                   APPN/HPR in IP Networks
          APPN Implementers' Workshop Closed Pages Document

Status of this Memo

Copyright Notice

Table of Contents

1.0 Introduction

   The APPN Implementers' Workshop (AIW) is an industry-wide consortium
   of networking vendors that develops Advanced Peer-to-Peer
   Networking(R) (APPN(R)) standards and other standards related to
   Systems Network Architecture (SNA), and facilitates high quality,
   fully interoperable APPN and SNA internetworking products. The AIW
   approved Closed Pages (CP) status for the architecture in this
   document on December 2, 1997, and, as a result, the architecture was
   added to the AIW architecture of record. A CP-level document is
   sufficiently detailed that implementing products will be able to
   interoperate; it contains a clear and complete specification of all
   necessary changes to the architecture of record. However, the AIW
   has procedures by which the architecture may be modified, and the AIW
   is open to suggestions from the internet community.

   The architecture for APPN nodes is specified in "Systems Network
   Architecture Advanced Peer-to-Peer Networking Architecture Reference"
   [1]. A set of APPN enhancements for High Performance Routing (HPR)
   is specified in "Systems Network Architecture Advanced Peer-to-Peer
   Networking High Performance Routing Architecture Reference, Version
   3.0" [2]. The formats associated with these architectures are
   specified in "Systems Network Architecture Formats" [3]. This memo
   assumes the reader is familiar with these specifications.

   This memo defines a method with which HPR nodes can use IP networks
   for communication, and the enhancements to APPN required by this
   method. This memo also describes an option set that allows the use
   of the APPN connection network model to allow HPR nodes to use IP
   networks for communication without having to predefine link
   connections.

   (R) 'Advanced Peer-to-Peer Networking' and 'APPN' are trademarks of
   the IBM Corporation.

1.1 Requirements

   The following are the requirements for the architecture specified in
   this memo:

   1. Facilitate APPN product interoperation in IP networks by
       documenting agreements such as the choice of the logical link
       control (LLC).

   2. Reduce system definition (e.g., by extending the connection
       network model to IP networks) -- Connection network support is an
       optional function.

   3. Use class of service (COS) to retain existing path selection and
       transmission priority services in IP networks; extend
       transmission priority function to include IP networks.

   4. Allow customers the flexibility to design their networks for low
       cost and high performance.

   5. Use HPR functions to improve both availability and scalability
       over existing integration techniques such as Data Link Switching
       (DLSw) which is specified in RFC 1795 [4] and RFC 2166 [5].

2.0 IP as a Data Link Control (DLC) for HPR

   This memo specifies the use of IP and UDP as a new DLC that can be
   supported by APPN nodes with the three HPR option sets: HPR (option
   set 1400), Rapid Transport Protocol (RTP) (option set 1401), and
   Control Flows over RTP (option set 1402). Logical Data Link Control
   (LDLC) Support (option set 2006) is also a prerequisite.

   RTP is a connection-oriented, full-duplex protocol designed to
   transport data in high-speed networks. HPR uses RTP connections to
   transport SNA session traffic. RTP provides reliability (i.e., error
   recovery via selective retransmission), in-order delivery (i.e., a
   first-in-first-out [FIFO] service provided by resequencing data that
   arrives out of order), and adaptive rate-based (ARB) flow/congestion
   control. Because RTP provides these functions on an end-to-end basis,
   it eliminates the need for these functions on the link level along
   the path of the connection. The result is improved overall
   performance for HPR. For a more complete description of RTP, see
   Appendix F of [2].

   This new DLC (referred to as the native IP DLC) allows customers to
   take advantage of APPN/HPR functions such as class of service (COS)
   and ARB flow/congestion control in the IP environment. HPR links
   established over the native IP DLC are referred to as HPR/IP links.

The following sections describe in detail the considerations and
enhancements associated with the native IP DLC.

2.1 Use of UDP and IP

The native IP DLC will use the User Datagram Protocol (UDP) defined
in RFC 768 [6] and the Internet Protocol (IP) version 4 defined in
RFC 791 [7].

Typically, access to UDP is provided by a sockets API. UDP provides
an unreliable connectionless delivery service using IP to transport
messages between nodes. UDP has the ability to distinguish among
multiple destinations within a given node, and allows port-number-
based prioritization in the IP network. UDP provides detection of
corrupted packets, a function required by HPR. Higher-layer
protocols such as HPR are responsible for handling problems of
message loss, duplication, delay, out-of-order delivery, and loss of
connectivity. UDP is adequate because HPR uses RTP to provide end-
to-end error recovery and in-order delivery; in addition, LDLC
detects loss of connectivity. The Transmission Control Protocol
(TCP) was not chosen for the native IP DLC because the additional
services provided by TCP such as error recovery are not needed.
Furthermore, the termination of TCP connections would require
additional node resources (control blocks, buffers, timers, and
retransmit queues) and would, thereby, reduce the scalability of the
design.

The UDP header has four two-byte fields. The UDP Destination Port is
a 16-bit field that contains the UDP protocol port number used to
demultiplex datagrams at the destination. The UDP Source Port is a
16-bit field that contains the UDP protocol port number that
specifies the port to which replies should be sent when other
information is not available. A zero setting indicates that no
source port number information is being provided. When used with the
native IP DLC, this field is not used to convey a port number for
replies; moreover, the zero setting is not used. IANA has registered
port numbers 12000 through 12004 for use in these two fields by the
native IP DLC; use of these port numbers allows prioritization in the
IP network. For more details of the use of these fields, see 2.6.1,
"IP Prioritization" on page 28.

The UDP Checksum is a 16-bit optional field that provides coverage of
the UDP header and the user data; it also provides coverage of a
pseudo-header that contains the source and destination IP addresses.
The UDP checksum is used to guarantee that the data has arrived
intact at the intended receiver. When the UDP checksum is set to

zero, it indicates that the checksum was not calculated and should
not be checked by the receiver. Use of the checksum is recommended
for use with the native IP DLC.

IP provides an unreliable, connectionless delivery mechanism. The IP
protocol defines the basic unit of data transfer through the IP
network, and performs the routing function (i.e., choosing the path
over which data will be sent). In addition, IP characterizes how
"hosts" and "gateways" should process packets, the circumstances
under which error messages are generated, and the conditions under
which packets are discarded. An IP version 4 header contains an 8-
bit Type of Service field that specifies how the datagram should be
handled. As defined in RFC 1349 [8], the type-of-service byte
contains two defined fields. The 3-bit precedence field allows
senders to indicate the priority of each datagram. The 4-bit type of
service field indicates how the network should make tradeoffs between
throughput, delay, reliability, and cost. The 8-bit Protocol field
specifies which higher-level protocol created the datagram. When
used with the native IP DLC, this field is set to 17 which indicates
the higher-layer protocol is UDP.

2.2 Node Structure

Figure 1 on page 6 shows a possible node functional decomposition for
transport of HPR traffic across an IP network. There will be
variations in different platforms based on platform characteristics.

The native IP DLC includes a DLC manager, one LDLC component for each
link, and a link demultiplexor. Because UDP is a connectionless
delivery service, there is no need for HPR to activate and deactivate
lower-level connections.

The DLC manager activates and deactivates a link demultiplexor for
each port and an instance of LDLC for each link established in an IP
network. Multiple links (e.g., one defined link and one dynamic link
for connection network traffic) may be established between a pair of
IP addresses. Each link is identified by the source and destination
IP addresses in the IP header and the source and destination service
access point (SAP) addresses in the IEEE 802.2 LLC header (see 6.0,
"Appendix - Packet Format" on page 37); the link demultiplexor passes
incoming packets to the correct instance of LDLC based on these
identifiers. Moreover, the IP address pair associated with an active
link and used in the IP header may not change.

LDLC also provides other functions (for example, reliable delivery of
Exchange Identification [XID] commands). Error recovery for HPR RTP
packets is provided by the protocols between the RTP endpoints.

The network control layer (NCL) uses the automatic network routing
(ANR) information in the HPR network header to either pass incoming
packets to RTP or an outgoing link.

All components are shown as single entities, but the number of
logical instances of each is as follows:

o   DLC manager -- 1 per node

o   LDLC -- 1 per link

o   Link demultiplexor -- 1 per port

o   NCL -- 1 per node (or 1 per port for efficiency)

o   RTP -- 1 per RTP connection

o   UDP -- 1 per port

o   IP -- 1 per port

Products are free to implement other structures. Products
implementing other structures will need to make the appropriate
modifications to the algorithms and protocol boundaries shown in this
document.
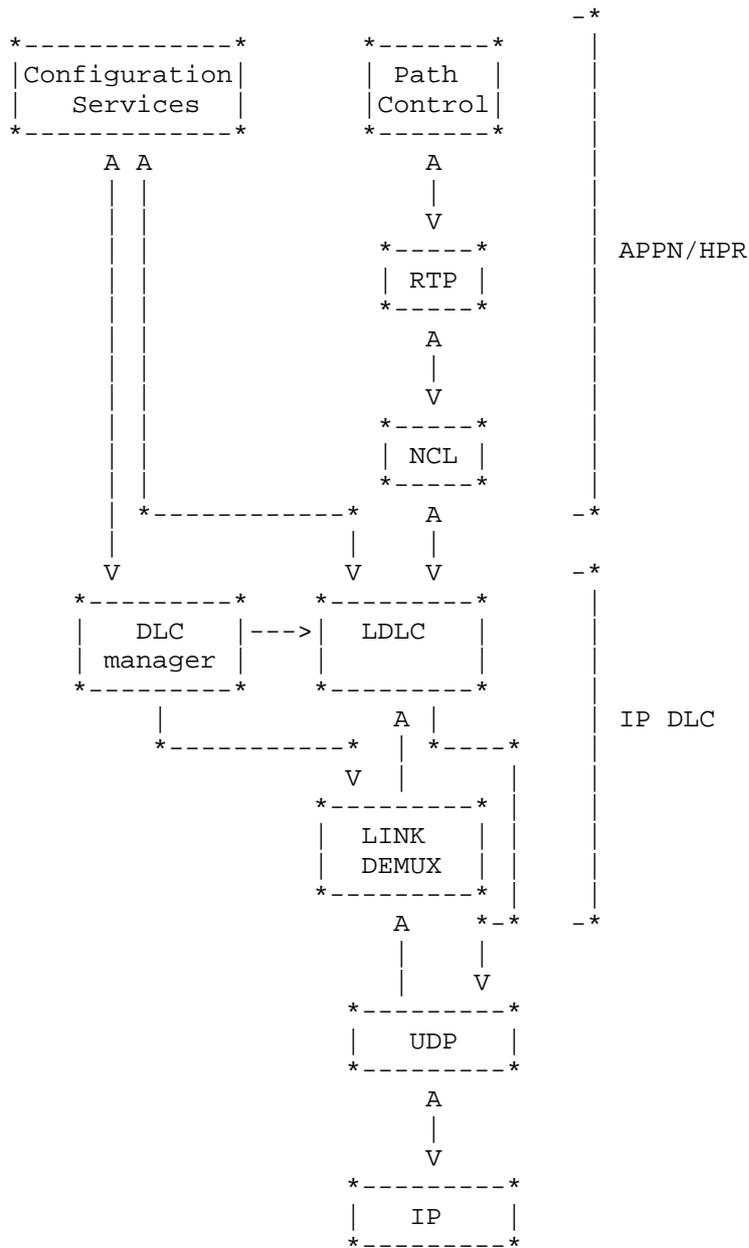
```
     ----------------------------------------------------------------------
                                              -*
     *-------------*          *-------*        |
     |Configuration|          | Path  |        |
     |  Services   |          |Control|        |
     *-------------*          *-------*        |
          A A                     A            |
          | |                     |            |
          | |                     V            |
          | |                  *-----*         |  APPN/HPR
          | |                  | RTP |         |
          | |                  *-----*         |
          | |                     A            |
          | |                     |            |
          | |                     V            |
          | |                  *-----*         |
          | |                  | NCL |         |
          | |                  *-----*         |
          | *-----------*         A         -*
          |             |         | |
          V             V         V V       -*
     *---------*    *---------*             |
     |   DLC   |--->|  LDLC   |             |
     | manager |    |         |             |
     *---------*    *---------*             |
          |             A |                 |  IP DLC
     *-----------*      | *----*            |
          V             |  |                |
     *---------*        |  |                |
     |  LINK   |        |  |                |
     | DEMUX   |        |  |                |
     *---------*        |  |                |
          A         *-*  -*
          |          |
          |          V
     *---------*
     |  UDP    |
     *---------*
          A
          |
          V
     *---------*
     |  IP     |
     *---------*

     ----------------------------------------------------------------------
                    Figure 1. HPR/IP Node Structure
```

2.3 Logical Link Control (LLC) Used for IP

   Logical Data Link Control (LDLC) is used by the native IP DLC. LDLC
   is defined in [2]. LDLC uses a subset of the services defined by
   IEEE 802.2 LLC type 2 (LLC2). LDLC uses only the TEST, XID, DISC,
   DM, and UI frames.

   LDLC was defined to be used in conjunction with HPR (with the HPR
   Control Flows over RTP option set 1402) over reliable links that do
   not require link-level error recovery. Most frame loss in IP
   networks (and the underlying frame networks) is due to congestion,
   not problems with the facilities. When LDLC is used on a link, no
   link-level error recovery is available; as a result, only RTP traffic
   is supported by the native IP DLC. Using LDLC eliminates the need
   for LLC2 and its associated cost (adapter storage, longer path
   length, etc.).

2.3.1 LDLC Liveness

   LDLC liveness (using the LDLC TEST command and response) is required
   when the underlying subnetwork does not provide notification of
   connection outage. Because UDP is connectionless, it does not
   provide outage notification; as a result, LDLC liveness is required
   for HPR/IP links.

   Liveness should be sent periodically on active links except as
   described in the following subsection when the option to reduce
   liveness traffic is implemented. The default liveness timer period
   is 10 seconds. When the defaults for the liveness timer and retry
   timer (15 seconds) are used, the period between liveness tests is
   smaller than the time required to detect failure (retry count
   multiplied by retry timer period) and may be smaller than the time
   for liveness to complete successfully (on the order of round-trip
   delay). When liveness is implemented as specified in the LDLC
   finite-state machine (see [2]) this is not a problem because the
   liveness protocol works as follows: The liveness timer is for a
   single link. The timer is started when the link is first activated
   and each time a liveness test completes successfully. When the timer
   expires, a liveness test is performed. When the link is operational,
   the period between liveness tests is on the order of the liveness
   timer period plus the round-trip delay.

   For each implementation, it is necessary to check if the liveness
   protocol will work in a satisfactory manner with the default settings
   for the liveness and retry timers. If, for example, the liveness
   timer is restarted immediately upon expiration, then a different
   default for the liveness timer should be used.

2.3.1.1 Option to Reduce Liveness Traffic

   In some environments, it is advantageous to reduce the amount of
   liveness traffic when the link is otherwise idle. (For example, this
   could allow underlying facilities to be temporarily deactivated when
   not needed.) As an option, implementations may choose not to send
   liveness when the link is idle (i.e., when data was neither sent nor
   received over the link while the liveness timer was running). (If
   the implementation is not aware of whether data has been received,
   liveness testing may be stopped while data is not being sent.)
   However, the RTP connections also have a liveness mechanism which
   will generate traffic. Some implementations of RTP will allow
   setting a large value for the ALIVE timer, thus reducing the amount
   of RTP liveness traffic.

   If LDLC liveness is turned off while the link is idle, one side of
   the link may detect a link failure much earlier than the other. This
   can cause the following problems:

   o   If a node that is aware of a link failure attempts to reactivate
        the link, the partner node (unaware of the link failure) may
        reject the activation as an unsupported parallel link between the
        two ports.

   o   If a node that is unaware of an earlier link failure sends data
        (including new session activations) on the link, it may be
        discarded by a node that detected the earlier failure and
        deactivated the link. As a result, session activations would
        fail.

   The mechanisms described below can be used to remedy these problems.
   These mechanisms are needed only in a node not sending liveness when
   the link is idle; thus, they would not be required of a node not
   implementing this option that just happened to be adjacent to a node
   implementing the option.

   o   (Mandatory unless the node supports multiple active defined links
        between a pair of HPR/IP ports and supports multiple active
        dynamic links between a pair of HPR/IP ports.) Anytime a node
        rejects the activation of an HPR/IP link as an unsupported
        parallel link between a pair of HPR/IP ports (sense data
        X'10160045' or X'10160046'), it should perform liveness on any
        active link between the two ports that is using a different SAP
        pair. Thus, if the activation was not for a parallel link but
        rather was a reactivation because one of these active links had
        failed, the failed link will be detected. (If the SAP pair for
        the link being activated matches the SAP pair for an active link,
        a liveness test would succeed because the adjacent node would

respond for the link being activated.) A simple way to implement
this function is for LDLC, upon receiving an activation XID, to
run liveness on all active links with a matching IP address pair
and a different SAP pair.

o   (Mandatory) Anytime a node receives an activation XID with an IP
    address pair and a SAP pair that match those of an active link,
    it should deactivate the active link and allow it to be
    reestablished. A timer is required to prevent stray XIDs from
    deactivating an active link.

o   (Recommended) A node should attempt to reactivate an HPR/IP link
    before acting on an LDLC-detected failure. This mechanism is
    helpful in preventing session activation failures in scenarios
    where the other side detected a link failure earlier, but the
    network has recovered.

2.4 IP Port Activation

The node operator (NO) creates a native IP DLC by issuing
DEFINE_DLC(RQ) (containing customer-configured parameters) and
START_DLC(RQ) commands to the node operator facility (NOF). NOF, in
turn, passes DEFINE_DLC(RQ) and START_DLC(RQ) signals to
configuration services (CS), and CS creates the DLC manager. Then,
the node operator can define a port by issuing DEFINE_PORT(RQ) (also
containing customer-configured parameters) to NOF with NOF passing
the associated signal to CS.

A node with adapters attached to multiple IP subnetworks may
represent the multiple adapters as a single HPR/IP port. However, in
that case, the node associates a single IP address with that port.
RFC 1122 [9] requires that a node with multiple adapters be able to
use the same source IP address on outgoing UDP packets regardless of
the adapter used for transmission.

```
   *-------------------------------------------*
   |  NOF                 CS                DLC |
   *-------------------------------------------*
        .  DEFINE_DLC(RQ)    .
   1    o---------------->o
        .  DEFINE_DLC(RSP)  |
   2    o<----------------*
        .  START_DLC(RQ)    .         create
   3    o---------------->o------------------>o
        .  START_DLC(RSP)   |                 .
   4    o<----------------*                   .
        .  DEFINE_PORT(RQ)  .                 .
   5    o---------------->o                   .
        .  DEFINE_PORT(RSP) |                 .
   6    o<----------------*                   .
```

                Figure 2. IP Port Activation

The following parameters are received in DEFINE_PORT(RQ):

o   Port name

o   DLC name

o   Port type (if IP connection networks are supported, set to shared
     access transport facility [SATF]; otherwise, set to switched)

o   Link station role (set to negotiable)

o   Maximum receive BTU size (default is 1461 [1492 less an allowance
     for the IP, UDP, and LLC headers])

o   Maximum send BTU size (default is 1461 [1492 less an allowance
     for the IP, UDP, and LLC headers])

o   Link activation limits (total, inbound, and outbound)

o   IPv4 supported (set to yes)

o   The local IPv4 address (required if IPv4 is supported)

o   IPv6 supported (set to no; may be set to yes in the future; see
     2.9, "IPv4-to-IPv6 Migration" on page 35)

o   The local IPv6 address (required if IPv6 is supported)

o   Retry count for LDLC (default is 3)

o   Retry timer period for LDLC (default is 15 seconds; a smaller
     value such as 10 seconds can be used for a campus network)

o   LDLC liveness timer period (default is 10 seconds; see 2.3.1,
     "LDLC Liveness" on page 7)

o   IP precedence (the setting of the 3-bit field within the Type of
     Service byte of the IP header for the LLC commands such as XID
     and for each of the APPN transmission priorities; the defaults
     are given in 2.6.1, "IP Prioritization" on page 28.)

2.4.1 Maximum BTU Sizes for HPR/IP

   When IP datagrams are larger than the underlying physical links
   support, IP performs fragmentation. When HPR/IP links are
   established, the default maximum basic transmission unit (BTU) sizes
   are 1461 bytes, which corresponds to the typical IP maximum
   transmission unit (MTU) size of 1492 bytes supported by routers on
   token-ring networks. 1461 is 1492 less 20 bytes for the IP header, 8
   bytes for the UDP header, and 3 bytes for the IEEE 802.2 LLC header.
   The IP header is larger than 20 bytes when optional fields are
   included; smaller maximum BTU sizes should be configured if optional
   IP header fields are used in the IP network. For IPv6, the default
   is reduced to 1441 bytes to allow for the typical IPv6 header size of
   40 bytes. Smaller maximum BTU sizes (but not less than 768) should
   be used to avoid fragmentation when necessary. Larger BTU sizes
   should be used to improve performance when the customer's IP network
   supports a sufficiently large IP MTU size. The maximum receive and
   send BTU sizes are passed to CS in DEFINE_PORT(RQ). These maximum
   BTU sizes can be overridden in DEFINE_CN_TG(RQ) or DEFINE_LS(RQ).

   The Flags field in the IP header should be set to allow
   fragmentation. Some products will not be able to control the setting
   of the bit allowing fragmentation; in that case, fragmentation will
   most likely be allowed. Although fragmentation is slow and prevents
   prioritization based on UDP port numbers, it does allow connectivity
   across paths with small MTU sizes.

2.5 IP Transmission Groups (TGs)

2.5.1 Regular TGs

   Regular HPR TGs may be established in IP networks using the native IP
   DLC architecture. Each of these TGs is composed of one or more
   HPR/IP links. Configuration services (CS) identifies the TG with the
   destination control point (CP) name and TG number; the destination CP

name may be configured or learned via XID, and the TG number, which
may be configured, is negotiated via XID. For auto-activatable
links, the destination CP name and TG number must be configured.

When multiple links (dynamic or defined) are established between a
pair of IP ports (each associated with a single IP address), an
incoming packet can be mapped to its associated link using the IP
address pair and the service access point (SAP) address pair. If a
node receives an activation XID for a defined link with an IP address
pair and a SAP pair that are the same as for an active defined link,
that node can assume that the link has failed and that the partner
node is reactivating the link. In such a case as an optimization,
the node receiving the XID can take down the active link and allow
the link to be reestablished in the IP network. Because UDP packets
can arrive out of order, implementation of this optimization requires
the use of a timer to prevent a stray XID from deactivating an active
link.

Support for multiple defined links between a pair of HPR/IP ports is
optional. There is currently no value in defining multiple HPR/IP
links between a pair of ports. In the future if HPR/IP support for
the Resource ReSerVation Protocol (RSVP) [10] is defined, it may be
advantageous to define such parallel links to segregate traffic by
COS on RSVP "sessions." Using RSVP, HPR would be able to reserve
bandwidth in IP networks. An HPR logical link would be mapped to an
RSVP "session" that would likely be identified by either a specific
application-provided UDP port number or a dynamically-assigned UDP
port number.

When multiple defined HPR/IP links between ports are not supported,
an incoming activation for a defined HPR/IP link may be rejected with
sense data X'10160045' if an active defined HPR/IP link already
exists between the ports. If the SAP pair in the activation XID
matches the SAP pair for the existing link, the optimization
described above may be used instead.

If parallel defined HPR/IP links between ports are not supported, an
incoming activation XID is mapped to the defined link station (if it
exists) associated with the port on the adjacent node using the
source IP address in the incoming activation XID. This source IP
address should be the same as the destination IP address associated
with the matching defined link station. (They may not be the same if
the adjacent node has multiple IP addresses, and the configuration
was not coordinated correctly.)

If parallel HPR/IP links between ports are supported, multiple
defined link stations may be associated with the port on the adjacent
node. In that case, predefined TG numbers (see "Partitioning the TG

Number Space" in Chapter 9 Configuration Services of [1]) may be used
to map the XID to a specific link station. However, because the same
TG characteristics may be used for all HPR/IP links between a given
pair of ports, all the link stations associated with the port in the
adjacent node should be equivalent; as a result, TG number
negotiation using negotiable TG numbers may be used.

In the future, if multiple HPR/IP links with different
characteristics are defined between a pair of ports using RSVP,
defined link stations will need sufficient configured information to
be matched with incoming XIDs. (Correct matching of an incoming XID
to a defined link station allows CS to provide the correct TG
characteristics to topology and routing services (TRS).) At that
time CS will do the mapping based on both the IP address of the
adjacent node and a predefined TG number.

The node initiating link activation knows which link it is
activating. Some parameters sent in prenegotiation XID are defined
in the regular link station configuration and not allowed to change
in following negotiation-proceeding XIDs. To allow for forward
migration to RSVP, when a regular TG is activated in an IP network,
the node receiving the first XID (i.e., the node not initiating link
activation) must also understand which defined link station is being
activated before sending a prenegotiation XID in order to correctly
set parameters that cannot change. For this reason, the node
initiating link activation will indicate the TG number in
prenegotiation XIDs by including a TG Descriptor (X'46') control
vector containing a TG Identifier (X'80') subfield. Furthermore, the
node receiving the first XID will force the node activating the link
to send the first prenegotiation XID by responding to null XIDs with
null XIDs. To prevent potential deadlocks, the node receiving the
first XID has a limit (the LDLC retry count can be used) on the
number of null XIDs it will send. Once this limit is reached, that
node will send an XID with an XID Negotiation Error (X'22') control
vector in response to a null XID; sense data X'0809003A' is included
in the control vector to indicate unexpected null XID. If the node
that received the first XID receives a prenegotiation XID without the
TG Identifier subfield, it will send an XID with an XID Negotiation
Error control vector to reject the link connection; sense data
X'088C4680' is included in the control vector to indicate the
subfield was missing.

For a regular TG, the TG parameters are provided by the node operator
based on customer configuration in DEFINE_PORT(RQ) and DEFINE_LS(RQ).
The following parameters are supplied in DEFINE_LS(RQ) for HPR/IP
links:

o   The destination IP host name (this parameter can usually be
     mapped to the destination IP address): If the link is not
     activated at node initialization, the IP host name should be
     mapped to an IP address, and the IP address should be stored with
     the link station definition. This is required to allow an
     incoming link activation to be matched with the link station
     definition. If the adjacent node activates the link with a
     different IP address (e.g., it could have multiple ports), it
     will not be possible to match the link activation with the link
     station definition, and the default parameters specified in the
     local port definition will be used.

o   The destination IP version (set to version 4, support for version
     6 may be required in the future; this parameter is only required
     if the address and version cannot be determined using the
     destination IP host name.)

o   The destination IP address (in the format specified by the
     destination IP version; this parameter is only required if the
     address cannot be determined using the destination IP host name.)

o   Source service access point address (SSAP) used for XID, TEST,
     DISC, and DM (default is X'04'; other values may be specified
     when multiple links between a pair of IP addresses are defined)

o   Destination service access point address (DSAP) used for XID,
     TEST, DISC, and DM (default is X'04')

o   Source service access point address (SSAP) used for HPR network
     layer packets (NLPs) (default is X'C8'; other values may be
     specified when multiple links between a pair of IP addresses are
     defined.)

o   Maximum receive BTU size (default is 1461; this parameter is used
     to override the setting in DEFINE_PORT.)

o   Maximum send BTU size (default is 1461; this parameter is used to
     override the setting in DEFINE_PORT.)

o   IP precedence (the setting of the 3-bit field within the Type of
     Service byte of the IP header for LLC commands such as XID and
     for each of the APPN transmission priorities; the defaults are
     given in 2.6.1, "IP Prioritization" on page 28; this parameter is
     used to override the settings in DEFINE_PORT)

o   Shareable with connection network traffic (default is yes for
     non-RSVP links)

o   Retry count for LDLC (default is 3; this parameter is used to
    override the setting in DEFINE_PORT)

o   Retry timer period for LDLC (default is 15 seconds; a smaller
    value such as 10 seconds can be used for a campus link; this
    parameter is used to override the setting in DEFINE_PORT)

o   LDLC liveness timer period (default is 10 seconds; this parameter
    is to override the setting in DEFINE_PORT; see 2.3.1, "LDLC ness"
    on page 7)

o   Auto-activation supported (default is no; may be set to yes when
    the local node has switched access to the IP network)

o   Limited resource (default is to set in concert with auto-
    activation supported)

o   Limited resource liveness timer (default is 45 sec.)

o   Port name

o   Adjacent CP name (optional)

o   Local CP-CP sessions supported

o   Defined TG number (optional)

o   TG characteristics

The following figures show the activation and deactivation of regular
TGs.

```
*-------------------------------------------------------------------*
|CS              DLC             LDLC            DMUX           UDP|
*-------------------------------------------------------------------*
 .                .                               .              .
 .CONNECT_OUT(RQ) .  create                       .              .
 o--------------->o-------------->o               .              .
 .                |       new LDLC                .              .
 .                o----------------------------->o               .
 CONNECT_OUT(+RSP)|                .              .              .
 o<--------------*                 .              .              .
 |           XID                   .          XID(CMD)          . XID
  *----------------------------->o----------------------------->o----->
```
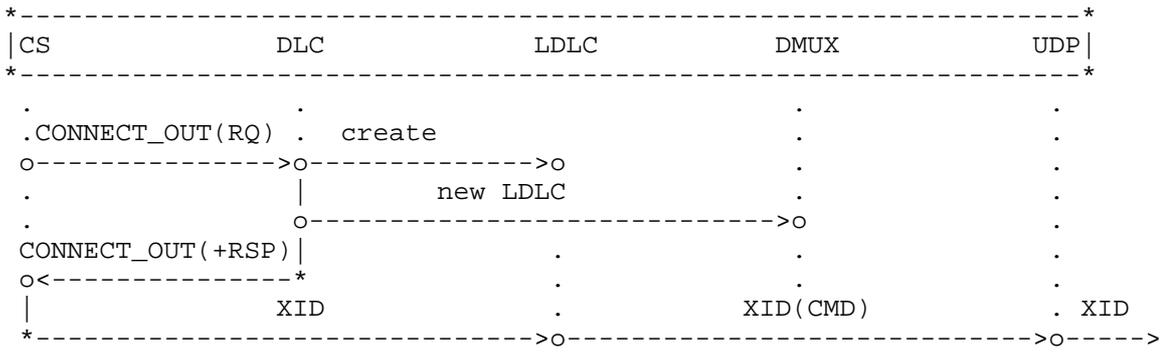
                  Figure 3. Regular TG Activation (outgoing)

   In Figure 3 upon receiving START_LS(RQ) from NOF, CS starts the link
   activation process by sending CONNECT_OUT(RQ) to the DLC manager.
   The DLC manager creates an instance of LDLC for the link, informs the
   link demultiplexor, and sends CONNECT_OUT(+RSP) to CS. Then, CS
   starts the activation XID exchange.

```
*-------------------------------------------------------------------*
|CS              DLC             LDLC            DMUX           UDP|
*-------------------------------------------------------------------*
 .                .                               .              .
 . CONNECT_IN(RQ) .           XID(CMD)            .    XID      . XID
 o<--------------o<---------------------------o<-------------o<-----
 | CONNECT_IN(RSP).   create                    .              .
 *-------------->o-------------->o               .              .
 .                |       new LDLC               .              .
 .                o----------------------------->o               .
 .                | XID(CMD)       .              .              .
 .                *-------------->o               .              .
 .               XID              |               .              .
 o<----------------------------*                 .              .
 |           XID                   .          XID(RSP)          . XID
  *----------------------------->o----------------------------->o----->
```
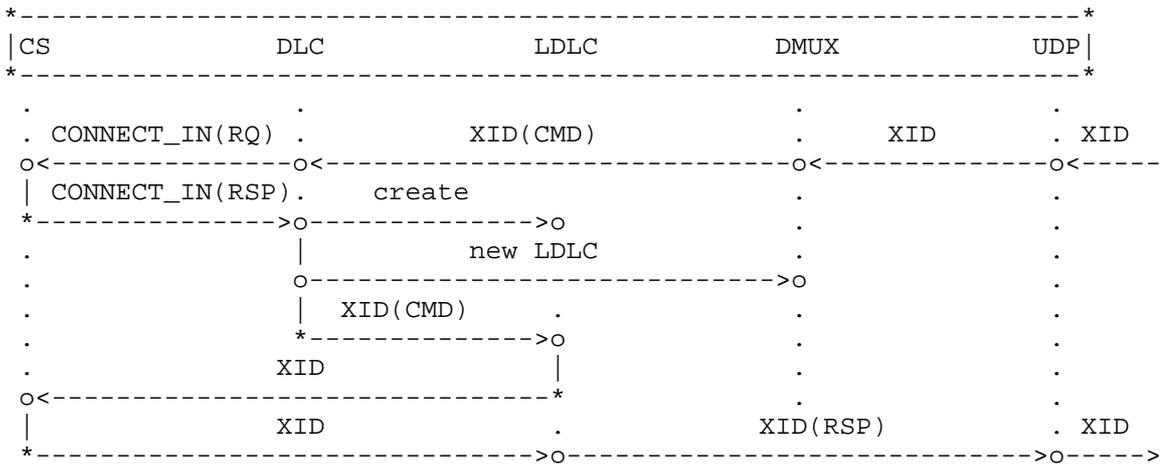
                  Figure 4. Regular TG Activation (incoming)

   In Figure 4, when an XID is received for a new link, it is passed to
   the DLC manager. The DLC manager sends CONNECT_IN(RQ) to notify CS
   of the incoming link activation, and CS sends CONNECT_IN(+RSP)
   accepting the link activation. The DLC manager then creates a new
   instance of LDLC, informs the link demultiplexor, and forwards the
   XID to to CS via LDLC. CS then responds by sending an XID to the
   adjacent node.

The two following figures show normal TG deactivation (outgoing and
incoming).

```
*---------------------------------------------------------------------*
|CS               DLC              LDLC            DMUX           UDP|
*---------------------------------------------------------------------*
 .                 .                .                .               .
 .               DEACT              .              DISC            . DISC
 o------------------------------>o------------------------------->o----->
 .               DEACT              .     DM         .     DM      . DM
 o<-----------------------------o<------------o<-------------o<-----
 | DISCONNECT(RQ) .    destroy   .                .               .
 *-------------->o------------->o                .               .
  DISCONNECT(RSP) |                              .               .
 o<--------------*                               .               .
```
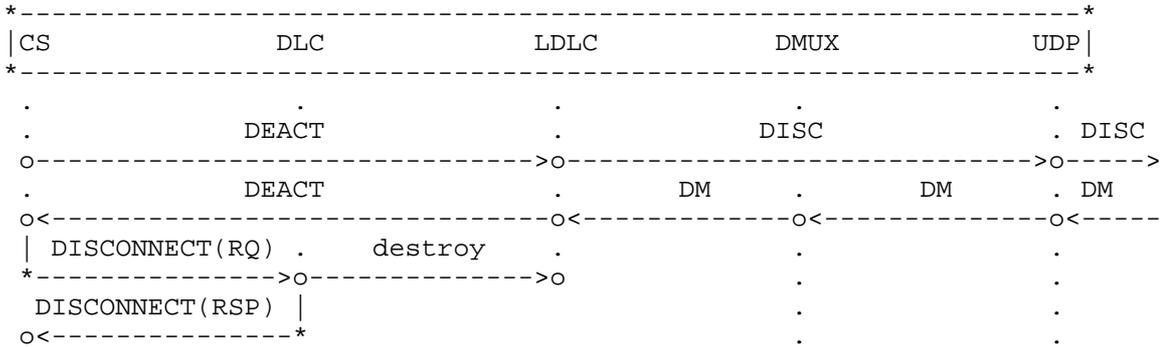
                 Figure 5. Regular TG Deactivation (outgoing)

   In Figure 5 upon receiving STOP_LS(RQ) from NOF, CS sends DEACT to
   notify the partner node that the HPR link is being deactivated. When
   the response is received, CS sends DISCONNECT(RQ) to the DLC manager,
   and the DLC manager deactivates the instance of LDLC. Upon receiving
   DISCONNECT(RSP), CS sends STOP_LS(RSP) to NOF.

```
*---------------------------------------------------------------------*
|CS               DLC              LDLC            DMUX           UDP|
*---------------------------------------------------------------------*
 .                 .                .                .               .
 .               DEACT              .     DISC       .     DISC    . DISC
 o<-----------------------------o<------------o<-------------o<-----
 |                 .                |               DM         . DM
 |                 .                *----------------------------->o----->
 | DISCONNECT(RQ) .    destroy     .                .               .
 *-------------->o------------->o                .               .
 .DISCONNECT(RSP) |                              .               .
 o<--------------*                               .               .
```
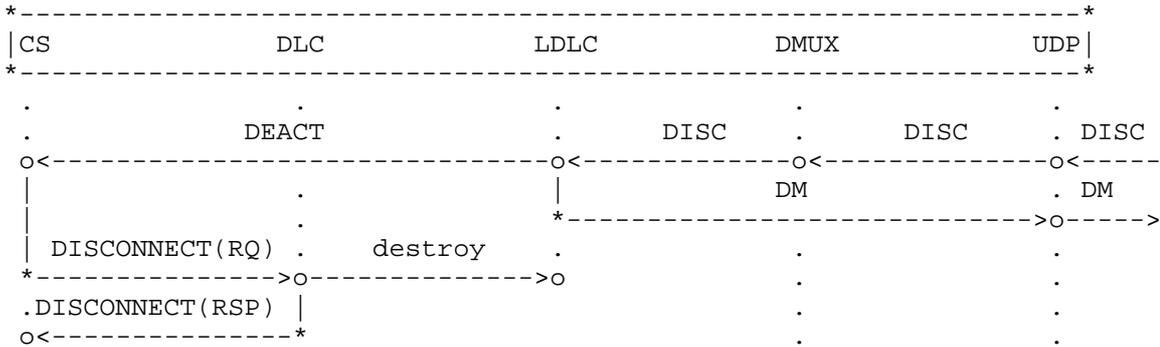
                 Figure 6. Regular TG Deactivation (incoming)

   In Figure 6, when an adjacent node deactivates a TG, the local node
   receives a DISC. CS sends STOP_LS(IND) to NOF. Because IP is
   connectionless, the DLC manager is not aware that the link has been
   deactivated. For that reason, CS also needs to send DISCONNECT(RQ)
   to the DLC manager; the DLC manager deactivates the instance of LDLC.

2.5.1.1 Limited Resources and Auto-Activation

   To reduce tariff charges, the APPN architecture supports the
   definition of switched links as limited resources. A limited-
   resource link is deactivated when there are no sessions traversing
   the link. Intermediate HPR nodes are not aware of sessions between
   logical units (referred to as LU-LU sessions) carried in crossing RTP
   connections; in HPR nodes, limited-resource TGs are deactivated when
   no traffic is detected for some period of time. Furthermore, APPN
   links may be defined as auto-activatable. Auto-activatable links are
   activated when a new session has been routed across the link.

   An HPR node may have access to an IP network via a switched access
   link. In such environments, it may be advisable for customers to
   define regular HPR/IP links as limited resources and as being auto-
   activatable.

2.5.2 IP Connection Networks

   Connection network support for IP networks (option set 2010), is
   described in this section.

   APPN architecture defines single link TGs across the point-to-point
   lines connecting APPN nodes. The natural extension of this model
   would be to define a TG between each pair of nodes connected to a
   shared access transport facility (SATF) such as a LAN or IP network.
   However, the high cost of the system definition of such a mesh of TGs
   is prohibitive for a network of more than a few nodes. For that
   reason, the APPN connection network model was devised to reduce the
   system definition required to establish TGs between APPN nodes.

   Other TGs may be defined through the SATF which are not part of the
   connection network. Such TGs (referred to as regular TGs in this
   document) are required for sessions between control points (referred
   to as CP-CP sessions) but may also be used for LU-LU sessions.

   In the connection network model, a virtual routing node (VRN) is
   defined to represent the SATF. Each node attached to the SATF
   defines a single TG to the VRN rather than TGs to all other attached
   nodes.

   Topology and routing services (TRS) specifies that a session is to be
   routed between two nodes across a connection network by including the
   connection network TGs between each of those nodes and the VRN in the
   Route Selection control vector (RSCV). When a network node has a TG
   to a VRN, the network topology information associated with that TG
   includes DLC signaling information required to establish connectivity
   to that node across the SATF. For an end node, the DLC signaling

information is returned as part of the normal directory services (DS)
process. TRS includes the DLC signaling information for TGs across
connection networks in RSCVs.

CS creates a dynamic link station when the next hop in the RSCV of an
ACTIVATE_ROUTE signal received from session services (SS) is a
connection network TG or when an adjacent node initiates link
activation upon receiving such an ACTIVATE_ROUTE signal. Dynamic
link stations are normally treated as limited resources, which means
they are deactivated when no sessions are using them. CP-CP sessions
are not supported on connections using dynamic link stations because
CP-CP sessions normally need to be kept up continuously.

Establishment of a link across a connection network normally requires
the use of CP-CP sessions to determine the destination IP address.
Because CP-CP sessions must flow across regular TGs, the definition
of a connection network does not eliminate the need to define regular
TGs as well.

Normally, one connection network is defined on a LAN (i.e., one VRN
is defined.) For an environment with several interconnected campus
IP networks, a single wide-area connection network can be defined; in
addition, separate connection networks can be defined between the
nodes connected to each campus IP network.

2.5.2.1 Establishing IP Connection Networks

Once the port is defined, a connection network can be defined on the
port. In order to support multiple TGs from a port to a VRN, the
connection network is defined by the following process:

1. A connection network and its associated VRN are defined on the
    port. This is accomplished by the node operator issuing a
    DEFINE_CONNECTION_NETWORK(RQ) command to NOF and NOF passing a
    DEFINE_CN(RQ) signal to CS.

2. Each TG from the port to the VRN is defined by the node operator
    issuing DEFINE_CONNECTION_NETWORK_TG(RQ) to NOF and NOF passing
    DEFINE_CN_TG(RQ) to CS.

Prior to implementation of Resource ReSerVation Protocol (RSVP)
support, only one connection network TG between a port and a VRN is
required. In that case, product support for the DEFINE_CN_TG(RQ)
signal is not required because a single set of port configuration
parameters for each connection network is sufficient. If a NOF
implementation does not support DEFINE_CN_TG(RQ), the parameters
listed in the following section for DEFINE_CN_TG(RQ), are provided by
DEFINE_CN(RQ) instead. Furthermore, the Connection Network TG

Numbers (X'81') subfield in the TG Descriptor (X'46') control vector
on an activation XID is only required to support multiple connection
network TGs to a VRN, and its use is optional.

```
  *-------------------------------------------------------*
  |   NO                      NOF                   CS    |
  *-------------------------------------------------------*
     DEFINE_CONNECTION_NETWORK(RQ)    DEFINE_CN(RQ)   .
        o----------------------->o----------------->o
     DEFINE_CONNECTION_NETWORK(RSP)   DEFINE_CN(RSP) |
        o<-----------------------o<-----------------*
   DEFINE_CONNECTION_NETWORK_TG(RQ) DEFINE_CN_TG(RQ)  .
        o----------------------->o----------------->o
  DEFINE_CONNECTION_NETWORK_TG(RSP) DEFINE_CN_TG(RSP)|
        o<-----------------------o<-----------------*
```
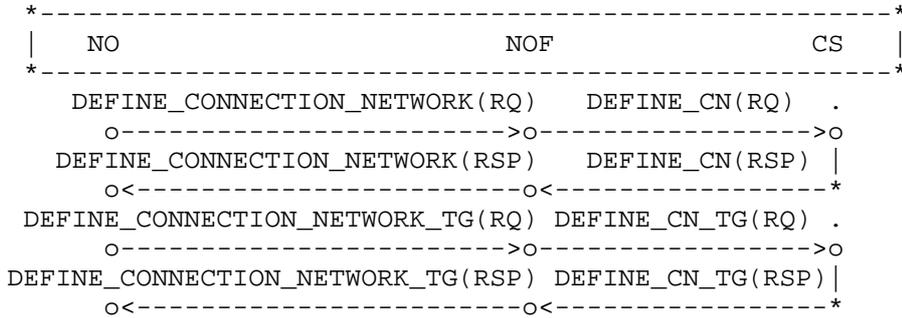
             Figure 7. IP Connection Network Definition

An incoming dynamic link activation may be rejected with sense data
X'10160046' if there is an existing dynamic link between the two
ports over the same connection network (i.e., with the same VRN CP
name). If a node receives an activation XID for a dynamic link with
an IP address pair, a SAP pair, and a VRN CP name that are the same
as for an active dynamic link, that node can assume that the link has
failed and that the partner node is reactivating the link. In such a
case as an optimization, the node receiving the XID can take down the
active link and allow the link to be reestablished in the IP network.
Because UDP packets can arrive out of order, implementation of this
optimization requires the use of a timer to prevent a stray XID from
deactivating an active link.

Once all the connection networks are defined, the node operator
issues START_PORT(RQ), NOF passes the associated signal to CS, and CS
passes ACTIVATE_PORT(RQ) to the DLC manager. Upon receiving the
ACTIVATE_PORT(RSP) signal from the DLC manager, CS sends a TG_UPDATE
signal to TRS for each defined connection network TG. Each signal
notifies TRS that a TG to the VRN has been activated and includes TG
vectors describing the TG. If the port fails or is deactivated, CS
sends TG_UPDATE indicating the connection network TGs are no longer
operational. Information about TGs between a network node and the
VRN is maintained in the network topology database. Information
about TGs between an end node and the VRN is maintained only in the
local topology database. If TRS has no node entry in its topology
database for the VRN, TRS dynamically creates such an entry. A VRN
node entry will become part of the network topology database only if

a network node has defined a TG to the VRN; however, TRS is capable
of selecting a direct path between two end nodes across a connection
network without a VRN node entry.

```
*----------------------------------------------------------------------*
|  CS                     TRS                 DLC              DMUX  |
*----------------------------------------------------------------------*
      .            ACTIVATE_PORT(RQ)           .      create
      o--------------------------------------->o---------------->o
      .            ACTIVATE_PORT(RSP)          |                 .
      o<--------------------------------------*                  .
      |  TG_UPDATE            .                .                 .
      *------------------->o                   .                 .
      .                     .                  .                 .
```
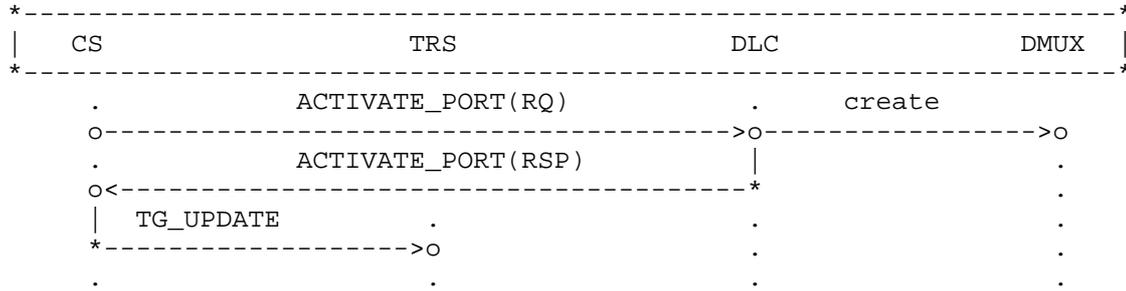
          Figure 8. IP Connection Network Establishment

The TG vectors for IP connection network TGs include the following
information:

   o   TG number

   o   VRN CP name

   o   TG characteristics used during route selection

        -    Effective capacity
        -    Cost per connect time
        -    Cost per byte transmitted
        -    Security
        -    Propagation delay
        -    User defined parameters

   o   Signaling information

        -    IP version (indicates the format of the IP header including
             the IP address)

        -    IP address

        -    Link service access point address (LSAP) used for XID, TEST,
             DISC, and DM

2.5.2.2 IP Connection Network Parameters

   For a connection network TG, the parameters are determined by CS
   using several inputs. Parameters that are particular to the local
   port, connection network, or TG are system defined and received in

DEFINE_PORT(RQ), DEFINE_CN(RQ), or DEFINE_CN_TG(RQ). Signaling
information for the destination node including its IP address is
received in the ACTIVATE_ROUTE request from SS.

The following configuration parameters are received in DEFINE_CN(RQ):

o   Connection network name (CP name of the VRN)

o   Limited resource liveness timer (default is 45 sec.)

o   IP precedence (the setting of the 3-bit field within the Type of
     Service byte of the IP header for LLC commands such as XID and
     for each of the APPN transmission priorities; the defaults are
     given in 2.6.1, "IP Prioritization" on page 28; this parameter is
     used to override the settings in DEFINE_PORT)

The following configuration parameters are received in
DEFINE_CN_TG(RQ):

o   Port name

o   Connection network name (CP name of the VRN)

o   Connection network TG number (set to a value between 1 and 239)

o   TG characteristics (see 2.6.3, "Default TG Characteristics" on
     page 30)

o   Link service access point address (LSAP) used for XID, TEST,
     DISC, and DM (default is X'04')

o   Link service access point address (LSAP) used for HPR network
     layer packets (default is X'C8')

o   Limited resource (default is yes)

o   Retry count for LDLC (default is 3; this parameter is used to
     override the setting in DEFINE_PORT)

o   Retry timer period for LDLC (default is 15 sec.; a smaller value
     such as 10 seconds can be used for a campus connection network;
     this parameter is used to override the setting in DEFINE_PORT)

o   LDLC liveness timer period (default is 10 seconds; this parameter
     is used to override the setting in DEFINE_PORT; see 2.3.1, "LDLC
     Liveness" on page 7)

o   Shareable with other HPR traffic (default is yes for non-RSVP
    links)

o   Maximum receive BTU size (default is 1461; this parameter is used
    to override the value in DEFINE_PORT(RQ).)

o   Maximum send BTU size (default is 1461; this parameter is used to
    override the value in DEFINE_PORT(RQ).)

The following parameters are received in ACTIVATE_ROUTE for
connection network TGs:

o   The TG pair

o   The destination IP version (if this version is not supported by
    the local node, the ACTIVATE_ROUTE_RSP reports the activation
    failure with sense data X'086B46A5'.)

o   The destination IP address (in the format specified by the
    destination IP version)

o   Destination service access point address (DSAP) used for XID,
    TEST, DISC, and DM

2.5.2.3 Sharing of TGs

Connection network traffic is multiplexed onto a regular defined IP
TG (usually used for CP-CP session traffic) in order to reduce the
control block storage. No XIDs flow to establish a new TG on the IP
network, and no new LLC is created. When a regular TG is shared,
incoming traffic is demultiplexed using the normal means. If the
regular TG is deactivated, a path switch is required for the HPR
connection network traffic sharing the TG.

Multiplexing is possible if the following conditions hold:

1. Both the regular TG and the connection network TG to the VRN are
    defined as shareable between HPR traffic streams.

2. The destination IP address is the same.

3. The regular TG is established first. (Because links established
    for connection network traffic do not support CP-CP sessions,
    there is little value in allowing a regular TG to share such a
    link.)

The destination node is notified via XID when a TG can be shared
between HPR data streams. At either end, upon receiving

ACTIVATE_ROUTE requesting a shared TG for connection network traffic,
CS checks its TGs for one meeting the required specifications before
initiating a new link. First, CS looks for a link established for
the TG pair; if there is no such link, CS determines if there is a
regular TG that can be shared and, if multiple such TGs exist, which
TG to choose. As a result, RTP connections routed over the same TG
pair may actually use different links, and RTP connections routed
over different TG pairs may use the same link.

2.5.2.4 Minimizing RSCV Length

The maximum length of a Route Selection (X'2B') control vector (RSCV)
is 255 bytes. Use of connection networks significantly increases the
size of the RSCV contents required to describe a "hop" across an
SATF. First, because two connection network TGs are used to specify
an SATF hop, two TG Descriptor (X'46') control vectors are required.
Furthermore, inclusion of DLC signaling information within the TG
Descriptor control vectors increases the length of these control
vectors. As a result, the total number of hops that can be specified
in RSCVs traversing connection networks is reduced.

To avoid unnecessarily limiting the number of hops, a primary goal in
designing the formats for IP signaling information is to minimize
their size. Additional techniques are also used to reduce the effect
of the RSCV length limitation.

For an IP connection network, DLC signaling information is required
only for the second TG (i.e., from the VRN to the destination node);
the signaling information for the first TG is locally defined at the
origin node. For this reason, the topology database does not include
DLC signaling information for the entry describing a connection
network TG from a network node to a VRN. The DLC signaling
information is included in the allied entry for the TG in the
opposite direction. This mechanism cannot be used for a connection
network TG between a VRN and an end node. However, a node
implementing IP connection networks does not include IP signaling
information for the first connection network TG when constructing an
RSCV.

In an environment where APPN network nodes are used to route between
legacy LANs and wide-area IP networks, it is recommended that
customers not define connection network TGs between these network
nodes and VRNs representing legacy LANs. Typically, defined links
are required between end nodes on the legacy LANs and such network
nodes which also act as network node servers for the end nodes.
These defined links can be used for user traffic as well as control
traffic. This technique will reduce the number of connection network
hops in RSCVs between end nodes on different legacy LANs.

Lastly, for environments where RSCVs are still not able to include
enough hops, extended border nodes (EBNs) can be used to partition
the network. In this case, the EBNs will also provide piecewise
subnet route calculation and RSCV swapping. Thus, the entire route
does not need to be described in a single RSCV with its length
limitation.

2.5.3 XID Changes

Packets transmitted over IP networks are lost or arrive out of order
more often than packets transmitted over other "link" technologies.
As a result, the following problem with the XID3 negotiation protocol
was exposed:
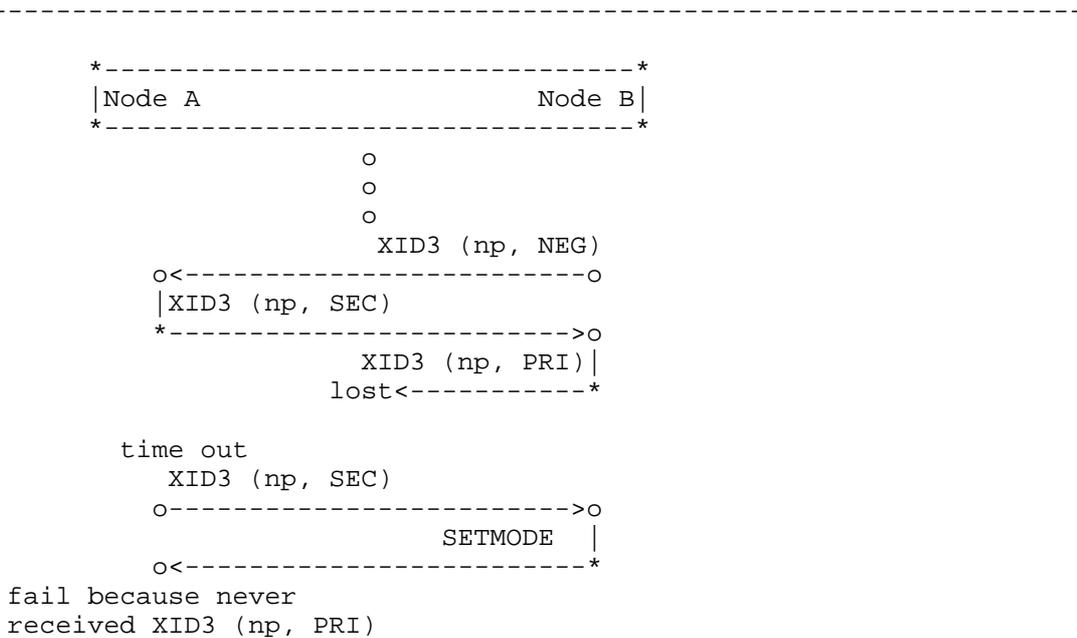
```
   -------------------------------------------------------------------

       *------------------------------*
       |Node A                  Node B|
       *------------------------------*
                       o
                       o
                       o
                        XID3 (np, NEG)
             o<------------------------o
             |XID3 (np, SEC)
             *------------------------>o
                        XID3 (np, PRI)|
                       lost<----------*

          time out
            XID3 (np, SEC)
             o----------------------->o
                             SETMODE  |
             o<-----------------------*
        fail because never
        received XID3 (np, PRI)

     Notation: np  - negotiation proceeding
               NEG - negotiable link station role
               SEC - secondary link station role
               PRI - primary link station role

   -------------------------------------------------------------------
                   Figure 9. XID3 Protocol Problem
```

In the above sequence, the XID3(np, PRI), which is a link-level
response to the received XID3(np, SEC), is lost. Node A times out
and resends the XID3(np, SEC) as a link-level command. When Node B

receives this command, it thinks that the XID3(np, PRI) was
successfully received by Node A and that the activation XID exchange
is complete. As a result, Node B sends SETMODE (SNRM, SABME, or
XID_DONE_RQ, depending upon the link type). When Node A receives
SETMODE, it fails the link activation because it has not received an
XID3(np, PRI) from Node B confirming that Node B does indeed agree to
be the primary. Moreover, there are similar problems with incomplete
TG number negotiation.

To solve the problems with incomplete role and TG number negotiation,
two new indicators are defined in XID3. The problems are solved only
if both link stations support these new indicators:

o   Negotiation Complete Supported indicator (byte 12 bit 0) -- this
    1-bit field indicates whether the Negotiation Complete indicator
    is supported. This field is meaningful when the XID exchange
    state is negotiation proceeding; otherwise, it is reserved. A
    value of 0 means the Negotiation Complete indicator is not
    supported; a value of 1 means the indicator is supported.

o   Negotiation Complete indicator (byte 12 bit 1) -- this 1-bit
    field is meaningful only when the XID exchange state is
    negotiation proceeding, the XID3 is sent by the secondary link
    station, and the Negotiation Complete Supported indicator is set
    to 1; otherwise, this field is reserved. This field is set to 1
    by a secondary link station that supports enhanced XID
    negotiation when it considers the activation XID negotiation to
    be complete for both link station role and TG number (i.e., it is
    ready to receive a SETMODE command from the primary link
    station.)

When a primary link station that supports enhanced XID negotiation
receives an XID3(np) with both the Negotiation Complete Supported
indicator and the Negotiation Complete indicator set to 1, the
primary link station will know that it can safely send SETMODE if it
also considers the XID negotiation to be complete. The new
indicators are used as shown in the following sequence when both the
primary and secondary link stations support enhanced XID negotiation.

```
        -----------------------------------------------------------------

            *-------------------------------*
            |Node A                  Node B |
            *-------------------------------*
                            o
                            o
                            o
                    XID3 (np, NEG, S, ^C)
    1           o<-----------------------o
                |XID3 (np, SEC, S, ^C)
    2           *----------------------->o
                    XID3 (np, PRI, S, ^C)|
    3                   lost <-----------*

            time out
                XID3 (np, SEC, S, ^C)
    4           o----------------------->o
                    XID3 (np, PRI, S, ^C)|
    5           o<----------------------*
                |XID3 (np, SEC, S, C)
    6           *----------------------->o
                                SETMODE  |
    7           o<----------------------*


    ^S indicates that byte 12 bit 0 is set to 0.
     S indicates that byte 12 bit 0 is set to 1.
    ^C indicates that byte 12 bit 1 is set to 0.
     C indicates that byte 12 bit 1 is set to 1.


        ----------------------------------------------------------------
```
Figure 10. Enhanced XID Negotiation

When Node B receives the XID in flow 4, it realizes that the Node A
does not consider XID negotiation to be complete; as a result, it
resends its current XID information in flow 5. When Node A receives
this XID, it responds in flow 6 with an XID that indicates XID
negotiation is complete. At this point, Node B, acting as the
primary link station, sends SETMODE, and the link is activated
successfully.

Migration cases with only one link station supporting enhanced XID
negotiation are shown in the two following sequences. In the next
sequence, only Node A (acting as the secondary link station) supports
the new function.

```
         -------------------------------------------------------------------

           *------------------------------*
           |Node A                  Node B|
           *------------------------------*
                         o
                         o
                         o
                      XID3 (np, NEG, ^S)
      1          o<------------------------o
                 |XID3 (np, SEC, S, ^C)
      2          *------------------------>o
                      XID3 (np, PRI, ^S)|
      3               lost <-----------*

             time out
               XID3 (np, SEC, S, ^C)
      4          o------------------------>o
                               SETMODE  |
      5          o<------------------------*
             fail


         ------------------------------------------------------------------
                       Figure 11. First Migration Case
```

The XID negotiation fails because Node B does not understand the new indicators and responds to flow 4 with SETMODE.

In the next sequence, Node B supports the new indicators but Node A does not.

```
     ------------------------------------------------------------------

          *-------------------------------*
          |Node A                   Node B|
          *-------------------------------*
                         o
                         o
                         o
                  XID3 (np, NEG, S, ^C)
     1         o<------------------------o
               |XID3 (np, SEC, ^S)
     2         *------------------------>o
                  XID3 (np, PRI, S, ^C)|
     3                  lost <----------*

            time out
              XID3 (np, SEC, ^S)
     4         o------------------------>o
                                 SETMODE |
     5         o<------------------------*
            fail


     ------------------------------------------------------------------
                    Figure 12. Second Migration Case
```

The XID negotiation fails because Nobe A does not understand the new
indicators and thus cannot indicate that it thinks XID negotiation is
not complete in flow 4. Node B understands that the secondary link
station (node A) does not support the new indicators and respond with
SETMODE in flow 5.

Products that support HPR/IP links are required to support enhanced
XID negotiation. Moreover, it is recommended that products
implementing this solution for HPR/IP links also support it for other
link types.

2.5.4 Unsuccessful IP Link Activation

Link activation may fail for several different reasons. When link
activation over a connection network or of an auto-activatable link
is attempted upon receiving ACTIVATE_ROUTE from SS, activation
failure is reported with ACTIVATE_ROUTE_RSP containing sense data
explaining the cause of failure. Likewise, when activation fails for
other regular defined links, the failure is reported with
START_LS(RSP) containing sense data.

   As is normal for session activation failures, the sense data is also
   sent to the node that initiated the session. At the APPN-to-HPR
   boundary, a -RSP(BIND) or an UNBIND with an Extended Sense Data
   control vector is generated and returned to the primary logical unit
   (PLU).

   At an intermediate HPR node, link activation failure can be reported
   with sense data X'08010000' or X'80020000'. At a node with route-
   selection responsibility, such failure can be reported with sense
   data X'80140001'.

   The following table contains the sense data for the various causes of
   link activation failure:

ERROR DESCRIPTION	SENSE DATA
Table 1 (Page 1 of 2). Native IP DLC Link Activation Failure Sense Data	
The link specified in the RSCV is not available.	X'08010000'
The limit for null XID responses by a called node was reached.	X'0809003A'
A BIND was received over a subarea link, but the next hop is over a port that supports only HPR links. The receiver does not support this configuration.	X'08400002'
The contents of the DLC Signaling Type (X'91') subfield of the TG Descriptor (X'46') control vector contained in the RSCV were invalid.	X'086B4691'
The contents of the IP Address and Link Service Access Point Address (X'A5') subfield of the TG Descriptor (X'46') control vector contained in the RSCV were invalid.	X'086B46A5'
No DLC Signaling Type (X'91') subfield was found in the TG Descriptor (X'46') control vector contained in the RSCV.	X'086D4691'
No IP Address and Link Service Access Point Address (X'A5') subfield was found in the TG Descriptor (X'46') control vector contained in the RSCV.	X'086D46A5'
Multiple sets of DLC signaling information were found in the TG Descriptor (X'46') control vector contained in the RSCV. IP supports only one set of DLC signaling information.	X'08770019'
Link Definition Error: A link is defined as not supporting HPR, but the port only supports HPR links.	X'08770026'
A called node found no TG Identifier (X'80') subfield within a TG Descriptor (X'46') control vector in a prenegotiation XID for a defined link in an IP network.	X'088C4680'

```
+-----------------------------------------------------------------------+
| Table 1 (Page 2 of 2). Native IP DLC Link Activation Failure Sense    |
|                        Data                                           |
+----------------------------------------------------------+------------+
| The XID3 received from the adjacent node does not        | X'10160031'|
| contain an HPR Capabilities (X'61') control vector.      |            |
| The IP port supports only HPR links.                     |            |
+----------------------------------------------------------+------------+
| The RTP Supported indicator is set to 0 in the HPR       | X'10160032'|
| Capabilities (X'61') control vector of the XID3          |            |
| received from the adjacent node.  The IP port supports   |            |
| only links to nodes that support RTP.                    |            |
+----------------------------------------------------------+------------+
| The Control Flows over RTP Supported indicator is set    | X'10160033'|
| to 0 in the HPR Capabilities (X'61') control vector of   |            |
| the XID3 received from the adjacent node.  The IP port   |            |
| supports only links to nodes that support control        |            |
| flows over RTP.                                          |            |
+----------------------------------------------------------+------------+
| The LDLC Supported indicator is set to 0 in the HPR      | X'10160034'|
| Capabilities (X'61') control vector of the XID3          |            |
| received from the adjacent node.  The IP port supports   |            |
| only links to nodes that support LDLC.                   |            |
+----------------------------------------------------------+------------+
| The HPR Capabilities (X'61') control vector received     | X'10160044'|
| in XID3 does not include an IEEE 802.2 LLC (X'80') HPR   |            |
| Capabilities subfield.  The subfield is required on an   |            |
| IP link.                                                 |            |
+----------------------------------------------------------+------------+
| Multiple defined links between a pair of switched        | X'10160045'|
| ports is not supported by the local node.  A link        |            |
| activation request was received for a defined link,      |            |
| but there is an active defined link between the paired   |            |
| switched ports.                                          |            |
+----------------------------------------------------------+------------+
| Multiple dynamic links across a connection network       | X'10160046'|
| between a pair of switched ports is not supported by     |            |
| the local node.  A link activation request was           |            |
| received for a dynamic link, but there is an active      |            |
| dynamic link between the paired switched ports across    |            |
| the same connection network.                             |            |
+----------------------------------------------------------+------------+
| Link failure                                             | X'80020000'|
+----------------------------------------------------------+------------+
| Route selection services has determined that no path     | X'80140001'|
| to the destination node exists for the specified COS.    |            |
+----------------------------------------------------------+------------+
```

2.6 IP Throughput Characteristics

2.6.1 IP Prioritization

   Typically, IP routers process packets on a first-come-first-served
   basis; i.e., no packets are given transmission priority. However,
   some IP routers prioritize packets based on IP precedence (the 3-bit
   field within the Type of Service byte of the IP header) or UDP port
   numbers. (With the current plans for IP security, the UDP port
   numbers are encrypted; as a result, IP routers would not be able to
   prioritize encrypted traffic based on the UDP port numbers.) HPR
   will be able to exploit routers that provide priority function.

   The 5 UDP port numbers, 12000-12004 (decimal), have been assigned by
   the Internet Assigned Number Authority (IANA). Four of these port
   numbers are used for ANR-routed network layer packets (NLPs) and
   correspond to the APPN transmission priorities (network, 12001; high,
   12002; medium, 12003; and low, 12004), and one port number (12000) is
   used for a set of LLC commands (i.e., XID, TEST, DISC, and DM) and
   function-routed NLPs (i.e., XID_DONE_RQ and XID_DONE_RSP). These
   port numbers are used for "listening" and are also used in the
   destination port number field of the UDP header of transmitted
   packets. The source port number field of the UDP header can be set
   either to one of these port numbers or to an ephemeral port number.

   The IP precedence for each transmission priority and for the set of
   LLC commands (including function-routed NLPs) are configurable. The
   implicit assumption is that the precedence value is associated with
   priority queueing and not with bandwidth allocation; however,
   bandwidth allocation policies can be administered by matching on the
   precedence field. The default mapping to IP precedence is shown in
   the following table:

```
+-----------------------------------------------+
| Table 2. Default IP Precedence Settings       |
+---------------------+-------------------------+
| PRIORITY            |      PRECEDENCE         |
+---------------------+-------------------------+
| LLC commands and    |         110             |
| function-routed NLPs|                         |
+---------------------+-------------------------+
| Network             |         110             |
+---------------------+-------------------------+
| High                |         100             |
+---------------------+-------------------------+
| Medium              |         010             |
+---------------------+-------------------------+
| Low                 |         001             |
+---------------------+-------------------------+
```

As an example, with this default mapping, telnet, interactive ftp,
and business-use web traffic could be mapped to a precedence value of
011, and batch ftp could be mapped to a value of 000.

These settings were devised based on the AIW's understanding of the
intended use of IP precedence. The use of IP precedence will be
modified appropriately if the IETF standardizes its use differently.
The other fields in the IP TOS byte are not used and should be set to
0.

For outgoing ANR-routed NLPs, the destination (and optionally the
source) UDP port numbers and IP precedence are set based on the
transmission priority specified in the HPR network header.

It is expected that the native IP DLC architecture described in this
document will be used primarily for private campus or wide-area
intranets where the customer will be able to configure the routers to
honor the transmission priority associated with the UDP port numbers
or IP precedence. The architecture can be used to route HPR traffic
in the Internet; however, in that environment, routers do not
currently provide the priority function, and customers may find the
performance unacceptable.

In the future, a form of bandwidth reservation may be possible in IP
networks using the Resource ReSerVation Protocol (RSVP), or the
differentiated services currently being studied by the Integrated
Services working group of the IETF. Bandwidth could be reserved for
an HPR/IP link thus insulating the HPR traffic from congestion
associated with the traffic of other protocols.

2.6.2 APPN Transmission Priority and COS

   APPN transmission priority and class of service (COS) allow APPN TGs
   to be highly utilized with batch traffic without impacting the
   performance of response-time sensitive interactive traffic.
   Furthermore, scheduling algorithms guarantee that lower-priority
   traffic is not completely blocked. The result is predictable
   performance.

   When a session is initiated across an APPN network, the session's
   mode is mapped into a COS and transmission priority. For each COS,
   APPN has a COS table that is used in the route selection process to
   select the most appropriate TGs (based on their TG characteristics)
   for the session to traverse. The TG characteristics and COS tables
   are defined such that APPN topology and routing services (TRS) will
   select the appropriate TG for the traffic of each COS.

2.6.3 Default TG Characteristics

   In Chapter 7 (TRS) of [1], there is a set of SNA-defined TG default
   profiles. When a TG (connection network or regular) is defined as
   being of a particular technology (e.g., ethernet or X.25) without
   specification of the TG's characteristics, parameters from the
   technology's default profile are used in the TG's topology entry.
   The customer is free to override these values via configuration.
   Some technologies have multiple profiles (e.g., ISDN has both a
   profile for switched and nonswitched.) Two default profiles are
   required for IP TGs. This many are needed because there are both
   campus and wide-area IP networks. As a result for each HPR/IP TG, a
   customer should specify, at minimum, campus or wide area. HPR/IP TGs
   traversing the Internet should be specified as wide-area links. If
   no specification is made, a campus network is assumed.

   The 2 IP profiles are as follows:

+--+
| Table 3. IP Default TG Characteristics |
+------------------+---------+----------+---------+---------+---------+
	Cost	Cost per	Security	Propa-	Effec-
	per	byte		gation	tive
	connect			delay	capacity
	time				
+------------------+---------+----------+---------+---------+---------+
| Campus | 0 | 0 | X'01' | X'71' | X'75' |
+------------------+---------+----------+---------+---------+---------+
| Wide area | 0 | 0 | X'20' | X'91' | X'43' |
+------------------+---------+----------+---------+---------+---------+

Typically, a TG is either considered to be "free" if it is owned or
leased or "costly" if it is a switched carrier facility. Free TGs
have 0 for both cost parameters, and costly TGs have 128 for both
parameters. For campus IP networks, the default for both cost
parameters is 0.

It is less clear what the defaults should be for wide area. Because
a router normally has leased access to an IP network, the defaults
for both costs are also 0. This assumes the IP network is not
tariffed. However, if the IP network is tariffed, then the customer
should set the cost per byte to 0 or 128 depending on whether the
tariff contains a component based on quantity of data transmitted,
and the customer should set the cost per connect time to 0 or 128
based on whether there is a tariff component based on connect time.
Furthermore, for switched access to the IP network, the customer
settings for both costs should also reflect the tariff associated
with the switched access link.

Only architected values (see "Security" in [1]) may be used for a
TG's security parameter. The default security value is X'01'
(lowest) for campus and X'20' (public switched network; secure in the
sense that there is no predetermined route the traffic will take) for
wide-area IP networks. The network administrator may override the
default value but should, in that case, ensure that an appropriate
level of security exists.

For wide area, the value X'91' (packet switched) is the default for
propagation delay; this is consistent with other wide-area facilities
and indicates that IP packets will experience both terrestrial
propagation delay and queueing delay in intermediate routers. This
value is suitable for both the Internet and wide-area intranets;
however, the customer could use different values to favor intranets
over the Internet during route selection. The value X'99' (long) may
be appropriate for some international links across the Internet. For
campus, the default is X'71' (terrestrial); this setting essentially
equates the queueing delay in IP networks with terrestrial
propagation delay.

For wide area, X'43' (56 kbs) is shown as the default effective
capacity; this is at the low-end of typical speeds for wide-area IP
links. For campus, X'75' (4 Mbs) is the default; this is at the
low-end of typical speeds for campus IP links. However, customers
should set the effective capacity for both campus and wide area IP
links based on the actual physical speed of the access link to the IP
network; for regular links, if both the source and destination access
speeds are known, customers should set the effective capacity based
on the minimum of these two link speeds. If there are multiple
access links, the capacity setting should be based on the physical

speed of the access link that is expected to be used for the link.

For the encoding technique for effective capacity in the topology
database, see "Effective Capacity" in Chapter 7, Topology and Routing
Services of [1]. The table in that section can be extended as
follows for higher speeds:

```
+-----------------------------------------------------------------------+
| Table 4. Calculated Effective Capacity Representations                 |
+-----------------------------------+-----------------------------------+
| Link Speed (Approx.)              | Effective Capacity                |
+-----------------------------------+-----------------------------------+
| 25M                               | X'8A'                             |
+-----------------------------------+-----------------------------------+
| 45M                               | X'91'                             |
+-----------------------------------+-----------------------------------+
| 100M                              | X'9A'                             |
+-----------------------------------+-----------------------------------+
| 155M                              | X'A0'                             |
+-----------------------------------+-----------------------------------+
| 467M                              | X'AC'                             |
+-----------------------------------+-----------------------------------+
| 622M                              | X'B0'                             |
+-----------------------------------+-----------------------------------+
| 1G                                | X'B5'                             |
+-----------------------------------+-----------------------------------+
| 1.9G                              | X'BC'                             |
+-----------------------------------+-----------------------------------+
```

2.6.4 SNA-Defined COS Tables

SNA-defined batch and interactive COS tables are provided in [1].
These tables are enhanced in [2] (see section 18.7.2) for the
following reasons:

o   To ensure that the tables assign reasonable weights to ATM TGs
     relative to each other and other technologies based on cost,
     speed, and delay

o   To facilitate use of other new higher-speed facilities - This
     goal is met by providing several speed groupings above 10 Mbps.
     To keep the tables from growing beyond 12 rows, low-speed
     groupings are merged.

Products implementing the native IP DLC should use the new COS
tables. Although the effective capacity values in the old tables are
sufficient for typical IP speeds, the new tables are valuable because
higher-speed links can be used for IP networks.

2.6.5 Route Setup over HPR/IP links

   The Resequence ("REFIFO") indicator is set in Route Setup request and
   reply when the RTP path uses a multi-link TG because packets may not
   be received in the order sent. The Resequence indicator is also set
   when the RTP path includes an HPR/IP link as packets sent over an IP
   network may arrive out of order.

   Adaptive rate-based congestion control (ARB) is an HPR Rapid
   Transport Protocol (RTP) function that controls the data transmission
   rate over RTP connections. ARB also provides fairness between the
   RTP traffic streams sharing a link. For ARB to perform these
   functions in the IP environment, it is necessary to coordinate the
   ARB parameters with the IP TG characteristics. This is done for IP
   links in a similar manner to that done for other link types.

2.6.6 Access Link Queueing

   Typically, nodes implementing the native IP DLC have an access link
   to a network of IP routers. These IP routers may be providing
   prioritization based on UDP port numbers or IP precedence. A node
   implementing the native IP DLC can be either an IP host or an IP
   router; in both cases, such nodes should also honor the priorities
   associated with either the UDP port numbers or the IP precedence when
   transmitting HPR data over the access link to the IP network.

```
-----------------------------------------------------------------------

*--------* access link *--------*      *--------*
|  HPR   |-------------|   IP    |-----|   IP    |
| node   |             | Router  |     | Router  |
*--------*             *--------*      *--------*
                           |               |
                           |               |
                           |               |
                           |               |
                       *--------*      *--------* access link *--------*
                       |   IP    |-----|   IP    |-------------|  HPR   |
                       | Router  |     | Router  |             | node   |
                       *--------*      *--------*              *--------*


-----------------------------------------------------------------------
```
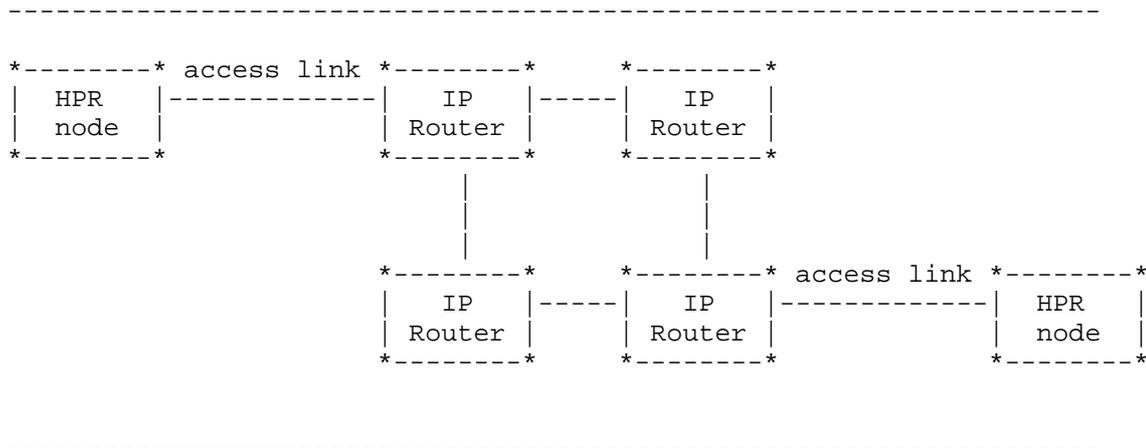   Figure 13. Access Links

   Otherwise, the priority function in the router network will be
   negated with the result being HPR interactive traffic delayed by
   either HPR batch traffic or the traffic of other higher-layer
   protocols at the access link queues.

2.7 Port Link Activation Limits

   Three parameters are provided by NOF to CS on DEFINE_PORT(RQ) to
   define the link activation limits for a port: total limit, inbound
   limit, and outbound limit. The total limit is the desired maximum
   number of active link stations allowed on the port for both regular
   TGs and connection network TGs. The inbound limit is the desired
   number of link stations reserved for connections initiated by
   adjacent nodes; the purpose of this field is to insure that a minimum
   number of link stations may be activated by adjacent nodes. The
   outbound limit is the desired number of link stations reserved for
   connections initiated by the local node. The sum of the inbound and
   outbound limits must be less than or equal to the total limit. If
   the sum is less than the total limit, the difference is the number of
   link stations that can be activated on a demand basis as either
   inbound or outbound. These limits should be based on the actual
   adapter capability and the node's resources (e.g., control blocks).

   A connection network TG will be reported to topology as quiescing
   when its port's total limit threshold is reached; likewise, an
   inactive auto-activatable regular TG is reported as nonoperational.
   When the number of active link stations drops far enough below the
   threshold (e.g., so that at least 20 percent of the original link
   activation limit has been recovered), connection network TGs are
   reported as not quiescing, and auto-activatable TGs are reported as
   operational.

2.8 Network Management

   APPN and HPR management information is defined by the APPN MIB (RFC
   2155 [11]) and the HPR MIB (RFC 2238 [13]). In addition, the SNANAU
   working group of the IETF plans to define an HPR-IP-MIB that will
   provide HPR/IP-specific management information. In particular, this
   MIB will provide a mapping of APPN traffic types to IP Type of
   Service Precedence values, as well as a count of UDP packets sent for
   each traffic type.

   There are also rules that must be specified concerning the values an
   HPR/IP implementation returns for objects in the APPN MIB:

   o   Several objects in the APPN MIB have the syntax IANAifType. The
       value 126, defined as "IP (for APPN HPR in IP networks)" should
       be returned by the following three objects when they identify an
       HPR/IP link:

       -   appnPortDlcType
       -   appnLsDlcType
       -   appnLsStatusDlcType

   o   Link-level addresses are reported in the following objects:

       -    appnPortDlcLocalAddr
       -    appnLsLocalAddr
       -    appnLsRemoteAddr
       -    appnLsStatusLocalAddr
       -    appnLsStatusRemoteAddr

       All of these objects should return ASCII character strings that
       represent IP addresses in the usual dotted-decimal format. (At
       this point it's not clear what the "usual...format" will be for
       IPv6 addresses, but whatever it turns out to be, that is what
       these objects will return when an HPR/IP link traverses an IP
       network.)

   o   The following two objects return Object Identifiers that tie
       table entries in the APPN MIB to entries in lower-layer MIBs:

       -    appnPortSpecific
       -    appnLsSpecific

       Both of these objects should return the same value: a RowPointer
       to the ifEntry in the agent's ifTable for the physical interface
       associated with the local IP address for the port. If the agent
       implements the IP-MIB (RFC 2011 [12]), this association between
       the IP address and the physical interface will be represented in
       the ipNetToMediaTable.

2.9 IPv4-to-IPv6 Migration

   The native IP DLC is architected to use IP version 4 (IPv4).
   However, support for IP version 6 (IPv6) may be required in the
   future.

   IP routers and hosts can interoperate only if both ends use the same
   version of the IP protocol. However, most IPv6 implementations
   (routers and hosts) will actually have dual IPv4/IPv6 stacks. IPv4
   and IPv6 traffic can share transmission facilities provided that the
   router/host at each end has a dual stack. IPv4 and IPv6 traffic will
   coexist on the same infrastructure in most areas. The version number
   in the IP header is used to map incoming packets to either the IPv4
   or IPv6 stack. A dual-stack host which wishes to talk to an IPv4
   host will use IPv4.

   Hosts which have an IPv4 address can use it as an IPv6 address using
   a special IPv6 address prefix (i.e., it is an embedded IPv4 address).
   This mapping was provided mainly for "legacy" application
   compatibility purposes as such applications don't have the socket

structures needed to store full IPv6 addresses. Two IPv6 hosts may
communicate using IPv6 with embedded-IPv4 addresses.

Both IPv4 and IPv6 addresses can be stored by the domain name service
(DNS). When an application queries DNS, it asks for IPv4 addresses,
IPv6 addresses, or both. So, it's the application that decides which
stack to use based on which addresses it asks for.

Migration for HPR/IP ports will work as follows:

An HPR/IP port is configured to support IPv4, IPv6, or both. If IPv4
is supported, a local IPv4 address is defined; if IPv6 is supported,
a local IPv6 address (which can be an embedded IPv4 address) is
defined. If both IPv4 and IPv6 are supported, both a local IPv4
address and a local IPv6 address are defined.

Defined links will work as follows: If the local node supports IPv4
only, a destination IPv4 address may be defined, or an IP host name
may be defined in which case DNS will be queried for an IPv4 address.
If the local node supports IPv6 only, a destination IPv6 address may
be defined, or an IP host name may be defined in which case DNS will
be queried for an IPv6 address. If both IPv4 and IPv6 are supported,
a destination IPv4 address may be defined, a destination IPv6 address
may be defined, or an IP host name may be defined in which case DNS
will be queried for both IPv4 and IPv6 addresses; if provided by DNS,
an IPv6 address can be used, and an IPv4 address can be used
otherwise.

Separate IPv4 and IPv6 connection networks can be defined. If the
local node supports IPv4, it can define a connection network TG to
the IPv4 VRN. If the local node supports IPv6, it can define a TG to
the IPv6 VRN. If both are supported, TGs can be defined to both
VRNs. Therefore, the signaling information received in RSCVs will be
compatible with the local node's capabilities unless a configuration
error has occurred.

3.0 References

[1] IBM, Systems Network Architecture Advanced Peer-to-Peer
Networking Architecture Reference, SC30-3442-04. Viewable at URL:
http://www.raleigh.ibm.com/cgi-bin/bookmgr/BOOKS/D50L0000/CCONTENTS

[2] IBM, Systems Network Architecture Advanced Peer-to-Peer
Networking High Performance Routing Architecture Reference, Version
3.0, SV40-1018-02. Viewable at URL: http://www.raleigh.ibm.com/cgi-
bin/bookmgr/BOOKS/D50H6001/CCONTENTS

[3]   IBM, Systems Network Architecture Formats, GA27-3136-16.
Viewable at URL: http://www.raleigh.ibm.com/cgi-
bin/bookmgr/BOOKS/D50A5003/CCONTENTS

[4]   Wells, L. and A. Bartky, "Data Link Switching: Switch-to-Switch
Protocol, AIW DLSw RIG: DLSw Closed Pages, DLSw Standard Version
1.0", RFC 1795, April 1995.

[5]   Bryant, D. and P. Brittain, "APPN Implementers' Workshop Closed
Pages Document DLSw v2.0 Enhancements", RFC 2166, June 1997.

[6]   Postel, J., "User Datagram Protocol", STD 6, RFC 768, August
1980.

[7]   Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[8]   Almquist, P., "Type of Service in the Internet Protocol Suite",
RFC 1349, July 1992.

[9]   Braden, R., "Requirements for Internet Hosts -- Communication
Layers", STD 3, RFC 1122, October 1989.

[10] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin,
"Resource ReSerVation Protocol (RSVP) -- Version 1 Functional
Specification", RFC 2205, September 1997.

[11] Clouston, B., and B. Moore, "Definitions of Managed Objects for
APPN using SMIv2", RFC 2155, June 1997.

[12] McCloghrie, K., "SNMPv2 Management Information Base for the
Internet Protocol using SMIv2", RFC 2011, November 1996.

[13] Clouston, B., and B. Moore, "Definitions of Managed Objects for
HPR using SMIv2", RFC 2238, November 1997.

4.0 Security Considerations

For HPR, the IP network appears to be a link. For that reason, the
SNA session-level security functions (user authentication, LU
authentication, session encryption, etc.) are still available for
use. In addition, as HPR traffic flows as UDP datagrams through the
IP network, IPsec can be used to provide network-layer security
inside the IP network.

There are firewall considerations when supporting HPR traffic using
the native IP DLC. First, the firewall filters can be set to allow
the HPR traffic to pass. Traffic can be restricted based on the
source and destination IP addresses and the destination port number;

the source port number is not relevant. That is, the firewall should
accept traffic with the IP addresses of the HPR/IP nodes and with
destination port numbers in the range 12000 to 12004. Second, the
possibility exists for an attack using forged UDP datagrams; such
attacks could cause the RTP connection to fail or even introduce
false data on a session. In environments where such attacks are
expected, the use of network-layer security is recommended.

5.0 Author's Address

Gary Dudley
C3BA/501
IBM Corporation
P.O. Box 12195
Research Triangle Park, NC 27709, USA

Phone: +1 919-254-4358
Fax: +1 919-254-6243
EMail: dudleyg@us.ibm.com

6.0 Appendix - Packet Format

6.1 HPR Use of IP Formats

```
+----------------------------------------------------------------------+
| 6.1.1  IP Format for LLC Commands and Responses                      |
|                                                                      |
|                  The formats described here are used for the         |
|                  following LLC commands and responses:  XID          |
|                  command and response, TEST command and response,    |
|                  DISC command, and DM response.                      |
+----------------------------------------------------------------------+
```

```
+----------------------------------------------------------------------+
| IP Format for LLC Commands and Responses                             |
+-------+-----+--------------------------------------------------------+
| Byte  | Bit | Content                                                |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| 0-p   |     | IP header (see note 1)                                 |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+1-  |     | UDP header (see note 2)                                |
| p+8   |     |                                                        |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+9-  |     | IEEE 802.2 LLC header (see note 3)                     |
|       |     | _____                                     |
| p+11  |     |                                                        |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+9   |     | DSAP:  same as for the base APPN (i.e., X'04' or an    |
|       |     | installation-defined value)                            |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+10  |     | SSAP:  same as for the base APPN (i.e., X'04' or an    |
|       |     | installation-defined value)                            |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+11  |     | Control: set as appropriate                            |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+12-n|     | Remainder of PDU:  XID3 or TEST information field, or  |
|       |     | null for DISC command and DM response                  |
+-------+-----+--------------------------------------------------------+
```

```
+-------+-----+--------------------------------------------------------+
|       |     |     Note 1:  Rules for encoding the IP header can be found |
|       |     |     in RFC 791.                                        |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
|       |     |     Note 2:  Rules for encoding the UDP header can be   |
|       |     |     found in RFC 768.                                  |
+-------+-----+--------------------------------------------------------+


+----------------------------------------------------------------------+
| IP Format for LLC Commands and Responses                             |
+-------+-----+--------------------------------------------------------+
| Byte  | Bit | Content                                                |
+-------+-----+--------------------------------------------------------+


+-------+-----+--------------------------------------------------------+
|       |     |     Note 3:  Rules for encoding the IEEE 802.2 LLC header |
|       |     |     can be found in ISO/IEC 8802-2:1994 (ANSI/IEEE Std |
|       |     |     802.2, 1994 Edition), Information technology -     |
|       |     |     Telecommunications and information exchange between |
|       |     |     systems - Local and metropolitan area networks -  |
|       |     |     Specific requirements - Part 2:  Logical Link Control. |
+-------+-----+--------------------------------------------------------+


+----------------------------------------------------------------------+
| 6.1.2  IP Format for NLPs in UI Frames                               |
|                                                                      |
|                 This format is used for either LDLC specific         |
|                 messages or HPR session and control traffic.         |
+----------------------------------------------------------------------+
+----------------------------------------------------------------------+
| IP Format for NLPs in UI Frames                                      |
+-------+-----+--------------------------------------------------------+
| Byte  | Bit | Content                                                |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| 0-p   |     | IP header (see note 1)                                 |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+1-  |     | UDP header (see note 2)                                |
| p+8   |     |                                                        |
+-------+-----+--------------------------------------------------------+
+-------+-----+--------------------------------------------------------+
| p+9-  |     | IEEE 802.2 LLC header                                  |
|       |     |     _____                           |
| p+11  |     |                                                        |
+-------+-----+--------------------------------------------------------+
```

```
+-------+-----+------------------------------------------------------+
| p+9   |     | DSAP:  the destination SAP obtained from the IEEE    |
|       |     | 802.2 LLC (X'80') subfield in the HPR Capabilities   |
|       |     | (X'61') control vector in the received XID3 (see note|
|       |     | 3)                                                   |
+-------+-----+------------------------------------------------------+
+-------+-----+------------------------------------------------------+
| p+10  |     | SSAP:  the source SAP obtained from the IEEE 802.2 LLC|
|       |     | (X'80') subfield in the HPR Capabilities (X'61')     |
|       |     | control vector in the sent XID3 (see note 4)         |
+-------+-----+------------------------------------------------------+
+-------+-----+------------------------------------------------------+
| p+11  |     | Control:                                             |
+-------+-----+-------+----------------------------------------------+
|       |     | X'03' | UI with P/F bit off                          |
+-------+-----+-------+----------------------------------------------+
+-------+-----+------------------------------------------------------+
| p+12-n|     | Remainder of PDU:  NLP                               |
+-------+-----+------------------------------------------------------+
+-------+-----+------------------------------------------------------+
|       |     | Note 1:  Rules for encoding the IP header can be found|
|       |     | in RFC 791.                                          |
+-------+-----+------------------------------------------------------+
+-------+-----+------------------------------------------------------+
|       |     | Note 2:  Rules for encoding the UDP header can be    |
|       |     | found in RFC 768.                                    |
+-------+-----+------------------------------------------------------+
+-------------------------------------------------------------------+
| IP Format for NLPs in UI Frames                                   |
+-------+-----+------------------------------------------------------+
| Byte  | Bit | Content                                              |
+-------+-----+------------------------------------------------------+
+-------+-----+------------------------------------------------------+
|       |     | Note 3:  The User-Defined Address bit is considered  |
|       |     | part of the DSAP.  The Individual/Group bit in the   |
|       |     | DSAP field is set to 0 by the sender and ignored by  |
|       |     | the receiver.                                        |
+-------+-----+------------------------------------------------------+
+-------+-----+------------------------------------------------------+
|       |     | Note 4:  The User-Defined Address bit is considered  |
|       |     | part of the SSAP.  The Command/Response bit in the   |
|       |     | SSAP field is set to 0 by the sender and ignored by  |
|       |     | the receiver.                                        |
+-------+-----+------------------------------------------------------+
```

7.0 Full Copyright Statement