can be assigned to a single line. Different ringing patterns identify the number called.

The Teen Service feature is only allowed on 500/2500 sets of residential and Centrex lines. Since the dv-Answer-Call message is not supported on 500/2500 sets, the answer message will not be allowed if the answering party specifies the SDN of a teen service call.

4.17.2.18 Call pickup and answer call interactions

Call pickup enables subscriber A to answer a call on subscriber B’s line from subscriber’s A’s set. The dv-Answer-Call message will not answer another DNs call in the call pickup group.

4.17.2.19 Ring again and answer call interactions

Ring again allows a calling party encountering a busy station to be notified when the busy station becomes idle and to be placed automatically in a ring again mode. The Ring Again feature is assigned to a key on MBS set. The Ring Again key can not be answered by dv-Answer-Call.

4.17.2.20 Emergency service (ESL and EBS) and hold call interactions

Once an ESL/ESB party has answered a call, the party can not hold the call that is made to the ESL. The line remains under the control of the ESL, and the call ends only when the ESL line releases. A dv-Hold-Call request is rejected with a Return Error of Call-Not-Holdable-By-User and the call is not affected.

4.17.2.21 Teen Service and hold call interactions

An SDN involved in an active call can not be placed on hold by the dv-Hold-Call message.

4.17.2.22 Call pickup and hold call interactions

A call pickup party involved in an active call can be placed on hold by the dv-Hold-Call message.

4.17.2.23 3WC and hold call interactions

When a dv-Hold-Call messages is sent to a 500/2500 set that is the controller of a 3WC call, the hold message is not processed and the call cannot be held. A return error is received with invalid call state.

4.17.2.24 Emergency service (ESL and ESB) and release call interactions

Once an ESL/ESB party has answered a call, the call is not releasable by the party that made the call to the ESL. The line remains under the control of the ESL, and the call ends only when the ESL line releases. A dv-Release-Call request is rejected with a Return Error of Call-Not-Releasable-By-User and the call is not affected.

4.17.2.25 Call waiting and release call interactions

If call waiting is active on the user’s DN at the time the dv-Release-Call message is being processed, then the currently active talking call will be released by CompuCALL, and the user will be reconnected/connected to the inactive call.

4.17.2.26 Held call and release call interactions

Currently there are a number of procedures to place a caller on hold. First, on a MBS set there is a hold key which will hold a call locally at the set. There is also a Call Hold (Access Code CHD) feature which places a party on hold and gives the holder dial tone which allows the user to initiate a new call. Finally, there exists a Permanent Hold (Access Code HLD) feature which simply holds the other party, but does not allow the holder to initiate another call.

- MBS hold
  - If a call is held by using the HOLD key on an MBS set, the party that was put on hold can manually release from the call, but the user that activated hold (holder) cannot manually release from the call. A dv-Release-Call allows either party to release.
- Permanent hold (HLD)
  - If a call is held by a party using the HLD feature, the held party can manually release from the call, but the holder cannot manually release before flashing to reconnect the call. dv-Release-Call allows a user to release from a call held by the HLD feature providing the other party activated the HLD feature. If the user activates the HLD feature, that

user will not be allowed to release using dv-Release-Call and a Return Error of Invalid-Call-State will be returned.

  - Call hold (CHD)
  - Similar to Three-Way Calling (3WC), Call hold allows the party that activates the CHD feature to make another call after holding the first party. For this reason, CHD scenarios will be treated the same as other multiple call scenarios.

- Local hold and Autohold
  - Local hold refers to holding a call using an MBS hold key. The hold key is not a programmable key, it is a special key that holds a call at the set the key was activated from rather than at the switch. Autohold, which results in the same treatment to the held party and the same local method of holding the call, refers to a party accessing another DN key while active on a different DN key.
  - When an Incalls Key call is held either by activating local hold or autohold, the call cannot be released manually until the agent presses the Incalls Key to activate the key again. If an agent that is active on an ACD call activates local hold, and then attempts to release the call manually, the attempt will be denied.
  - If local hold or autohold is active then the party can release the call via dv-Release-Call.
- Kset Music on hold (KSMOH)
  - KSMOH is a feature that provides music or an announcement to a party that is put on hold. If a party activates MBS hold or accesses another DN key which invokes autohold, the held party will be connected to music or an announcement. dv-Release-Call will act exactly the same with KSMOH active as it does with MBS hold and autohold. A dv-Release-Call allows either party to release.

4.17.2.27 3WC restrictions and release call

dv-Release-Call is available in a limited capacity to users in multiple call scenarios initiated by 3WC, CXR, ACDXFER, QCK, and CompuCALL initiated 3WC. Many different scenarios and interactions occur with different conference features as well as different sets. These interactions are beyond the scope of the initial dv-Release-Call offering.

For simplification, the 3WC abbreviations of the first party being party B, the controller being party A, and the third (consult) party being C are used.

A party is considered a controller from the point of flashing or hitting the 3WC key until the call collapses back to a two party call. A party is the controller in both consult and conference scenarios.

The following restrictions applies in multiple party scenarios:

- If party B is the dv-Release-Call user, then the user will be allowed to release using dv-Release-Call.
- If the user is party A in the call (controller), then a release request with the value of releaseType either releaseThisParty or releaseAllParties while in consult mode will release the consult leg call. Alternatively, a release request during conference will be handled according to the value of the releaseType (releaseThisParty or releaseAllParties), the difference being that one type may cause the call to be transferred, while the other type will cause all three parties to be released from the conference call.
- If the user is Party C, then that party will be allowed to release using dv-Release-Call.

The following describe different 3WC scenarios with dv-Release-Call:

- Originator calls terminator. Terminator answers and presses 3WC key and dial tone heard. Originator is held and DN lamp flashes. dv-Release-Call sent to the switch on behalf of the terminator. Return result sent to host. Originator is still held and DN flashes.
- Originator calls terminator. Terminator answers and presses 3WC key and dial tone heard. Originator is held and DN lamp flashes. dv-Release-Call sent to the switch on behalf of the originator. Return result sent to host. The first leg is cleared and the terminator still receives dial tone but the call is moved to the first DN key.

  Note: The above restrictions apply not only to the 3WC feature, but all multiple party features including CXR, ACDXFER, QCK, and CompuCALL dv-Add-Party and dv-Conference-Party features. As mentioned previously, Call hold (CHD) is also treated like 3WC.

4.17.2.28 Make call and release call

An outgoing call initiated by a dv-Make-Call request that is receiving information such as announcements from the switch will be releasable using dv-Release-Call.

The new capability of aborting an unconfirmed dv-Make-Call will be an available function with the dv-Release-Call.

The dv-Release-Call will provide the user with the capability to abort a dv-Make-Call that is in the alerting stage. If the user invokes a dv-Make-Call command and enters the alerting state, then decides the call is no longer desired, the party can abort the dv-Make-Call using dv-Release-Call.

Since every Class 2 operation requires a response, a response must be sent for not only the dv-Release-Call message, but also the dv-Make-Call message that was aborted. A Return Result will be returned for the dv-Release-Call message if completed successfully. A Return Error will be sent for the dv-Make-Call invoke since the command was not completed. The error reason for the dv-Make-Call will be Make-Call-Aborted. The new dv-Make-Call error will be sent in all cases where the dv-Make-Call operation was terminated during the alerting stage. This includes aborting using dv-Release-Call, the ACD agent going Not-Ready, and other actions that cause the set to become unavailable during the alerting stage.

4.17.2.29dv-Make-Call Errors

Make-Call-Aborted -an invocation of the dv-Make-Call has been aborted at the alerting for confirmation stage, this error is sent regardless of the reason the dv-Make-Call was aborted. (i.e., dv-Release-Call, ACD agent going Not-Ready, etc.).

4.17.2.30 Teen Service and release call

A SDN can not be released by the dv-Release-Call message.

4.17.2.31 Call pickup and release call

The Call Pickup party can be released by the dv-Release-Call message.

4.17.2.32 Attendant console and release call

An agent cannot release from an Attendant Console once the call has been answered using the dv-Release-Call message. Before the attendant console answers the call the agent can still abandon the call using dv-Release-Call.

4.17.2.33 Teen Service SDN and Call-Conferenced-U message

In the case where a Teen Service SDN is the consult DN of a 3WC/CXR, the consult DN in the dv-Call-Conferenced-U message will be that of the Teen Service primary DN(PDN) and not a SDN.

The consult DN in the dv-Call-Consult-Originated-U message will be provided in the dialed digits format, the dialed digits will contain the Teen Service SDN.

4.17.2.34 Calling name delivery

The 3WC call events (Call-Consult-Originated,-U, Call-Conferenced-U, and Call-Transferred-U) will not provide the calling name and hence Calling Name Delivery or Calling Name Delivery Blocking feature will not have any impact.

4.17.2.35 Key set music on hold (KSMOH)

Using the dv-Hold-Call and dv-Add-Party messages on sets with key set music on hold option and appropriate datafill, will provide music to the caller.

4.17.2.36 Feature access code

If the 3WC/CXR controller hits the 3WC/CXR key and dials a feature access code, the dv-Call-Consult-Originated-U message will not be sent to the switch.

4.17.2.37 Call forward universal (CFU)/ call forward don’t answer (CFDA)

When the controller activates 3WC/CXR to initiate a consult leg call with party C, and party C has activated CFU or CFDA to party D, the display on party D shows “Call Fwd”. The dv-Call-Transferred-U message will be sent from the switch to party D when the call is transferred even though the display on party D shows “Call Fwd” and not “Call Transfer”.

4.17.3 Set interactions

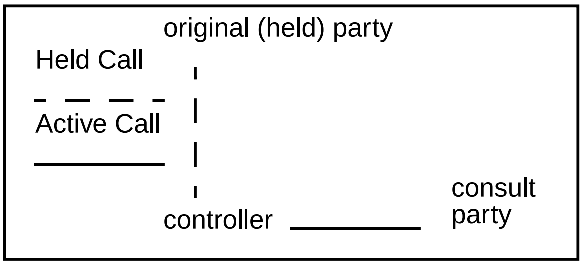
4.17.3.1 Interactions with dv-Make-Call

- The dv-Make-Call request requires the agent set to be idle (logged in and logically on-hook), with or without the ACDNR feature activated as specified in the MakeCallType subparameter. However, by assigning call forcing to an ACD set, it is possible to send dv-Make-Call and originate the call automatically while the set is physically off hook, provided that the line is logically in the idle state (Note: call forcing does not apply to 500/2500 sets).
- Customers can optionally choose distinctive ringing on their sets for inbound and/or outbound calls. The distinctive ringing does not apply when Call Forcing is in effect. In the case where the agent only makes outbound calls, the host can specify "CallingAgentReserved" in the dv-Make-Call request which essentially "reserves" the agent for that outbound call. The agent must correspondingly activate the ACDNR feature on the set. If the host specifies "CallingAgentNotReserved" in the dv-Make-Call request, then the agent can receive either inbound or outbound calls announced by their distinctive rings. In that case, the agent must not activate the ACDNR feature. In order for the dv-Make-Call request to proceed, the agent set must be in the proper state as specified in the dv-Make-Call request.
- If the agent changes the set state, i.e., the ACDNR feature is activated or deactivated or the set goes logically off-hook, after the switch has verified the dv-Make-Call request, the call will proceed as if the set state did not change. For example, if the agent deactivates the ACDNR feature after the switch has verified dv-Make-Call with MakeCallType: "CallingAgentReserved", the dv-Make-Call request will still proceed.
- A dv-Make-Call request while activating the Walkaway feature would result in a Return Error with reason "Make-Call-Mismatched-State".
- The display with respect to MakeCall originated calls will on the originator's set, a message will be displayed showing that the call is an outgoing call to the destination DN,
- The display with respect to MakeCall originated calls will on the terminator's set, the lamps and display will be updated as if the originator had called the destination directly.

4.17.3.2 Interactions with dv-Add-Party, dv-Transfer-Party, dv-Drop-Party and dv-Conference-Party

dv-Add-Party: An ACD agent (controller) is active on a call with an original call party, the host computer sends a dv-Add-Party (AddConsultParty or AddConsultforConf) message to add a consult party to the call. The connection between the controller and the original call party will be put on hold and the controller will be connected to the consult party (Figure 65).

Figure 65 Consult call



In this add party scenario if all the parties are MBS sets:

- There will be no display or lamp changes to the original call party set.
- There will be no changes to the controller's key lamps (the 3WC key will not light up and the DN key lamp will stay on). The controller's display will be updated with the consult party's call info (name and number).
- The consult party's display and lamp will be updated as if the controller had called the consult party directly.

Display set updates:

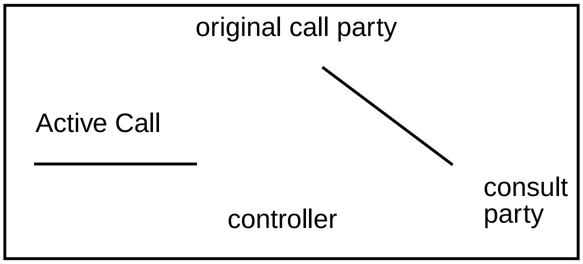
- The display with respect to set generated 3WC's will not be updated for SCAI generated 3WC's. That is the calling number, name, and reason display will be the same as expected, but if a lamp exists on the SCAI controller's set for the 3WC feature, it will not be updated.
- The original held party of the add party will see no display or lamp changes, while the consult party's display will be updated as if the SCAI controller had called the consult party directly.
- Whenever the consult or conference calls are collapsed the re-connected parties display will be updated reflecting who they are actually connected to. If the call was transferred, then the call transfer reason will be displayed on the parties' set involved in the transferred call. No reason is displayed when a drop party request is completed; only the number and name as appropriate will be displayed.
- If the user has a data terminal connected to the host computer, then they should rely on the host to keep them informed of the progress of their CompuCALL call control requests.

- On residential 2500 type display sets the reason is not displayed. On IBN/PSETS the display is updated accordingly.

Event interactions:

- If the original calling party releases the call (depressing the Release key on the MBS set or going on-hook on 500/2500 sets), the connection between the controller and the original calling party will be released. The controller will stay connected to the consult party. At this point, any dv-Transfer-Party or dv-Drop-Party messages will be denied with a Return Error message.
- If the consult party releases the call, the connection between the controller and the consult party will be released and the controller will be re-connected to the original call party. The controller's display will be updated with the original call party's call info. At this point any dv-Transfer-Party or dv-Drop-Party will be denied with a Return Error message.
- If the controller releases the call (goes on-hook, or depresses the RLS key), the consult party will be dropped and the original call party will ringback the controller. There would not be CompuCALL messages sent for the ringback.
- Any CONF key depression (e.g., 3WC on MBS sets) or flash (on 500/2500 sets) from the controller set will be ignored.
- Any other key depression (e.g., hold key) will proceed normally.
- dv-Transfer-Party: Given the above add party scenario the host computer may send a dv-Transfer-Party. In this case, the original call party will be connected to the consult party and the controller will drop out from the call, see Figure 66.

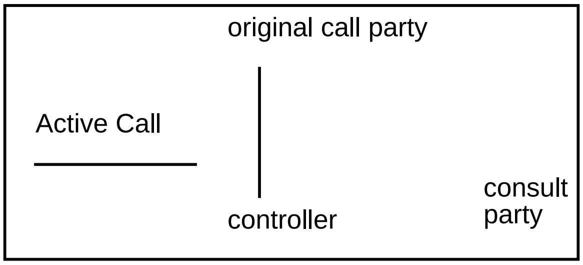
Figure 66 Transferred call



In this scenario, the original calling party's and the consult party's displays will be updated with the call info of the other party's on the top line and call transfer reason on the bottom line.

dv-Drop-Party: Given the add party scenario, the host computer may send dv-Drop-Party; the consult party will be dropped and the controller will be re-connected to the original call party, see Figure 67.

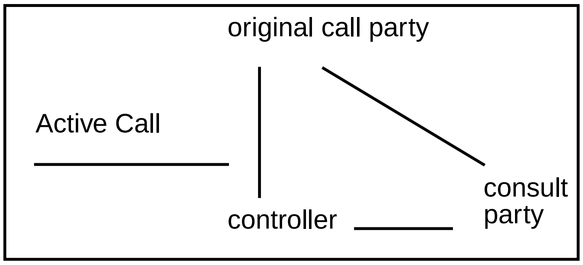
Figure 67 Dropped consult party



In this case, the controller's display will be updated with the call info of the original call party.

dv-Conference-Party: Given the add party scenario, the host may send dv-Conference-Party, the consult party will be conferenced in to the call between the controller and the original call party, see Figure 68.

Figure 68 Conferenced call



In this case the displays of all three parties will be updated to indicate a conference call.

4.17.3.3 ACD Walkaway feature interaction with 500/2500 agent positions and TPAC messages

Due to 500/2500 telephone set limitations, the ACD Walkaway feature is not available to agents operating from these positions. The CompuCALL dv-Set-Feature (Not Ready) message, will now provide this capability through Third Party Agent Control (TPAC). Upon a change in state for an agent position, it is reported for 500/2500 and MBS agent positions alike. All the conditions and feature operation of the ACD Walkaway feature for the MBS are now applicable to 500/2500 positions, if activated through TPAC.

4.17.3.4 3WCPUB feature

SCAI generated consult or conference calls are not compatible with the 3WCPUB feature. The 3WCPUB feature will be ignored by SCAI 3WC.

4.17.3.5 911

After a dv-Add-Party to a 911 (ESL/ESB), dv-Drop-Party is not allowed after the call is answered by the ESL/ESB.

4.17.3.6 Programming features

Features like speed call and call forward cannot be programmed when there is an active call on the set. As a result, programming features are blocked when SCAI 3WC or MakeCall is active.

4.17.3.7 Terminating features

SCAI 3WC or MakeCall will interact with termination features (for example call forwarding) the same as normal. As a result, there is no need for any special codes for these interactions.

4.17.3.8 Feature activation

No feature can be activated by either a dv-MakeCall or dv-Add-Party request (e.g., speed call, call pick up).

4.17.3.9 Translation features

Authcodes and account codes are blocked for dv-Add-Party requests, but are allowed via MakeCall.

Features like DISA are blocked as are those that require further digit collection before the connection has been established for dv-MakeCall or dv-Add-Party requests.

None of the translation type features (e.g., speed call, LNR--last number redial) can be used by dv-Add-Party or dv-MakeCall to set up the translations call access codes. But, once the appropriate access codes have been datafilled or set up via the phone, MakeCall or AddParty may contain the appropriate translation access code as the destination address.

last number redial (LNR): on a successful dv-Add-Party or dv-MakeCall, the destAddress will be stored as the last number dialed from the controller's set.

4.17.3.10 Transfer recall

FXR recall timer is not used by the SCAI transfer feature. As the 3WC feature, only CXR recall timer is functioning on this feature.

4.17.3.11 Hold key

The hold key is a local hold and does not affect any of these SCAI call control ROs. Hence if the controlling party hits the hold key while talking to an incoming call, the controlling party holds that incoming call. While the call is still being held, if a dv-Add-Party is sent and is successful, the hold key now holds the connection to the add-on party. The add-on party receives silence (unless music was datafilled). If the controlling party hits the DN key to re-access the call, it is not the add-on call.

4.17.3.12 Other DN keys

The same effect as described for the hold key occurs by hitting another DN key. This puts the incoming call on hold. If a dv-Add-Party is sent at this point, an add-on call is made but the connection to the add-on call is held while the controlling party is still occupied on the other DN. Hitting the DN key which originated the add party request will permit the controlling party to now talk with the add-on call. The original incoming call was put on hold by the dv-Add-Party request.

4.17.3.13 Answer call and 500/2500 sets

The dv-Answer-Call RO is not supported on any 500/2500 sets except 500/2500 sets with the call waiting option can answer the call waited call by sending the dv-Answer-Call message.

4.17.3.14 Maestro sets and hold call

The display and lamps are not updated when using dv-Hold-Call on Maestro sets.

On Maestro sets, with set based hold active, SCAI 3WC (Add, Drop, Transfer) messages will work.

Suppose a Maestro set is placed on hold by using the hold key. Next, a dv-Hold-Call message is sent to hold the Maestro set. As a result, both Maestro set based hold and CompuCALL hold (dv-Hold-Call) are active. The set based hold is deactivated either by using the hold key press or flash key press, and there is no dv-Call-Unheld message sent to the host. At this point, speech path is not re-establish because the CompuCALL hold is still active. CompuCALL hold is deactivated by using the flash key press or sending a dv-Unhold-Call.

4.17.3.15 Release call with attendant consoles

Calls involving a user and an attendant console (AC) are not releasable using dv-Release-Call. If a user is connected to an AC or ringing an AC, a dv-Release-Call request is not processed, a Return Error of Call-Not-Releasable-By-User is returned to the host, and the call continues without being affected.

4.17.3.16 Add party with 3WC/CXR key on EBS sets

If the EBS line has 3WC/CXR key and is associated to a CompuCALL session, the lamp update will occur as mentioned below when the host computer does 3WC or call transfer.

- dv-Add-Party: When the dv-Add-Party is successfully processed and the Return-Result is sent to the host computer, a three way consult call is initiated. The key on which 3WC or call transfer exists will be determined and the LAMP turned ON(solid). With dv-Add-Party, the user cannot toggle between the first leg and the second leg(consult leg) either manually or using CAS and hence there is no necessity to flash the DN key lamp when the caller is put on auto-hold. The result will be such that both the DN Key lamp and the 3WC/CXR lamp will be solid. If the consulted DN answers the call and releases the consult call, the controller’s 3WC/CXR lamp will be turned off. Similarly, if the Controller releases the consult call, the 3WC/CXR lamp will be turned off and the DN key will be rung to alert the user to answer the first leg call. Except for the lamp update, this behavior of releasing and alerting to answer the first leg call exists with CAS feature.

- dv-Conference-Party: The 3WC/CXR lamp will be turned off.
- dv-Drop-Party, dv-Transfer-Party: When either of the messages are successfully processed, the 3WC/CXR lamp will be turned off.

4.17.3.17 Display set updates with MakeCall originated calls

The display with respect to MakeCall originated calls will (for display sets):

- on the originator’s set, a message will be displayed showing that the call is an outgoing call to the destination DN,
- on the terminator’s set, the lamps and display will be updated as if the originator had called the destination directly

4.17.3.18 Display set updates with call transferred calls

The display with respect to dv-Call-Transferred-U event:

When the display of the consult party is updated with call transfer then the dv-Call-Transferred-U event message is sent to the host.

The otherPtyDN parameter will be included in this message only in the case of a transfer to a two-party call.

If the consult party is transferred into a conference call, the otherPtyDN parameter will not be included. This has been done in order to keep constant with the set display information: in this case, the consult party’s set display shows ‘CONFERENCE / CALL TRANSFER’ rather than the DN of the first leg party.

4.17.3.19 Make call beep Hset and XPM

The XPM software load of the peripheral to which the originating lines belong, must be XPM05 or above for the beep HSET option to work.

4.17.3.20 Calling number delivery blocking(CNDB) and suppress feature interactions with 3WC call event messages

The firstLegDN of the dv-Call-Conferenced-U message will not be provided to the controller if the controller is the terminator of the first leg call. If the controller is the originator of the first leg call the firstLegDN will be provided in the dv-Call-Conferenced-U message.

For intra-switch second leg call, the ControllerDN field of the dv-Call-Conferenced-U message will not be provided to the consult party if such blocking features interact.

The otherPtyDN of the dv-Call-Transferred-U message will not be provided if the CNDB and suppress features are datafilled.

4.17.4 Restrictions and limitations

This section summarizes the restrictions and limitations associated with this offering of DMS-100 Meridian CompuCALL options.

4.17.4.1 Reason for call release

No indication is available in the dv-Call-Released-U message to distinguish whether the agent or the caller in a two-party call has cleared an established call. In a call which involves more than two parties, no CompuCALL messages are sent to the host when a non-CompuCALL party drops out.

4.17.4.2 Original calling number for transferred/forwarded call

The OrigCallingNumber parameter for a call which has been transferred to the CompuCALL environment from a non-CompuCALL line on the same switch will be the DN of the transferring line (i.e., not the DN of the original calling line). The exception is calls which are transferred from a Meridian attendant console on the switch; in this case, the OrigCallingNumber parameter will be the DN of the original calling line.

If a call is forwarded to the CompuCALL environment by a non-CompuCALL party on the same switch, the OrigCallingNumber parameter for this call is the calling DN to the forwarding line, or the calling DN to the first forwarding line in a multi-forwarding situation (before entering the CompuCALL environment). This is different, however, for the OrigCallingNumber parameter for a call which has been forwarded to the CompuCALL environment from another switch over an SS7 or ISDN PRI trunk will be the DN of the original calling line.

4.17.4.3 CompuCALL and MDC 3WC/call transfer

CompuCALL call party messages cannot be used interchangeably with MDC 3WC/call transfer. Although they provide similar functionality, they are not interchangeable on a per call basis. The user cannot perform any actions from the MBS/2500 set in conjunction with CompuCALL signaling functions for the same call. For example, if the agent uses the dv-Add-Party to consult a third party, then dv-Drop-Party or dv-Transfer-Party must be used to revert it to a two-party call. Therefore, the agent cannot start a CompuCALL call using CompuCALL call party messages and finish it using the MBS/2500 set or vice versa, and similarly the user cannot start a 3WC/call transfer using the MBS/2500 set and continue using CompuCALL call party messages.

Additionally, when using CompuCALL signaling to perform the equivalent of MDC 3WC/Call Transfer, the lamps of an MBS set will not be updated. If an ACD agent is using a data terminal connected to the host computer, then they should rely on the host to keep them informed on the progress of their Meridian ACD CompuCALL requests. The display capabilities will be the same as existing MDC 3WC/call transfer.

> ***Note 1:*** The 3WC/call transfer capability must be assigned to the set (3WC/CXR) or on the customer group (CXFER) for CompuCALL call party messages to be used for those sets. If only 3WC is subscribed (i.e., without transferring capabilities), the CompuCALL Call Party feature is available for consultation and/or conferencing, not transferring. If CompuCALL call party call with transferring is needed, then the call transfer capability should also be assigned to the set or the customer group to which the agent belongs. If the Call Transfer capability is not enabled for the set but transfer is requested via dv-Transfer-Party, the consult party will be dropped. (If call transfer is requested via the voice set in the MDC scenario, all parties in the call will be released.)

> ***Note 2:*** The hold key on the voice set is a local hold and does not affect the CompuCALL messages. Hence if the controller depresses the hold key while talking on an incoming call, the controller keeps that incoming call on hold. If a dv-Add-Party message is sent and successful while the incoming call is still being held, the hold key now holds the connection to the consult party as well as the connection to the original call party. The consult party is receiving silence (unless music was datafilled). The controller can now depress the Incalls key to talk to the consult party. When a dv-Drop-Party message is sent via the host, the consult party is dropped and the controller is re-connected to the original calling party. If the Release key on the voice set is depressed instead of sending a dv-Drop-Party message, the consult party is released and the controller will be ringing for the original calling party.

The same effect as described above occurs when a secondary DN (SDN) key is depressed during an active call. This puts the incoming call on hold. If a dv-Add-Party is sent at this point, a consult call is made but the connection to the consult party is held while the controller is still occupied on the SDN. Depressing the Incalls key at this time will permit the controller to talk to the consult party. The switch put the original call party on hold when the dv-Add-Party message was received. The controller can retrieve the connection to the original calling party the same way as described above when the hold key is used.

There is no need for a 3WC/CXR feature to be on the set. It is not used.

If the controller is in a conference call and hits the RLS key (EBS) or goes on-hook (EBS or 500/2500), the consult leg drops out of the call and the original two-party call member (party-B) re-rings the controller (ringback). In a normal 3WC scenario, the controller drops out of the conference call and a transfer occurs.

The existing Centrex and residential 3WC, call transfer and call conference feature restrictions will apply as well as any imposed by the CompuCALL ECM application.

The add/drop/transfer/conference messages must be subscribed to in table SCAISSRV. If a message is not subscribed to, then a request will be denied, and CompuCALL will send a Return Error specifying that the message is not allowed.

An add party will only be allowed to originate from a DN that has a call established (the specified originator's DN is connected to another party and in a talking state). Thus, if an originator is still ringing, dialing, outpulsing, etc. on another call then the add party will not be allowed. The switch will send a return error specifying that the line is in an invalid call state.

The drop/transfer/conference party messages are all enabled by first performing an add party operation to set up a consult leg call on the line. If either drop, transfer or conference party is requested on behalf of a DN without having first set up that DN as a SCAI controller via add party, a return error will be sent from the switch specifying that the DN is in an invalid call state.

4.17.4.4 MDC/RES MakeCall

The following restrictions and limitations will apply to residential and Centrex line originated MakeCalls:

So-called predictive capabilities, in which calls are set up while the target line is still busy on other call(s) is not supported.

Lines that are the targets (e.g., originating end) for MakeCall calls must be part of the business group identified during ECM session establishment (dv-Appl-Logon). Furthermore, these lines must have a valid DN (single appearance directory number) which has the appropriate CompuCALL line option assigned to the line.

The line specified by the origination address must be idle at the time the dv-Make-Call request is received. If not, the request will be not allowed. Note that being off-hook and listening to dial tone is not considered idle.

When an outbound Makecall call is originated on behalf of a line which is associated to a CompuCALL session, a call released message will be sent to the associated session when the either party releases the call.

The MakeCall message must be subscribed to in table SCAISSRV. If the message is not subscribed to, then a MakeCall request will be denied, and CompuCALL will send a Return Error specifying that the MakeCall is not allowed.

The MakeCall message must be subscribed to by the line via the ECM line sub-option.

MakeCall will only be supported for single line appearance directory numbers (DNs). Thus an origination address specifying a MADN DN, UCD DN, ACD DN, or Teen Service SDN will result in a return error indicating an invalid origination address.

If calls are terminated to the SDN of the Teen Service feature, the associated call event messages will be sent for that call specifying the SDN as the terminating device ID. Only the primary DN needs to be associated (the SDNs cannot be associated).

Conversely, calls may not be originated from the SDN of the Teen Service feature via MakeCall, nor may the Teen Service SDN be specified as a SCAI conferencing/transfer controller.

Enhanced Teen Service is not currently supported.

If automatic call distribution (ACD) originates a call via the MakeCall message and terminates to a residential line, the call offered message, if subscribed to for outgoing calls will be sent. In addition, the call offered message will be sent for add party consult calls.

If a host logs on with a service version that existed prior to service version SCAI07, then the residential or Centrex originated MakeCall request will be denied as it always has been for those previous versions.

4.17.4.5 Meridian ACD CompuCALL and translation type features

None of the translation type features (e.g., speed call) can be used by the dv-Add-Party to setup the consult party.

4.17.4.6 Meridian ACD CompuCALL and Conference Splitting

When the consult call is established by dv-Add-Party, the controller cannot alternate between the original call party and the consult party.

4.17.4.7 Meridian ACD CompuCALL and Agent Set Features

Features like Authcodes, account codes, and DISA cannot be used with dv-Add-Party.

Last number redial (LNR) cannot be used by dv-Add-Party. However on a successful dv-Add-Party, the DestAddress will be stored as the last number dialed from the controller set.

4.17.4.8 Meridian ACD CompuCALL and MDC Attendant Console

Calls transferred by an MDC Console cannot be redirected with dv-Call-Redirect.

4.17.4.9 Agent-to-Agent transfer

In the case of an ACD call transfer by one agent from his or her Incalls key to another agent's Incalls key, i.e., using the Transfer-to-Incalls-Key feature, the ACDDN parameter in the dv-Call-Queued-U, dv-Call-Offered-U and dv-Call-Answered-U messages for the receiving agent for the transferred call will be the primary ACDDN of the ACD group to which this agent belongs. There is currently no explicit indication to the host that this is an agent-to-agent transfer. Calls which have been transferred by the Transfer-to-Incalls-Key feature cannot be redirected.

4.17.4.10 InvokeID management and dv-Call-Received-C message

When the dv-Call-Received-C timer expires without a response from the host, its InvokeID is cleared in the switch. Similarly if the caller abandons before time-out, the dv-Call-Released-U message will be sent and the InvokeID will be cleared. This results in a reject with reason "Unrecognized-Linked-identifier" to any subsequent response from the host using this InvokeID.

The value of the time-out must be datafilled in DMS-100. It ranges from 1 to 30 seconds. The recommended value is 2 seconds.

4.17.4.11CallID interactions with restarts

When cold or reload restarts occur, the CallID will be reset to one.

4.17.4.12 dv-Transfer-Party Interactions with Transfer Recall feature

After an ACD agent (controller) with a Transfer Recall feature has sent a dv-Transfer-Party via the host to transfer a call to a consult party, a dv-Call-Released-U message with ReleaseReason "CallTransferred" is sent for this controller. If the consult party does not answer, the controller is recalled, i.e., both the controller and the consult party are rung at the same time. No additional dv-Call-Offered-U message is sent for the controller. If the consult party answers the call at that time, the call will be treated as if it was a normal transferred call from the controller, and no further dv-Call-Released-U will be sent to the controller. However, if the controller answers this recalled call, only a dv-Call-Released-U message with ReleaseReason "PartyDroppedNoAnswer" is sent for the consult party and no further dv-Call-Released-U message is sent for the controller.

4.17.4.13 Zero length HostCallData

If the switch receives a message which contains the HostCallData parameter with a length of zero (0), it would respond with a Return Error with reason "MissingHostCallData".

If the host's application program does not wish to include HostCallData, the parameter should not be included in the respective message. If the host's application program wishes to initialize (i.e., blank out) the HostCallData parameter, the application should perform this using valid information which is at the discretion of the application.

4.17.4.14 ACD agent position logIn via CompuCALL dv-Set-Feature (logIn function) message

The line state of the agent position, whether 500/2500 set or the InCalls key on an MBS, must be idle when logging in an agent position using the dv-Set-Feature (LogIn function) message.

The dv-Set-Feature (LogIn function) message can only be used to log in an agent position and not to be used to undo or toggle log out requests (either manually or via CompuCALL). This series of actions of toggling between login/logout is supported through manual activation but not through the CompuCALL dv-Set-Feature (LogIn/LogOut function) messages.

4.17.4.15 ACD origination via MakeCall

If automatic call distribution (ACD) originates a call via the MakeCall message and terminates to a residential line, the call offered message, if subscribed to for outgoing calls will be sent. In addition, the call offered message will be sent for add party consult calls.

4.17.4.16 LOB feature with CompuCALL

The two restrictions that apply to the feature remain after its integration with CompuCALL:

- If the agent enters more than five LOB codes and erases the last one, it also erases the fourth because the feature software stores a maximum of three LOB codes. Since CompuCALL has no such restrictions, the LOB codes reported by CompuCALL (through the dv-LOB-Event-U) message can be different from the ones reported by ACD MIS.
- If the agent does not enter any LOB code at all, the ACD MIS reports the default code for that group if the switch datafills one. CompuCALL does not report any code.

4.17.4.17 Release call and multiple calls

If a user is involved in a multiple call scenario, other than via 3WC, for example via ACD observe or 6 way conferencing, then the user will not be allowed to release any of the calls using dv-Release-Call and an error message will be sent to the host.

4.17.4.18 Answer call use

Only incoming calls and dv-Make-Call alerting may be answered via the dv-Answer-Call message. The dv-Answer-Call message may not be used to answer the other-end party of outgoing calls.

4.17.4.19 ACD agent position Ready/Not Ready activation via CompuCALL dv-Set-Feature (Ready/Not Ready function) message

For ACD agent positions to activate the Ready or Not Ready agent functions, the following criteria must be met:

- the ACD agent must be logged in, either manually or via CompuCALL
- the line state of the InCalls key of an MBS agent position must be idle or active on a call; the line state of the 500/2500 agent position must be idle or active on a call
- the agent position may not be in a transient state such as being involved in another feature e.g., walkaway
- the agent position must have the ACD Not Ready feature available.

4.17.4.20 Meridian ACD CompuCALL Call Redirection Message and Off-Board Processor Access features

The CompuCALL application service function, call redirection, does not support the redirection of a call to a number which requires a query to an off-board processor. This may disallow calls to be redirected to an 800 or DMS AIN number.

4.17.4.21 Residential and Centrex call event messages

The following restrictions and limitations will apply to Residential and Centrex incoming and outgoing call event messages:

- Each event message and parameters of each required by the host application must be subscribed to in table SCAISSRV.
- The following messages will be sent for all incoming calls: call offered, call answered and call released.
- The following messages will be sent for all outgoing calls: set off hook and call released. The only exception here is that the set off hook message will not be sent for MakeCall originated calls or add party originated consult leg calls.
- The set off hook message is really an off-hook signal; however, when we are notified that the set has gone off-hook we are not yet aware of the digits-dialed, so we cannot say who the terminator is.
- If both the originator and terminator of a call are associated to a CompuCALL session, then the originator's associated host will receive the supported outgoing call event messages. The terminator's associated host will receive the supported incoming call event messages even if both of the parties are associated to the same host computer.
- The incoming event messages (call offered, call answered and call released) will not be delivered if call forward intragroup (CFI), call forward universal (CFU) or call forward busy (CFB) is activated on an

associated busy line. Conversely, if a associated line has call forward don't answer (CFD) activated and a call comes in to the line, a call offered message will be sent. If the line is not answered and the call is forwarded, a call released message will be sent with reason of call forwarded.

- Network call ID is not passed from switch to switch.
- Since the ACD call event messages are not enhanced to include the new dialable number and name fields, the ACD name is not available to send in the residential and Centrex call history.
- Since the RES/CTX messages are not enhanced to include the new functionality, status query message and the response messages are not available for these lines.

4.17.4.22 Local CallID wrap-around

Given that the CallID has a fixed maximum number of bits, it will still wrap around at some point in time for a given application. The time and frequency of occurrence depend on the traffic rate of calls reported on across the CompuCALL interface(s) for the switch.

After the CallIDs have been used and released, they are not reused until they wrap around #7FFFFFFF (Hex) other different CallIDs. Therefore, assuming a present maximum of 15.2 CompuCALL calls per second continuous operation, the CallIDs will not be reused in approximately 4.5 years. Implicit in this statement is the need for the host to audit its CallIDs and remove those which have remained for a period of more than a reasonable duration, e.g., 7 days.

4.17.4.23 Maximum number of redirections

A maximum of 2 consecutive call redirections is allowed for a given call within a switch after which the call remains at the last redirected destination. The redirection counter is reset (set to zero) under call forward don't answer conditions only. In other words, if the call was involved with any other type of call forwarding, the redirection counter is not reset.

4.17.4.24 BusinessGroupId restriction and CompuCALL

In host-originated CompuCALL service capabilities (Third Party Call Control, resource query), it is not necessary for the respective DN for which a service is requested to be in an associated environment. Therefore, the BusinessGroupId is used to check the integrity of the service request. This applies to the messages in the Third Party Call Control and resource query capabilities.

4.17.4.25 Interswitch Call Forwarding and CompuCALL

The forwarding party parameter is supported for forwarded calls between switches in an SS7 network but is only partially supported in a PRI network. The difference is that only the "first" forwarding party information, especially in a multi-forward situation, is available from the PRI network.

4.17.4.26 ACD agent supervisor PositionID limitations with TPCC and TPAC

The ACD agent supervisor may also operate as an ACD agent and hence, when using the TPCC or TPAC is limited to use its "Agent PositionID" and not its "Supervisor PositionID". The TPCC ROs affected are: dv-Add-Party, dv-Drop-Party, dv-Transfer-Party, and dv-Conference-Party. The TPAC RO affected is dv-Set-Feature with the following functions: Agent LogIn, Agent LogOut, Agent Ready, and Agent Not Ready.

4.17.4.27 CTX/ACD sets

The follow business sets are supported by CompuCALL 3WC/CXR lamp synchronization and the Make-Call use of Calling_Agent_Buzz_Base and Calling_Agent_Beep_Hset: M5009 MBS, M5112 MBS, M5209 MBS, M5312 MBS, M5212 MBS, EBS, and PSET with Display. The MBS II series of telephone sets, including the M5008 Meridian Business set, M5208 Meridian Business set, M5216 Meridian Business set and M5316 Meridian Business set is supported.

4.17.4.28 3WC/CXR and 500/2500 sets

3WC/CXR lamp synchronization for 500/2500 sets is not supported. Make Call with no alerting on 500/2500 sets is not supported. 500/2500 sets do not support LOB features and LOB events.

4.17.4.29 NICM over PRI trunks

NICM adds the capability to transfer the NICM information over ISUP/SS7 network to the designated DMS-100 switch. Prior to NA11, the NICM information was not available for extended calls (routed, redirected, forwarded, overflowed, transferred) from one DMS-100 to another DMS-100 (interswitching).

NA11 provides the functionality that enables the transfer of NICM information over PRI trunks when the call involves an SL-100 as either the originating or the terminating node. With this functionality the ICM customers on a DMS-100 switch and an SL-100 PBX receive coordinated voice and data (CVD) for networked calls. Messages contain NICM information uder the following circumstances:

- NICM is subscribed at the Switch level
- NICM is subscribed at the end user level
- The ISUP/SS7 network uses the ANSI ISUP and NTNA PRI protocol variant
- SCAI version is SCAI12 and above

4.17.4.30 Meridian Business Set (MBS) display

The ReservAgt request through the NICM does not update the display of the reserved agent.

4.17.4.31 Set-Feature and ICM associations

Set-Feature reserves or unreserves an agent despite ecm-association.

4.17.4.32 NetNodeID subparameter

For NICM the NetNodeID subparameter is CCS7 OPC.

4.17.4.33 2500 ACD agent originating calls

A 500/2500 ACD agent does not send a message to a host when a call is originated from a set that does not have an ECM option assigned. An ACD agent can not have the ECM option. The ECM option is necessary in order to send CTXEVENT and RESEVENT service messages. They do not contain the SETOFFHOOK message.

When a 500/2500 ACD agent ends an ACD call, the variable wrap interval begins. If the agent originates and terminates a call before the interval expires, the switch does not send a dv-agent-setaction-u message with status as interval ended. This occurs only if there are no calls in the ACD queue. If the agent originates a call during the interval and terminates after the interval ends, the switch does not send a dv-agent-setaction-u message. The reason is the agent is not idle at the time of interval expiration.

4.17.4.34 CTX/ACD sets

The feature ICM Variable Wrap Reporting supports the following business sets: M5009 MBS, M5112 MBS, M5209 MBS, M5312 MBS, M5212 MBS, EBS, and PSET with display. Also the MBS II series of telephone sets: M5008 Meridian Business Set, M5208 Meridian Business Set, M5216 Meridian Business Set, and M5316 Meridian Business Set.

4.17.4.35 500/2500 sets

The feature ICM Variable Wrap Reporting support these sets because they are capable of sending agent status to the switch through star codes.

4.17.4.36 SCAI workqueue

Prior to NA012, the SCAI workqueue handled a maximum 128 messages. In high traffic scenarios, the messages exceed this limit. As a result, those messages that exceeded the limit were lost.

In NA012, the ICM software dynamically assigns temporary incoming/outgoing workqueues (from a pool of up to ten temporary incoming and outgoing queues) when the number of messages in the incoming/outgoing SCAI workqueues reaches a threshold value of 128. The pool of ten temporary incoming and outgoing queues is created at the time of datafilling the first ten links in the table SCAICOMS. This pool serves any of the 271 links which experiences high traffic. Each temporary queue can support 512 more messages. The total message enqueuing capacity can support up to 640 messages. Since the ICM processes dequeue and enqueue messages from the queues at the same time, the actual message handling capacity can exceed 640 messages.

4.17.4.37 Continuity Test Enhancements

Prior to NA012, the switch sent the continuity messages at continuity test activation. The maximum number of messages are equal to the number of reattempts after waiting for the response timer for each attempt. The default values for the response timer was 10 secs and for the number of re-attempts was 5.

In NA012, parameters in table SCAICOMS allow customers to program the following fields:

- AUDINTVL allows customers to datafill the audit interval of the continuity test (1 to 720 minutes).
- RESPTIME allows customers to datafill the time for which the switch waits for a response from the host computer after the switch sends the continuity message (1 to 30 seconds).
- NUMATMPT allows customers to datafill the maximum number of times the switch sends a continuity test message for a particular link. If the switch does not receive a RETURN RESULT from the host, then the switch sends the next continuity test message. This repeats for the NUMATMPT (1 through 10) times. If the switch receives a RETURN ERROR/RETURN REJECT response, the switch sends the next continuity test message immediately
- TERMINET allows customers to datafill, whether or not to terminate the session when the host does not send an appropriate response (Y or N).

5.0 Application service parameter definitions

5.1 RO parameter introduction

This chapter defines the Remote Operation (RO) parameters in Chapter 4.0 Application service options. The parameters are covered alphabetically.

For each parameter, this section provides:

- the format (integer, choice)

  Note: "Choice" is defined as a parameter type. Section 5.2 provides the formats of each parameter type. Section 5.3 provides the secondary parameters in the parameter types.
- the range of operation (e.g., 0 to 127)
- the ranges of RETURN-ERROR (64 to 127) and REJECT (128 to 255), where applicable

  Note: Where "integer" format is specified, the field is variable length and the minimum number of bytes to send the number is used. The range of each parameter reflects the present status of service implementation. This does not preclude this range from being changed (increased) in the future, if necessary.
- the static or dynamic call-related information nature of the parameter where applicable
- a brief description with context ("uniquely identifies a host application in a given customer's network");" customer" refers to the end customer -the customer who uses the service
- the assignment ("assigned by the customer and datafilled against that customer in DMS"), where applicable
- a cross-reference to the specific RO messages in this chapter which may contain the parameter

5.2 Parameters

This section provides definitions of the remote operation (primary) parameters in the application service functions described in Chapter 7.0 Service model.

5.2.1 ApplicationID

- Integer, range 1 to 32 767 (one or two byte), encoded in 2's complement.
- Any value in the range of 32 768 to 65 535 or with a value of 0 causes a RETURN-ERROR for this parameter.
- The REJECT range is a value of more than 65 535.
- Uniquely identifies a host in a customer's network within the set of public and private switches with which a customer's host application can establish a session; functionally equal to the "UserID" for mainframe computers.
- This parameter does not specify the actual physical host computer since a host application may not always reside on the same physical host.
- Assigned by the customer
- Used in the dv-Appl-Logon RO and the RETURN-RESULT of the dv-Appl-Continuity-Test RO

5.2.2 AnsweringParty

- Choice, 4 items, AddressType format
- The AddressType is the only mandatory parameter which can be specified as a position ID or stationNumber.
- Identifies the user that the host is making the answer request on behalf of.
- Used in the dv-Answer-Call RO

5.2.3 AssociatedAgent

- The maximum number of ACD agent positions that are associated with a single ACD group is 1024.
- It is an optional parameter.
- Identifies an ACD agent position id
- Used in dv-DN-Associate RO

5.2.4 AssociatedDN

- Octet string, 10 bytes
- Encoded in ASCII to indicate the DN to be associated
- The range of RETURN-ERROR is from 11 to 50 bytes. The reject range is a length of more than 50 bytes.

- The AssociatedDN in North America is a 10-digit North American Numbering Plan (NANP) DN unique in the North American public switched telephone network (including PBXs with Direct Inward Dialing) if a public service (ACD calls placed as public network calls). However, for a private service (calls placed only as "on-net" private network calls), the AssociatedDN is unique in a customer's private network or in a given public or private switch. In this case, even where callers dial an abbreviated private DN like an extension number, it is assumed that the switch has (and can provide the host application with) a corresponding DN which is the DN the host application should specify in the dv-DN-Associate message. Numbering plans other than NANP are currently under consideration for international implementations; these numbering plans will be addressed in later BCS releases:
- Assigned by the telephone company
- Used in the dv-DN-Associate RO

5.2.5 AuxDeviceDN

- The parameter is optional
- If the parameter is requested in Table SCAISSRV, the value of the parameter is 0 in the dv-EMK-U message.

5.2.6 BusinessGroupID

- Integer, range 1 to 4 194 304 (one to three byte), encoded in 2's complement.
- The value of RETURN-ERROR for this parameter is 0. The reject range is a value of more than 4 194 304.
- Uniquely identifies an meridian business group (MBG) customer within a public network.

  Note: An MBG customer is a telephone company customer who uses public facilities to carry customer specific information. This parameter uniquely identifies a given telephone company MBG customer across a number of switches in a given network, and not only within a given switch. This is so a host application which establishes application level sessions with multiple switches only needs to be datafilled with a BusinessGroupID per network rather than per switch.
- Assigned by the telephone company
- Used in the dv-Appl-Logon RO

5.2.7 CDN

- The parameter is mandatory.
- Identifies the address of the CDN which is in the station number format

- Optional AddressType format.
- state-identifies if the CDN is default, controlled, revert to default:
  - — DEFAULT-initially all of the calls are in the DEFAULT state
  - — CONTROLLED-the host computer handles incoming calls
  - — REVERT TO DEFAULT-calls in the CDN queues are routed to the default ACD DN
- setCDNStateError-identifies the error that causes the CDN state request to fail.

5.2.8 HoldingParty

- Optional, AddressType format
- The AddressType is the only mandatory parameter which can be specified as a position ID or stationNumber.
- Identifies the user that the host is making the hold request on behalf of
- Used in the dv-Hold-Call RO

5.2.9 Destination

- Identifies the destination to terminate
- Has a maximum of 32 digits
- Defined as a choice parameter
- Optional DestinationInfo format

5.2.10 DNOperation

- Enumerated, 3 items
- This parameter must be provided by the host application and specifies if the specified DN is to be added to, removed from, or replace the current set of DNs associated with the session.
  - — Add
    - – The specified DN is added to the set of DNs associated with the session. If no DN is specified in the message, the current set is unaltered.
  - — Delete
    - – The specified DN is deleted from the set of DNs associated with the session. If no DN is specified, the current set is unaltered.
  - — Define
    - – This subparameter is only used for ACD DN association. Only the specified ACD DN is associated with the current session. All

previous ACD associations are deleted. If no ACD DN is specified, all of the customer's ACD DNs not currently associated with another session are associated with the current session.

The dv-DN-Associate message with the DEFINE operation is only supported for ACD DNs:

- A dv-DN-Associate message with the DEFINE operation and no DN in the associated DN parameter associates the host computer to all the ACD DNs within the host's customer group which are not associated with another session.
- A dv-DN-Associate message with the DEFINE operation and an ACD DN in the associated DN parameter disassociates all the ACD DNs which are associated by that host computer except for that specified ACD DN.
- A dv-DN-Associate message with the DEFINE operation and a non-ACD DN in the associated DN parameter fails with an Invalid DN error message.
- Used in the dv-DN-Associate RO.

5.2.11 Emergency key activity

- A mandatory parameter
- Can select the parameter in Table SCAISSRV for EMKEVENT message category
- 47 bytes long and type sequence

5.2.12 FLAG

- applies to dv-Agent-Status-U and dv-CDN-Status-U
- 1 byte long decimal
- for CDN status parameter indicates all done
- for Agent status parameter indicates upload in progress

5.2.13 LOBCode

- Octet string
- 3 bytes in length
- Contains 0x03, 0x05 and 0x01 representing the LOBCode value of ‘351’
- For the MIS event message, the agent can enter a maximum of three codes during the call.

5.2.14 LOBTime

- A constructor and optional parameter

- 9 bytes in length
- Associates a time-stamp with the message

5.2.15 NetworkNodeID

- Integer, range 0 to 32 767 (one or two byte), encoded in 2's complement
- The reject range is a value of more than 32 767.
- Uniquely identifies a switch (CO switch or PBX) in a customer's network (in the set of public and private switches with which a customer's host applications can establish an application-to-application level session).
- The customer assigns the NetworkNodeID to assure uniqueness in his network which may consist of public and private nodes.
- Datafilled against the customer in DMS
- Used in the dv-Appl-Logon RO

5.2.16 Password

- Used for security purposes
- Variable length IA5String (1 to a maximum of 8 characters)
- The range of RETURN-ERROR is from 11 to 50 characters. The reject range is a value of more than 50.
- Has a one-to-one correspondence with the BusinessGroupID, i.e., all Service IDs of a customer will be under the same password
- Assigned by the customer and datafilled against the customer on the DMS
- Used in the dv-Appl-Logon RO.

5.2.17 Postcall Status

- Indicates the status of the agent once the agent’s side of the call releases.
- 0 = null
- 1 = variable
- 2 = zero interval
- 3 = release guard started
- 4 = not ready
- 5 = logged out
- dv-Call-Released-U

5.2.18 ReleasingParty

- Optional, AddressType format.
- The AddressType is the only mandatory parameter which can be specified as a position ID or stationNumber.
- Identifies the user that the host is making the release request
- Used in the dv-Release-Call RO.

5.2.19 ReleaseType

- Enumerated, 2 items, both optional
- Consists of releaseThisParty and releaseAllParties
- Specifies whether to release this party alone or release all parties in a call
- Used in the dv-Release-Call RO

5.2.20 ServiceID

- Integer, range 0 to 255 (one or two byte), encoded in 2's complement
- The return error range is a value between 255 and 65 535. The reject range is a value of more than 65 535.
- Uniquely identifies a service profile for a customer
- Used by the host to specify which service profile to establish a session
- Assigned by the telephone company
- Used in the dv-Appl-Logon RO
- ServiceVersion
- Integer, range 0 to 65 535 (one byte), encoded in 2's complement
- The REJECT range is a value of more than 65 535.
- Uniquely identifies the application level signaling version on a specific switch
- Used in the dv-Appl-Logon RO.

5.2.21 STATUS

- applies to dv-Agent-Status-U and dv-CDN-Status-U
- for Agent status, parameter is constuctor length 21 bytes decimal
- for CDN status, parameter is a primitive 1 byte long
- status of a CDN: default, controlled, revert to default

5.2.22 Supervisor’s PosID

- Optional parameter
- identifies the position of the supervisor
- can select in table SCAISSRV
- the value is NIL in the dv-EMK-U message

5.2.23 TreatmentCompleted

- Audios is the only valid treatment to be included.
- Audio route is included with the AUDIO key word.
- Third Party Queue Control (TPQC) is the new service category.
- The host request multiple RANs and Musics to a call.

5.2.24 UnheldParty

- Mandatory, AddressType format
- The AddressType is the mandatory parameter which can be specified as a position ID or stationNumber which has been unheld.
- Used in the dv-Call-Unheld RO

5.2.25 UnholdingParty

- Optional, AddressType format
- The AddressType is the only mandatory parameter which can be specified as a position ID or stationNumber.
- Identifies the user that the host is making the unhold request on behalf of
- Used in the dv-Unhold-Call RO

5.3 acdDN

- Octet string, 10 bytes
- Contains dynamic call-related information
- This parameter is the 10-digit primary or supplementary ACD DN associated with the incoming ACD call for the current CompuCALL party; this may be the called number received by the switch (e.g. the SS7 Called Party Number), which may or may not be the actual number dialed by the calling party (e.g. in the case of 800 service it is the 10 digit NANP routing number translated from the dialed 800 number).
- Unique in the North American public switched telephone network (including PBXs with Direct Inward Dialing) in the case of a public ACD service (i.e., ACD calls placed as public network calls); unique within a

customer's private network or within a given public or private switch for a given customer in the case of a private ACD service (i.e., ACD calls placed only as "on-net" private network calls) —Note that even in the latter case where callers dial an abbreviated private ACD number, such as an extension number, it is assumed that the switch has (and can provide the host application with) a corresponding 10-digit primary or supplementary ACD DN which, in the case of a primary ACDDN, is the same as the AssociatedDN parameter specified by the host application in the dv-DN-Associate message.

- Assigned by the operating company
- Used in the following ROs:
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U
  - — dv-Call-Queued-U
  - — dv-Call-Received-C
  - — dv-Call-Released-U

5.3.1 acdDNQuery

- Sequence, 5 items
- Contains static and dynamic ACD resource-related information
- Provides the status of the ACD group, its queues and its agent resources associated with a specified ACD DN. It includes at least one field:
  - — time -the time when the ACD information was provided
  - — queryAddress-the ACD DN for which information was requested
  - — grpPrimDN -the primary ACD DN of the ACD group associated with the ACD DN for which information was requested
  - — grpStat -the status of the ACD group
  - — queryInfo -the group, queue and agent information for the ACD group
- Used in the dv-Resource-Query RO

5.3.2 acdGroup

- Choice, 4 items; AddressType format
- Contains dynamic call-related or agent-related information
- The choice for ACDGroup with Meridian ACD CompuCALL options is StationNumber.

- Uniquely identifies the current ACD group in the CompuCALL environment to which the call has been routed or an event had been reported. It is the Primary DN of the ACD group.
- Assigned by the operating company
- Used in the following ROs:
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U
  - — dv-Call-Queued-U
  - — dv-Call-Received-C
  - — dv-Call-Released-U
  - — dv-Agent-Logged-In-U
  - — dv-Agent-Logged-Out-U
  - — dv-Agent-Ready-U
  - — dv-Agent-Not-Ready-U

5.3.3 addPartyType

- Enumerated, 2 items
- Indicates the type of conference or consult call to be set up. The values are:
  - — AddConsultParty-instructs DMS-100 to put the existing call on hold and initiate a call to a 3rd party. It also implies that this call is not intended to be conferenced at that time, so conference resources are not needed.
  - — AddConsultforConf- similar to AddConsultParty request to put the existing call on hold and initiate a third party call, plus it also requests conference resources to be reserved for the call.
- Used in the dv-Add-Party RO

5.3.4 applicationData

- Sequence, 3 items
- Used as a constructor in dv-Make-Call
- Includes one or more of the following subparameters:
  - — MakeCallType
  - — AuthCodeDigits
  - — AcctCodeDigits
- Used in dv-Make-Call RO

5.3.5 callHistoryInfo

- Sequence, 4 items
- Includes one or more of the following subparameters:
  - CallType - if the call to the current CompuCALL party is consult, transfer, conference, overflow (by ACD), redirect (by ACD), or forwarded
  - OrigInboundDN - the first DN at which the call is directed when it enters the CompuCALL environment; if an outbound ACD call initiated by dv-Make-Call, this is irrelevant and is not provided; in the deviceAddressType (if a transferred call, this is the DN of the party who transferred the call)
  - PrevApplicationID - identifies the application ID for the session of the associated environment to which the party extending the call belongs--must be on the same switch), contains the ApplicationID of the associated environment to which the CompuCALL party transferring, overflowing, redirecting, or forwarding the call belongs; this host application is the current "owner" of the call and is identified so the second host application knows where to look for existing call-related information; if an outbound ACD call, this may be the ApplicationID which initiated the dv-Make-Call message
  - HostCallData - a "generic" parameter carrying miscellaneous call-related information sent by a host application which is extending this call. If a call is redirected by a host, this parameter may be provided by that host at that time, and might include the redirection reason.
  - ReturnedtoQueue-indicates the route call is returned to the same ACD queue. When the call is returned to queue it retains the original arrival time. The call is placed at the head of the queue.
- Used in the following ROs:
  - dv-Call-Answered-U
  - dv-Call-Offered-U
  - dv-Call-Queued-U
  - dv-Call-Received-C

5.3.6 callMode

- Enumerated, 2 items
- Contains dynamic call-related information
- Indicates the modality of the call at the time of its arrival
- The values supported are CallWaited and Teen Service:

- — callWaited is delivered if the call is waiting indicating that the call has been "waited" while the line to which the call is offered is busy on an existing call. The line must also have the Call Waiting feature datafilled.
- — teenService is delivered if the call were terminated to a Teen Service Secondary DN.

- Used in the following MDC/RES ROs:
  - — .dv-Call-Offered-U b.1.15
  - — dv-Call-Answered-U

5.3.7 calledAbtNo

- Choice, 4 items; AddressType format
- networkCallID, deviceID, and CalledAt-no are the choices for calledAbtNo with Meridian ACD CompuCALL options. The other choice causes a RETURN-ERROR.
- Identifies the digits entered into a voice response unit (VRU)
- Assigned by the operating company
- Used in the following RO:
  - — dv-Add-Party

5.3.8 destAddress

- Choice, 4 items; AddressType format
- DialedDigits is the choice for destAddress with Meridian ACD CompuCALL options. The other choices cause a RETURN-ERROR.
- Identifies the terminator (other-end) of the call to be set up
- Assigned by the operating company
- Used in the following ROs:
  - — dv-Make-Call
  - — dv-Add-Party

5.3.9 deviceID

- Choice, 4 items; AddressType format
- Contains dynamic call-related information
- PositionID and StationNumber are the choices for DeviceID with Meridian CompuCALL options.
- Uniquely identifies the device for a customer work station in a switch

- Assigned by the operating company
- Used in the following ROs:
  - dv-Set-Offhook-U
  - dv-Call-Answered-U
  - dv-Call-Offered-U
  - dv-Call-Released-U
  - dv-Message-Waiting-U
  - dv-DN-Query-U

5.3.9.1 DNQueryFilter (CHG. NUMBER)

- Bitstring, 3 items
- DN Query Filter is an optional parameter.
- This filter lets the host select which optional information parameters (DN association info, line state info, and message waiting info.) to send in the DN Query Result (this filter does not guarantee subscription to each parameter, only that what is subscribed to may be filtered out on a per-message basis).
- The following three parameters may be selected via this filter:
  - DN association info (associationInfo)
  - line state info (lineStateInfo)
  - message waiting info (msgWaitInfo)
- If the filter isn't sent, all optional information parameters will be sent in the DN Query Result.
- Used in the dv-DN-Query-U RO
- Examples of DNQueryFilter

Figure 69 shows all of the different combinations and permutations of the bit stream of the DN Query Filter, the hex value of the bit stream and the meaning of the DN Query Filter. Bit 1 represents the Association Information, bit 2 represents the Line State Information and bit 3 represents the Message Waiting Information. If bit 1 is set to the value 1 then the host is requesting to query about the DN Association information. Whereas, if bit 2 is set to the value of 1 then the host is requesting to query about the Line State information. If bit 3 is set to the value of 1 then the host is requesting to query the Message Waiting Information. A return error with reason invalid filter will be sent to the host if anything other than the following hex values are in the DN Query Filter: 01, 02, 03, 04, 05, 06, 07.

Figure 69 Bit stream of the DN Query Result

Bits						**HexDN Query Filter**	
0	0	00	0	0	01	01	Association Info
0	0	00	0	0	11	03	Association Info, Line State Info
0	0	00	0	1	01	05	Association Info, Msg Waiting Info
0	0	00	0	1	11	07	Association Info, Line State Info, MsgWaiting Info
0	0	00	0	0	10	02	Line State Info
0	0	00	0	1	00	04	MsgWaiting Info
0	0	00	0	1	10	06	Line State Info, MsgWaiting Info
8	7	65	4	3	21		

5.3.10 dropPartyType

- Enumerated, 1 item
- Indicates who is to be dropped and how. This is a request to drop the consult party and reestablish the connection between the original calling party and the controller. DropConsultParty is the only valid DropPartyType.
- Used in the dv-Drop-Party RO

5.3.11 forwardingParty

- Sequence, 4 items; ForwardingPartyType
- Contains information on the first and the last forwarding party in the call. It identifies the forwarding number and forwarding reason of the parties who forwarded the call the first and the last time before terminating on the current CompuCALL party. This information is provided only if presentation of the DN is allowed as subscribed by the respective parties who forwarded the call. This parameter includes one or more of the following subparameters:

  — FirstFwdNumber

  FirstFwdReason

  — LastFwdNumber

  — LastFwdReason
- Used only in the ForwardingParty parameter in the following ROs for Meridian ACD CompuCALL:

  — dv-Call-Answered-U

  — dv-Call-Offered-U

  — dv-Call-Queued-U

- — dv-Call-Received-C

5.3.12 grpPrimDN

- Choice, 4 items; AddressType format
- Contains static ACD group's primary DN
- StationNumber is the choice for grpPrimDN with Meridian ACD CompuCALL options. The other choices cause a RETURN-ERROR.
- Unique in the North American public switched telephone network (including PBXs with Direct Inward Dialing). If a public ACD service (ACD calls placed as public network calls), unique in a customer's private network or in a public or private switch for a customer.
- Assigned by the operating company
- Used in the dv-Resource-Query RO

5.3.13 grpStat

- Enumerated, 5 items
- Contains dynamic ACD resource information
- Indicates the current operational status of the ACD group associated with the ACDDN for which information is requested. It has the following ACD group states:
  - — Accepting Calls - the ACD group is in service and incoming calls are being serviced or queued.
  - — Overflow - the ACD group is servicing calls currently queued in the Incoming-Call Queue. New calls may be rerouted to a customer-defined destination, another ACD group, an announcement, or a nonACD DN.
  - — Night Service - the ACD group is not servicing calls. Calls may be rerouted to a night service route, if assigned, or given an announcement. Night Service takes effect when all agents are logged off or placed in this state by the supervisor (on the ACD supervisor set).
  - — Controlled InterFlow the ACD group is only servicing calls currently in the Incoming-Call Queue. New calls are not queued for this ACD group but are rerouted to a Controlled InterFlow Route. The ACD supervisor places the ACD group in this state by activating the Controlled Interflow feature.
  - — Status Reroute the ACD group is not servicing or queuing calls. New calls are rerouted to the customer-specified overflow route. The calls currently in the Incoming-Call Queue for this ACD group before the group entering this state remain in queue til answered or treated for

exceeding the customer-specified threshold. An ACD group enters this state due to an equipment fault affecting the agents' telephone lines of this ACD group.

- Used in the dv-Resource-Query RO

5.3.14 logInID

- Implicit IA5String, 4 bytes
- Contains static agent-related information
- Uniquely identifies the ACD agent requesting the association to a PositionID
- Used in the dv-Set-Feature and dv-Agent-Logged-In-U ROs

5.3.15 MessageWaitInfo

- Sequence, 2 items
- Consists of 2 subparameters:
  - — Message Waiting Type (one byte)
  - — Message Waiting Indication (one byte)
- Used in the following ROs:
  - — dv-Message-Waiting-U
  - — dv-DN-Query RETURN RESULT-U

5.3.16 networkCallID

- Sequence, 2 items
- Contains static call-related information
- Consists of 2 subparameters:
  - — NetworkNodeID (one or two bytes)
  - — LocalCallID (one to four bytes)
- A key element of Meridian CompuCALL options service-related messages which provides a unique and reliable identifier for every CompuCALL call. This identifier is used for any information exchange between the DMS and the customer's host application(s) regarding that specific call during its "lifetime."
- Uniquely identifies an incoming call for a customer across all the public and private switches in the customer's network. It cross-references other messages relating to the same call.
- Used in the following ROs:
  - — dv-SET-Offhook-U

— dv-Call-Answered-U

— dv-Call-Offered-U

— dv-Call-Queued-U

— dv-Call-Released-U

— dv-EMK-U

— dv-Call-Received-C

— dv-Call-Redirect

— dv-Make-Call RETURN-RESULT

— dv-Add-Party RETURN-RESULT

— dv-Transfer-Party RETURN-RESULT

— dv-Drop-Party RETURN-RESULT

— dv-Conference-Party RETURN-RESULT

— dv-Route-Call RETURN-RESULT

5.3.17 origAddress

- Choice, 4 items; AddressType format
- Identifies the originator of the call to be set up or the action to be performed
- PositionID is the choice for OrigAddress for Meridian CompuCALL options. Other choices cause a RETURN-ERROR. Uniquely identifies an agent position (not the actual agent) for a customer in a switch.
- Used in the following ROs:

— dv-Make-Call

— dv-Add-Party

— dv-Transfer-Party

— dv-Drop-Party

— dv-Conference-Party

— dv-Set-Feature

5.3.18 origCallingNumber

- Octet string, 6 to 12 bytes
- Contains static call-related information
- Encoded following the format of the Calling Party Number field in Q.931 messages (Section D of the NT ISDN Basic Rate Access Interface Specification, NIS S208-6, Issue 1.0). Contains the DN of the calling line. This may be a 10-digit North American Numbering Plan (NANP) DN. Or

if a call from a PBX, this may be the listed Directory Number (LDN) of the PBX or a private (4-digit extension) number.

- The following values are set if applicable. Otherwise, this parameter isn't sent if the number is unavailable or suppressed.
  - — Type of Number = "000" (TON Unknown)
  - — Number Plan Identification = "0000" (NPI Unknown)
  - — Presentation Indicator = "00" (Presentation Allowed)
  - — Screening Indicator = "11" (Network Provided)
- The address digits provide the originating DN unless it is unavailable. If the origCallingNumber is unavailable or is suppressed, it is not sent.
- Assigned by the operating company
- Used in the following ROs:
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U
  - — dv-Call-Queued-U
  - — dv-Call-Received-C

Note: The origCallingNumber parameter follows the format of the Q.931 Calling Party Number Field and is encoded as a single variable length data element in these ROs and parsed by the host.

Bits 87654321	
0/1Type of NumberNumbering Plan Ext.Identification	Octet 1
1Present.000Screening Ext.IndicatorIndicator	2
	3a*
= 0 Digits = Spare (IA5 characters)	4* etc.
* This octet may be omitted.	
The number digit in octet 4 precedes the number digit in octet 5, etc.	

5.3.18.1origChargeNumber

- Octet string, 10 bytes
- Contains static call-related information

- Has the same value as the Bellcore TR-394 SS7 ISUP Charge Number parameter (also called as "ANI" - Automatic Number Identification) which is a 10-digit number consisting of the caller's three digit NPA plus a seven-digit "billing number".
- Assigned by the operating company of call origination
- Provided if received on an incoming trunk with TR394 SS7
- Used in the following ROs
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U
  - — dv-Call-Queued-U
  - — dv-Call-Received-C

5.3.18.2 origDeviceID

- Set of, 3 items, deviceAddressType format
- Provided in the deviceAddressType format for terminating calls -- same as in the Call Offered message
- Contains static call-related information
- Used in the following ROs:
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U

5.3.19 origInboundDN for ACD

- Octet string, 10 bytes
- Contains static call-related information
- Contains the first DN where the call is directed when it enters the CompuCALL environment. It remains the same as long as it stays in the CompuCALL environment even if it is transferred, overflowed, redirected, or forwarded.
- Used in the CallHistoryInfo parameter in the following ROs:
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U
  - — dv-Call-Queued-U
  - — dv-Call-Received-C

5.3.19.1 origInboundDN for MCD/RES

- Octet string, 10 bytes

- Contains static call-related information
- Contains the first DN where the call is directed when it enters the CompuCALL environment. It remains the same as long as it stays in the CompuCALL environment even if it is transferred, overflowed, redirected, or forwarded.
- Used in the CallHistoryInfo parameter in the following ROs:
  - — dv-Call-Answered-U
  - — dv-Call-Offered-U
  - — dv-Call-Queued-U
  - — dv-Call-Received-C

5.3.20 posID

Choice, 4 items; AddressType format

- Contains dynamic event-related information
- The choice for ACDGroup with Meridian ACD CompuCALL options is PositionID.
- Uniquely identifies the ACD agent position in the CompuCALL environment to which an event was reported
- Assigned by the operating company
- Used in the following ROs:
  - — dv-Agent-Logged-In-U
  - — dv-Agent-Logged-Out-U
  - — dv-Agent-Ready-U
  - — dv-Agent-Not-Ready-U

5.3.21 privateData (Used in the dv-Set-Feature RO)

- Choice, 1 item, SetFeaturePrivateData
- Contains the private data specific to a particular function only applicable to ACD agent functions. The choices of PrivateData are:
  - — logInPrivateData
  - — logOutPrivateData
  - — readyPrivateData
  - — notReadyPrivateData
- Only the notReady agent function requires private data. The subparameter for notReadyPrivateData is described below:

  - — walkawayReason identifies the reason an ACD agent position entered the Not Ready state. The parameter type is implicit IA5STRING and must be a three-digit string with value in the range from 000 to 999.
- Used in the dv-Set-Feature RO.

5.3.22 privateData (Used in the dv-Agent-Ready-U RO)

- Implicit, ReadyPrivateData
- Contains the private data specific to a ACD Agent Ready function. The subparameters of ReadyPrivateData are:
- ACDGroupUniquely identifies the ACD group in the CompuCALL environment to which an ACD agent position is associated. The ACD Group parameter is the Primary DN of the ACD group. It is of type StationNumber and is coded as an implicit OCTECT STRING.
- Used in the dv-Agent-Ready-U RO

5.3.23 privateData (Used in the dv-Agent-Not-Ready-U RO)

- Implicit, NotReadyPrivateData
- This parameter contains the private data specific to a ACD Agent Ready function. The subparameters of NotReadyPrivateData are:
  - — ACDGroupUniquely identifies the ACD group in the CompuCALL environment to which an ACD agent position is associated. The acdGroup parameter is the primary DN of the ACD group. It is of type StationNumber and is coded as an implicit OCTECT STRING.
  - — WalkawayReasonIdentifies the reason an ACD agent position entered the Not Ready state. The parameter type is implicit IA5STRING and must be a three-digit string with values in the range of 000 to 999.
- Used in the dv-Agent-Not-Ready-U RO

5.3.24 queryAddress

- Choice, 4 items; AddressType
- Contains the ACD group's primary or supplementary DN
- StationNumber is the choice for queryAddress with Meridian ACD CompuCALL options. Other choices cause a RETURN-ERROR.
- Unique in the North American public switched telephone network (including PBXs with Direct Inward Dialing) in the case of a public ACD service (i.e. ACD calls placed as public network calls) unique in a customer's private network or in a given public or private switch for a customer.
- Assigned by the operating company

- Used in the dv-Resource-Query RO

5.3.25 Query Filter

- the host selects required information
- there are three parameter choices: AllAgents, AllCDNs, AllDevices

5.3.26 queryInfo

- Sequence, 3 items; QueryTypes
- Contains dynamic ACD resource-related information.
- Contains the current status of the ACD group, ACD queues and ACD agents associated with the ACDDN for which information was requested by the host in the dv-Resource-Query message. It includes one or more of the following subparameters:
  - — GrpInfo - provides ACD group-related information
  - — QInfo - provides ACD queue-related information
  - — AgtInfo - provides ACD agent-related information
- Used in the dv-Resource-Query RO.

5.3.27 redirectDestination

- Choice, 4 items; AddressType format
- DialedDigits is the choice for RedirectDestination for Meridian CompuCALL options. The other choices cause a RETURN-ERROR.
- Specifies the destination of a redirected call
- Used in dv-Call-Redirect RO

5.3.28 releaseReason

- Enumerated, 9 items
- Indicates why a call has been released from a CompuCALL party belonging to an associated environment:
  - — Call Cleared (one party in a two-party CompuCALL Call disconnected after being answered)
  - — Caller Transferred (the CompuCALL party transferred the call and disconnected)
  - — Call Overflowed (call has been overflowed away from the CompuCALL party by an ACD "overflow" feature)
  - — Call Abandoned (caller disconnected before call was answered by the CompuCALL party - while in the Incoming Call Queue)

  - — Call Redirected (the call has been redirected away from the CompuCALL party by the host application using Call Redirection)
  - — Call Forwarded (call has been forwarded to another station by the MDC "call forward" feature, see Note)
  - — Call Picked-up (call has been picked up by another station after call was offered to the MDC CompuCALL party)
  - — PartyDropped (the CompuCALL party in an established 3-way call is dropped or released)
  - — PartyDroppedNoAnswer (the CompuCALL party is a consult party in a 3-way call and is dropped or released before answering)
- Used in dv-Call-Released-U RO

Note: dv-Call-Released-U is only sent for a call forwarded by the "CallForwardDon'tAnswer" feature.

5.3.29 set-Feature-Info

- Choice, 1 item; SetFeatureInformation
- Identifies the function of the request to be performed on behalf of a telephone set. The functions only apply to ACD agent positions.
- Used in the dv-Set-Feature RO

5.3.30 time

- Sequence, 3 items; SwitchTimeOfDay
- Contains the current switch time expressed in a twenty-four hour format and includes one or more of the following subparameters:
  - — Hour
  - — Minute
  - — Second
- Used in the dv-Resource-Query RO

5.4 Parameter types

This section defines the parameter types in Section 5.2. Each parameter type consists of one or more choices, sequences, or enumerated items. The allowed choices, sequences, or enumerated items depend on the individual parameters and are defined in Section 5.2.4.

- DNQueryFilter
- DNAssocInfo
- LineStateInfo

- MsgWaitType

5.4.1 AddressType

AddressType consists of the following four choices:

- positionID
- dialedDigits
- stationNumber
- q.931Address

5.4.2 AgtInfoType

AgtInfoType is a sequence with ACD Agent resource-related information. This parameter type provides status information on all the agents assigned in the ACD group associated with the ACD DN for which information is requested by the host. The sequence includes one or more of the following subparameters:

- agtsLgd
- agtsBsy
- agtsIdle
- agtsNR
- agtsCTQ

5.4.3 CallType (Change number)

CallType is an IMPLICIT ENUMERATED type.
The sequence includes one or more of the following subparameters:

- callTransferred
- callOverflowed
- callRedirected
- callForwarded
- callConsult
- callConference

5.4.4 deviceAddressType

- Applies to MDC/RES lines
- Sequence, 2 items
- Contains dynamic call-related information
- Lets us send a set of one or more values representing the calling device.
- The set includes the following parameters:

- — dialedDigits containing the DN in dial number delivery format
- — stationNumber containing directory number format
- — name containing the caller’s name

- Used in the following ROs:
  - — dv-Call-Offered-U
  - — dv-Call-Answered-U

5.4.4.1 dnAssociationInfo

DN Association Info is one of the following: associated to this session, associated to another session, not associated, or not allowed.

5.4.5 ForwardingPartyType

ForwardingPartyType is a sequence containing Call Forward-related information applicable to the call. This parameter type is a sequence with one or more of the following subparameters:

- firstFwdNumber
- firstFwdReason
- lastFwdNumber
- lastFwdReason

5.4.6 GrpInfoType

GrpInfoType consists of the following sequence:

- maxPQSize
- maxCTQSize

5.4.7 LineStateInfo

Line State Info is one of the following: idle, origination, dialling, talking, ringing, treatment, held, or lockedOut.

5.4.8 MessageWaitingType

Message Waiting Type is an enumerated type, 2 items. It specifies the type of Message Waiting. Currently the Message Waiting and Executive Message Waiting types are supported.

5.4.9 Message Waiting Indication

Identifies if the message waiting indication will be activated or deactivated.

5.4.10 Message Wait Info

Message Waiting Information contains the following optional parameters:

- Message Waiting Type: specifies the type of Message Waiting. The Message Waiting and Executive Message Waiting types are supported.

- Message Waiting Indication: identifies if the message waiting indication will be activated or deactivated.

5.4.11 QInfoType

QInfoType contains the queue information of an ACD group. This parameter type is a sequence with one or more of the following subparameters:

- acdDNPrio
- pQSize
- cTQSize
- outQSize
- queueInfo

5.4.12 QueryTypes

QueryType contains information on a specific ACD group pertaining to the group, agents and queue status. This parameter type is a sequence including one or more of the following subparameters:

- grpInfo
- qInfo
- agtInfo

5.4.13 QueuesByPriority

QueuesByPriority contains information on the number of calls queued in each of the ACD group's priority queues. This parameter type is a sequence with one or more of the following subparameters:

- Priority
- PQAtThisPrio

5.4.14 Q.931AddressType

Q.931AddressType is used in the AddressType parameter type and consists of the following sequence:

- digits
- numberTypeNumberPlan
- presentationScreeningIndicator

5.4.15 SwitchTimeOfDay

SwitchTimeOfDay indicates the current switch time expressed in a twenty-four-hour format. This parameter type is a sequence with one or more of the following subparameters:

- hours
- minutes

- seconds

5.5 Secondary parameters

This section defines the secondary parameters in Sections 5.1.0.and 5.2.0 and is listed alphabetically.

Add: Name, AssocQueryInfo, MsgWaitQueryInfo, LineStInfo(p.16, FN8189)

5.5.1 treatmentType

- There are 8 valid treatment types; AUDIO, RAN, Music, Silence, Ringback, Busy, Fastbusy, and Disconnect.
- The route range of RAN and Music types is 0 to 512.
- Any route within the route range results in the playing of the audio group.

5.5.2 acctCodeDigits

- ASCII String, range 1 to 14 ASCII digits.
- The range of RETURN-ERROR for this parameter is from 0, or 15 to 50. The REJECT range is a value of more than 50.
- Carries a string of IA5-encoded digits that correspond to the account code digits.
- Assigned by the operating company.

5.5.3 acdDNPrio

- Integer, range 0 to 3
- Contains static ACD DN-related information
- Assigned by the customer. Datafilled against the customer's ACD DN in the DMS.
- Specifies the priority associated with the ACD DN.
- Used in the QInfoType format

5.5.4 agentFeature

- Choice, 4 items; SetAgentFeature
- Identifies the function of the request to be performed on behalf of an ACD agent position. It has the following choices:
  - — logIn-ACD Agent Login function
  - — logOut-ACD Agent Logout function
  - — ready-ACD Agent Ready function
  - — notReady-ACD Agent Not Ready function

- The logIn functionality requires the following subparameters:
  - — loginID - identifies the ACD agent to be logged into the position at a PositionID. The parameter type is an implicit IA5STRING and must be four (4) digits in length with a range of 0001 to 9999.
  - — password - contains the password for agent login security. Type is an implicit IA5STRING and must be four (4) digits with a range of 0001 to 9999.
  - — The password parameter may be an application-specific option.
- Used in the dv-Set-Feature RO

5.5.5 agtsBsy

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Specifies the number of agents associated with the ACD DN currently in BUSY state (active on a call or reserved for a call) and unavailable for new calls.
- Used in the AgtInfoType format.

5.5.6 agtsCTQ

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Specifies the number of agents of an ACD group associated with the ACD DN who currently have calls queued for them in the Call Transfer Queue.
- Used in the AgtInfoType format

5.5.7 agtsIdle

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Specifies the number of agents associated with the ACD DN currently in IDLE state and available for new calls.
- Used in the AgtInfoType format

5.5.8 agtInfo

- Sequence, 5 items; AgtInfoType
- Contains dynamic ACD Agent resource-related information
- Provides information on all the agents assigned in the ACD group associated with the ACD DN for which information is requested by the host. It includes one or more of the following subparameters:
  - — agtsLgd - number of agents currently logged in
  - — agtsBsy - number of agents currently active on calls
  - — agtsIdle - number of agents currently available to answer calls

- — agtsNRT - number of agents currently in NotReady state
- — agtsCTQ - number of agents with calls currently queued for them in the Call Transfer Queue

- Used in the QueryTypes format

5.5.9 agtsLgd

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement
- The number of agents currently logged in an ACD group associated with the ACD DN for which information is requested by the host.
- Used in the AgtInfoType format.

5.5.10 agtsNR

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement.
- The number of agents of an ACD group associated with the ACD DN for which information is requested by the host currently in NotReady state and temporarily unavailable for new calls.
- Used in the AgtInfoType format

5.5.11 authCodeDigits

- ASCII String, range 0 to 14 ASCII digits
- RETURN-ERROR range is from 15 to 50. REJECT range is more than 50.
- Carries IA5-encoded digits corresponding to the authorization code digits
- Assigned by the operating company

5.5.12 callType

- Enumerated, 4 items
- Indicates whether the call has been:

- — transferred
- — overflowed
- — redirected
- — forwarded

5.5.13 ctqSize

- Integer; range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Contains dynamic ACD Queue-related information
- Specifies the number of queue slots occupied by calls in the ACD group's Call Transfer Queue

- Used in the QInfoType format

5.5.14 dialedDigits

- ASCII string, range 1 to 21 characters
- The range of RETURN-ERROR is from 22 to 50 characters. The REJECT range is a value of more than 50 characters.
- Carries a string of IA5-encoded digits corresponding to the dialed digits. The digits are translated and routed according to the Business Group (currently equivalent to DMS customer group) and Network Class of Service (NCOS) datafill set up for the originator of the call.

5.5.14.1 AdnAssociationInfo

- DN Association Info is one of the following:
  - associated to this session
  - associated to another session
  - not associated
  - not allowed

5.5.15 firstFwdNumber

- Choice, 4 items; AddressType format
- Contains static call-related information
- Encoded following the format of the Calling Party Number field in Q.931 messages (Section D of the Nortel Networks ISDN Basic Rate Access Interface Specification, NIS S208-6, Issue 1.0, April 1991). Contains a DN which may be a 10-digit North American Numbering Plan (NANP) directory number (or in the case of a call from a PBX, this may be the listed Directory Number (LDN) of the PBX), or a private number (4-digit extension number).
- Assigned by the operating company
- Used in the ForwardingPartyType format

5.5.16 firstFwdReason

- Enumerated, 4 items
- Indicates why a call has been forwarded by the first party (original called party) prior to terminating on the current CompuCALL party:
  - Unknown-The forwarding reason is not known or is unavailable
  - UserBusy-The first party for whom the call was intended, was busy, hence, the call was forwarded for this reason

- — NoReply-The first party for whom the call was intended, was not answering up to a specified time limit, hence, the call was forwarded for this reason
- — Unconditional -The first party for whom the call was intended unconditionally forwarded all calls

- Used in the ForwardingPartyType format

5.5.17 hours

- Integer, range 0 to 23
- Used in the SwitchTimeOfDay format.

5.5.18 grpInfo

- Sequence, 2 items; GrpInfoType
- Contains static ACD resource-related information on the assigned values for the ACD group with regard to the maximum Incoming Call Queue size and maximum Call Transfer Queue size. It includes one or more of the following subparameters:
  - — MaxPQSize - indicates the value for the maximum number of calls which may be queued in the Incoming Call Queue.
  - — MaxCTQSize - indicates the value for the maximum number of calls which may be queued in the Call Transfer Queue.
- Assigned by the customer and datafilled against the customer's ACD group in the DMS
- Used in the QueryTypes format

5.5.19 hostCallData

- Octet string, maximum 10 bytes
- Contains dynamic call-related information
- A "generic" subparameter to carry miscellaneous call-related host information; one host can use it to pass information to another host

Once a host supplies information in this parameter, the switch saves and sends it in any subsequent switch-host messages regarding the CallID with this parameter. This is overwritten in the switch when a host related to the call supplies it in a host-switch message. A HostCallData with null content is not rejected by the DMS-100 switch. The host can use it to reset a previous HostCallData string to null:

- Uniqueness is up to the customer
- Assigned by the customer and carried transparently by DMS

5.5.20 lastFwdNumber

- Choice, 4 items; AddressType format
- Contains static call-related information
- Encoded following the format of the Calling Party Number field in Q.931 messages (Section D of NT ISDN Basic Rate Access Interface Specification, NIS S208-6, Issue 1.0). Contains the calling line’s DN. This may be a 10-digit North American Numbering Plan (NANP) DN (or if a call from a PBX, this may be the listed DN (LDN) of the PBX), or a private number (4-digit extension number).
- Assigned by the operating company
- Used in the ForwardingPartyType format

5.5.21 lastFwdReason

- Enumerated, 4 items
- Indicates why a call has been forwarded by the last terminating party before terminating on the current CompuCALL party:
  - Unknown (forwarding reason is unknown or is unavailable)
  - UserBusy (last terminating party was busy so the call was forwarded)
  - NoReply(last terminating party was not answering up to a specified time limit so the call was forwarded)
  - Unconditional (last terminating party is unconditionally forwarding all calls)
- Used in the ForwardingPartyType format

5.5.21.1 LineStateInfo

Line State Info is one of the following:

- idle
- origination
- dialing
  - talking
  - ringing
  - treatment
  - held
  - lockedOut

5.5.22 localCallID

- Integer, from 1 to 4 Bytes, encoded in 2's complement

- Contains static call-related information
- Uniquely identifies an incoming call within a the DMS-100 switch
- The maximum number of possible values for LocalCallID is, 231-1, i.e., 2,147,483,647. The same LocalCallID is not likely to be used for two active calls simultaneously.
- Assigned by the switch on a per-call basis
- Used in the networkCallId parameter

5.5.23 makeCallType

- Enumerated, 3 items
- Includes the following available "types" of MakeCall:
  - "CallingAgentOnline"-the agent (operator) must be "on-line" (off-hook and idle) before the call proceeds.
  - "CallingAgentReserved"-the ACDNR feature on the agent (operator) set must be activated before the call proceeds.
  - "CallingAgentNotReserved"-the ACDNR feature on the agent (operator) set must NOT be activated before the call proceeds.
  - "CallingAgentBuzzBase"-a buzz will be applied to the base of the telephone set and the setup of the call will proceed to route the call to the destination.
  - "CallingAgentBeepHset"-a tone will be applied to the head set and the setup of the call will proceed to route the call to the destination.
- Used in the applicationData parameter.

5.5.24 maxCTQSize

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Contains static ACD group information
- Assigned by the customer and datafilled against the customer's ACD group in the DMS
- Specifies the value for the maximum number of calls queued in the Call Transfer Queue
- Used in the GrpInfoType format

5.5.25 maxPQSize

- Integer, range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Contains static ACD group information

- Assigned by the customer and datafilled against the customer's ACD group in the DMS
- Specifies the value for the maximum number of calls queued in the Incoming Call Queue
- Used in the GrpInfoType format

5.5.25.1 MessageWaitingInfo

Message Waiting Information is one of the following:

- Message Waiting Activated
- Message Waiting Deactivated
  - Executive Message Waiting Activated
  - Executive Message Waiting Deactivated
  - Not Allowed

5.5.26 minutes

- Integer, range 0 to 59
- Used in the SwitchTimeOfDay format

5.5.27 name

- b

5.5.28 networkNodeID

- Integer, range 0 to 32 767 (one or two bytes), encoded in 2's complement
- Contains static call-related information
- The reject range for this parameter is a value of more than 32 767
- Uniquely identifies a switch (CO switch or PBX) in a customer's network, in other words, within the set of public and private switches with which a customer's host applications can establish a CompuCALL session.
- Assigned by the customer and datafilled against the customer in DMS

5.5.29 numberTypeNumberPlan

- Octet string, 1 byte
- Used in the Q.931AddressType format
- Contains the Type of Number and the numbering plan identification adopted from the Q.931 messages
  - The type of number indicates the type of that number. The values are:
    - 000 unknown

- 001 international number
- 010 national number
- 100 subscriber number

— Numbering plan identification indicates the number plan for that address. The values follow:

- 0000 unknown
- 0001 telephone/ISDN numbering plan
- 1001 private numbering plan

Note: The associated Q.931 field is encoded as a single variable length data element in these ROs and parsed by the host.

Bits 87654321	
0/1Type of NumberNumbering Plan Ext.Identification	Octet 1
1Present.000Screening Ext.IndicatorIndicator	2
	3a*
= 0 Digits = Spare (IA5 characters)	4* etc.
* This octet may be omitted.	
The number digit in octet 4 precedes the number digit in octet 5, etc.	

5.5.30 outqSize

Integer; range 0 to 32 767 (one or two bytes) encoded in 2's complement

- Contains dynamic ACD Queue-related information
- Specifies the number of queue slots occupied by calls in the ACD group's Overflow Out Queues
- Used in the QInfoType format

5.5.31 password

- IA5string, 4 bytes
- Specifies the password or security in the ACD agent position LogIn function
- The value of the password ranges from 0001 to 9999 decimal.

- Used in the LogInInfoType format

5.5.32 positionID

- Integer, range 0 to 9999 (one or two byte), encoded in 2's complement
- The REJECT range for this parameter is a value of more than 65 535.
- Uniquely identifies an ACD agent position for a customer in a switch
- Assigned by the operating company

5.5.33 pqSize

- Integer; range 0 to 32 767 (one or two bytes) encoded in 2's complement
- Contains dynamic ACD Queue-related information
- Specifies the number of queue slots occupied by calls in the ACD group's Incoming Call, Call Transfer and Overflow Out Queues.
- Used in the QInfoType format.

5.5.34 presentationScreeningIndicator

- Octet string, 1 byte.
- Used in the Q.931AddressType format.
- Contains the presentation indicator and screening indicator.
- The presentation indicator specifies whether the digits are available or suppressed by the subscriber. The values follow:
  - — 00 presentation allowed
  - — 01 presentation restricted
- The screening indicator shows if the digits are provided by the user or network. If user-provided, the digits are verified in the originating switch and indicated pass or fail. If the user-provided digits failed, they are sent with the network-provided digits. The values follow:
  - — 00 user-provided, not screened
  - — 01 user-provided, verified and passed
  - — 01 user-provided, verified and failed
  - — 11 network-provided

Note: The associated Q.931 field is encoded as a single variable length data element in these ROs and parsed by the host.

Bits 87654321	
0/1Type of NumberNumbering Plan Ext.Identification	Octet 1
1Present.000Screening Ext.IndicatorIndicator	2
	3a*
= 0 Digits = Spare (IA5 characters)	4* etc.
* This octet may be omitted.	
The number digit in octet 4 precedes the number digit in octet 5, etc.	

5.5.35 prevApplicationID

Integer, range 1 to 32 767 (one or two byte), encoded in 2's complement

- Contains dynamic CompuCALL call-related information
- Contains the ApplicationID for the Associated environment and is used if the CompuCALL party transferring, overflowing, redirecting, and forwarding the call belongs to the Associated environment. Otherwise, this parameter is not used.
- Used within the callHistoryInfo parameter

5.5.36 priority

- Integer; range 0 to 3
- Contains static ACD queue-related information
- Specifies the priority of the ACD Queue
- Assigned by the customer and datafilled against the customer's ACD queues in the DMS
- Used in the QueuesByPriority format

5.5.37 pQAtThisPrio

- Integer; range 32 767 (one or two bytes) encoded in 2's complement
- Contains dynamic ACD queue-related information

- Specifies the number of queue slots occupied by calls in the ACD queue. This value includes information on the Call Transfer Queue and Overflow Out Queue.
- Used in the QueuesByPriority format

5.5.38 queueInfo

- Sequence, 2 items; QueuesByPriority
- Contains dynamic ACD queue-related information

Contains current queue status information for the ACD group associated with the specified ACD DN and includes one or more subparameters:

Priority of the queue

PQAtThisPrio - the number of calls queued at the priority

- Used in the QInfoType format.

5.5.39 qInfo

- Sequence, 5 items; QInfoType
- Contains dynamic ACD queue-related information
- Contains current queue status information for the ACD group associated with the ACDDN for which information was requested by the host. It includes one or more of the following subparameters:
  - — ACDDNPrio specifies the priority associated with the ACDDN for which information was requested.
  - — PQSizespecifies the number of queue slots occupied by calls in the ACD group's Incoming-Call queue. Also includes calls queued in the Call Transfer.
  - — CTQSizespecifies the number of queue slots occupied by calls in the ACD group's Call Transfer call queue.
  - — OutQSizespecifies the number of queue slots occupied by calls in the ACD group's Overflow Out queue.
  - — QueueInfoprovides the queue information on the number of calls queued in each of the ACD group's Priority Queues.
- Used in the QueryTypes format.

5.5.40 q.931Address

- Sequence, 3 items; Q.931AddressType
- Used in the AddressType format

- Contains the address information which uniquely identifies a DN, It follows the format of the Q.931 Calling Party Number field.
- The address information includes one or more of the following subparameters:
  - — digits
  - — NumberTypeNumberPlan
  - — PresentationScreeningIndicator
- The following values are set if applicable.
  - — type of number
  - — number plan identification
  - — presentation indicator
  - — Screening Indicator

Note: The q.931 Address follows the format of the Q.931 Calling Party Number Field and is encoded as a single variable length data element in these ROs and parsed by the host.

Bits 87654321	
0/1Type of NumberNumbering Plan Ext.Identification	Octet 1
1Present.000Screening Ext.IndicatorIndicator	2
	3a*
= 0 Digits = Spare (IA5 characters)	4* etc.
* This octet may be omitted.	
The number digit in octet 4 precedes the number digit in octet 5, etc.	

5.5.41 seconds

- Integer, range 0 to 59
- Used in the SwitchTimeOfDay format

5.5.42 stationNumber

- Octet string, 10 bytes

- Contains digits uniquely designating a station line or the Primary DN of an ACD Group in the CompuCALL environment. It uniquely identifies the line with which the station associates.
- Assigned by the operating company

5.5.43 walkawayReason

- Implicit IA5string, 3 bytes
- Identifies the reason for an ACD agent position to enter the Not Ready state
- The value ranges from 000 to 999
- The value is entered by the ACD agent
- Used in the following formats:
  - — NotReadyPrivateDataType (dv-Set-Feature RO for Agent Not Ready function)
  - — NotReadyPrivateData (dv-Agent-Not-Ready-U RO)

6.0 Call walkthrough procedures

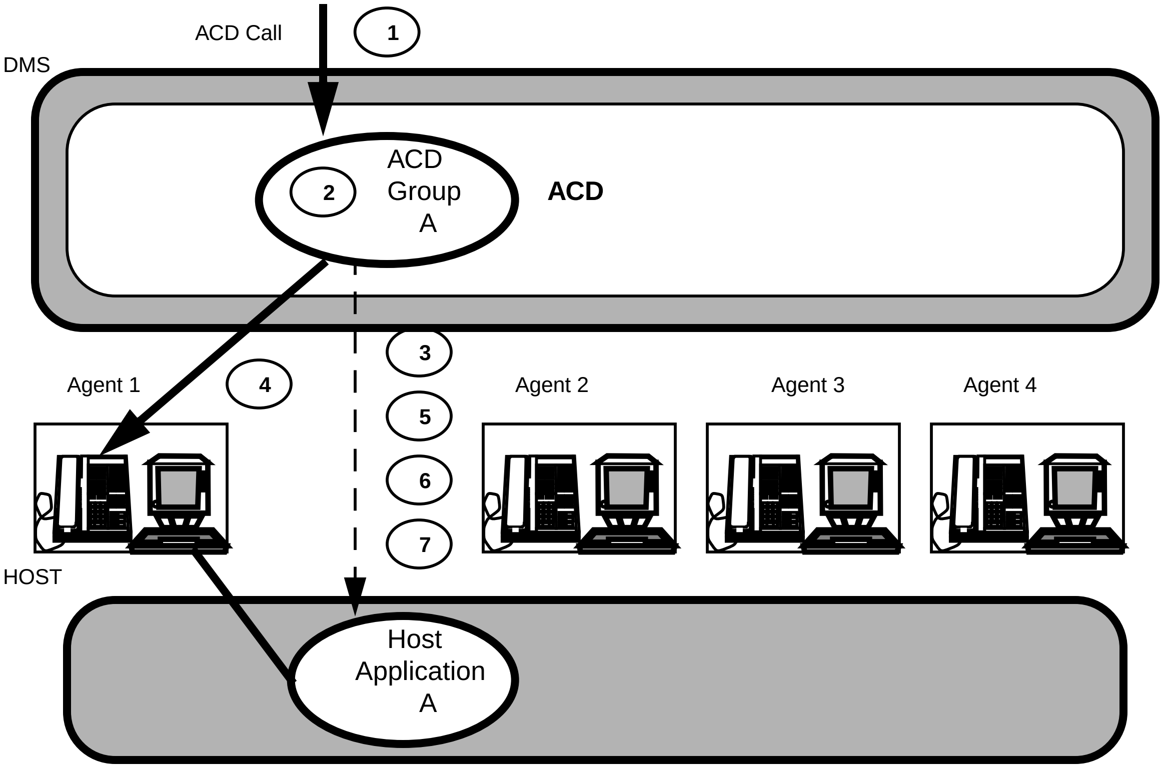
6.1 Call walkthrough procedures

Following are typical Meridian CompuCALL scenarios and associated messages generated by the switch and host. This is not an extensive list of Meridian CompuCALL message sequences, but a reference for the more common or exceptional scenarios.

Each host application is assumed to have an ACD group or MDC line and its own associated environment with which it has an established session. For example, host application A has an associated environment with ACD group A, ACD agent 1, and MDC station 1.

6.2 Meridian ACD CompuCALL options: no call redirection

6.2.1 Normal call



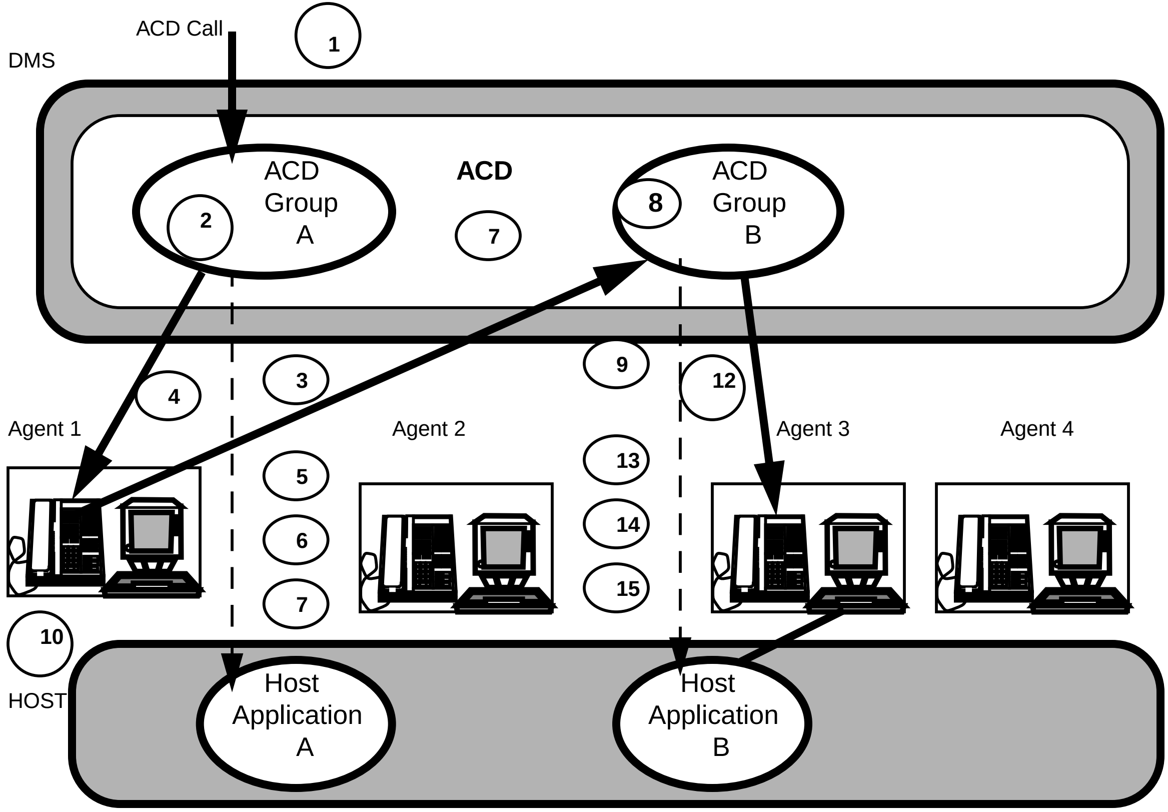
1) ACD call received by switch
2) Call queued for agent (if necessary)
3) dv-Call-Queued-U sent to Host Application A (if call queued)
   •NetworkCallID - new
   •ACDDN - primary or supplementary DN of ACD Group A
   •OrigCallingNumber - if available
   •OrigChargeNumber - if available
   •ACDGroup - StationNumber = primary DN of ACD Group A
   --> Host Application A retrieves caller information/prepares screen display
4) Agent 1 selected - call offered
5) dv-Call-Offered-U sent to Host Application A
   •same parameters as in 3, plus
   •DeviceID - PositionID = Agent 1
   --> Host Application A displays call record on Agent 1's terminal
6) dv-Call-Answered-U sent to Host Application A when agent 1 answers call
   •same parameters as in 5

6) (Cont'd)
   --> Host Application A is confirmed that Agent 1 has answered call
7) dv-Call-Released-U sent to Host Application A when call clears
   •NetworkCallID - same as in 3
   •ReleaseReason = CallCleared
   •ACDDN - same as in 3
   •ACDGroup - same as in 3
   •DeviceID - same as in 5
   --> Host Application A 'closes' call record (immediately after agent follow-up work)

Note - If call abandoned:

a) before agent selected (i.e., while in queue), host receives
   dv-Call-Queued-U and
   dv-Call-Released-U (ReleaseReason = CallAbandoned)
b) after agent selected (e.g. while agent's phone ringing), host receives
   dv-Call-Queued-U,
   dv-Call-Offered-U, and
   dv-Call-Released-U (ReleaseReason = CallAbandoned)

6.2.2 Single call transfer



1) - 6) Same as Normal Call in 6.2.1

7) Agent 1 initiates consult call to ACD Group B

8) Call queued for agent

9) dv-Call-Queued-U sent to Host Application B
   - •NetworkCallID - same as in 3
   - •ACDDN - primary or supplementary DN of ACD Group B
   - •OrigCallingNumber - same as in 3
   - •OrigChargeNumber - same as in 3
   - •CallHistoryInfo
     - -CallType = CallTransferred
     - -OrigInboundDN = primary or supplementary DN of ACD Gp. A
     - -PrevApplicationID = Host Application A
   - •ACDGroup - StationNumber = primary DN of ACD Group B
   - --> Host Application B obtains call record from Host Application A and prepares screen display for agent (Note - All scenarios assume inter-host application communication)

10) Agent 1 completes the call transfer and disconnects (i.e., blind transfer)

11) dv-Call-Released-U sent to Host Application A
   - •NetworkCallID - same as in 3
   - •ReleaseReason = CallTransferred
   - •ACDDN - same as in 3
   - •ACDGroup - same as in 3
   - •DeviceID - same as in 5
   - --> provides 'closure' regarding agent 1's involvement with the call to Host Application A

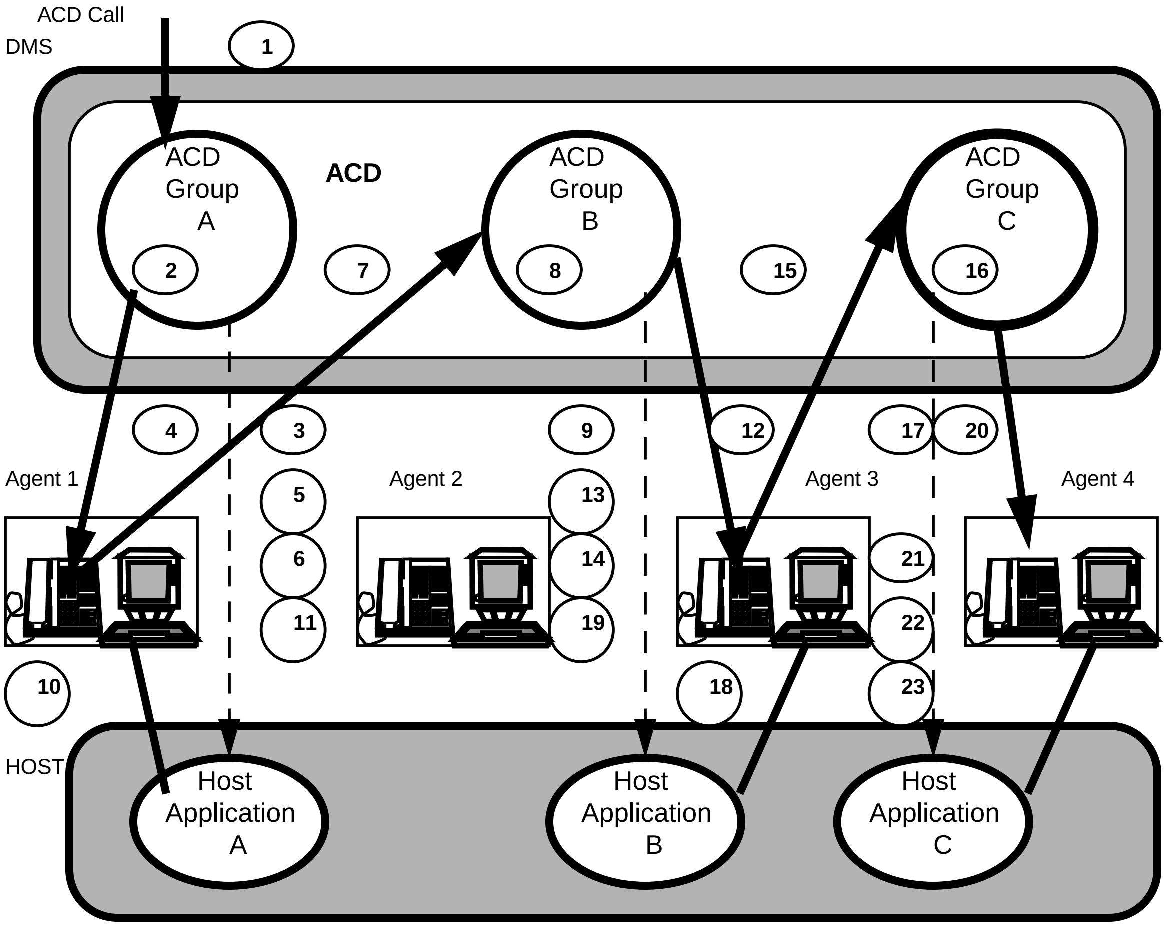
12) Agent 3 selected - call offered

13) dv-Call-Offered-U sent to Host Application B
   - •same parameters as in 9, plus
   - •DeviceID - PositionID = Agent 3
   - --> Host Application B displays call record on Agent 3's terminal

14) dv-Call-Answered-U sent to Host Application B when agent 3 answers call
   - • same parameters as in 13
   - --> Host Application B is confirmed that Agent 3 has answered call

15) dv-Call-Released-U sent to Host Application B when call clears
   - •NetworkCallID - same as in 3
   - •ReleaseReason = CallCleared
   - •ACDDN - same as in 9
   - •ACDGroup - same as in 9
   - •DeviceID - same as in 13
   - --> Host Application B 'closes' call record

6.2.3 Multiple call transfer transfer with consultation



1) - 14) Same as single call transfer in 6.2.2

15) Agent 3 initiates consult call to ACD Group C using the voice set

16) Call queued for agent

17) dv-Call-Queued-U sent to Host Application C
- •NetworkCallID - same as in 3
- •ACDDN - primary or supplementary DN of ACD Group C
- •OrigCallingNumber - same as in 3
- •OrigChargeNumber - same as in 3
- •CallHistoryInfo
  - -CallType = CallTransferred
  - -OrigInboundDN = primary or supplementary DN of ACD Gp. A
  - -PrevApplicationID = Host Application B
- •ACDGroup - StationNumber = primary DN of ACD Group C
- --> Host Application C obtains current call record from Host Application B and prepares screen display for agent

18) Agent 3 completes the call transfer and disconnects (i.e., blind transfer)

19) dv-Call-Released-U sent to Host Application B
- •NetworkCallID - same as in 3
- •ReleaseReason = CallTransferred

19) (Cont'd)
- •ACDDN - same as in 9
- •ACDGroup - same as in 9
- •DeviceID - same as in 13
- --> provides 'closure' regarding agent 1's involvement with the call to Host Application B

20) Agent 4 selected - call offered

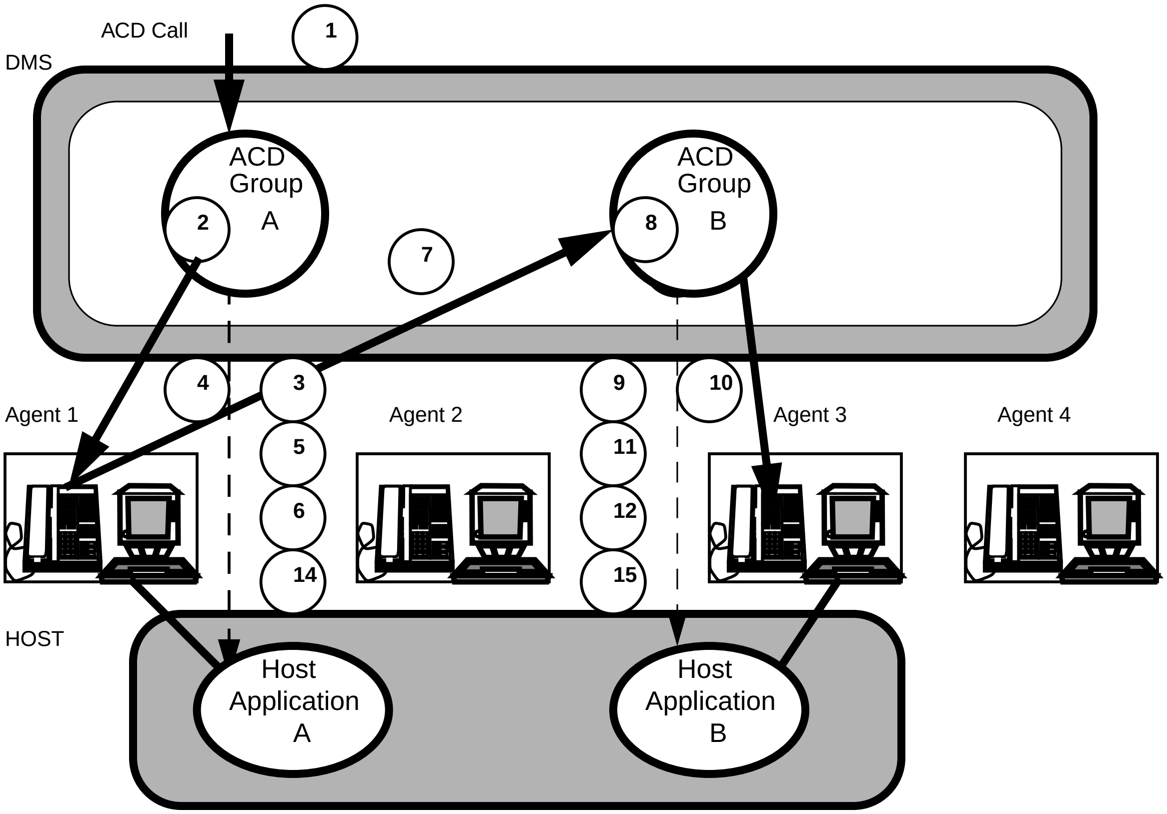
21) dv-Call-Offered-U sent to Host Application C
- •same parameters as in 17, plus
- •DeviceID - PositionID = Agent 4
- --> Host Application C displays call record on Agent 4's terminal

22) dv-Call-Answered-U sent to Host Application C when Agent 4 answers call
- • same parameters as in 21
- --> Host Application C is confirmed that Agent 4 has answered call

23) dv-Call-Released-U sent to Host Application C when call clears
- •NetworkCallID - same as in 3
- •ReleaseReason = CallCleared
- •ACDDN - same as in 17
- •ACDGroup - same as in 17
- •DeviceID - same as in 21
- --> Host Application C 'closes' call record

Note: If no Host Application B:
- Steps 9,13,14 & 19 omitted
- No PrevApplicationID in Step 17

6.2.4 Transfer with consultation



1) - 9) Same as single call transfer in 6.2.2
10) Agent 3 selected - call offered
11) dv-Call-Offered-U sent to Host Application B
- same parameters as in 9, plus
- DeviceID - PositionID = Agent 3

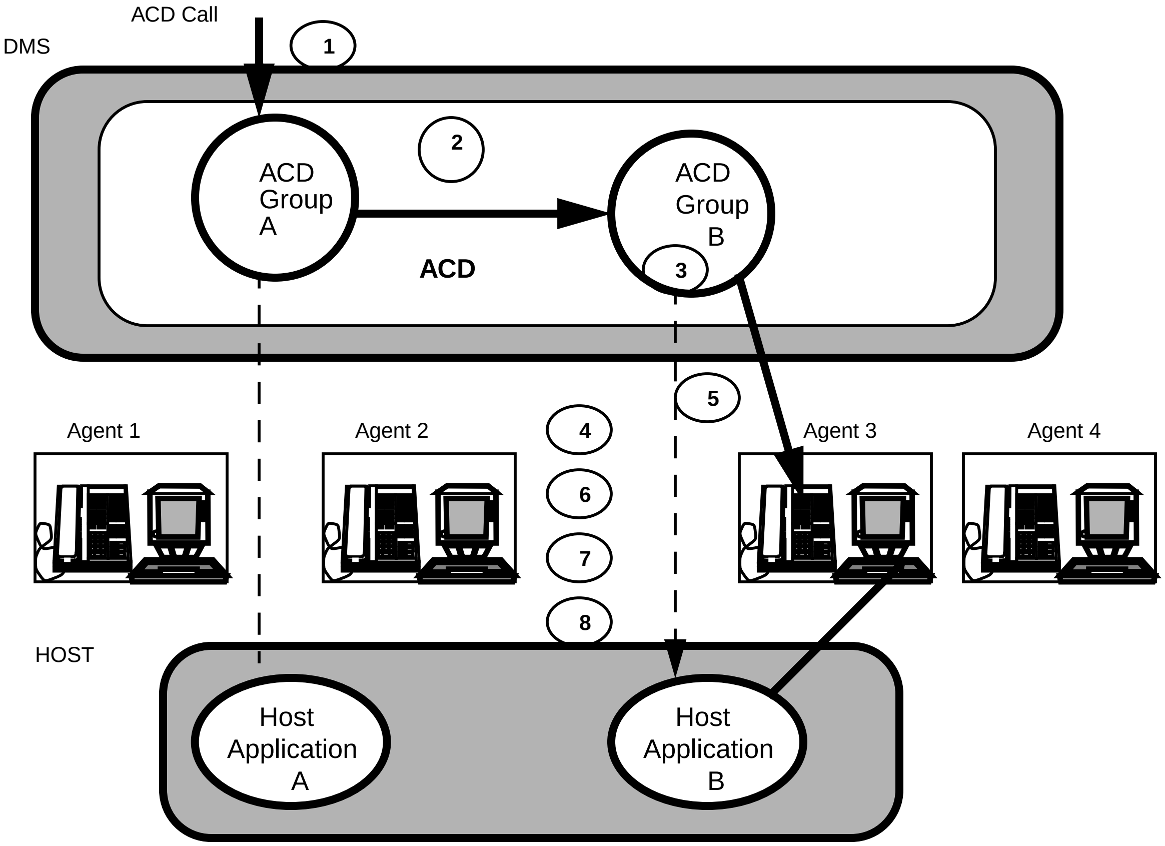
--> Host Application B displays call record on Agent 3's terminal (e.g. shared screen)
12) dv-Call-Answered-U sent to Host Application B when Agent 3 answers call
- same parameters as in 11

--> Host Application B is confirmed that Agent 3 has answered call
13) Agent 1 consults with Agent 3, with caller on hold; after private consultation, Agent 1 completes the call transfer and disconnects
14) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- ACDDN - same as in 3
- ACDGroup - same as in 3
- DeviceID - same as in 5

--> provides 'closure' regarding agent 1's involvement with the call to Host Application A
15) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- ACDDN - same as in 9
- ACDGroup - same as in 9
- DeviceID - same as in 11

--> Host Application B 'closes' call record

6.2.5 Single call overflow (immediate)

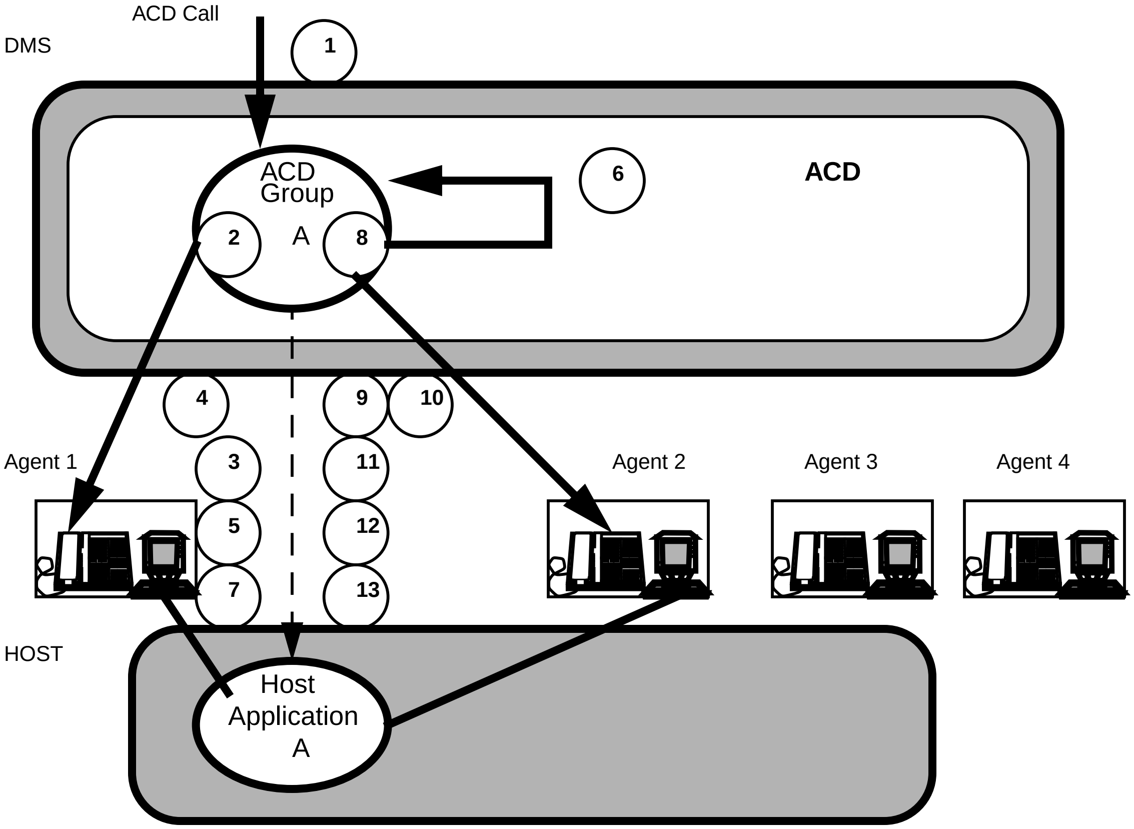


1) ACD call received by switch
2) Call overflows to ACD Group B (Note: This is immediate overflow, before the call is queued for ACD Group A)
3) Call queued for agent (if necessary)
4) dv-Call-Queued-U sent to Host Application B (if call queued)
   - NetworkCallID - new
   - ACDDN - primary or supplementary DN of ACD Group B
   - OrigCallingNumber - if available
   - OrigChargeNumber - if available
   - CallHistoryInfo
     - CallType = CallOverflowed
     - OrigInboundDN = primary or supplementary DN of ACD Gp. A
     - PrevApplicationID = Host Application A
   - ACDGroup - StationNumber = primary DN of ACD Group B
5) Agent 3 selected - call offered
6) dv-Call-Offered-U sent to Host Application B
   - same parameters as in 4, plus
   - DeviceID - PositionID = Agent 3

   --> Host Application B displays call record on Agent 3's terminal
7) dv-Call-Answered-U sent to Host Application B when Agent 3 answers call
   - same parameters as in 6
   - Host Application B is confirmed that Agent 3 has answered call
8) dv-Call-Released-U sent to Host Application B when call clears
   - NetworkCallID - same as in 4
   - ReleaseReason = CallCleared
   - ACDDN - same as in 4
   - ACDGroup - same as in 4
   - DeviceID - same as in 6

   --> Host Application B 'closes' call record

6.2.6 Ring threshold overflow



1) - 4) Same as Normal Call in 6.2.1

5) dv-Call-Offered-U sent to Host Application A
- same parameters as in 3, plus
- DeviceID - PositionID = Agent 1

6) Ring Threshold feature overflows call back into ACD Group A

Note: Example is where agent is doing follow-up work on previous call, forgets to press Not-Ready key, and Ring Threshold timer expires when agent doesn't answer new call. Same procedure applies where agent presses Not-Ready or Make-Set-Busy key while being offered call.

7) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallOverflowed
- ACDDN - same as in 3
- ACDGroup - same as in 3
- DeviceID - same as in 5

--> Host Appl. A retains call record

8) Call queued for agent (if necessary)

9) dv-Call-Queued-U sent to Host Application A (if call queued)
- NetworkCallID - same as in 3
- ACDDN - primary or supplementary DN of ACD Group A
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3

9) (Cont'd)
- CallHistoryInfo
  - CallType = CallOverflowed
  - OrigInboundDN = primary or supplementary DN of ACD Group A
  - PrevApplicationID = Host Appl.A
- ACDGroup - StationNumber = primary DN of ACD Group A

--> Host Appl. A retrieves previous call record/prepares screen display

10) Agent 2 selected - call offered

11) dv-Call-Offered-U sent to Host Application A
- same parameters as in 9, plus
- DeviceID - PositionID = Agent 2

--> Host Application A displays call record on Agent 2's terminal

12) dv-Call-Answered-U sent to Host Application A
- same parameters as in 11

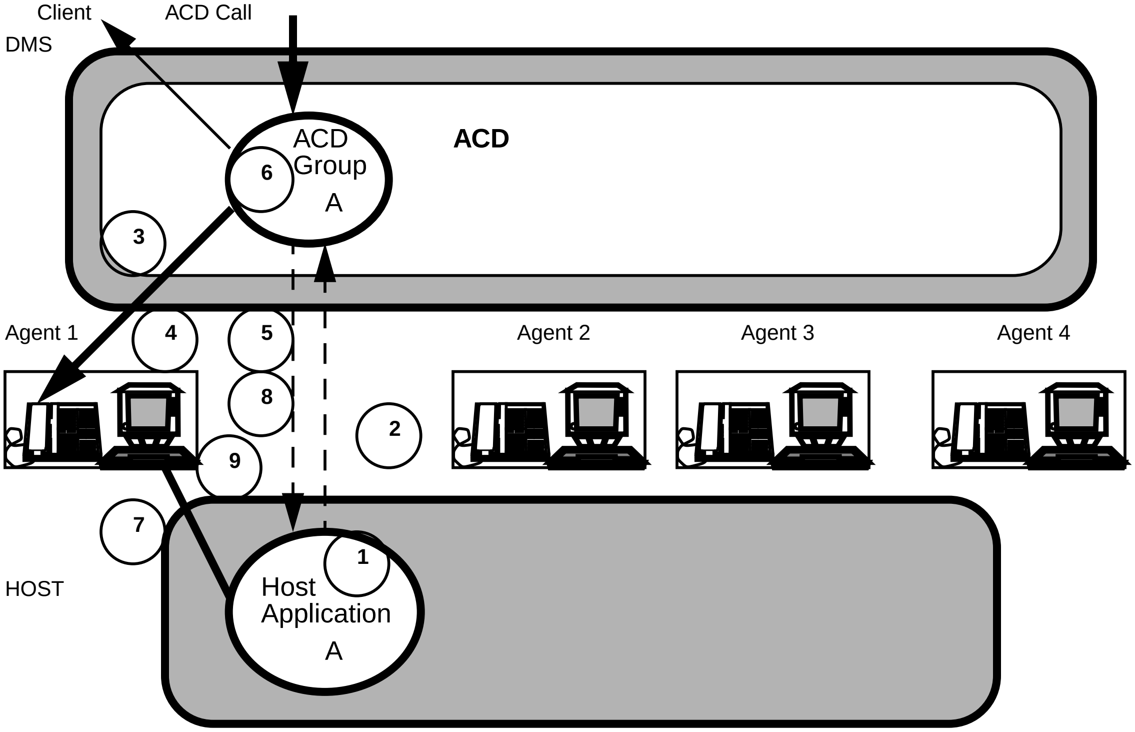
--> Host Application A is confirmed that Agent 2 has answered call

13) dv-Call-Released-U sent to Host Application A when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- ACDDN - same as in 9
- ACDGroup - same as in 9
- DeviceID - same as in 11

--> Host Application A 'closes' call record

6.3 Meridian ACD CompuCALL options: Third Party Call Control

6.3.1 MakeCall using the host



Procedure 1

1) An outgoing call is initiated on behalf of an agent - initiated by agent or host from 'call list'
2) dv-Make-Call sent to the switch.
- OrigAddress - PositionID = Agent 1
- DestAddress - DialedDigits = Client DN
- ApplicationData
  - MakeCallType = CallingAgentOnline
  - AuthCodeDigits - new (optional)
  - AcctCodeDigits - new (optional)

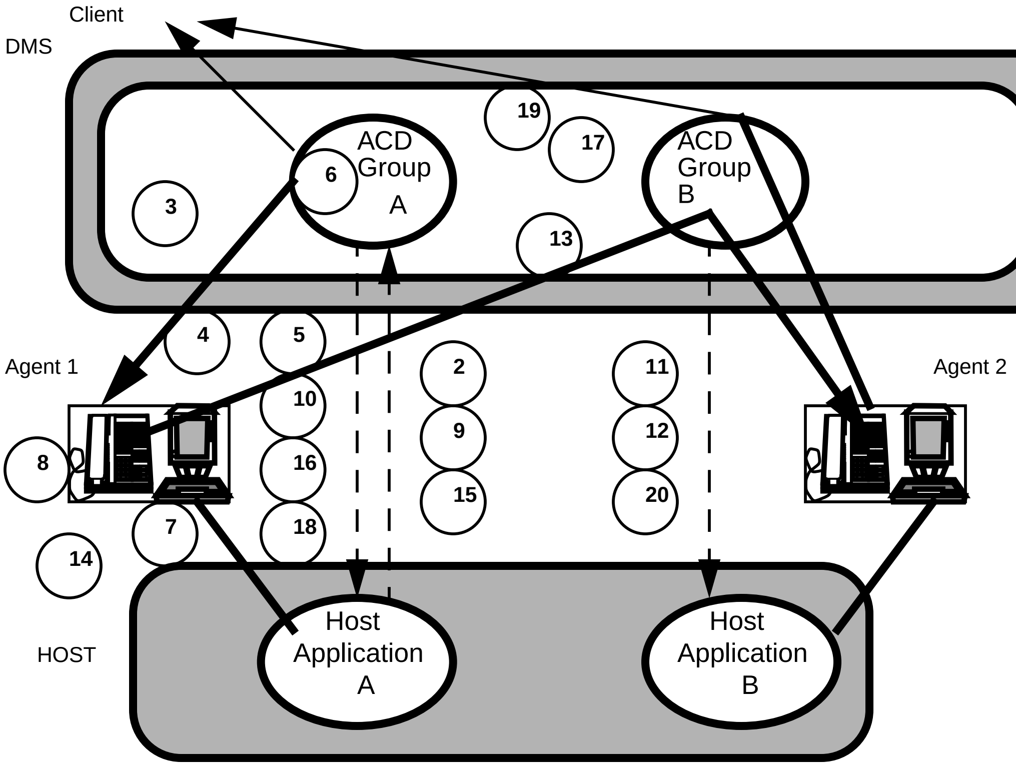
3) Switch verifies request and authorization code, and alerts agent 1
4) Agent 1 accepts call (goes off-hook)
5) RETURN-RESULT sent to Host Application A (acknowledges 2)
- NetworkCallID - new

6) The switch attempts call set- up (with agent connected)
7) Host displays client's file on Agent 1's terminal
8) dv-Call-Released-U sent to Host Application A when call clears
- NetworkCallID = same as in 5
- ReleaseReason = CallCleared
- ACDDN - unavailable
- ACDGroup - StationNumber - unavailable
- DeviceID - PositionID = Agent 1

--> Host Application A 'closes' call record
9) Agent 1 may enter the result of call attempt (success or failure/reason) to host
--> Host updates customer's file and /or 'call list'

Note: dv-Call-Answered-U doesn't apply when an outbound call initiated by dv-Make-Call is answered although the latter requires acceptance of the call by the agent.

6.3.2 MakeCall with transfer using the host



1) - 7) Same as MakeCall using the host (6.3.1)

8) Agent 1 (controller) decides to add a consult party (Agent 3) to the call

9) dv-Add-Party sent to the switch; host requests a consult call to be established
- •AddPartyType - AddConsultParty or AddConsultforConf* (puts existing call on hold)
- •OrigAddress - PositionID = Agent 1
- •DestAddress - DialedDigits = Agent 3 (consult party)

Note: Consult party does not have to be an ACD agent.

10) RETURN-RESULT sent to Host Application A (indicating valid request)
- •NetworkCallID - same as 5

11) dv-Call-Offered-U sent to Host B
- NetworkCallID - same as in 5
- ACDDN - primary or supplementary DN of ACD Group B
- OrigCalling Number - Agent 1's Incalls Key DN
- OrigCharge Number - not provided
- CallHistoryInfo
  - CallType = CallTransferred
  - OrigInboundDN = Not provided
  - PrevApplicationID = Host Appl. A
- ACDGroup - StationNumber = primary DN of ACD Group B
- DeviceID - PositionID = Agent 3

12) dv-Call-Answered-U sent to Host Application B when Agent 3 answers call
- • same parameters as in 11

--> Host displays client record on Agent 3's terminal

13) The controller (Agent 1) is connected to the consult party (Agent 3) but cannot talk to the held party (client)

--> The controller's and consult party's displays are updated with call info if they have MBS sets

Note: Scenario assumes idle consult party (Agent 3); therefore, no queuing is needed

14) The controller (Agent 1) initiates call transfer

15) dv-Transfer-Party sent to the switch; Host requests call transfer
- •OrigAddress - PositionID = Agent 1

16) RETURN-RESULT sent to Host Application A (indicating valid request)
- •NetworkCallID - same as in 5

17) The controller is dropped and the held party is transferred to the consult party

--> The consult party's display is updated with call info if agent has MBS set

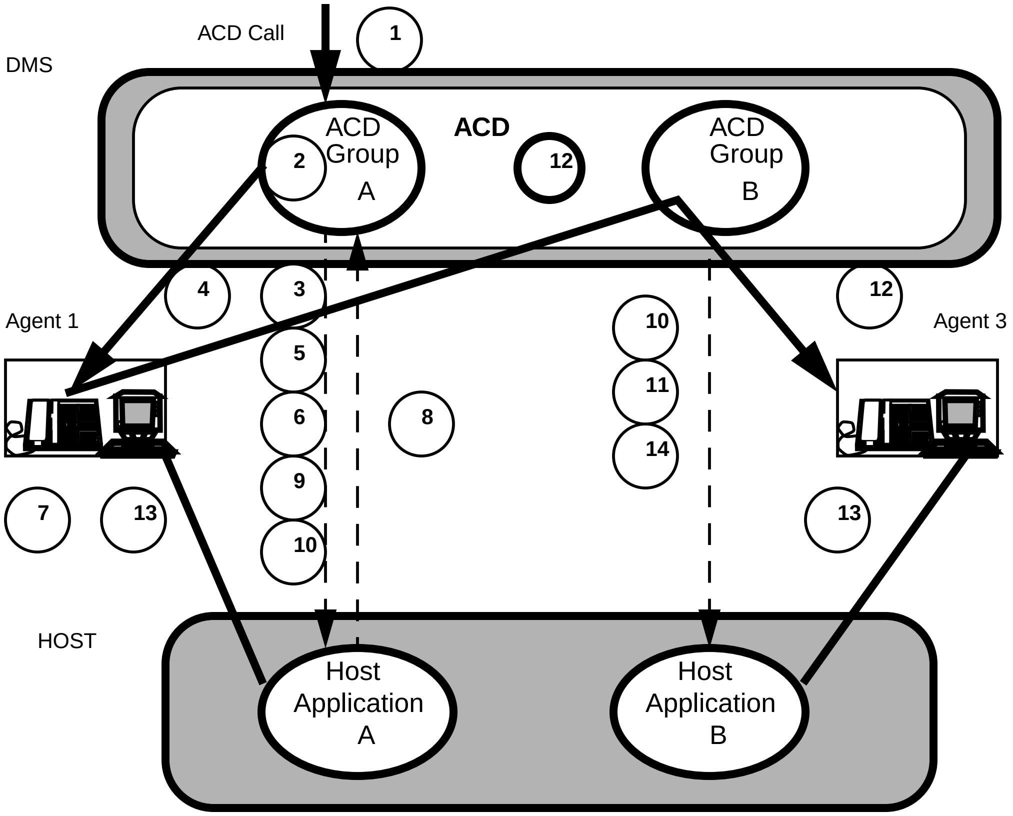
18) dv-Call-Released-U sent to Host A
- •NetworkCallID - same as in 5
- •ReleaseReason = CallTransferred
- •ACDDN - not available
- •ACDGroup - not available
- •DeviceID - PositionID = Agent 1

19) Agent 3 talks to client

20) dv-Call-Released-U sent to Host Application B when call clears
- •NetworkCallID = same as in 5
- •ReleaseReason = CallCleared
- •ACDDN = primary or supplementary DN of ACD Group B
- •ACDGroup - StationNumber = primary DN of ACD Group B
- •DeviceID - PositionID = Agent 3

--> Host Application B 'closes' call record

6.3.3 Consultation using the host (use of RLS key)



1) - 6) Same as in 6.2.1

7) Agent 1 (controller) decides to add a consult party (Agent 3) to the call

8) dv-Add-Party sent to the switch; host requests a consult call to be established

- AddPartyType - AddConsultParty or AddConsultforConf* (puts call on hold)
- OrigAddress - PositionID = Agent 1
- DestAddress - DialedDigits = Agent 3 (Consult Party)

Note: Consult party does not have to be an ACD agent.

9) RETURN-RESULT sent to Host Application A (indicating valid request)

- NetworkCallID - same as 3

10) dv-Call-Offered-U sent to Host B

- NetworkCallID - same as in 3
- ACDDN - primary or supplementary DN of ACD Group B
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo

- CallType = CallTransferred
- OrigInboundDN = primary or supplementary DN of ACD Group A
- PrevApplicationID = Host Application A

• ACDGroup - StationNumber = primary DN of ACD Group B

- DeviceID - PositionID = Agent 3

11) dv-Call-Answered-U sent to Host Application B when Agent 3 answers call

- same parameters as in 10)

* See notes on p.4-6.22.

11) (Cont'd)

--> Host displays client record on Agent 3's terminal

12) The controller (Agent 1) is connected to the consult party (Agent 3) but can’t talk to the held party

--> controller's and consult party's displays are updated with call info if they have MBS sets

Note: Scenario assumes idle consult party (Agent 3); therefore, no transfer queuing is needed

13) The consult party (agent 3) or the controller (agent 1) releases the consult call using RLS key

--> Controller (agent 1) is rerung for held party & answers

Note: No messages are sent for Controller (agent 1)

14) dv-Call-Released-U sent to Host Application B

- NetworkCallID - same as in 3
- ReleaseReason = PartyDropped
- ACDDN = primary or supplementary DN of ACD Group B
- ACDGroup - StationNumber = primary DN of ACD Group B
- DeviceID - PositionID = Agent 3

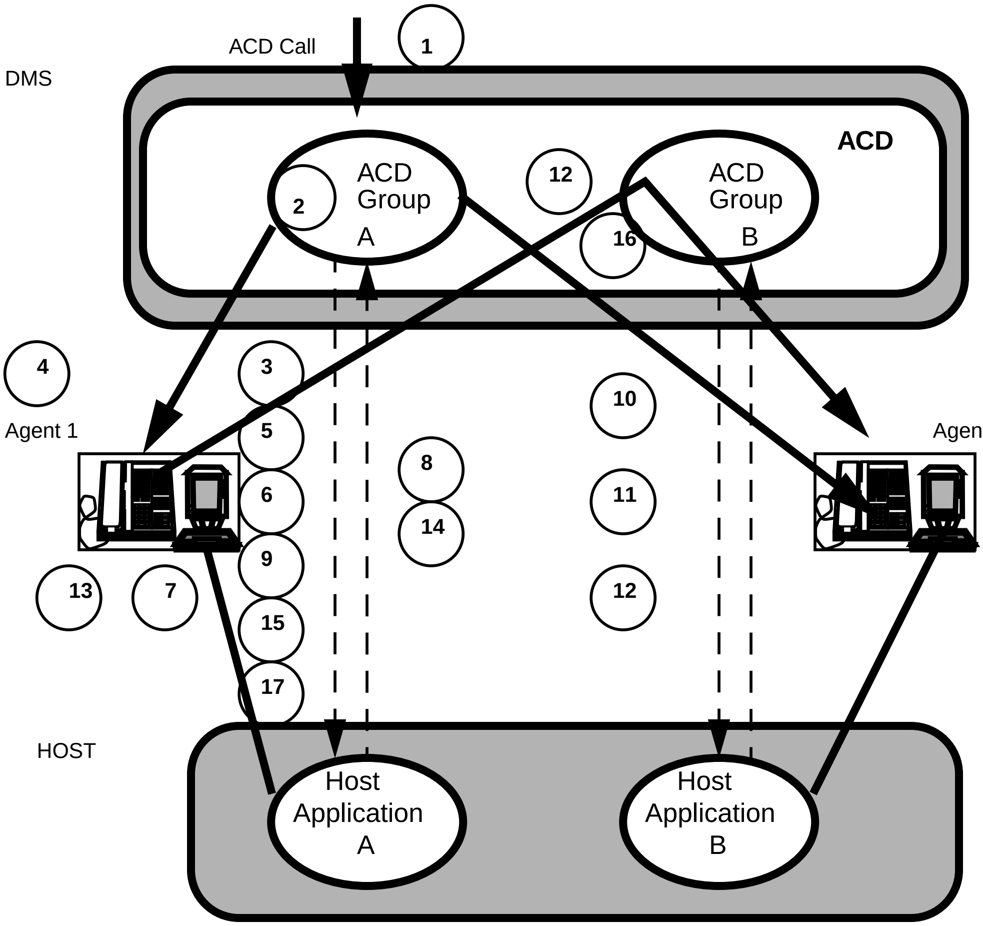
--> Controller (agent 1) is reconnected to the held party. Controller's and held party's displays are updated with call info if they have MBS sets.

15) dv-Call-Released-U sent to Host Application A when call clears

- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- ACDDN = primary or supplementary DN of ACD Group A
- ACDGroup - StationNumber = primary DN of ACD Group A
- DeviceID - PositionID = Agent 1

--> Host Application A 'closes' call record

6.3.4 Transfer with consultation using the host



1) - 12) Same as Consultation using the Host (use of RLS key) in 6.3.3

13) The controller initiates call transfer

14) dv-Transfer-Party sent to the switch; Host requests call transfer
- OrigAddress - PositionID = Agent 1

15) RETURN-RESULT sent to Host Application A (indicating valid request)
- NetworkCallID - same as in 3

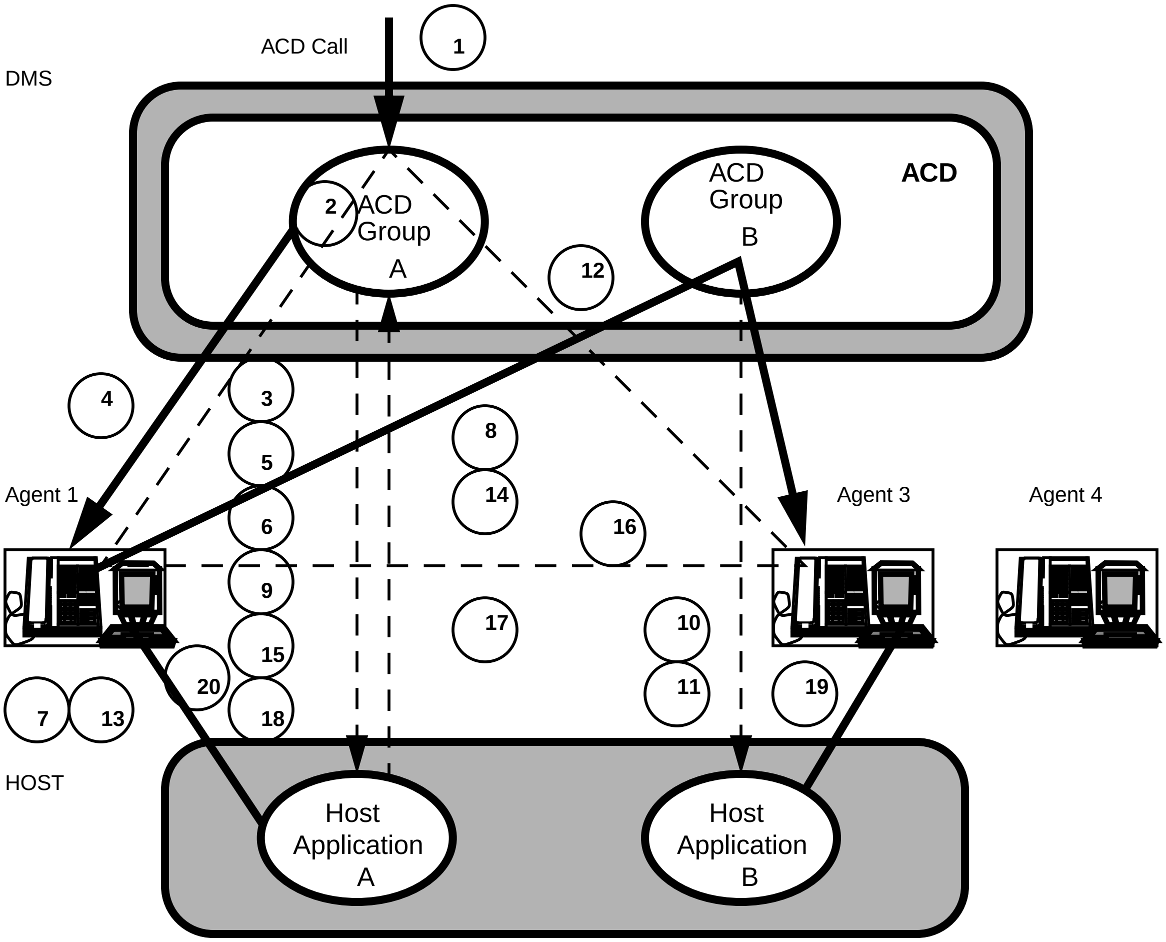
16) The controller (Agent 1) is dropped and the held party (client) is transferred to the consult party (Agent 3)
--> The consult party's display is updated with call info if agent has MBS set

17) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- ACDDN = primary or supplementary DN of ACD Group A
- ACDGroup - StationNumber = primary DN of ACD Group A
- DeviceID - PositionID = Agent 1

18) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- ACDDN = primary or supplementary DN of ACD Group B
- ACDGroup - StationNumber = primary DN of ACD Group B
- DeviceID - PositionID = Agent 3

--> Host Application B "closes" call record

6.3.5 Conference with consultation using the host



1) - 12) Same as Consultation using the host (use of RLS key) in 6.3.3

13) The controller initiates call conference

14) dv-Conference-Party sent to the switch; Host requests call conference
- OrigAddress - PositionID = Agent 1

15) RETURN-RESULT sent to Host Application A (indicating valid request) • NetworkCallID - same as in 3

16) The controller, held party, and consult party are all connected and active on a conference call.
--> The controller's and consult party's displays are updated with call status if they have MBS sets

17) dv-Drop-Party sent to switch to drop consult party
- DropPartyType - DropConsultParty
- OrigAddress - PositionID = Agent 1

18) RETURN-RESULT sent to Host Application A (indicating valid request) • NetworkCallID - same as in 3

19) dv-Call-Released-U sent to Host Application B
- NetworkCallID - same as in 3

19) (Cont'd)
- ReleaseReason = PartyDropped
- ACDDN = primary or supplementary DN of ACD Group B
- ACDGroup - StationNumber = primary DN of ACD Group B
- DeviceID - PositionID = Agent 3

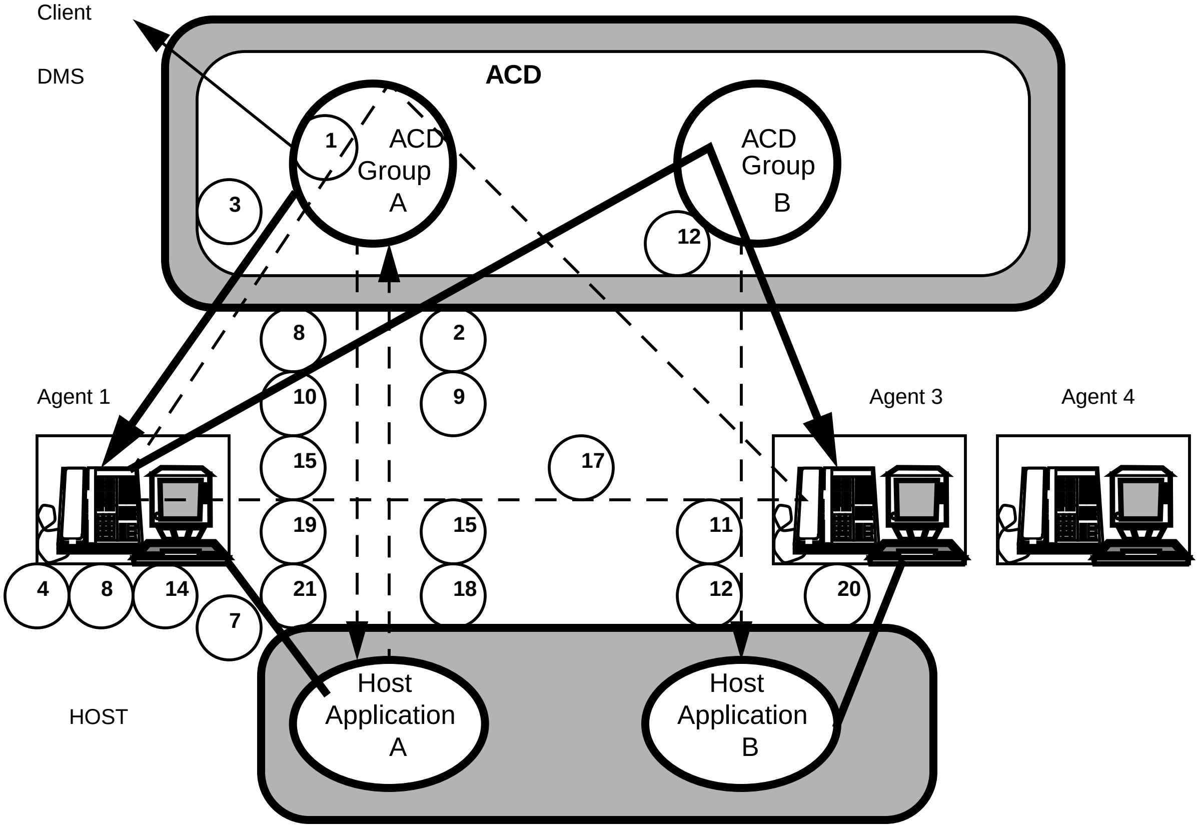
--> Host Application B 'closes' call record

20) dv-Call-Released-U sent to Host Application A when call to client clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- ACDDN = primary or supplementary DN of ACD Group A
- ACDGroup - StationNumber = primary DN of ACD Group A
- DeviceID - PositionID = Agent 1

--> Host Application A 'closes' call record

Note: When all three parties are on the conference call, the controller can choose to transfer call, and steps 17-20 will be replaced by steps 14-18 in Transfer with consultation using the host scenario in 6.3.5.

6.3.6 MakeCall and conference using the host



1) - 7) Same as MakeCall using the Host in 6.3.1

8) Agent 1 (controller) decides to add a consult party (Agent 3) to the call

9) dv-Add-Party sent to the switch; Host requests a consult call to be established
- AddPartyType - AddConsultParty or AddConsultforConf* (puts existing call on hold)
- OrigAddress - PositionID = Agent 1
- DestAddress - DialedDigits = Agent 3 (consult party)

Note: Consult party does not have to be an ACD agent.

10) RETURN-RESULT sent to Host Application A (indicating valid request)
- NetworkCallID - same as 5

11) dv-Call-Offered-U sent to Host Application B
- NetworkCallID - same as in 5
- ACDDN - primary or supplementary DN of ACD Group B
- OrigCallingNumber - Agent 1's Incalls key DN
- OrigChargeNumber - not provided

11) (Cont'd)
- CallHistoryInfo
  - CallType = CallTransferred
  - OrigInboundDN = Not provided
  - PrevApplicationID = Host Application A
- ACDGroup - StationNumber = primary DN of ACD Group B
- DeviceID - PositionID = Agent 3

12) dv-Call-Answered-U sent to Host Application B when Agent 3 answers call
- same parameters as in 11)

--> Host displays client record on Agent 3's terminal

13) The controller (Agent 1) is connected to the consult party (Agent 3) but can’t talk to the held party

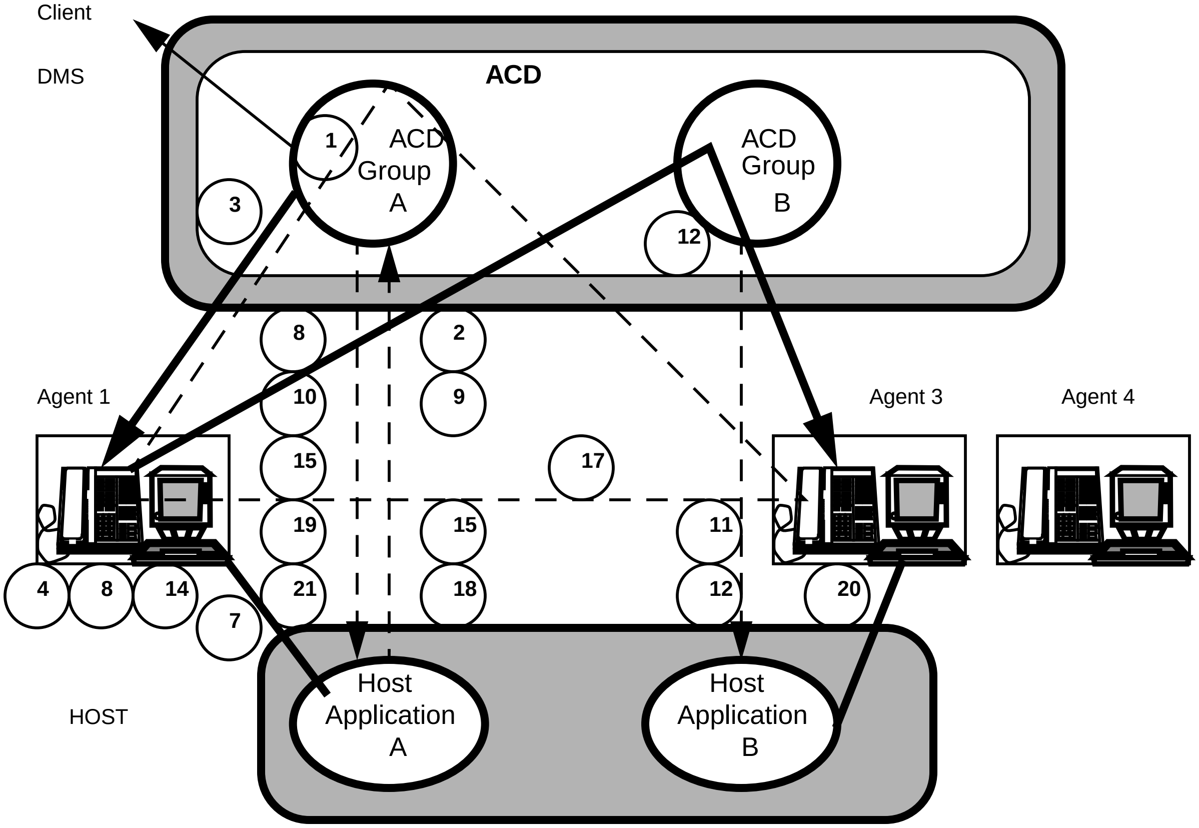
--> controller's and consult party's displays are updated with call info if they have MBS sets

Note: Scenario assumes idle consult party (Agent 3); therefore, no transfer queuing is needed

14) The controller initiates call conference via the host

(To be continued on the following page)

(continued)



(Continued from previous page)

15) dv-Conference-Party sent to the switch; Host requests call conference
- OrigAddress - PositionID = Agent 1

16) RETURN-RESULT sent to Host Application A (indicating valid request)
- NetworkCallID - same as in 5

17) The controller, held party, and consult party are all connected and active on a conference call. --> The controller's and consult party's displays are updated with call info if they have MBS sets.

18) dv-Drop-Party sent to switch; Host requests to drop consult party (for Agent 1)
- DropPartyType - DropConsultParty
- OrigAddress - PositionID = Agent 1

19) RETURN-RESULT sent to Host Application A (indicating valid request)
- Network CallID - same as in 5

20) dv-Call-Released-U sent to Host Application B
- NetworkCallID = same as in 5
- ReleaseReason = PartyDropped
- ACDDN = primary or supplementary DN of ACD Group B
- ACDGroup - StationNumber = primary DN of ACD Group B
- DeviceID - PositionID = Agent 3

--> Host Application B 'closes' call record

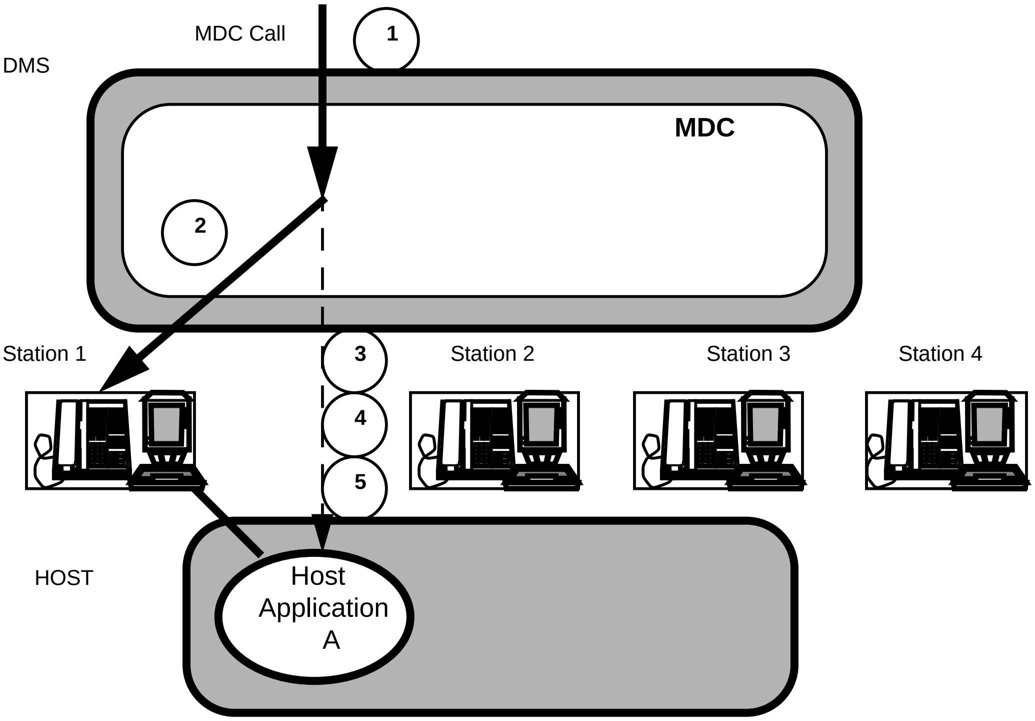
21) dv-Call-Released-U sent to Host Application A when call to client clears
- NetworkCallID - same as in 5
- ReleaseReason = CallCleared
- ACDDN - not available
- ACDGroup - not available
- DeviceID - PositionID = Agent 1

--> Host Application A 'closes' call record

Note: When all three parties are on the conference call, the controller can choose to transfer call, and steps 18-21 will be replaced by steps 13-18 in Transfer with consultation using the Host scenario (6.3.5).

6.4 MDC CompuCALL options

6.4.1 Normal call



Procedure 2

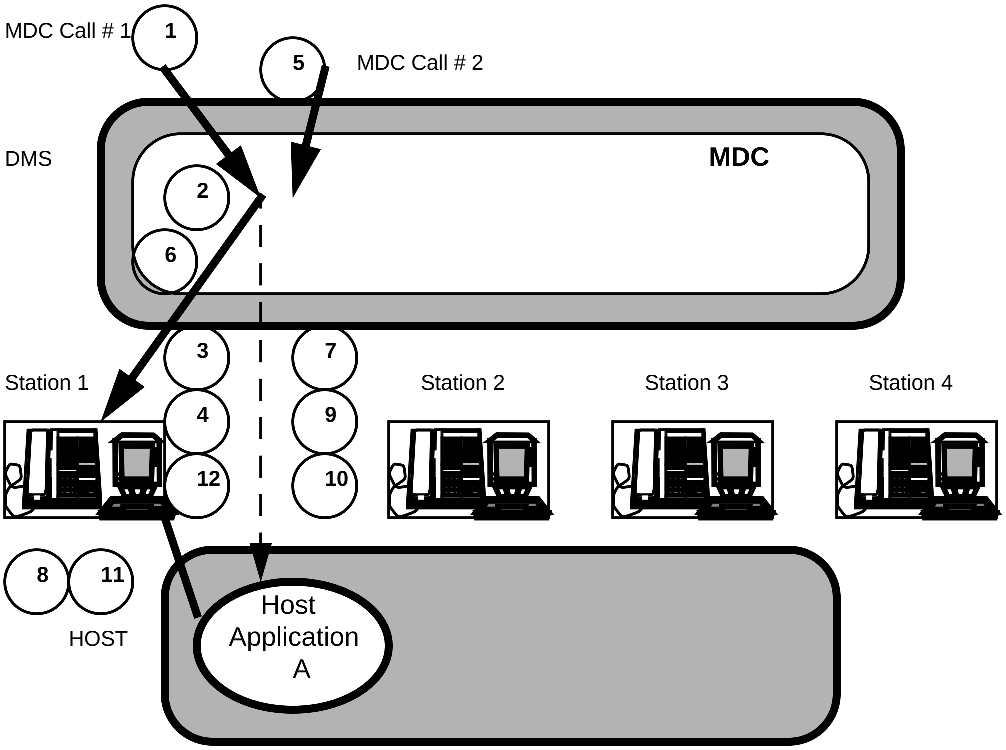
1) Incoming call received by switch for DN at Station 1
2) Call offered to idle Station 1
3) dv-Call-Offered-U sent to Host Application A
   - NetworkCallID - new
   - OrigCallingNumber - if available
   - OrigChargeNumber - if available
   - DeviceID - StationNumber = Station 1

   --> Host Application A displays call record on Station 1's terminal
4) dv-Call-Answered-U sent to Host Application A when Station 1 answers call

   --> The same parameters as in 3
5) dv-Call-Released-U sent to Host Application A when call clears
   - NetworkCallID - same as in 3
   - ReleaseReason = CallCleared
   - DeviceID - StationNumber = Station 1

   --> Host Application A closes call record

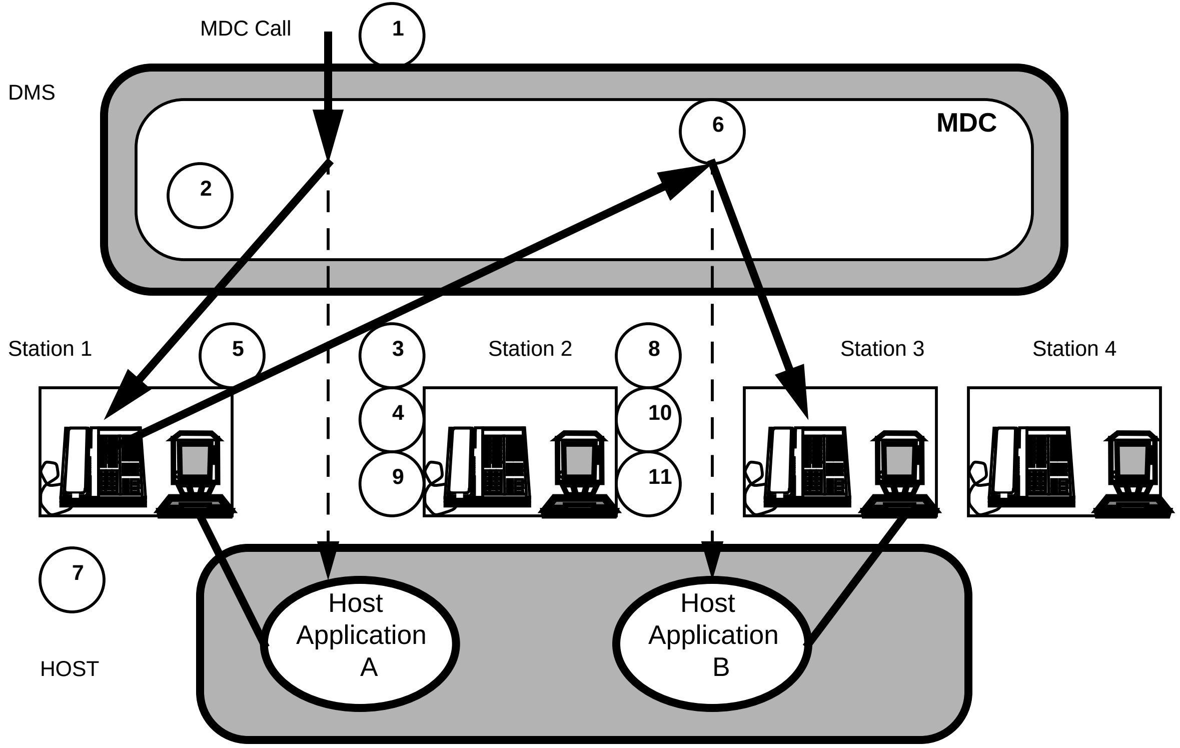
6.4.2 Normal call with call wait



1) Incoming call #1 received by switch for DN at Station 1
2) Call #1 offered to Station 1
3) dv-Call-Offered-U sent to Host Application A regarding call #1
   •NetworkCallID - new
   •OrigCallingNumber - if available
   •OrigChargeNumber - if available
   •DeviceID - StationNumber = Station 1
   --> Host Application A displays call #1 record on Station 1's terminal
4) dv-Call-Answered-U sent to Host Application A when Station 1 answers call #1
   The same parameters as in 3
5) Incoming call #2 received by switch for DN at Station 1 which has Call Waiting feature active
6) Call #2 offered to Station 1 as waited call
7) dv-Call-Offered-U sent to Host Application A regarding call #2
   •NetworkCallID - new
   •OrigCallingNumber - new, if available
   •OrigChargeNumber - new, if available
   •CallMode = CallWaited
   •DeviceID - StationNumber = Station 1
   --> Host Application A retrieves caller #2 information/prepares screen display
8) Station 1 puts call #1 on hold and answers call #2
9) dv-Call-Answered-U sent to Host Application A when Station 1 answers call #2
   • The same parameters as in 7
   --> Host Application A displays call #2 record on station 1's terminal
10) dv-Call-Released-U sent to Host Application A when call #2 clears
   •NetworkCallID - same as in 7
   •ReleaseReason = CallCleared
   •DeviceID - StationNumber = Station 1
   --> Host Application A 'closes' call #2 record and redisplays call #1 record on Station 1's terminal
11) Station 1 returns to call #1
12) dv-Call-Released-U sent to Host Application A when call #1 clears
   •NetworkCallID - same as in 3
   •ReleaseReason = CallCleared
   •DeviceID - StationNumber = Station 1
   --> Host Application A 'closes' call #1 record

Note: After Step 9, Station 1 can choose to put call #2 on hold and return to call #1, and steps 11 and 12 will precede step 10. Assuming the display on station 1's data terminal can be controlled by the worker at station 1.

6.4.3 Intra-switch single call transfer



Procedure 3

1) - 4) Same as Normal call in 6.4.1
5) Station 1 initiates call transfer to Station 3 using the voice set
6) Call offered to Station 3
7) Station 1 completes the call transfer and disconnects (i.e., blind transfer)
8) dv-Call-Offered-U sent to Host Application B
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo
  - CallType = CallTransferred
  - OrigInboundDN = Station 1
  - PrevApplicationID = Host Application A
- DeviceID - StationNumber = Station 3

--> Host Application B displays call record on Station 3's terminal

9) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- DeviceID - StationNumber = Station 1

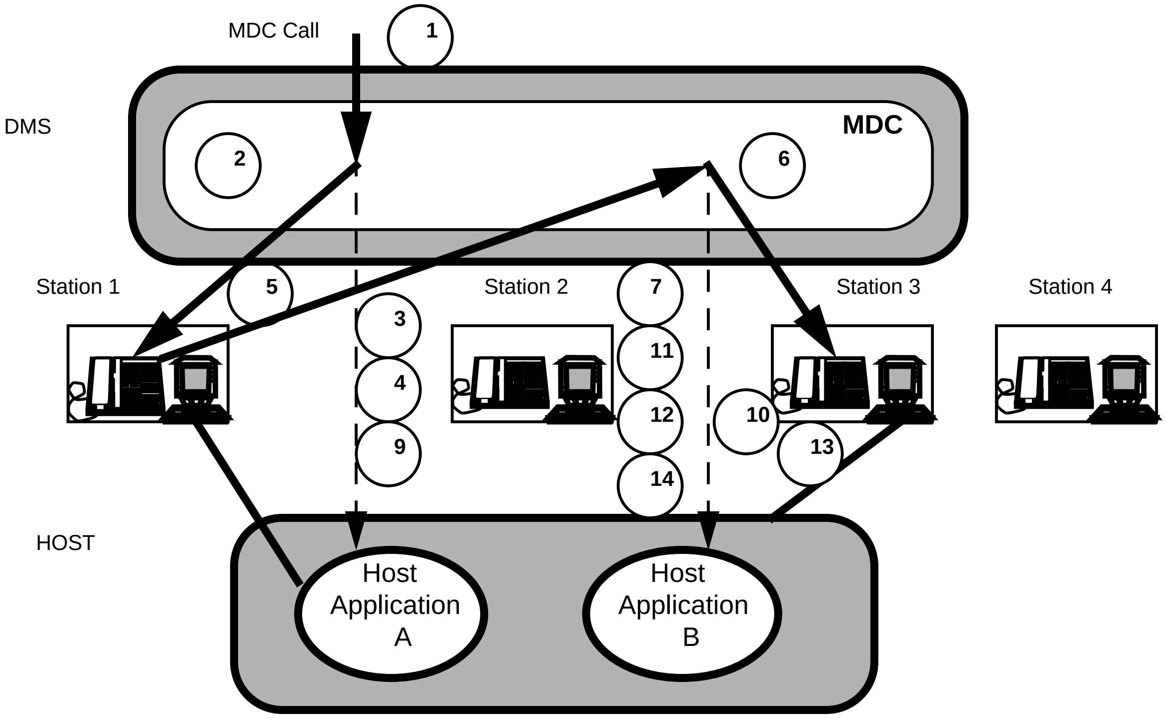
--> provides 'closure' relating to Station 1's involvement with the call to Host Application A

dv-Call-Answered-U sent to Host Application B when Station 3 answers call
- same parameters as in 8

11) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

--> Host Application B closes call record

6.4.4 Intra-switch single call transfer on call wait



1) - 4) Same as Normal call in 6.4.1

5) Station 1 initiates call transfer to Station 3

6) Call offered to Station 3 as waited call (call #2) while Station 3 is active on an existing call (call #1)

7) dv-Call-Offered-U sent to Host B regarding the transferred call (call #2)
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo
  - CallType = CallTransferred
  - OrigInboundDN = Station 1
  - PrevApplicationID = Host Application A
- CallMode = CallWaited
- DeviceID - StationNumber = Station 3

--> Host Application B retrieves caller information/ prepares screen display regarding transferred call

8) Station 1 completes the call transfer and disconnects (i.e., blind transfer)

9) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- DeviceID - StationNumber = Station 1

--> provides 'closure' relating to Station 1's involvement with the call to Host A

10) Station 3 puts existing call (call #1) on hold and answers transferred call(call#2)

11) dv-Call-Answered-U sent to Host Application B when Station 3 answers the call

--> The same parameters as in 7

--> Host Application A displays call #2 record on station 1's terminal

12) dv-Call-Released-U sent to Host Application B when call #2 clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber =Station 3

--> Host Appl. B 'closes' call record and re-displays call #1 record on station 1's terminal

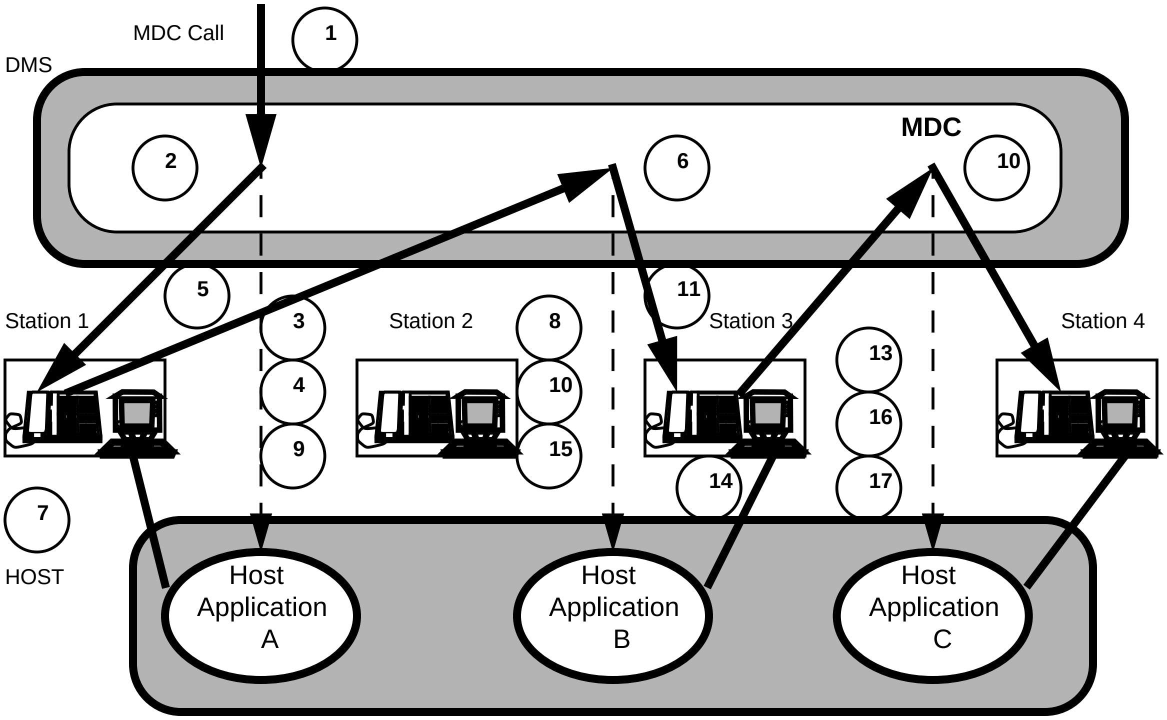
13) Station 3 returns to call #1

14) dv-Call-Released-U sent to Host Application B when call #1 clears
- NetworkCallID - same as previous for call #1
- ReleaseReason = CallCleared
- DeviceID - StationNumber =Station 3

--> Host B 'closes' call record for call #1

Note: After step 11, Station 3 can choose to put call #2 on hold and return to existing call #1, then steps 13 and 14 will precede step 12. Assuming the display on Station 3's data terminal can be controlled by the worker at Station 3.

6.4.5 Intra-switch multiple call transfer



1) - 10) Same as Intra-switch single call transfer in 6.4.3

11) Station 3 initiates call transfer to Station 4 using the voice set

12) Call offered to Station 4

13) dv-Call-Offered-U sent to Host Application C
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo
  - CallType = CallTransferred
  - OrigInboundDN = Station 1
  - PrevApplicationID = Host Application B
- DeviceID - StationNumber = Station 4

--> Host Application C obtains call record from Host Application B and displays call record on Agent 4's terminal (e.g., shared screen)

14) Station 3 completes the call transfer and disconnects (i.e., blind transfer)

15) dv-Call-Released-U sent to Host Application B
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- DeviceID - StationNumber = Station 3

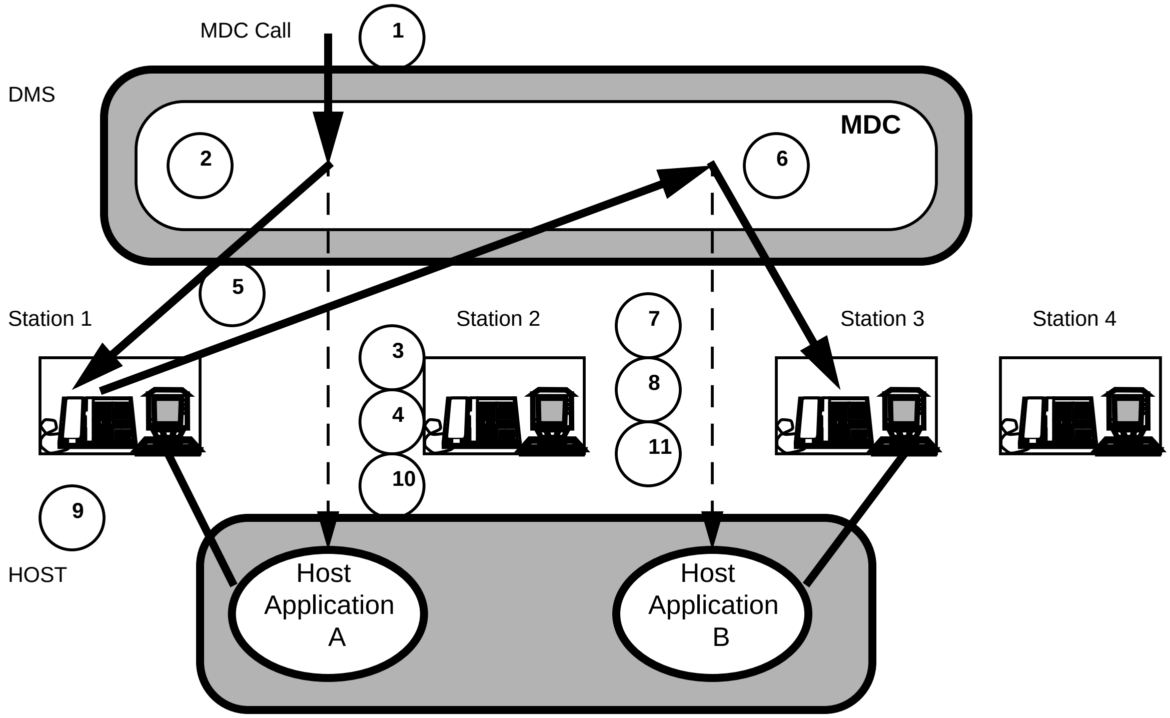
--> provides 'closure' relating to Station 3's involvement with the call to Host Application B

16) dv-Call-Answered-U sent to Host Application C when Station 4 answers call
- same parameters as in 13

17) dv-Call-Released-U sent to Host Application C when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 4

--> Host Application C 'closes' call record

6.4.6 Intra-switch transfer with consultation



Procedure 4

1) - 6) Same as Intra-switch single call transfer in 6.4.3

7) dv-Call-Offered-U sent to Host Application B
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo
  - CallType = CallTransferred
  - OrigInboundDN = Station 1
  - PrevApplicationID = Host Application A
- DeviceID - StationNumber = Station 3

--> Host Application B obtains call record from Host Application A and displays it on Station 3's terminal (e.g., shared screen)

8) dv-Call-Answered-U sent to Host Application B when Station 3 answers call
- same parameters as in 7

--> Host Application B is confirmed that Station 1 has answered call

9) Station 1 consults with Station 3, with caller on hold; after private consultation, Station 1 completes the call transfer and disconnects

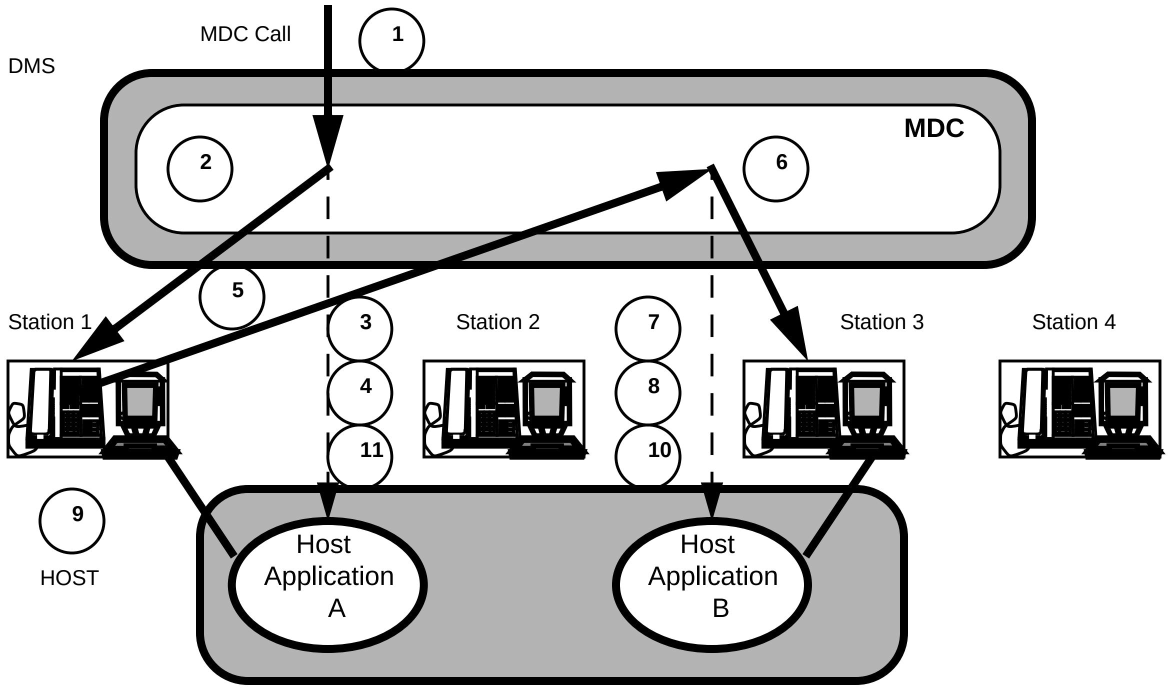
10) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- DeviceID - StationNumber = Station 1

--> provides 'closure' on previous messages to Host Application A

11) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

--> Host Application B 'closes' call record

6.4.7 Intra-switch call consultation



1) - 4) Same as Normal Call in 6.4.1

5) Station 1 initiates consult call to Station 3 w/ the voice set

6) Call offered to Station 3

7) dv-Call-Offered-U sent to Host B
•NetworkCallID - same as in 3
•OrigCallingNumber - same as in 3
•OrigChargeNumber - same as in 3
•CallHistoryInfo
- CallType = CallTransferred
- OrigInboundDN = Station 1
- PrevApplicationID = Host Application A
•DeviceID - StationNumber = Station 3
--> Host B gets call record from Host A and displays it on Station 3's terminal (shared)

(Note: the CallType in this case is to indicate the call can be transferred or conferenced as opposed to has been transferred or conferenced.)

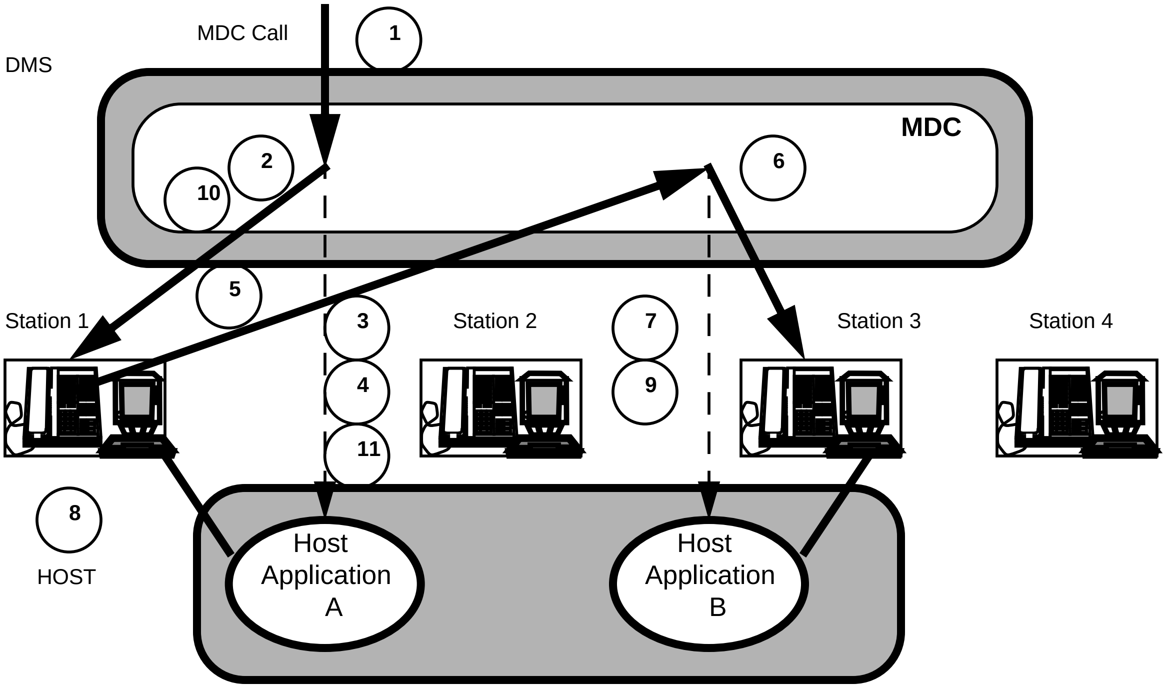
8) dv-Call-Answered-U sent to Host Application B when Station 3 answers call
• same parameters as in 7
--> Host Application B is confirmed that Station 3 has answered call

9) Station 1 consults with Station 3, with original call party on hold; after private consultation, Station 1 releases consult party

10) dv-Call-Released-U sent to Host Application B
•NetworkCallID - same as in 3
•ReleaseReason = PartyDropped
•DeviceID - StationNumber = Station 3
--> provides 'closure' relating to Station 3's involvement with the call to Host Application B

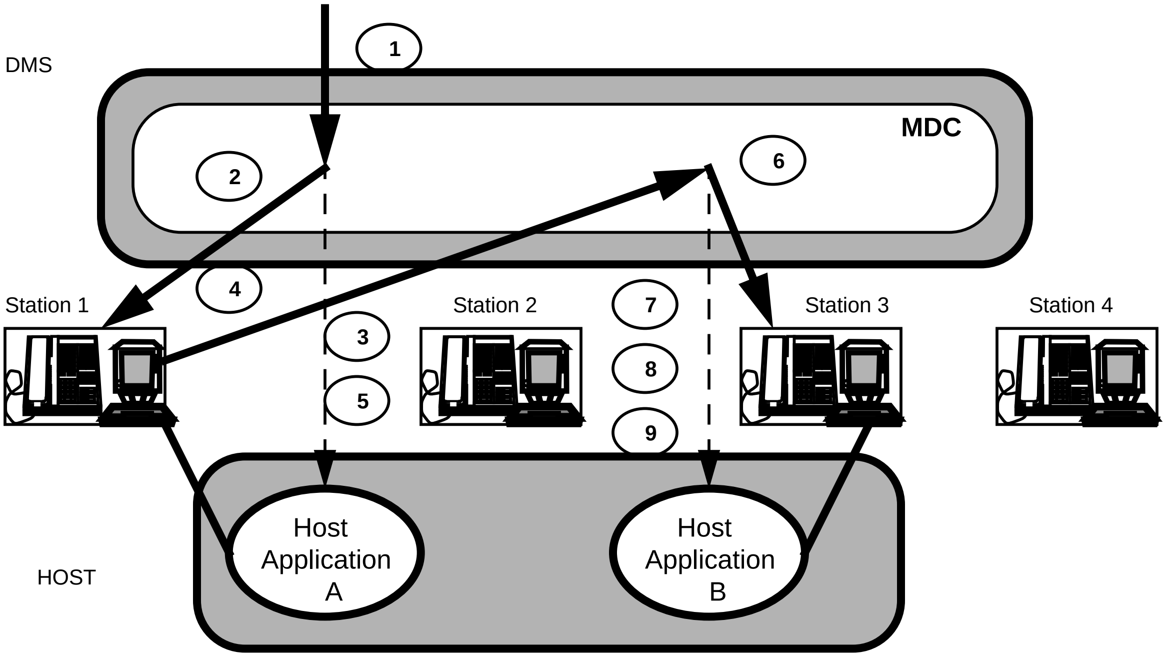
11) dv-Call-Released-U sent to Host Application A when call clears
•NetworkCallID - same as in 3
•ReleaseReason = CallCleared
•DeviceID - StationNumber = Station 1
--> Host Application A 'closes' call record

6.4.8 Intra-switch consult party dropped before answer



1) - 4) Same as Normal Call in 6.4.1
5) Station 1 initiates consult call to Station 3 using the voice set
6) Call offered to Station 3
7) dv-Call-Offered-U sent to Host Application B
•NetworkCallID - same as in 3
•OrigCallingNumber - same as in 3
•OrigChargeNumber - same as in 3
•CallHistoryInfo
- CallType = CallTransferred*
- OrigInboundDN = Station 1
- PrevApplicationID = Host Application A
•DeviceID - StationNumber = Station 3
--> Host Application B obtains call record from Host Application A and displays it on Station 3's terminal (e.g., shared screen)
8) Station 1 drops consult party before consult party answers
9) dv-Call-Released-U sent to Host Application B
•NetworkCallID - same as in 3
•ReleaseReason = PartyDroppedNoAnswer
•DeviceID - StationNumber = Station 3
--> provides 'closure' relating to Station 3's involvement with the call to Host Application B
10) Station 1 is reconnected to original call party
11) dv-Call-Released-U sent to Host Application A when call clears
•NetworkCallID - same as in 3
•ReleaseReason = CallCleared
•DeviceID - StationNumber = Station 1
--> Host Application A 'closes' call record

6.4.9 Intra-switch single call forward



1) - 3) Same as Normal Call in 6.4.1
4) Call Forward Don't Answer (CFD) feature forwards the call to station 3
5) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallForwarded
- DeviceID - StationNumber = Station 1

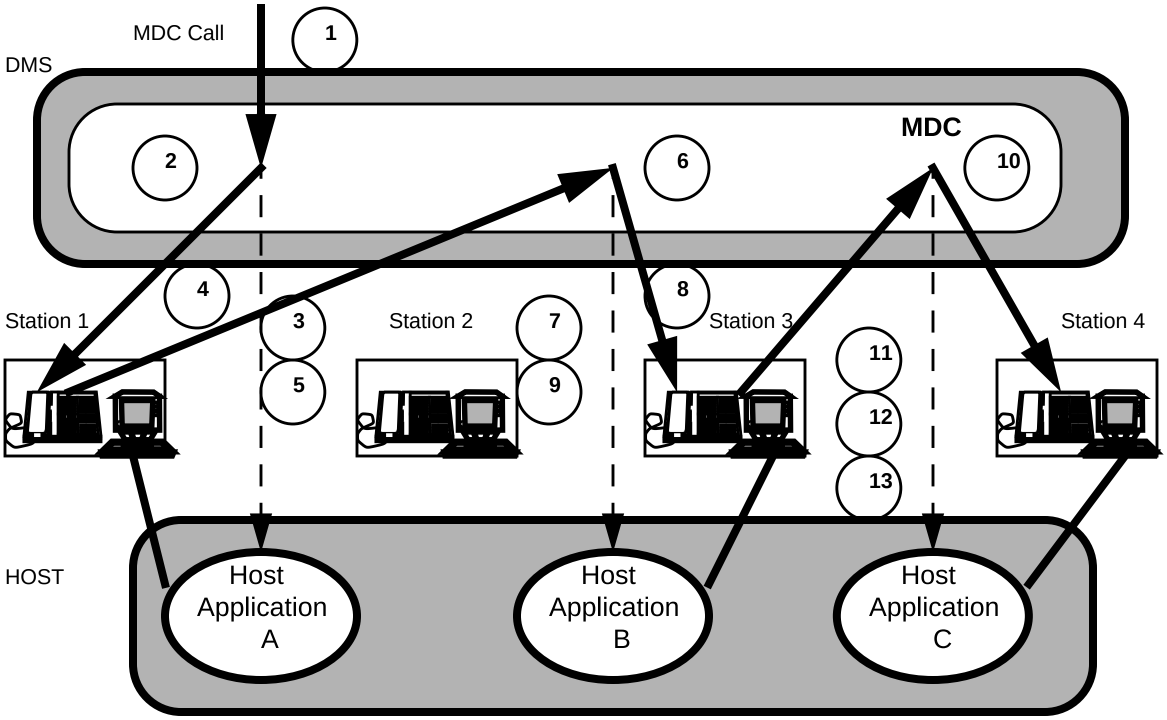
--> provides 'closure' relating to Station 1's involvement with the call to Host Application A
6) Call offered to Station 3
7) dv-Call-Offered-U sent to Host Application B
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo
  - CallType = CallForwarded
  - OrigInboundDN = Station 1
  - PrevApplicationID = Host Application A
- DeviceID - StationNumber = Station 3

--> Host Application B displays call record on Station 3's terminal
8) dv-Call-Answered-U sent to Host Application B when Station 3 answers the call
- The same parameters as in dv-Call-Offered-U in 7

--> Host Application B is confirmed that Station 3 has answered call
9) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

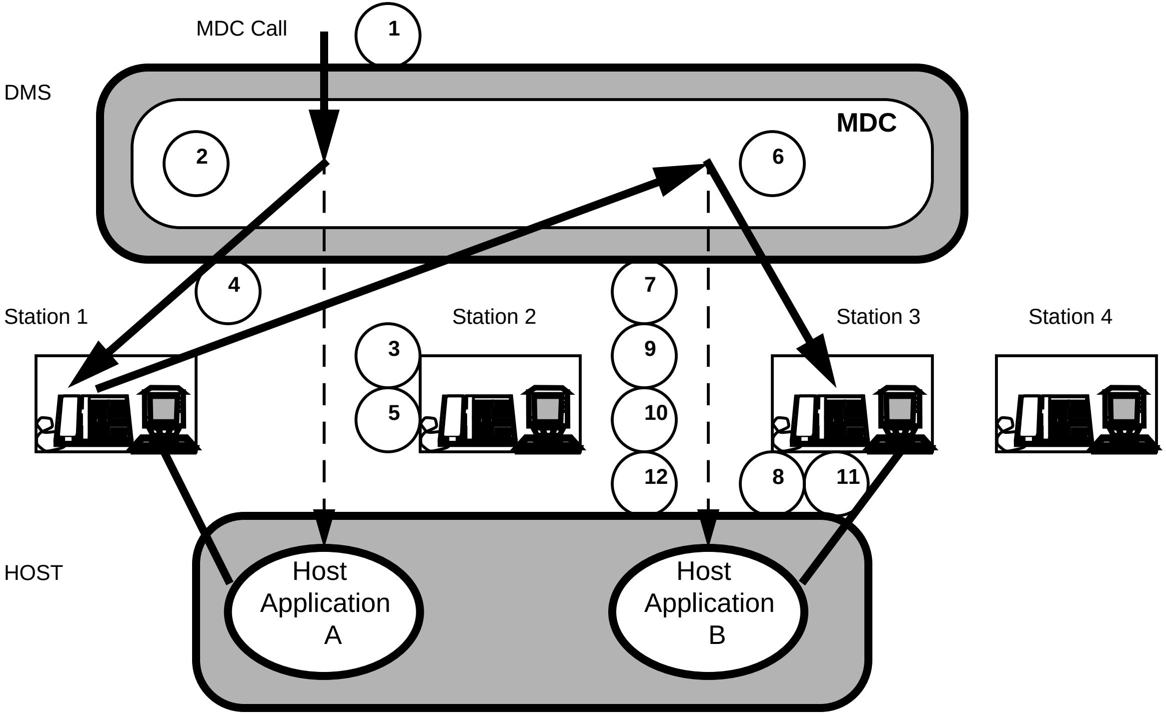
--> Host Application B closes call record

6.4.10 Intra-switch multiple call forward



1) - 7) Same as Intra-switch single call forward in 6.4.9
8) Call Forward Don't Answer (CFD) feature forwards the call at Station 3 to Station 4
9) dv-Call-Released-U sent to Host Application B
•NetworkCallID - same as in 3
•ReleaseReason = CallForwarded
•DeviceID - StationNumber = Station 3
--> provides 'closure' relating to Station 3's involvement with the call to Host Application B
10) Call offered to Station 4
11) dv-Call-Offered-U sent to Host Application C
•NetworkCallID - same as in 3
•OrigCallingNumber - same as in 3
•OrigChargeNumber - same as in 3
•CallHistoryInfo
- CallType = CallForwarded
- OrigInboundDN = Station 1
- PrevApplicationID = Host Application B
•DeviceID - StationNumber = Station 4
--> Host Application C displays call record on Station 4's terminal
12) dv-Call-Answered-U sent to Host Application C when Station 4 answers the call
--> The same parameters as in 11
--> Host Application C is confirmed that Station 4 has answered call
13) dv-Call-Released-U sent to Host Application C when call clears
•NetworkCallID - same as in 3
•ReleaseReason = CallCleared
•OrigChargeNumber - same as in 3
•DeviceID - StationNumber = Station 4
--> Host Application C 'closes' call record

6.4.11 Intra-switch single call forward on call wait



1) - 3) Same as Normal Call in 6.4.1

4) Call Forward Don't Answer (CFD) feature forwards the call to station 3

5) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallForwarded
- DeviceID - StationNumber=Station 1

--> provides 'closure' relating to Station 1's involvement with the call to Host Application A

6) Call offered to Station 3 as waited call (call #2) while Station 3 is busy on existing call (call #1)

7) dv-Call-Offered-U sent to Host Application B
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- CallHistoryInfo
  - CallType = CallForwarded
  - OrigInboundDN = Station 1
  - PrevApplicationID = Host Application A
- CallMode = CallWaited
- DeviceID - StationNumber=Station 3

--> Host Application B prepares call #2 screen display for Station 3

8) Station 3 puts existing call (call #1) on hold and answers forwarded call (call #2)

9) dv-Call-Answered-U sent to Host Application B when Station 3 answers call #2
- The same parameters as in 7

--> Host Application B displays call #2 record for Station 3's terminal

10) dv-Call-Released-U sent to Host Application B when call #2 clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

--> Host Application B 'closes' call #2 record

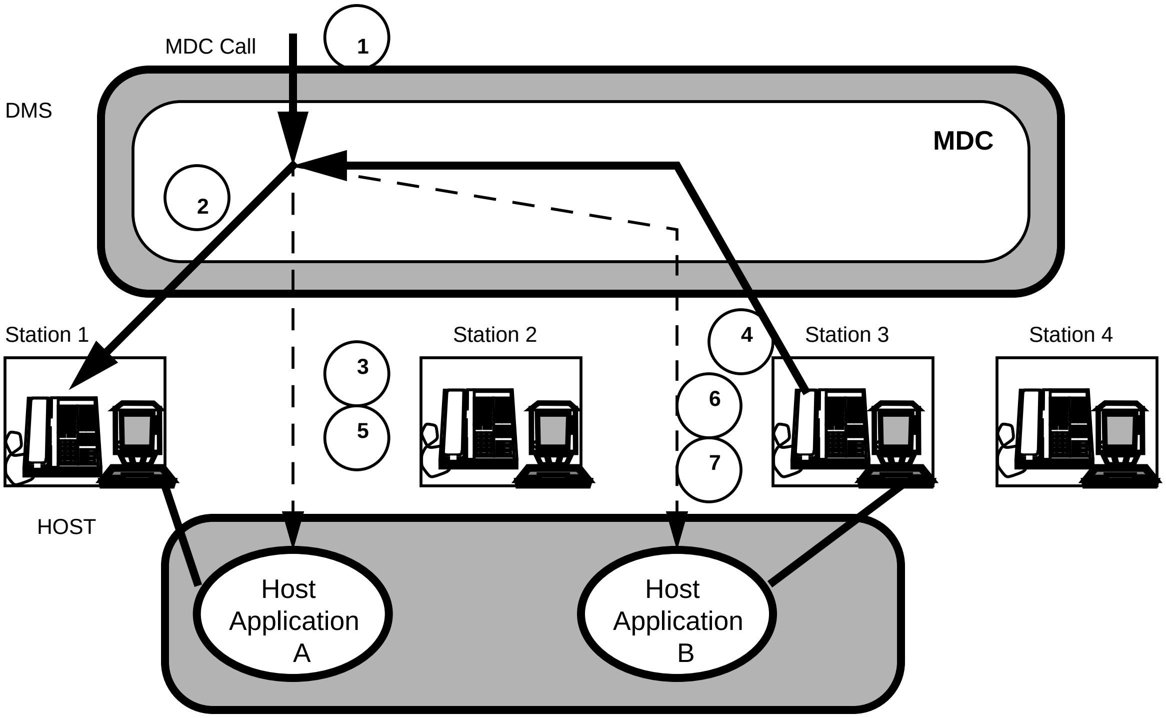
11) Station 3 returns to call #1

12) dv-Call-Released-U sent to Host Application B when call #1 clears
- NetworkCallID - same as previous for call #1
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

--> Host Application B 'closes' call #1 record

Note: After step 9, Station #3 can choose to put call #2 on hold and return to existing call #1, then steps 11 & 12 will precede step 10. Assuming the display on Station 3's terminal can be controlled by the worker at Station 3.

6.4.12 Intra-switch single call pick-up



Procedure 5

1) - 3) Same as Normal call in 6.4.1
4) Station 3 initiates call pick-up
5) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallPickedUp
- DeviceID - StationNumber = Station 1

--> provides 'closure' relating to Station 1's involvement with the call to Host Application A

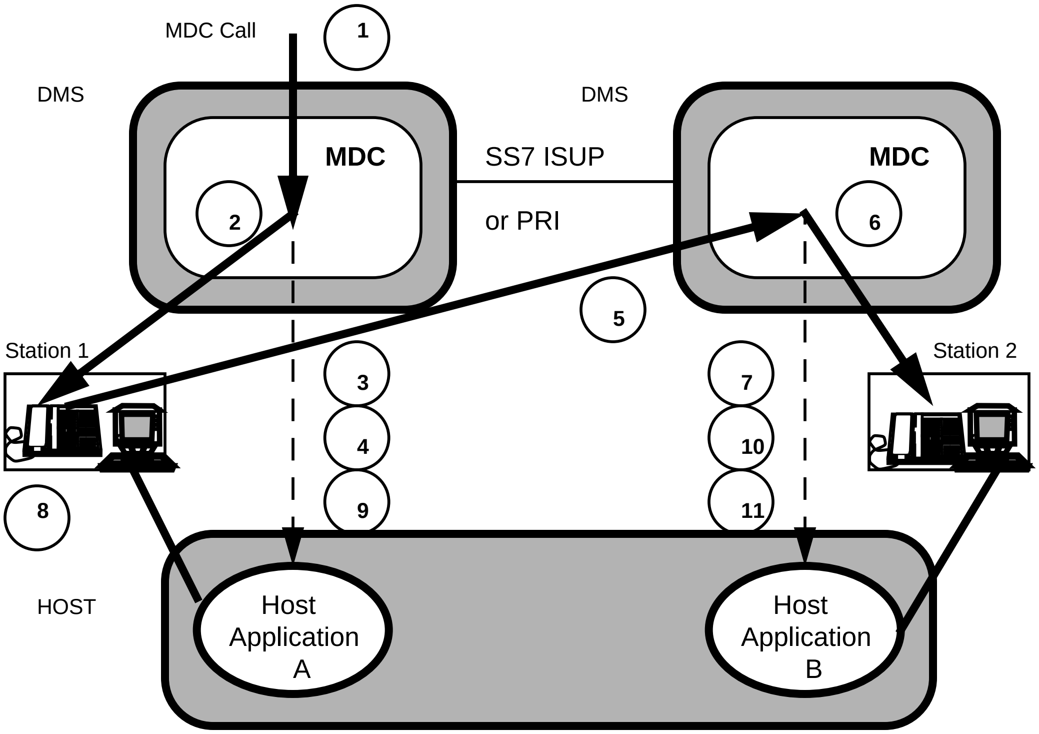
6) dv-Call-Answered-U sent to Host Application B when Station 3 answers the call
- NetworkCallID - same as in 3
- OrigCallingNumber - same as in 3
- OrigChargeNumber - same as in 3
- DeviceID - StationNumber = Station 3

--> Host Application B displays call record on Station 3's terminal

7) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 3
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

--> Host Application B closes call record

6.4.13 Inter-switch call transfer



Procedure 6

1) - 4) Same as Normal Call in 6.4.1
5) Station 1 initiates call transfer to Station 3
6) Call offered to Station 3
7) dv-Call-Offered-U sent to Host Application B
- NetworkCallID - new (not the same as in 3)
- OrigCallingNumber - transferring station's DN (only if over ISUP or PRI trunk)
- OrigChargeNumber - transferring station's ANI (only if over ISUP trunk)
- DeviceID - StationNumber - Station 3

Note: Since inter-switch SS7 signaling does not currently support 'networked CompuCALL', inter-switch transferred call is treated by second switch as a new call, and transfer of data session must be based on the transferring station's OrigCallingNumber/ OrigChargeNumber (complex and unreliable) or manually initiated/coordinated by the two stations via the host using transfer with consultation.

8) Station 1 completes the call transfer and disconnects (i.e., blind transfer)
9) dv-Call-Released-U sent to Host Application A
- NetworkCallID - same as in 3
- ReleaseReason = CallTransferred
- DeviceID - StationNumber = Station 1

--> provides 'closure' relating to Station 1's involvement with the call to Host Application A

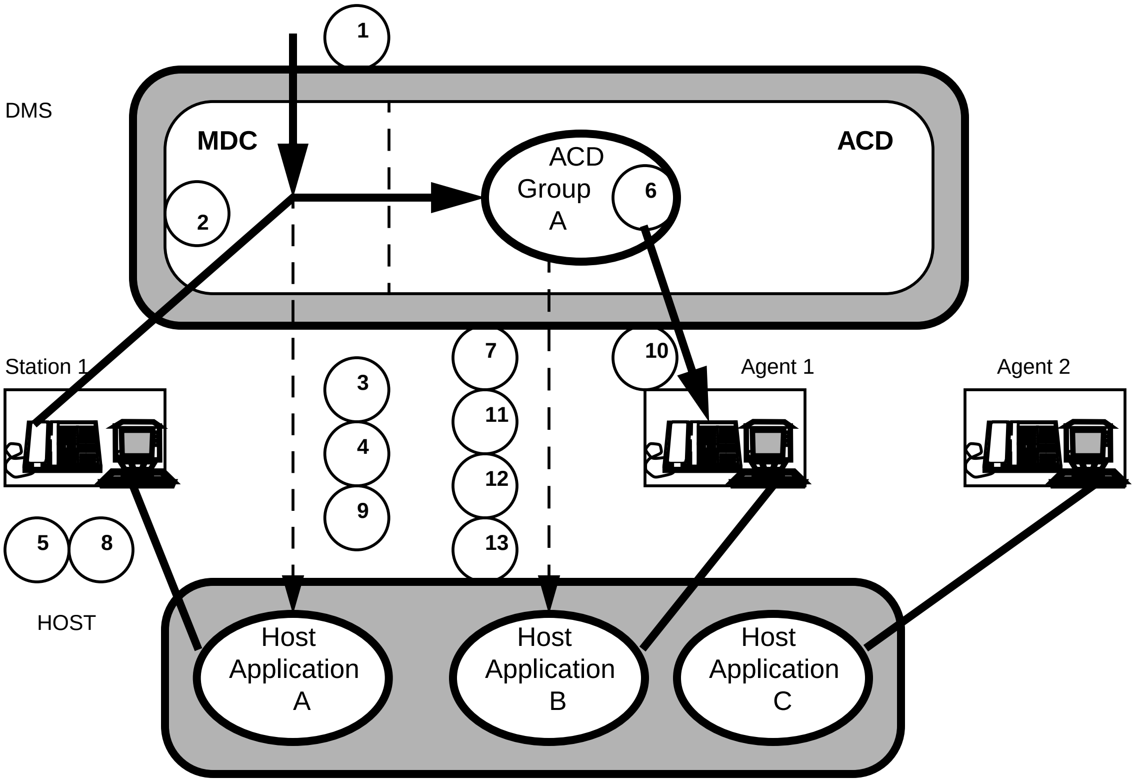
10) dv-Call-Answered-U sent to Host Application B when Station 3 answers call
- same parameters as in 7

--> Host Application B is confirmed that Station 3 has answered call

11) dv-Call-Released-U sent to Host Application B when call clears
- NetworkCallID - same as in 7
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 3

--> Host Application B 'closes' call record

6.4.14 Intra switch MDC call transferred to an ACD line



1) - 4) Same as Normal Call in 6.4.1

5) Station 1 initiates call transfer to ACD Group A using the voice set

6) ACD call received by switch and queued for agent (if necessary)

7) dv-Call-Queued-U sent to Host Application B (if call queued)
   •NetworkCallID - same as in 3
   •ACDDN - primary or supplementary DN of ACD Group A
   •OrigCallingNumber - same as in 3
   •OrigChargeNumber - same as in 3
   •CallHistoryInfo
   - CallType = CallTransferred
   - OrigInboundDN = Station 1's DN
   - PrevApplicationID = Host Application A
   •ACDGroup - StationNumber = primary DN of ACD Group A
   --> Host Application B obtains caller record from Host Application A and prepares screen display for agent (Note - All scenarios assume inter-host application communication)

8) Station 1 completes the call transfer and disconnects (i.e., 'blind transfer')

9) dv-Call-Released-U sent to Host Application A
   •NetworkCallID - same as in 3
   •ReleaseReason = CallTransferred
   •DeviceID - StationNumber = Station 1
   --> provides "closure" relating to Station 1's involvement with the call to Host Application A

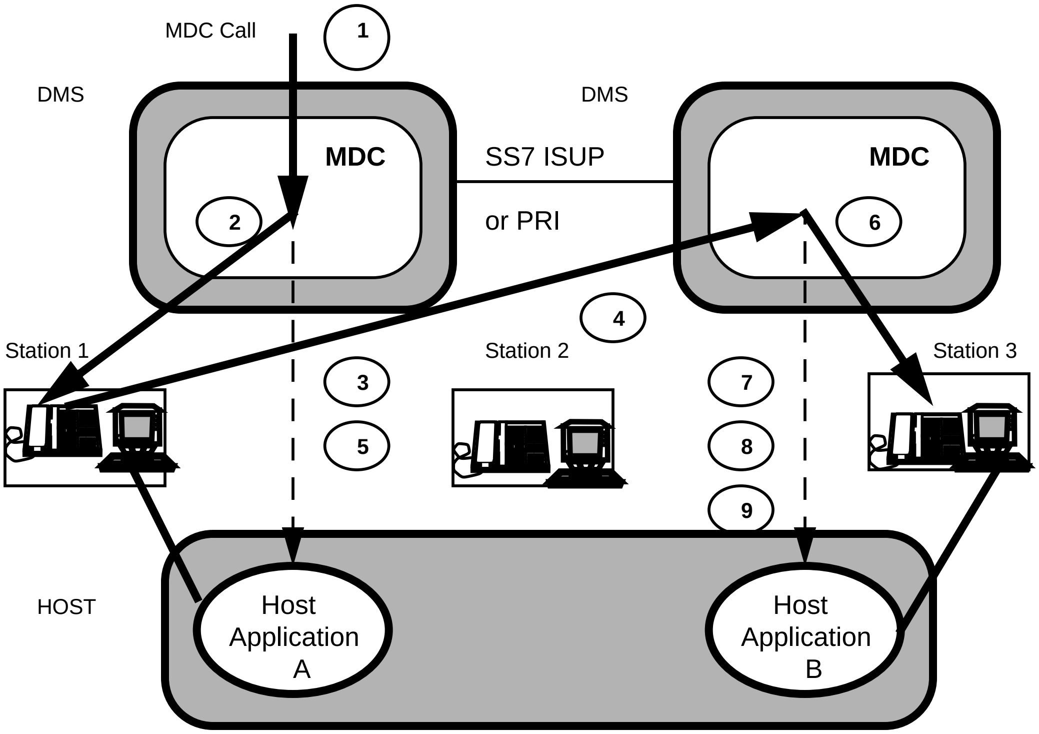
10) Agent 1 selected - call offered

11) dv-Call-Offered-U sent to Host Application B
   •same parameters as in 7, plus
   •DeviceID - PositionID = Agent 1
   --> Host Application B displays call record on Agent 1's terminal

12) dv-Call-Answered-U sent to Host Application B when agent 1 answers call
   •same parameters as in 11
   --> Host Application B is confirmed that Agent 1 is answering call

13) dv-Call-Released-U sent to Host Application B when call clears
   •NetworkCallID - same as in 3
   •ReleaseReason = CallCleared
   •ACDDN - same as in 7
   •ACDGroup - same as in 7
   •DeviceID - same as in 11
   --> Host Application B 'closes' call record (immediately after agent follow-up work)

6.4.15 Inter-switch single call forward



1) - 3) Same as Normal Call in 6.4.1

4) Call Forward Don't Answer (CFD) feature forwards the call to station 3 other DMS
--> call forwarding parameters: FirstFwdNumber (Station 1) and FirstFwdReason (NoReply) are sent to other DMS via SS7/PRI parameters: Orig Call Number (OCN) and Original Redirection in Redirection Indicators (RNI)

5) dv-Call-Released-U sent to Host Application A
•NetworkCallID - same as in 3
•ReleaseReason = CallForwarded
•DeviceID - StationNumber = Station 1
--> provides 'closure' relating to Station 1's involvement with the call to Host Application A

6) Call offered to Station 3

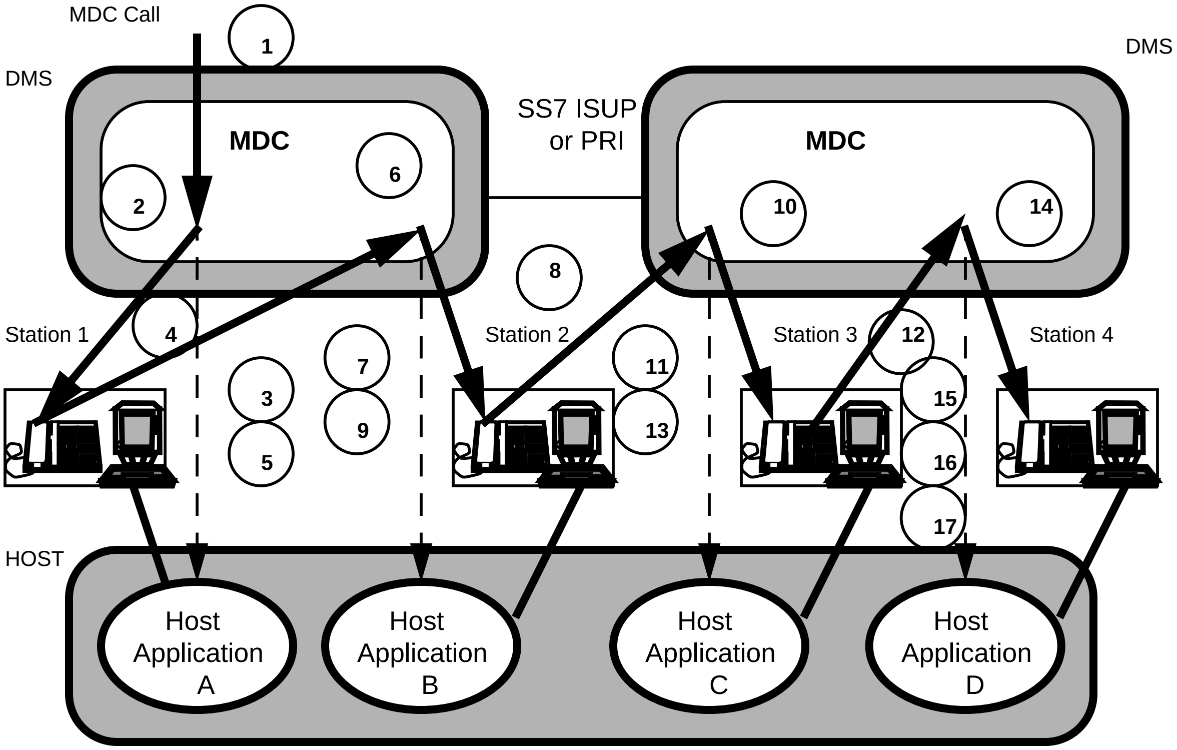
7) dv-Call-Offered-U sent to Host Application B
•NetworkCallID - new
•OrigCallingNumber - same as in 3 (ISUP/PRI)
•OrigChargeNumber - same as in 3
(ISUP only)

7) (Cont'd)
•DeviceID - StationNumber = Station 3
--> Host Application B displays call record on Station 3's terminal

8) dv-Call-Answered-U sent to
Host Application B when Station 3 answers the call
•The same parameters as in dv-Call-Offered-U in 7
--> Host Application B is confirmed that Station 3 has answered call

9) dv-Call-Released-U sent to Host Application B when call clears
•NetworkCallID - same as in 7
•ReleaseReason = CallCleared
•DeviceID - StationNumber = Station 3
--> Host Application B 'closes' call record

6.4.16 Inter-switch multiple call forward



Procedure 7

1) - 7) Same as Intra-switch single call forward in 6.4.9

8) Call Forward Don't Answer (CFD) feature forwards the call at Station 2 to Station 3
--> call forwarding parameters: FirstFwdNumber (Station 1), FirstFwdReason (NoReply), LastFwdNumber (Station 2) and LastFwdReason (NoReply) are sent to other DMS via SS7/PRI parameters: Orig Call Number (OCN), Original Redirection in Redirection Indicators (RNI), Redirecting Number (RGN) and the Redirecting Reason also in the RNI

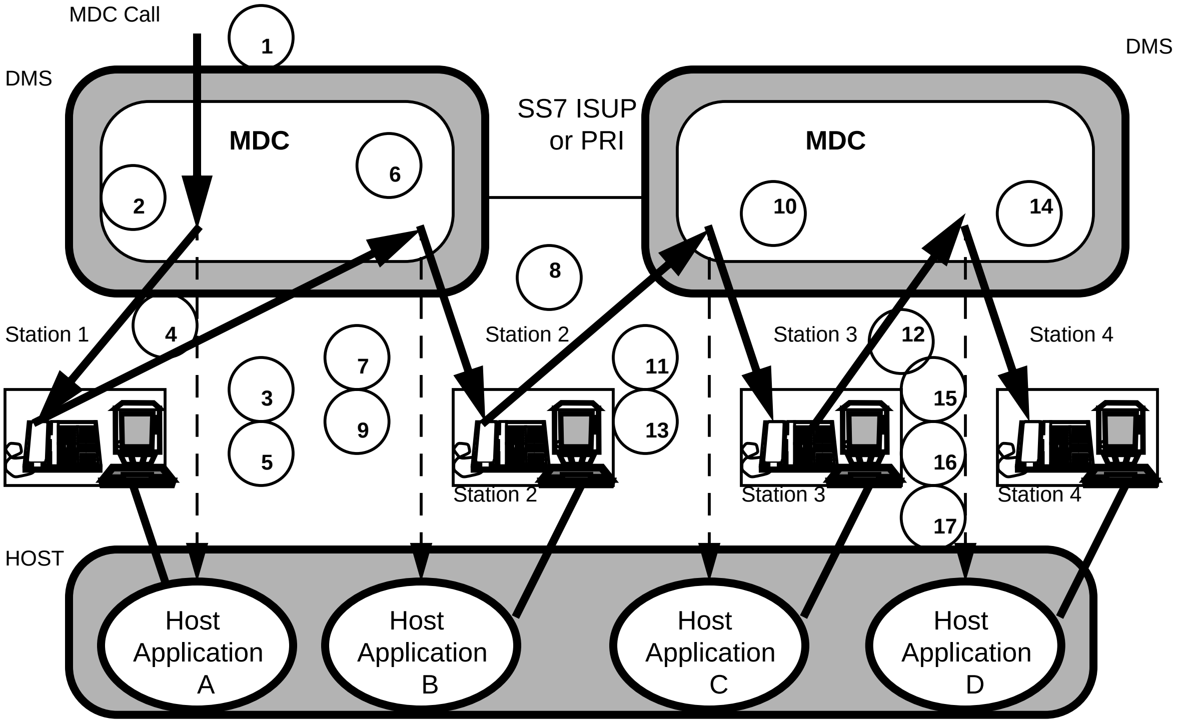
9) dv-Call-Released-U sent to Host Application B
- NetworkCallID - same as in 3
- ReleaseReason = CallForwarded
- DeviceID - StationNumber = Station 2
--> provides 'closure' relating to Station 2's involvement with the call to Host Application B

10) Call offered to Station 3

11) dv-Call-Offered-U sent to host
Application C
- NetworkCallID - new
- OrigCallingNumber - same as in 3 (ISUP/PRI)
- OrigChargeNumber - same as in 3 (ISUP only)
- DeviceID - StationNumber = Station 3

12) Call Forward Don't Answer (CFD) feature forwards the call to station 4

Procedure 7 (continued)



13) dv-Call-Released-U Application C sent to Host
- NetworkCallID - same as in 11
- ReleaseReason = CallForward
- DeviceID - StationNumber = Station 3

--> provides 'closure' relating to Station 3's involvement with the call to Host Application C

14) Call offered to Station 4

15) dv-Call-Offered-U sent to Host Application D
- NetworkCallID - same as in 11
- OrigCallingNumber - same as in 11
- OrigChargeNumber - same as in 11
- CallHistoryInfo
  - CallType = CallForwarded
  - OrigInboundDN = Station 3
  - PrevApplicationID = Host Application C
- DeviceID - StationNumber = Station 4
- ForwardingParty
  - FirstFwdNumber = Station 1
  - FirstFwdReason = NoReply
  - LastFwdNumber = Station 3
  - LastFwdReason = NoReply

--> Host Application D displays call record on Station 4's terminal

16) dv-Call-Answered-U sent to Host Application D when Station 4 answers the call
- The same parameters as in dv-Call-Offered-U in 15

--> Host Application D is confirmed that Station 4 has answered call

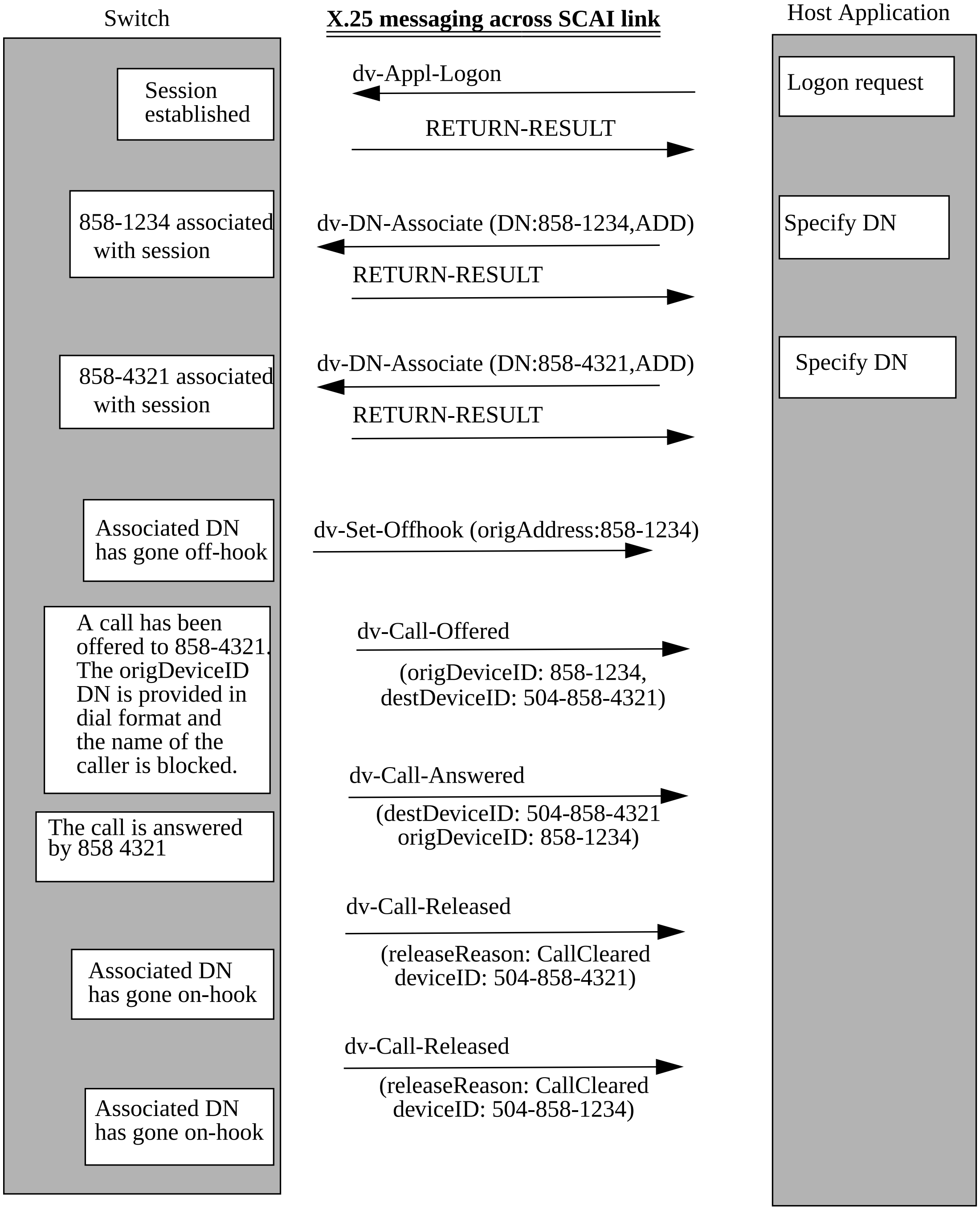
17) dv-Call-Released-U sent to Host Application D when call clears
- NetworkCallID - same as in 11
- ReleaseReason = CallCleared
- DeviceID - StationNumber = Station 4

--> Host Application D 'closes' call record

6.5 ICM sessions

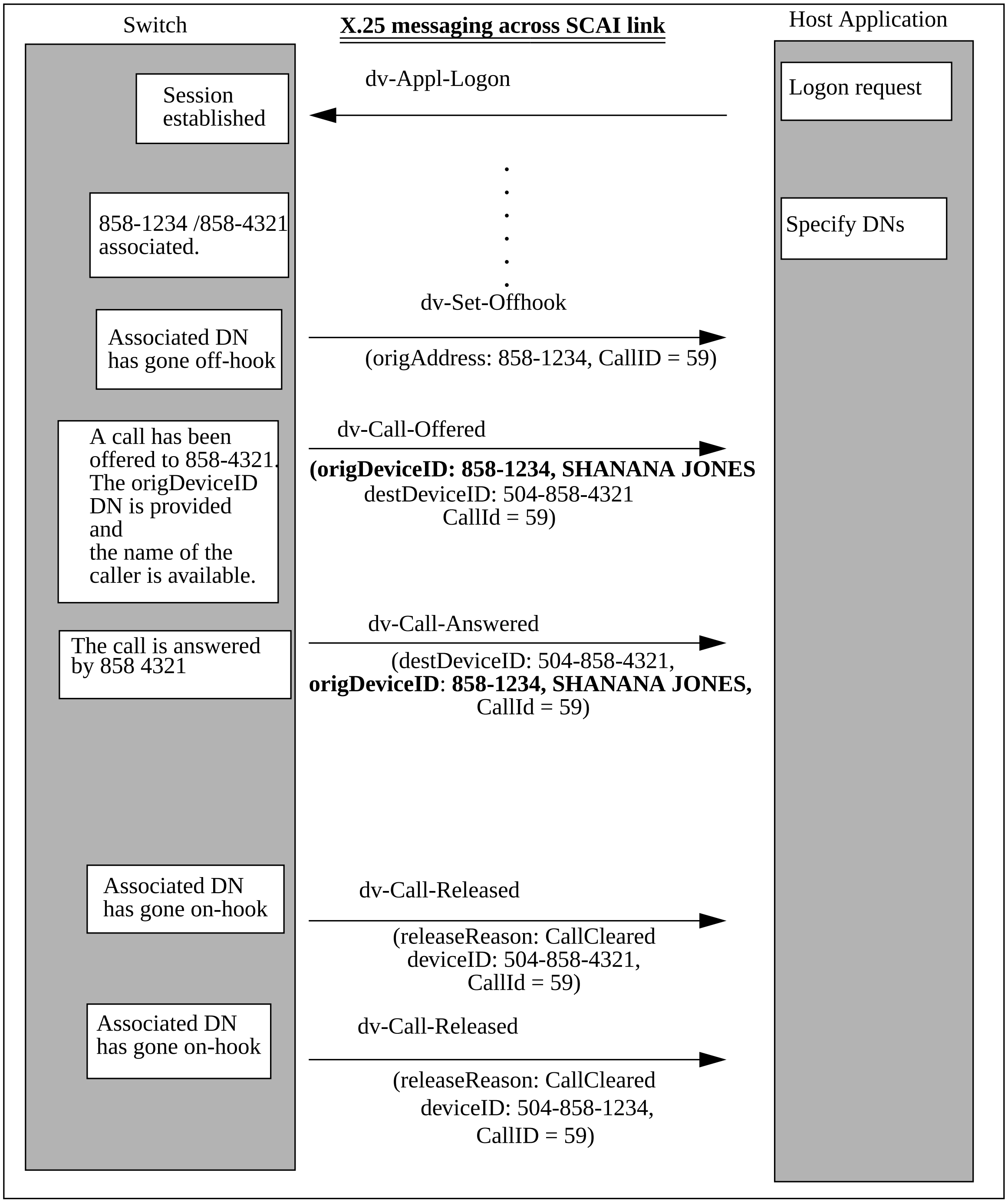
6.5.1 Normal call

This example shows a switch-host session of call event messages. First, the host initiates an application session with the DMS-100. Next, the host associates 858-1234 and 858-4321 with the current session. 858-1234 goes off-hook and calls 858-4321 which is answered.


Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
RETURN-RESULT
858-1234 associated with session
dv-DN-Associate (DN:858-1234,ADD)
Specify DN
RETURN-RESULT
858-4321 associated with session
dv-DN-Associate (DN:858-4321,ADD)
Specify DN
RETURN-RESULT
Associated DN has gone off-hook
dv-Set-Offhook (origAddress:858-1234)
A call has been offered to 858-4321. The origDeviceID DN is provided in dial format and the name of the caller is blocked.
dv-Call-Offered
(origDeviceID: 858-1234, destDeviceID: 504-858-4321)
The call is answered by 858 4321
dv-Call-Answered
(destDeviceID: 504-858-4321 origDeviceID: 858-1234)
dv-Call-Released
(releaseReason: CallCleared deviceID: 504-858-4321)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared deviceID: 504-858-1234)
Associated DN has gone on-hook

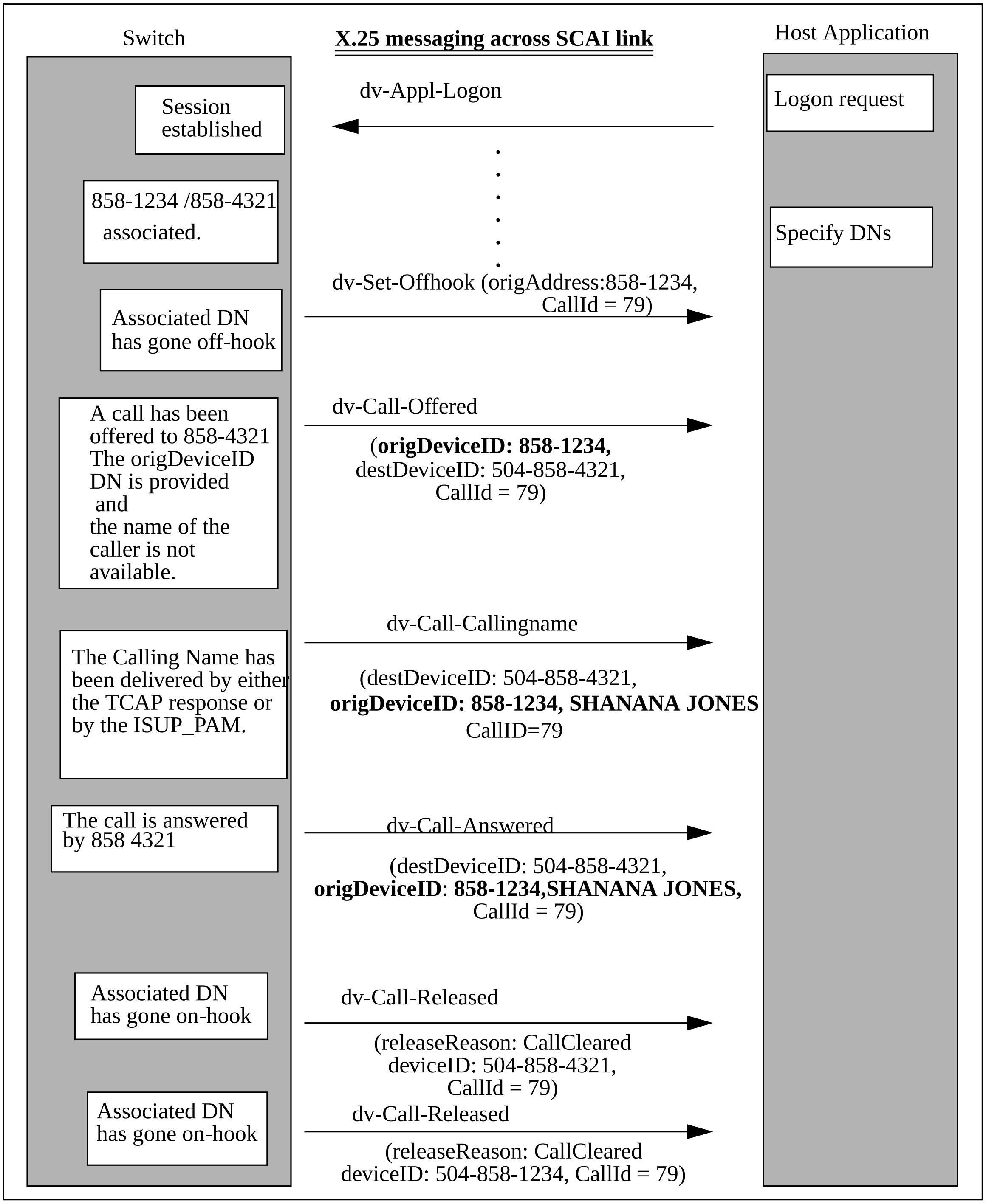
6.5.2 Normal call without Callname message

The following example shows a switch-host session of the call event messages for the above cases. First, the host initiates an application session with the DMS-100. Next, the host associates 858-1234 and associates 858-4321 with the current session. The associated DN, 858-1234, goes off-hook and calls 858-4321. 858-4321 is answered.


Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
858-1234 /858-4321 associated.
Specify DNs
dv-Set-Offhook
Associated DN has gone off-hook
(origAddress: 858-1234, CallID = 59)
dv-Call-Offered
A call has been offered to 858-4321. The origDeviceID DN is provided and the name of the caller is available.
(origDeviceID: 858-1234, SHANANA JONES
destDeviceID: 504-858-4321
CallId = 59)
dv-Call-Answered
The call is answered by 858 4321
(destDeviceID: 504-858-4321,
origDeviceID: 858-1234, SHANANA JONES,
CallId = 59)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared
deviceID: 504-858-4321,
CallId = 59)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared
deviceID: 504-858-1234,
CallID = 59)

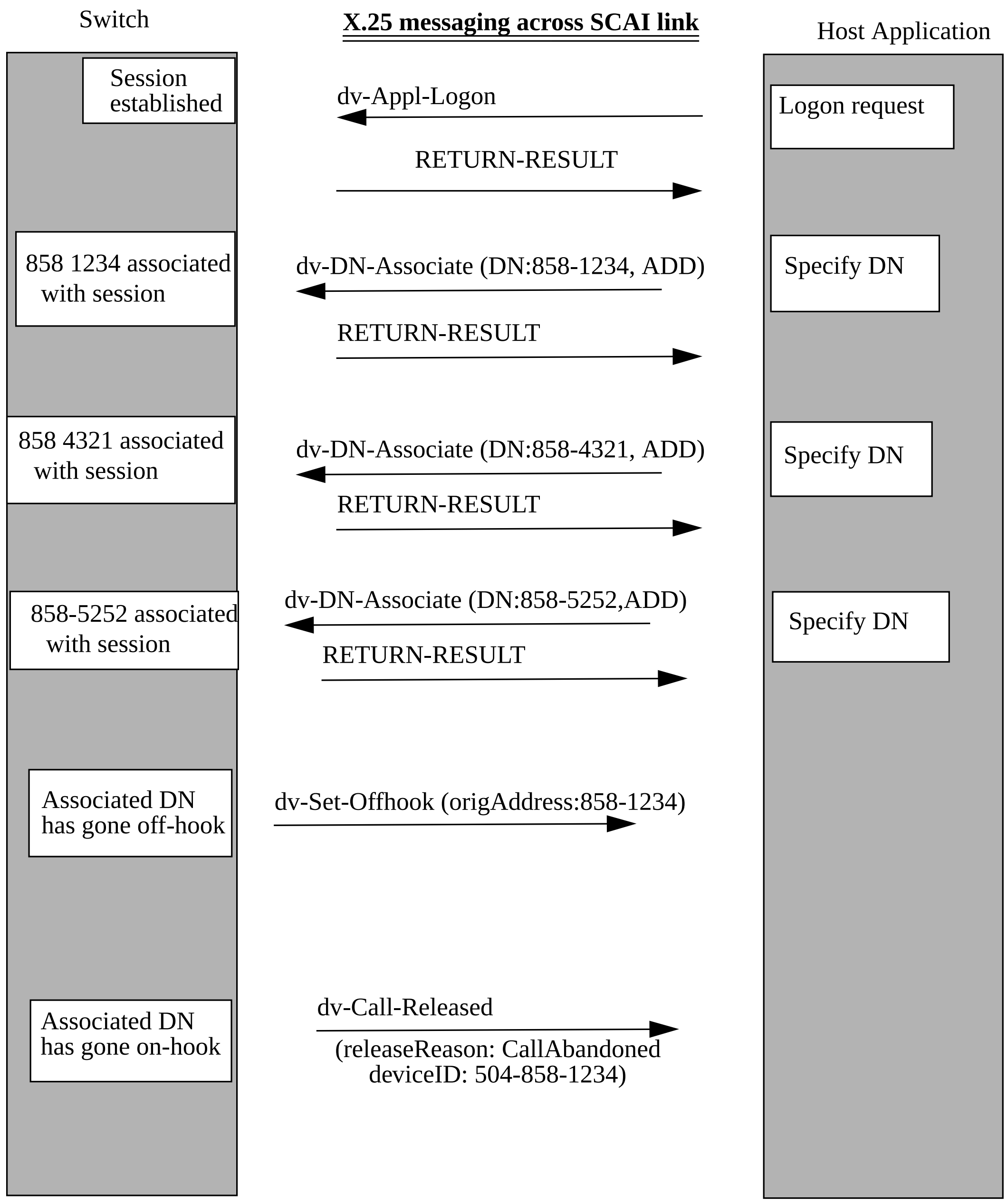
6.5.3 Normal call with Callname message

The following example shows a switch-host session of the call event messages. First, the host initiates an application session with the DMS-100. Next, the host associates 858-1234 and associates 858-4321 with the current session. The associated DN, 858-1234, goes off-hook and calls 858-4321. 858-4321 is answered.


Switch
X.25 messaging across SCAI link
Host Application
dv-Appl-Logon
Session established
Logon request
858-1234 /858-4321 associated.
Specify DNs
dv-Set-Offhook (origAddress:858-1234, CallId = 79)
Associated DN has gone off-hook
dv-Call-Offered
A call has been offered to 858-4321 The origDeviceID DN is provided and the name of the caller is not available.
(origDeviceID: 858-1234, destDeviceID: 504-858-4321, CallId = 79)
dv-Call-Callingname
The Calling Name has been delivered by either the TCAP response or by the ISUP_PAM.
(destDeviceID: 504-858-4321, origDeviceID: 858-1234, SHANANA JONES CallID=79
The call is answered by 858 4321
dv-Call-Answered
(destDeviceID: 504-858-4321, origDeviceID: 858-1234,SHANANA JONES, CallId = 79)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared deviceID: 504-858-4321, CallId = 79)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared deviceID: 504-858-1234, CallId = 79)

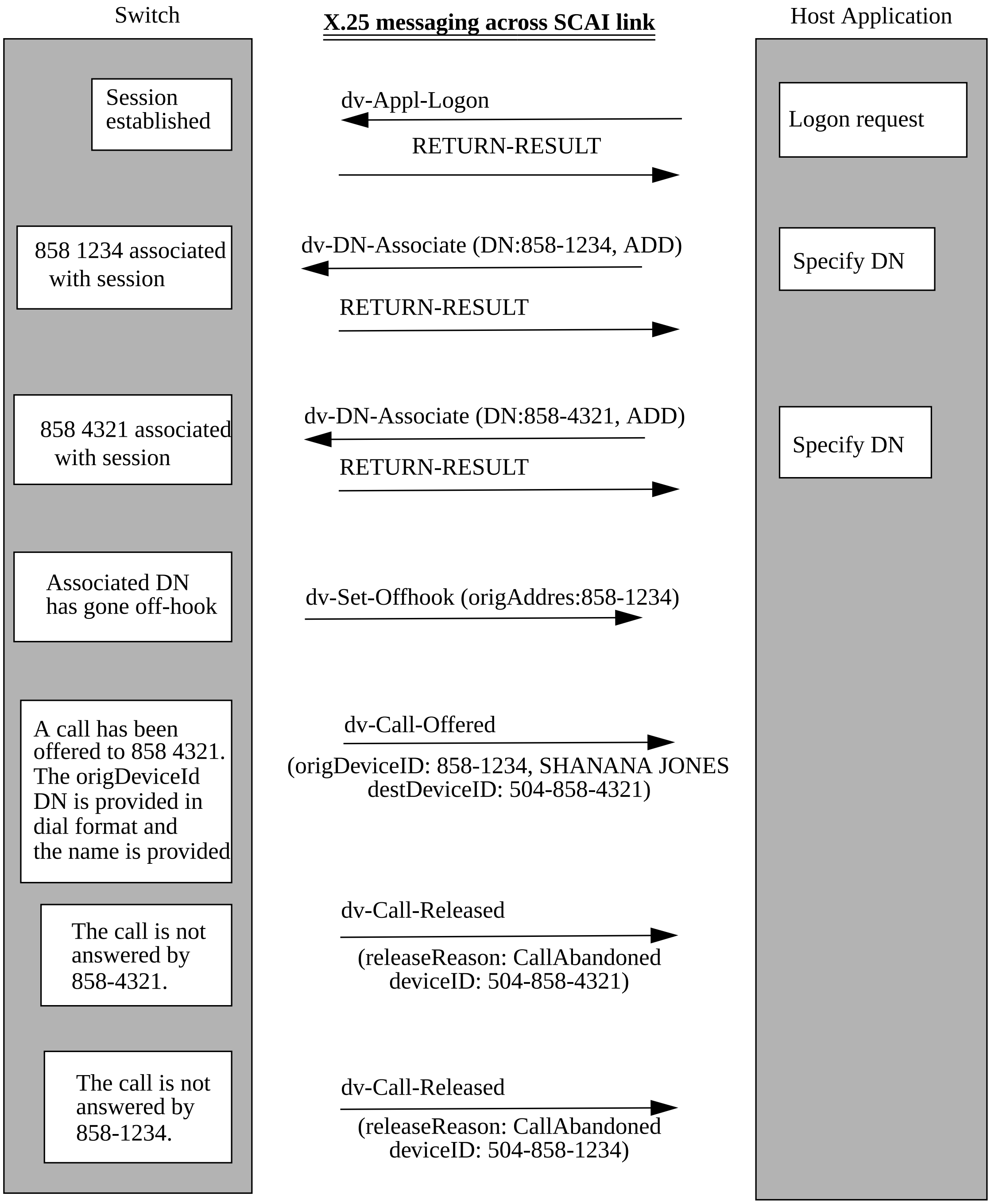
6.5.4 Busy line interactions

This example shows a switch-host session of a call to a busy line. The host initiates an application session with the DMS and associates 858-4321, 858-1234, 858-5252. 858-1234 goes off-hook and calls 858-4321which is involved in another call, so no SCAI message is sent.


Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
RETURN-RESULT
858 1234 associated with session
dv-DN-Associate (DN:858-1234, ADD)
Specify DN
RETURN-RESULT
858 4321 associated with session
dv-DN-Associate (DN:858-4321, ADD)
Specify DN
RETURN-RESULT
858-5252 associated with session
dv-DN-Associate (DN:858-5252,ADD)
Specify DN
RETURN-RESULT
Associated DN has gone off-hook
dv-Set-Offhook (origAddress:858-1234)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallAbandoned deviceID: 504-858-1234)

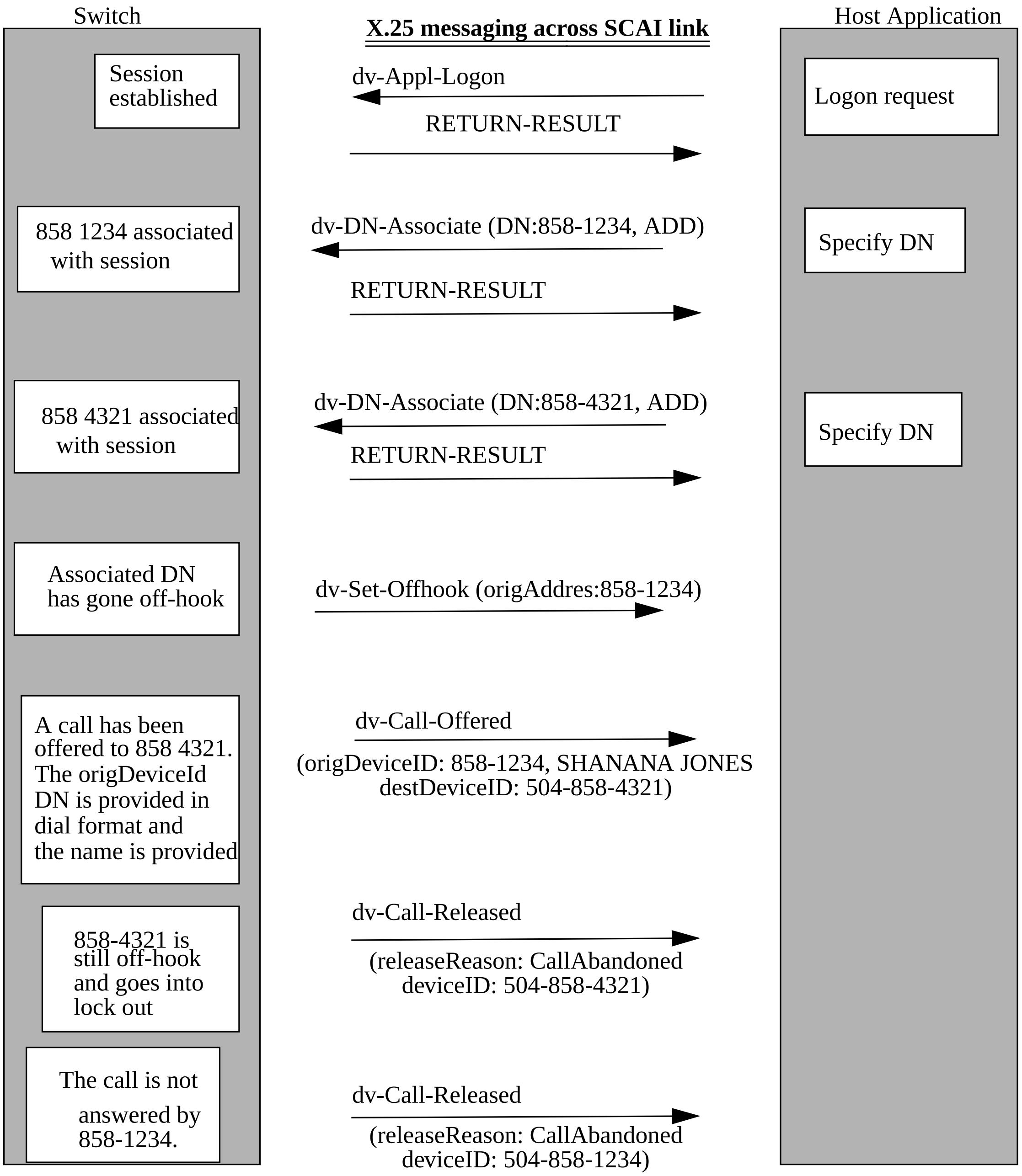
6.5.5 Call not answered

The following example shows a switch-host session of the call event messages. First, the host initiates an application session with the DMS-100 and associates 858-4321 and 858-1234 with the current session. 858-1234, goes off-hook and calls 858-4321. The person at 858-4321 does not answer the call.

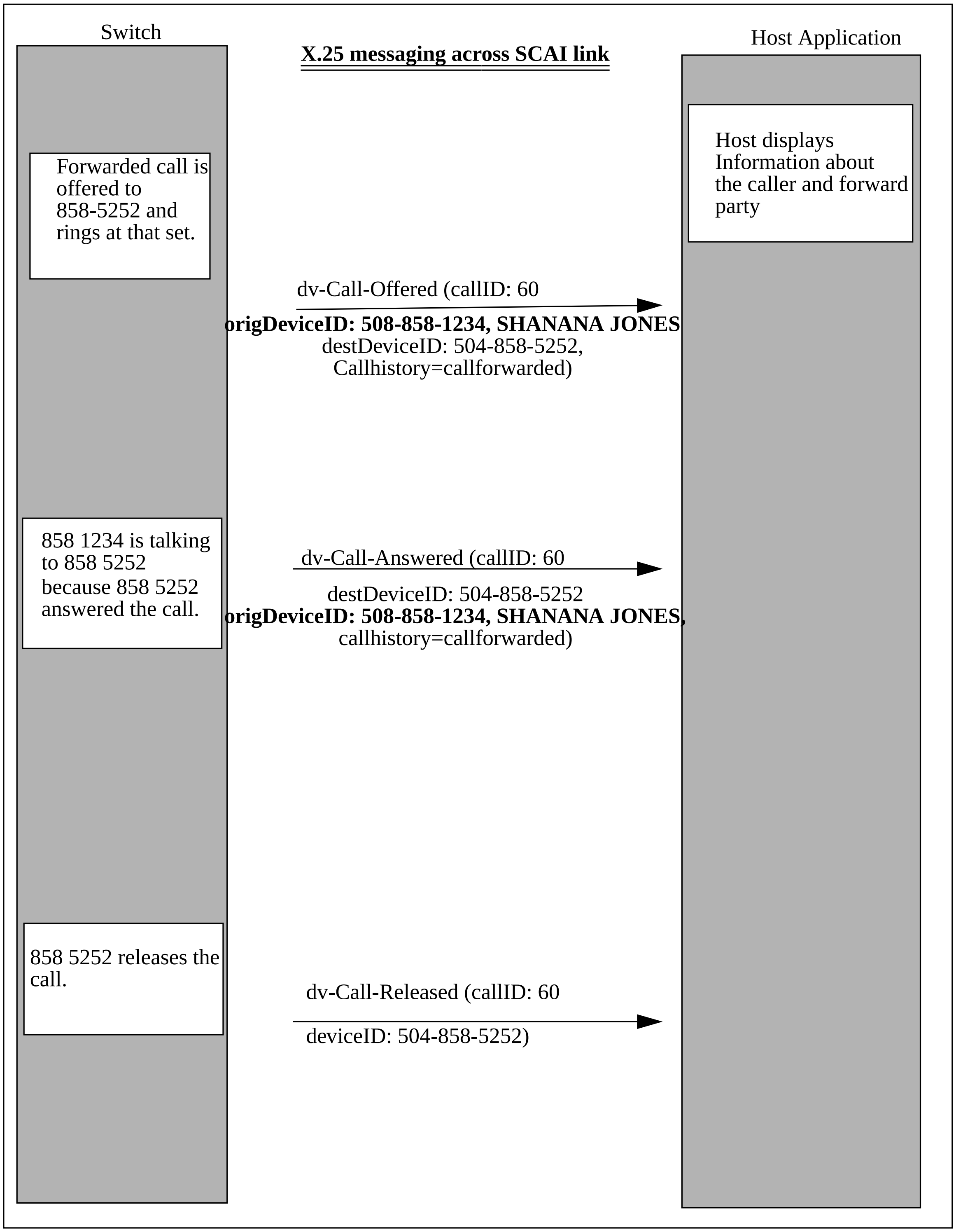

Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
RETURN-RESULT
Logon request
858 1234 associated with session
dv-DN-Associate (DN:858-1234, ADD)
RETURN-RESULT
Specify DN
858 4321 associated with session
dv-DN-Associate (DN:858-4321, ADD)
RETURN-RESULT
Specify DN
Associated DN has gone off-hook
dv-Set-Offhook (origAddres:858-1234)
A call has been offered to 858 4321. The origDeviceId DN is provided in dial format and the name is provided
dv-Call-Offered
(origDeviceID: 858-1234, SHANANA JONES destDeviceID: 504-858-4321)
The call is not answered by 858-4321.
dv-Call-Released
(releaseReason: CallAbandoned deviceID: 504-858-4321)
The call is not answered by 858-1234.
dv-Call-Released
(releaseReason: CallAbandoned deviceID: 504-858-1234)

6.5.6 Lock-out

The host initiates an application session with the DMS and associates 858-4321 and 858-1234 with the current session. 858-1234 goes off-hook and calls 858-4321. 858-4321 answers the call but 858-4321 remains off-hook. 858-4321 receives a lock-out treatment for remaining off-hook. The call released message is sent to the host when a call is released, not when a phone goes on-hook. 858-1234 goes on-hook but 858-4321 remains off-hook. 858-4321 receives a lock-out treatment for remaining off-hook. The Call Released message is sent to the host when a call is released, not when a phone goes on-hook.

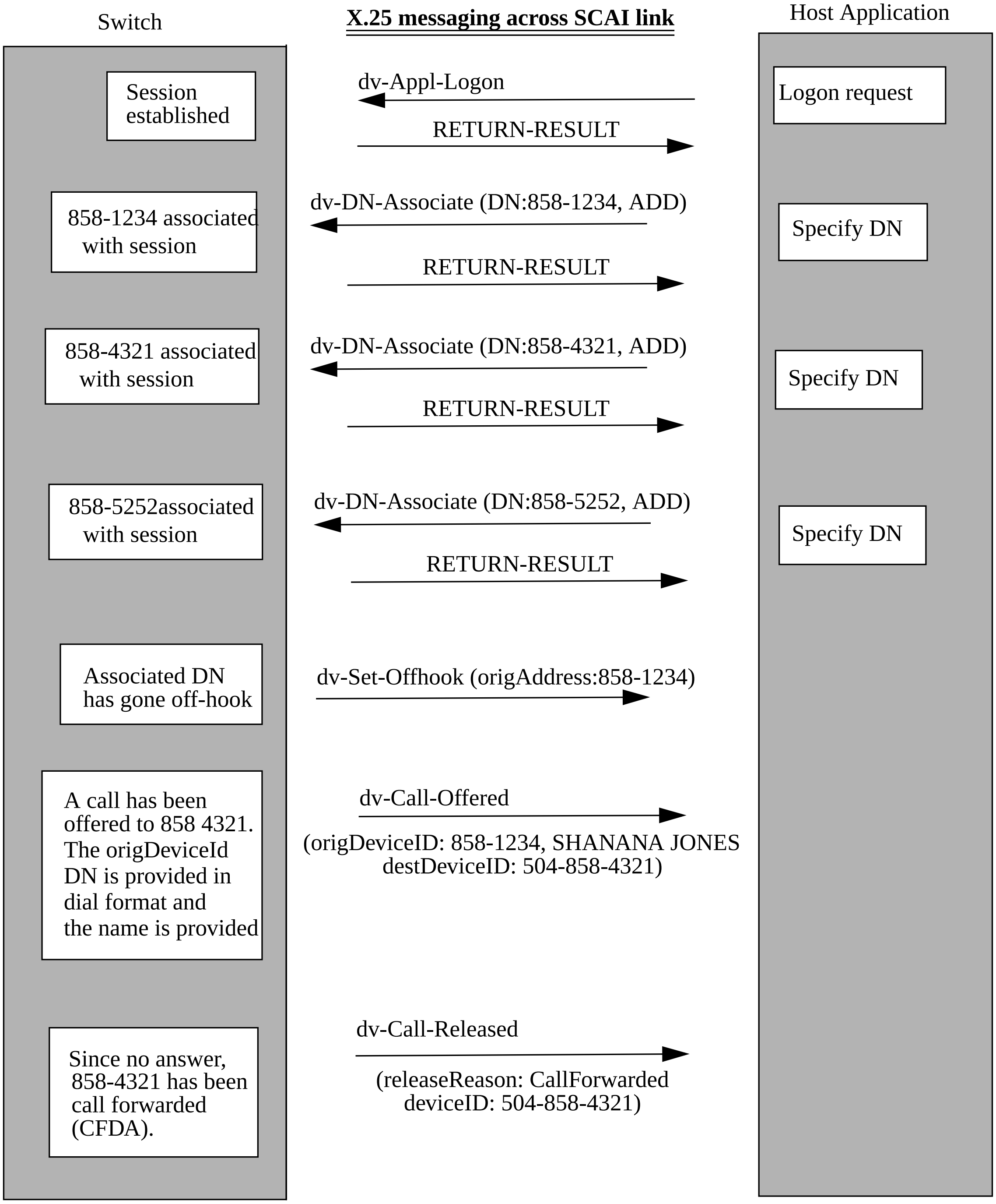

Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
RETURN-RESULT
858 1234 associated with session
dv-DN-Associate (DN:858-1234, ADD)
Specify DN
RETURN-RESULT
858 4321 associated with session
dv-DN-Associate (DN:858-4321, ADD)
Specify DN
RETURN-RESULT
Associated DN has gone off-hook
dv-Set-Offhook (origAddres:858-1234)
A call has been offered to 858 4321. The origDeviceId DN is provided in dial format and the name is provided
dv-Call-Offered
(origDeviceID: 858-1234, SHANANA JONES destDeviceID: 504-858-4321)
858-4321 is still off-hook and goes into lock out
dv-Call-Released
(releaseReason: CallAbandoned deviceID: 504-858-4321)
The call is not answered by 858-1234.
dv-Call-Released
(releaseReason: CallAbandoned deviceID: 504-858-1234)

6.5.7 Callforwarded call

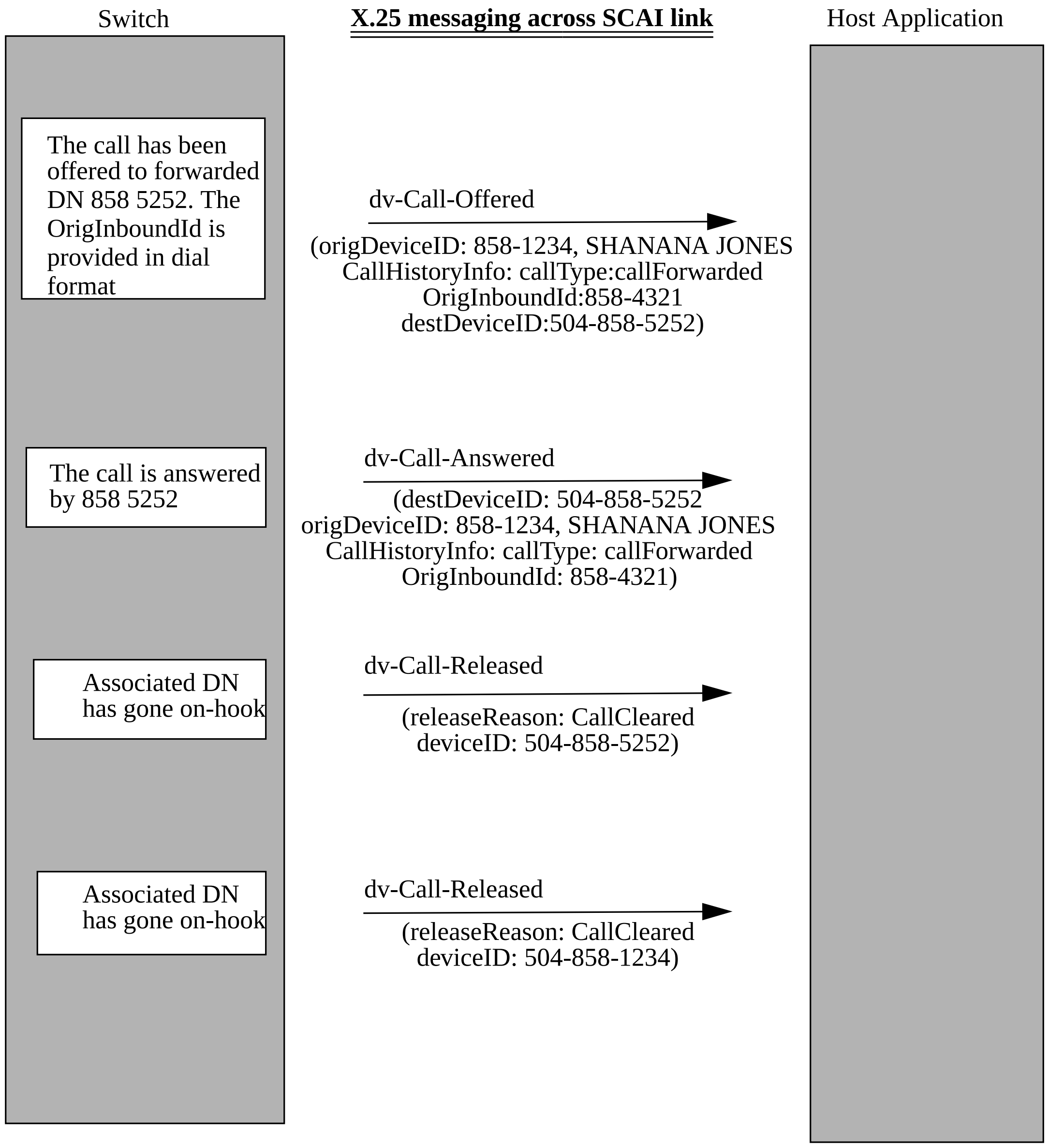

Switch
X.25 messaging across SCAI link
Host Application
Forwarded call is offered to 858-5252 and rings at that set.
Host displays Information about the caller and forward party
dv-Call-Offered (callID: 60
origDeviceID: 508-858-1234, SHANANA JONES
destDeviceID: 504-858-5252,
Callhistory=callforwarded)
858 1234 is talking to 858 5252 because 858 5252 answered the call.
dv-Call-Answered (callID: 60
destDeviceID: 504-858-5252
origDeviceID: 508-858-1234, SHANANA JONES,
callhistory=callforwarded)
858 5252 releases the call.
dv-Call-Released (callID: 60
deviceID: 504-858-5252)

6.5.8 Call forward don’t answer

This example shows a switch-host session of a Call Forward Don’t Answer call. The host initiates an application session with the DMS and associates 858-4321, 858-1234 and 858-5252. 858-1234 goes off-hook and calls 858-4321. 858-4321 does not answer, so the call is forwarded to 858-5252 which answers the call.

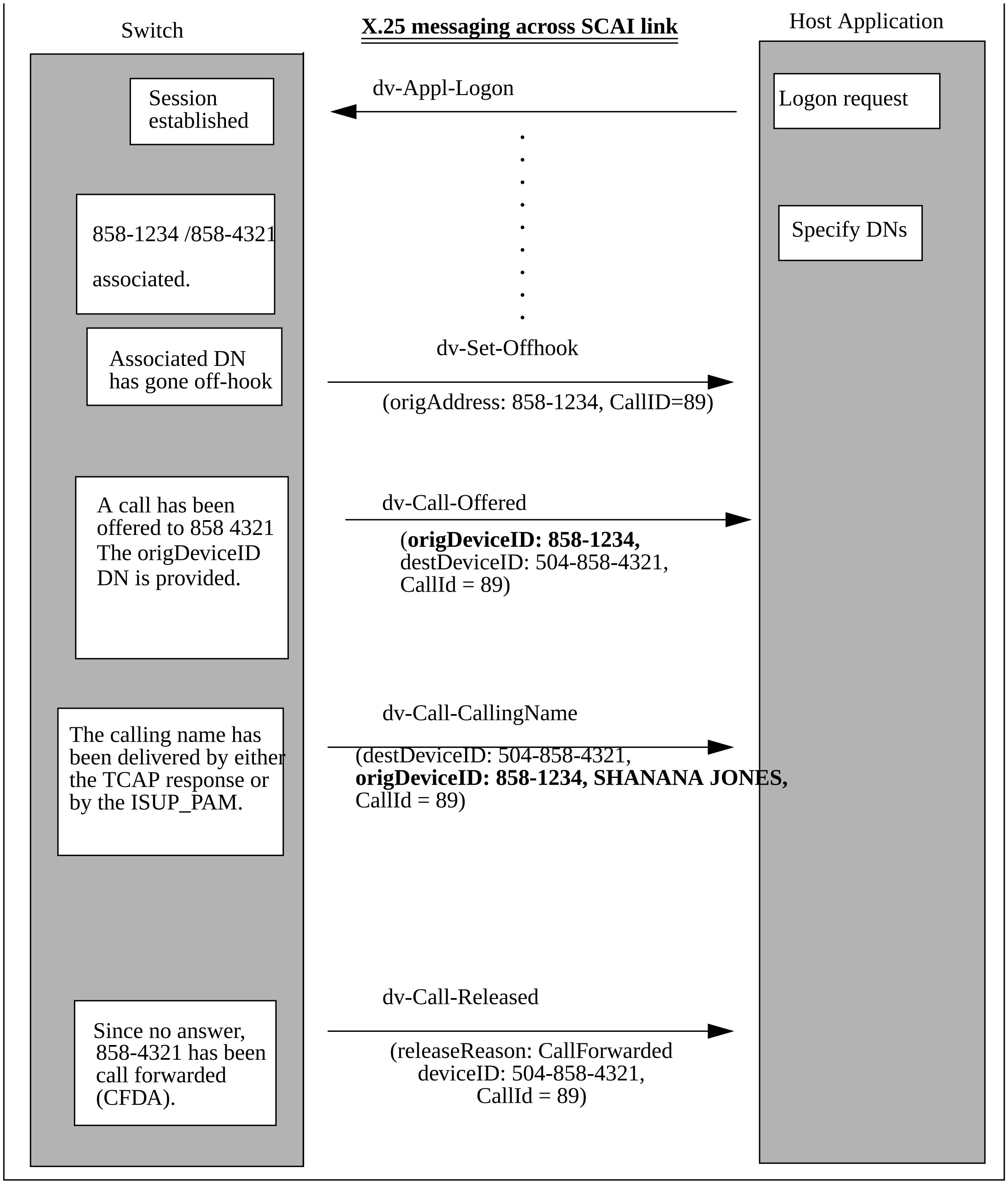

Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
RETURN-RESULT
858-1234 associated with session
dv-DN-Associate (DN:858-1234, ADD)
Specify DN
RETURN-RESULT
858-4321 associated with session
dv-DN-Associate (DN:858-4321, ADD)
Specify DN
RETURN-RESULT
858-5252associated with session
dv-DN-Associate (DN:858-5252, ADD)
Specify DN
RETURN-RESULT
Associated DN has gone off-hook
dv-Set-Offhook (origAddress:858-1234)
A call has been offered to 858 4321. The origDeviceId DN is provided in dial format and the name is provided
dv-Call-Offered
(origDeviceID: 858-1234, SHANANA JONES destDeviceID: 504-858-4321)
Since no answer, 858-4321 has been call forwarded (CFDA).
dv-Call-Released
(releaseReason: CallForwarded deviceID: 504-858-4321)

Call forward don’t answer(Continued)


Switch
X.25 messaging across SCAI link
Host Application
The call has been offered to forwarded DN 858 5252. The OrigInboundId is provided in dial format
dv-Call-Offered
(origDeviceID: 858-1234, SHANANA JONES
CallHistoryInfo: callType:callForwarded
OrigInboundId:858-4321
destDeviceID:504-858-5252)
The call is answered by 858 5252
dv-Call-Answered
(destDeviceID: 504-858-5252
origDeviceID: 858-1234, SHANANA JONES
CallHistoryInfo: callType: callForwarded
OrigInboundId: 858-4321)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared
deviceID: 504-858-5252)
Associated DN has gone on-hook
dv-Call-Released
(releaseReason: CallCleared
deviceID: 504-858-1234)

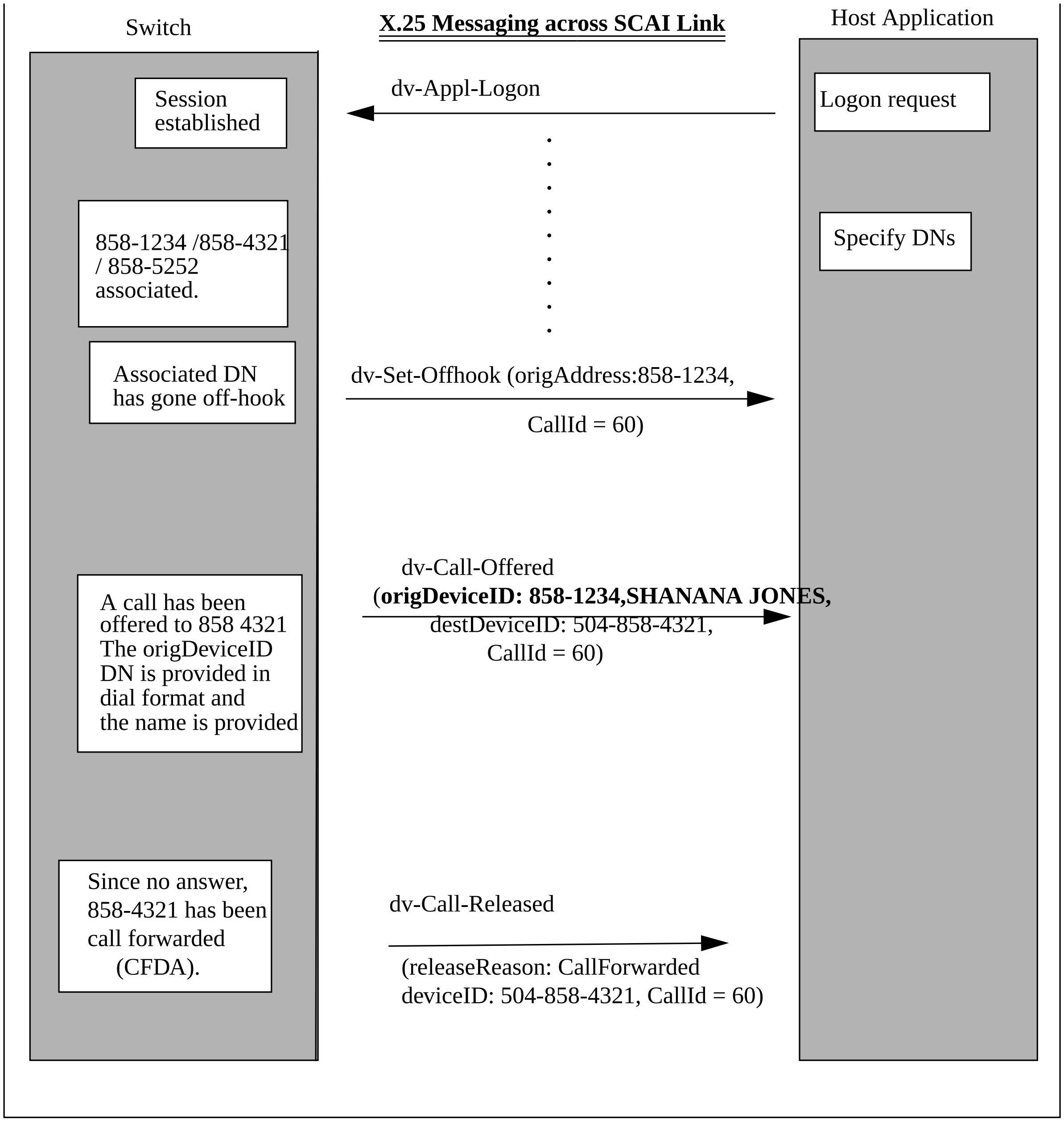
6.5.9 Call Forward Don’t Answer with CallingName

The following example shows a switch-host session of a Call Forward Don’t Answer call. First, the host initiates an application session with the DMS-100 and associates 858-4321, 858-1234 and 858-5252. 858-1234 goes off-hook and calls 858-4321. Assume the Callname is available at the time of sending the call offered message.The call-offered message is sent.


Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
858-1234 /858-4321 associated.
Specify DNs
Associated DN has gone off-hook
dv-Set-Offhook
(origAddress: 858-1234, CallID=89)
A call has been offered to 858 4321 The origDeviceID DN is provided.
dv-Call-Offered
(**origDeviceID: 858-1234,** destDeviceID: 504-858-4321, CallId = 89)
The calling name has been delivered by either the TCAP response or by the ISUP_PAM.
dv-Call-CallingName
(destDeviceID: 504-858-4321, **origDeviceID: 858-1234, SHANANA JONES,** CallId = 89)
Since no answer, 858-4321 has been call forwarded (CFDA).
dv-Call-Released
(releaseReason: CallForwarded deviceID: 504-858-4321, CallId = 89)

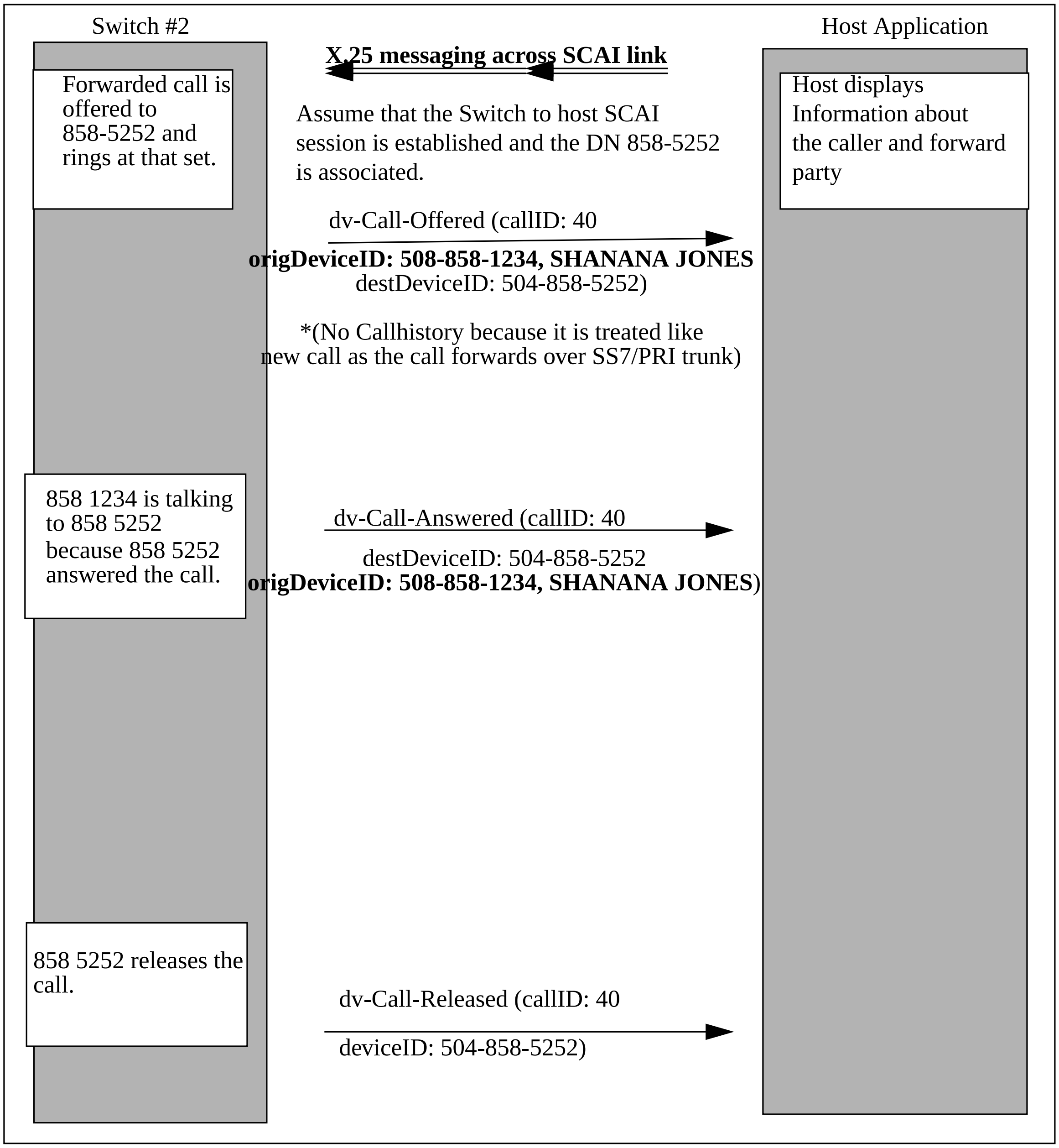
6.5.10 Call Forward without CallingName

The following example shows a switch-host session of a Call Forward Don’t Answer call. First, the host initiates an application session with the first DMS-100 and associates 858-4321 and 858-1234; the host also initiates an application session with the second DMS-100 and associates 858-5252; 858-1234 goes off-hook and calls 858-4321. Assume the Callname is NOT available at the time of sending the call offered message.The call-offered message is sent. The dv-Call-Callingname-U message is also sent. However, the person at 858-4321 does not answer, so the call is forwarded to 858-5252 over an ISUP trunk. The person at 858-5252 answers the call.


Switch
X.25 Messaging across SCAI Link
Host Application
Session established
dv-Appl-Logon
Logon request
858-1234 /858-4321 / 858-5252 associated.
Specify DNs
Associated DN has gone off-hook
dv-Set-Offhook (origAddress:858-1234,
CallId = 60)
A call has been offered to 858 4321 The origDeviceID DN is provided in dial format and the name is provided
dv-Call-Offered
(**origDeviceID: 858-1234,SHANANA JONES,**
destDeviceID: 504-858-4321,
CallId = 60)
Since no answer, 858-4321 has been call forwarded (CFDA).
dv-Call-Released
(releaseReason: CallForwarded
deviceID: 504-858-4321, CallId = 60)

6.5.11Inter-switch forwarded call

The call history Iinformation in the dv-Call-Offered-U and the dv-Call-Answered-U messages are lost for calls that are forwarded to another switch (i.e.. over SS7/PRI network) and only consist of the information pertaining to the call after the forwarding took place.



Forwarded call is offered to 858-5252 and rings at that set.

Host displays Information about the caller and forward party

Assume that the Switch to host SCAI session is established and the DN 858-5252 is associated.

dv-Call-Offered (callID: 40
origDeviceID: 508-858-1234, SHANANA JONES
destDeviceID: 504-858-5252)

*(No Callhistory because it is treated like new call as the call forwards over SS7/PRI trunk)

858 1234 is talking to 858 5252 because 858 5252 answered the call.

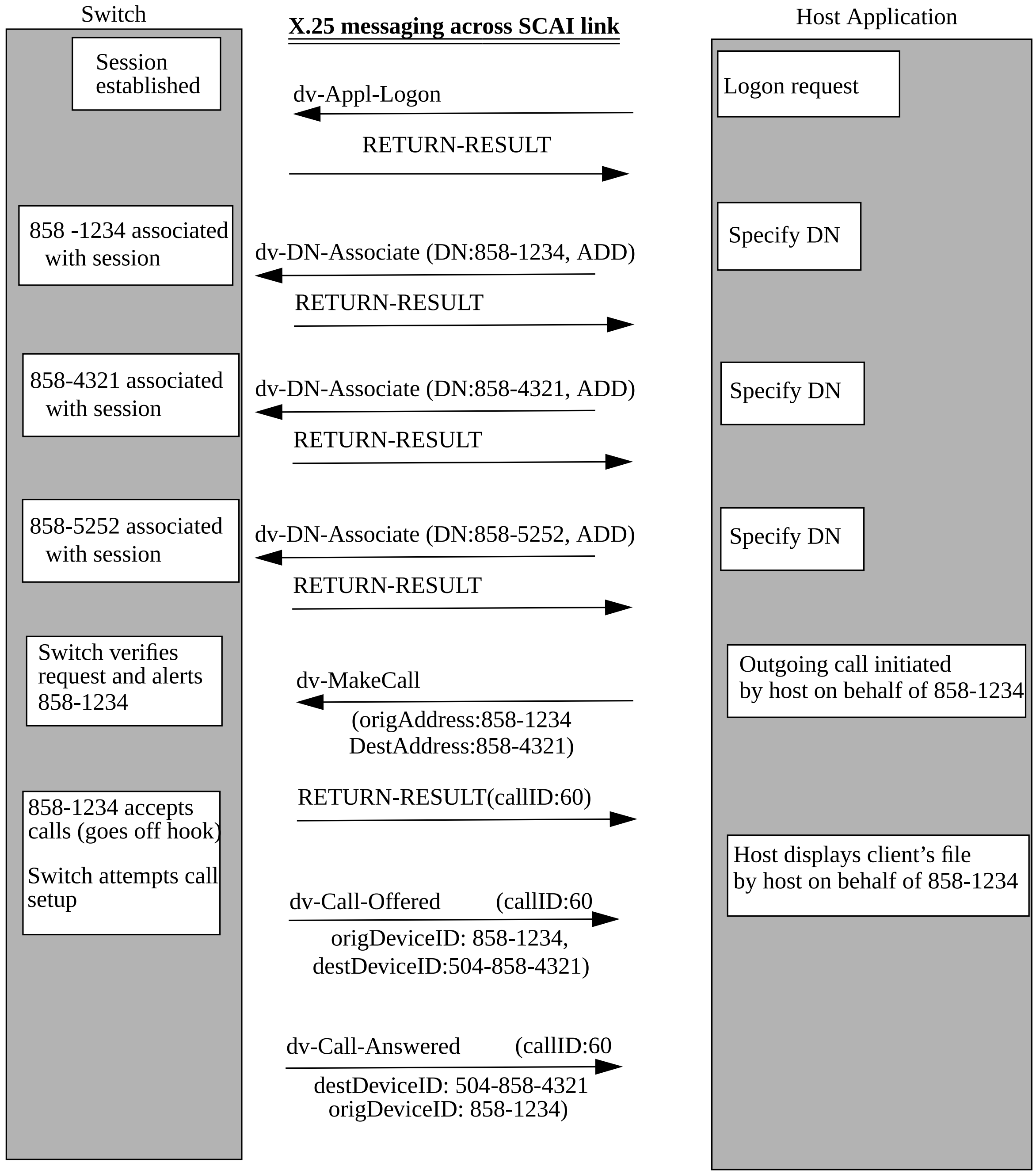
dv-Call-Answered (callID: 40
destDeviceID: 504-858-5252
origDeviceID: 508-858-1234, SHANANA JONES)

858 5252 releases the call.

dv-Call-Released (callID: 40
deviceID: 504-858-5252)

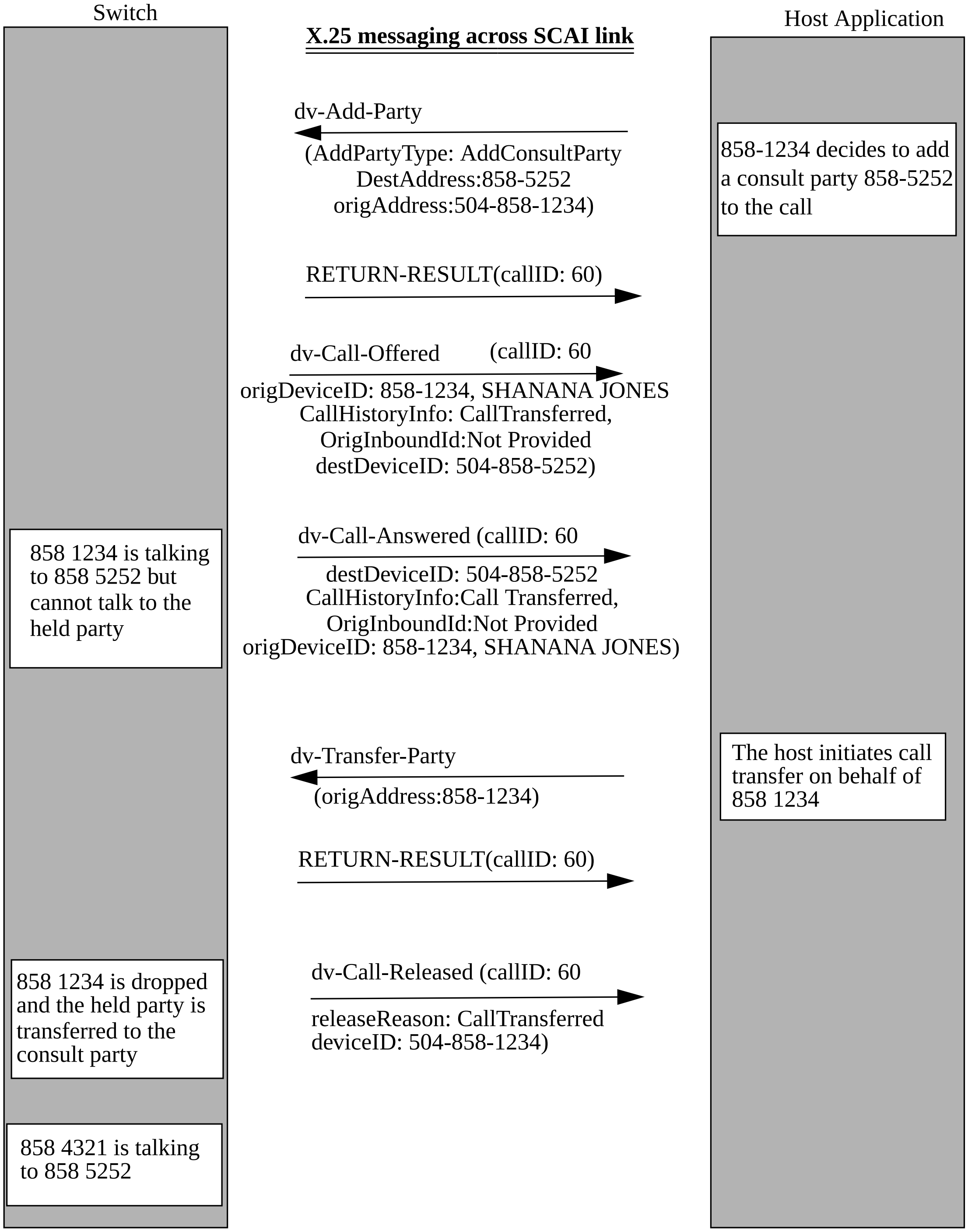
6.5.12 Third Party Control

First, the host initiates an application session with the DMS-100 and associates 858-4321, 858-1234, and 858-5252. The host initiates an outgoing call on the behalf of 858-1234. The call is answered by 858-4321. The host decides to add a consult party 858-5252 to the call on behalf of 858-1234.


Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
RETURN-RESULT
858 -1234 associated with session
dv-DN-Associate (DN:858-1234, ADD)
Specify DN
RETURN-RESULT
858-4321 associated with session
dv-DN-Associate (DN:858-4321, ADD)
Specify DN
RETURN-RESULT
858-5252 associated with session
dv-DN-Associate (DN:858-5252, ADD)
Specify DN
RETURN-RESULT
Switch verifies request and alerts 858-1234
dv-MakeCall
(origAddress:858-1234 DestAddress:858-4321)
Outgoing call initiated by host on behalf of 858-1234
858-1234 accepts calls (goes off hook)
Switch attempts call setup
RETURN-RESULT(callID:60)
Host displays client's file by host on behalf of 858-1234
dv-Call-Offered (callID:60
origDeviceID: 858-1234,
destDeviceID:504-858-4321)
dv-Call-Answered (callID:60
destDeviceID: 504-858-4321
origDeviceID: 858-1234)

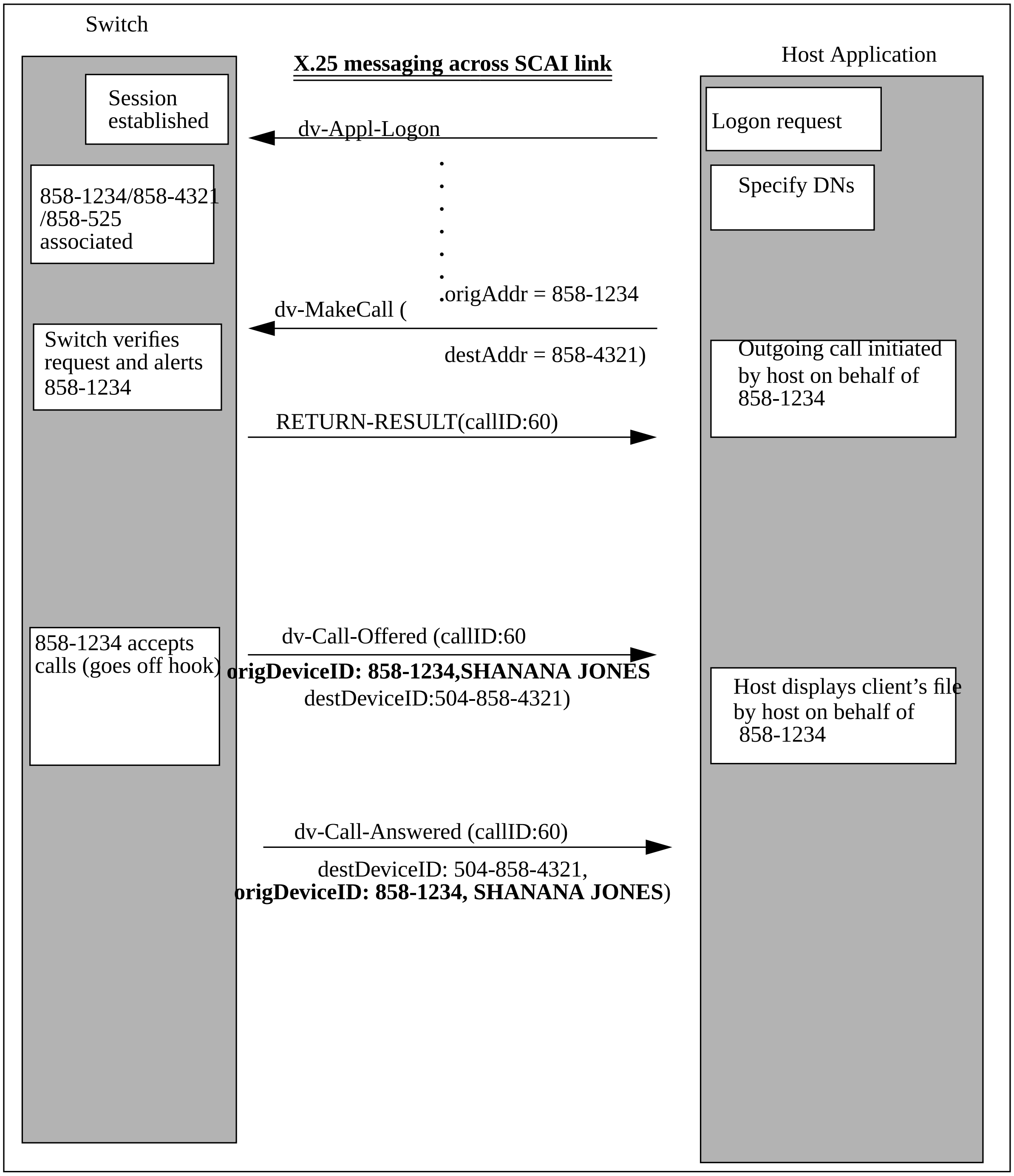
Third Party Control (continued)

The OrigInboundId information in the call history information parameter is not provided in the Call Offered message or the Call Answered message. The person at 858-5252 answers the call. The host initiates call transfer.

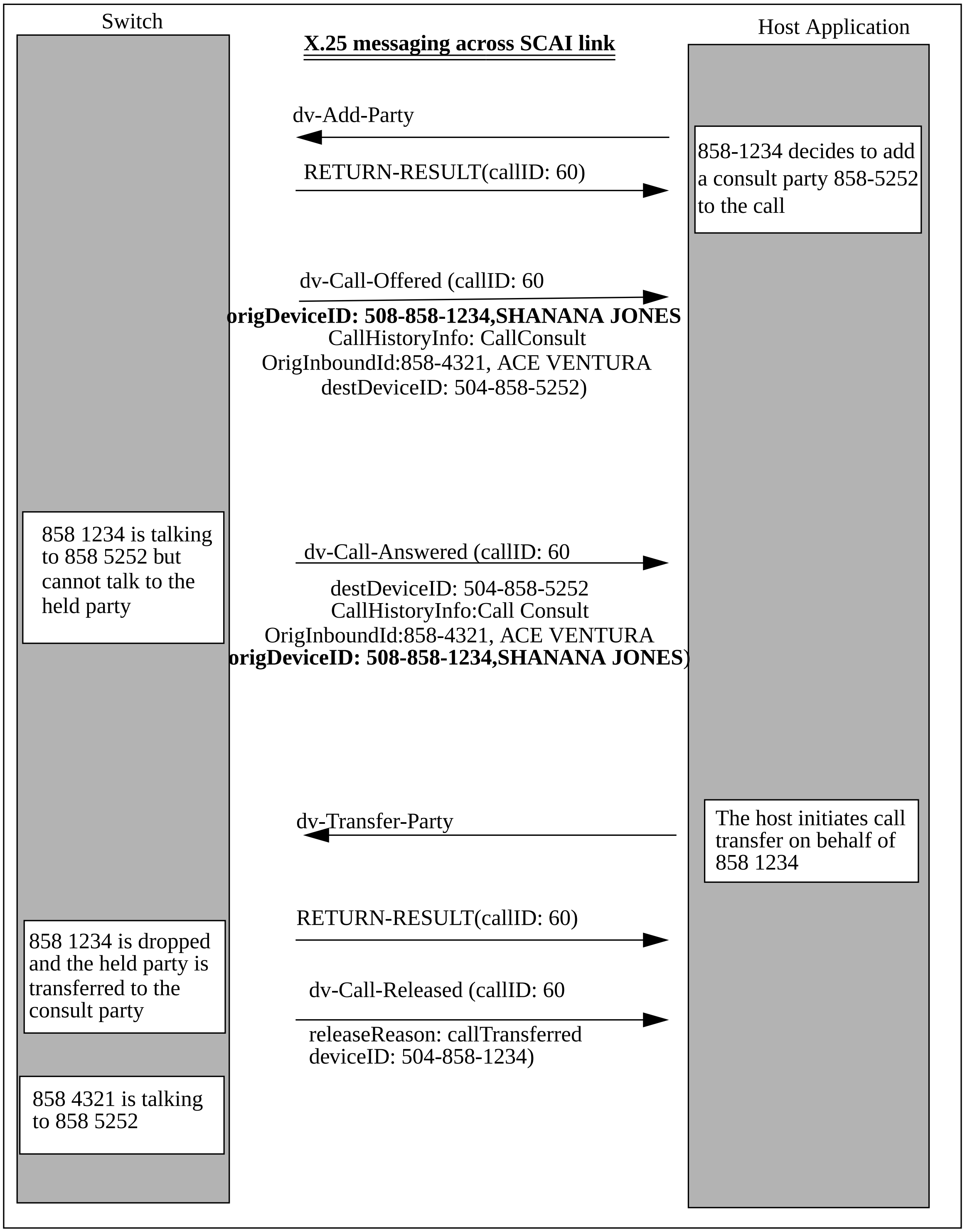

Switch
Host Application
X.25 messaging across SCAI link
dv-Add-Party
(AddPartyType: AddConsultParty
DestAddress:858-5252
origAddress:504-858-1234)
858-1234 decides to add a consult party 858-5252 to the call
RETURN-RESULT(callID: 60)
dv-Call-Offered (callID: 60
origDeviceID: 858-1234, SHANANA JONES
CallHistoryInfo: CallTransferred,
OrigInboundId:Not Provided
destDeviceID: 504-858-5252)
858 1234 is talking to 858 5252 but cannot talk to the held party
dv-Call-Answered (callID: 60
destDeviceID: 504-858-5252
CallHistoryInfo:Call Transferred,
OrigInboundId:Not Provided
origDeviceID: 858-1234, SHANANA JONES)
dv-Transfer-Party
(origAddress:858-1234)
The host initiates call transfer on behalf of 858 1234
RETURN-RESULT(callID: 60)
858 1234 is dropped and the held party is transferred to the consult party
dv-Call-Released (callID: 60
releaseReason: CallTransferred
deviceID: 504-858-1234)
858 4321 is talking to 858 5252

6.5.13 Third Party Control without Callname

The following example shows a switch-host session of a three party control. First, the host initiates an application session with the DMS-100 and associates 858-4321, 858-1234, and 858-5252. The host initiates an outgoing call on the behalf of 858-1234. The call is answered by 858-4321.


Switch
X.25 messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
858-1234/858-4321 /858-525 associated
Specify DNs
origAddr = 858-1234
dv-MakeCall (
destAddr = 858-4321)
Switch verifies request and alerts 858-1234
Outgoing call initiated by host on behalf of 858-1234
RETURN-RESULT(callID:60)
858-1234 accepts calls (goes off hook)
dv-Call-Offered (callID:60
origDeviceID: 858-1234,SHANANA JONES
destDeviceID:504-858-4321)
Host displays client's file by host on behalf of 858-1234
dv-Call-Answered (callID:60)
destDeviceID: 504-858-4321,
origDeviceID: 858-1234, SHANANA JONES)

Third Party Control without Callname (continued)



dv-Add-Party

858-1234 decides to add a consult party 858-5252 to the call

RETURN-RESULT(callID: 60)

dv-Call-Offered (callID: 60
origDeviceID: 508-858-1234,SHANANA JONES
CallHistoryInfo: CallConsult
OrigInboundId:858-4321, ACE VENTURA
destDeviceID: 504-858-5252)

858 1234 is talking to 858 5252 but cannot talk to the held party

dv-Call-Answered (callID: 60
destDeviceID: 504-858-5252
CallHistoryInfo:Call Consult
OrigInboundId:858-4321, ACE VENTURA
origDeviceID: 508-858-1234,SHANANA JONES)

dv-Transfer-Party

The host initiates call transfer on behalf of 858 1234

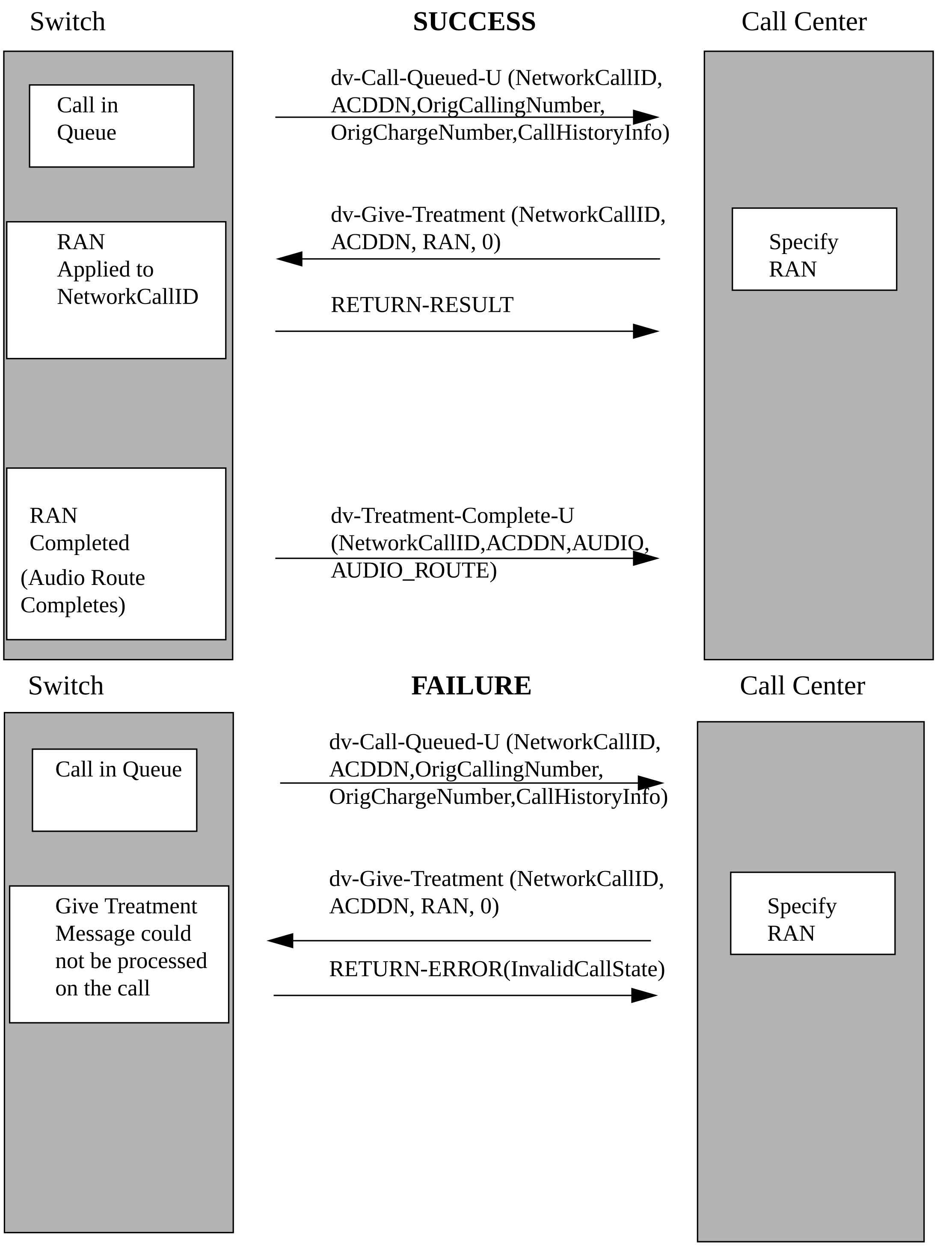
RETURN-RESULT(callID: 60)

858 1234 is dropped and the held party is transferred to the consult party

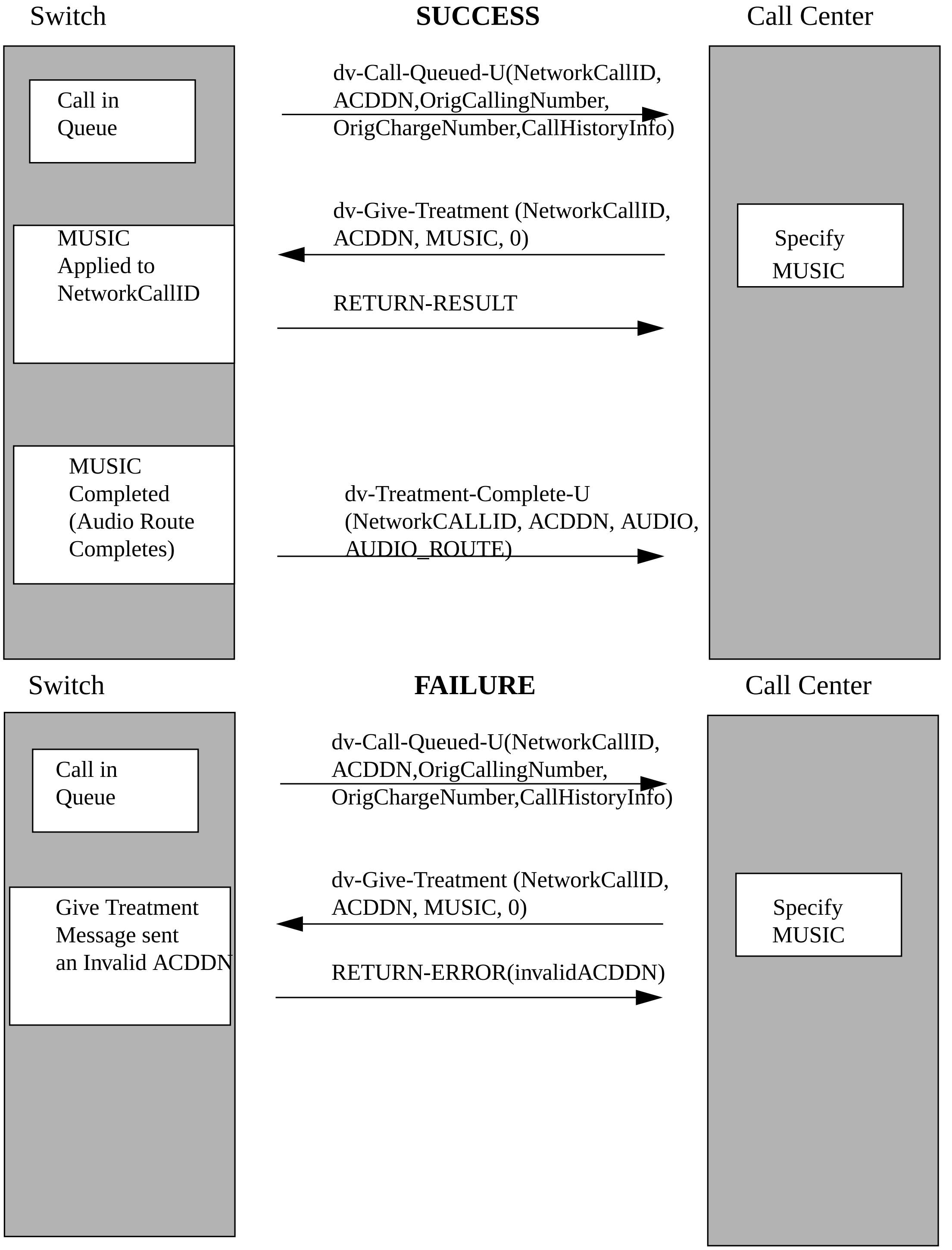
dv-Call-Released (callID: 60
releaseReason: callTransferred
deviceID: 504-858-1234)

858 4321 is talking to 858 5252

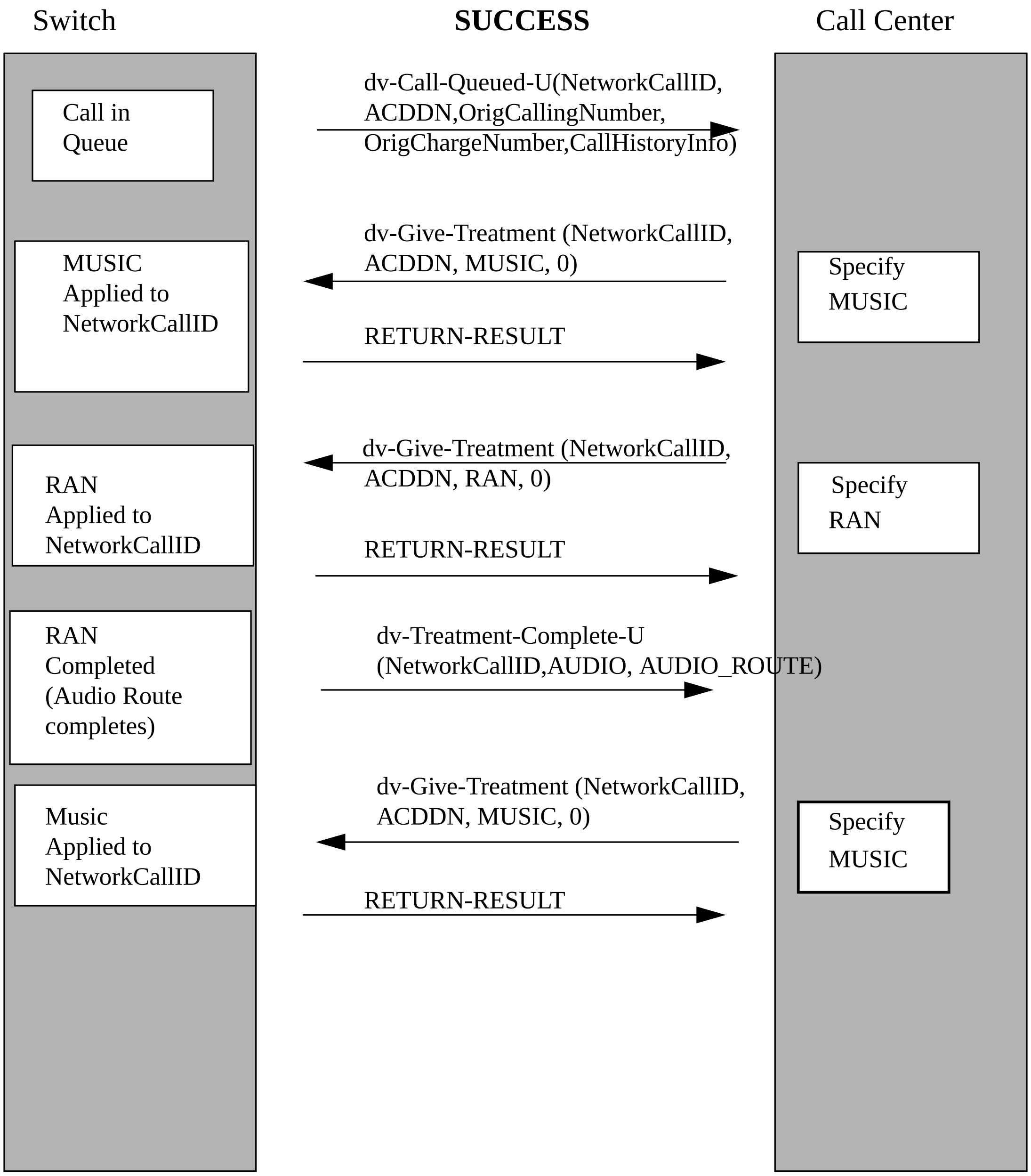
6.5.14Give Treatment (RAN) & Treatment_Complete (RAN)


Switch
SUCCESS
Call Center
Call in Queue
dv-Call-Queued-U (NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN, RAN, 0)
Specify RAN
RAN Applied to NetworkCallID
RETURN-RESULT
RAN Completed
(Audio Route Completes)
dv-Treatment-Complete-U (NetworkCallID,ACDDN,AUDIO, AUDIO_ROUTE)
Switch
FAILURE
Call Center
Call in Queue
dv-Call-Queued-U (NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN, RAN, 0)
Specify RAN
Give Treatment Message could not be processed on the call
RETURN-ERROR(InvalidCallState)

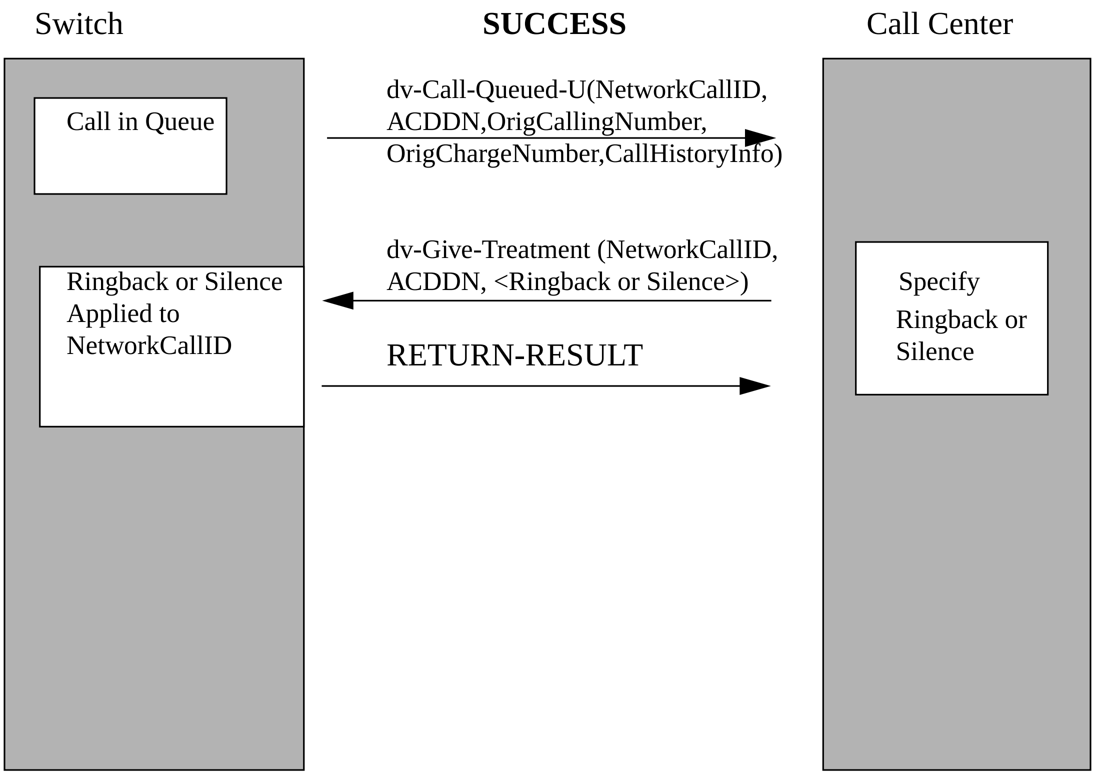
Give Treatment (MUSIC) sample session

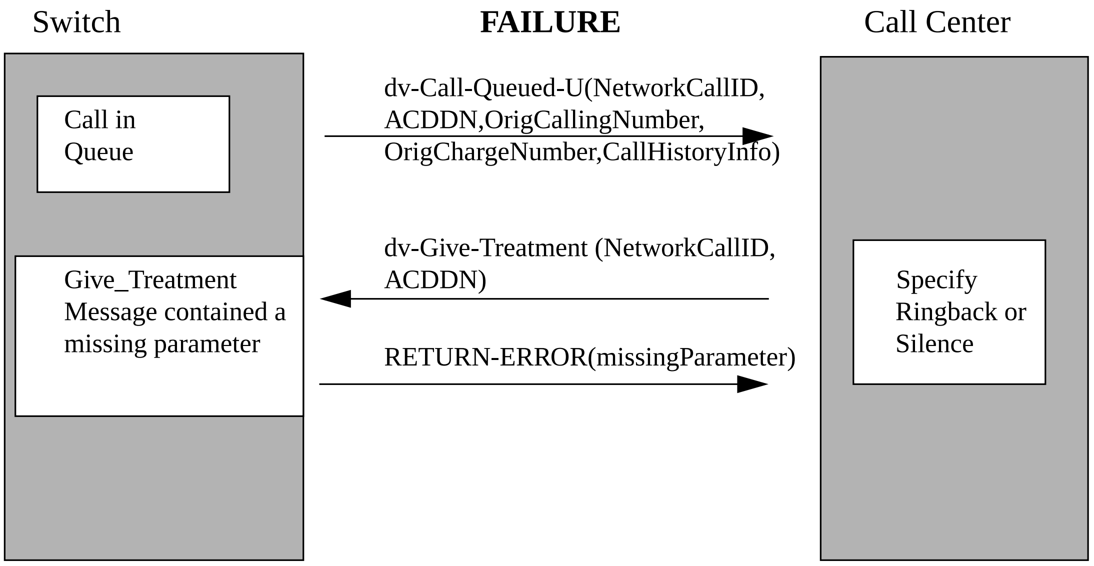

Switch
SUCCESS
Call Center
Call in Queue
dv-Call-Queued-U(NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN, MUSIC, 0)
Specify MUSIC
MUSIC Applied to NetworkCallID
RETURN-RESULT
MUSIC Completed (Audio Route Completes)
dv-Treatment-Complete-U (NetworkCALLID, ACDDN, AUDIO, AUDIO_ROUTE)
Switch
FAILURE
Call Center
Call in Queue
dv-Call-Queued-U(NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN, MUSIC, 0)
Specify MUSIC
Give Treatment Message sent an Invalid ACDDN
RETURN-ERROR(invalidACDDN)

6.5.15 Give Treatment (MUSIC & RAN Interaction) sample session

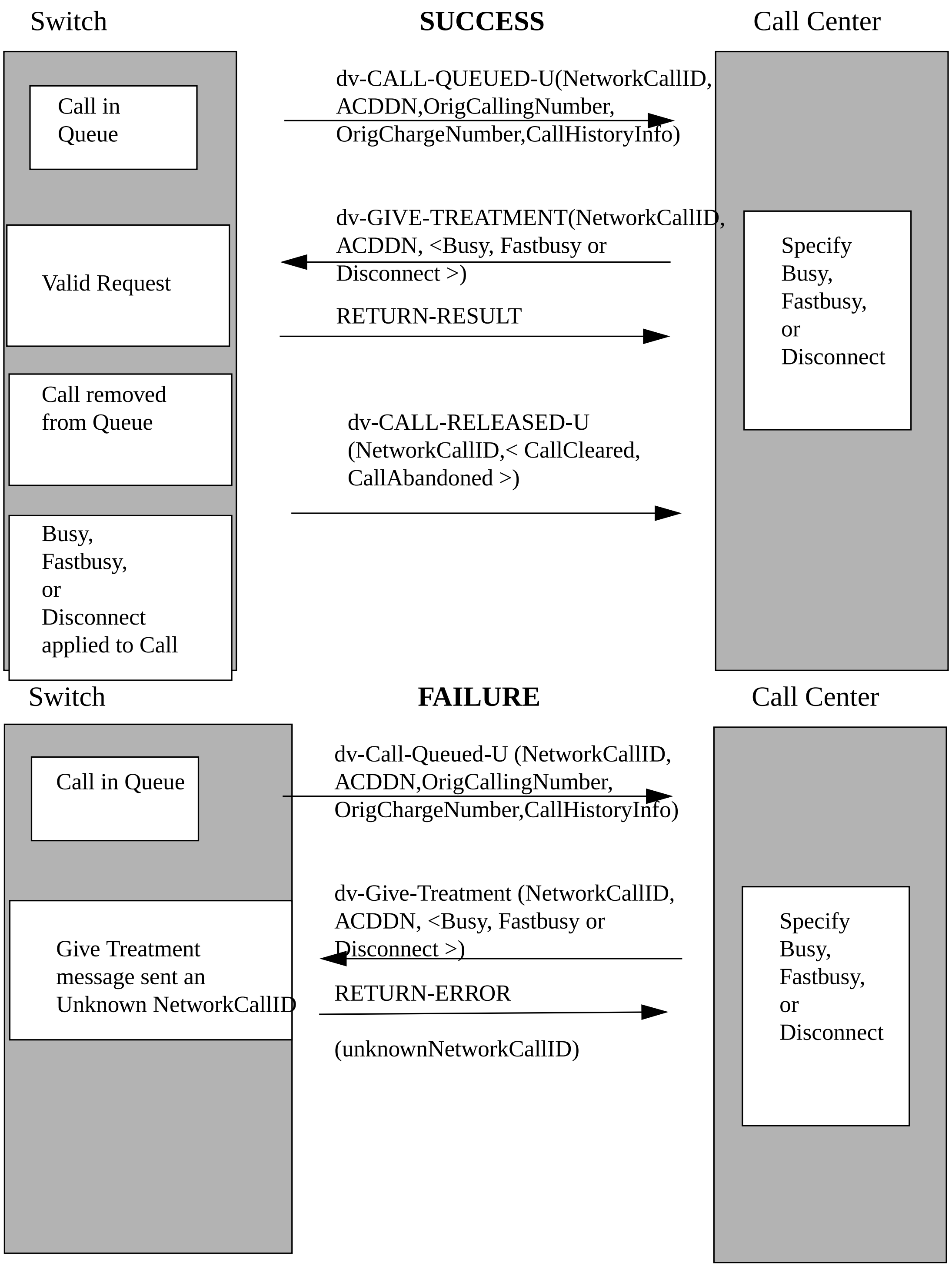

Switch
SUCCESS
Call Center
dv-Call-Queued-U(NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
Call in Queue
dv-Give-Treatment (NetworkCallID, ACDDN, MUSIC, 0)
Specify MUSIC
MUSIC Applied to NetworkCallID
RETURN-RESULT
dv-Give-Treatment (NetworkCallID, ACDDN, RAN, 0)
Specify RAN
RAN Applied to NetworkCallID
RETURN-RESULT
RAN Completed (Audio Route completes)
dv-Treatment-Complete-U (NetworkCallID,AUDIO, AUDIO_ROUTE)
dv-Give-Treatment (NetworkCallID, ACDDN, MUSIC, 0)
Specify MUSIC
Music Applied to NetworkCallID
RETURN-RESULT

6.5.16 Give Treatment (Ringback, Silence) sample session

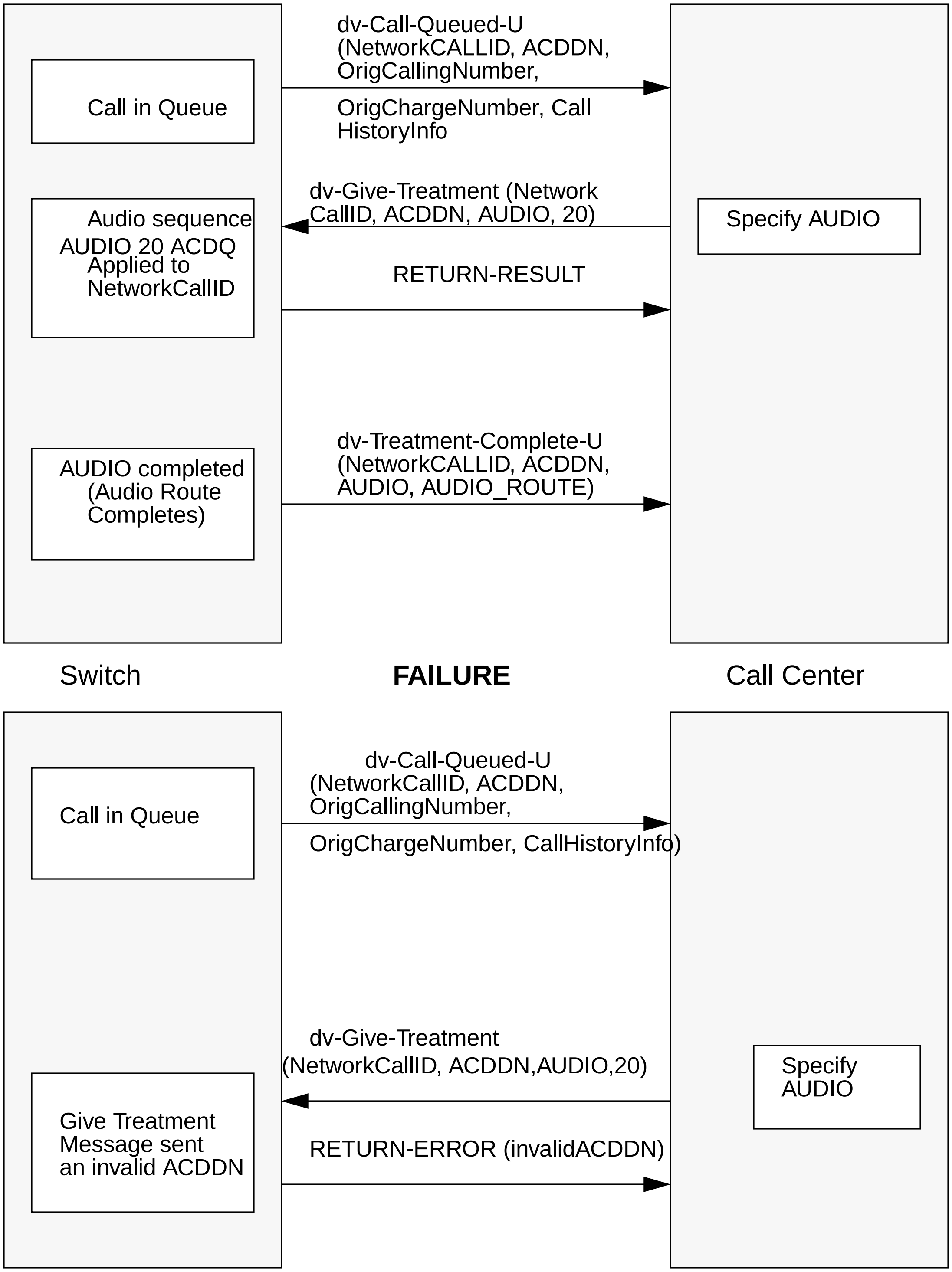

Switch
SUCCESS
Call Center
Call in Queue
dv-Call-Queued-U(NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN, <Ringback or Silence>)
Ringback or Silence Applied to NetworkCallID
Specify Ringback or Silence
RETURN-RESULT


Switch
FAILURE
Call Center
Call in Queue
dv-Call-Queued-U(NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN)
Give_Treatment Message contained a missing parameter
Specify Ringback or Silence
RETURN-ERROR(missingParameter)

6.5.17 Give Treatment (Busy, Fastbusy, Disconnect) sample session

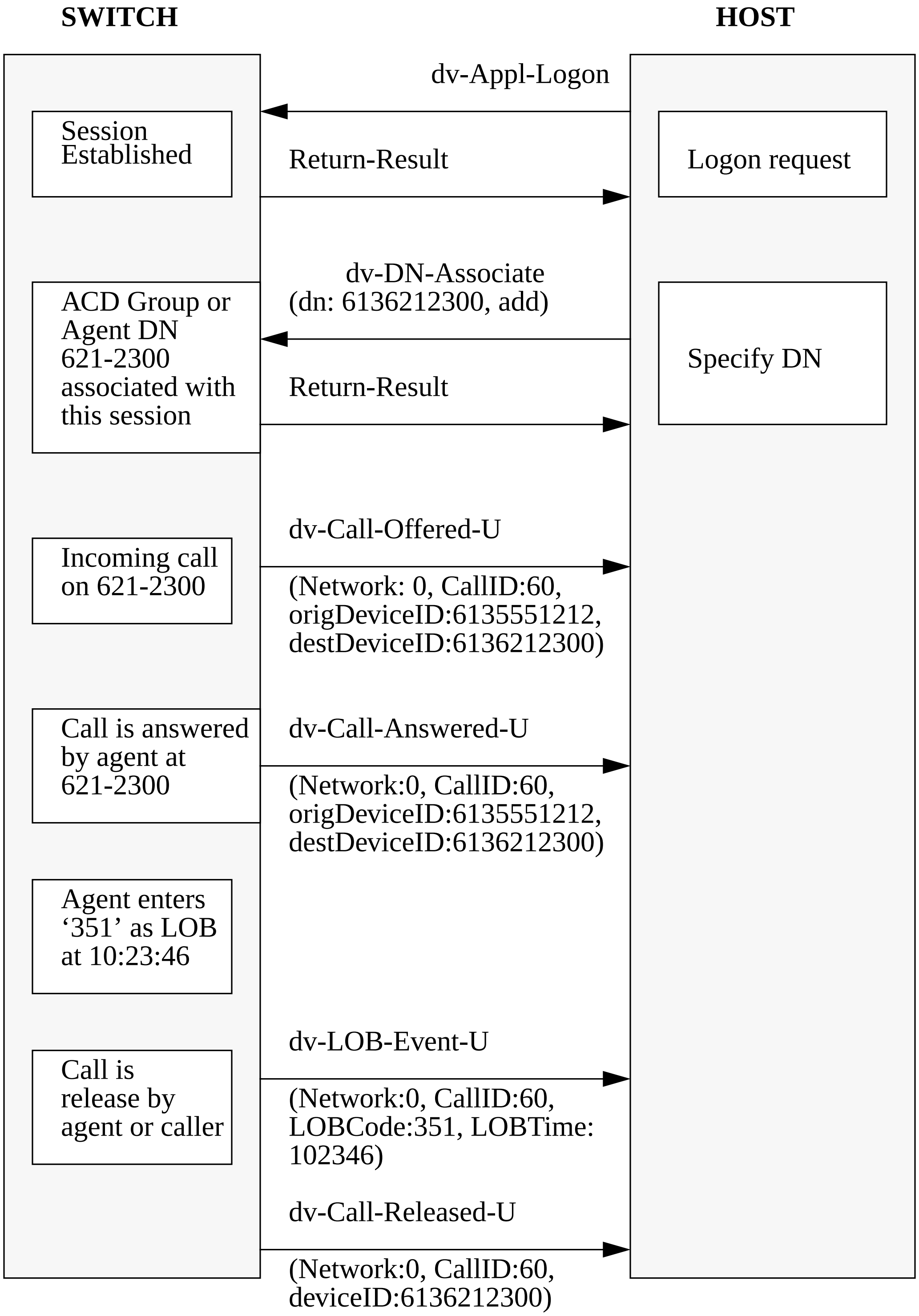

Switch
SUCCESS
Call Center
Call in Queue
dv-CALL-QUEUED-U(NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-GIVE-TREATMENT(NetworkCallID, ACDDN, <Busy, Fastbusy or Disconnect >)
Valid Request
Specify Busy, Fastbusy, or Disconnect
RETURN-RESULT
Call removed from Queue
dv-CALL-RELEASED-U (NetworkCallID,< CallCleared, CallAbandoned >)
Busy, Fastbusy, or Disconnect applied to Call
Switch
FAILURE
Call Center
Call in Queue
dv-Call-Queued-U (NetworkCallID, ACDDN,OrigCallingNumber, OrigChargeNumber,CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN, <Busy, Fastbusy or Disconnect >)
Specify Busy, Fastbusy, or Disconnect
Give Treatment message sent an Unknown NetworkCallID
RETURN-ERROR
(unknownNetworkCallID)

6.5.18 dv-Give-Treatment (ACDDN, NetworkCallID, AUDIO, <0 - 512>)


Call in Queue
dv-Call-Queued-U (NetworkCALLID, ACDDN, OrigCallingNumber,
OrigChargeNumber, Call HistoryInfo
dv-Give-Treatment (Network CallID, ACDDN, AUDIO, 20)
Specify AUDIO
Audio sequence AUDIO 20 ACDQ Applied to NetworkCallID
RETURN-RESULT
dv-Treatment-Complete-U (NetworkCALLID, ACDDN, AUDIO, AUDIO_ROUTE)
AUDIO completed (Audio Route Completes)
Switch
FAILURE
Call Center
Call in Queue
dv-Call-Queued-U (NetworkCallID, ACDDN, OrigCallingNumber,
OrigChargeNumber, CallHistoryInfo)
dv-Give-Treatment (NetworkCallID, ACDDN,AUDIO,20)
Specify AUDIO
Give Treatment Message sent an invalid ACDDN
RETURN-ERROR (invalidACDDN)

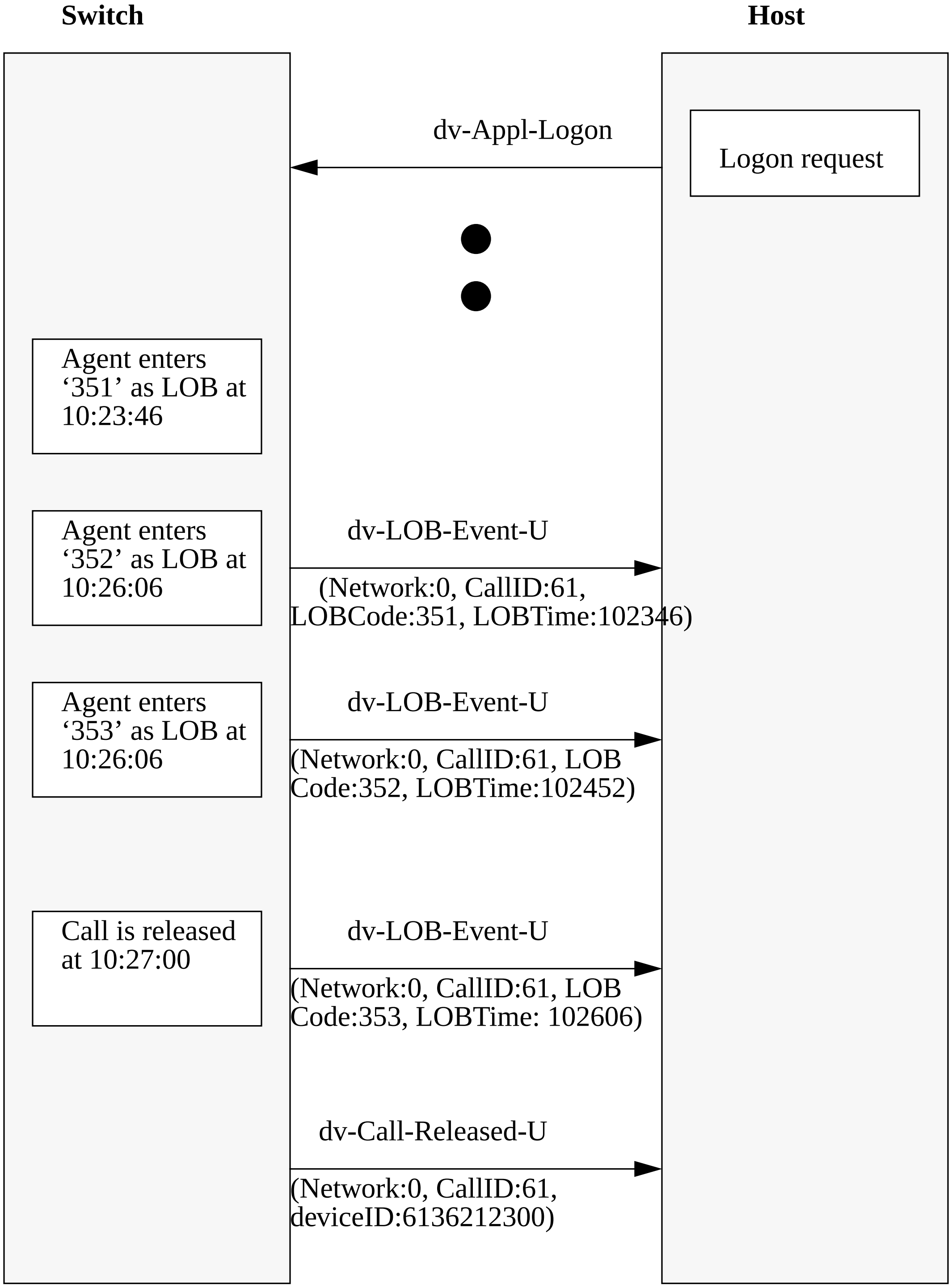
6.5.19 dv-LOB-Event-U (scenario 1)

Scenario 1 shows a session where dv-LOB-Event-U message is sent as the agent enters the code between the start and the end of the call.


SWITCH
HOST
dv-Appl-Logon
Session Established
Return-Result
Logon request
dv-DN-Associate (dn: 6136212300, add)
ACD Group or Agent DN 621-2300 associated with this session
Return-Result
Specify DN
dv-Call-Offered-U
Incoming call on 621-2300
(Network: 0, CallID:60, origDeviceID:6135551212, destDeviceID:6136212300)
Call is answered by agent at 621-2300
dv-Call-Answered-U
(Network:0, CallID:60, origDeviceID:6135551212, destDeviceID:6136212300)
Agent enters ‘351’ as LOB at 10:23:46
dv-LOB-Event-U
Call is release by agent or caller
(Network:0, CallID:60, LOBCode:351, LOBTime: 102346)
dv-Call-Released-U
(Network:0, CallID:60, deviceID:6136212300)

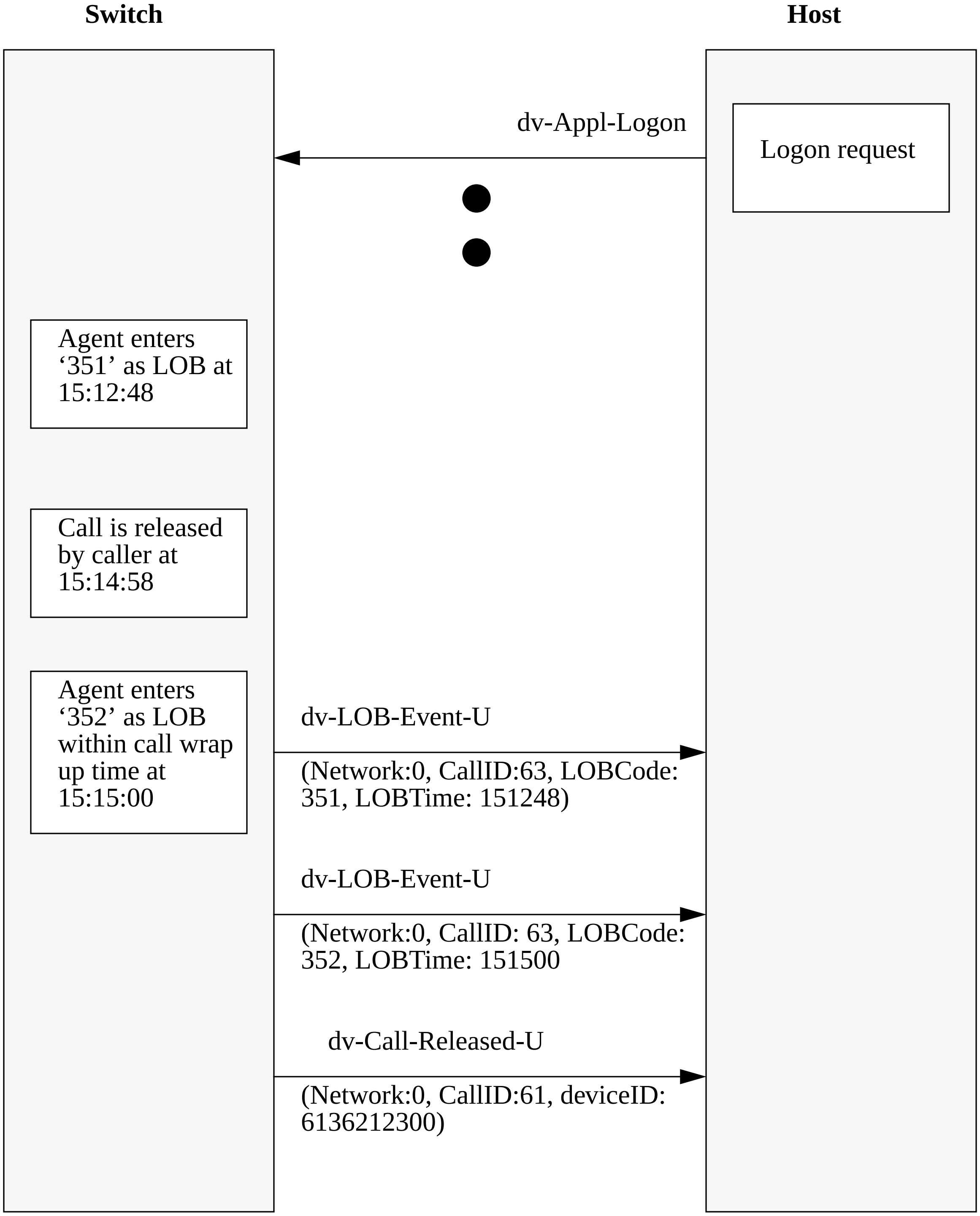
6.5.20 dv-LOB-Event-U (scenario 2)

Scenario 2, below, shows the sequence of LOB event messages when the agent enters more than one LOB code during a call.

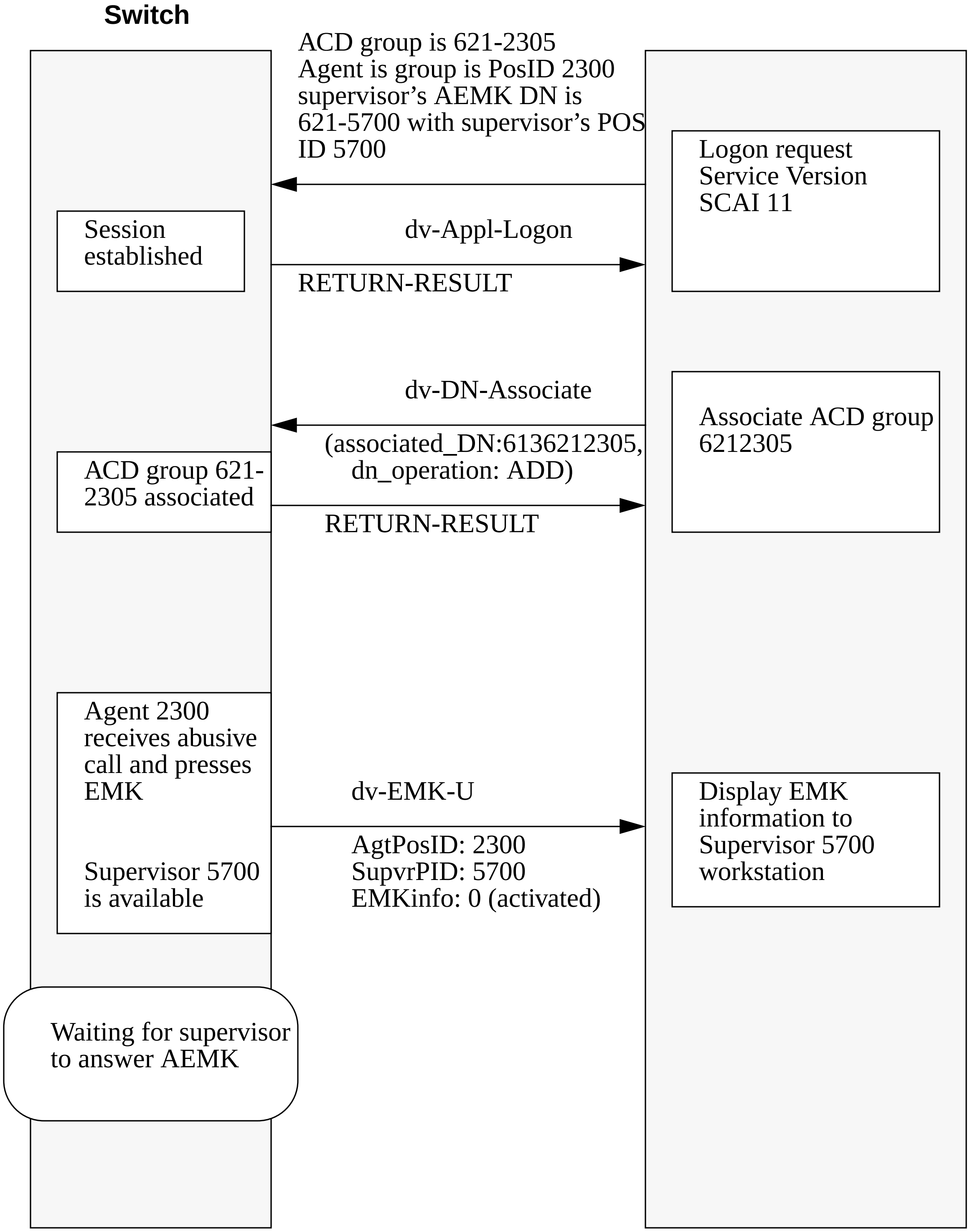

Switch
Host
dv-Appl-Logon
Logon request
Agent enters '351' as LOB at 10:23:46
Agent enters '352' as LOB at 10:26:06
dv-LOB-Event-U
(Network:0, CallID:61, LOBCode:351, LOBTime:102346)
Agent enters '353' as LOB at 10:26:06
dv-LOB-Event-U
(Network:0, CallID:61, LOB Code:352, LOBTime:102452)
Call is released at 10:27:00
dv-LOB-Event-U
(Network:0, CallID:61, LOB Code:353, LOBTime: 102606)
dv-Call-Released-U
(Network:0, CallID:61, deviceID:6136212300)

6.5.21 dv-LOB-Event-U (scenario 4)

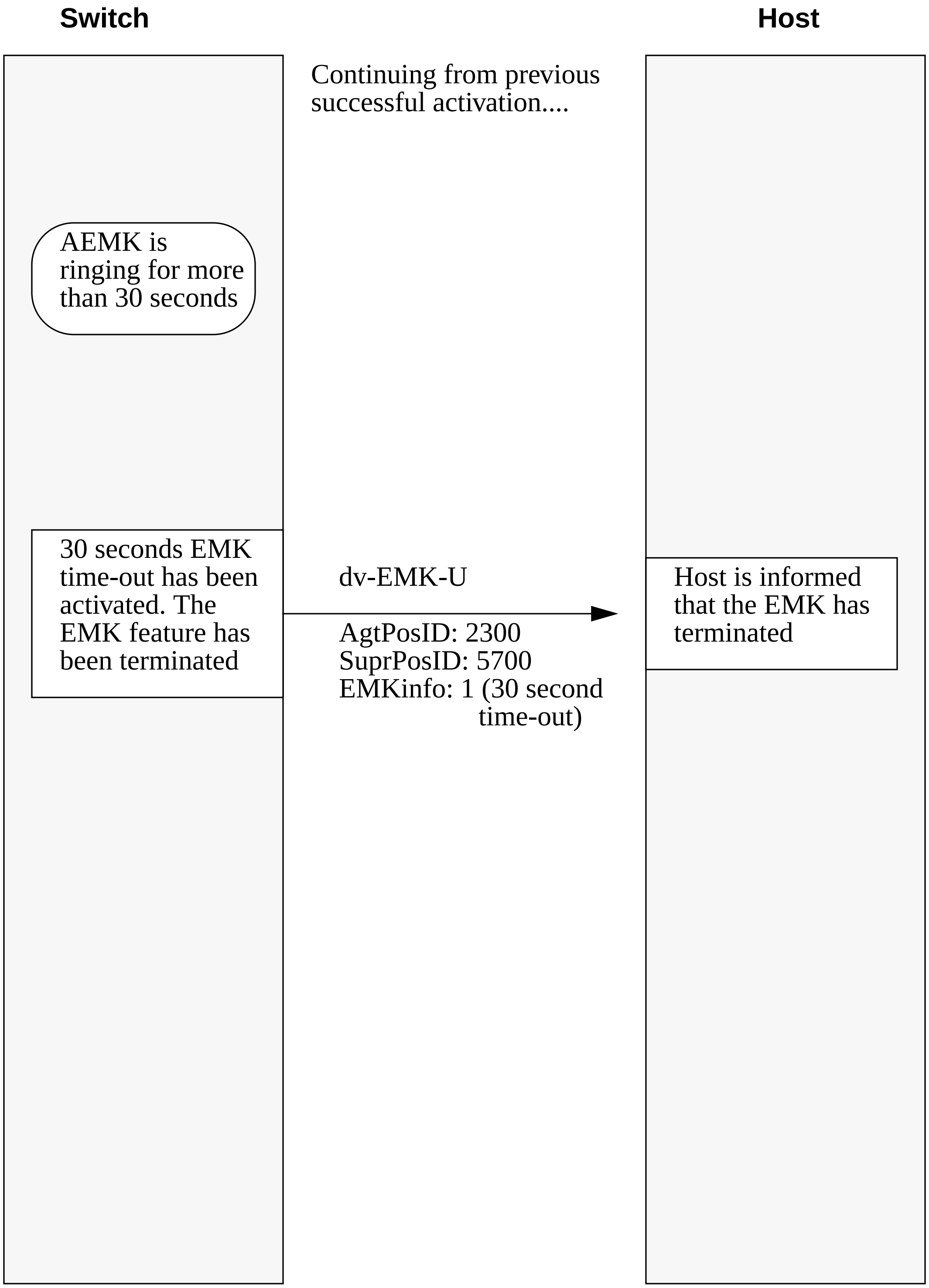
Scenario 4, below, shows the sequence of messages when the agent enters an LOB code after the caller releases the call. This LOB code must be entered during the release guard time of 2.5 seconds.


Switch
Host
dv-Appl-Logon
Logon request
Agent enters ‘351’ as LOB at 15:12:48
Call is released by caller at 15:14:58
Agent enters ‘352’ as LOB within call wrap up time at 15:15:00
dv-LOB-Event-U
(Network:0, CallID:63, LOBCode: 351, LOBTime: 151248)
dv-LOB-Event-U
(Network:0, CallID: 63, LOBCode: 352, LOBTime: 151500
dv-Call-Released-U
(Network:0, CallID:61, deviceID: 6136212300)

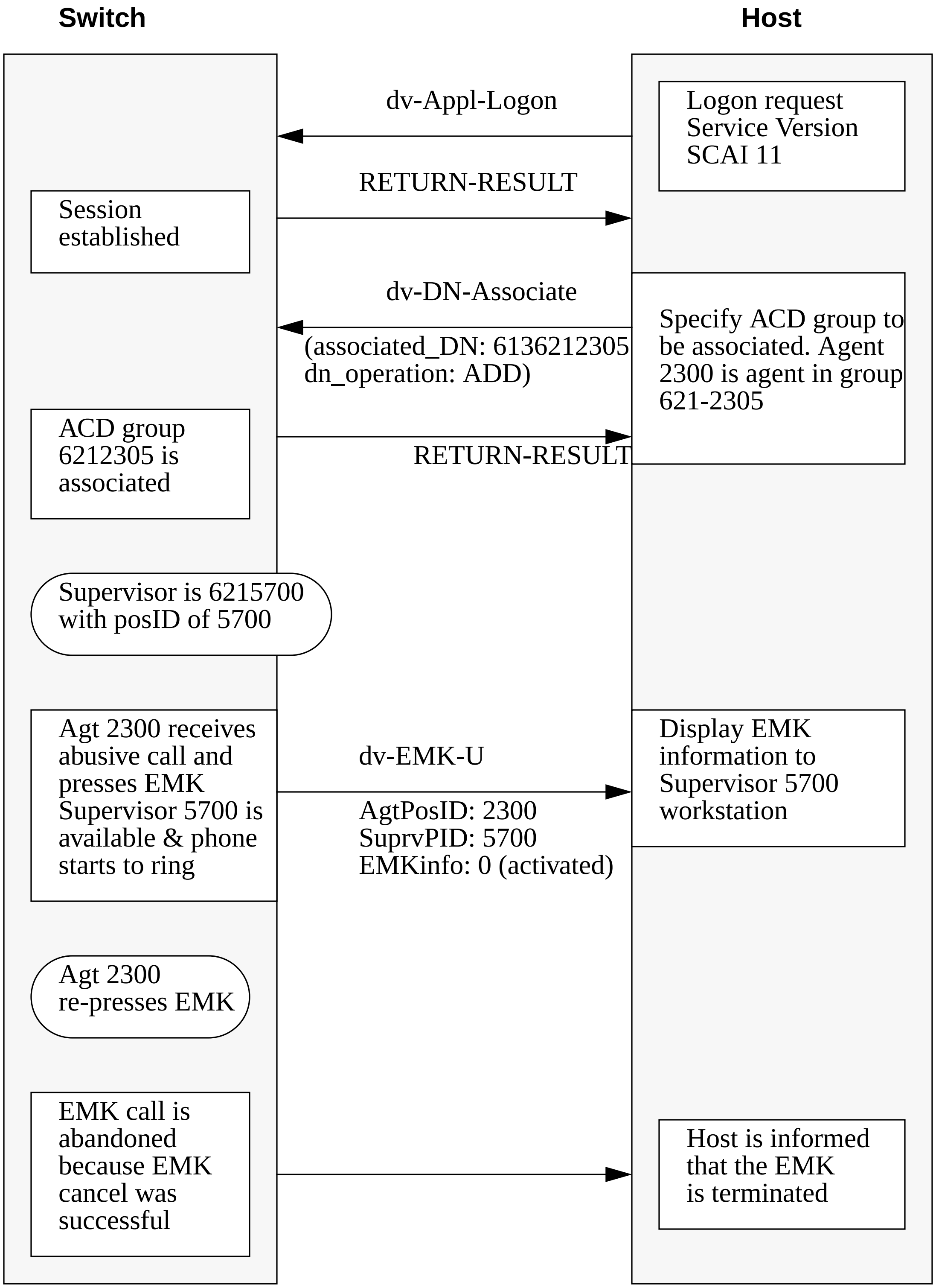
6.5.22 dv-EMK-U successful activation



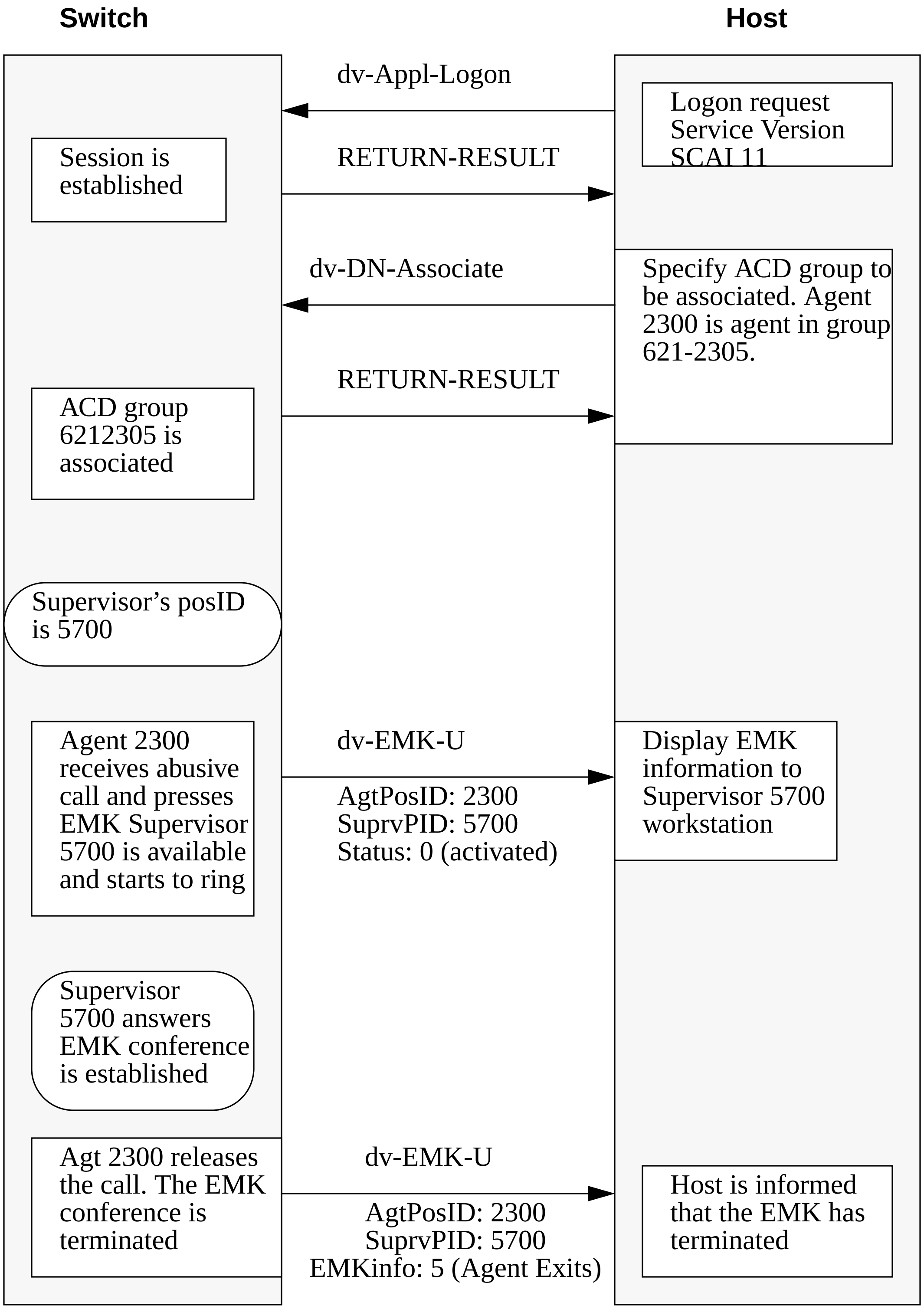
6.5.23 dv-EMK-U activation 30 second time-out


Switch
Host
Continuing from previous successful activation....
AEMK is ringing for more than 30 seconds
30 seconds EMK time-out has been activated. The EMK feature has been terminated
dv-EMK-U
AgtPosID: 2300
SuprPosID: 5700
EMKinfo: 1 (30 second time-out)
Host is informed that the EMK has terminated

6.5.24 dv-EMK-U agent cancels

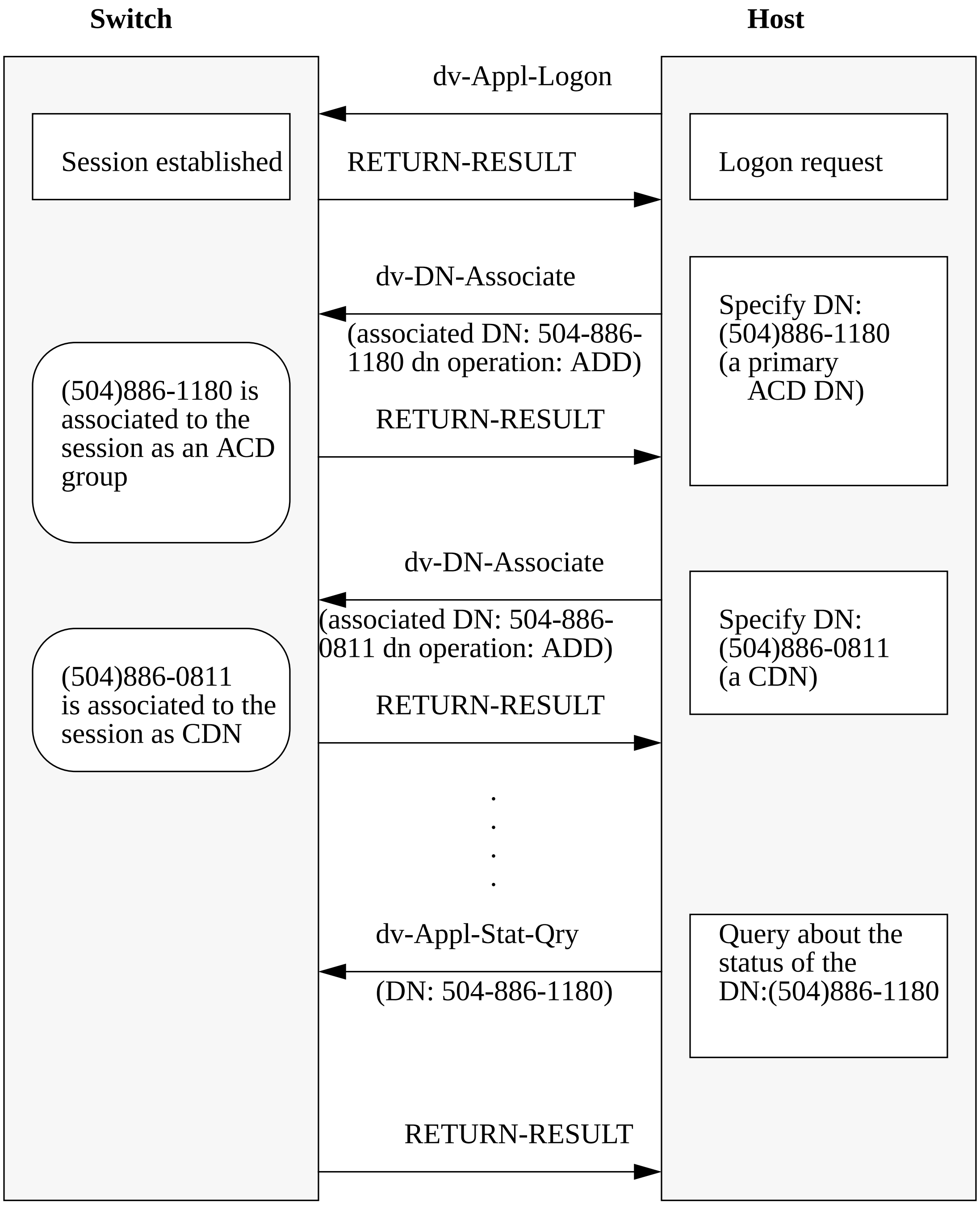


6.5.25 dv-EMK-U agent releases

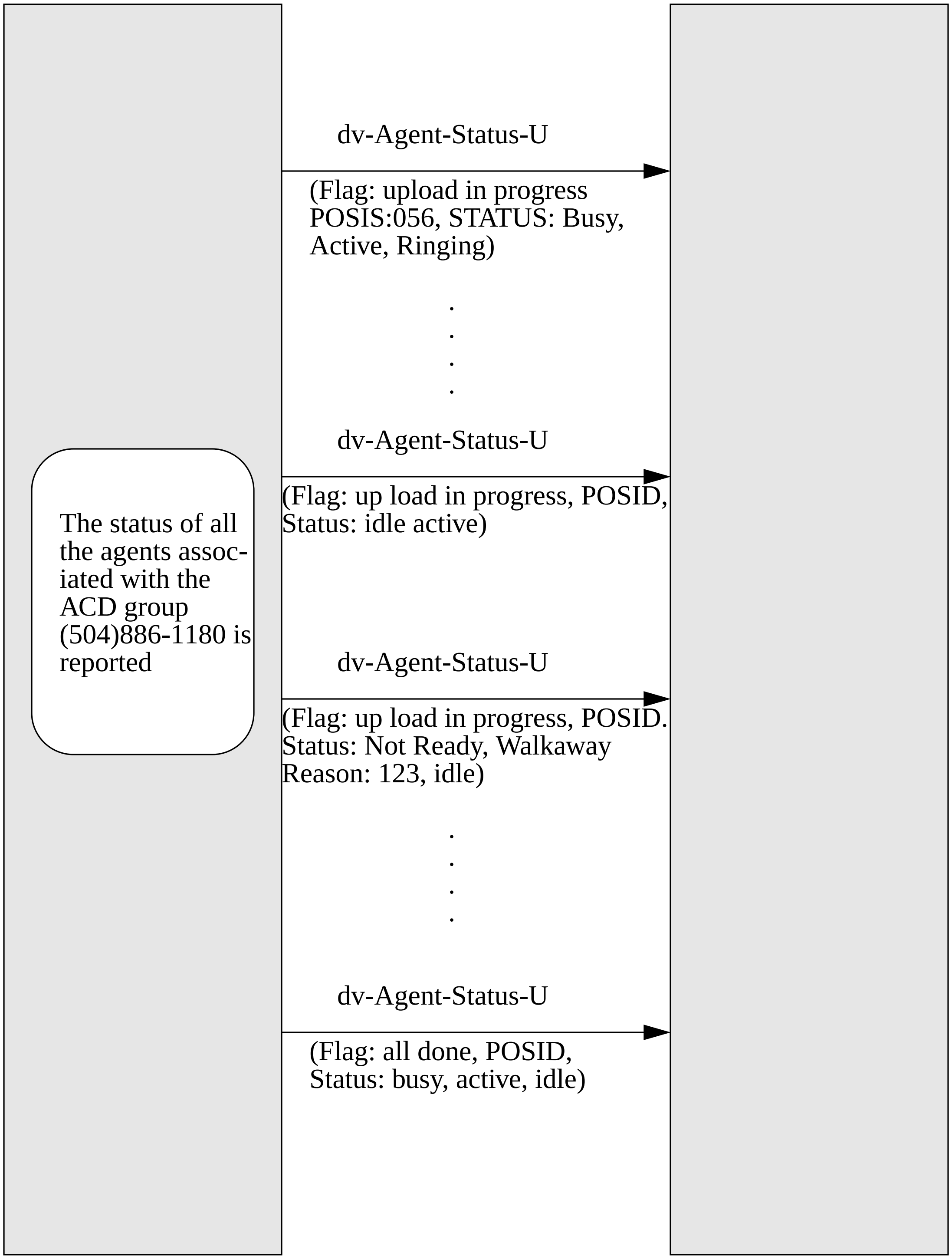

Switch
Host
dv-Appl-Logon
Logon request
Service Version
SCAI 11
Session is established
RETURN-RESULT
dv-DN-Associate
Specify ACD group to be associated. Agent 2300 is agent in group 621-2305.
RETURN-RESULT
ACD group 6212305 is associated
Supervisor's posID is 5700
Agent 2300 receives abusive call and presses EMK Supervisor 5700 is available and starts to ring
dv-EMK-U
AgtPosID: 2300
SuprvPID: 5700
Status: 0 (activated)
Display EMK information to Supervisor 5700 workstation
Supervisor 5700 answers EMK conference is established
Agt 2300 releases the call. The EMK conference is terminated
dv-EMK-U
AgtPosID: 2300
SuprvPID: 5700
EMKinfo: 5 (Agent Exits)
Host is informed that the EMK has terminated

6.5.26 dv-Appl-Stat-Query with ACDDN as DN parameter

The following example shows a switch-host session when the associated DN is an ACD DN. First, the host initiates an application session with the DMS-100. Next, the host associates(504) 886 -1180 (ACD DN) and (504)886 - 0811(CDN) with the current session. Now when the host sends the query message requesting information about the DN (504)886-1180, the switch reports the status of all the agents associated to this group.

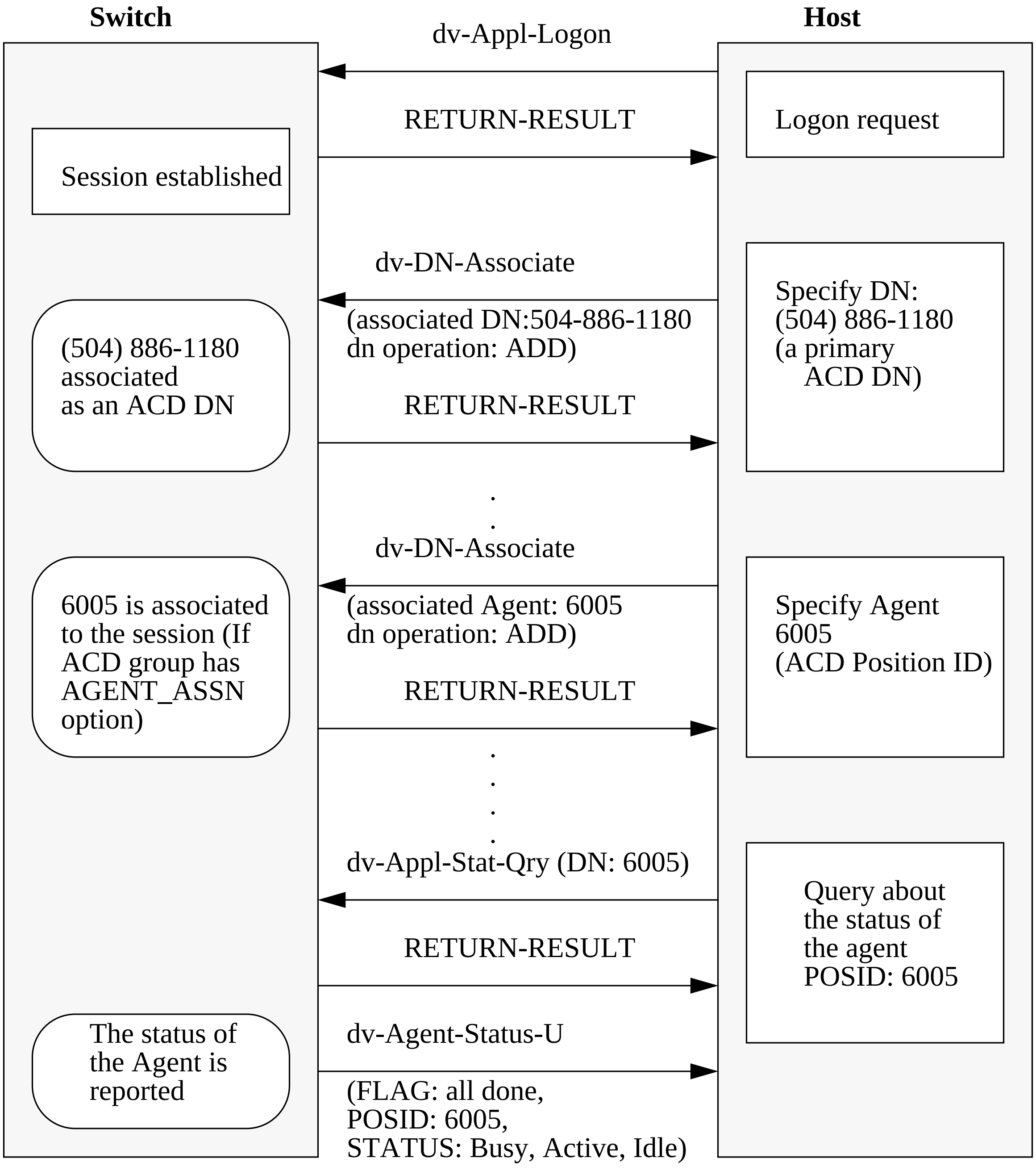

Switch
Host
dv-Appl-Logon
Session established
RETURN-RESULT
Logon request
dv-DN-Associate
(associated DN: 504-886-1180 dn operation: ADD)
Specify DN: (504)886-1180 (a primary ACD DN)
(504)886-1180 is associated to the session as an ACD group
RETURN-RESULT
dv-DN-Associate
(associated DN: 504-886-0811 dn operation: ADD)
Specify DN: (504)886-0811 (a CDN)
(504)886-0811 is associated to the session as CDN
RETURN-RESULT
dv-Appl-Stat-Qry
(DN: 504-886-1180)
Query about the status of the DN:(504)886-1180
RETURN-RESULT

continued

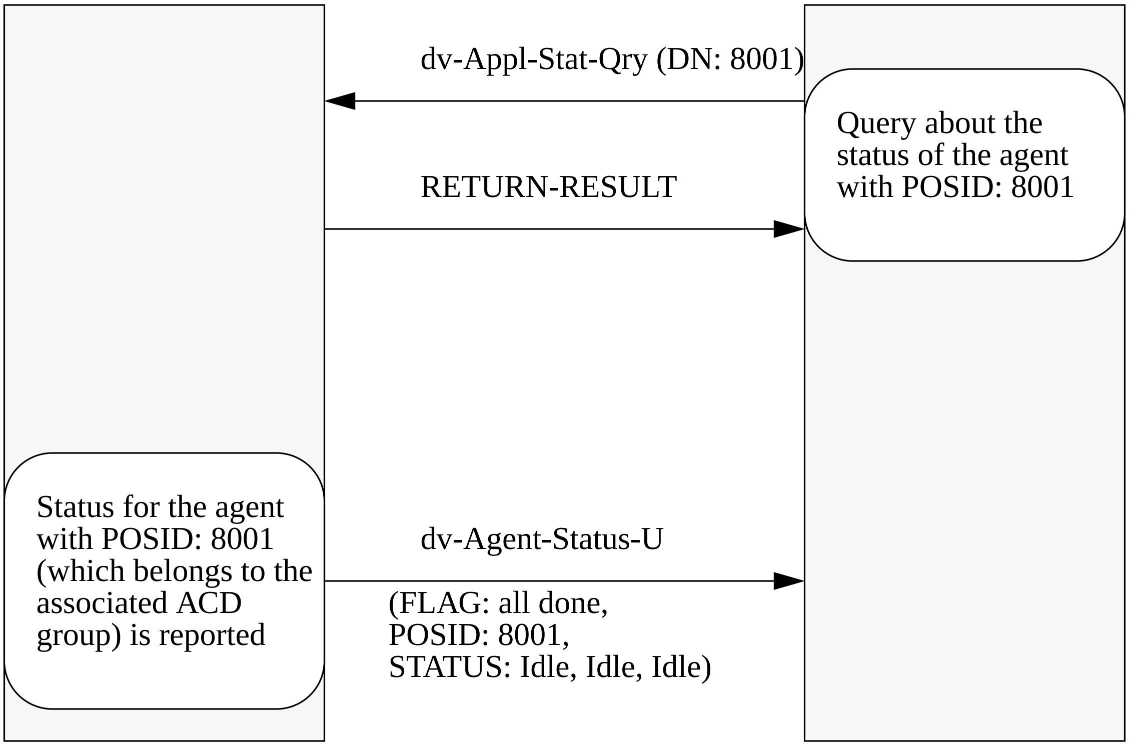

dv-Agent-Status-U
(Flag: upload in progress
POSIS:056, STATUS: Busy,
Active, Ringing)
.
.
.
.
dv-Agent-Status-U
(Flag: up load in progress, POSID,
Status: idle active)
The status of all the agents associated with the ACD group (504)886-1180 is reported
dv-Agent-Status-U
(Flag: up load in progress, POSID.
Status: Not Ready, Walkaway
Reason: 123, idle)
.
.
.
.
dv-Agent-Status-U
(Flag: all done, POSID,
Status: busy, active, idle)

6.5.27 dv-Appl-Stat-Qry with PosID as DN parameter

The following example shows a switch-host session when the POSITION_ID of an agent is specified in the Status Query message. If the agent is associated, then the Switch Report Agent Status message for that agent is sent back to the host. Here after initiating the switch-host session, host associates an ACD group (504)886-1180 and an agent (POSID 6005) with the current session. Now when the host requests the status of the agent specifying the POSID 6005 as the DN parameter, the status of the agent is reported in the dv_Agent_Status_U message. When the host specifies the POSID 8001 (which belongs to the associated ACD group) as the DN parameter, the status of the agent with POSID 8001 is reported.

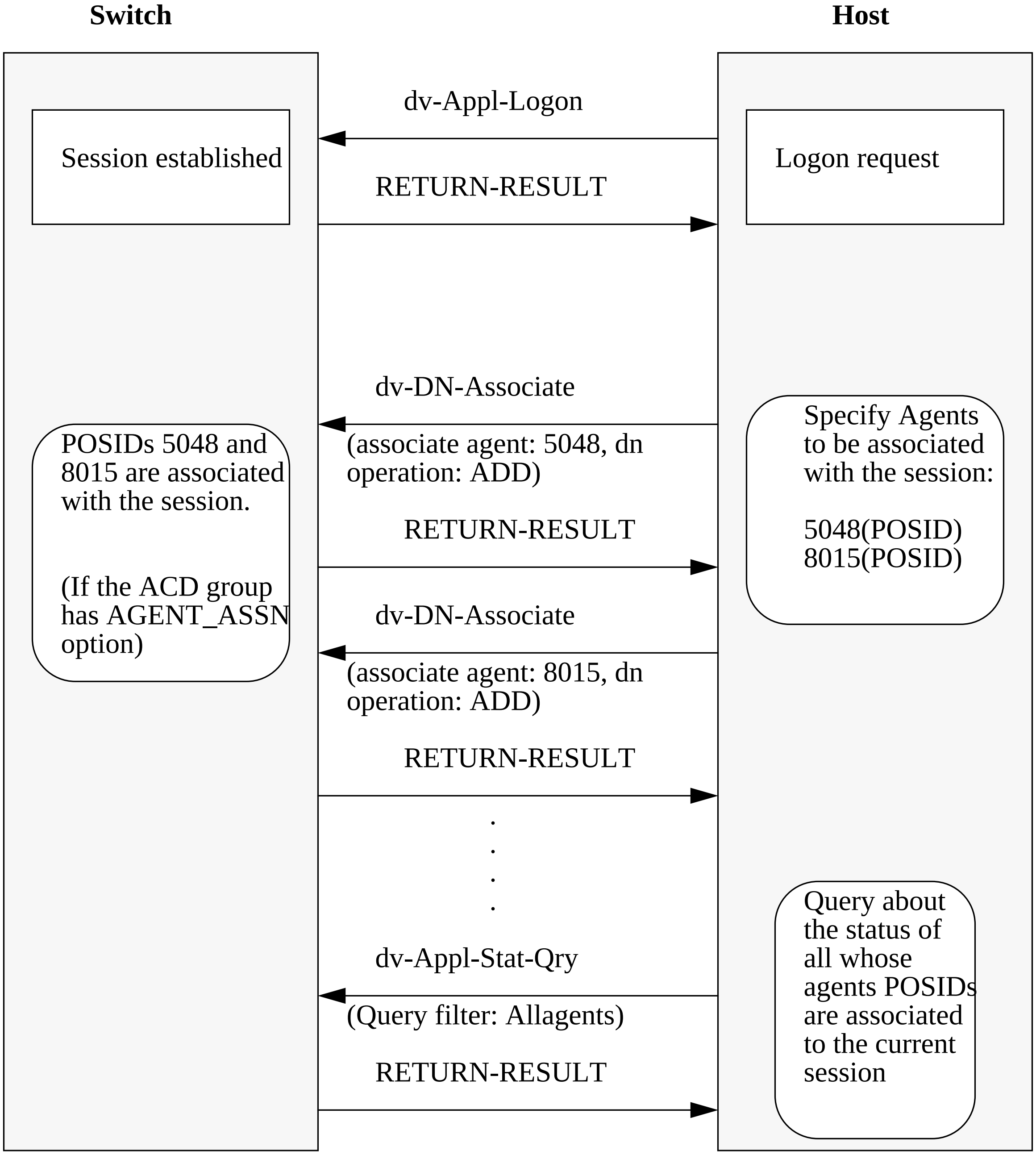


continued

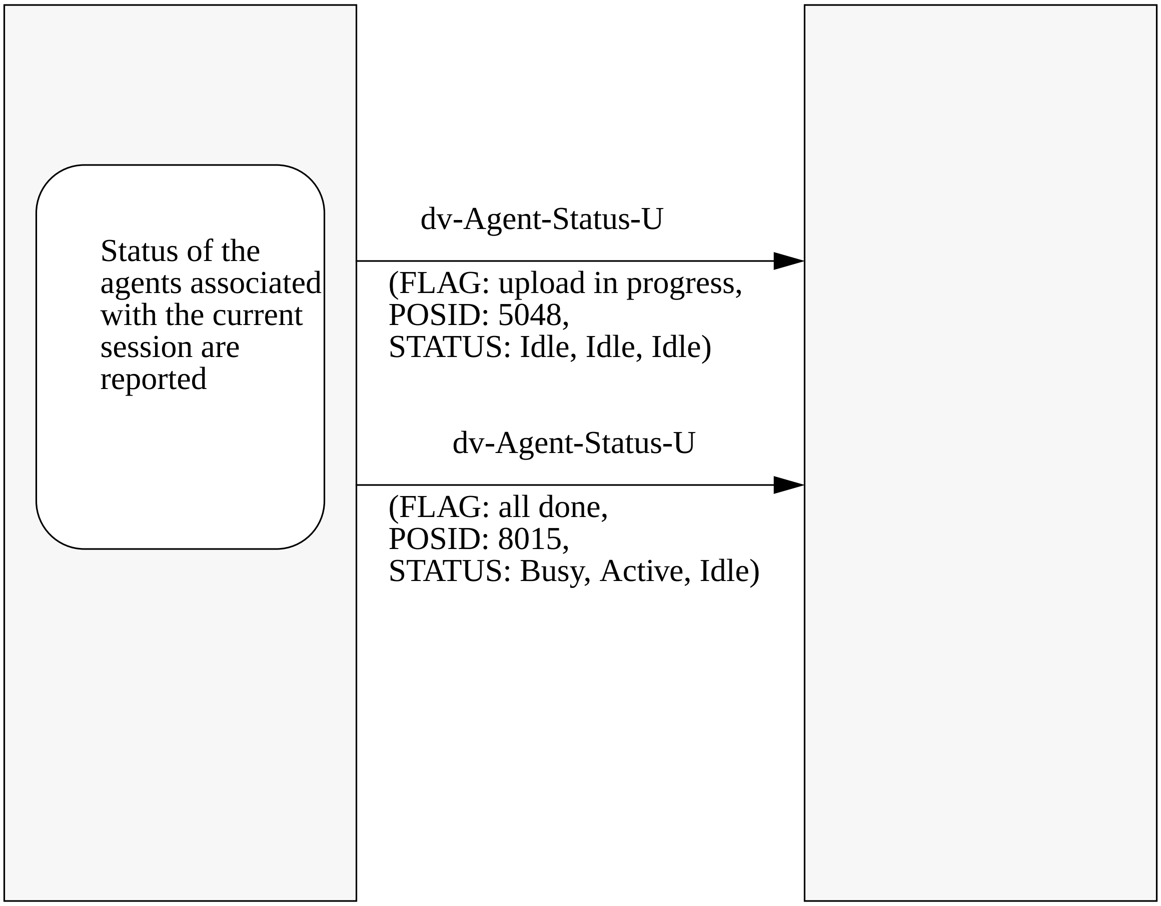

dv-Appl-Stat-Qry (DN: 8001)
Query about the status of the agent with POSID: 8001
RETURN-RESULT
Status for the agent with POSID: 8001 (which belongs to the associated ACD group) is reported
dv-Agent-Status-U
(FLAG: all done,
POSID: 8001,
STATUS: Idle, Idle, Idle)

6.5.28 dv-Appl-Stat-Qry with AllAgents as query filter parameter

If the host specifies AllAgents as a parameter in the Query filter, the agent information for all the associated agents (associated by the POSID) will be sent by the switch. The following is a typical example of a switch host session where the host after initiating the application session with the switch associates two agents with POSIDs as 5048 and 8015 respectively with the current session. When the parameter AllAgents is sent with the query message, the status of those two agents associated to the current session are reported by the switch.

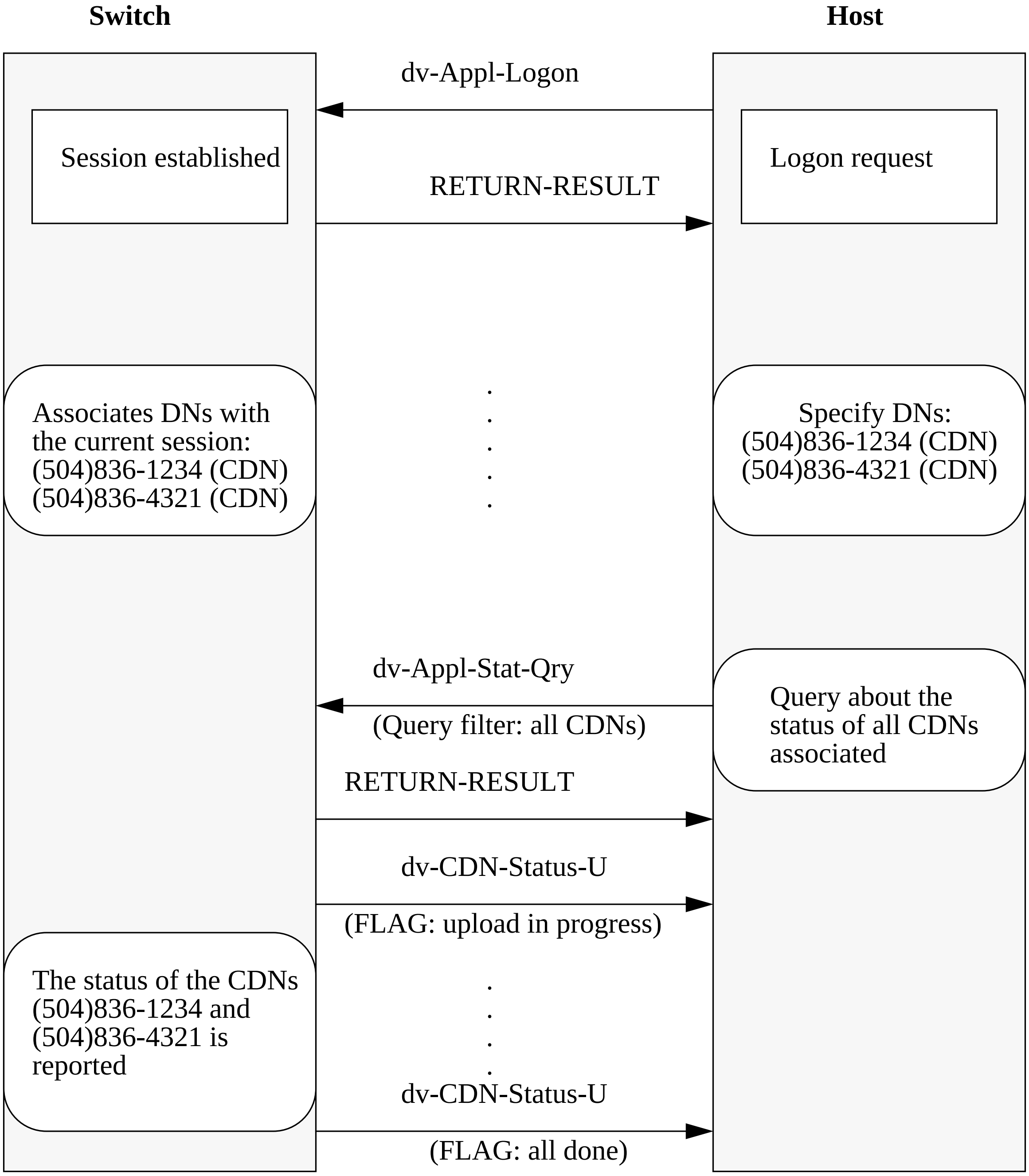

Switch
Host
dv-Appl-Logon
Session established
Logon request
RETURN-RESULT
dv-DN-Associate
Specify Agents to be associated with the session:
5048(POSID)
8015(POSID)
POSIDs 5048 and 8015 are associated with the session.
(If the ACD group has AGENT_ASSN option)
(associate agent: 5048, dn operation: ADD)
RETURN-RESULT
dv-DN-Associate
(associate agent: 8015, dn operation: ADD)
RETURN-RESULT
.
.
.
.
dv-Appl-Stat-Qry
Query about the status of all whose agents POSIDs are associated to the current session
(Query filter: Allagents)
RETURN-RESULT

continued


Status of the agents associated with the current session are reported
dv-Agent-Status-U
(FLAG: upload in progress,
POSID: 5048,
STATUS: Idle, Idle, Idle)
dv-Agent-Status-U
(FLAG: all done,
POSID: 8015,
STATUS: Busy, Active, Idle)

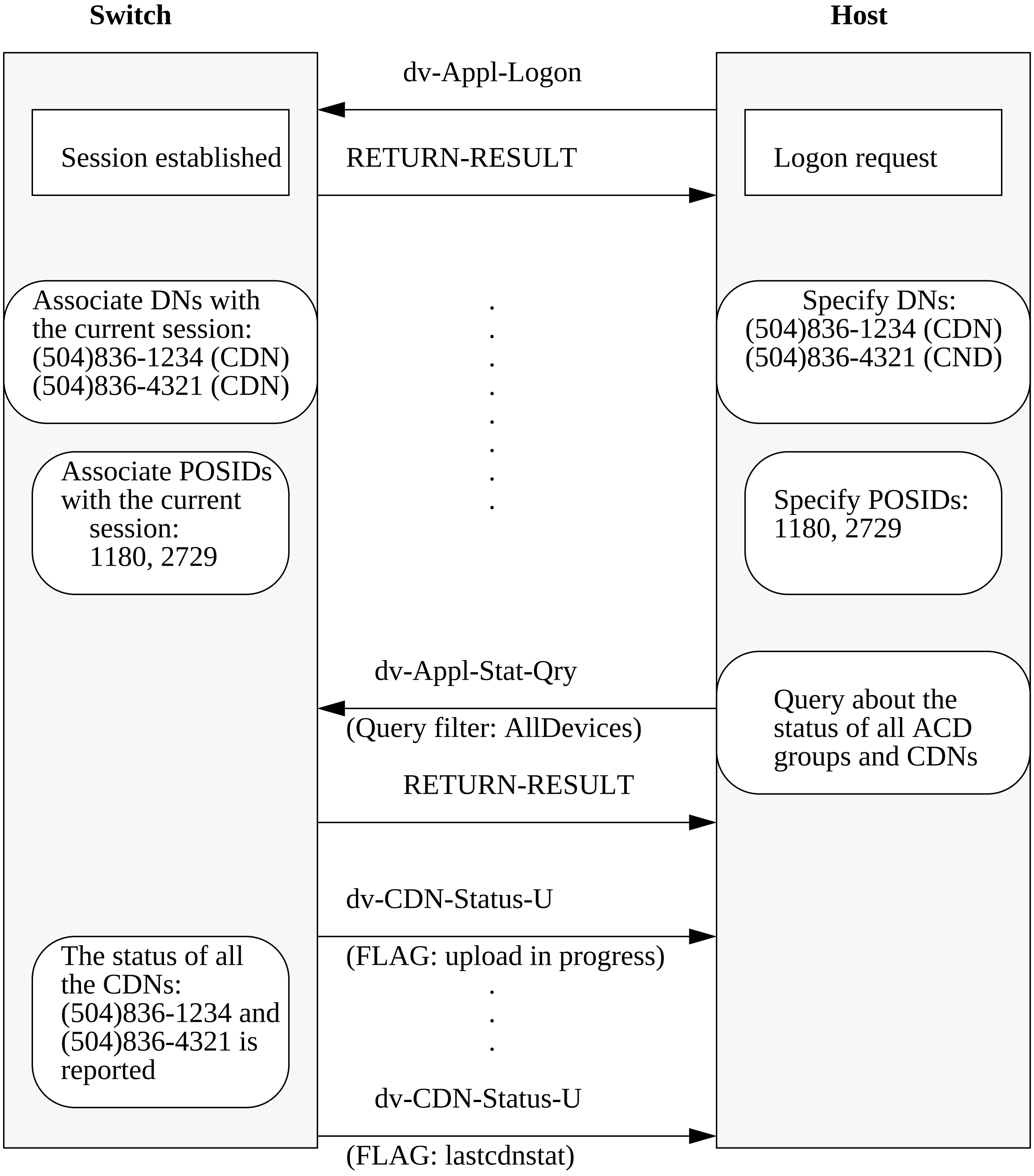
6.5.29 dv-Appl-Stat-Qry with AllCDNs as query filter parameter

If the Host specifies AllCDNs as a query filter parameter, the status of all the associated CDNs will be reported. The following is a typical example of a switch host session where the host after initiating the application session with the switch associates two CDNs (504)836-1234 and (504)836-4321 with the current session. When the parameter AllCDNs is sent with the query message, the status of these two associated CDNs is reported by the switch.

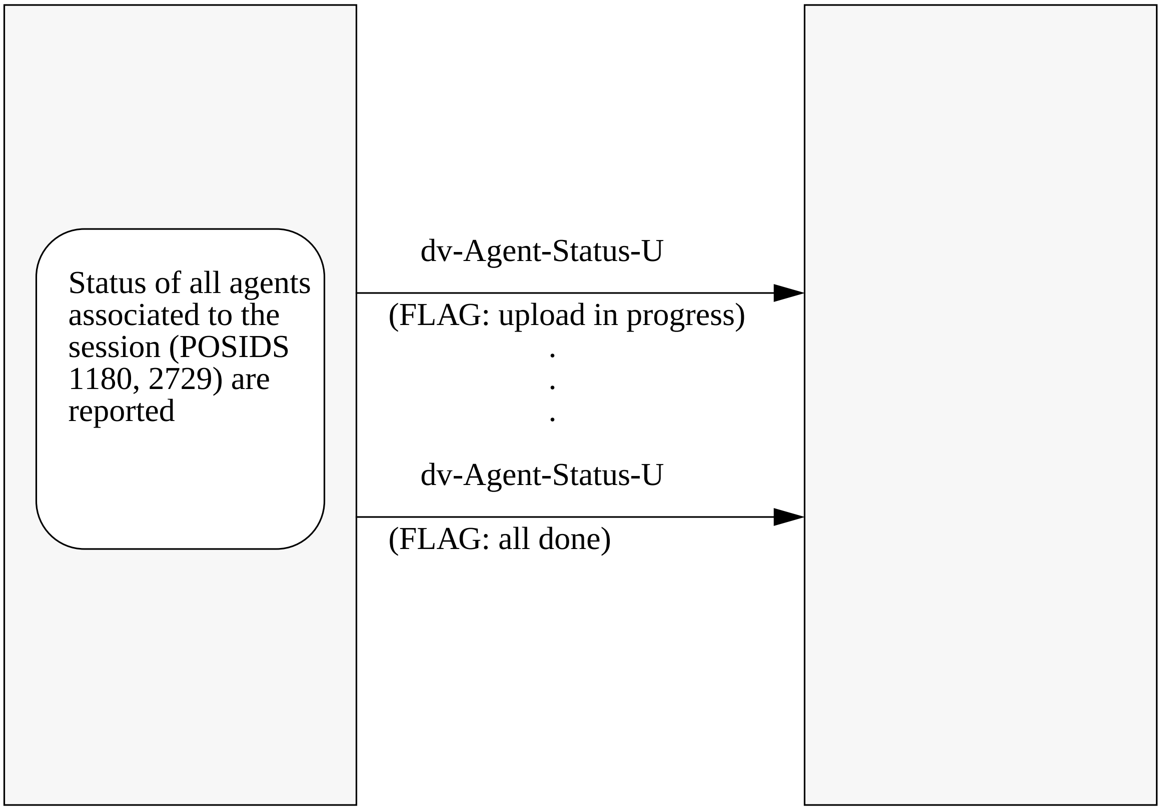

Switch
Host
dv-Appl-Logon
Session established
Logon request
RETURN-RESULT
Associates DNs with the current session: (504)836-1234 (CDN) (504)836-4321 (CDN)
Specify DNs: (504)836-1234 (CDN) (504)836-4321 (CDN)
dv-Appl-Stat-Qry
(Query filter: all CDNs)
Query about the status of all CDNs associated
RETURN-RESULT
dv-CDN-Status-U
(FLAG: upload in progress)
The status of the CDNs (504)836-1234 and (504)836-4321 is reported
dv-CDN-Status-U
(FLAG: all done)

6.5.30 dv-Appl-Stat-Qry with AllDevices as query filter parameter

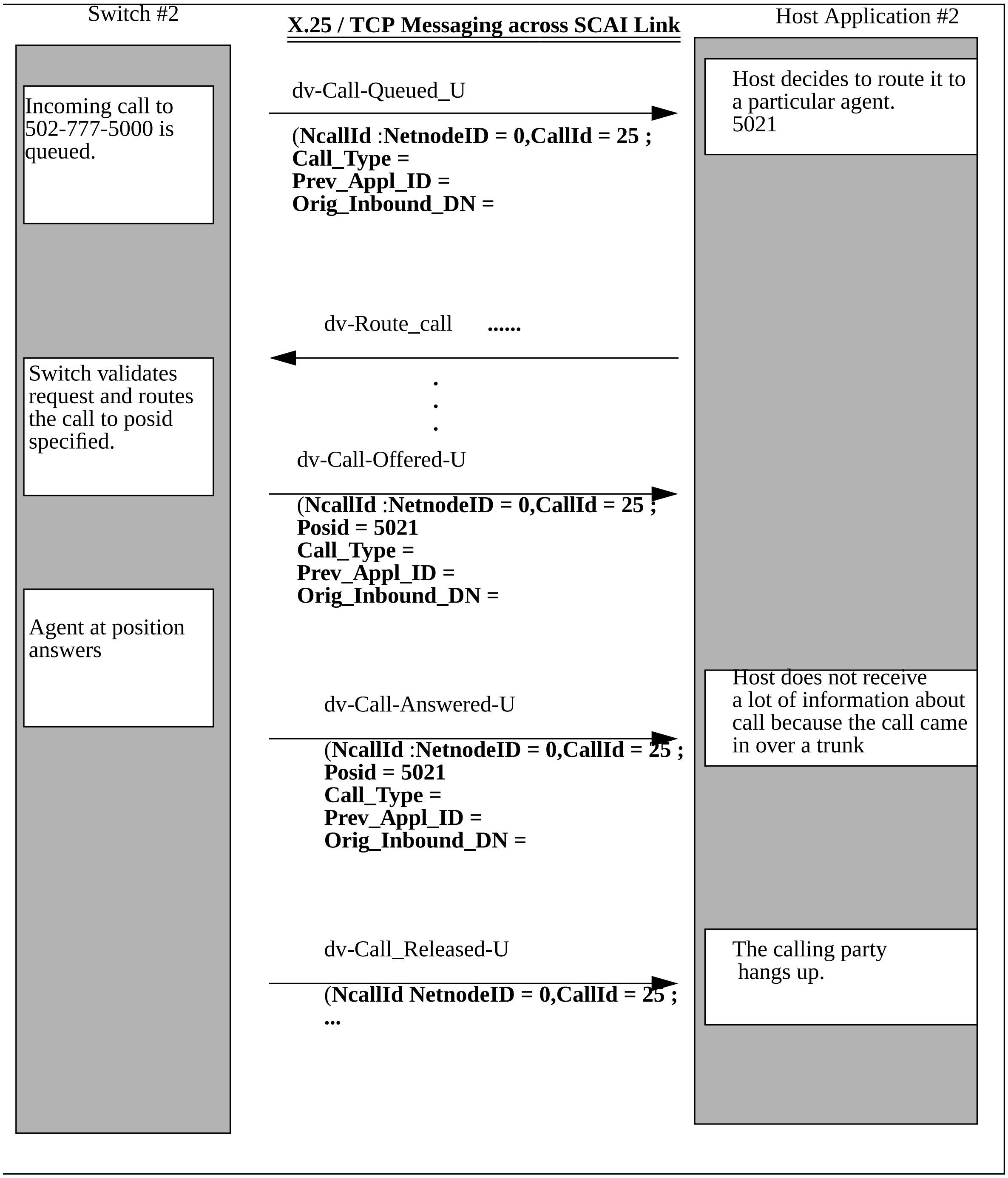
If the Host specifies AllDevices as a Query filter parameter, the status of all the associated POSIDs and CDNs will be reported. The following shows a typical example of a switch host session where the host after initiating the application session with the switch associates two CDNs (504)836-1234 and (504)836-4321. It also associates two agents by associating the POSIDs 1180 and 2729. Hence the status of the two CDNs as well as that of the two associated agents are going to be reported separately in the response messages.


Switch
Host
dv-Appl-Logon
Session established
RETURN-RESULT
Logon request
Associate DNs with the current session: (504)836-1234 (CDN) (504)836-4321 (CDN)
Specify DNs: (504)836-1234 (CDN) (504)836-4321 (CND)
Associate POSIDs with the current session: 1180, 2729
Specify POSIDs: 1180, 2729
dv-Appl-Stat-Qry
(Query filter: AllDevices)
Query about the status of all ACD groups and CDNs
RETURN-RESULT
dv-CDN-Status-U
(FLAG: upload in progress)
The status of all the CDNs: (504)836-1234 and (504)836-4321 is reported
dv-CDN-Status-U
(FLAG: lastcdnstat)

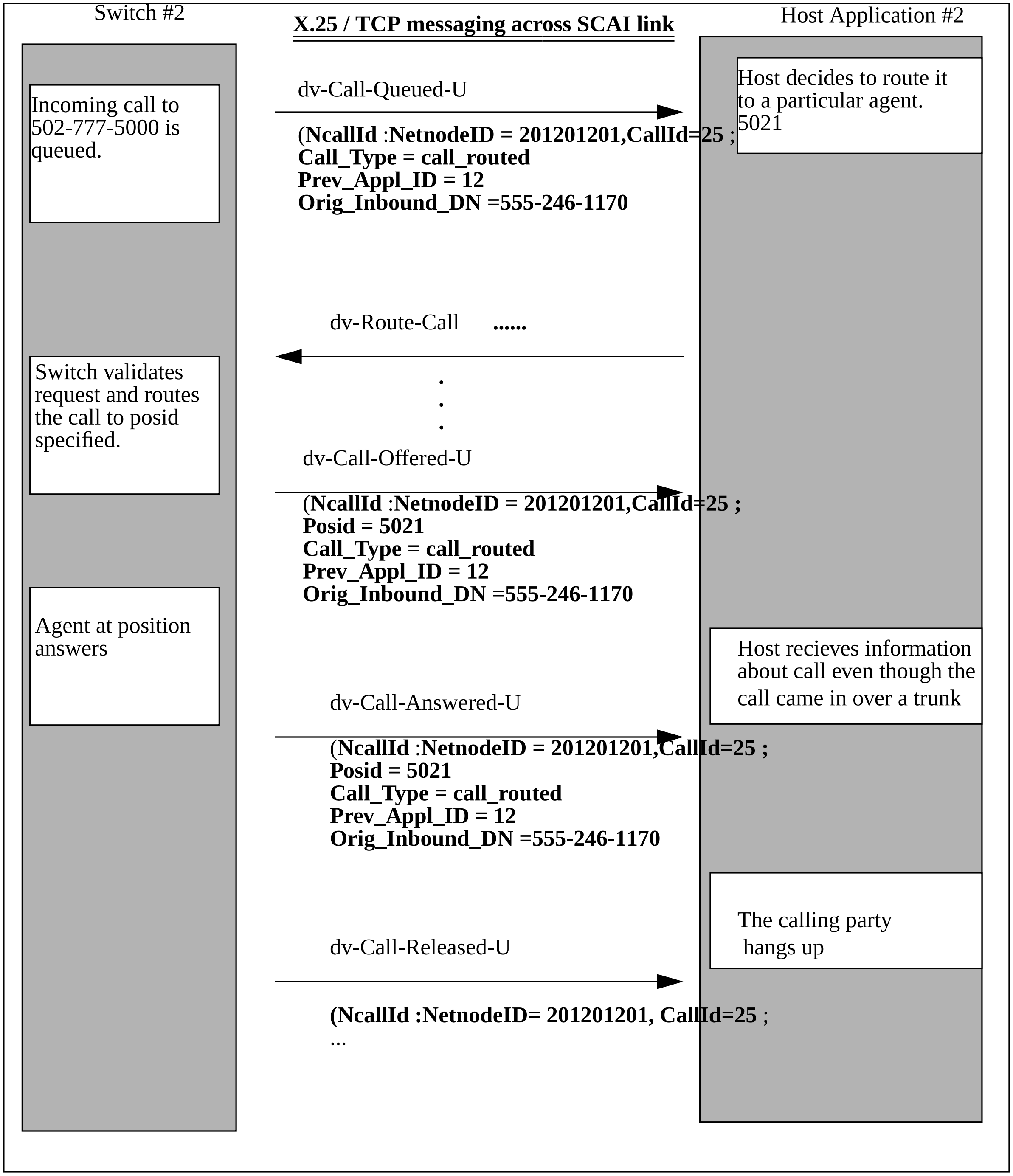
continued


Status of all agents associated to the session (POSIDS 1180, 2729) are reported
dv-Agent-Status-U
(FLAG: upload in progress)
.
.
.
dv-Agent-Status-U
(FLAG: all done)

6.5.31Extended call without NICM funtionality


Switch #2
X.25 / TCP Messaging across SCAI Link
Host Application #2
Incoming call to 502-777-5000 is queued.
dv-Call-Queued_U
(NcallId :NetnodeID = 0,CallId = 25 ;
Call_Type =
Prev_Appl_ID =
Orig_Inbound_DN =
Host decides to route it to a particular agent. 5021
dv-Route_call ......
Switch validates request and routes the call to posid specified.
. . .
dv-Call-Offered-U
(NcallId :NetnodeID = 0,CallId = 25 ;
Posid = 5021
Call_Type =
Prev_Appl_ID =
Orig_Inbound_DN =
Agent at position answers
dv-Call-Answered-U
(NcallId :NetnodeID = 0,CallId = 25 ;
Posid = 5021
Call_Type =
Prev_Appl_ID =
Orig_Inbound_DN =
Host does not receive a lot of information about call because the call came in over a trunk
dv-Call_Released-U
(NcallId NetnodeID = 0,CallId = 25 ;
...
The calling party hangs up.

6.5.32 Extended call with NICM funtionality



Incoming call to 502-777-5000 is queued.

dv-Call-Queued-U

(**NcallId** :**NetnodeID = 201201201,CallId=25** ;
Call_Type = call_routed
Prev_Appl_ID = 12
Orig_Inbound_DN =555-246-1170

Host decides to route it to a particular agent. 5021

dv-Route-Call **......**

. . .

Switch validates request and routes the call to posid specified.

dv-Call-Offered-U

(**NcallId** :**NetnodeID = 201201201,CallId=25 ;**
Posid = 5021
Call_Type = call_routed
Prev_Appl_ID = 12
Orig_Inbound_DN =555-246-1170

Agent at position answers

Host recieves information about call even though the call came in over a trunk

dv-Call-Answered-U

(**NcallId** :**NetnodeID = 201201201,CallId=25 ;**
Posid = 5021
Call_Type = call_routed
Prev_Appl_ID = 12
Orig_Inbound_DN =555-246-1170

The calling party hangs up

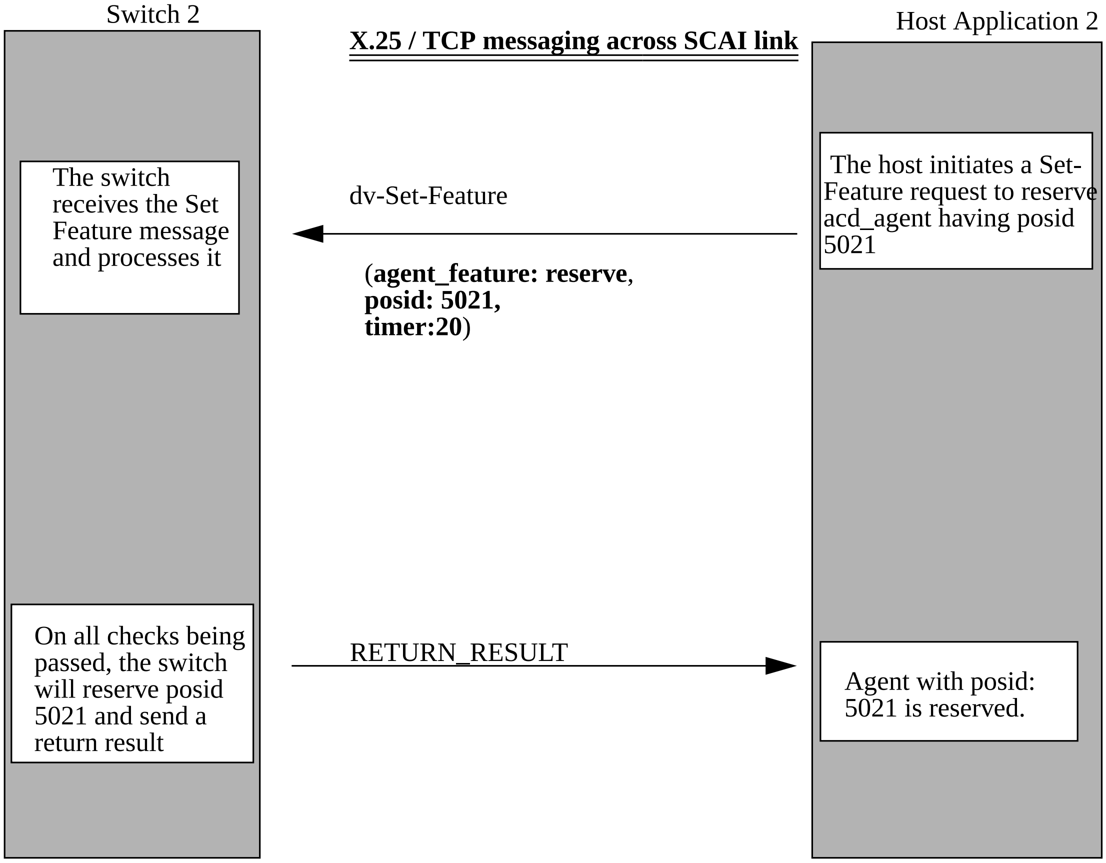
dv-Call-Released-U

(NcallId :NetnodeID= 201201201, CallId=25 ;
...

6.5.33 Set_Feature (reserved) with Return Result

In this case, the host1 wishes to enhance the call to the host2 which has a session (associated) to the switch2. The host1 should be able to inform the host2 (via LAN / WAN connectivity) that a call has been enhanced to host2. Hence host2 will prepare to receive the call by reserving an agent.Therefore host2 sends a dv-Set-Feature message (with reserve parameter) and timer value requesting the switch to reserve the specified agent.

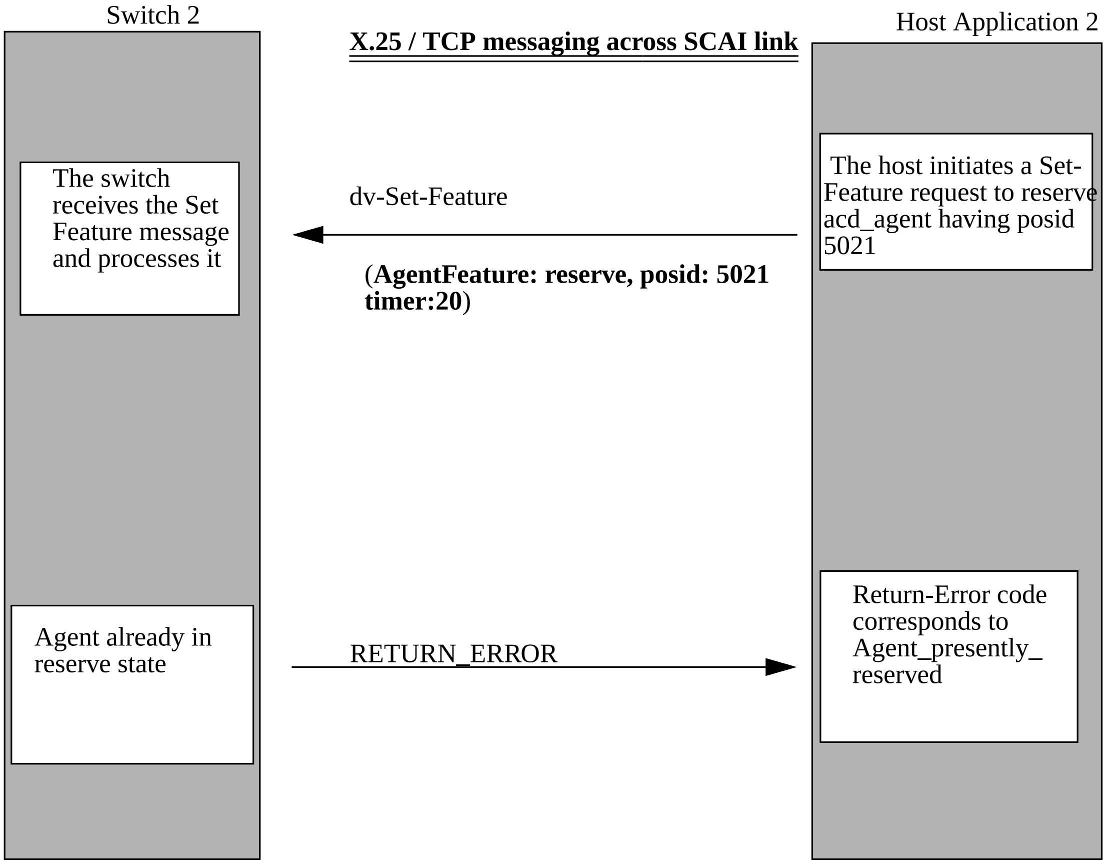
The switch2 will reserve the specified agent, if all subscription and other checks are passed and the agent can be reserved and upon doing so it will return to the host2 a return-result message.


Switch 2
X.25 / TCP messaging across SCAI link
Host Application 2
The switch receives the Set Feature message and processes it
dv-Set-Feature
The host initiates a Set-Feature request to reserve acd_agent having posid 5021
(**agent_feature: reserve**, **posid: 5021, timer:20**)
On all checks being passed, the switch will reserve posid 5021 and send a return result
RETURN_RESULT
Agent with posid: 5021 is reserved.

6.5.34 Set_Feature (reserved) with Return Error (agent_already reserved)

In this case, the host1 wishes to enhance the call to the host2 which has a session (associated) to the switch2. The host1 should be able to inform the host2 (via LAN / WAN connectivity) that a call has been enhanced to host2. Hence host2 will prepare to receive the call by reserving an agent. Therefore, host2 sends a dv_SET_FEATURE message (with reserve parameter) and timer value requesting the switch to reserve the specified agent.

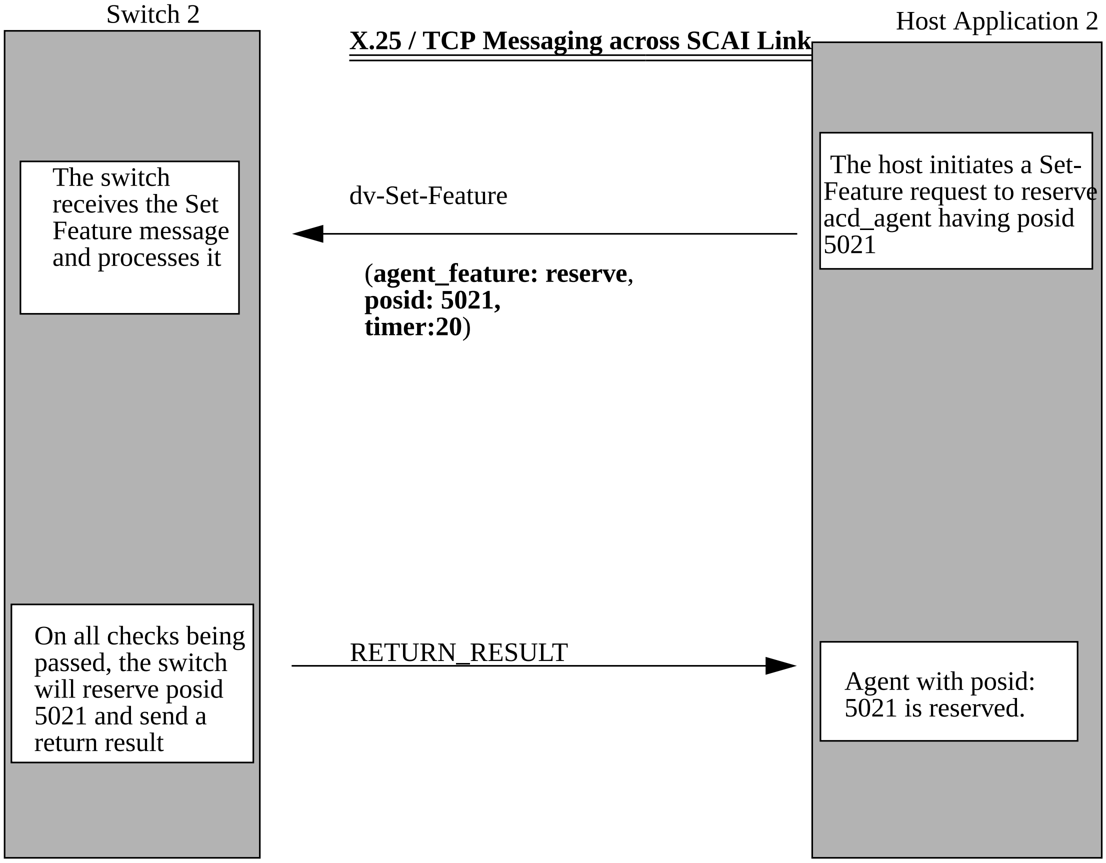
The switch2 finds the agent already reserved and hence returns error message.


Switch 2
X.25 / TCP messaging across SCAI link
Host Application 2
The switch receives the Set Feature message and processes it
dv-Set-Feature
The host initiates a Set-Feature request to reserve acd_agent having posid 5021
(AgentFeature: reserve, posid: 5021 timer:20)
Agent already in reserve state
RETURN_ERROR
Return-Error code corresponds to Agent_presently_ reserved

6.5.35Set Feature (unreserved) with Return Result

In this case, the host1 wishes to enhance the call to the host2 which has a session (associated) to the switch2. The host1 should be able to inform the host2 (via LAN / WAN connectivity) that a call has been enhanced to host2. Hence host2 will prepare to receive the call by reserving an agent. Therefore, host2 sends a dv-Set-Feature message (with reserve parameter) and timer value requesting the switch to reserve the specified agent.

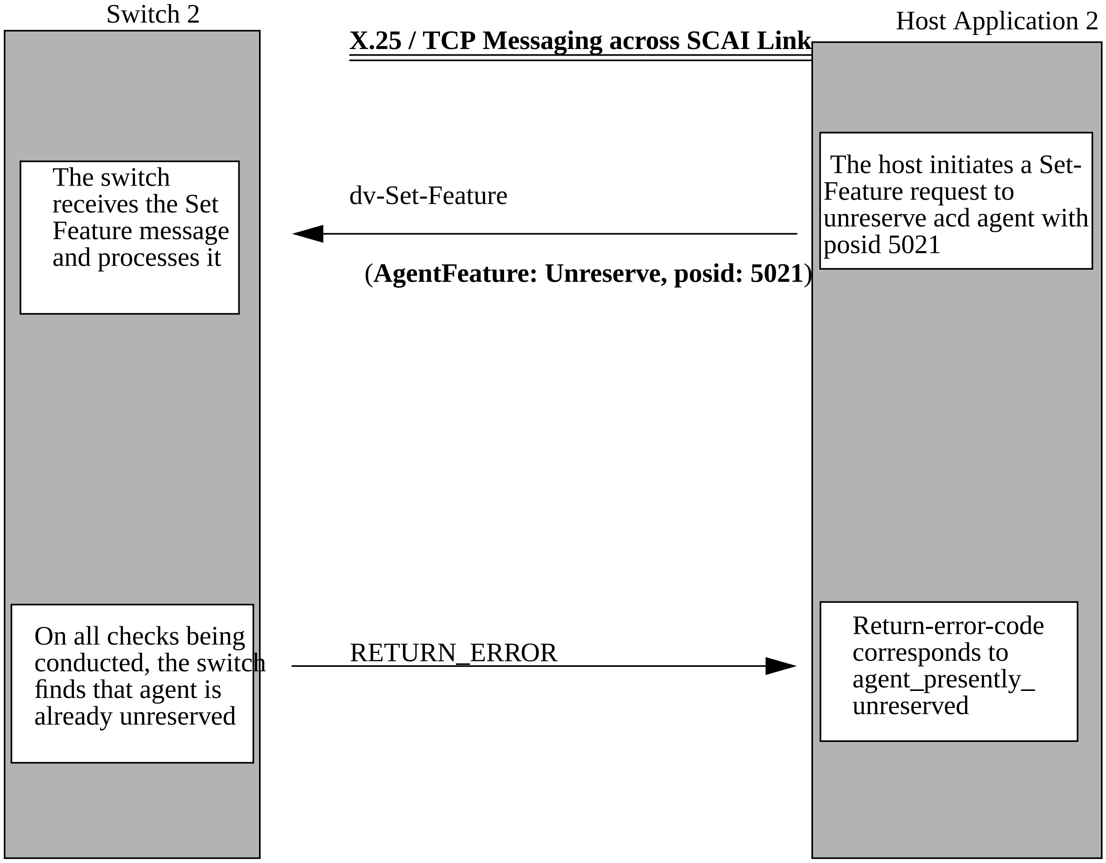
The switch2 will reserve the specified agent, if all subscription and other checks are passed and the agent can be reserved and upon doing so it will return to the host2 a return-result message.


Switch 2
Host Application 2
X.25 / TCP Messaging across SCAI Link
The switch receives the Set Feature message and processes it
dv-Set-Feature
The host initiates a Set-Feature request to reserve acd_agent having posid 5021
(agent_feature: reserve, posid: 5021, timer:20)
On all checks being passed, the switch will reserve posid 5021 and send a return result
RETURN_RESULT
Agent with posid: 5021 is reserved.

6.5.35.1 Set Feature (unreserved) with RE

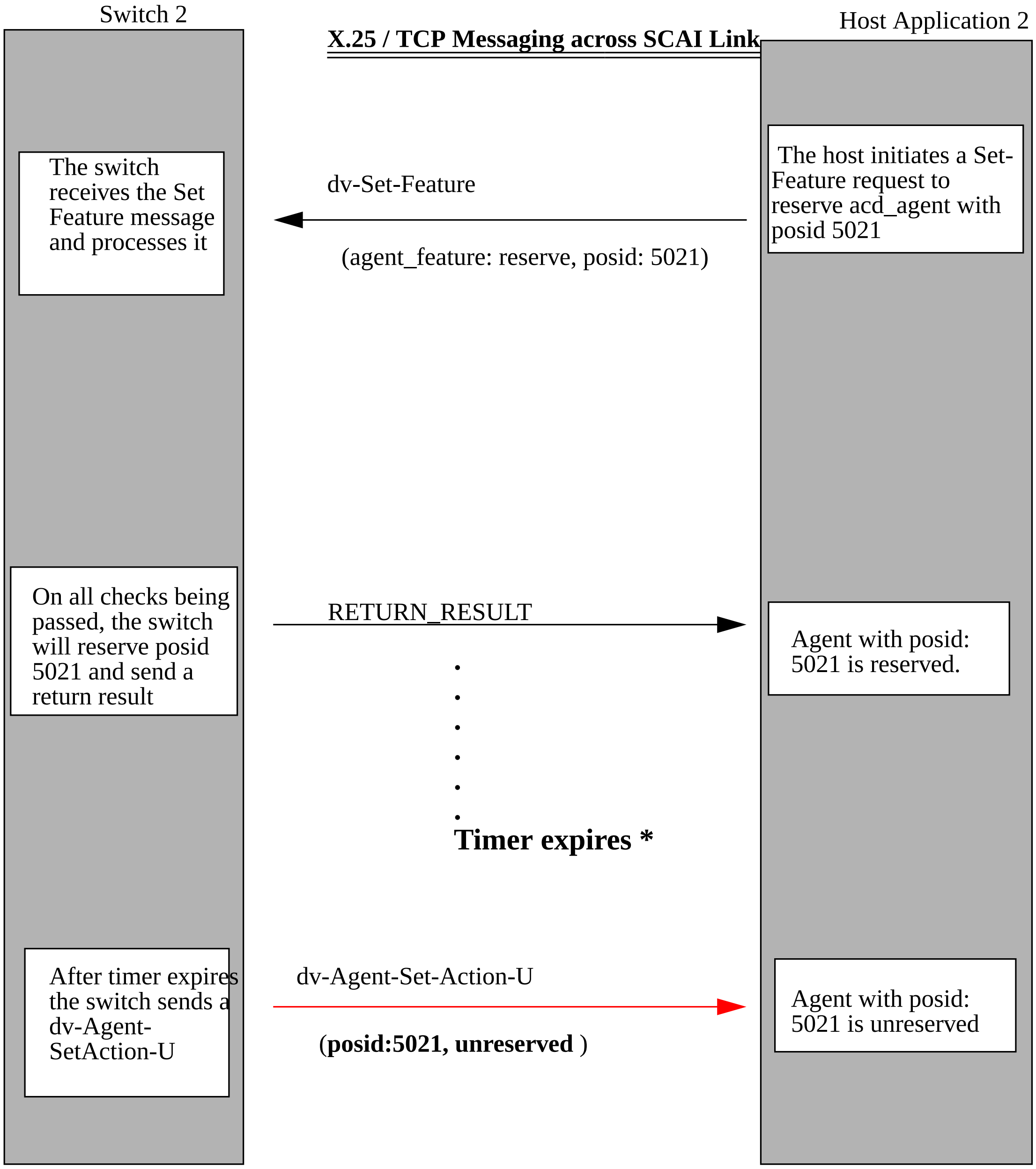
In this case, the host2 has already reserved an agent, when host1 informs (via LAN / WAN connectivity) that the reserved is not needed anymore, it may be unreserved. Therefore, host2 sends a dv-Set-Feature message (with unreserve parameter) requesting the switch to unreserve the specified agent.

The switch2 finds that agent is already in unreserved state and returns error message.


Switch 2
Host Application 2
X.25 / TCP Messaging across SCAI Link
The switch receives the Set Feature message and processes it
dv-Set-Feature
(AgentFeature: Unreserve, posid: 5021)
The host initiates a Set-Feature request to unreserve acd agent with posid 5021
On all checks being conducted, the switch finds that agent is already unreserved
RETURN_ERROR
Return-error-code corresponds to agent_presently_ unreserved

6.5.36 dv-Agent-SetAction-U message

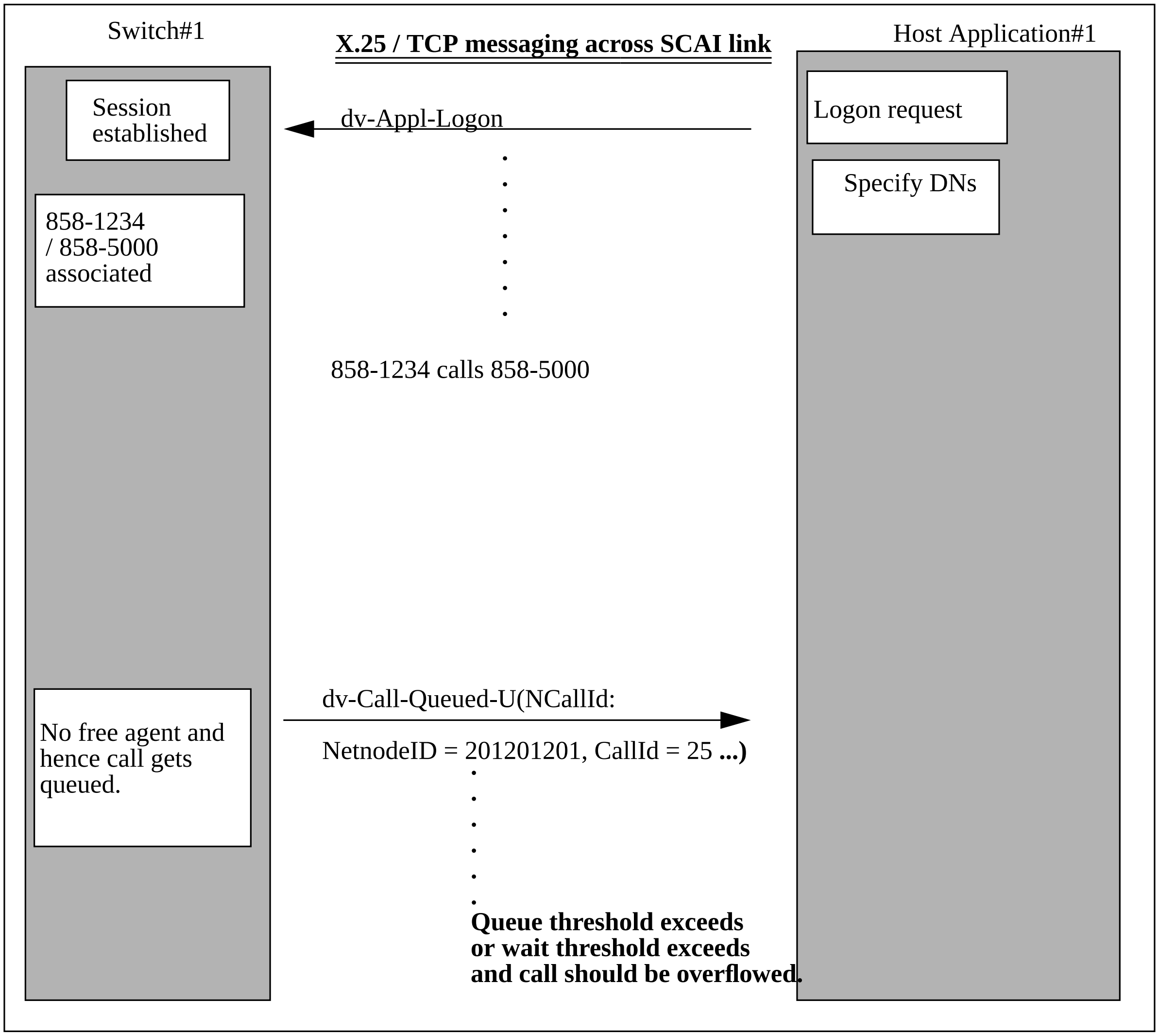
In this case, an agent been reserved using the Set-Feature(unreserve) message. After a certain preset time i.e., when timer expires, after approximately 20 seconds. (This timer starts when agent gets into reserved state) the switch2 decides to unreserve the agent. Hence it will unreserve the agent and send a dv-Unreserved-U message to the host2 informing the host2 that the agent with specified posid, which was reserved has now been unreserved.


Switch 2
X.25 / TCP Messaging across SCAI Link
Host Application 2
The switch receives the Set Feature message and processes it
dv-Set-Feature
(agent_feature: reserve, posid: 5021)
The host initiates a Set-Feature request to reserve acd_agent with posid 5021
On all checks being passed, the switch will reserve posid 5021 and send a return result
RETURN_RESULT
Agent with posid: 5021 is reserved.
Timer expires *
After timer expires the switch sends a dv-Agent-SetAction-U
dv-Agent-Set-Action-U
(posid:5021, unreserved)
Agent with posid: 5021 is unreserved

6.5.37 Overflowed from Queue or NACD (ACD customers only)

In this case, party A calls ACD group, and there are no idle agents available to answer the incoming call. The incoming call from party A will be placed in the queue. On reaching the maximum queue or wait thresholds, the call from party A will be overflowed to another ACD group.

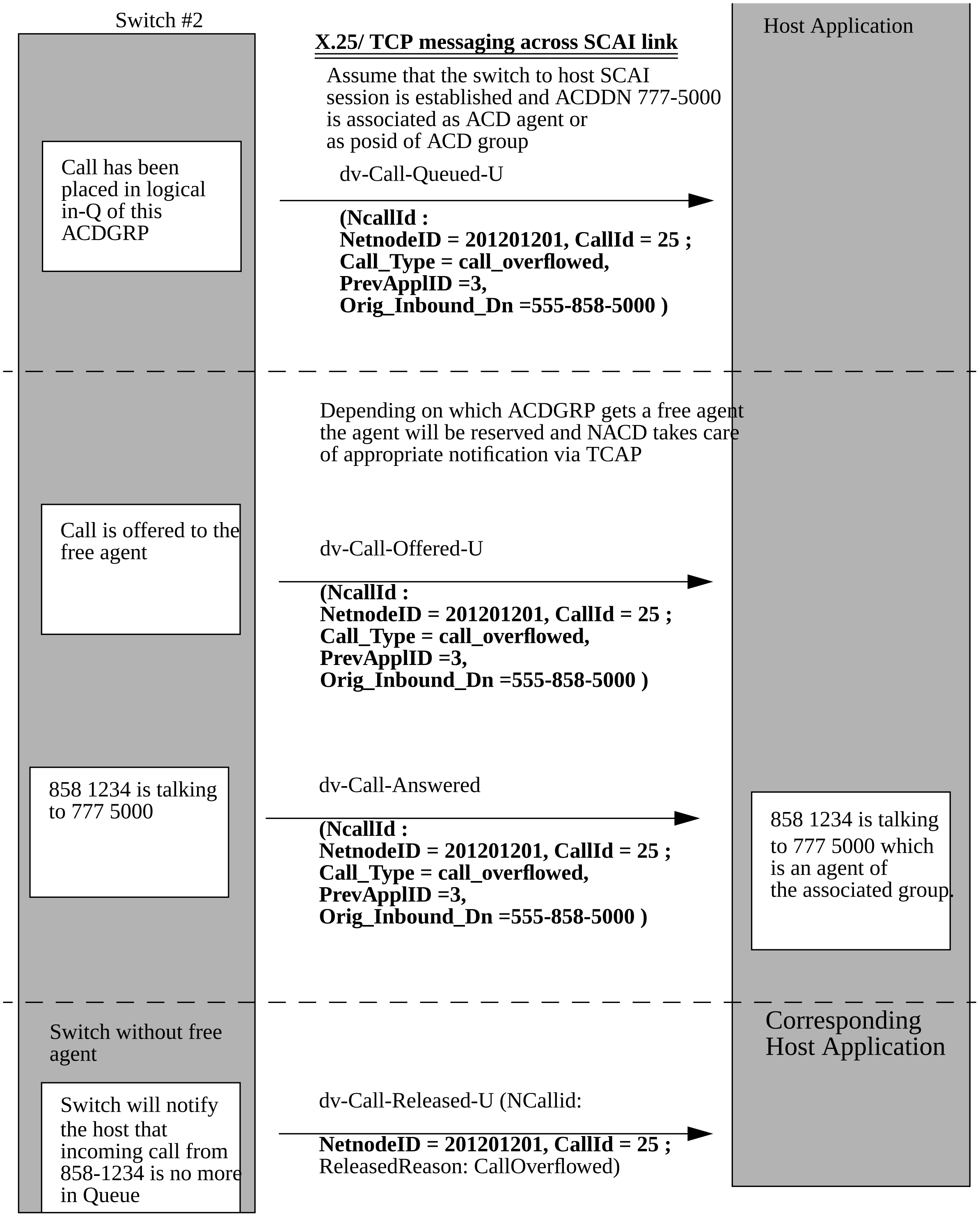
The messaging that takes place is shown in figure.


Switch#1
X.25 / TCP messaging across SCAI link
Host Application#1
Session established
dv-Appl-Logon
Logon request
Specify DNs
858-1234 / 858-5000 associated
858-1234 calls 858-5000
dv-Call-Queued-U(NCallId:
NetnodeID = 201201201, CallId = 25 ...)
No free agent and hence call gets queued.
Queue threshold exceeds
or wait threshold exceeds
and call should be overflowed.

6.5.38 Call Overflowed (Second Leg Significant)

The switch1 now sends IAM message to switch2 with the optional NETICM parameters encoded. The switch2 will reserve (**NACD reservation**) the first idle agent, and the rest of the ISUP messaging continues as in NACD.

When the call is finally offered to the agent at switch2, the following messaging will place between host2 and switch2.


Switch #2
Host Application
X.25/ TCP messaging across SCAI link
Corresponding Host Application

Assume that the switch to host SCAI session is established and ACDDN 777-5000 is associated as ACD agent or as posid of ACD group

Call has been placed in logical in-Q of this ACDGRP

dv-Call-Queued-U

(NcallId : NetnodeID = 201201201, CallId = 25 ; Call_Type = call_overflowed, PrevApplID =3, Orig_Inbound_Dn =555-858-5000)

Depending on which ACDGRP gets a free agent the agent will be reserved and NACD takes care of appropriate notification via TCAP

Call is offered to the free agent

dv-Call-Offered-U

(NcallId : NetnodeID = 201201201, CallId = 25 ; Call_Type = call_overflowed, PrevApplID =3, Orig_Inbound_Dn =555-858-5000)

858 1234 is talking to 777 5000

dv-Call-Answered

(NcallId : NetnodeID = 201201201, CallId = 25 ; Call_Type = call_overflowed, PrevApplID =3, Orig_Inbound_Dn =555-858-5000)

858 1234 is talking to 777 5000 which is an agent of the associated group.

Switch without free agent

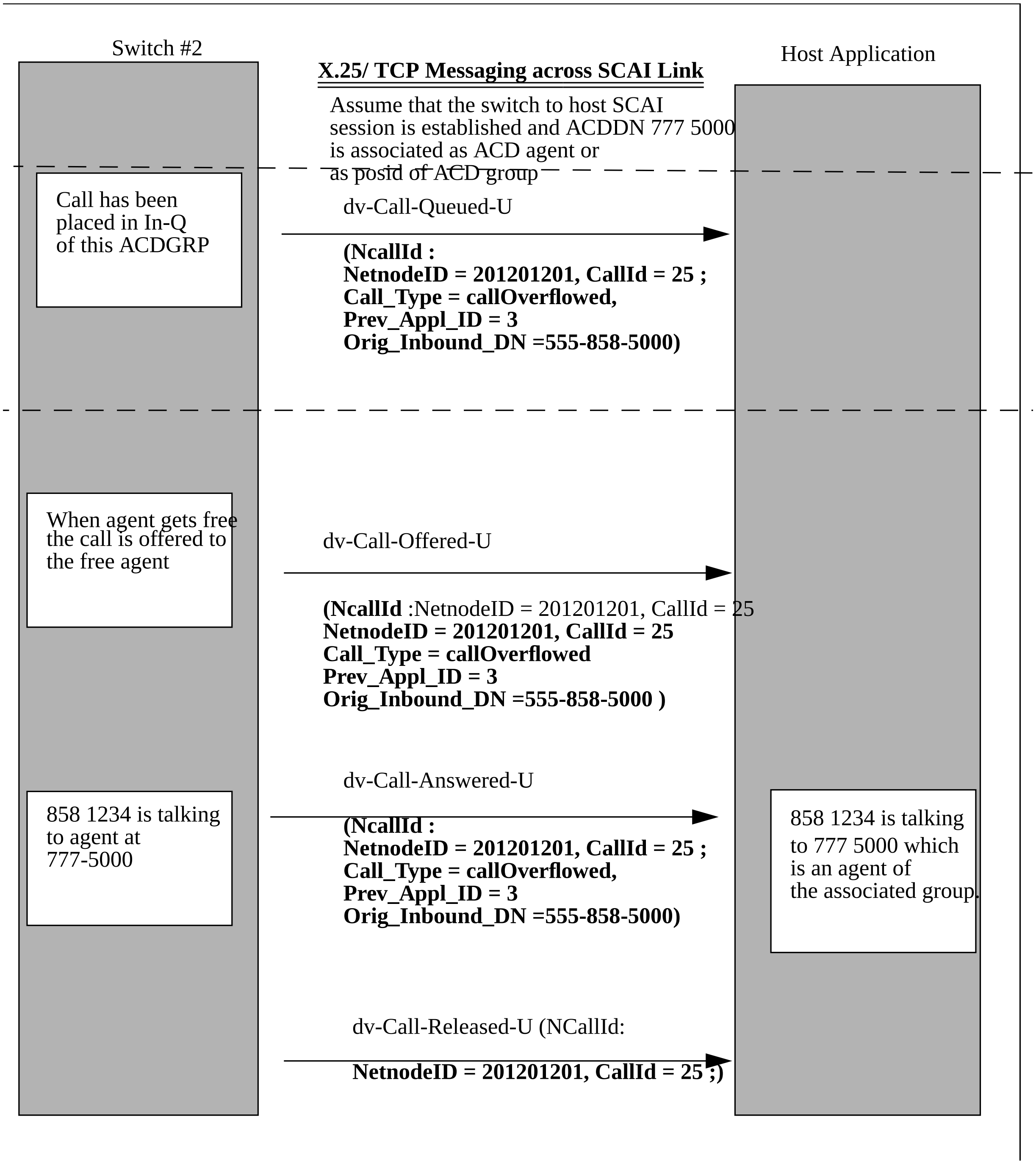
Switch will notify the host that incoming call from 858-1234 is no more in Queue

dv-Call-Released-U (NCallid:

NetnodeID = 201201201, CallId = 25 ;
ReleasedReason: CallOverflowed)

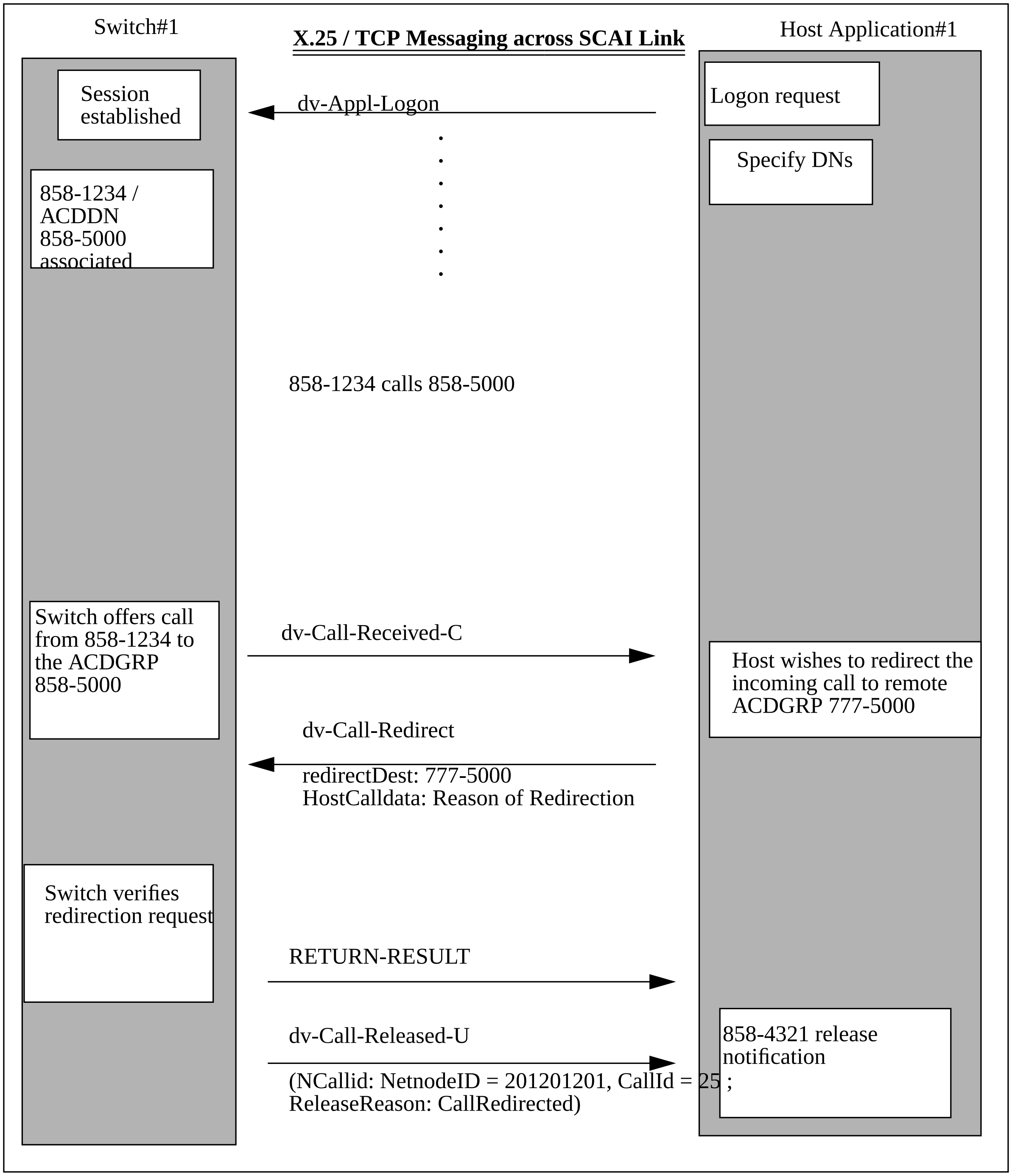
6.5.39 Overflowed - immediate (ACD customers only)

In this case party A calls ACD group, and there are no idle agents available to answer the incoming call. Also, the wait threshold or the queue threshold, of the queue has already exceeded. Party A will be overflowed immediately to another ACD group (actually NACD).There will be no ICM specific messaging to the host from which the call immediately overflowed.

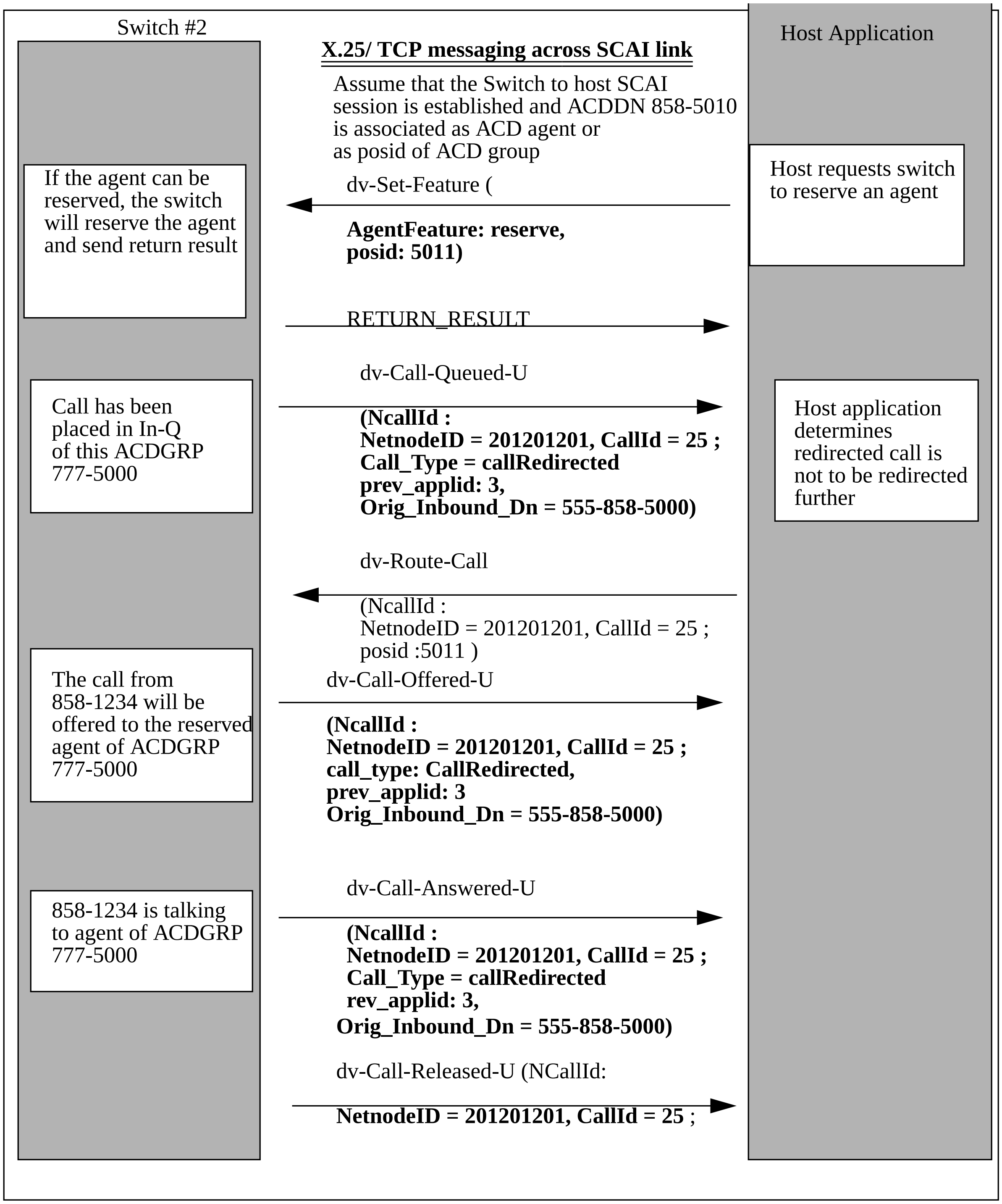

Switch #2
X.25/ TCP Messaging across SCAI Link
Host Application
Assume that the switch to host SCAI session is established and ACDDN 777 5000 is associated as ACD agent or as posid of ACD group
Call has been placed in In-Q of this ACDGRP
dv-Call-Queued-U
(NcallId : NetnodeID = 201201201, CallId = 25 ; Call_Type = callOverflowed, Prev_Appl_ID = 3 Orig_Inbound_DN =555-858-5000)
When agent gets free the call is offered to the free agent
dv-Call-Offered-U
(NcallId :NetnodeID = 201201201, CallId = 25 NetnodeID = 201201201, CallId = 25 Call_Type = callOverflowed Prev_Appl_ID = 3 Orig_Inbound_DN =555-858-5000)
858 1234 is talking to agent at 777-5000
dv-Call-Answered-U
(NcallId : NetnodeID = 201201201, CallId = 25 ; Call_Type = callOverflowed, Prev_Appl_ID = 3 Orig_Inbound_DN =555-858-5000)
858 1234 is talking to 777 5000 which is an agent of the associated group.
dv-Call-Released-U (NCallId: NetnodeID = 201201201, CallId = 25 ;)

6.5.40 Redirected (with agent reserved) (ACD customers only)

In this case party A calls ACD group, and the ACDGRP wishes to redirect the call to another ACDGRP on remote switch. The host at the remote has already reserved an agent (host1 informs host2 via LAN/WAN connectivity) and hence call is directly offered to the reserved agent. The messaging that takes place will be described.


Switch#1
X.25 / TCP Messaging across SCAI Link
Host Application#1
Session established
dv-Appl-Logon
Logon request
Specify DNs
858-1234 / ACDDN 858-5000 associated
858-1234 calls 858-5000
Switch offers call from 858-1234 to the ACDGRP 858-5000
dv-Call-Received-C
Host wishes to redirect the incoming call to remote ACDGRP 777-5000
dv-Call-Redirect
redirectDest: 777-5000
HostCalldata: Reason of Redirection
Switch verifies redirection request
RETURN-RESULT
dv-Call-Released-U
858-4321 release notification
(NCallid: NetnodeID = 201201201, CallId = 25 ;
ReleaseReason: CallRedirected)

6.5.41Call Redirected (Second Leg Significant)



Switch #2

X.25/ TCP messaging across SCAI link

Host Application

Assume that the Switch to host SCAI session is established and ACDDN 858-5010 is associated as ACD agent or as posid of ACD group

Host requests switch to reserve an agent

dv-Set-Feature (

AgentFeature: reserve, posid: 5011)

If the agent can be reserved, the switch will reserve the agent and send return result

RETURN_RESULT

dv-Call-Queued-U

(NcallId :
NetnodeID = 201201201, CallId = 25 ;
Call_Type = callRedirected
prev_applid: 3,
Orig_Inbound_Dn = 555-858-5000)

Call has been placed in In-Q of this ACDGRP 777-5000

Host application determines redirected call is not to be redirected further

dv-Route-Call

(NcallId :
NetnodeID = 201201201, CallId = 25 ;
posid :5011)

dv-Call-Offered-U

(NcallId :
NetnodeID = 201201201, CallId = 25 ;
call_type: CallRedirected,
prev_applid: 3
Orig_Inbound_Dn = 555-858-5000)

The call from 858-1234 will be offered to the reserved agent of ACDGRP 777-5000

dv-Call-Answered-U

(NcallId :
NetnodeID = 201201201, CallId = 25 ;
Call_Type = callRedirected
rev_applid: 3,
Orig_Inbound_Dn = 555-858-5000)

858-1234 is talking to agent of ACDGRP 777-5000

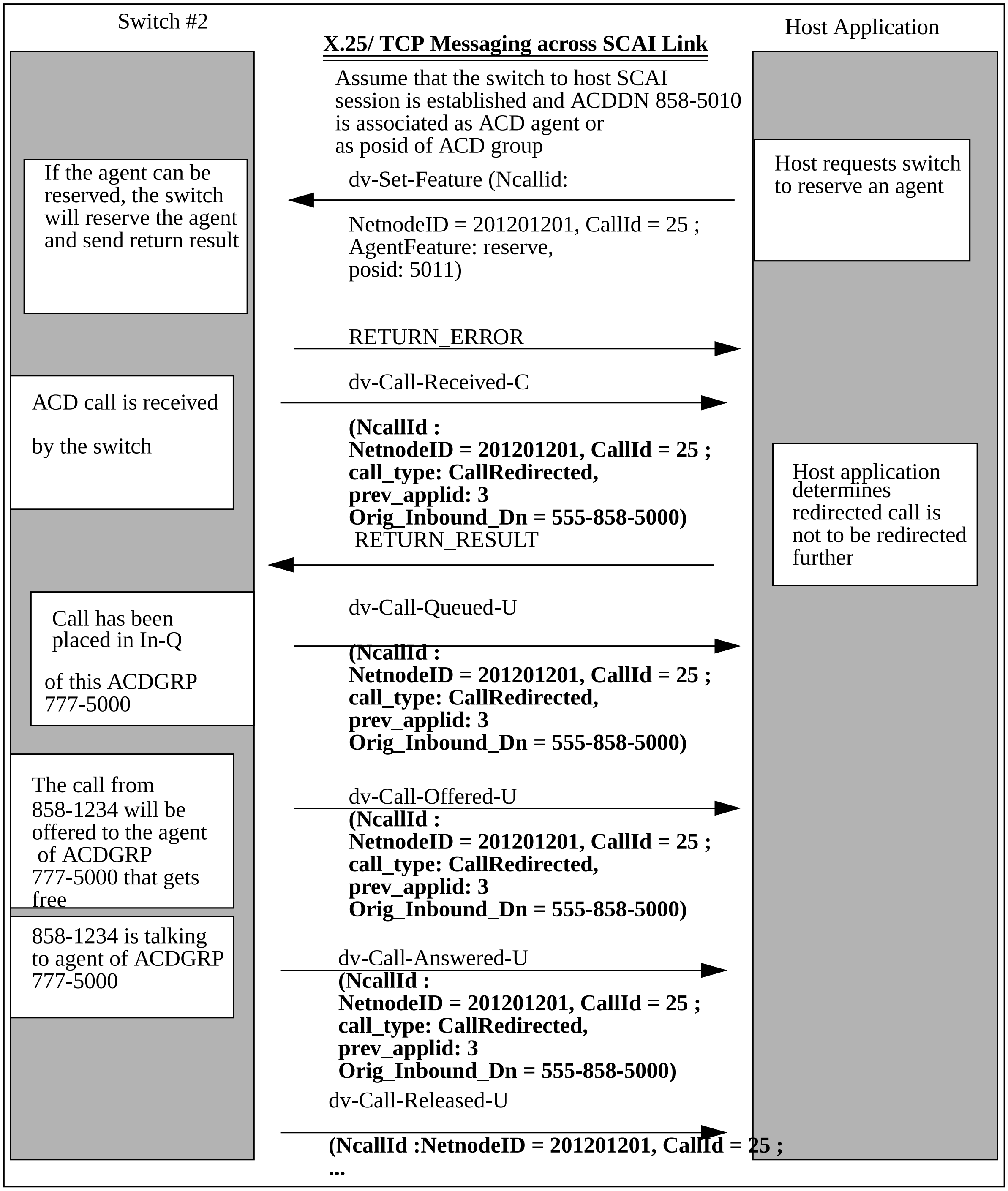
dv-Call-Released-U (NCallId:

NetnodeID = 201201201, CallId = 25 ;

6.5.42 Redirected with agent not reserved (ACD customers only)

In this case party A calls ACD group, and the ACDGRP wishes to redirect the call to another ACDGRP on remote switch. The host associated to remote switch cannot reserve an agent and hence the call may be queued at the remote switch. The messaging that takes place for the second leg is described.

Note: This scenario may be a rare one with Symposium management, because if the host at the target failed to reserve an agent, then the host at the source will not redirect it to the target host at all.



Assume that the switch to host SCAI session is established and ACDDN 858-5010 is associated as ACD agent or as posid of ACD group

Host requests switch to reserve an agent

dv-Set-Feature (Ncallid:

NetnodeID = 201201201, CallId = 25 ;
AgentFeature: reserve,
posid: 5011)

If the agent can be reserved, the switch will reserve the agent and send return result

RETURN_ERROR

ACD call is received

by the switch

dv-Call-Received-C

(NcallId :
NetnodeID = 201201201, CallId = 25 ;
call_type: CallRedirected,
prev_applid: 3
Orig_Inbound_Dn = 555-858-5000)

Host application determines redirected call is not to be redirected further

RETURN_RESULT

Call has been placed in In-Q

of this ACDGRP 777-5000

dv-Call-Queued-U

(NcallId :
NetnodeID = 201201201, CallId = 25 ;
call_type: CallRedirected,
prev_applid: 3
Orig_Inbound_Dn = 555-858-5000)

The call from 858-1234 will be offered to the agent of ACDGRP 777-5000 that gets free

dv-Call-Offered-U
(NcallId :
NetnodeID = 201201201, CallId = 25 ;
call_type: CallRedirected,
prev_applid: 3
Orig_Inbound_Dn = 555-858-5000)

858-1234 is talking to agent of ACDGRP 777-5000

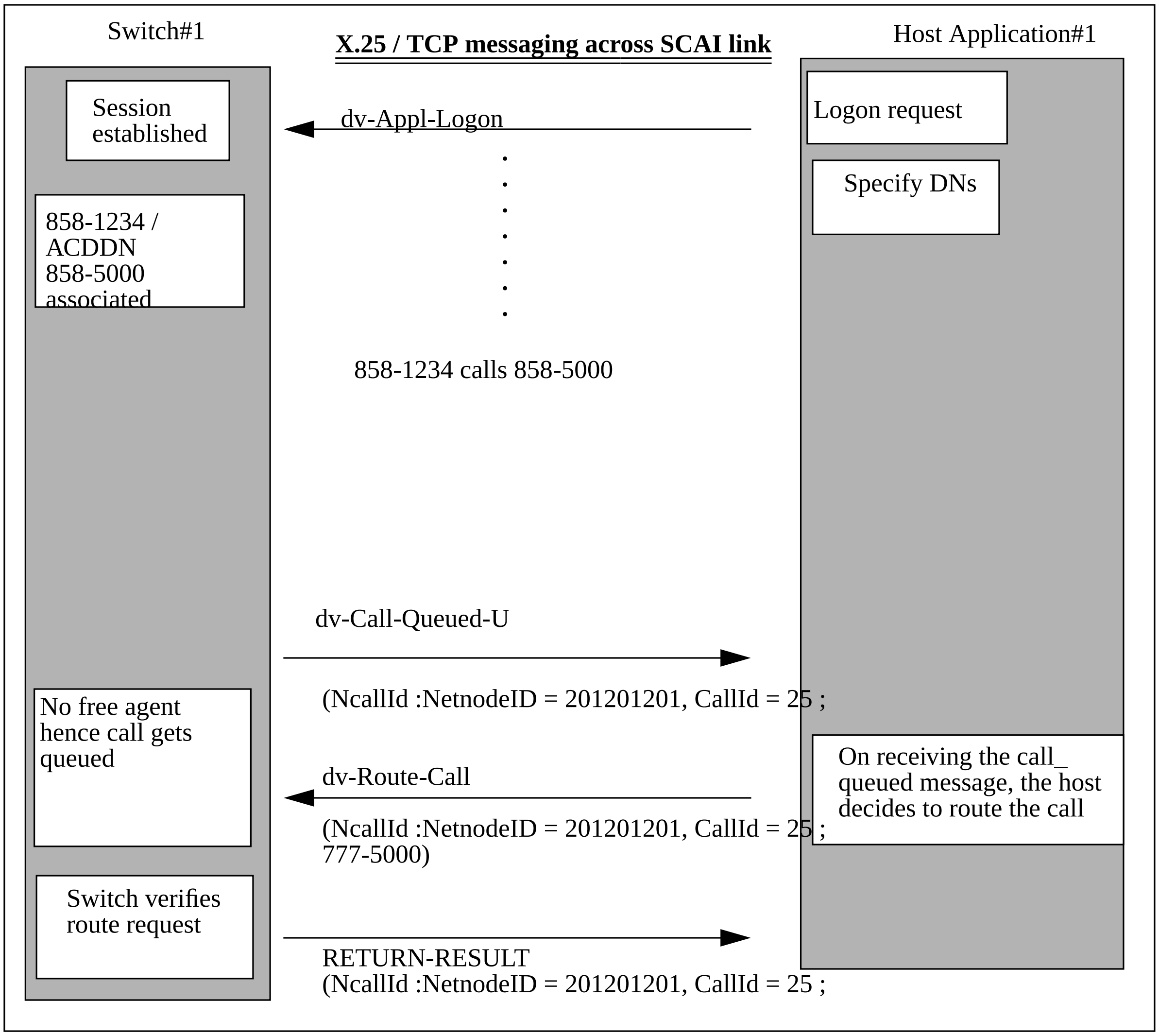
dv-Call-Answered-U
(NcallId :
NetnodeID = 201201201, CallId = 25 ;
call_type: CallRedirected,
prev_applid: 3
Orig_Inbound_Dn = 555-858-5000)

dv-Call-Released-U

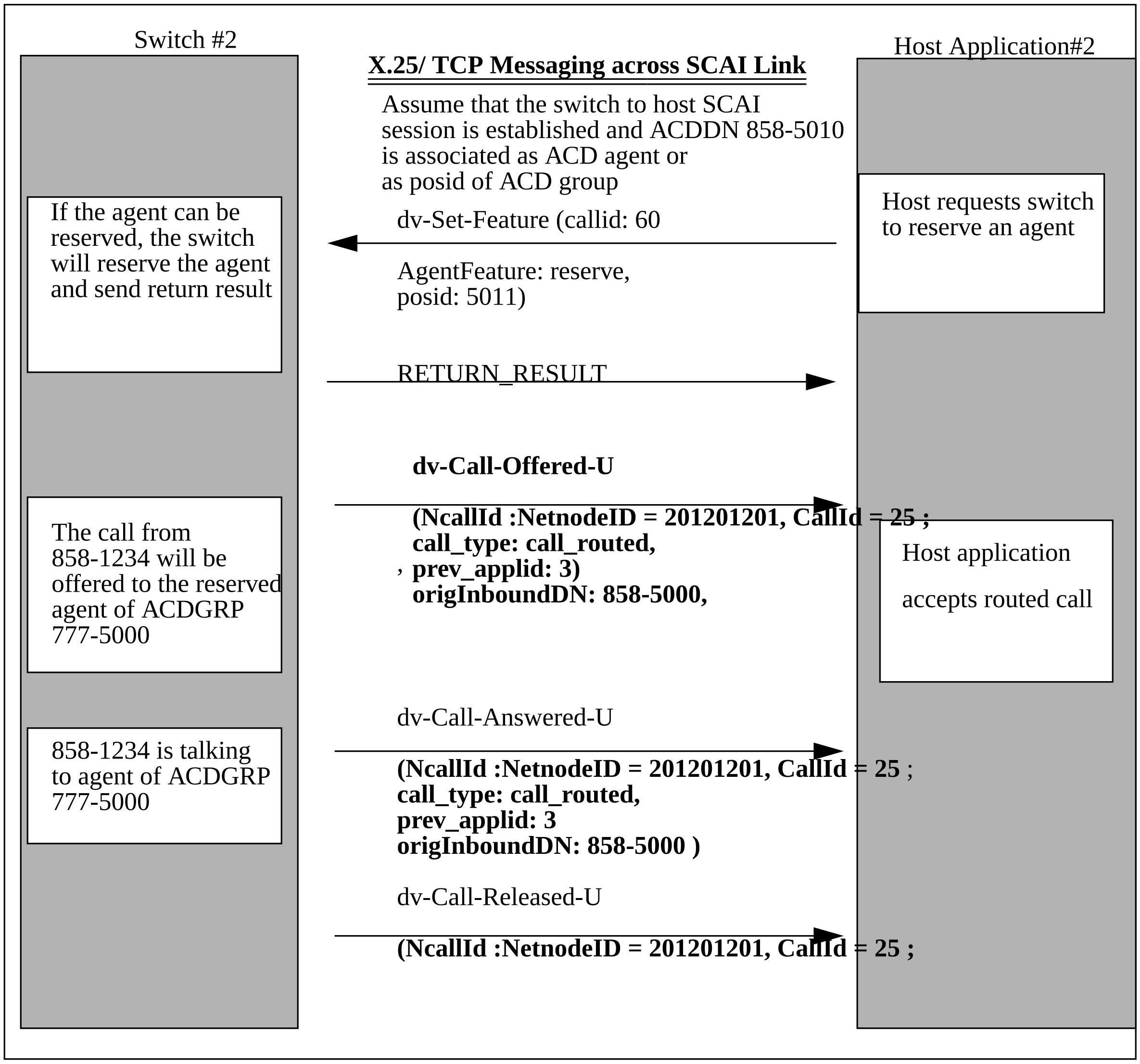
(NcallId :NetnodeID = 201201201, CallId = 25 ;
...

6.5.43 CallRouting - with agent reserved (ACD customers)

In this case party A calls party B, where party B is an ACDDN. When no agents are available, the incoming call gets queued. The host computer may decide to route the call to another party C at remote switch, which is another ACDDN. On routing the call, the host1 will also inform to host2 that a call is to be offered to host2. The host2 will therefore request the switch2 to reserve a specified agent. The switch validates the request and reserves the specified agent. In this case the call will be directly offered to the reserved agent.


Switch#1
X.25 / TCP messaging across SCAI link
Host Application#1
Session established
dv-Appl-Logon
Logon request
Specify DNs
858-1234 / ACDDN 858-5000 associated
858-1234 calls 858-5000
dv-Call-Queued-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;
No free agent hence call gets queued
dv-Route-Call
On receiving the call_ queued message, the host decides to route the call
(NcallId :NetnodeID = 201201201, CallId = 25 ; 777-5000)
Switch verifies route request
RETURN-RESULT
(NcallId :NetnodeID = 201201201, CallId = 25 ;

6.5.44 CallRouted by host computer with reserved agent (second Leg Significant)


Switch #2
Host Application#2
X.25/ TCP Messaging across SCAI Link
Assume that the switch to host SCAI session is established and ACDDN 858-5010 is associated as ACD agent or as posid of ACD group
Host requests switch to reserve an agent
dv-Set-Feature (callid: 60
AgentFeature: reserve, posid: 5011)
If the agent can be reserved, the switch will reserve the agent and send return result
RETURN_RESULT
dv-Call-Offered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: call_routed, prev_applid: 3) origInboundDN: 858-5000,
Host application accepts routed call
The call from 858-1234 will be offered to the reserved agent of ACDGRP 777-5000
dv-Call-Answered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: call_routed, prev_applid: 3 origInboundDN: 858-5000)
858-1234 is talking to agent of ACDGRP 777-5000
dv-Call-Released-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;

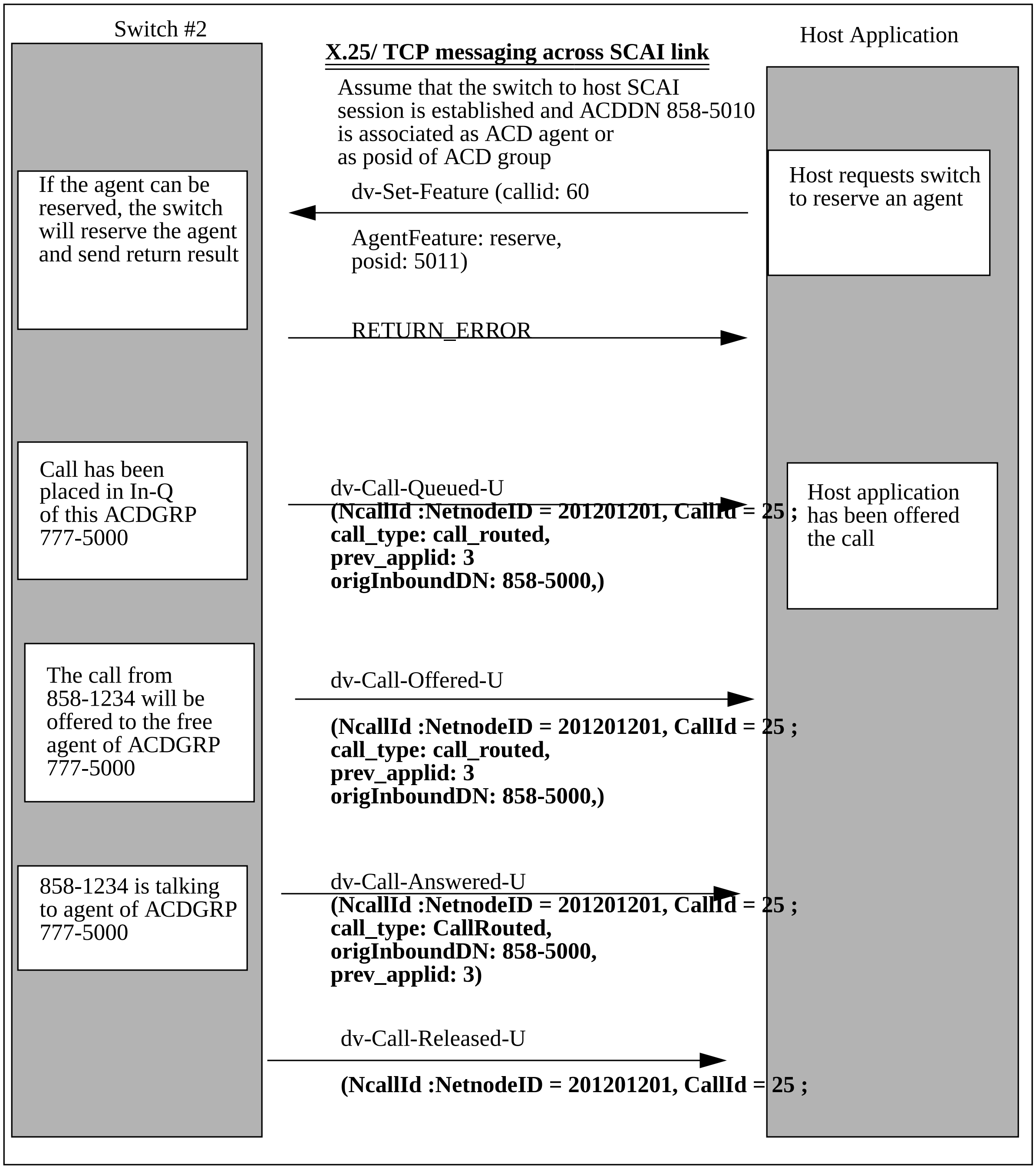
6.5.45 CallRouting with no reserved agent (ACD customers)

In this case party A calls party B, where party B is an ACDDN. When no agents are available, the incoming call gets queued. The host computer may decide to route the call to another party C at remote switch, which is another ACDDN.

On routing the call, the host1 will also inform to host2 that a call is to be offered to host2. The host2 will therefore request the switch to reserve a specified agent. The switch is unable to reserve the specified agent and sends return error to the host. In this case the call will be placed in the incoming queue of the remote group.

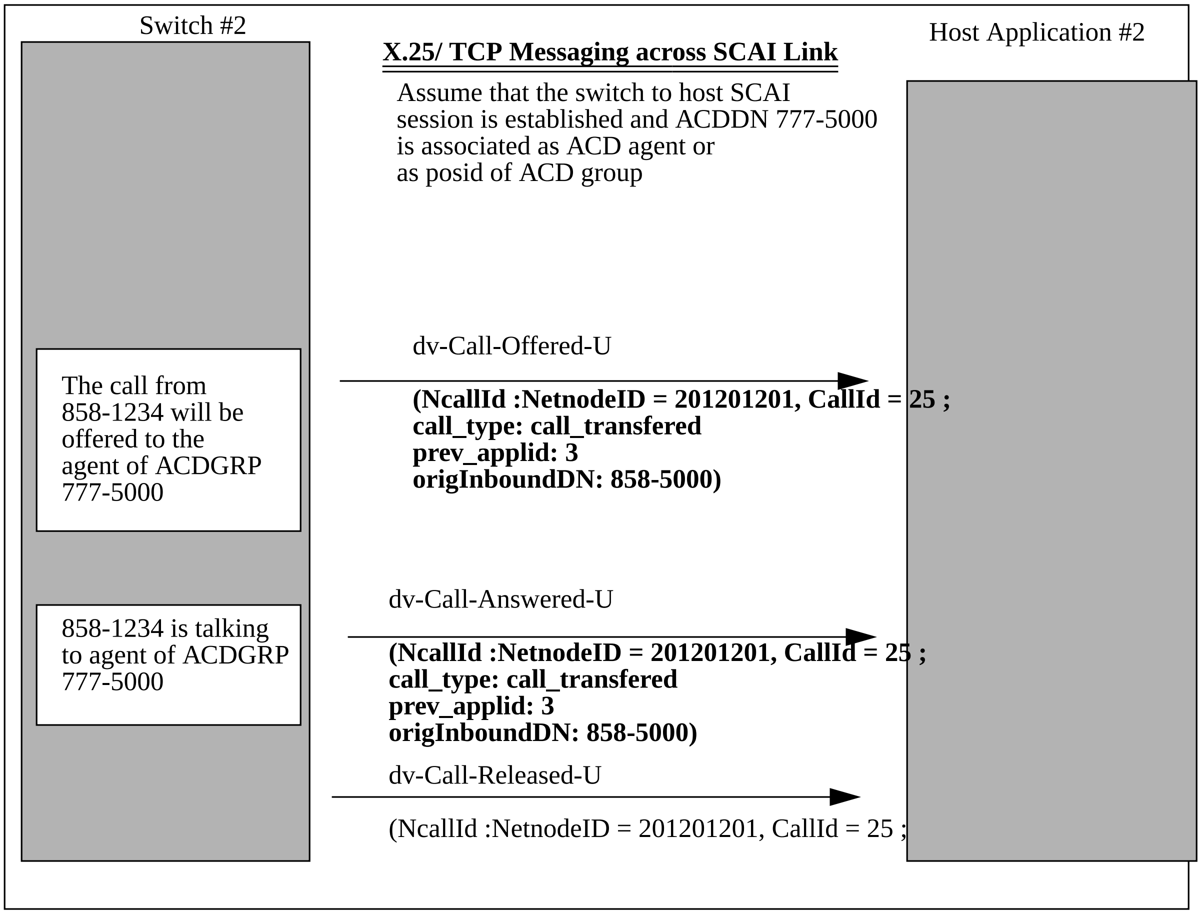
The messaging for the above is as shown

This scenario may be a rare one with Symposium management, because if the host at the target failed to reserve an agent, then the host at the source will not redirect it to this host at all.


Switch #2
Host Application
X.25/ TCP messaging across SCAI link
Assume that the switch to host SCAI session is established and ACDDN 858-5010 is associated as ACD agent or as posid of ACD group
If the agent can be reserved, the switch will reserve the agent and send return result
Host requests switch to reserve an agent
dv-Set-Feature (callid: 60
AgentFeature: reserve, posid: 5011)
RETURN_ERROR
Call has been placed in In-Q of this ACDGRP 777-5000
dv-Call-Queued-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: call_routed, prev_applid: 3 origInboundDN: 858-5000,)
Host application has been offered the call
The call from 858-1234 will be offered to the free agent of ACDGRP 777-5000
dv-Call-Offered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: call_routed, prev_applid: 3 origInboundDN: 858-5000,)
858-1234 is talking to agent of ACDGRP 777-5000
dv-Call-Answered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: CallRouted, origInboundDN: 858-5000, prev_applid: 3)
dv-Call-Released-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;

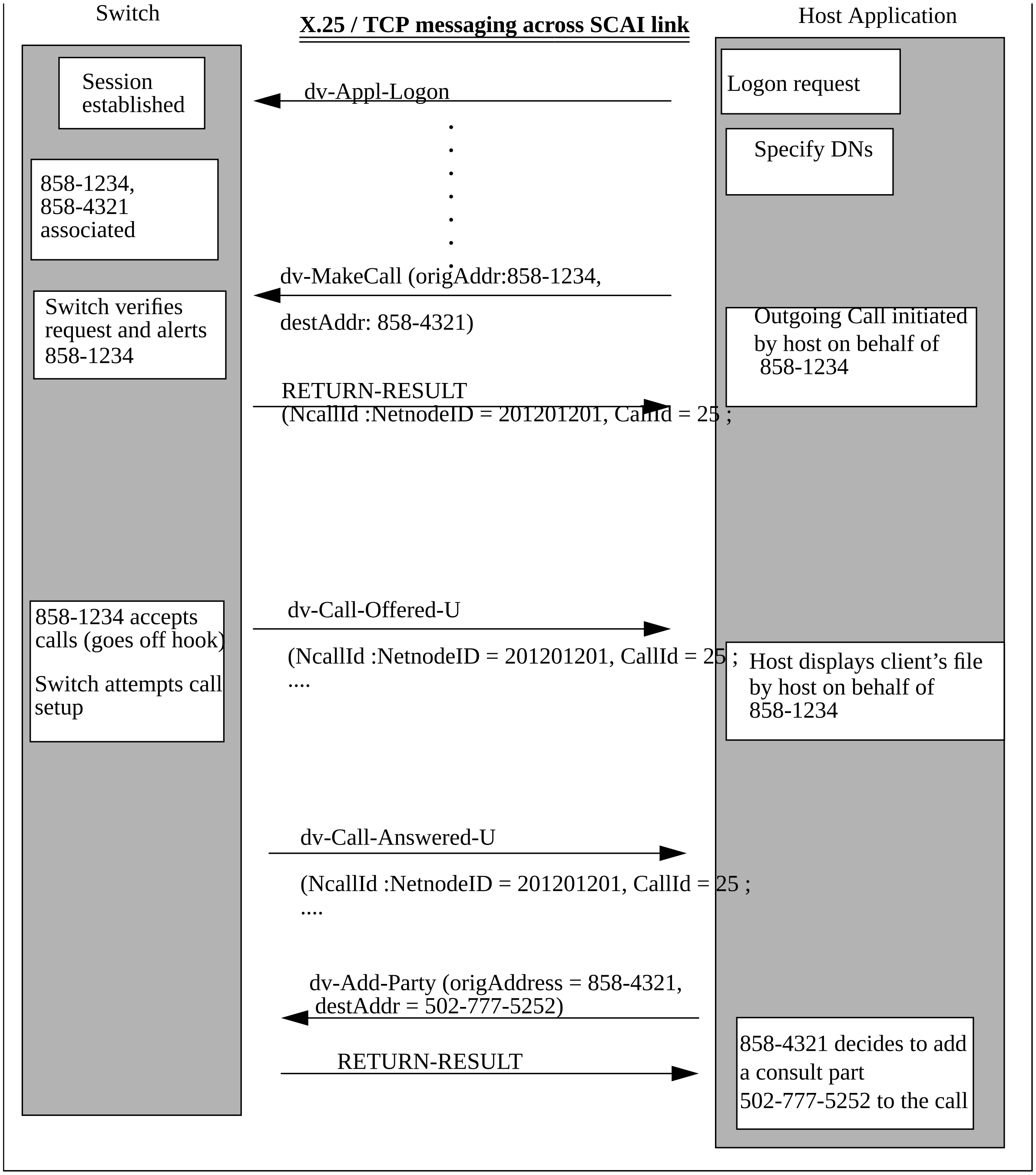
6.5.46 Transferred (ACD customers)

In this case party A calls party B, where party B is an ACD agent, and party B wishes to consult with party C which is associated to a session on another switch. In this case party A calls party B, where party B is an MDC/ACD agent, and party B wishes to consult with party C (ACD/MDC agent) which is associated to a session on another switch. For messaging between party A and B is similar to a 2way call. Once partyB initiates 3WC (either via ICM or manually) the following messaging take place.

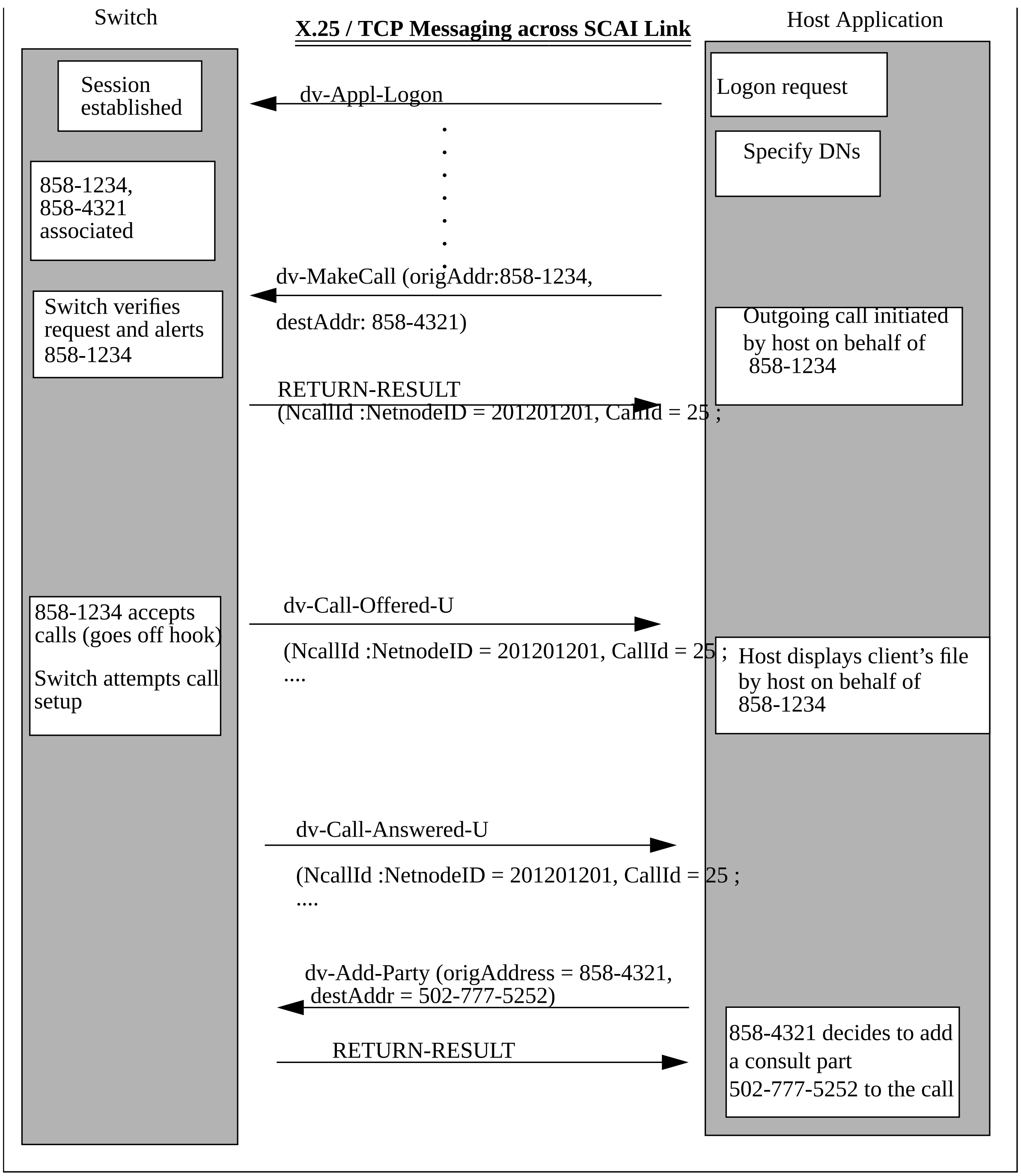

Switch #2
Host Application #2
X.25/ TCP Messaging across SCAI Link
Assume that the switch to host SCAI session is established and ACDDN 777-5000 is associated as ACD agent or as posid of ACD group
The call from 858-1234 will be offered to the agent of ACDGRP 777-5000
dv-Call-Offered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: call_transfered prev_applid: 3 origInboundDN: 858-5000)
858-1234 is talking to agent of ACDGRP 777-5000
dv-Call-Answered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ; call_type: call_transfered prev_applid: 3 origInboundDN: 858-5000)
dv-Call-Released-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;

6.5.47 Transferred (MDC/RES customers)

In this case party A calls party B, where party B is an MDC agent, and party B wishes to consult with party C which is associated to a session on another switch. For messaging between party A and B is similar to a 2way call. Once the partyB initiates 3WC (either via ICM or manually) the following messaging takes place as shown.


Switch
X.25 / TCP messaging across SCAI link
Host Application
Session established
dv-Appl-Logon
Logon request
Specify DNs
858-1234, 858-4321 associated
dv-MakeCall (origAddr:858-1234,
destAddr: 858-4321)
Switch verifies request and alerts 858-1234
Outgoing Call initiated by host on behalf of 858-1234
RETURN-RESULT
(NcallId :NetnodeID = 201201201, CallId = 25 ;
858-1234 accepts calls (goes off hook)
Switch attempts call setup
dv-Call-Offered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;
....
Host displays client’s file by host on behalf of 858-1234
dv-Call-Answered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;
....
dv-Add-Party (origAddress = 858-4321,
destAddr = 502-777-5252)
RETURN-RESULT
858-4321 decides to add a consult part 502-777-5252 to the call

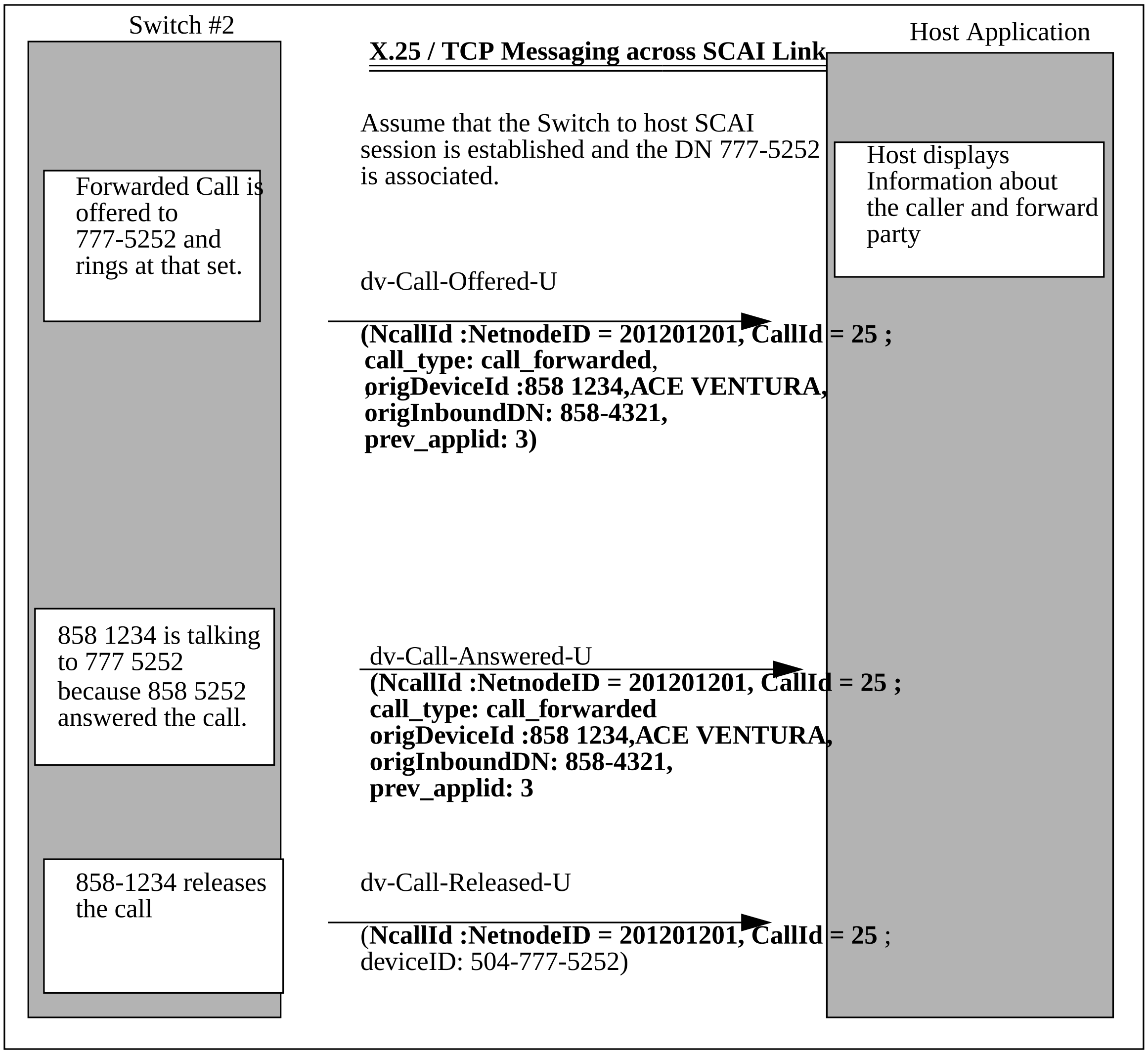
6.5.48 Call_Transferred (Second Leg Significant)


Switch
X.25 / TCP Messaging across SCAI Link
Host Application
Session established
Logon request
dv-Appl-Logon
Specify DNs
858-1234, 858-4321 associated
dv-MakeCall (origAddr:858-1234,
destAddr: 858-4321)
Switch verifies request and alerts 858-1234
Outgoing call initiated by host on behalf of 858-1234
RETURN-RESULT
(NcallId :NetnodeID = 201201201, CallId = 25 ;
dv-Call-Offered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;
....
858-1234 accepts calls (goes off hook)
Switch attempts call setup
Host displays client's file by host on behalf of 858-1234
dv-Call-Answered-U
(NcallId :NetnodeID = 201201201, CallId = 25 ;
....
dv-Add-Party (origAddress = 858-4321,
destAddr = 502-777-5252)
RETURN-RESULT
858-4321 decides to add a consult part 502-777-5252 to the call

6.5.49 Forwarded (MDC customers only)

In this case, party A calls party B, where party B is an MDC agent, and party B has activated Call Forwarded Don’t Answer to a number which resides on another switch. For messaging between party A and B is similar to a 2way call. Once the call forward time expires and the party B has not answered, call will be offered to the remote number as shown.

> ***Note:*** Only in case of CFDA (and not CFU, CFB, CFI) the incoming messages are observed for an associated line. If the line is not answered and the call is forwarded, a Call Released message will be sent with reason of Call Forwarded.



Assume that the Switch to host SCAI session is established and the DN 777-5252 is associated.

Host displays Information about the caller and forward party

Forwarded Call is offered to 777-5252 and rings at that set.

dv-Call-Offered-U

(NcallId :NetnodeID = 201201201, CallId = 25 ;
call_type: call_forwarded,
origDeviceId :858 1234,ACE VENTURA,
origInboundDN: 858-4321,
prev_applid: 3)

858 1234 is talking to 777 5252 because 858 5252 answered the call.

dv-Call-Answered-U

(NcallId :NetnodeID = 201201201, CallId = 25 ;
call_type: call_forwarded
origDeviceId :858 1234,ACE VENTURA,
origInboundDN: 858-4321,
prev_applid: 3

858-1234 releases the call

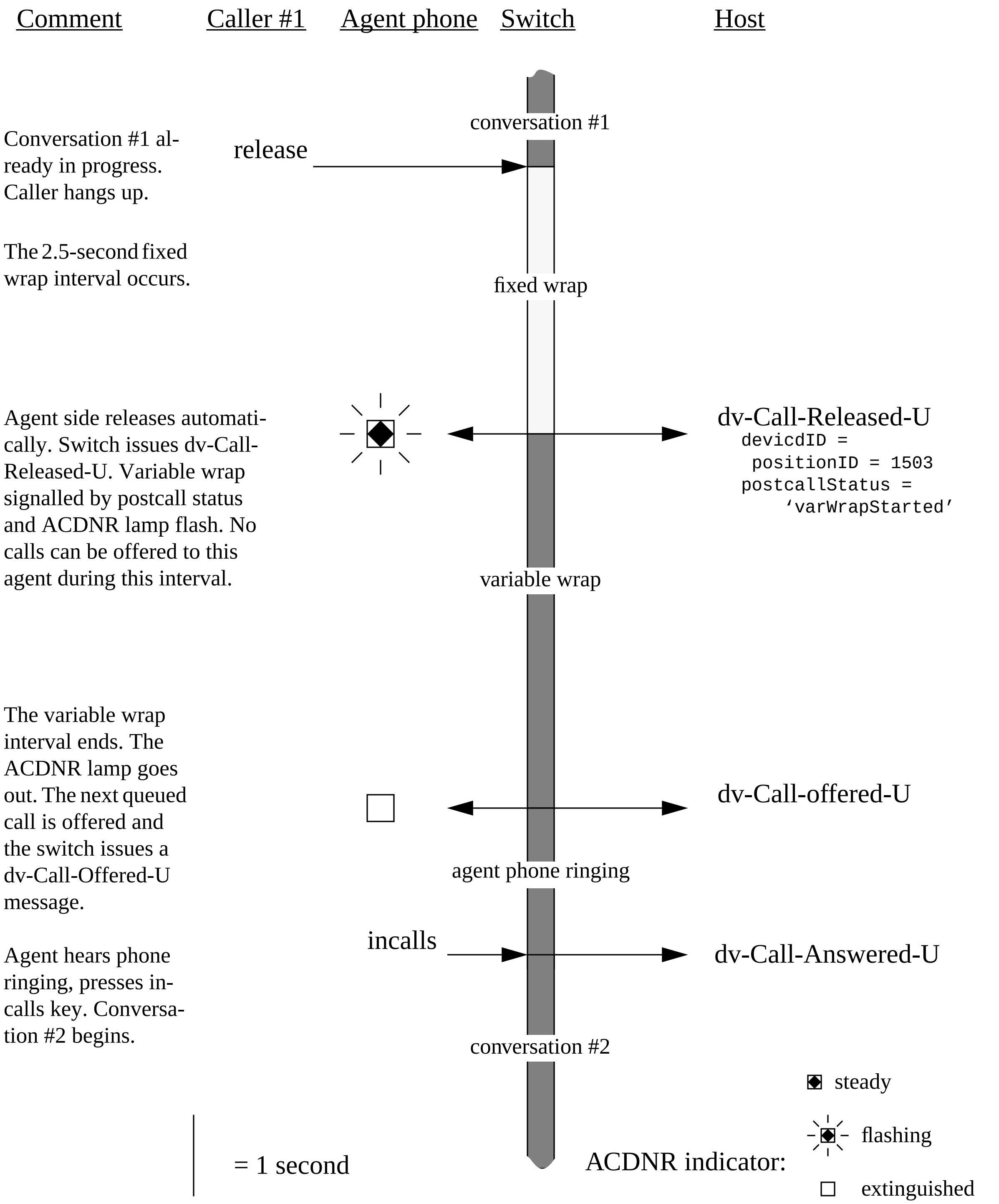
dv-Call-Released-U

(**NcallId :NetnodeID = 201201201, CallId = 25** ;
deviceID: 504-777-5252)

6.6 ICM Variable Wrap - message flow example sessions

6.6.1 Caller release, variable wrap, queued call (see Scenario 1)

Scenario 1



Conversation #1 already in progress. Caller hangs up.

The 2.5-second fixed wrap interval occurs.

Agent side releases automatically. Switch issues dv-Call-Released-U. Variable wrap signalled by postcall status and ACDNR lamp flash. No calls can be offered to this agent during this interval.

```
devicdID =
 positionID = 1503
postcallStatus =
    ‘varWrapStarted’
```

The variable wrap interval ends. The ACDNR lamp goes out. The next queued call is offered and the switch issues a dv-Call-Offered-U message.

Agent hears phone ringing, presses incalls key. Conversation #2 begins.

6.6.2 Caller release, variable wrap, no queued call

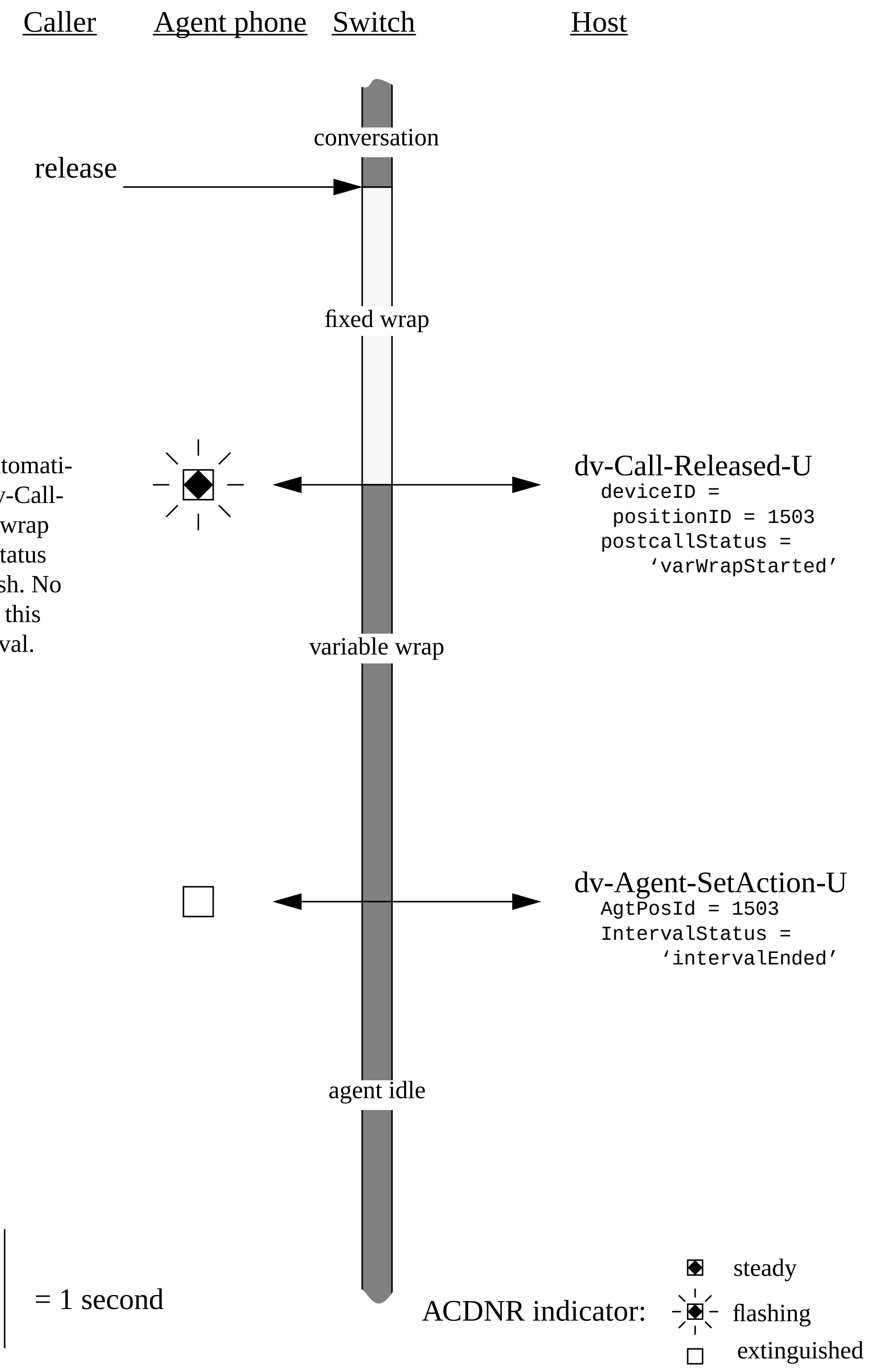
Comment

Conversation already in progress. Caller hangs up.

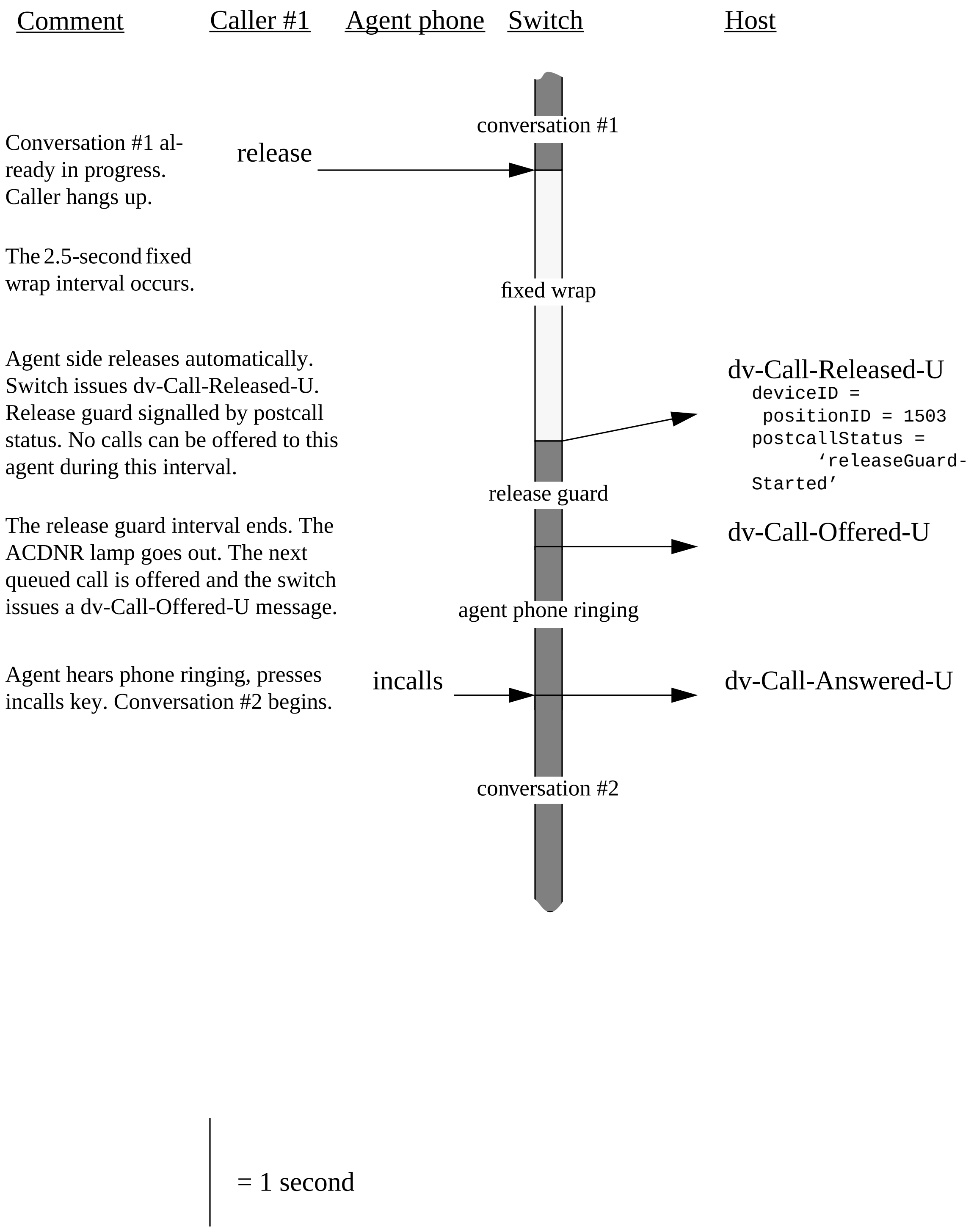
The 2.5-second fixed wrap interval occurs.

Agent side releases automatically. Switch issues dv-Call-Released-U. Variable wrap signalled by postcall status and ACDNR lamp flash. No calls can be offered to this agent during this interval.

The variable wrap interval ends. The ACDNR lamp goes out. Since there is no queued call, agent goes idle. dv-Agent-SetAction-U signals host of agent availability.



6.6.3 Caller release, release guard, queued call



Conversation #1 already in progress. Caller hangs up.

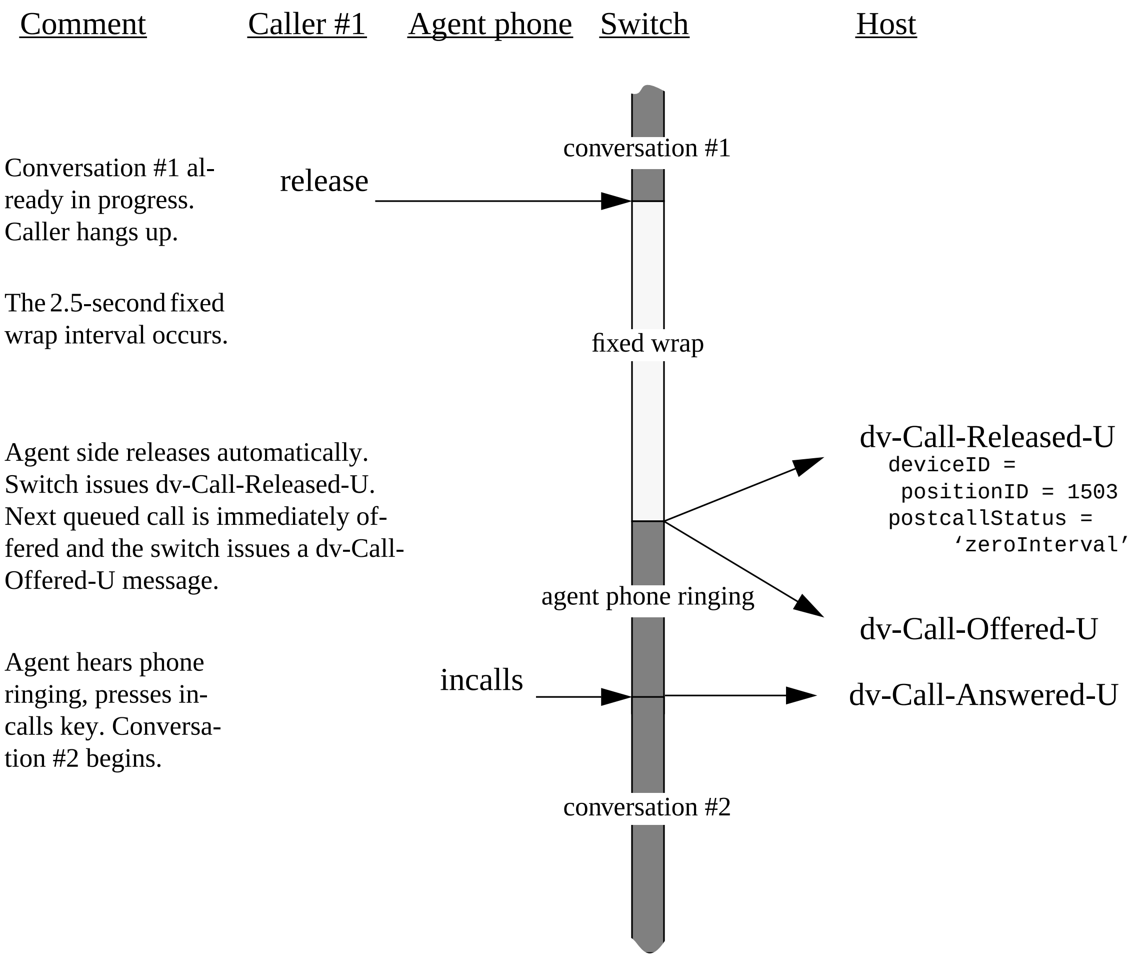
The 2.5-second fixed wrap interval occurs.

Agent side releases automatically. Switch issues dv-Call-Released-U. Release guard signalled by postcall status. No calls can be offered to this agent during this interval.

The release guard interval ends. The ACDNR lamp goes out. The next queued call is offered and the switch issues a dv-Call-Offered-U message.

Agent hears phone ringing, presses incalls key. Conversation #2 begins.

6.6.4 Caller release, zero vaiable wrap, queued call



Conversation #1 already in progress. Caller hangs up.

The 2.5-second fixed wrap interval occurs.

Agent side releases automatically. Switch issues dv-Call-Released-U. Next queued call is immediately offered and the switch issues a dv-Call-Offered-U message.

Agent hears phone ringing, presses incalls key. Conversation #2 begins.

= 1 second

7.0 Service model

7.1 Overview

This section describes the CompuCALL service model to provide a framework for the service functions and customer applications in Chapter 4.0 Application service options and CompuCALL session management procedures in Chapter 3.0 Session management options. The Service Model is described in terms of:

- the intent of the CompuCALL options
- key basic concepts
- key service design principles governing CompuCALL service-related procedures including, where relevant, exceptions to those principles

This Service Model is specifically for the CompuCALL options in this document. It is not a generic service model for future offerings of the CompuCALL options. Therefore, this model is subject to change in future releases of this document.

7.2 Intent

The CompuCALL options specified support "call center" applications.

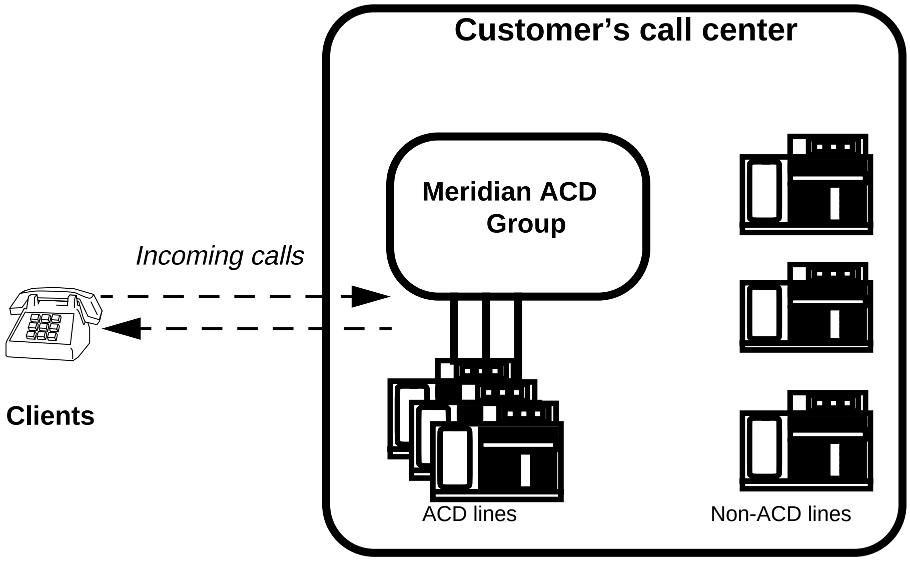
A call center is a group of ACD and/or non-ACD lines belonging to a customer which receive calls from and/or originate calls to "clients" of that customer (Figure 70). Call centers may serve either internal clients (belonging to the same customer) or external clients. However, call center clients are always considered to be outside the call center, i.e., not including members of the call center itself. Examples of call centers are:

- an airline reservation department (external clients)
- a manufacturer's service department (external clients)
- a computer "help desk" (internal clients)

Call centers may provide one or more functions. For example, an insurance company call center might handle calls regarding both policy changes and claims. Callers may dial different numbers to access the different functions of a call center. Or callers may dial a single number and one of the call center functions may be to "screen" incoming calls via interaction with the caller to determine the desired service.

A call center may include more than one ACD group and may consist of ACD groups and non-ACD lines all on the same switch or spread across two or more switches. The Meridian CompuCALL options herein support customer call centers with one or more ACD groups and/or a number of non-ACD lines all on a single switch. A customer may have more than one call center on the same switch. However, these call centers are considered unrelated. For example, a customer might have a customer relations department call center handling calls from external clients, plus a benefits department call center handling calls from employees. The Meridian CompuCALL options specified herein are for situations where a customer has one or more call centers on a switch.

Figure 70 Call center



Current release intended for “call center” applications

7.3 Concepts

7.3.1 CompuCALL customer

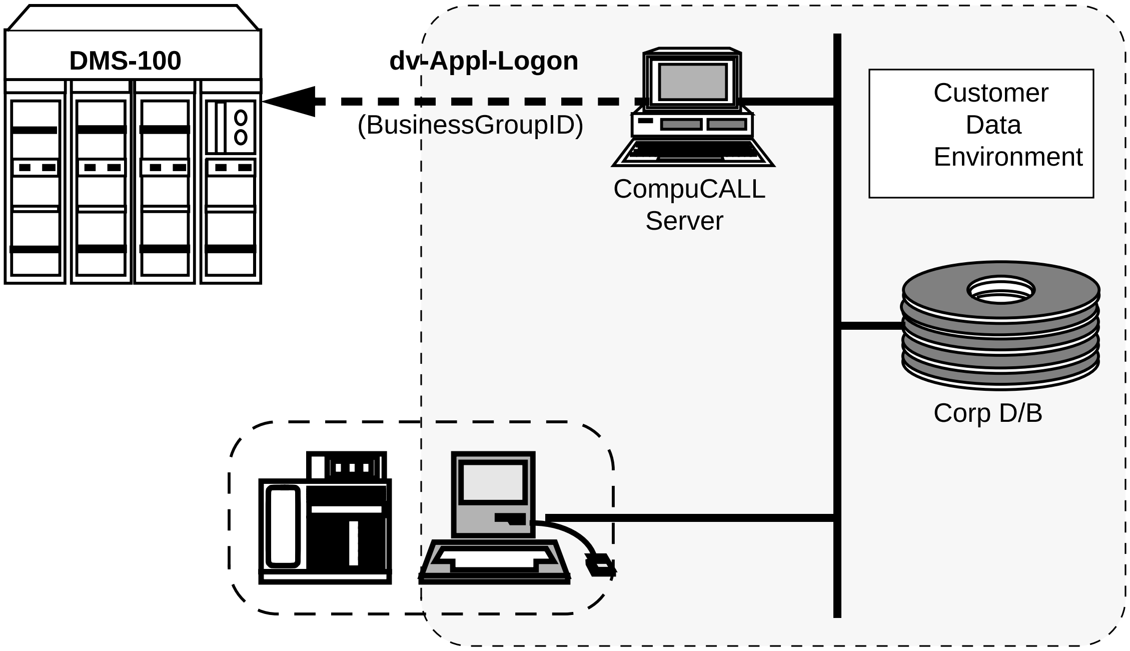
A "CompuCALL customer" is an Operating Company business customer (a company or a department within a company) who subscribes to one or more Meridian CompuCALL options. "Customer" refers to a CompuCALL customer. Currently CompuCALL services exist for the use of four types of customers:

- Automatic Call Distribution (ACD)
- Meridian Digital Centrex (MDC)
- Residential (RES)
- SL-100 Private Branch Exchange (PBX)

A CompuCALL customer's data environment (the customer's "host computer") establishes a CompuCALL session with the DMS using the dv-Appl-Logon message (Please refer to Section 3.2.1, " dv-Appl-Logon (operation value "1" hex) message," on page 85.). This message contains the BusinessGroupID parameter which identifies the customer and is mapped by the switch into an internal DMS-100 customer group (Figure 71).

There is a one-to-one correspondence between the BusinessGroupID and the internal DMS-100 customer group. However, a CompuCALL customer may have more than one DMS-100 customer group on a switch, possibly configured as a "customer Group family". Therefore, a CompuCALL customer may establish multiple CompuCALL sessions with the switch using different BusinessGroupIDs, each corresponding to a different DMS-100 customer Group. In addition, each BusinessGroupID may have multiple CompuCALL sessions established where each CompuCALL session is associated (Please refer to Section 3.2.2, "dv-DN-Associate (operation value "2" hex) message," on page 94.) with different ACD Groups (DN-associated with different ACD DNs) or non-ACD lines.

Figure 71 CompuCALL customer

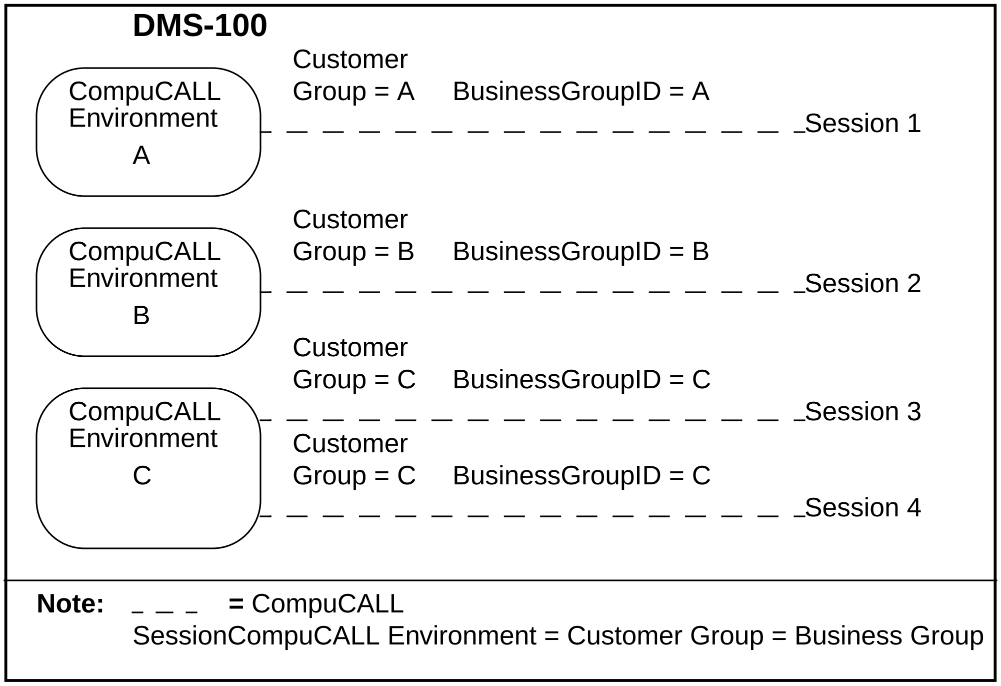


BusinessGroupID = DMS-100 Customer Group
CompuCALL Customer can have > 1 BGID (CG)

7.3.2CompuCALL environment

A "CompuCALL environment" exists on a single switch and is created when a CompuCALL customer establishes one or more active CompuCALL sessions with that switch using a given BusinessGroupID. A customer can have more than one CompuCALL session for a given CompuCALL environment (where these sessions all are established using the same BusinessGroupID). In addition, a customer can create multiple CompuCALL environments on the same switch by establishing multiple CompuCALL sessions with different BusinessGroupIDs. There is a one-to-one correspondence between BusinessGroupIDs, DMS-100 customer groups and CompuCALL environments. The relationship among CompuCALL sessions, BusinessGroupIDs, DMS-100 customer groups, and CompuCALL environments is shown in Figure 72.

Figure 72 CompuCALL environmental relationships



A CompuCALL environment consists of a number of ACD groups and related ACD lines and/or a number of non-ACD lines. A CompuCALL environment on a switch is comprised of (Please refer to Figure 73, “CompuCALL environment,” on page 710.):

- all the ACD groups on that switch which have been assigned to the DMS-100 customer Group corresponding to that CompuCALL environment
- all the ACD lines on that switch which are logged in and which have been assigned to one of the ACD groups in “a”
- all the MDC lines on that switch which have been assigned to the DMS-100 customer Group which corresponds to that CompuCALL environment and which have been "DN-associated" (Section 7.3.4, “Associated environment,” on page 711) with one of that CompuCALL environment's CompuCALL sessions

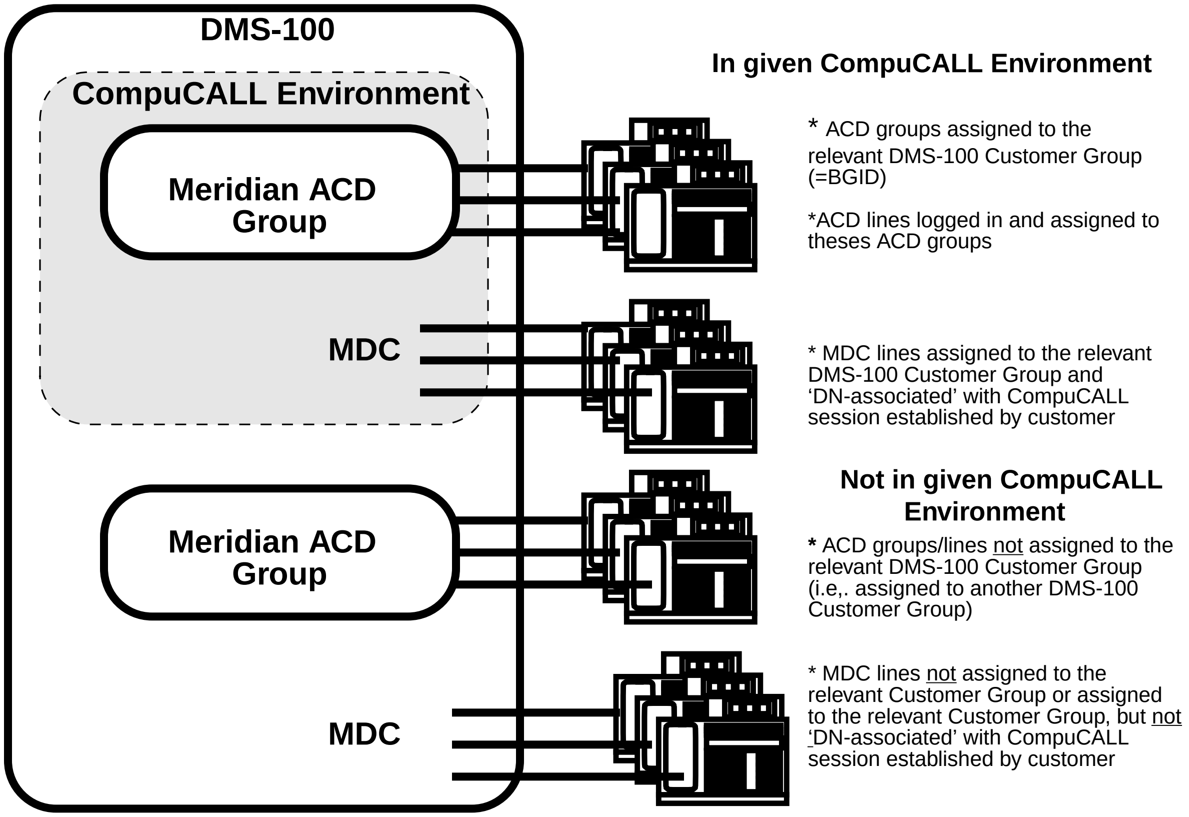
  Note: MDC lines refers to the non-ACD lines supported by the Meridian MDC CompuCALL Option (see Section 4.1, “Application service overview,” on page 121).

The CompuCALL environment is significant since:

- A call which enters or is originated in a customer's CompuCALL environment becomes a CompuCALL call (Section Figure 76 , “Relationship among CC, CE, and AE,” on page 715).

- Acustomer can use Third Party Call Control (TPCC) for the ACD lines in his CompuCALL environment.
- A customer can receive CompuCALL call-related event messages for calls to or from ACD groups, ACD lines, and MCS lines in his environment if the customer subscribed to those messages.

Figure 73 CompuCALL environment



In given CompuCALL Environment

* ACD groups assigned to the relevant DMS-100 Customer Group (=BGID)

*ACD lines logged in and assigned to theses ACD groups

* MDC lines assigned to the relevant DMS-100 Customer Group and ‘DN-associated’ with CompuCALL session established by customer

Not in given CompuCALL Environment

* ACD groups/lines not assigned to the relevant DMS-100 Customer Group (i.e,. assigned to another DMS-100 Customer Group)

* MDC lines not assigned to the relevant Customer Group or assigned to the relevant Customer Group, but not ‘DN-associated’ with CompuCALL session established by customer

7.3.3 Party and CompuCALL party

Party refers to any network addressable entity which can initiate or terminate a call (a "call" is defined in Section 7.3.5). Examples of parties are:

- a normal business line
- an AC
- an ACD line (addressed directly or indirectly via its ACD group)

The CompuCALL TPCC messages dv-Make-Call, dv-Add-Party, dv-Transfer-Party, dv-Conference-Party, and dv-Drop-Party (Chapter 1, Section 2.1.3) all specify a party which already is, or is requested to be, involved with a call.

A party is involved in a call when that party:

- initiates a call (goes off hook)
- is identified by the switch as the destination of a call
- is being alerted of a call attempt
- is connected to a call

A party identified as the destination of a call is involved with the call even if the call hasn’t been answered by that party. For example, an ACD group is involved with a call which has been routed to it, even if the call hasn’t been queued for that ACD group.

A "CompuCALL party" is a party in a customer's CompuCALL environment, i.e., an ACD group, ACD line, or MDC line. Therefore, all of the parties in a CompuCALL environment (Figure 73) are CompuCALL parties.

CompuCALL parties may, in turn, contain as members other CompuCALL parties. An ACD group in a customer's CompuCALL environment is a CompuCALL party. Furthermore, each ACD line assigned to that ACD group is also a CompuCALL party.

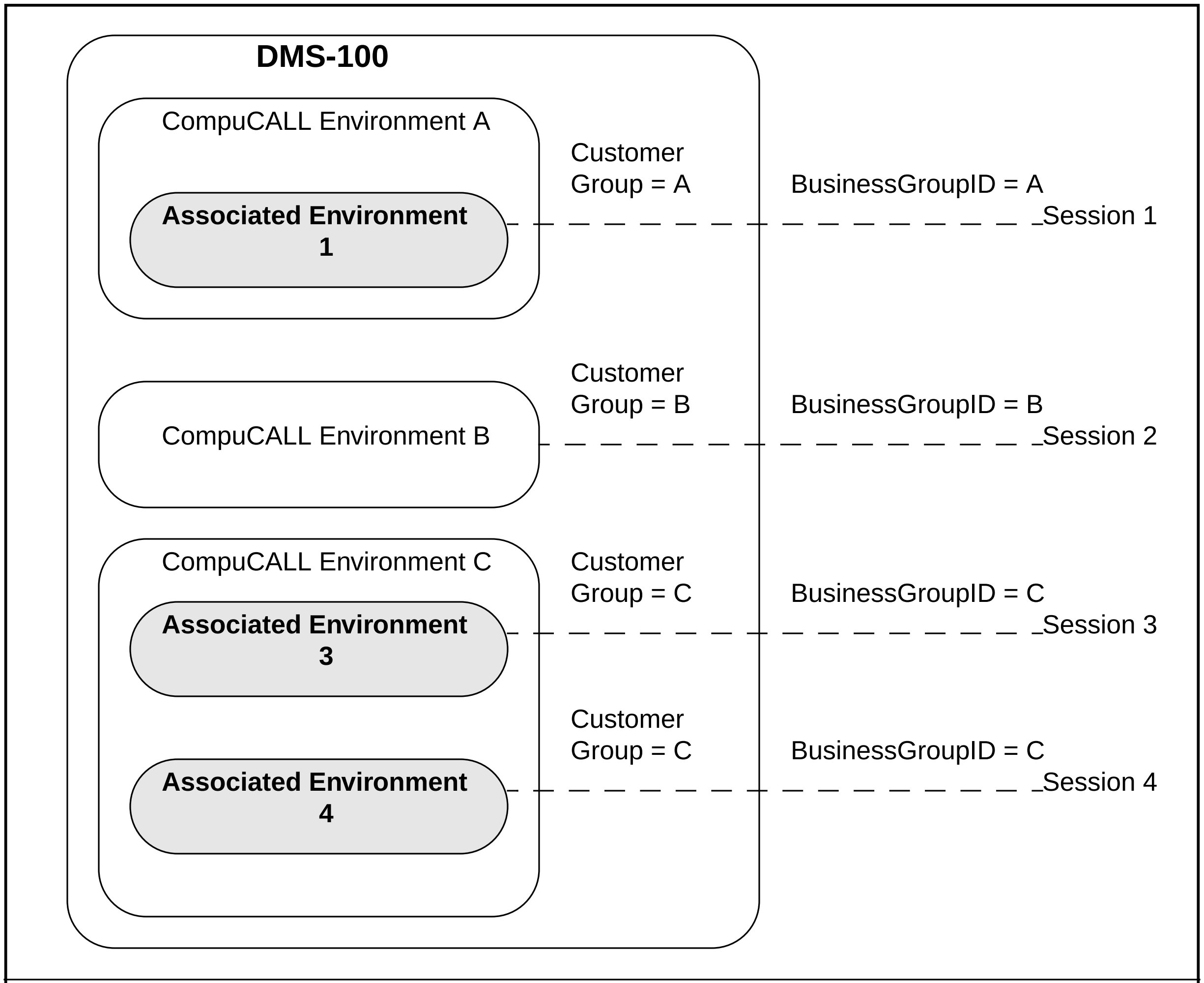
7.3.4 Associated environment

An "Associated environment" exists on a single switch and consists of those ACD groups and MDC lines in a customer's CompuCALL environment (CompuCALL parties) which have been "DN-associated" with a given CompuCALL session. "DN-associated" refers to the customer's host computer sending the dv-DN-Associate message over that session, as described in Section 3.2.2. When an ACD group is DN-associated with a CompuCALL session, all of the ACD lines which are logged in and assigned to that group are automatically implicitly DN-associated with that session.

There is a one-to-many correspondence between Associated environments and CompuCALL sessions. Since a CompuCALL environment may correspond to more than one CompuCALL session (i.e., all established using the same BusinessGroupID), a CompuCALL environment may consist of more than one associated environment. An associated environment may consist of one or more CompuCALL sessions that are established using the same BusinessGroupID and the same/different service IDs. The relationship between CompuCALL sessions, associated environments, and CompuCALL environments is shown in Figure 74.

> ***Note:*** A CompuCALL session may exist without any ACD groups or MDC lines having been DN-associated with it (i.e., with no corresponding Associated environment). Session 2 in Figure 70 is an example of a CompuCALL session with no corresponding Associated environment.

Figure 74 CompuCALL sessions, associated environments, and CompuCALL environments



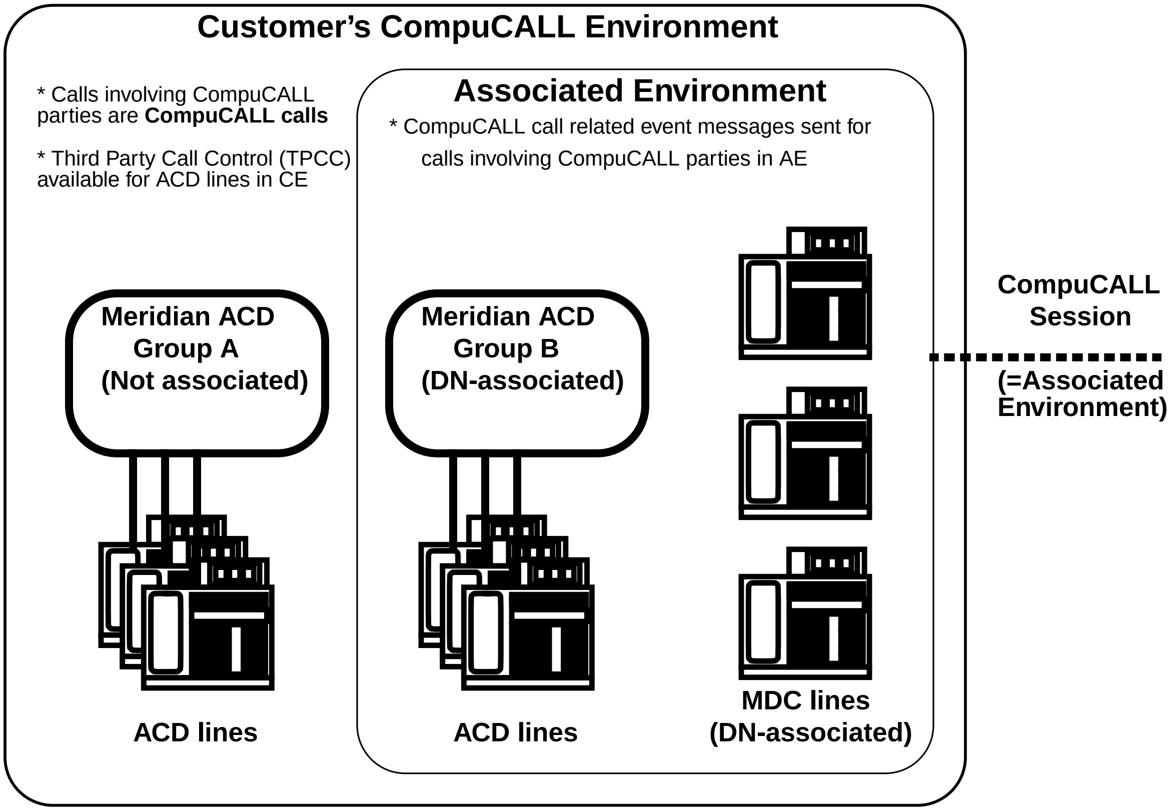
Note: _ _ _ _ CompuCALL Session
Associated Environment = CompuCALL Session

An ACD group or MDC line can be DN-associated with only one CompuCALL session at any point in time. Therefore, associated environments never overlap, or include the same CompuCALL parties even if they are in the same CompuCALL environment.

ACD groups do not have to be DN-associated with a CompuCALL session to be part of a CompuCALL environment. However, MDC lines must be DN-associated with a CompuCALL session to be part of a CompuCALL environment. Therefore, a customer's CompuCALL environment may include ACD groups which are DN-associated (within an associated environment) as well as other ACD groups which are not DN-associated and therefore not within any associated environment.

Figure 71 below shows a customer's CompuCALL environment and one of its associated environments and the significance of each. CompuCALL call-related event messages are sent only for calls to or from ACD groups, ACD lines, and MDC lines (i.e., CompuCALL parties) which are in an associated environment.

Figure 75 Significance of customer's CompuCALL environment and associated environment


Customer's CompuCALL Environment
* Calls involving CompuCALL parties are **CompuCALL calls**
* Third Party Call Control (TPCC) available for ACD lines in CE
Associated Environment
* CompuCALL call related event messages sent for calls involving CompuCALL parties in AE
Meridian ACD Group A (Not associated)
Meridian ACD Group B (DN-associated)
ACD lines
ACD lines
MDC lines (DN-associated)
CompuCALL Session
(=Associated Environment)

Reasons why a CompuCALL customer might have multiple associated environments in the same CompuCALL environment include:

- the customer wants different service profiles for different ACD groups and establishes multiple CompuCALL sessions using different ServiceIDs (Please refer to Section 3.1, “Session management options overview,” on page 83.)
- the customer requires multiple CompuCALL sessions to carry the volume of CompuCALL messages for a given CompuCALL environment

7.3.5 Meridian CompuCALL options support of call centers

7.3.5.1 Relationship between call centers and CompuCALL environments

The Meridian CompuCALL options support situations where:

- there is a one-to-one correspondence between a CompuCALL environment and a customer's call center (all the ACD groups and/or MDC lines in the call center are assigned to the same DMS-100 customer group, and no ACD group and/or MDC line in another call center is assigned to the same DMS-100 customer group)
- a call center corresponds to more than one CompuCALL environment (the call center includes ACD groups and/or MDC lines assigned to different DMS-100 customer groups)
- a CompuCALL environment corresponds to more than one call center (all the ACD groups and/or MDC lines in these call centers are assigned to the same DMS-100 customer group)

Figure 72 shows various examples of the possible relationships among call centers, CompuCALL environments, and associated environments.

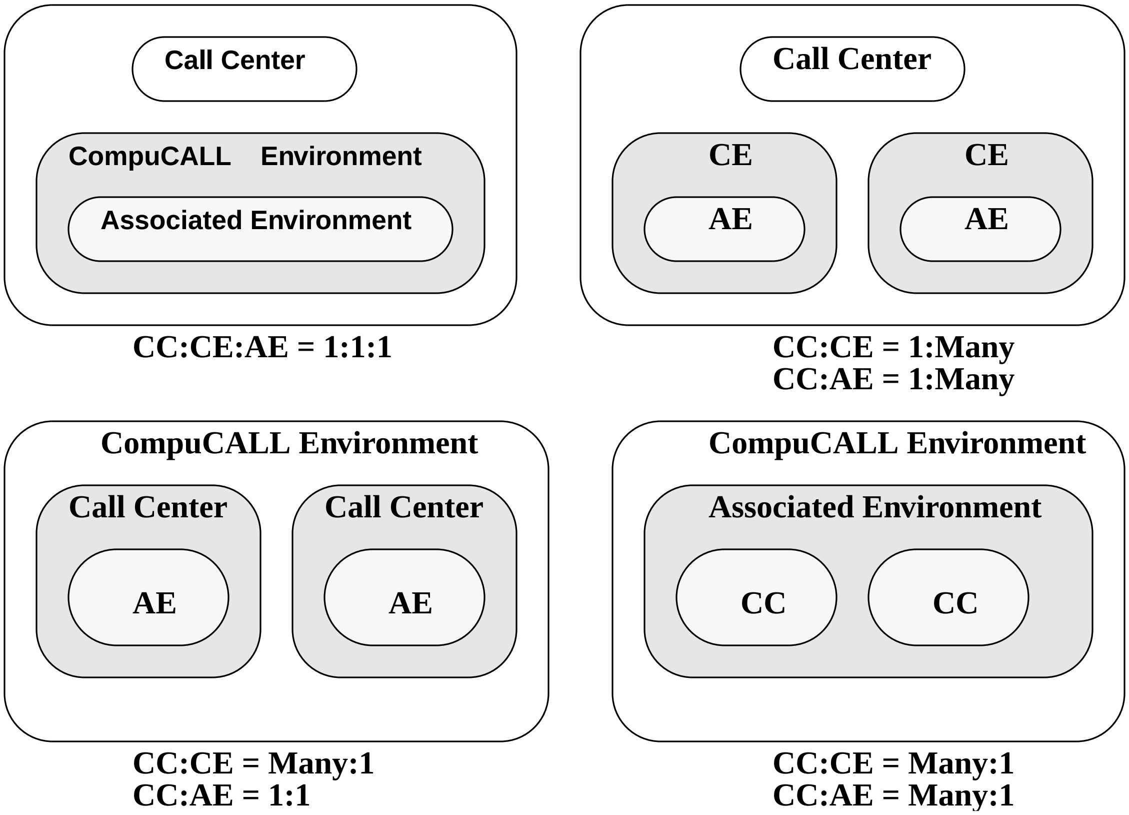
7.3.5.2 Relationship between call centers and associated environments

The Meridian CompuCALL options support situations where:

- there is a one-to-one correspondence between a customer's call center and an associated environment (the call center is served by a single CompuCALL session with which some or all of the ACD groups and/or MDC lines in that call center have been DN-associated)
- a customer's call center includes more than one Associated environment (the call center is served by multiple CompuCALL sessions, with different ACD groups and/or MDC lines in that call center—assigned to the same or different DMS-100 customer Groups—having been DN-associated with each session)
- an Associated environment includes multiple call centers belonging to the same customer (a single CompuCALL session serves multiple call centers whose ACD groups and/or MDC lines are all assigned to the same DMS-100 customer group)

Figure 72 shows examples of the possible relationships among call centers(CC), CompuCALL environments(CE) and associated environments(AE).

Figure 76 Relationship among CC, CE, and AE



CC:CE:AE = 1:1:1

CC:CE = 1:Many
CC:AE = 1:Many

CC:CE = Many:1
CC:AE = 1:1

CC:CE = Many:1
CC:AE = Many:1

Note: CE:AE always 1:1 or 1:Many

7.3.6 Call and CompuCALL call

There are many different definitions of a call. Here call corresponds to the CompuCALL NetworkCallID and refers to a circuit-switched connection involving one or more network addressable entities (Section Figure 73). A call exists when it is initiated (by the originating party going off-hook) and therefore may (temporarily) involve only one party. A call ceases to exist when no party is involved with the call (all parties hang up). The relationship between a call and the parties involved with the call is not fixed. Parties may be added to a call (call transfer) or dropped from a call (dropped from a three-way conference). For example, a call is considered to be the same call even if the original calling and called parties are no longer involved with the call.

A CompuCALL call is a call which is routed to or originated by a CompuCALL party.

A CompuCALL call is significant since:

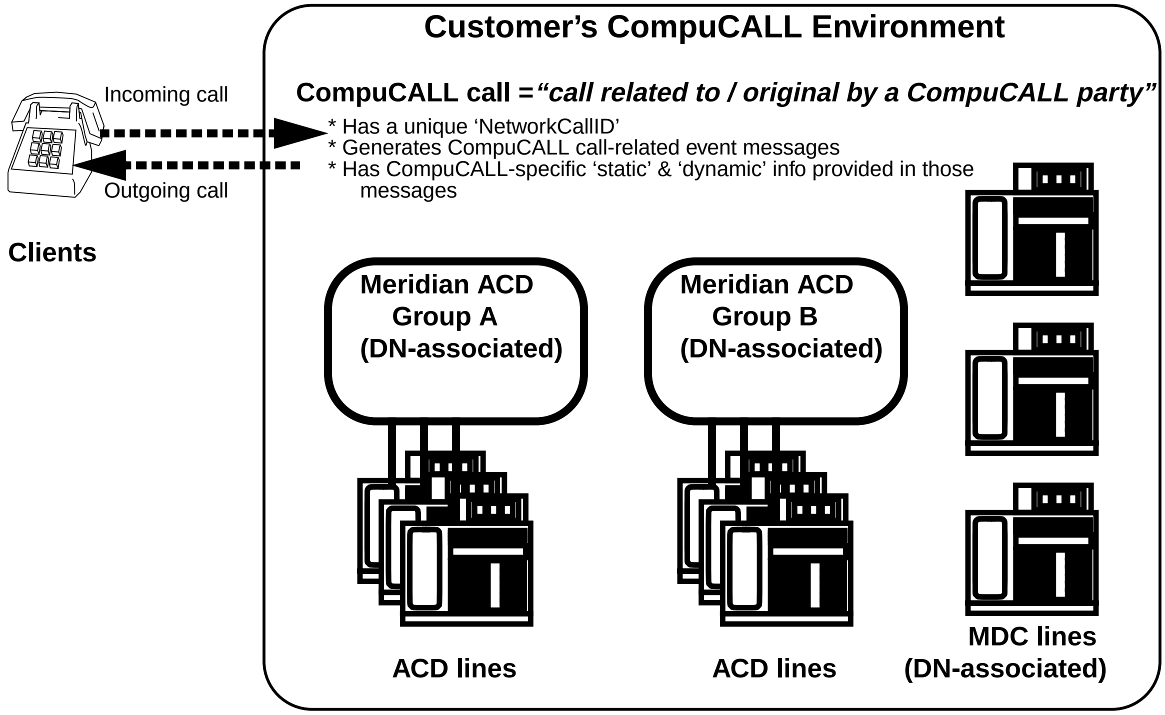
1 It has a NetworkCallID uniquely identifying the call for the customer. It is used for every event and call control message exchanged between the switch and the customer's host computer over any CompuCALL session and related to that call. The NetworkCallID for a CompuCALL call is therefore (like the call) not tied to a particular CompuCALL party.

2 It generates CompuCALL call-related event messages (assuming the customer subscribed to those messages and the CompuCALL party is in one of the customer's associated environments).

3 It has CompuCALL-specific information related to the call in those messages.

The Meridian CompuCALL options are for call center applications where the initial two party call crosses a CompuCALL environment boundary. In other words, they are intended for calls which start out as either incoming calls to ACD groups or MDC lines in a customer's call center from clients outside the call center and not CompuCALL parties, or outgoing calls from ACD lines or MDC lines in a call center to such clients. This is shown in Figure 73.

Figure 77 CompuCALL call


Customer's CompuCALL Environment
CompuCALL call = "call related to / original by a CompuCALL party"
* Has a unique 'NetworkCallID'
* Generates CompuCALL call-related event messages
* Has CompuCALL-specific 'static' & 'dynamic' info provided in those messages
Incoming call
Outgoing call
Clients
Meridian ACD Group A (DN-associated)
Meridian ACD Group B (DN-associated)
ACD lines
ACD lines
MDC lines (DN-associated)

Current release intended for calls which cross CompuCALL Environment boundary

Every parameter in a CompuCALL call-related event message for a CompuCALL call contains one of two types of information:

1 "static information" which is never updated as long as the call remains in the CompuCALL environment. This can be visualized as a "snapshot" of the call that never changes even if the call is transferred, overflowed, forwarded, or redirected within the CompuCALL environment (that is to say between ACD groups, ACD lines, and MDC lines within the CompuCALL environment). Static information is "captured" either:

a. when a call enters a customer's CompuCALL environment, or

b. when a call is originated by an ACD or MDC line in a customer's CompuCALL environment.

An example of static information is the OrigCallingNumber parameter. For a call in "a", this is the calling number for the call when it enters the CompuCALL environment. For a call in "b", it is the DN of the ACD or MDC line (within the CompuCALL environment) which originated the call.

2 "dynamic information" which may be updated while the call remains in the CompuCALL environment. An example is the current DN to which a CompuCALL call is being routed (this will change if the call is transferred, overflowed, forwarded, or redirected within the CompuCALL environment)

7.3.7 Intra-CompuCALL calls

For call center applications of CompuCALL it is assumed that:

- Most CompuCALL calls will involve one and only one CompuCALL party on the same switch (i.e., incoming calls to the customer's call center from "non-CompuCALL party" clients on the same or another switch or outgoing calls from the customer's call center to these clients).
- Some CompuCALL calls will start out as in "a," but become calls where at least two of the parties involved with the call at the same time are CompuCALL parties on the same switch, but where the client (a "non CompuCALL party" on the same or another switch) remains involved with the call until it clears (e.g., calls which are transferred, forwarded, redirected, or overflowed within the customer's call center).
- Very few calls will start out as CompuCALL calls where all of the parties involved with the call are CompuCALL parties on the same switch (i.e., between two parties in the same or different call centers on the same switch).
- Very few calls will become CompuCALL calls where all of the parties involved with the call are CompuCALL parties on the same switch (i.e., the client is no longer involved with the call).

Since the Meridian CompuCALL options are for call center applications, the service functionality meets the major market requirements for call center applications of the product. In other words, the calls in "a" and "b". The current release does not support CompuCALL calls which are at any time entirely "intra-CompuCALL calls" - i.e., CompuCALL calls where all parties involved with the calls are CompuCALL parties within the same or different CompuCALL environments on the same switch.

In other words, the current release is not intended to support calls that are entirely within the same call center, or calls between different call centers on the same switch. Chapter 4.0 Application service options is in the context of call center applications of CompuCALL and does not explicitly address intra-CompuCALL calls. Many of the CompuCALL procedures will apply if customers use CompuCALL for intra-CompuCALL calls. However, these procedures will not necessarily adequately support customer applications of CompuCALL for intra-CompuCALL calls (e.g., maintaining coordination of voice and data for transferred calls). Future offerings of the Meridian CompuCALL options will be enhanced as appropriate, including support of intra-CompuCALL calls if justified by market requirements.

7.4 CompuCALL service design principles

This Section describes the key service design principles governing the service-related procedures for the Meridian CompuCALL options specified in the current release of this document including, where relevant, the exceptions to those principles. The following service design principles are essentially explicit statements of general procedures which are only implicitly apparent from the detailed procedures addressed in Chapter 4.0 Application service options. Detailed procedures which are explicitly addressed in Chapter 4.0 Application service options are not included in this Chapter.

7.4.1 CompuCALL call-related event messages

Principle 1.1 - The CompuCALL call-related event messages dv-Call-Received-C, dv-Call-Queued-U, dv-Call-Offered-U, dv-Call-Answered-U, and dv-Call-Released-U relate to a specific CompuCALL party (i.e., ACD group, ACD line, or MDC line) in a customer's associated environment. They report a change in the status of that CompuCALL party's involvement with a given CompuCALL call, (i.e., an event has occurred).

> ***Note:*** A CompuCALL call is, by definition, a call to/from a CompuCALL party (Section 7.3.5.1).

Principle 1.2 - CompuCALL call-related event messages are sent only over the CompuCALL session with which the specific CompuCALL party to which the messages relate has been "DN-associated" (Section 7.3.4).

> ***Note:*** A given CompuCALL party can be associated with at most one CompuCALL session at any point in time. In other words, CompuCALL

call-related event messages are never "broadcast" over multiple CompuCALL sessions (Please refer to Section 3.2.2, “dv-DN-Associate (operation value "2" hex) message,” on page 94.).

Principle 1.3 - The CompuCALL call-related event messages which are sent for a given CompuCALL party contain no explicit application addressing. In other words, the switch assumes that all messages sent over a given CompuCALL session are received by a single "logical host application" which has established that session.

Note: This does not preclude the customer from having multiple real host applications using the same CompuCALL session. However, in this case the customer must have an Application Program Interface (API) or equivalent which serves the host applications and routes and/or "broadcasts" the event messages received from the switch to these applications as appropriate. To facilitate this, every CompuCALL call-related event message identifies the CompuCALL party (i.e., ACD group or MDC line) for which the message is being sent. The CompuCALL party is identified by the same DN that was in the AssociatedDN parameter in the dv-DN-Associate message which DN-associated that CompuCALL party with that session. This provides a form of implicit message addressing. If an ACD line, the primary ACD DN of the ACD group to which the line is assigned is provided (as well as the identity of the specific ACD line).

Principle 1.4 - At most, one CompuCALL call-related event message of each type (e.g., dv-Call-Offered-U) is sent for a given CompuCALL call with respect to a CompuCALL party involved in the call.

Note 1: A call may leave a given CompuCALL party and the return to the same CompuCALL party. In this case the same message may be sent more than once. For
example, a dv-Call-Queued-U message is sent when a call is placed in the Incoming calls Queue of an ACD group. If the call is subsequently re-queued for the same ACD group after an agent has been selected to handle the call (by the Ring Threshold feature), another dv-Call-Queued-U message for the call will be sent. However, in this case, the call can be considered to have undergone two distinct events - first to have logically left the ACD group and second to have been routed back to the same ACD group.

Note 2: The same message type for a given call may also be sent more than once for different CompuCALL parties involved with the call (e.g., call transfer).

Principle 1.5 - The CompuCALL call-related event messages pertaining to a given CompuCALL party for a given call are sequential. They are always sent in the same order for a given call routed to a given CompuCALL party. For example, the messages sent for a call which is routed to an ACD group, queued for an agent, handled by an agent, and then cleared would be dv-Call-Received-C or dv-Call-Queued-U (depending on whether or not the call is subject to call redirection), dv-Call-Offered-U, dv-Call-Answered-U, and finally dv-Call-Released-U.

> ***Note:*** Not all possible messages may be sent. The actual CompuCALL call-related event messages sent over a given CompuCALL session for a given CompuCALL party are determined by the ServiceID specified when the session was established (see Chapter 3, Section 1.0).

Principle 1.6 - The CompuCALL call-related event messages dv-Call-Queued-U, dv-Call-Offered-U, dv-Call-Answered-U, and dv-Call-Released-U are unconfirmed (i.e., "-U" = "unconfirmed"). In other words, no positive acknowledgment (RETURN-RESULT) or negative acknowledgment (RETURN-ERROR) is expected when a message is sent and if sent back will be responded to with a REJECT message with reason "unrecognized invocation". These messages are therefore all of ROSE Operation Class 5 (Section 1.5, “Application layer designer guide,” on page 43). Therefore, sending any of these event messages has no impact on the progress of the call within the switch.

> ***Note:*** A positive or negative acknowledgment is required for the dv-Call-Received-C message (i.e., "C" = "confirmed"). However, this message is for call control purposes (call redirection) as well as event reporting. This message is of ROSE Operation Class 2.

7.4.2 Parameters in CompuCALL call-related event messages

Principle 2.1 - The parameters provided in CompuCALL call-related event messages for a given CompuCALL party are dependent only on the ServiceID for the session over which those messages are sent - i.e., which parameters the customer has subscribed to and wishes to receive over that session. They are independent of whether or not any previous event messages for the same call have been sent for the same or any other CompuCALL parties involved with the call.

Exception - The presence and contents of the CallHistoryInfo parameter do depend on whether or not a dv-Call-Received-C message relating to the call has been sent previously. If this message has been sent and the call has been redirected, then the CallHistoryInfo parameter is present (i.e., the call has been "extended") and may contain the HostCallData subparameter.

Principle 2.2 - The static information for a CompuCALL call (NetworkCallID) is retained unchanged as long as the call is in a customer's

CompuCALL environment, regardless of whether and how many times the call is extended in that environment. "Extended" refers to transferred, overflowed, forwarded, or redirected calls.

> ***Note:*** The call does not have to remain in an Associated environment in the CompuCALL environment (Section 7.3.5.1).

Principle 2.3 - There is only one view of the static information for a CompuCALL call during its lifetime since it relates to the call and not to a given CompuCALL party involved with the call. In other words, parameters such as the NetworkCallID and OrigCallingNumber included in any CompuCALL call-related event message for a CompuCALL call contain the same values as long as the call remains in the same CompuCALL environment.

Principle 2.4 - There may be more than one view of the dynamic information for a CompuCALL call during its lifetime since it may relate to a given CompuCALL party's involvement with a call, not to the overall call. For example, the DeviceID indicates a specific line (ACD or MDC). In the case of a call which involves two CompuCALL parties and which is cleared, a dv-Call-Released-U message containing a different DeviceID will be sent for each CompuCALL party.

Principle 2.5 - A call becomes a CompuCALL call when it enters a CompuCALL environment (i.e., when it is routed to a CompuCALL party in that CompuCALL environment). At that time the call is assigned a new NetworkCallID and the static information related to the call is created. Also, a call is a CompuCALL call when it is originated by a CompuCALL party - either by an ACD line using dv-Make-Call or by an MDC line using the telephone set.

Exception 1 - A CompuCALL call which is "extended" by a CompuCALL party to another CompuCALL party in a different CompuCALL environment on the same switch is treated the same as a call "extended" in the same CompuCALL environment - i.e., the existing static information related to the call (NetworkCallID) is retained and is not updated. This allows the coordination of voice and data to be retained even when calls are "extended" between CompuCALL parties assigned to different DMS-100 customer groups, and therefore provides CompuCALL customers with the maximum flexibility in configuring their call centers.

Exception 2 - A call which leaves a CompuCALL environment and is "extended" back into the same CompuCALL environment without leaving the switch may not be treated as a new CompuCALL call. It may be treated as though it had been "extended" in the same CompuCALL environment. In other words, the event messages sent after the call reenters the CompuCALL environment may contain the previous static information (NetworkCallID, OrigCallingNumber) and will include the CallHistoryInfo parameter.

Principle 2.6 - The CallHistoryInfo parameter is included in CompuCALL call-related event messages for a CompuCALL party only when the call has been extended to that CompuCALL party from another CompuCALL party on the same switch. The CallHistoryInfo parameter includes subparameters CallType, OrigInboundDN, PrevApplicationID, and HostCallData.

> ***Note 1:*** A CompuCALL call forwarded by CompuCALL party A to CompuCALL party B in the same or another CompuCALL environment on the same switch using either the Call Forward Universal (CFU) or Call Forward Busy (CFB) feature is treated by CompuCALL as if it had come directly from the forwarded party to party B. In other words, the CompuCALL call-related event messages sent for party B will not contain the CallHistoryInfo parameter (unless the call to party A already has CallHistoryInfo). However, that if party A uses the Call Forward Don't Answer (CFD) feature, the call is treated as "extended" from party A to party B and the messages sent for party B will contain the CallHistoryInfo parameter.

> ***Note 2:*** A CompuCALL call routed to an ACD group which is a CompuCALL party and then handled by an ACD line assigned to that group (also a CompuCALL party) as part of normal ACD call handling, the call is not considered to have left the ACD group (which would generate a dv-Call-Released-U message) and been "extended" to an ACD line (which would generate a CallHistoryInfo parameter).

Exception - The CallHistoryInfo parameter may be included for calls leaving a CompuCALL environment and subsequently "extended" into the same or another CompuCALL environment by a non-CompuCALL party if the call stays in the same switch.

7.4.3 CompuCALL call control messages

Principle 3.1 - The scope of the Third Party Call Control (TPCC) messages for ACD lines dv-Make-Call, dv-Add-Party, dv-Transfer-Party, dv-Conference-Party, and dv-Drop-Party is the CompuCALL environment and not the Associated environment. A TPCC message must be sent over a CompuCALL session which was established using the BusinessGroupID which corresponds to the DMS-100 customer group to which the ACD line for which the message is being sent has been assigned. A TPCC message for a given ACD line is not restricted to being sent over the session with which the ACD group to which the ACD line is assigned has been DN-associated. In fact, the ACD group to which the ACD line is assigned does not need to be DN-associated with any active session.

> ***Note 1:*** An ACD line must be logged in to an ACD group in order for these TPCC messages to be used for that line.

> ***Note 2:*** It is the customer's responsibility to coordinate the various host applications which generate TPCC messages for a given ACD line, whether they use the same session or multiple sessions.
>
> ***Note 3:*** These TPCC messages are not supported for MDC lines in the current release of this document.

Principle 3.2 - The CompuCALL call redirection message dv-Call-Redirect must be sent by the host computer to the Aasociated environment to which the ACD group which has received the call belongs (i.e., over the same CompuCALL session on which the corresponding dv-Call-Received-C message was received). Note that dv-Call-Redirect is sent on behalf of an ACD group and not an ACD line.

Principle 3.3 - The switch always retains responsibility for CompuCALL calls, even though the customer's host computer may use CompuCALL to control these calls. For example:

- The CompuCALL call redirection capability gives the customer's host computer the opportunity to redirect a CompuCALL call which has been routed to a given ACD group, using the dv-Call-Redirect message which is sent in response to the dv-Call-Received-C message. A response from the host computer is not required in order for the switch to continue routing the call. If a RETURN-RESULT or dv-Call-Redirect message is not received by the switch within a customer defined time period from when the dv-Call-Received-C message was sent, the switch will continue to route the call to the ACD group for which the dv-Call-Received-C message was sent.
- When the switch receives a dv-Call-Redirect message, it verifies that the RedirectDestination parameter corresponds to a valid translation before redirecting the call (i.e., a line, trunk, or treatment on the same switch). However, the switch does not verify that the destination is idle (e.g., could be a busy line).
- When the switch receives two conflicting TPCC messages at the same time (e.g., two dv-Make-Call messages for the same ACD line), it arbitrates. The first message processed by the switch is accepted, and the second one is responded to with a RETURN-ERROR message (e.g., "Not-Idle").

Principle 3.4 - The CompuCALL TPCC messages for ACD lines complement (not replace) the equivalent existing telephone set features. For example, the same ACD agent can transfer calls by using at any time either the telephone set or the CompuCALL dv-Add-Party, dv-Transfer-Party, dv-Conference-Party, and dv-Drop-Party messages. However, this must be performed on a call-by-call basis - i.e., a call transfer which is initiated using the phone cannot be completed using CompuCALL and vice versa.

8.0 Appendix List of service functions and parameters

This appendix provides a listing of all the services, functions and parameters used in this document.

8.1 CompuCALL service options

The following are the CompuCALL service options covered in this release:

- CompuCALL OA&M
- CompuCALL session management
- Meridian ACD CompuCALL
- MDC CompuCALL

8.1.1 CompuCALL functions and capabilities by service

In this section, all the application service functions used by each CompuCALL service are listed under the application service options. For Meridian ACD CompuCALL options and MDC CompuCALL options, the application service functions are categorized according to each application service capability.

CompuCALL session management options:

- dv-APPL-Logoff
- dv-APPL-Logon
- dv-DN-Associate

CompuCALL OA&M options:
dv-Appl-Continuity-Test

Meridian ACD CompuCALL options:

- Coordinated voice and data:
  - dv-Call-Answered-U
  - dv-Call-Offered-U
  - dv-Call-Queued-U
  - dv-Call-Released-U

  - dv-Set-Offhook-U
- Third Party Call Control:
  - dv-Answer-Call
  - dv-Hold-Call
  - dv-Release-Call
  - dv-Unhold-Call
  - dv-Call-Unheld-U
  - dv-Call-Consult-Originated-U
  - dv-Conferenced-U
  - dv-Call-Transferred-U
  - dv-Add-Party
  - dv-Conference-Party
  - dv-Drop-Party
  - dv-Make-Call
  - dv-Transfer-Party
- Call routing
  - dv-Call-Received-C
  - dv-Call-Redirect
- Resource status
  - dv-Resource-Query
- Third Party Agent Control:
  - dv-Set-Feature
  - dv-Agent-Logged-In-U
  - dv-Agent-Logged-Out-U
  - dv-Agent-Ready-U
  - dv-Agent-Not-Ready-U
  - dv-Agent-SetAction-U
- Third Party Queue Control:
  - dv-Route-Call
  - dv-Give-Treatment
  - dv-Treatment-Complete-U
  - dv-Set-CDN-State

MDC CompuCALL options:

- Coordinated voice and data:
  - dv-Call-Answered-U
  - dv-Call-Offered-U
  - dv-Call-Released-U
  - dv-Set-Offhook-U
- Message waiting notification:
  - dv-Message-Waiting-U
- Call initiation:
  - dv-Make-Call
- SCAI 3WC and CXR:
  - dv-Add-Party
  - dv-Transfer-Party
  - dv-Drop-Party
  - dv-Conference-Party
  - dv-Call-Consult-Originated-U
  - dv-Call-Conferenced-U
  - dv-Call-Transferred-U
- SCAI call control:
  - dv-Answer-Call
  - dv-Release-Call
  - dv-Hold-Call
  - dv-Unhold-Call
  - dv-Call-Unheld-U
- DN-Query:
  - dv-DN-Query

8.2 CompuCALL functions and their operation values

In this section all the functions used in CompuCALL are listed in alphabetical order and their operation values (in decimal) are provided.

• dv-Add-Party	12
• dv-Agent-Logged-In-U	18
• dv-Agent-Logged-Out-U	20

dv-Agent-Ready-U	21
dv-Agent-Not-Ready-U	22
dv-Agent-SetAction-U	44
dv-Answer-Call	23
dv-Appl-Continuity-Test	03
dv-Appl-Logoff	04
dv-Appl-Logon	01
dv-Call-Answered-U	16
dv-Call-Conferenced-U	32
dv-Call-Consult-Originated-U	31
dv-Call-Offered-U	05
dv-Call-Queued-U	06
dv-Call-Received-C	10
dv-Call-Redirect	11
dv-Call-Released-U	07
dv-CDN-State	37
dv-Give-Treatment	35
dv-Route-Call	34
dv-Treatment-Complete-U	36
dv-Call-Transferred-U	33
dv-Call-Unheld-U	30
dv-Conference-Party	15
dv-DN-Associate	02
dv-Drop-Party	14
dv-EMK-U	43
dv-Hold-Call	28
dv-LOB-Event-U	42
dv-Make-Call	09
dv-Release-Call	24
dv-Resource-Query	19
dv-Set-Feature	17
dv-Set-Offhook	25

• dv-Transfer-Party	13
• dv-Unhold-Call	29

8.2.1CompuCALL message change history

This section lists the name changes, introductions, or removals of CompuCALL messages in the current software release. It is listed below in alphabetical order with references to the documentation release of the occurrence.

- I = Implemented
- U = Unchanged
- E = Enhanced
- NA = Not Available CompuCALL Parameters by Function

CompuCALL Message	**CompuCALL Interface Specification Release**							
	01.01			02.01	03.01	03.02	04.01	05.01
CCM	**30**	**31**	**32**	**33**	**34**	**34**	**35**	**36/01**
continuity Return Result	I	U	U	E	U	U	U	U
continuity Return Error	I	U	U	U	U	U	U	U
dv-Add-Party				I	E	U	U	U
dv-Appl-Logoff	I	U	U	U	U	U	U	U
dv-Appl-Logoff-U	I	U	U	NA	NA	NA	NA	NA
dv-Appl-Logon	I	U	E	U	U	U	U	U
dv-DN-Associate	I	U	U	U	E	U	E	U
dv-Call-Answered-U					I	U	E	U
dv-Call-Offered-U	I	E	E	U	E	U	E	U
dv-Call-Received-U			I	U	E	U	E	U
dv-Call-Redirect			I	U	E	U	U	U
dv-Call-Queued-U		I	E	U	E	U	E	U
dv-Call-Released-U	I	E	E	U	E	U	U	U
dv-Conference-Party					I	U	U	U
dv-Drop-Party				I	U	U	U	U
dv-Make-Call			I	U	E	U	U	U
dv-Resource-Query						U	I	U

CompuCALL Message	CompuCALL Interface Specification Release							
	01.01			02.01	03.01	03.02	04.01	05.01
dv-Transfer-Party				I	U	U	U	U
Received-C Return Error			I	U	U	U	U	U
Received-C Reject			I	U	U	U	U	U
Received-C Return Result			I	U	U	U	U	U
Reject	I	U	U	U	U	U	U	U
Res.-Query Return Result							I	U
dv-Set-Feature								I
Set-Feature Return Result								I
Set-Feature Return Error								I
dv-Agent-Logged-In-U								I
dv-Agent-Logged-Out-U								I
dv-Agent-Ready-U								I
dv-Agent-Not-Ready-U								I

CompuCALL Message	CompuCALL Interface Specification Release		
	6.0	**6.0**	**6.0**
CCM	05	06	07
continuity Return Result	U	U	U
continuity Return Error	U	U	U
dv-Add-Party	E	U	U
dv-Appl-Continuity-Test	U	U	U
dv-Appl-Logoff	U	U	U
dv-Appl-Logoff-U	NA	NA	NA
dv-Appl-Logon	U	U	U
dv-DN-Associate	E	U	U
dv-Call-Answered-U	E	U	U
dv-Call-Offered-U	E	U	U
dv-Call-Received-C	U	U	U

CompuCALL Message	CompuCALL Interface Specification Release		
	6.0	6.0	6.0
CCM	05	06	07
dv-Call-Redirect	U	U	U
dv-Call-Queued-U	U	U	U
dv-Call-Released-U	E	U	U
dv-Conference-Party	E	U	U
dv-Drop-Party	E	U	U
dv-Make-Call	E	U	E
dv-Resource-Query	U	U	U
dv-Transfer-Party	E	U	U
Received-C Return Error	U	U	U
Received-C Reject	U	U	U
Received-C Return Result	U	U	U
Reject	U	U	U
Res.-Query Return Result	U	U	U
dv-Set-Feature	U	U	U
Set-Feature Return Result	U	U	U
Set-Feature Return Error	U	U	U
dv-Agent-Logged-In-U	U	U	U
dv-Agent-Logged-Out-U	U	U	U
dv-Agent-Ready-U	U	U	U
dv-Agent-Not-Ready-U	U	U	U
dv-Set-OffHook-U	I	U	U
dv-Message-Waiting-Indication	I	U	U
dv-DN-Query	I	U	U
dv-Answer-Call		I	U
dv-Hold-Call		I	U
dv-UnHold-Call		I	U

CompuCALL Message	CompuCALL Interface Specification Release		
	6.0	**6.0**	**6.0**
CCM	05	06	07
dv-Call-Unheld-U		I	U
dv-Release-Call		I	U
dv-Call-Consult-Originated-U			I
dv-Call-Conferenced-U			I
dv-Call-Transferred-U			I

In this section, all the parameters used in a function are listed between parentheses following the function. Some parameters called constructors have nested parameters which are listed between parentheses. Some functions have an associated response from the recipient. These are indicated as RETURN-RESULT with parameters shown in parentheses. All the parameters may not be included or required in these messages. These messages are listed in alphabetical order.

- dv-Add-Party (AddPartyType, OrigAddress, DestAddress) RETURN-RESULT (NetworkCallID)
- dv-Agent-Logged-In-U (ACDGroup, PosID, LoginID)
- dv-Agent-Logged-Out-U (ACDGroup, PosID)
- dv-Agent-Ready-U (PosID, PrivateData (ACDGroup))
- dv-Agent-SetAction-U (AgtPosID, NtwSetAct)
- dv-Agent-Not-Ready-U (PosID, PrivateData (ACDGroup, WalkawayReason))
- dv-Appl-Continuity-Test RETURN-RESULT (ApplicationID)
- dv-Appl-Logoff
- dv-Appl-Logon (ApplicationID, BusinessGroupID,NetworkNodeID, Password, ServiceID, ServiceVersion)
- dv-Call-Answered-U (NetworkCallID, DeviceID, ACDDN, OrigCallingNumber, OrigChargeNumber, CallHistoryInfo (CallType, OrigInboundDN, PrevApplicationID, HostCallData), ACDGroup, CallMode, ForwardingParty1)
- dv-Call-Conferenced-U (NetworkCallID, ContollerDN, ConsultDN, FirstLegDN)
- dv-Call-Consult-Originated-U (NetworkCallID, DeviceID, ConsultDN)

- dv-Call-Offered-U (NetworkCallID, ACDDN, OrigDeviceID, OrigChargeNumber, CallHistoryInfo (CallType, OrigInboundDN, PrevApplicationID, HostCallData), ACDGroup, CallMode, destDeviceID, Forwarding Party, dialedDigits, stationNumber, name, PositionID1)
- dv-Call-Transferred-U (NetworkCallID, DeviceID, OtherPtyDN)
- dv-Call-Queued-U (NetworkCallID, ACDDN, OrigCallingNumber, OrigChargeNumber, CallHistoryInfo(CallType, OrigInboundDN, PrevApplicationID, HostCallData), ACDGroup, ForwardingParty1)
- dv-Call-Received-C (NetworkCallID, ACDDN, OrigCallingNumber, OrigChargeNumber, CallHistoryInfo(CallType, OrigInboundDN, PrevApplicationID, HostCallData), ACDGroup, ForwardingParty1)
- RETURN-RESULT
- dv-Call-Redirect (NetworkCallID, RedirectDestination, HostCallData)
- dv-Call-Released-U (NetworkCallID, ReleaseReason, ACDDN, ACDGroup, DeviceID)
- dv-CDN-State (CDN, state)
- dv-Conference-Party (OrigAddress)
  RETURN-RESULT (NetworkCallID)
- dv-DN-Associate (AssociatedDN, DNOperation)
  RETURN-RESULT
- dv-Drop-Party (DropPartyType, OrigAddress)
  RETURN-RESULT (NetworkCallID)
- dv-EMK-U (NetworkCallID, AgtPosID, AgtDN, SuprvPID, SuprvDN, AuxDN, EMKInfo)
- dv-Give-Treatment (NetworkCallID, ACDDN, treatmentType)
- dv-LOB-Event-U (NetworkCallID, AgentPosID, LOBCode, LOBTime)
- dv-Make-Call (OrigAddress, DestAddress, ApplicationData (MakeCallType, AuthCodeDigits, AcctCodeDigits))
  RETURN-RESULT (NetworkCallID)
- dv-Resource-Query (QueryAddress)
  RETURN-RESULT (Time, QueryAddress, GrpPrimDN, GrpStat,QueryInfo)
- dv-Route-Call (NetworkCallID, ACDDN, destination)

  RETURN-RESULT (networkCallID)
- dv-Set-Feature (OrigAddress, Set-Feature-Info (AgentFeature), PrivateData (AgentPrivateData))
  RETURN-RESULT

- dv-Transfer-Party (OrigAddress)
  RETURN-RESULT (NetworkCallID)
- dv-Treatment-Complete-U (NetworkCallID, ACDDN, TreatmentCompleted)

  RETURN-RESULT (TRTRAN, TRTMUS, TRTRNG, TRTSIL, TRTDSC, TRTBUS, TRTFBY)

9.0 Acronyms

3WC	Three Way Call
ACD	Automatic Call Distribution
ACDDN	Automatic Call Distribution Directory Number
ACSE	Association Control Service Element
AE	Application Entity
AEMK	Answer Emergency Key
AIN	Advanced Intelligent Network
ANSI	American National Standards Institute
AP	Application Process
API	Application Program Interface
ASE	Application Service Element
ASN	Abstract Syntax Notation
ASN.1	Abstract Syntax Notation 1
BCS	Batch Change Supplement
BGID	Business Group Identifier
BRI	Basic Rate Interface
CAS	Computer Aided Signalling
CCITT	Consultative Committee on International Telegraph and Telephone
CDN	Controlled Directory Number

CIF	Controlled Interflow
CLASS	Customer Local Area Signalling Services
CLID	Calling Line Identification (ID)
CNA	Canadian
CNAB	Calling Name Delivery Blocking
CNDB	Calling Number Delivery Blocking
CFB	Call Forward Busy
CFD	Call Forward Don’t Answer
CFI	Call Forward Intragroup
CFU	Call Forward Universal
CPU	Call Pickup
CTX	Meridian Digital Centrex (same as Centrex)
CVD	Coordinated Voice and Data
CWT	Call Waiting
CXR	Call Transfer
DCE	Data Communications Equipment
DDN	Dial Number Delivery
DMS	Digital Multiplex System
DN	Directory Number
DNA	Data Network Address
DSP	Down-Stream Processor
DTE	Data Terminal Equipment
DV	Data Voice
ECM	Extended Call Management
EMK	(ACD) Emergency Key

EBS	Electronic Business Set
EMPC	Enhanced Multi-Protocol Card
ESDN	Enhanced Secondary DN
ICCM	Integrated Call Center Manager
ICM	Intelligent Call Manager
ID	Identifier
IOC	DMS I/O Controller
IOD	Input/Output Device
ISDN	Integrated Service Digital Network
ISUP	ISDN User Part
IVR	Interactive Voice Response Unit
LCS	Local Concrete Syntax
LOB	Line of Business
LDN	Listed Directory Number
LSB	Least Significant Bit
MACD	Meridian ACD
MADN	Multiple Appearance Directory Number
MAP	Maintenance and Administrative Position
MBS	Meridian Business Set
MBG	Multi Business Group
MDC	Meridian Digital Centrex
MFT	Meridian Feature Transparency
MM	Man Machine interface
MPC	Multi-Protocol Card
MSB	Most Significant Bit

MWT	Message Waiting
NCOS	Network Class of Service
NA	North America
NANP	North American Numbering Plan
NICM	Network Intelligent Call Manager
NT	Northern Telecom
OM	Operations Measurement
OSI	Open System Interconnection
POSID	Position ID
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephone Network
RCHD	Residential Call Hold
RE	Return Error
REJ	Reject
RES	Residential Line
RLS	Release
RO	Remote Operation
ROSE	Remote Operation Service Element
RR	Return Result
SCA	Selective Call Acceptance
SACF	Single Association Control Function
SCAI	Switch-Computer Applications Interface (T1S1 SCAI)
SCRJ	Selective Call Rejection
SDN	Secondary Directory Number
SO	Service Orders

SMDI	Simplified Message Desk Interface
SMDR	Station Message Detailed Recording
SUS	Suspended Service
SVC	Switched Virtual Circuit
TCAP	Transaction Capabilities Application Part
TPAC	Third Party Agent Control
TPCC	Third Party Call Control
TPQC	Third Party Queue Control
Telco	Telephone Company -- Switch Service Provider
UCD	Universal Call Distribution

10.0 Bibliography

Information Processing Systems - Open Systems Interconnection -Application Layer Structure- ISO 9545 (defines relationships among AE, ASE, and AP)

Reference Model of Open Systems Interconnection for CCITT Applications - CCITT Recommendation X.200 (ISO 7498).
Use of X.200 Series Protocols in CCITT Modifications - CCITT Recommendation X.220.

Information Processing Systems - Text Communication -
Remote Operations Part 1: Model Notation, and Service Definition -CCITT Recommendation X.219 (ISO 9072-1)
Remote Operations Part 2: Protocol Specification -CCITT Recommendation X.229 (ISO 9072-2)

Information Processing Systems - Open Systems Interconnection -
Specification of Abstract Syntax Notation One (ASN.1) -CCITT Recommendation X.208 (ISO 8824)
Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1) - CCITT Recommendation X.209 (ISO 8825)
ISDN Basic Rate User-Network Interface Specification NIS S208-5, Issue 1.0

Nortel Networks

International Standard ISO 7776: Information processing systems - Data communication - High-level data link control procedures - Description of the X.25 LAPB-compatible DTE data link procedures.
International Standard ISO 8208: Data Communication - X.25 packet level protocol for data terminal equipment.

CompuCALL / Meridian SCAI

Interface Specification

NORTEL NETWORKS CONFIDENTIAL: The information contained in this document is the property of Nortel Networks. Except as specifically authorized in writing by Nortel Networks, the holder of this document shall keep the information contained herein confidential and shall protect same in whole or in part from disclosure and dissemination to third parties and use same for evaluation, operation, and maintenance purposes only.

Information subject to change without notice
DMS, Meridian and CompuCALL are trademarks of Nortel Networks

Document number: NIS-Q218-1
Document version: 10.01
Document status: Standard
Date: March 2000
Printed in United States of America

NORTEL NETWORKS™

How the world shares ideas.