Expire in six months

David Bryant
3Com
Corp Paul
Brittain
Data Connection Ltd.

Revised 1/97

APPN Implementer's Workshop
Closed Pages Document

DLSw v2.0 Enhancements

<draft-rfced-info-bryant-00.txt>

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

This memo is the proposed CP document for the DLSw V2.0 enhancements, incorporating all comments received on the AP document. Please post any objections to the granting of CP state or minor editorial corrections to this document on the DLSw mail exploder by 3 February 1997.

If no objections are received, this document will be updated in the week of 3 **February 1997 to accept all the revisions and the change this section to** state that CP status has been granted. The final CP document will then be posted to the DLSw mail exploder and placed on the AIW FTP site.

After CP state has been granted, a copy of this document will be republished as an informational RFC. The changes to the CP document required for publications as an RFC are as follows:

- change header/footers to RFC formats
- change this section to 'standard' informational RFC text
- remove use of illegal text formatting (underline, superscript etc. -

though a copy of
the Word document will be maintained to produce a PostScript version of the RFC)
- convert to NROFF format.

[I will be contacting the RFC editors in the meantime to check that there is nothing else that they will need us to change - Ed.]

The distribution of this memo is unlimited.

Abstract

This document specifies

- a set of extensions to RFC 1795 designed to improve the scalability of DLSw
- clarifications to RFC 1795 in the light of the implementation experience to-date.

It is assumed that the reader is familiar with DLSw and RFC 1795. No effort has been made to explain these existing protocols or associated terminology.

This document was developed in the DLSw Related Interest Group (RIG) of the APPN Implementers Workshop (AIW). If you would like to participate in future DLSw discussions, please subscribe to the DLSw RIG mailing lists by sending a mail to majordomo@raleigh.ibm.com specifying 'subscribe aiw-dlsw' as the body of the message.

1.Introduction

This document defines v2.0 of Data Lnk Switching (DLSw) in the form of a set of enhancements to RFC 1795. These enhancements are designed to be fully backward compatible with existing RFC 1795 implementations. As a compatible set of enhancements to RFC 1795, this document does not replace or supersede RFC 1795.

The bulk of these enhancements address scalability issues in DLSw v1.0. Reason codes have also been added to the HALT_DL and HALT_DL_NOACK SSP messages in order to improve the diagnostic information available.

Finally, the appendix to this document lists a number of clarifications to RFC 1795 where the implementation experience to-date has shown that the original RFC was ambiguous or unclear. These clarifications should be read alongside RFC 1795 to obtain a full specification of the base v1.0 DLSw standard.

2.HALT Reason codes

RFC 1795 provides no mechanism for a DLSw to communicate to its peer the reason for dropping a circuit. DLSw v2.0 adds reason code fields to the HALT_DL and HALT_DL_NOACK SSP messages to carry this information. The reason code is carried as 6 bytes of data after the existing SSP header. The format of these bytes is as shown below.

Byte	Description
0-1	Generic HALT reason code in byte normal format
2-5	Vendor-specific detailed reason code

The generic HALT reason code takes one of the following decimal values (which are chosen to match the disconnect reason codes specified in the DLSw MIB).

1 - Unknown error
2 - Received DISC from end-station
3 - Detected DLC error with end-station
4 - Circuit-level protocol error (e.g., pacing)
5 - Operator-initiated (mgt station or local console)

The vendor-specific detailed reason code may take any value.

All V2.0 DLSws must include this information on all HALT_DL and HALT_DL_NOACK messages sent to v2.0 DLSw peers. For backwards compatibility with RFC 1795, DLSw V2.0 implementations must also accept a HALT_DL or HALT_DL_NOACK message received from a DLSw peer that does not carry this information (i.e. RFC 1795 format for these SSP messages).

3.Scope of Scalability Enhancements

The DLSw Scalability group of the AIW identified a number of scalability issues associated with existing DLSw protocols as defined in RFC 1795:

z Administration

RFC 1795 implies the need to define the transport address of all DLSw peers at each DLSw. In highly meshed situations (such as those often found in NetBIOS networks), the resultant administrative burden is undesirable.

z Address Resolution

RFC 1795 defines point to point TCP (or other reliable transport protocol) connections between DLSw peers. When attempting to discover the location of an unknown resource, a DLSw sends an address resolution packet to each DLSw peer over these connections. In highly meshed configurations, this can result in a very large number of packets in the transport network. Although each packet is sent individually to each DLSw peer, they are each identical in nature. Thus the transport network is burdened with excessive numbers of identical packets. Since the transport network is most commonly a wide area network, where bandwidth is considered a precious resource, this packet duplication is undesirable.

z Broadcast Packets

In addition to the address resolution packets described above, RFC 1795 also propagates NetBIOS broadcast packets into the transport network. The UI frames of NetBIOS are sent as LAN broadcast packets. RFC 1795 propagates these packets over the point to point transport connections to each DLSw peer. In the same manner as above, this creates a large number of identical packets in the transport network, and hence is undesirable. Since NetBIOS UI frames can be sent by applications, it is difficult to predict or control the rate and quantity of such traffic. This compounds the undesirability of the existing RFC 1795 propagation method for these packets.

z TCP (transport connection) Overhead

As defined in RFC 1795, each DLSw maintains a transport connection to its DLSw peers. Each transport connection guarantees in order packet delivery. This is accomplished using acknowledgment and sequencing algorithms which require both CPU and memory at the DLSw endpoints in direct proportion to the number transport connections. The DLSw Scalability group has identified two scenarios where the number of transport connections can become significant resulting in excessive overhead and corresponding equipment costs (memory and CPU). The first scenario is found in highly meshed DLSw configurations where the number of transport connections approximates n2 (where n is the number of DLSw peers). This is typically found in DLSw networks supporting NetBIOS. The second scenario is found in networks where many remote locations communicate to few central sites. In this case, the central sites must support n transport connections (where n is the number of remote sites). In both scenarios the resultant transport connection overhead is considered undesirable depending upon the value of n.

z LLC2 overhead

RFC 1795 specifies that each DLSw provides local termination for the LLC2 (SDLC or other SNA reliable data link protocol) sessions traversing the SSP. Because these reliable data links provide guaranteed in order packet delivery, the memory and CPU overhead of maintaining these connections can also become significant. This is particularly undesirable in the second scenario described above, because the number of reliable connections maintained at the central site is the aggregate of the connections maintained at each remote site.

It is not the intent of this document to address all the undesirable scalability issues associated with RFC 1795. This paper identifies protocol enhancements to RFC 1795 using the inherent multicast capabilities of the underlying transport network to improve the scalability of RFC 1795.

It is believed that the enhancements defined, herein, address many of the issues identified above, such as administration, address resolution, broadcast packets, and, to a lesser extent, transport overhead. This paper does not address LLC2 overhead. Subsequent efforts by the AIW and/or DLSw Scalability group may address the unresolved scalability issues.

While it is the intent of this paper to accommodate all transport protocols as best as is possible, it is recognized that the multicast capabilities of many protocols is not yet well defined, understood, or implemented. Since TCP is the most prevalent DLSw transport protocol in use today, the DLSw Scalability group has chosen to focus its definition around IP based multicast services. This document only addresses the implementation detail of IP based multicast services.

This proposal does not consider the impacts of IPv6 as this was considered too far from widespread use at the time of writing.

4.Overview of Scalability Enhancements

This paper describes the use of multicast services within the transport network to improve the scalability of DLSw based networking. There are only a few main components of this proposal:

z Single session TCP connections

RFC 1795 defines a negotiation protocol for DLSw peers to choose either two unidirectional or one bi-directional TCP connection. DLSws implementing the enhancements described in this document must support and use(whenever required and possible)a single bi-directional TCP connection between DLSw peers. That is to say that the single tunnel negotiation support of RFC 1795 is a prerequisite function to this set of enhancements. Use of two unidirectional TCP connections is only allowed (and required)for migration purposes when communicating with DLSw peers that do not implement these enhancements.

This document also specifies a faster method for bringing up a single TCP connection between two DLSw peers than the negotiation used in RFC 1795. This faster method, detailed in section 6.2.1, must be used where both peers are known to support DLSw v2.0.

z TCP connections on demand

Two DLSw peers using these enhancements will only establish a TCP connection when necessary. SSP connections to DLSw peers which do not implement these enhancements are assumed to be established by the means defined in RFC 1795. DLSws implementing v2.0 utilize UDP based transport services to send address resolution packets (CANUREACH_ex, NETBIOS_NQ_ex, etc.). If a positive response is received, then a TCP connection is only established to the associated DLSw peer if one does not already exist. Correspondingly, TCP connections are brought down when there are no circuits to a DLSw peer for an implementation defined period of time.

z Address resolution through UDP

The main thrust of this paper is to utilize non-reliable transport and the inherent efficiencies of multicast protocols whenever possible and applicable to reduce network overhead. Accordingly, the address resolution protocols of SNA and NetBIOS are sent over the non-reliable transport of IP, namely UDP. In addition, IP multicast/unicast services are used whenever address resolution packets must be sent to multiple destinations. This avoids the need to maintain TCP SSP connections between two DLSw peers when no circuits are active. CANUREACH_ex and ICANREACH_ex packets can be sent to all the appropriate DLSw peers without the need for pre-configured peers or pre-established TCP/IP connections. In addition, most multicast services (including TCP's MOSPF, DVMRP, MIP, etc.) replicate and propagate messages only as necessary to deliver to all multicast members.   This avoids duplication and excessive bandwidth consumption in the transport network.

To further optimize the use of WAN resources, address resolution responses are sent in a directed fashion (i.e., unicast) via UDP transport whenever possible.   This avoids the need to setup or maintain TCP connections when they are not required. It also avoids the bandwidth costs associated with broadcasting.

Note: It is also permitted to send some address resolution traffic over existing TCP connections. The conditions under which this is permitted are detailed in section 7.

z NetBIOS broadcasts over UDP

In the same manner as above, NetBIOS broadcast packets are sent via UDP (unicast and multicast) whenever possible and appropriate. This avoids the need to establish TCP connections between DLSw peers when there are no circuits required.   In addition, bandwidth in the transport network is conserved by utilizing the efficiencies inherent to multicast service implementation. Details covering identification of these packets and proper propagation methods are described in section 10.

5.Multicast Groups and Addressing

IP multicast services provides an unreliable datagram oriented delivery service to multiple parties. Communication is accomplished by sending and/or listening to specific 'multicast' addresses. When a given node sends a packet to a specific address (defined to be within the multicast address range), the IP network (unreliably) delivers the packet to every node listening on that address.

Thus, DLSws can make use of this service by simply sending and receiving (i.e., listening for) packets on the appropriate multicast addresses. With careful planning and implementation, networks can be effectively partitioned and network overhead controlled by sending and listening on different addresses groups. It is not the intent of this paper to define or describe the techniques by which this can be accomplished. It is expected that the networking industry (vendors and end users alike) will determine the most appropriate ways to make use of the functions provided by use of DLSw multicast transport services.

5.1 Using Multicast Groups

The multicast addressing as described above can be effectively used to limit the amount of broadcast/multicast traffic in the network. It is not the intent of this document to describe how individual DLSw/SSP implementations would assign or choose group addresses.   The specifics of how this is done and exposed to the end user is an issue for the specific implementor. In order to provide for multivendor interoperability and simplicity of configuration, however, this paper defines a single IP multicast address, 224.0.10.000, to be used as a default DLSw multicast address. If a given implementation chooses to provide a default multicast address, it is recommended this address be used. In addition, this address should be used for both transmitting and receiving of multicast SSP messages. Implementation of a default multicast address is not, however, required.

5.2 DLSw Multicast Addresses

For the purpose of long term interoperability, the AIW has secured a block of IP multicast addresses to be used with DLSw. These addresses are listed below:

Address Range	Purpose
224.0.10.000	**Default multicast address**
224.0.10.001-191	User defined DLSw multicast groups
224.0.10.192-255	Reserved for future use by the DLSw RIG in DLSw enhancements

6.DLSw Message Transports

With the introduction of DLSw Multicast Protocols, SSP messages are now sent over two distinct transport mechanisms: TCP/IP connections and UDP services. Furthermore, the UDP datagrams can be sent to two different kinds of IP addresses: unique IP addresses (generally associated with a specific DLSw), and multicast IP addresses (generally associated with a group of DLSw peers).

6.1 TCP/IP Connections on Demand

As is the case in RFC 1795, TCP/IP connections are established between DLSw peers. Unlike RFC 1795, however, TCP/IP connections are only established to carry reliable circuit data (i.e., LLC2 based circuits). Accordingly, a

TCP/IP connection is only established to a given DLSw peer when the first circuit to that DLSw is required (i.e., the origin DLSw must send a CANUREACH_CS to a target DLSw peer and there is no existing TCP connection between the two). In addition, the TCP/IP connection is brought down an implementation defined amount of time after the last active (not pending) circuit has terminated. In this way, the overhead associated with maintaining TCP connections is minimized.

With the advent of TCP connections on demand, the activation and deactivation of TCP connections becomes a normal occurrence as opposed to the exception event it constitutes in RFC 1795. For this reason, it is recommended that implementations carefully consider the value of SNMP traps for this condition.

6.1.1 TCP Connections on Demand Race Conditions

Non-circuit based SSP packets (e.g.,CANUREACH_ex, etc.) may still be sent/received over TCP connections after all circuits have been terminated. Taking this into account implementations should still gracefully terminate these TCP connections once the connection is no longer supporting circuits. This may require an implementation to retransmit request frames over UDP when no response to a TCP based unicast request is received and the TCP connection is brought down. This is not required in the case of multicast requests as these are received over the multicast transport mechanism.

6.2 Single Session TCP/IP Connections

RFC 1795 defines the use of two unidirectional TCP/IP sessions between any pair of DLSw peers using read port number 2065 and write port number 2067. Additionally, RFC 1795 allows for implementations to optionally use only one bi-directional TCP/IP session. Using one TCP/IP session between DLSw peers is believed to significantly improve the performance and scalability of DLSw protocols. Performance is improved because TCP/IP acknowledgments are much more likely to be piggy-backed on real data when TCP/IP sessions are used bi-directionally. Scalability is improved because fewer TCP control blocks, state machines, and associated message buffers are required. For these reasons, the DLSw enhancements defined in this paper REQUIRE the use of single session TCP/IP sessions.

Accordingly, DLSws implementing these enhancements must carry the TCP Connections Control Vector in their Capabilities Exchange. In addition, the TCP Connections Control Vector must indicate support for 1 connection.

6.2.1 Expedited Single Session TCP/IP Connections

In RFC 1795, single session TCP/IP connections are accomplished by first establishing two uni-directional TCP connections, exchanging capabilities, and then bringing down one of the connections. In order to avoid the unnecessary flows and time delays associated with this process, a new single session bi-directional TCP/IP connection establishment algorithm is defined.

6.2.1.1TCP Port Numbers

DLSws implementing these enhancements will use a TCP destination port of **2067 (as opposed to RFC 1795 which uses 2065) for single session TCP** connections. The source port will be a random port number using the established TCP norms which exclude the possibility of either 2065 or 2067.

6.2.1.2TCP Connection Setup

DLSw peers implementing these enhancements will establish a single session TCP connection whenever the associated peer is known to support this capability. To do this, the initiating DLSw simply sends a TCP setup request to destination port 2067. The receiving DLSw responds accordingly and the TCP three way handshake ensues. Once this handshake has completed, each DLSw is notified and the DLSw capabilities exchange ensues. As in RFC 1795, no flows may take place until the capabilities exchange completes.

6.2.1.3Single Session Setup Race Conditions

The new expedited single session setup procedure described above opens up the possibility of a race condition that occurs when two DLSw peers attempt to setup single session TCP connections to each other at the same time. To avoid the establishment of two TCP connections, the following rules are applied when establishing expedited single session TCP connections:

1. If an inbound TCP connect indication is received on port 2067 while an outbound TCP connect request (on port 2067) to the same DLSw (IP address) is in process or outstanding, the DLSw with the higher IP address will close or reject the connection from the DLSw with the lower IP address.
2. To further expedite the process, the DLSw with the lower IP address may choose (implementation option) to close its connection request to the DLSw with the higher address when this condition is detected.
3. If the DLSw with the lower IP address has already sent its capabilities exchange request on its connection to the DLSw with the higher IP address, it must resend its capabilities exchange request over the remaining TCP connection from its DLSw peer (with the higher IP address).
4. The DLSw with the higher IP address must ignore any capabilities exchange request received over the TCP connection to be terminated (the one from the DLSw with the lower IP address).

6.2.1.4TCP Connections with Non-Multicast Capable DLSw peers

During periods of migration, it is possible that TCP connections between multicast capable and non-multicast capable DLSw peers will occur. It is also possible that multicast capable DLSws may attempt to establish TCP connections with partners of unknown capabilities (e.g., statically defined peers). To handle these conditions the following additional rules apply to expedited single session TCP connection setup:

1.1.If the capability of a DLSw peer is not known, an implementation may choose to send the initial TCP connect request to either port 2067 (expedited single session setup) or port 2065 (standard RFC 1795 TCP setup).

2.2.If a multicast capable DLSw receives an inbound TCP connect request on port 2065 while processing an outbound request on 2067 to the same DLSw, the sending DLSw will terminate its 2067 request and respond as defined in RFC 1795 with an outbound 2065 request (standard RFC 1795 TCP setup).

3.If a multicast capable DLSw receives an indication that the DLSw peer is not multicast capable (the port 2067 setup request times out or a port not recognized rejection is received), it will send another connection request using port 2065 and the standard RFC 1795 session setup protocol.

6.3 UDP Datagrams

As mentioned above, UDP datagrams can be sent two different ways: unicast (e.g., sent to a single unique IP address) or multicast (i.e., sent to an IP multicast address). Throughout this document, the term UDP datagram will be used to refer to SSP messages sent over UDP, while unicast and multicast SSP messages will refer to the specific type/method of UDP packet transport. In either case, standard UDP services are used to transport these packets. In order to properly parse the inbound UDP packets and deliver them to the SSP state machines, all DLSw UDP packets will use the destination port of 2067.
In addition, the checksum function of UDP remains optional for DLSw SSP messages. It is believed that the inherent CRC capabilities of all data link transports will adequately protect SSP packets during transmission. And the incremental exposure to intermediate nodal data corruption is negligible. For further information on UDP packet formats see the "Frame Formats" section.

6.3.1 Vendor Specific Functions over UDP

In order to accommodate vendor specific capabilities over UDP transport, a new SSP packet format has been defined. This new packet format is required because message traffic of this type is not necessarily preceded by a capabilities exchange. Accordingly, DLSw's wishing to invoke a vendor specific function may send out this new SSP packet format over UDP.

Because this packet can be sent over TCP connections and non-multicast capable nodes may not be able to recognize it, implementations may only send this packet over TCP to DLSw peers known to understand this packet format (i.e., multicast capable). To avoid this situation in the future, DLSws implementing these enhancements must ignore SSP packets with an unrecognized DLSw version number in the range of x'31' to x'3F'. Further information and the precise format for this new packet type is described below in the "Frame Formats" section.

6.3.2 Unicast UDP Datagrams

Generically speaking, a unicast UDP datagram is utilized whenever an SSP message (not requiring reliable transport) must be sent to a unique set (not all) of DLSw peers. This avoids the overhead of having to establish and maintain TCP connections when they are not required for reliable data transport.

A typical example of when unicast UDP might be used would be an ICANREACH_ex response from a peer DLSw (with which no TCP connection currently exists). In this case, the sending DLSw knows the IP address of the intended receiver and can simply send the response via unicast UDP. In addition, there are a number of NetBIOS cases where unicast UDP is used to handle UI frames directed to a specific DLSw (e.g., NetBIOS STATUS_RESPONSE). Further detail is provided in the NetBIOS section of this document.

6.3.3 Multicast UDP Datagrams

In a broad sense, multicast UDP datagrams are used whenever a given SSP message must be sent to multiple DLSw peers. In the case of SNA, this is primarily the CANUREACH_ex packets. In the case of NetBIOS, multicast datagrams are used to send broadcast UI frames such as NetBIOS user datagrams and broadcast datagrams.

Note, however, it is sometimes possible to avoid broadcasting certain NetBIOS frames that would otherwise be broadcast in the LAN environment. This is typically accomplished using name caching techniques not described in this paper. In cases of this type when a single destination DLSw can be determined, unicast transport can be used to send the 'broadcast' NetBIOS frame to a single destination. A more detailed listing of NetBIOS SSP packets and transport methods can be found in the NetBIOS section of this document.

6.4 Unicast UDP Datagrams in Lieu of IP Multicast

Because the use of IP multicast services is actually a function of IP itself and not DLSw proper, it is possible for implementations to simply make use of the UDP transport mechanisms described in this paper without making direct use of the IP multicast function. While this is not considered to be as efficient as using multicast transport mechanisms, this practice is not explicitly prohibited.

Implementations which choose to make use of UDP transport in this manner must first know the IP address of all the potential target DLSw peers and send individual unicast packets to each. How this information is obtained and/or maintained is outside the scope of this paper.

As a matter of compliance, implementers need not send SSP packets outbound over UDP as there are some conditions where this may not be necessary or desirable. It is, however, required that implementers provide an option to receive SSP packets over UDP.

6.5 TCP Transport

Despite the addition of UDP based packet transport, TCP remains the fundamental form of communications between DLSw peers. In particular, TCP is still used to carry all LLC2 based circuit data.

Throughout this document wherever UDP unicast (not multicast) is discussed, the reader should be aware that TCP may be used instead. Moreover, it is strongly recommended that TCP be used in preference to UDP whenever a TCP connection to the destination already exists. Implementations, however, should be prepared to receive SSP packets from either transport (TCP or UDP).

7.Migration Support

It is anticipated that some networks will experience a transition stage where both RFC 1795 (referred to as 'non-multicast' DLSws) and 'multicast capable' DLSws will exist in the network at the same time. It will be important for these two DLSw node types to interoperate and thus the following accommodations for non-multicast DLSws are required:

7.1 Capabilities Exchange

In order to guarantee both backward and forward capability, DLSws which implement these multicast enhancements will carry a "Multicast Capabilities" Control Vector in their capabilities exchange (see RFC 1795 for an explanation of capabilities exchange protocols). Presence of the Multicast Capabilities control vector indicates support for the protocols defined in this document on a per DLSw peer basis. Conversely, lack of the Multicast Capabilities control vector indicates no support for these extensions on a per DLSw peer basis.

Additionally, nodes implementing these enhancement will carry a modified DLSw Version control vector (x'82') indicating support for version 2 release 0.

Lastly, presence of these control vectors mandates a TCP Connections Control Vector indicating support for 1 TCP connection in the same Capabilities exchange.

If a multicast capable DLSw receives a Capabilities Exchange CV that includes the Multicast Capabilites CV but does not meet the above criteria, it must reject the capabilities exchange by sending a negative response as described in section 11.1.1.

7.2 Connecting to Non-Multicast Capable Nodes

It is assumed that TCP connections to DLSw peers which do not support multicast services are established by some means outside the scope of this paper (i.e., non-multicast partner addresses are configured by the customer). TCP connections must be established and maintained to down level nodes in the exact same manner as RFC 1795 requires, establishes, and maintains them. And because non-multicast DLSw peers will not indicate support for multicast services in their capabilities exchange, a multicast capable DLSw will know all its non-multicast peers.

7.3 Communicating with Multicast Capable Nodes

Because non-multicast nodes will not receive SSP frames via UDP (unicast or multicast) transmission, SSP messages to these DLSw peers must be sent over TCP connections.   Therefore, nodes which implement the multicast protocol enhancements must keep track of which DLSw peers do not support multicast extensions (as indicated in the capabilities exchange). When a given packet is sent out via multicast services, it must also be sent over multicast UDP(to reach other multicast capable DLSw peers) and over the TCP connection to each non-multicast node.   And although the multicast service requires periodic retransmissions (for reliability reasons), this is not the case with TCP connections to non-multicast nodes. Therefore, multicast capable DLSws should not resend SSP packets over TCP transport connection but rather, rely upon TCP to recover any lost packets. Furthermore, communications with non-multicast nodes should be in exact compliance with RFC 1795 protocols.

When sending a unicast UDP message, it is important to know that the destination DLSw supports multicast services. This knowledge can be obtained from previous TCP connections/capabilities exchanges or inferred from a previously received UDP message, but how this information is obtained is outside the scope of this paper. In the latter case, if the DLSw is non-multicast, then there would be a TCP connection to it and it would be known to be non-multicast. If it is multicast capable and a TCP connection is in existence, then its level is known (via the prior capabilities exchange). If its capabilities are not known and there is not an existing TCP connection, then it can be implied to be multicast capable by virtue of a cached entry but no active TCP connection (e.g., TCP peer on demand support). This inference, however, could be erroneous in cases where the TCP connection (to a non-multicast DLSw) has failed for some reason. But normal UDP based unicast verification mechanisms will detect no active path to the destination and circuit setup will proceed correctly (i.e., succeed or fail in accordance with true connectivity).

8.SNA Support

Note: This paper does not attempt to address the unique issues presented by SNA/HPR and its non-ERP data links

In SNA protocols the generalized packet sequence of interest is a test frame exchange followed by an XID exchange. In all cases, DLSw uses the CANUREACH_ex and ICANREACH_ex SSP packets to complete address resolution and circuit establishment. The following table describes how these packets are transported via UDP between two multicast capable DLSw peers.

Message Event	Action	Transport Mechanism	Retry
TEST	SEND CANUREACH_ex	Multicast/Unicast	Yes
TEST RESPONSE	SEND ICANREACH_ex	Unicast	No

The following paragraphs provide more detail on how UDP transport and multicast protocol enhancements are used to establish SNA data links.

8.1 Address Resolution

When a DLSw receives an incoming test frame from an attached data link, the assumption is that this is an exploratory frame in preparation for an XID exchange and link activation. The DLSw must determine a correlation between the destination LSAP (mac and sap pairing) and some other DLSw in the transport network.   This paper generically refers to this process as "address resolution".

8.2 Explorer frames

Address resolution messages may be sent over a TCP connection to a multicast capable DLSw peer if such a connection already exists in order that they take advantage of the guaranteed delivery of TCP. This is particularly recommended for ICANREACH_ex frames.

8.3 Circuit Setup

Circuit setup is accomplished in the same manner as described in RFC 1795. More specifically, CANUREACH_cs, ICANREACH_cs, REACH_ACK, XIDFRAME, etc. are all sent over the TCP connection to the appropriate DLSw.   This, of course, assumes the existence of a TCP connection between the DLSw peers. If the sending DLSw (sending a CANUREACH_cs) detects no active TCP connection to the DLSw peer, then a TCP connection setup is initiated and the packet sent. All other circuit setup (and takedown) related sequences are now passed over the TCP connection.

8.4 Example SNA SSP Message Sequence

The following diagram provides an example sequence of flows associated with an SNA LLC circuit setup.   All flows and states described below correspond precisely with those defined in RFC 1795. The only exception is the addition of a TCP connection setup and DLSw capabilities exchange that occurs when the origin DLSw must send a CANUREACH_CS and no TCP connection yet exists to the target DLSw peer.

```
 ======                            ___                           ======
 |    |        ---------        __/   \__        ---------        |    |
 |    |      __|  _|_  |__     /   IP    \    __|  _|_  |__      |    |
 ======        |   |   |      <  Network  >     |   |   |        ======
/______\       ---------       \__     __/      ---------       /______\
 Origin       Origin DLSw         \___/        Target DLSw      Target
 Station        partner                          partner        Station

              disconnected                    disconnected

TEST_cmd      DLC_RESOLVE_C    CANUREACH_ex               TEST_cmd
```

```
----------->  ----------->      ----------->               ---------->
   TEST_rsp   DLC_RESOLVE_R    ICANREACH_ex                 TEST_rsp
 <---------    <-----------   <-----------               <----------
null XID      DLC_XID
----------->  ----------->
              circuit_start

                        TCP Connection Setup
                        <-------------->
                        Capabilities Exch.
                        <-------------->

                         CANUREACH_cs    DLC_START_DL
                         ----------->    ----------->
                                      resolve_pending
                                ICANREACH_cs    DLC_DL_STARTED
                                <-----------    <-------------
           circuit_established                ircuit_pending
                              REACH_ACK
                              ----------->  circuit_established

                              XIDFRAME         DLC_XID       null XID
                              ----------->     --------->    -------->
        XID        DLC_XID        XIDFRAME         DLC_XID         XID
  <--------   <-----------    <-----------    <-----------    <--------
    XIDs         DLC_XIDs      XIDFRAMEs        DLC_XIDs        XIDs
<---------->  <----------> <-------------> <------------->  <---------->
SABME         DLC_CONTACTED   CONTACT         DLC_CONTACT     SABME
----------->  ----------->    ----------->    ----------->    -------->
              connect_pending                 contact_pending

          UA     DLC_CONTACT     CONTACTED    DLC_CONTACTED          UA
  <---------   <-----------   <-----------    <-----------    <--------
                  connected                        connected
IFRAMEs       DLC_INFOs        IFRAMEs        DLC_INFOs       IFRAMEs
<---------->  <----------->  <-------------> <------------->  <-------->
```

8.5 UDP Reliability

It is important to note, that UDP (unicast and multicast)transport services do not provide a reliable means of delivery. Existing RFC 1795 protocols guarantee the delivery (or failure notification) of CANUREACH_ex and ICANREACH_ex messages. UDP will not provide the same level of reliability. It is, therefore, possible that these messages may be lost in the network and (CANUREACH_ex) retries will be necessary.

8.5.1 Retries

Test Frames are generally initiated by end stations every few seconds. Many existing RFC 1795 DLSw implementations take advantage of the reliable

SSP TCP connections and filter out end station Test frame retries when a CANUREACH_ex is outstanding. Given the unreliable nature of UDP transport for these messages, however, this filtering technique may not be advisable. Neither RFC 1795 nor this paper address this issue specifically. It is simply noted that the UDP transport mechanism is unreliable and implementations should take this into account when determining a scheme for Test frame filtering and explorer retries. Accordingly, the "Retry" section in the table above only serves as an indicator of situations where retries may be desirable and/or necessary, but does not imply any requirement to implement retries. Also note, that retry logic only applies to non-response type packets. It is not appropriate to retry response type SSP packets (i.e., ICANREACH_ex) as there is no way of knowing if the original response was ever received (and whether retry is necessary). So in the case of SNA, CANUREACH_ex messages may need retry logic and ICANREACH_ex messages do not.

9.NetBIOS

With the introduction of DLSw Multicast transport, all multicast NetBIOS UI frames are carried outside the TCP connections between DLSw peers (i.e., via UDP datagrams). The following table defines the various NetBIOS UI frames and how they are transported via UDP between multicast capable DLSw peers:

		Transport	
Message Event	Action	Mechanism	Retry
ADD_GROUP_NAME_QUERY	SEND DATAFRAME	Multicast	Yes
ADD_NAME_QUERY	SEND NETBIOS_ANQ	Multicast	Yes
ADD_NAME_RESPONSE	SEND NETBIOS_ANR	Unicast1	No
NAME_IN_CONFLICT	SEND DATAFRAME	Multicast	No
STATUS_QUERY	SEND DATAFRAME	Unicast/Multicast2	Yes
STATUS_RESPONSE	SEND DATAFRAME	Multicast5	No
TERMINATE_TRACE (x'07')	SEND DATAFRAME	Multicast	No
TERMINATE_TRACE (X'13')	SEND DATAFRAME	Multicast	No
DATAGRAM	SEND DATAFRAME3	Unicast/Multicast2	No
DATAGRAM_BROADCAST	SEND DATAFRAME	Multicast	No
NAME_QUERY	SEND NETBIOS_NQ_ex	Unicast/Multicast2	Yes
NAME_RECOGNIZED	SEND NETBIOS_NR_ex	Unicast4	No

Note 1:
Upon receipt of an ADD_NAME_RESPONSE frame, a NETBIOS_ANR SSP message is returned via unicast UDP to the originator of the NETBIOS_ANQ message.

Note 2:
These frames may be sent either Unicast or Multicast UDP. If the implementation has sufficient cached information to resolve the NetBIOS datagram destination to a single DLSw peer, then the SSP message can and should be sent via unicast. If the cache does not contain such information then the resultant SSP message must be sent via multicast UDP.

Note 3:
Note that this frame is sent as either a DATAFRAME or DGRMFRAME according to the rules as specified in RFC 1795.

Note 4:
Upon receipt of a NAME_RECOGNIZED frame, a NETBIOS_NR_ex SSP message is returned via unicast UDP to the originator of the NETBIOS_NQ_ex frame. Notice that although the NAME_RECOGNIZED frame is sent as an All Routes Explorer (source routing LANs only) frame, the resultant NETBIOS_NR_ex is sent as a unicast UDP directed response to the DLSw originating the NETBIOS_NQ_ex. This is because there is no value in sending NETBIOS_NR_ex as a multicast packet in the transport network. The use of ARE transmission in the LAN environment is to accomplish some form of load sharing in the source routed LAN environment. Since no analogous capability exists in the (TCP) transport network, it is not necessary to emulate this function there. It is important to note, however, that when converting a received NETBIOS_NR_ex to a NAME_RECOGNIZED frame, the DLSw sends the NAME_RECOGNIZED frame onto the LAN as an ARE (source routing LANs only) frame. This preserves the source route load sharing in the LAN environments on either side of the DLSw transport network.

Note 5:
Although RFC 1795 does not attempt to optimize STATUS_RESPONSE processing, it is possible to send a STATUS_RESPONSE as a unicast UDP response. To do this, DLSws receiving an incoming SSP DATAFRAME containing a STATUS_QUERY must remember the originating DLSw's address and STATUS_QUERY correlator. Then upon receipt of the corresponding STATUS_RESPONSE, the DLSw responds via unicast UDP to the originating DLSw(using the remembered originating DLSw address). Note, however, that in order to determine whether a frame is a STATUS_QUERY, all multicast capable DLSw implementations will need to parse the contents of frames that would normally be sent as DATAFRAME SSP messages.

All other multicast frames are sent into the transport network using the appropriate multicast group address.

9.1 Address Resolution

Typical NetBIOS circuit setup using multicast services is essentially the same as specified in RFC 1795. The only significant difference is that NETBIOS_NQ_ex messages are sent via UDP to the appropriate unicast/multicast IP address and the NETBIOS_NR_ex is sent via unicast UDP to the DLSw originating the NETBIOS_NQ_ex.

9.2 Explorer Frames

Address resolution messages may be sent over a TCP connection to a multicast capable partner if such a connection already exists in order that they take advantage of the guaranteed delivery of TCP. This is particularly recommended for NETBIOS_NR_ex frames.

9.3 Circuit Setup

Following successful address resolution, a NetBIOS end station typically sends a SABME frame to initiate a formal LLC2 connection. Receipt of this message results in normal circuit setup as described in RFC 1795 (and the SNA case described above). That is to say that the CANUREACH_cs messages etc. are sent on a TCP connection to the appropriate DLSw peer. If no such TCP connection exists, one is brought up.

9.4 Example NetBIOS SSP Message Sequence

The following diagram provides an example sequence of flows associated with a NetBIOS circuit setup. All flows and states described below correspond precisely with those defined in RFC 1795. The only exception is the addition of a TCP connection setup and DLSw capabilities exchange that occurs when the origin DLSw must send a CANUREACH_cs and no TCP connection yet exists to the target DLSw peer.

```
 ======                            ___                            ======
 |    |        ---------        __/   \__       ---------         |    |
 |    |      __|  _|_  |__     /  IP     \    __|  _|_  |__       |    |
 ======        |   |   |      <  Network  >     |   |   |         ======
/______\       ---------       \__     __/      ---------        /______\
 Origin       Origin DLSw         \___/        Target DLSw       Target
 Station        partner                          partner         Station

              disconnected                     disconnected

NAME_QUERY    DLC_DGRM        NETBIOS_NQ_ex   DLC_DGRM        NAME_QUERY
----------->  ----------->    ----------->    ----------->    --------->

   NAME_RECOG     DLC_DGRM       NETBIOS_NR_ex     DLC_DGRM      NAME_RECOG
 <-----------   <------------    <-----------    <-----------  <---------

SABME         DLC_CONTACTED
----------->  ----------->
               circuit_start

                        TCP Connection Setup
                        <------------->
                        Capabilities Exch.
                        <------------->

                         CANUREACH_cs    DLC_START_DL
                         ----------->    ----------->
                                      resolve_pending


                                ICANREACH_cs    DLC_DL_STARTED
                                <-----------    <-----------
```

```
            circuit_established                circuit_pending
                              REACH_ACK
                              ----------->   circuit_established

                              CONTACT         DLC_CONTACT     SABME
                              ----------->    ----------->    --------->
             connect_pending                 contact_pending

          UA   DLC_CONTACT       CONTACTED    DLC_CONTACTED           UA
  <---------  <-----------    <-----------    <-----------    <---------
                connected                       connected

   IFRAMEs       DLC_INFOs       IFRAMEs        DLC_INFOs       IFRAMEs
<------------> <------------> <------------>  <------------>  <-------->
```

9.5 Multicast Reliability and Retries

In the case of NetBIOS, many more packets are being sent via UDP than in the SNA case. Therefore, the exposure to the unreliability of these services is greater than that of SNA. For address resolution frames, such as NAME_QUERY, etc., successful message delivery is an issue. In addition, the retry interval for these types of frames is considerably shorter than SNA with the defaults being: retry interval = 0.5 seconds and retry count = 6. Once again, neither RFC 1795 or this paper attempt to address the issue of LAN frame filtering optimizations. This issue is outside the scope of this paper. But it is important for implementers to recognize the inherent unreliable nature of UDP transport services for frames of this type and to implement retry schemes that are appropriate to successful operation. Again, it is only appropriate to consider retry of non-response type packets. Specific NetBIOS messages where successful message delivery is considered important (and retries possibly necessary) are indicated in the table above with an "Yes" in the "Retry" column.

10. Sequencing

It is important to note that UDP transport services do not provide guaranteed packet sequencing like TCP does for RFC 1795. In a steady state network, in order packet delivery can be generally assumed. But in the presence of network outages and topology changes, packets may take alternate routes to the destination and arrive out of sequence with respect to their original transmission order. For SNA address resolution this should not be a problem given that there is no inherent significance to the order of packets being transmitted via UDP.

In the case of NetBIOS, in order delivery is not guaranteed in the normal case (e.g., LANs). This is because LAN broadcasting mechanisms suffer the same problems of packet sequencing as do WAN multicast mechanisms. But one might argue the greater likelihood of topology related changes in the WAN environment and thus a greater level of concern. The vast majority of NetBIOS UI frames (being handled via UDP and Multicast) have correlator values and do not rely upon packet sequencing.

The only NetBIOS frames of special note would be: DATAGRAM, DATAGRAM_BROADCAST, and STATUS_RESPONSE. In the case of DATAGRAM and DATAGRAM_BROADCAST it is generally assumed that datagrams do not provide any guarantee of in order packet delivery. Thus applications utilizing this NetBIOS service are assumed to have no dependency on in order packet delivery.   STATUS_RESPONSE can actually be sent as a sequence of STATUS_RESPONSE messages. In cases where this occurs, the STATUS_RESPONSE will be exposed to potential out of sequence delivery.

11. Frame Formats

11.1 Multicast Capabilities Control Vector

This control vector is carried in the Capabilities Exchange Request. When present, it must be accompanied by a TCP Connections Control Vector indicating support for 1 TCP/IP connection and a DLSw version CV indicating support for version 2 release 0. Like all control vectors in this SSP message, it is an LT structure. LT structures consist of a 1 byte length field followed by a 1 byte type field.   The length field includes itself as well as the type and data fields.

Byte Bit   Description
0   **0-7**   Length, in binary, of the Multicast Capabilities control vector (inclusive of this byte, always 3)

1   **0-7**   Type: x'8C'

2   **0-7**   Multicast Version Number:
A binary numerical representation of the level of multicast services provided. The protocols as identified in this document constitute version one.   Accordingly, x'01' is encoded in this field. Any subsequent version must provide the services of all previous versions.

The intended use of this CV for Multicast support is to detect when the multicast CUR_ex flows will suffice between partners. If this CV is present in a CAPEX from a partner, that partner is also multicast capable and therefore does not need to receive CANUREACH_ex messages over the TCP link that exists between them (and there must be one or else the CAPEX would not have flowed) because it will receive the multicast copies.

A DLSw includes this control vector on a peer-wise basis. That is to say, that a DLSw implementation may support multicast services but choose not to indicate this in its capabilities exchange to all partners. Therefore, a DLSw may include this capabilities CV with some DLSw peers and not with others. Not including this vector can be used to force TCP connections with other multicast capable nodes and degrade to normal RFC 1795 operations. This capability is allowed to provide greater network design flexibility.

When sending this capabilities exchange control vector, the following rules apply:

ID	Required @ Startup	Length	Repeatable*	Allowed @ Runtime	Order	Content
0x8C	Y	0x03	N	N	5+	Multicast Capabilities

*Note: "Repeatable" means a Control Vector is repeatable within a single message.

11.1.1 DLSw Capabilities Negative Response

DLSws that implement these enhancements must provide support for both multicast version 1 and single TCP connections. This means that the capabilities exchange request must contain a DLSw Version ID control vector (x'82') indicating support for version 2 release 0, a Multicast Capabilities control vector, and the TCP Connections control vector indicating support for 1 TCP connection within a given capabilities exchange. If a multicast capable DLSw receives a capabilities exchange with a Multicast Capabilities, but either a missing or inappropriate TCP Connections CV (i.e., connections not equal to one)or DLSw Version control vector, then the inbound capabilities exchange should be rejected with a DLSw capabilities exchange negative response (see RFC 1795) using the following new reason code:

x'000D'Inconsistent DLSw Version, Multicast Capabilities, and TCP
Connections CV
received on the inbound Capabilities exchange

11.2 UDP Packets

SSP frame formats are defined in RFC 1795. Multicast protocol enhancements do not change these formats in any way. The multicast protocol enhancements, however, do introduce the notion of SSP packet transport via UDP. In this case, standard UDP services and headers are used to transport SSP packets.

The following section describes the proper UDP header for DLSw SSP packets.

Byte   Description
0-1   Source Port address
             In DLSw multicast protocols, this particular field is not
             relevant. It may be set to any value.

2-3   Destination Port address
          Always set to 2067

4-5   Length

```
6-7         Checksum
             The standard UDP checksum value.  Use of the UDP checksum
             function is optional.
```

11.3 Vendor Specific UDP Packets

In order to accommodate the addition of vendor specific functions over UDP transport, a new SSP packet header has been defined. As described above, it is possible to receive these packets over both UDP and TCP (when a TCP connection already exists).

It is important to note that the first 4 bytes of this packet match the format of existing RFC 1795 SSP packets. This is done so that implementations in the future can expect that the DLSw "Version Number" is found in byte one and that the following bytes describe the packet header and message length.

Furthermore, to assist DLSws in detecting 'out-of-sync' conditions whereby packet or parsing errors lead to improper length interpretations in the TCP datastream, valid DLSw version numbers will be restricted to the range of x'31' through x'3F' inclusive.

DLSw multicast Vendor Specific frame format differs from existing RFC 1795 packets in the following ways:

1) The "Version Number" field is set to x'32' (ASCII '2') and now represents a packet type more than a DLSw version number. More precisely, it is permitted and expected that DLSw may send packets of both types (x'31' and x'32').

2) The message length field is followed by a new 3 byte field that contains the specific vendor's IEEE Organizationally Unique Identifier (OUI).

3) All fields following the new OUI field are arbitrary and defined by implementers.

The following section defines this new packet format:

```
Byte        Description
0           DLSw packet type, Always set to x'32'

1          Header Length
           Always 7 or higher

2-3         Message Length
           Number of bytes within the data field following the header.


4-6         Vendor specific OUI
             The IEEE Organizationally Unique Identifier (OUI) associated
             with the vendor specific function in question.
```

```
7-n        Defined by the OUI owner
```

12. Compliance Statement

All DLSw v2.0 implementations must support

- Halt reason codes
- the Multicast Capabilities control vector in the DLSw capabilities exchanges messages.

The presence of the Multicast Capabilities control vector in a capabilities exchange message implies that the DLSw that issued the message supports all the scalability enhancements defined in this document. These are:

- use of multicast IP (if it is available in the underlying network)
- use of 2067 as the destination port for UDP and TCP connections
- single tunnel bring-up of TCP connections to DLSw peers
- peer-on-demand
- quiet ignore of all unrecognized vendor-specific UDP/TCP packets.

13. Security Considerations

This document addresses only scalability problems in RFC 1795. No attempt is made to define any additional security mechanisms. Note that, as in RFC 1795, a given implementation may still choose to refuse TCP connections from DLSw peers that have not been configured by the user. The mechanism by which the user configures this behavior is not specified in this document.

14. Acknowledgements

This specification was developed in the DLSw Related Interest Group (RIG) of the APPN Implementers Workshop. This RIG is chaired by Louise Herndon-Wells (lhwells@cup.portal.com) and edited by Paul Brittain (pjb@datcon.co.uk).

Much of the work on the scalability enhancements for v2.0 was developed by Dave Bryant (3COM).

Other significant contributors to this document include:

Frank Bordonaro (Cisco)
Jon Houghton (IBM)
Steve Klein (IBM)
Ravi Periasamy (Cisco)
Mike Redden (Proteon)
Doug Wolff (3COM)

Many thanks also to all those who participated in the DLSw RIG sessions and mail exploder discussions.

If you would like to participate in future DLSw discussions, please subscribe to the DLSw RIG mailing lists by sending a mail to majordomo@raleigh.ibm.com specifying 'subscribe aiw-dlsw' as the body of the message.

If you would like further information on the activities of the AIW, please refer to the AIW web site at http://www.raleigh.ibm.com/app/aiwhome.htm.

15. Authors' Addresses

The editor of this document is:

Paul Brittain
Data Connection Ltd
Windsor House
Pepper Street
Chester
CH1 1DF
UK

tel: +44 1244 313440
email: pjb@datcon.co.uk

Much of the work on this document was created by:

David Bryant
3Com Corporation
5400 Bayfront Plaza MS 2418
Santa Clara, CA 95052

tel: (408) 764-5272
email: David_Bryant@3mail.3com.com

16.

16. Appendix - Clarifications to RFC 1795

This appendix attempts to clarify the areas of RFC 1795 that have proven to be ambiguous or hard to understand in the implementation experience to-date. These clarifications should be read in conjunction with RFC 1795 as this document does not reproduce the complete text of that RFC.

The clarifications are ordered by the section number in RFC 1795 to which they apply. Where one point applies to more than one place in RFC 1795, it is listed below by the first relevant section.

If any implementers encounter further difficulties in understanding RFC **1795 or these clarifications, they are encouraged to query the DLSw mail** exploder (see section 1.1) for assistance.

3. Send Port

It is not permitted for a DLSw implementation to check that the send port used by a partner is 2067. All implementations must accept connections from partners that do not use this port.

3 TCP Tunnel bringup

The paragraph below the figure should read as follows:

> Each Data Link Switch will maintain a list of DLSw capable routers and their status (active/inactive). Before Data Link Switching can occur between two routers, they must establish two TCP connections between them. These connections are treated as half duplex data pipes. A Data Link Switch will listen for incoming connections on its Read Port (2065), and initiate outgoing connections on its Write Port (2067). Each Switch is responsible for initiating one of the two TCP connections. After the TCP connections are established, SSP messages are exchanged to establish the capabilities of the two Data Link Switches. Once the exchange is complete, the DLSw will employ SSP control messages to establish end-to-end circuits over the transport connection. Within the transport connection, DLSw SSP messages are exchanged. The message formats and types for these SSP messages are documented in the following sections.

3.2 RII bit in SSP header MAC addresses

The RII bit in MAC addresses received from the LAN must be set to zero before forwarding in the source or destination address field in a SSP message header. This requirement aims to avoid ambiguity of circuit IDs. It is also recommended that all implementations ignore this bit in received SSP message headers.

3.3 Transport IDs

All implementations must allow for the DLSw peer varying the Transport ID up to and including when the ICR_cs message flows, and at all times reflect the most recent TID received from the partner in any SSP messages sent. The TID cannot vary once the ICR_cs message has flowed.

3.4 LF bits

LF-bits should be propagated from LAN to SSP to LAN (and back) as per a bridge (i.e. they can only be revised downwards at each step if required).

3.5 KEEPALIVE messages

The SSP KEEPALIVE message (x1D) uses the short ("infoframe") version of the SSP header. All DLSw implementation must support receipt and quiet ignore of this message, but there is not requirement to send it. There is no response to a KEEPALIVE message.

3.5 MAC header for Netbios SSP frames

The MAC header is included in forwarded SSP Netbios frames in the format described below:

- addresses are always in non-canonical format
- src/dest addresses are as per the LLC frame
- AC/FC bits may be reset and must be ignored
- SSAP, DSAP and command fields are included
- RII bit in src address is copied from the LLC frame
- the RIF length is not extended to include padding
- all RIFs are padded to 18 bytes so that the data is in a consistent place.

3.5,7 Unrecognized control vectors

All implementations should quietly ignore unrecognized control vectors in any SSP messages. In particular, unrecognized SSP frames or unrecognized fields in a CAPEX message should be quietly ignored without dropping the TCP connection.

5.4 Use of CUR-cs/CUR-ex

The SSAP and DSAP numbers in CUR_ex messages should reflect those actually used in the TEST (or equivalent) frame that caused the CUR_ex message to flow. This would mean that the SAP numbers in a 'typical' CUR_ex frame for SNA traffic switched from a LAN will be a source SAP of x04 and a destination SAP of x00.

The CUR_cs frame should only be sent when the DSAP is known. Specifically, CUR_ex should be used when a NULL XID is received that is targeted at DSAP zero, and CUR_cs when a XID specifying the (non-zero) DSAP is received.

Note that this does not mean that an implementation can assume that the DSAP on a CUR_ex will always be zero. The ICR_ex must always reflect the SSAP and DSAP values sent on the CUR_ex. This is still true even if an implementation always sends a TEST with DSAP = x00 on its local LAN(s) in response to a CUR_ex to any SAP.

An example of a situation where the CUR_ex may flow with a non-zero DSAP is when there is an APPN stack local to the DLSw node. The APPN stack may then issue a connection request specifying the DSAP as a non-zero value. This would then be passed on the CUR_ex message.

7.6.1 Vendor IDs

The Vendor ID field in a CAPEX may be zero. However, a zero Vendor Context ID is not permitted, which implies that an implementation that uses a zero ID cannot send any vendor-specific CVs (other than those specified by other vendors that do have a non-zero ID)

7.6.3 Initial Pacing Window

The initial pacing window may be 1. There is no requirement on an implementation to use any minimum value for the initial pacing window.

7.6.7 TCP Tunnel bringup

The third paragraph should read:

> If TCP Connections CV values agree and the number of connections is one, then the DLSw with the higher IP address must tear down the TCP connections on its local port 2065. This connection is torn down after a CAPEX response has been both sent and received. After this point, the remaining TCP connection is used to exchange data in both directions.

7.7 CAPEX negative responses

If a DLSw does not support any of the options specified on a CAPEX received from a partner, or if it thinks the CAPEX is malformed, it must send a CAPEX negative response to the partner. The receiver of a CAPEX negative response is then responsible for dropping the connection. It is not permitted to drop the link instead of sending a CAPEX negative response.

8.2 Flow Control ACKs

The first flow-control ack (FCACK) does not have to be returned on the REACH_ACK even if the ICR_cs carried the FCIND bit. However it should be returned on the first SSP frame flowing for that circuit after the REACH_ACK.