Network Working Group J. Halpern
Request for Comments: 1223 NSC
May 1991

OSI CLNS and LLC1 Protocols on Network Systems HYPERchannel

Status of this Memo

The intent of this document is to provide a complete discussion of the protocols and techniques used to transmit OSI CLNS and LLC1 datagrams (and any associated higher level protocols) on Network Systems Corporation's HYPERchannel equipment. This document is intended for network planners and implementers who are already familiar with the OSI protocol suite and the techniques used to carry OSI traffic on standard networks such as 802.3.

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

Table of Contents

Goals of this Document

In this document, we have three major technical objectives:

1. To standardize the encapsulation of LLC1 packets over HYPERchannel. The format will be used for OSI CLNS and for any other protocols using LLC1 over HYPERchannel. (Note that if one desires to use the LLC1/SNAP combination for TCP/IP, this is the format to use. This represents an alternative to the native mode for TCP/IP over HYPERchannel, allowing for sharing the medium at the LLC1 layer.)

2. To describe how multicast protocols such as ES-IS and IS-IS shall operate over HYPERchannel. As a medium, HYPERchannel does not support either broadcast or multicast. Therefore, special techniques are needed to handle these protocols. Note that these techniques do not allow general multicast, although any specific problem may be solved by a generalization of these methods.

3. To make use of a standardized "message type" field in bytes 8 and 9 of the HYPERchannel network message. To permit better interoperability, NSC maintains a "network protocol registry" where any interested party may obtain a unique value in byte 8 (or bytes 8 and 9) for their own public, private, commercial or proprietary protocol. Lists of assigned protocol type numbers and their "owners" would be periodically published by NSC and are available to interested parties.

HYPERchannel Network Messages

Unlike most datagram delivery systems, the HYPERchannel network message consists of two parts:

```
 Message Proper
+---------------------+
|                     |
|                     |
|                     |
|     16-64 bytes     |
|                     |
|                     |
|                     |
+---------------------+

 Associated Data
+-----------------------------------------------------+
|                                                     |
|                                                     |
|                                                     |
|                                                     |
|                                                     |
|                                                     |
|           Unlimited length                          |
|                                                     |
|                                                     |
|                                                     |
|                                                     |
|                                                     |
|                                                     |
+-----------------------------------------------------+
```

The first part is a message header that can be up to 64 bytes in length. The first 16 bytes contain information required for the delivery of the entire message, and the remainder can be used by higher level protocols. The second part of the message, the "Associated Data," can be optionally included with the message proper. In most cases (transmission over HYPERchannel-50 trunks) the length of the associated data is literally unlimited. Others (such as HYPERchannel-10 or transmission within a local HYPERchannel-50 A400 adapter) limit the size of the Associated Data to 4K bytes. If the information sent can be contained within the Message Proper, then the Associated Data need not be sent.

HYPERchannel lower link protocols treat messages with and without Associated Data quite differently; "Message only" transmissions are sent using abbreviated protocols and can be queued in the receiving network adapter, thus minimizing the elapsed time needed to send and receive the messages. When associated data is provided, the HYPERchannel-50 adapters free their logical resources towards driving the host interface and coaxial trunks at maximum speed, so that data can flow through the transmitting channel, the coaxial cable, and the receiving channel concurrently. Thus HYPERchannel-50 can approach the nominal burst speed of the computer host interface when sending large data blocks over an extended period.

Message Proper Header

The first 16 bytes of the network Message Proper are examined by the network adapters to control delivery of the network message. The message format is as follows:

```
  byte   Message Proper
       +-------------------------------------------------------------+
    0  |      Trunks to Try            |        Message Flags        |
       |   TO trunks  |  FROM trunks   |                         |A/D|
       +--------------+----------------+-------------------------+---+
    2  |         TO Domain #           |         TO Network #        |
       |                               |                             |
       +-------------------------------+-----------------------------+
    4  |         TO Unit #             |        Logical To           |
       |                               |         (port number)       |
       +-------------------------------+-----------------------------+
    6  |        From Unit #            |        Logical From         |
       |                               |         (port number)       |
       +-------------------------------+-----------------------------+
    8  |                         Message type                        |
       |                           0x0B01                            |
       +-------------------------------+-----------------------------+
   10  |          FROM Domain #        |       FROM Network #        |
       |                               |                             |
       +-------------------------------+-----------------------------+
   12  |          True Unit            |         age count           |
       |                               |                             |
       +-------------------------------+-----------------------------+
   14  |      Header End Offset        |      Next Header Offset     |
       |        (16)                   |        (16)                 |
       +-------------------------------+-----------------------------+
   16  |   LLC1 destination SAP        |   LLC1 source SAP           |
       |      (0xFE for CLNP)          |      (0xFE for CLNP)        |
       +-------------------------------+-----------------------------+
   18  |   LLC1 function code          |                             |
       |      (0x03 for normal data)   |Start of upper layer protocol|
       +-------------------------------+                             +
   20  |        from bytes 19-63 of the message proper               |
       |        and continuing in the associated data                |
       |        (For OSI this is CLNP, then transport etc.)          |
       +-------------------------------+-----------------------------+
```

Trunks to Try

Consists of two four bit masks indicating which of four possible HYPERchannel-50 coaxial data trunks are to be used to transmit the message and to return it. If a bit in the mask is ON, then the adapter firmware will logically AND it with the mask of installed trunk interfaces and use the result as a candidate list of interfaces.

Whenever one of the internal "frames" are sent to communicate with the destination adapter, the transmission hardware electronically selects the first non-busy trunk out of the list of candidates. Thus selection of a data trunk is best performed by the adapter itself rather than by the host. Dedicating trunks to specific applications only makes sense in very critical real time applications such as streaming data directly from high speed overrunable peripherals.

A second Trunk mask is provided for the receiving adapter when it sends frames back to the transmitter, as it is possible to build asymmetric configurations of data trunks where trunk 1 on one box is connected to the trunk 3 interface of a second. Such configurations are strongly discouraged, but the addressing structure supports it if needed.

The "trunks to try" field is only used by HYPERchannel-50. To assure maximum interoperability, a value of 0xFF should be placed in this field to assure delivery over any technology. The newer DX series units determine the trunk mask on their own, but this field is preserved for use with A series equipment.

Message Flags

Contains options in message delivery. There are several bits defined by the hardware. However, only the A/D bit will be described here. Other bits are used only for special diagnostic or management purposes. If there is a need to set them, check the specific Network Systems manuals on their meanings. In the absence of such need, all bits other than A/D shall be set to zero on transmission, and not examined upon receipt of a message.

ASSOCIATED DATA PRESENT (A/D) is ON if an Associated Data block follows the Message Proper. 0 if only a message proper is present in the network message. The value of this bit is enforced by the network adapter firmware.

TO Domain Number

This is the most significant byte of the four byte hyperchannel address. It selects an NSC addressing domain, among a set of domains. If this and the network number both refer to the local domain and network, they may be set to 0.

TO Network Number

This is the destination network number. It identifies the network within the selected domain, where the destination unit resides. If the destination is in the local domain and network, both the TO domain and TO network numbers may be set to zero.

TO Unit

Upon arrival at the destination domain and network, this is the unit number of the destination HYPERchannel adapter. The combination of Domain, Network, and Unit uniquely identify a single adapter in a HYPERchannel network. For compatibility with existing HYPERchannel equipment, when sending a message to a destination outside the local domain and network, set this byte to 0, and store the actual destination unit number in the True Unit field.

Logical To

This field further identifies which process the message is intended for. With some hardware, the bottom bits select a machine from among several. When sending a message to an N400, the bottom two bits of this field select which of four attached hosts the message is destined for. Within a host, the logical to field selects a destination process. This is used in conjunction with the Message Type field to insure that messages are delivered to the correct place. The Logical TO field identifies a process, which then checks the Message Type to insure that it understands the message. This also allows for running two processes, both of which understand the same protocol.

From Unit

This identifies the Unit number from which this message was sent.

Logical From

This identifies the host and process who originated this message.

Message Type

The following two bytes are reserved for NSC. Users have been encouraged to put a zero in byte 8 and anything at all in byte 9 so as to not conflict with internal processing of messages by NSC firmware. In the past, this field has been loosely defined as carrying information of interest to NSC equipment carrying the message and not as a formal protocol type field. For example, an 0xFF00 in bytes 8 and 9 of the message will cause the receiving adapter to loop back the message without delivering it to the attached host.

NSC now uses both bytes 8 and 9 as a formal "protocol type" designator. Major protocols will be assigned a unique value in byte 8 that will (among good citizens) not duplicate a value generated by a different protocol. Minor protocols will have 16 bit values assigned to them so that we won't run out when 256 protocols turn up. Any interested party could obtain a protocol number or numbers by application to NSC. In this document, protocol types specific to OSI LLC1 are assigned. Specifically, the sixteen bit value 0x0B01 in bytes 8 and 9 shall identify LLC1 packets.

True Unit

This field is used to handle addressing outside of the local domain and network. For compatibility with previous NSC hardware, one may not put the destination unit number in the TO Unit field if the destination domain or network are not the local ones. In that case, one puts zero in the TO Unit field, and puts the destination Unit number into the TRUE unit field. NSC Link devices will adjust the message when it arrives at the destination domain and network so that the destination unit number appears in the TO Unit field.

Age Count

This field serves as a "time to live" in that it prevents datagrams from endlessly circulating about in an improperly configured network. Each time a message with this format passes through a bridge, the Age Count is decremented by one. When the result is zero, the message is discarded by the bridge. Therefore, this byte should be set to 255 when a message is originated, and ignored when a message is received.

Next Header Offset and Header End Offset

These fields are used by the hardware to determine if any special addressing is present. No special addressing forms are permitted in conjunction with LLC1. Therefore, these fields shall always be set to 16. Receivers may count on the LLC1 information beginning at offset 16 in the message proper.

LLC1 Data

The LLC1 Information begins at byte 16 of the message, for 3 bytes. The contains the LLC1 destination and source SAPs, followed by the LLC1 type identifier (usually 03 for unnumbered information.)

Higher Layer Protocol Data

Higher layer protocol information follows immediately after the LLC1 header in the message proper, and flows into the associated data. For purposes of this document, this is OSI CLNP, but it may be any protocol which uses LLC1.

TO Addresses and Open Driver Architecture

Since not all 16 bits of the TO address are used for the physical delivery of the network message, the remainder are considered "logical" in that their meaning is physically determined by host computer software or (in cases such as the FIPS data channel) by hardware in the host interface.

Since HYPERchannel is and will be used to support a large variety of general and special purpose protocols, it is desirable that several independent protocol servers be able to independently share the HYPERchannel network interface. The implementation of many of NSC's device drivers as well as those of other parties (such as Cray Research) support this service. Each protocol server that wishes to send or receive HYPERchannel network messages logically connects to a HYPERchannel device driver by specifying the complete 16 bit TO address it will own in the sense that any network message with that TO address will be delivered to that protocol server.

The logical TO field serves a function similar to the TYPE byte in the Ethernet message header, but differs from it in that the width of the logical TO field varies from host to host, and that no values of the logical TO address are reserved for particular protocols. On the other hand, it is possible to have several "identical" protocols (such as two independent copies of OSI with different HYPERchannel addresses) sharing the same physical HYPERchannel interface. This makes NSC's addressing approach identical to the OSI concept that the protocol server to reach is embedded within the address, rather than the IP notion of addressing a "host" and identifying a server through a message type.

Since the HYPERchannel header also has a "message type" field, there is some ambiguity concerning the respective roles of the message type and logical TO fields:

- o The logical TO field is always used to identify the protocol server which will receive the message. Once a server has specified the complete TO address for the messages it wishes to receive, the message will not be delivered to a different protocol server regardless of the contents of the message type field.

- o Although the type field cannot change the protocol server at the final destination of the message, the type field can be used by intermediate processes on the network to process the message before it reaches the server destination. An obvious example is the 0xFF00 message loopback type function, where network processing to loop back the message results in nondelivery to the TO address. In the future, intermediate nodes may process in transit messages based on the message type only for purposes such as security validation, aging of certain datagrams, and network management.

Broadcasting

NSC message forwarding protocols use low level link protocols to negotiate transmission of a message to its next destination on the network. Furthermore, NSC network boxes often fan out so that several hosts share the same network transmission equipment as in the A400 adapter. Both these characteristics mean that providing a genuine broadcast capability is not a trivial task, and in fact no NSC technology supports a broadcast capability.

However, the OSI ES-IS and IS-IS protocols require a broadcast capability to operate. Therefore, in order to support these protocols, some form of broadcast emulation must be used.

ES-IS

The End System to Intermediate System routing protocol is used by end systems to decide where to send packets. In the specified protocol, multicast messages are used so that end systems learn about intermediate systems, and intermediate systems learn about end systems. End systems normally then transmit any packets, whose correct mac destination is unknown, to a random intermediate system which then forwards the packet and tells the originator where to send future packets.

There are two situations which are distinct but related for support of this protocol over HYPERchannel. These are distinguished by whether or not there are any real intermediate systems on the HYPERchannel network.

ES-IS with Intermediate Systems

If there are one or more intermediate systems on the HYPERchannel, then the behavior is simply to emulate multicast.

END SYSTEM SUPPORT Each end system is profiled with a list of intermediate systems on the HYPERchannel. It is desirable but not necessary that this list be complete, as the future support for IS-IS will forward the necessary information to all the intermediate systems. Given the profiled list, whenever the end system wishes to originate an ESH packet (End System Hello), it will send individual copies to each intermediate system it knows about.

On most systems, these individual packets should be spaced out in time so as not to interfere with the normal transmission of OSI and other HYPERchannel messages. For end systems, an inter-packet time of 0.1 seconds is probably appropriate.

Note that if the End System receives ISH packets (Intermediate System Hello) from an IS on HYPERchannel not in its static list, it should add that to the list of systems it will send ESH packets to. The address of the new intermediate system should be remembered for the holding time in the ISH, just as with the normal operation of ES-IS.

INTERMEDIATE SYSTEMS Intermediate systems on the HYPERchannel shall also be profiled with the addresses of all the other intermediate systems on the HYPERchannel. This list is used here and in the IS-IS protocol. For the IS-IS protocol operation, it is important that the list be complete.

The list of intermediate systems is used, with ES-IS, by an intermediate system only in that it probably is also an end system. As such, it must send ESH packets to all the other intermediate systems. (The presumption that an IS is also an ES is driven by the long term requirements for network management. If you have an upper layer stack, such as is required for CMIP, you are an end system.)

Each intermediate system will keep a list of the end systems it knows about. These are the systems it has received ESH packets from. Whenever the IS sends ISH packets, it sends them individually to each ES it has heard from. In addition, it sends the ISH to any end systems which it believes, on the basis of IS-IS or other methods, are on the HYPERchannel.

Note that these packets must also be spread out in time to avoid causing congestion. However, given that the number of these is much higher than the number generated by End Systems, the time between transmissions should be selected by the IS developer to fit the sustainable I/O rates of the system. Make sure you can get at the very least one, and preferably two or three, useful packets in between each ISH copy being sent.

ES-IS without an Intermediate System

When there is no intermediate system, one or more systems must serve as address managers. These are referred to in draft ISO OSI documents as SNARE, for SubNetwork Address Resolution Entities.

END SYSTEM SUPPORT As in the previous case, each end system must be profiled with a list of intermediate systems. This list must contain all of the systems which will be serving as address managers on this network. The reason for this is that, since the address managers are not true intermediate systems, they are not running IS-IS and will not be exchanging lists of end systems they know about. There may well be several systems for redundancy and reliability.

SNARE The systems selected as address managers must appear, to the other end systems, as intermediate systems. This means that each one must send out ISH packets to all the end systems which it hears from. Each of these systems must record all the information from the ESH packets they receive. When a packet for an End System is received at a SNARE, it must behave as an IS. Specifically, it must forward the packet to the correct destination end system, and send a redirect message back to the originator, informing the originator of the correct SNPA (HYPERchannel address) for the end system.

Note that these systems are certainly end systems as well, and must send ESH packets to all the intermediate systems on the IS list, which must be complete.

ES-IS FORMAT SPECIFICATION

All ES-IS PDUS shall be formatted as specified in ISO 9542. They are then sent using LLC1 and the encapsulation specified earlier in this document for transmitting LLC1 over HYPERchannel.

RD PDUS When generating Redirect pdus, which contain HYPERchannel SNPAs (addresses), the SNPA shall be represented in four bytes. This shall be used even on a small HYPERchannel network containing only one domain and one network number.

QC FUNCTION There is no support for the ES-IS query configuration capability when using HYPERchannel. All systems must have at least one configured intermediate system, which shall be either a true IS or a SNARE.

IS-IS

The proposed IS-IS protocol for OSI (DP 10589) when run on a LAN requires broadcast capability. Because of the nature of the process for nominating the designated IS on a LAN, and other special features of this protocol, it is important never to partition the set of intermediate systems on a HYPERchannel network.

The implementation therefore is very simple. An intermediate system on HYPERchannel runs the IS-IS protocol directly. However, when it goes to send a message, it consults the profiled list of all level 1 ISs on the HYPERchannel or of all level 2 ISs on the HYPERchannel, and then sends individual copies of the message to each destination. This multiple transmission should be transparent to the IS-IS protocol itself.

Note that as with ES-IS on an intermediate system, it is important to space out the individual message transmissions. On most networks, spacing of 0.1 seconds will work well.

References

+1+ ISO IS 9542 - End system to intermediate system routing exchange protocol

+2+ ISO DP 10589 - Intermediate system to Intermediate system Infra-Domain routing exchange protocol

Security Considerations

Security issues are not discussed in this memo.

Author's Address

Joel M. Halpern
Principal Engineer
Network Systems Corporation MS033
7600 Boone Avenue North
Brooklyn Park, AN 55428

Phone: (612) 424-1606

Email: jmh@anubis.network.com