Network Working Group
Request for Comments: 2106
Category: Informational

S. Chiang
J. Lee
Cisco Systems, Inc.
H. Yasuda
Mitsubishi Electric Corp.

February 1997

Data Link Switching Remote Access Protocol

Status of this Memo

Abstract

This memo describes the Data Link Switching Remote Access Protocol that is used between workstations and routers to transport SNA/ NetBIOS traffic over TCP sessions. Any questions or comments should be sent to drap@cisco.com.

1. Introduction

Since the Data Link Switching Protocol, RFC 1795, was published, some software vendors have begun implementing DLSw on workstations. The implementation of DLSw on a large number of workstations raises several important issues that must be addressed. Scalability is the major concern. For example, the number of TCP sessions to the DLSw router increases in direct proportion to the number of workstations added. Another concern is efficiency. Since DLSw is a switch-to-switch protocol, it is not efficient when implemented on workstations.

DRAP addresses the above issues. It introduces a hierarchical structure to resolve the scalability problems. All workstations are clients to the router (server) rather than peers to the router. This creates a client/server model. It also provides a more efficient protocol between the workstation (client) and the router (server).

2. Overview

2.1. DRAP Client/Server Model

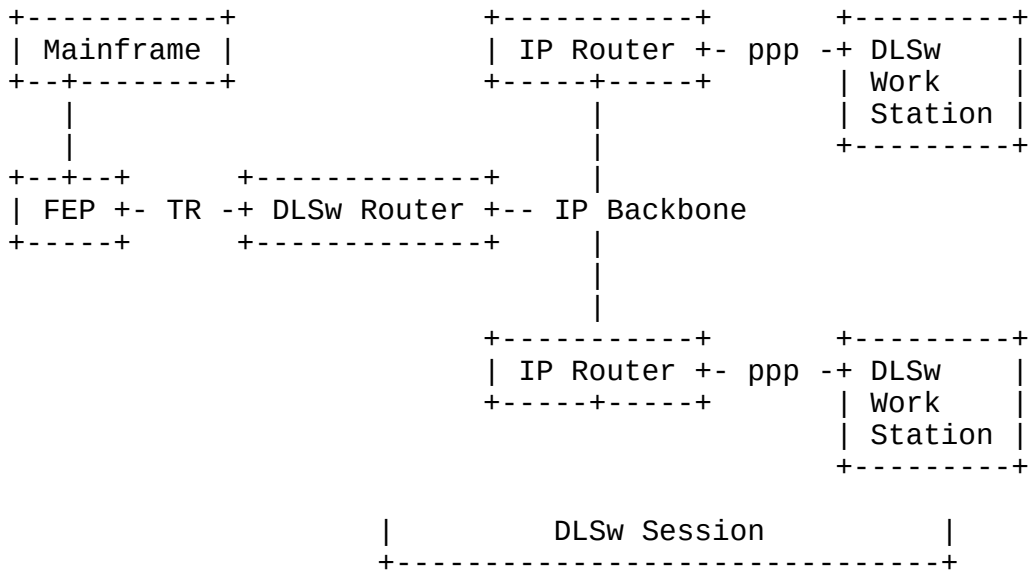

Figure 2-1. Running DLSw on a large number of workstations creates a scalability problem.

Figure 2-1 shows a typical DLSw implementation on a workstation. The workstations are connected to the central site DLSw router over the IP network. As the network grows, scalability will become an issue as the number of TCP sessions increases due to the growing number of workstations.

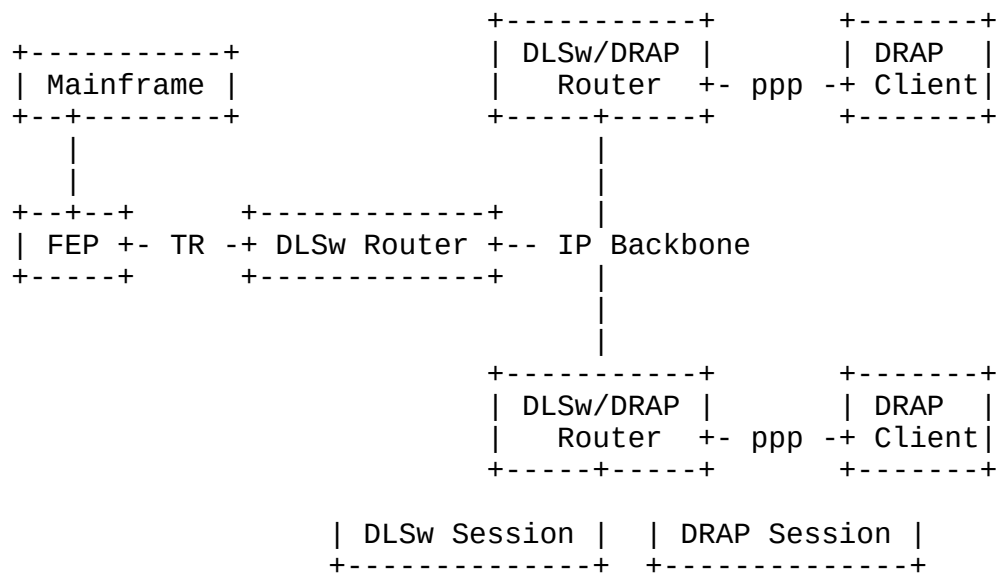

Figure 2-2. DLSw Remote Access Protocol solves the scalability problem.

In a large network, DRAP addresses the scalability problem by significantly reducing the number of peers that connect to the central site router. The workstations (DRAP client) and the router (DRAP server) behave in a Client/Server relationship. Workstations are attached to a DRAP server. A DRAP server has a single peer connection to the central site router.

2.2. Dynamic Address Resolution

In a DLSw network, each workstation needs a MAC address to communicate with a FEP attached to a LAN. When DLSw is implemented on a workstation, it does not always have a MAC address defined. For example, when a workstation connects to a router through a modem via PPP, it only consists of an IP address. In this case, the user must define a virtual MAC address. This is administratively intensive since each workstation must have an unique MAC address.

DRAP uses the Dynamic Address Resolution protocol to solve this problem. The Dynamic Address Resolution protocol permits the server to dynamically assign a MAC address to a client without complex configuration.

For a client to initiate a session to a server, the workstation sends a direct request to the server. The request contains the destination MAC address and the destination SAP. The workstation can either specify its own MAC address, or request the server to assign one to it. The server’s IP address must be pre-configured on the workstation. If IP addresses are configured for multiple servers at a

workstation, the request can be sent to these servers and the first one to respond will be used.

For a server to initiate a session to a client, the server sends a directed request to the workstation. The workstation must pre-register its MAC address at the server. This can be done either by configuration on the server or registration at the server (both MAC addresses and IP addresses will be registered).

2.3. TCP Connection

The transport used between the client and the server is TCP. Before a TCP session is established between the client and the server, no message can be sent. The default parameters associated with the TCP connections between the client and the server are as follows:

Socket Family	AF_INET	(Internet protocols)
Socket Type	SOCK_STREAM	(stream socket)
Port Number	1973	

There is only one TCP connection between the client and the server. It is used for both read and write operations.

3. DRAP Format

3.1. General Frame Format

The General format of the DRAP frame is as follows:

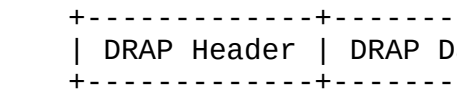

Figure 3-1. DRAP Frame Format

The DRAP protocol is contained in the DRAP header, which is common to all frames passed between the DRAP client and the server. This header is 4 bytes long. The next section will explain the details.

The next part is the DRAP Data. The structure and the size are based on the type of messages carried in the DRAP frame. The DRAP data is used to process the frame, but it is optional.

The third part of the frame is the user data, which is sent by the local system to the remote system. The size of this block is variable and is included in the frame only when there is data to be sent to the remote system.

3.2. Header Format

The DRAP header is used to identify the message type and the length of the frame. This is a general purpose header used for each frame that is passed between the DRAP server and the client. More information is needed for frames like CAN_U_REACH and I_CAN_REACH, therefore, it is passed to the peer as DRAP data. The structure of the DRAP data depends on the type of frames, and will be discussed in detail in later sections.

The DRAP Header is given below:

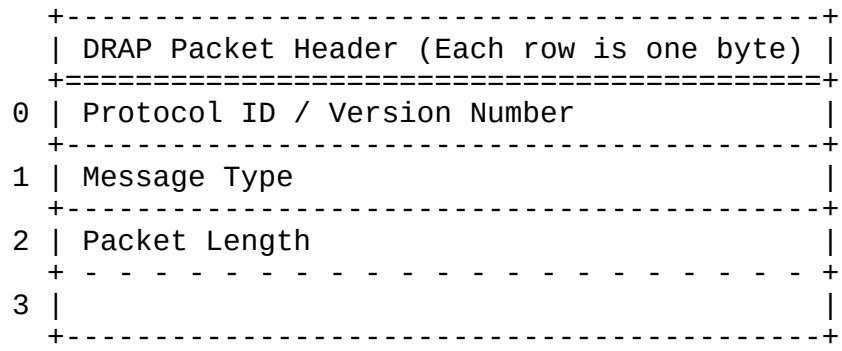

Figure 3-2. DRAP Header Format

- The Protocol ID uses the first 4 bits of this field and is set to "1000".
- The Version Number uses the next 4 bits in this field and is set to "0001".
- The message type is the DRAP message type.
- The Total Packet length is the length of the packet including the DRAP header, DRAP data and User Data. The minimum size of the packet is 4, which is the length of the header.

3.3. DRAP Messages

Most of the Drap frames are based on the existing DLSw frames and have the same names. The information in the corresponding DRAP and DLSw frames may differ; but the functionalities are the same. Thus the DLSw State Machine is used to handle these DRAP frames. Some new DRAP frames were created to handle special DRAP functions. For example, the new DRAP frames, I_CANNOT_REACH and START_DL_FAILED provide negative acknowledgment. The DLSw frames not needed for DRAP, are dropped.

The following table lists and describes all available DRAP messages:

DRAP Frame Name	Code	Function
CAN_U_REACH	0x01	Find if the station given is reachable
I_CAN_REACH	0x02	Positive response to CAN_U_REACH
I_CANNOT_REACH	0x03	Negative response to CAN_U_REACH
START_DL	0x04	Setup session for given addresses
DL_STARTED	0x05	Session Started
START_DL_FAILED	0x06	Session Start failed
XID_FRAME	0x07	XID Frame
CONTACT_STN	0x08	Contact destination to establish SABME
STN_CONTACTED	0x09	Station contacted - SABME mode set
DATA_FRAME	0x0A	Connectionless Data Frame for a link
INFO_FRAME	0x0B	Connection oriented I-Frame
HALT_DL	0x0C	Halt Data Link session
HALT_DL_NOACK	0x0D	Halt Data Link session without ack
DL_HALTED	0x0E	Session Halted
FCM_FRAME	0x0F	Data Link Session Flow Control Message
DGRM_FRAME	0x11	Connectionless Datagram Frame for a circuit
CAP_XCHANGE	0x12	Capabilities Exchange Message
CLOSE_PEER_REQUEST	0x13	Disconnect Peer Connection Request
CLOSE_PEER_RESPONSE	0x14	Disconnect Peer Connection Response
PEER_TEST_REQ	0x1D	Peer keepalive test request
PEER_TEST_RSP	0x1E	Peer keepalive response

Table 3-1. DRAP Frames

3.4. DRAP Data formats

The DRAP data is used to carry information required for each DRAP frame. This information is used by the Server or the Client and it does not contain any user data. The DRAP data frame types are listed in the following sections. Please note that the sender should set the reserved fields to zero and the receiver should ignore these fields.

3.4.1. CAN_U_REACH, I_CAN_REACH, and I_CANNOT_REACH Frames

These frame types are used to locate resources in a network. A CAN_U_REACH frame is sent to the server to determine if the resource is reachable. The server responds with an I_CAN_REACH frame if it can reach the workstation identified in the CAN_U_REACH frame, or with an I_CANNOT_REACH if the station is not reachable. The server should not send the CAN_U_REACH frame to the clients. When a server receives an explorer whose destination is a known client, the server should respond to it directly.

Field Name	Information
Message Type	0x01, 0x02, or 0x03
Packet Length	0x0C

Figure 3-3. CAN_U_REACH, I_CAN_REACH, and I_CANNOT_REACH Header

	Field Name (Each row is one byte)
0	Target MAC Address
1	
2	
3	
4	
5	
6	Source SAP
7	Reserved

Figure 3-4. CAN_U_REACH, I_CAN_REACH, and I_CANNOT_REACH Data

The MAC Address field carries the MAC address of the target workstation that is being searched. This is a six-byte MAC Address field. The same MAC Address is returned in the I_CAN_REACH and the I_CANNOT_REACH frames.

Byte 6 is the source SAP. The destination SAP is set to zero when an explorer frame is sent to the network.

If the sender did not receive a positive acknowledgment within a recommended threshold value of 60 seconds, the destination is considered not reachable.

3.4.2. START_DL, DL_STARTED, and START_DL_FAILED Frames

These frame types are used by DRAP to establish a link station (circuit). The START_DL frame is sent directly to the server that responds to the CAN_U_REACH frame. When the server receives this frame, it establishes a link station with the source and destination

addresses and saps provided in the START_DL frame. If the circuit establishment is successful, a DL_STARTED frame is sent back as a response. A failure will result in a START_DL_FAILED response. The server can also send START_DL frames to clients, to establish circuits.

Field Name	Information
Message Type	0x04, 0x05, or 0x06
Packet Length	0x18

Figure 3-5. START_DL, DL_STARTED, and START_DL_FAILED Header

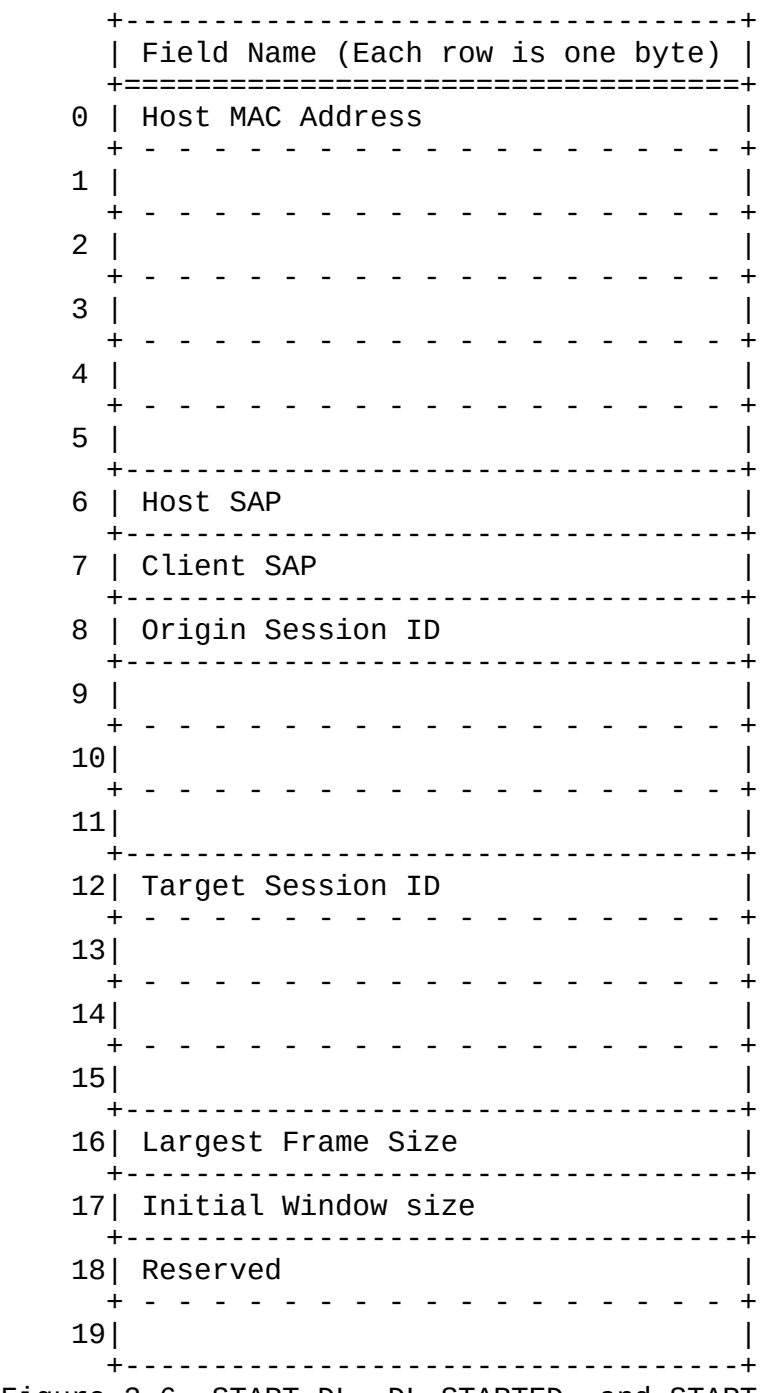

Figure 3-6. START_DL, DL_STARTED, and START_DL_FAILED Data

The Host MAC address is the address of the target station if the session is initiated from the client, or it is the address of the originating station if the session is initiated from the server.

The next two fields are the Host and Client SAPs. Each is one byte long. The Host SAP is the SAP used by the station with the Host MAC address. The Client SAP is the SAP used by the client.

The Origin Session ID, is the ID of the originating station that initiates the circuit. The originating station uses this ID to identify the newly created circuit. Before the START_DL frame is sent to the target station, the originating station sets up a control block for the circuit. This link station information is set because DRAP does not use a three-way handshake for link station establishment. In the DL_STARTED and the START_DL_FAILED messages, the Origin Session ID is returned as received in the START_DL frame. The Target Session ID is set by the target station and returned in the DL_STARTED message.

The Target Session ID is not valid for the START_DL and the START_DL_FAILED frame, and should be treated as Reserved fields. In the DL_STARTED frame, it is the session ID that is used to set up this circuit by the target station.

The Largest Frame Size field is used to indicate the maximum frame size that can be used by the client. It is valid only when it is set by the server. The Largest Frame Size field must be set to zero when a frame is sent by the client. Both START_DL and DL_STARTED use the Largest Frame Size field and only its rightmost 6 bits are used. The format is defined in the IEEE 802.1D Standard, Annex C, Largest Frame Bits (LF). Bit 3 to bit 5 are base bits. Bit 0 to bit 2 are extended bits. The Largest Frame Size field is not used in the START_DL_FAILED frame and must be set to zero.

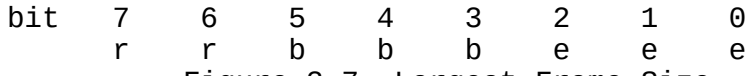

Figure 3-7. Largest Frame Size

Please note that if the client is a PU 2.1 node, the client should use the maximum I-frame size negotiated in the XID3 exchange.

The Initial window size in the START_DL frame gives the receive window size on the originating side, and the target DRAP station returns its receive window size in the DL_STARTED frame. The field is reserved in the START_DL_FAILED frame. The usage of the window size is the same as the one used in DLSw. Please refer to RFC 1795 for details.

The last two bits are reserved for future use. They must be set to zero by the sender and ignored by the receiver.

If the sender of the START_DL frame did not receive a START_DL_FAILED frame within a recommended threshold value of 60 seconds, the connection is considered unsuccessful.

3.4.3. HALT_DL, HALT_DL_NOACK, and DL_HALTED Frames

These frame types are used by DRAP to disconnect a link station. A HALT_DL frame is sent directly to the remote workstation to indicate that the sender wishes to disconnect. When the receiver receives this frame, it tears down the session that is associated with the Original Session ID and the Target Session ID provided in the HALT_DL frame. The receiver should respond with the DL_HALTED frame. The DL_HALTED frame should use the same Session ID values as the received HALT_DL message without swapping them. The HALT_DL_NOACK frame is used when the response is not required.

Field Name	Information
Message Type	0x0C, 0x0D, or 0x0E
Packet Length	0x10

Figure 3-8. HALT_DL, HALT_DL_NOACK, and DL_HALTED Header

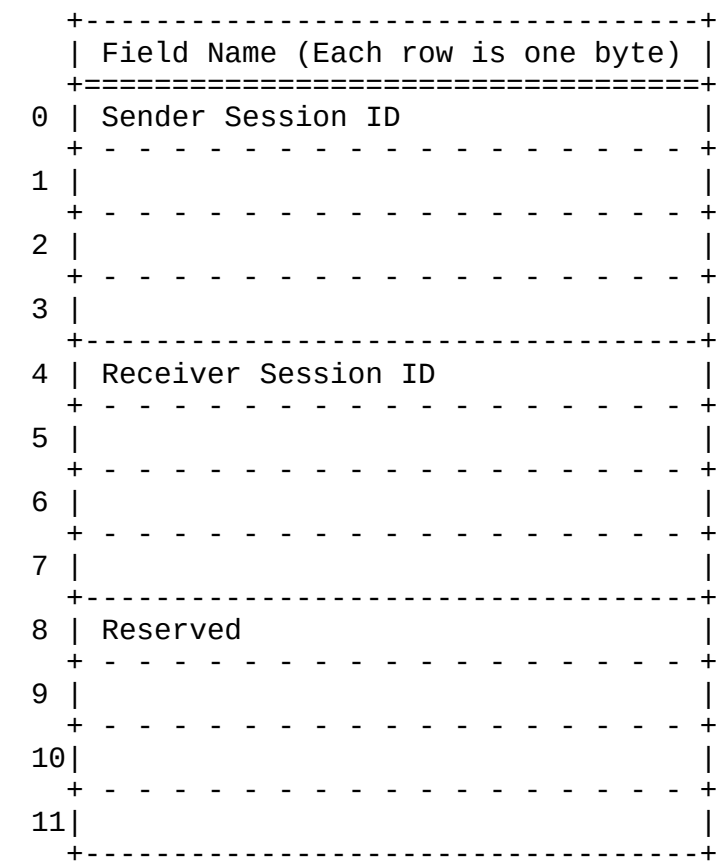

Figure 3-9. START_DL, DL_STARTED, and START_DL_FAILED Data

3.4.4. XID_FRAME, CONTACT_STN, STN_CONTACTED, INFO_FRAME, FCM_FRAME, and DGRM_FRAME

These frame types are used to carry the end-to-end data or establish a circuit. The Destination Session ID is the Session ID created in the START_DL frame or the DL_STARTED frame by the receiver. The usage of the flow control flag is the same as the one used in DLSw. Please refer to RFC 1795 for details.

Field Name	Information
Message Type	Based on Message type
Packet Length	0x0C + length of user data

Figure 3-10. Generic DRAP Header

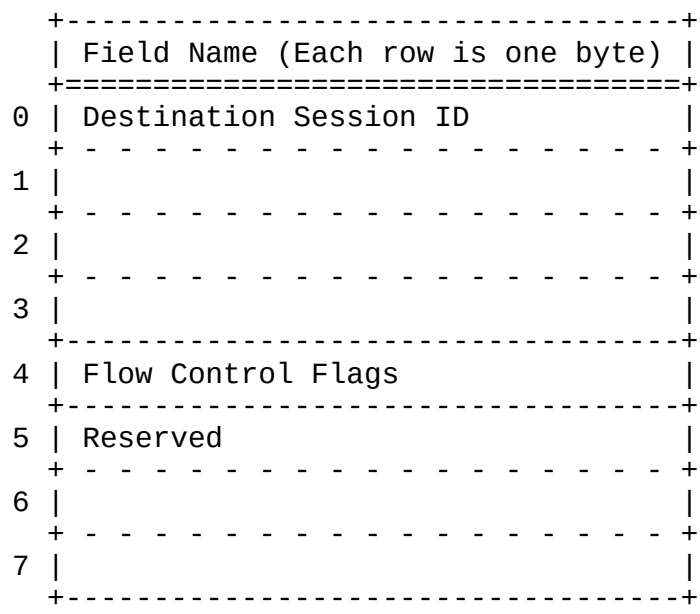

Figure 3-11. Generic DRAP Data Format

3.4.5. DATA_FRAME

This frame type is used to send connectionless SNA and NetBIOS Datagram (UI) frames that do not have a link station associated with the source and destination MAC/SAP pair. The difference between DGRM_FRAME and DATA_FRAME is that DGRM_FRAME is used to send UI frames received for stations that have a link station opened, whereas DATA_FRAME is used for frames with no link station established.

Field Name	Information
Message Type	0x0A
Packet Length	0x10 + Length of user data

Figure 3-12. DATA_FRAME Header

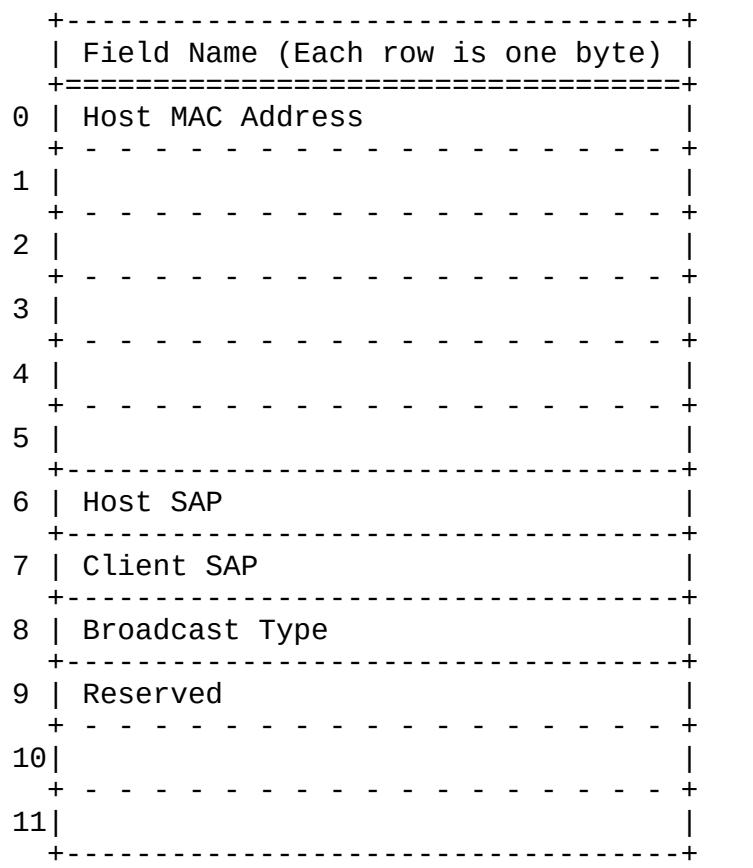

Figure 3-13. DATA_FRAME Data Format

The definition of the first 8 bytes is the same as the START_DL frame. The Broadcast Type field indicates the type of broadcast frames in use; Single Route Broadcast, All Route Broadcast, or Directed. The target side will use the same broadcast type. In the case of Directed frame, if the RIF information is known, the target peer can send a directed frame. If not, a Single Route Broadcast frame is sent.

3.4.6. CAP_XCHANGE Frame

In DRAP, the capability exchange frame is used to exchange the client's information, such as its MAC address, with the server. If a DRAP client has its own MAC address defined, it should put it in the MAC address field. Otherwise, that field must be set to zero.

When the DRAP server receives the CAP_XCHANGE frame, it should cache the MAC address if it is non zero. The DRAP server also verifies that the MAC address is unique. The server should return a CAP_XCHANGE response frame with the MAC address supplied by the client if the MAC

address is accepted. If a client does not have its own MAC address, the server should assign a MAC address to the client and put that address in the CAP_XCHANGE command frame.

A client should record the new MAC address assigned by the server and return a response with the assigned MAC address. If the client cannot accept the assigned MAC address, another CAP_XCHANGE command with the MAC address field set to zero should be sent to the server. The server should allocate a new MAC address for this client.

During the capability exchange, both the client and the server can send command frames. The process stops when either side sends a CAP_XCHANGE response frame. When the response frame is sent, the MAC address in the CAP_XCHANGE frame should be the same as the one in the previous received command. The sender of the CAP_XCHANGE response agrees to use the MAC address defined in the previous command.

The number of CAP_XCHANGE frames that need to be exchanged is determined by the client and the server independently. When the number of exchange frames has exceeded the pre-defined number set by either the server or the client, the session should be brought down.

The flag is used to show the capability of the sender. The following list shows the valid flags:

0x01 NetBIOS support. If a client sets this bit on, the server will pass all NetBIOS explorers to this client. If this bit is not set, only SNA traffic will be sent to this client.

0x02 TCP Listen Mode support. If a client supports TCP listen mode, the server will keep the client's MAC and IP addresses even after the TCP session is down. The cached information will be used for server to connect out. If a client does not support TCP listen mode, the cache will be deleted as soon as the TCP session is down.

0x04 Command/Response. If this bit is set, it is a command, otherwise, it is a response.

The values 0x01 and 0x02 are used only by the client. When a server sends the command/response to a client, the server does not return these values.

Starting with the Reserved field, implementors can optionally implement the Capability Exchange Control Vector. Each Capability Exchange Control Vector consists of three fields: Length (1 byte), Type (1 byte), and Data (Length - 2 bytes). Two types of Control Vectors are defined: SAP_LIST and VENDOR_CODE (described below). To

ensure compatibility, implementors should ignore the unknown Control Vectors instead of treating them as errors.

0x01 SAP_LIST. Length: 2+n bytes, where n ranges from 1 to 16.
This control vector lists the SAPs that the client can support. The maximum number of SAPs a client can define is 16. Therefore, the length of this Control Vector ranges from 3 to 18. If the SAP_LIST is not specified in the capability exchange, the server assumes that the client can support all the SAP values. For example, if a client can only support SAP 4 and 8, then the following Control Vectors should be sent: "0x04, 0x01, 0x04, 0x08". The first byte indicates the length of 4. The second byte indicates the control vector type of SAP_LIST. The last two bytes indicate the supported SAP values; 0x04 and 0x08. This Control Vector is used only by the client. If the server accepts this Control Vector, it must return the same Control Vector to the client.

0x02 VENDOR_CODE. Length: 6 bytes.
Each vendor is assigned a vendor code that identifies the vendor. This Control Vector does not require a response.

After the receiver responds to a Control Vector, if the capability exchange is not done, the sender does not have to send the same Control Vector again.

Field Name	Information
Message Type	0x12
Packet Length	0x1C

Figure 3-14. CAP_XCHANGE Header

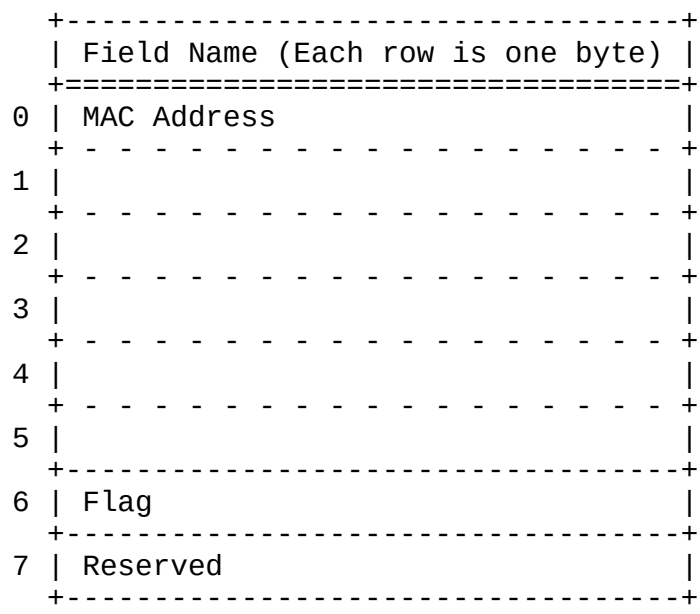

Figure 3-15. CAP_XCHANGE Data Format

3.4.7. CLOSE_PEER_REQ Frames

This frame is used for peer connection management and contains a reason code field. The following list describes the valid reason codes:

0x01 System shutdown. This indicates shutdown in progress.

0x02 Suspend. This code is used when there is no traffic between the server and the client, and the server or the client wishes to suspend the TCP session. When the TCP session is suspended, all circuits should remain intact. The TCP session should be re-established when new user data needs to be sent. When the TCP session is re-established, there is no need to send the CAP_XCHANGE frame again.

0x03 No MAC address available. This code is sent by the server when there is no MAC address is available from the MAC address pool.

Field Name	Information
Message Type	0x13
Packet Length	0x08

Figure 3-16. CLOSE_PEER_REQ Header

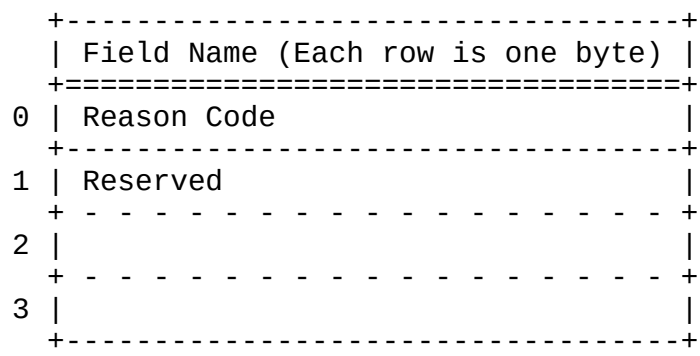

Figure 3-17. CLOSE_PEER_REQ Data Format

3.4.8. CLOSE_PEER_RSP, PEER_TEST_REQ, and PEER_TEST_RSP Frames

These three frames are used for peer connection management. There is no data associated with them.

- CLOSE_PEER_RSP
  CLOSE_PEER_RSP is the response for CLOSE_PEER_REQ.

- PEER_TEST_REQ and PEER_TEST_RSP
  PEER_TEST_REQ and PEER_TEST_RSP are used for peer level keepalive. Implementing PEER_TEST_REQ is optional, but PEER_TEST_RSP must be implemented to respond to the PEER_TEST_REQ frame. When a PEER_TEST_REQ frame is sent to the remote station, the sender expects to receive the PEER_TEST_RSP frame in a predefined time interval (the recommended value is 60 seconds). If the PEER_TEST_RSP frame is not received in the predefined time interval, the sender can send the PEER_TEST_REQ frame again. If a predefined number of PEER_TEST_REQ frames is sent to the remote station, but no PEER_TEST_RSP frame is received (the recommended number is 3), the sender should close the TCP session with this remote station and terminate all associated circuits.

Field Name	Information
Message Type	0x14, 0x1D, or 0x1E
Packet Length	0x04

Figure 3-18. CLOSE_PEER_RSP, PEER_TEST_REQ, and PEER_TEST_RSP DRAP

4. References

[1] Wells, L., Chair, and A. Bartky, Editor, "DLSw: Switch-to-Switch Protocol", RFC 1795, October 1993.

[2] IEEE 802.1D Standard.

Authors' Addresses

Steve T. Chiang
InterWorks Business Unit
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA 95134

Phone: (408) 526-5189
EMail: schiang@cisco.com

Joseph S. Lee
InterWorks Business Unit
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA 95134

Phone: (408) 526-5232
EMail: jolee@cisco.com

Hideaki Yasuda
System Product Center
Network Products Department
Network Software Products Section B
Mitsubishi Electric Corp.
Information Systems Engineering Center
325, Kamimachiya Kamakura Kanagawa 247, Japan

Phone: +81-467-47-2120
EMail: yasuda@eme068.cow.melco.co.jp