

Network Working Group  
INTERNET DRAFT  
Expires June 1999

J.-M. Pittet  
Silicon Graphics Inc.  
December 1998

IP and ARP over HIPPI-6400 (GSN)  
<draft-pittet-gsnlan-00.txt>

#### Status of this Memo

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

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

To view the entire list of current Internet-Drafts, please check the "lid-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

#### Abstract

The ANSI T11.1 task force has standardized HIPPI-6400 also known as Gigabyte System Network (GSN), a physical-level, point-to-point, full-duplex, link interface for reliable, flow-controlled, transmission of user data at 6400 Mbit/s, per direction. A parallel copper cable interface for distances of up to 40 m is specified in HIPPI-6400-PH [1]. Connections to a longer-distance optical interface are standardized in HIPPI-6400-OPT [3].

HIPPI-6400-PH [1] defines the encapsulation of IEEE 802.2 LLC PDUs [10] and by implication, IP on GSN. Another T11.1 standard describes the operation of HIPPI-6400 physical switches HIPPI-6400-SC [2]. T11.1 chose to leave HIPPI-6400 networking issues largely outside the scope of their standards; this document specifies the use of HIPPI-6400 switches as IP local area networks. This document further specifies a method for resolving IP addresses to HIPPI-6400 hardware addresses (HARP) and for emulating IP broadcast in a logical IP subnet (LIS) as a direct extension of HARP.

Furthermore it is the goal of this memo to define a IP and HARP that will allow interoperability for HIPPI-800 and HIPPI-6400 equipment both broadcast and non-broadcast capable networks.

## TABLE OF CONTENTS

1. Introduction
2. Definitions
  - 2.1 Global concepts used
  - 2.2 Glossary
3. IP Subnetwork Configuration
  - 3.1 Background
  - 3.2 HIPPI LIS Requirements
4. Internet Protocol
  - 4.1 Packet Format
    - 4.1.1 IEEE 802.2 LLC
    - 4.1.2 SNAP
    - 4.1.3 Packet diagrams
  - 4.2 HIPPI-6400 Hardware address: Universal LAN MAC address (ULA)
  - 4.3 Maximum Transmission Unit - MTU
5. HIPPI Address Resolution Protocol - HARP
  - 5.1 HARP Algorithm
    - 5.1.1 Selecting the authoritative HARP service
    - 5.1.2 HARP registration phase
    - 5.1.3 HARP operational phase
  - 5.2 HARP Client Operational Requirements
  - 5.3 Receiving Unknown HARP Messages
  - 5.4 HARP Server Operational Requirements
  - 5.5 HARP and Permanent ARP Table Entries
  - 5.6 HARP Table Aging
6. HARP Message Encoding
  - 6.1 Generic IEEE 802 ARP Message Format
  - 6.2 HIPARP Message Formats
    - 6.2.1 Example Message encodings:
    - 6.2.2 HARP\_NAK message format
7. Broadcast and Multicast
  - 7.1 Protocol for an IP Broadcast Emulation Server - PIBES
  - 7.2 IP Broadcast Address
  - 7.3 IP Multicast Address
  - 7.4 A Note on Broadcast Emulation Performance
8. HARP for Scheduled Transfer
9. Security
10. Open Issues
11. HARP Examples
  - 11.1 Registration Phase of Client Y on Non-broadcast Hardware
  - 11.2 Registration Phase of Client Y on Broadcast Capable Hardware
  - 11.3 Operational Phase (phase II)
    - 11.3.1 Successful HARP\_Resolve example
    - 11.3.2 Non-successful HARP\_Resolve example
12. References
13. Acknowledgments
14. Author's Address

## 1. Introduction

HIPPI-6400 is a duplex data channel that can transmit and receive data simultaneously at nearly 6400 megabits per second. HIPPI-6400 data transfers are cut up into micropackets which is composed of 32 data bytes and 64 bits of control information. HIPPI-6400 uses four multiplexed virtual channels. These virtual channels are allocated to control traffic, low latency traffic, and bulk traffic.

Using small packets and the virtual channels make that very large file transfers can not lock out a host or switch port for interactive traffic. Link control and look ahead flow control is done with Admin-micropackets that have the same size as data micropackets. HIPPI-6400 guarantees in order delivery of data at full data speed. It also supports link-level end to end checksumming and credit based flow control.

HIPPI-6400 defines a 20-bit interface for either copper or fiber-optic cables operating at 500 Mhz. It has a raw bandwidth of 10'000 Mb/s in each direction. This provides a payload bandwidth of 6400 Mb/s in each direction. [8]

Gigabyte System Network(TM) (GSN) is a marketing name for HIPPI-6400. It is a trademark of the High Performance Networking Forum (HNF; <http://www.hnf.org>) for use by its member companies that supply products complying to ANSI HIPPI-6400 standards.

HIPPI-6400-SC [2] defines 2 types of switches: bridging and non-bridging switches. the bridging switches are required to support hardware broadcast and the non-bridging are not. This memo allows for a coherent implementation of IP and HARP with bot types of switches.

## 2 Definitions

### 2.1 Global concepts used

In the following discussion, the terms "requester" and "target" are used to identify the node initiating the address resolution request and the node whose address it wishes to discover, respectively. This document will use HIPPI-800 and HIPPI-6400 when referring to concepts that apply to one or the other technology. The term HIPPI will be used when referring to both technologies.

### 2.2 Glossary

Broadcast

A distribution mode which transmits a message to all nodes. Particularly also the node sending the message.

#### Classical/Conventional

Both terms are used with respect to networks, including Ethernet, FDDI, and other 802 LAN types, as distinct from HIPPI-SC LANs.

#### Destination

The HIPPI-6400 node that receives data from a HIPPI-6400 Source.

#### HARP

HARP (HIPPI Address Resolution Protocol describes the whole set of HIPPI-6400 address resolution encodings and algorithms defined in this memo. HARP is a combination and adaptation of the Internet Address Resolution Protocol (ARP) RFC-826 [15] and Inverse ARP (InARP) [5] (see section 5). HARP also describes the HIPPI (800 and 6400) specific version of ARP (i.e. the protocol and the HIPPI specific encoding).

#### HRAL

The HARP Request Address List (see section 3.2).

#### Hardware (HW) address

The hardware address consisting of an ULA (see section 4.2)

#### Host

An entity, usually a computer system, that may have one or more HIPPI nodes and which may serve as a client or a HARP server.

#### Node

An entity consisting of one HIPPI Source/Destination dual simplex pair that is connected by parallel or serial HIPPI to a HIPPI-SC switch and that transmits and receives IP datagrams. A node may be an Internet host, bridge, router, or gateway.

## PIBES

The Protocol for Internet Broadcast Emulation Server (see section 7).

## Source

The HIPPI node that generates data to send to a HIPPI Destination.

## Universal LAN MAC Address (ULA)

A 48-bit globally unique address, administered by the IEEE, assigned to each node on an Ethernet, FDDI, 802 network, or HIPPI-SC LAN.

### 3. IP Subnetwork Configuration

#### 3.1 Background

ARP (address resolution protocol) as defined in [15] was meant to work on the 'local' cable. This definition gives the ARP protocol a local logical IP subnet (LIS) scope. In the LIS scenario, each separate administrative entity configures its hosts and routers within the LIS. Each LIS operates and communicates independently of other LIS's on the same HIPPI-6400 network.

HARP has LIS scope only and serves all nodes in the LIS. Communication to nodes located outside of the local LIS is usually provided via an IP router. This router is a HIPPI-6400 node attached to the HIPPI-6400 network that is configured as a member of one or more LIS's. This configuration MAY result in a number of disjoint LIS's operating over the same HIPPI-6400 network. Using this model, nodes of different IP subnets SHOULD communicate via an intermediate IP router even though it may be possible to open a direct HIPPI-6400 connection between the two IP members over the HIPPI-6400 network. This is an consequence of using IP and choosing to have multiple LIS's on the same HIPPI-6400 fabric.

By default, the HARP method detailed in section 5 and the classical LIS routing model MUST be available to any IP member client in the LIS.

#### 3.2 HIPPI LIS Requirements

The requirement for IP members (hosts, routers) operating in a HIPPI-6400 LIS configuration is:

- o All members of the LIS SHALL have the same IP network/subnet address and address mask [4].

The following list identifies the set of HIPPI-6400-specific parameters that MUST be implemented in each IP station connected to the HIPPI-6400 network:

- o HIPPI-6400 Hardware Address:

The HIPPI-6400 hardware address (a ULA) of an individual IP endpoint (i.e. a network adapter within a host) MUST be unique in the whole LIS.

- o HARP Request Address List (HRAL):

The HRAL is an ordered list of one or more addresses identifying the address resolution service(s). All HARP clients MUST be configured identically to have the same addresses(es) in the HRAL.

The HRAL MUST be the same for all nodes within a LIS. The HRAL MUST contain at least one, and MAY contain more than one HIPPI-6400 address, identifying the individual HARP service(s) that have authoritative responsibility for resolving HARP requests of all IP members located within the LIS. An LIS MUST have at least one HARP service entry configured and available to all members of the LIS.

By default the first address SHOULD be the reserved address for broadcast FF:FF:FF:FF:FF:FF.

Therefore, the HRAL entries are sorted in the following order:

- 1st : broadcast address (FF:FF:FF:FF:FF:FF),
- 2nd : official HARP server address (00:01:3B:FF:FF:E0),
- 3rd & on: any additional HARP server addresses will be sorted in decreasing order.

All HARP clients MUST be configured identically to have the same HRAL. Which identifies the selected HARP service.

An example of such a list:

- 1st entry: FF:FF:FF:FF:FF:FF
- 2nd entry: 00:01:3B:FF:FF:E0
- 3rd entry: <Alternate-HARP-server-ula>
- ...

Manual configuration of the addresses and address lists presented in this section is implementation dependent and further details are beyond the scope of this memo; i.e. this memo does not require any

further configuration method on how to structure the list. However, for an implementation designed in compliance with this memo, these addresses MUST be configured completely on the client, as appropriate for the LIS, prior to use by any service or operation detailed in this memo.

#### 4. Internet Protocol

##### 4.1 Packet format

The HIPPI-6400 packet format for Internet datagrams shall conform to the HIPPI-6400-PH [1] standard (see section 7 "Message structure" of []). The length of a HIPPI-6400-PH packet, including headers and trailing fill, shall be a multiple of 32 bytes as required by HIPPI-6400-PH.

ALL IP Datagrams shall be carried on HIPPI-6400-PH Virtual Channel 1 (VC1). Since HIPPI-6400-PH has a 32-byte granularity, IP Datagrams must also provide the data payload with a 32-byte granularity. If a user's data is not an integral multiple of 32-byte units, then the necessary zero fill padding SHALL be added.

D\_ULA Destination ULA SHALL be the ULA of the destination node.

S\_ULA Source ULA SHALL be the ULA of the requesting node.

M\_len Set to the IEEE 802 packet (e.g. IP or HARP message) length + 8 Bytes to account for the LLC/SNAP header length. The HIPPI-6400-PH [1] length parameter shall not include the pad.

##### 4.1.1 IEEE 802.2 LLC

The IEEE 802.2 LLC Header SHALL begin in the first byte after M\_len.

The LLC values SHALL be

SSAP	0xAA	170	(8 bits)
DSAP	0xAA	170	(8 bits)
CTL	0x03	3	(8 bits)

for a total length of 3 bytes. The 0x03 CTL value indicates the presence of a SNAP header.

##### 4.1.2 SNAP

The OUI value for Organization Code SHALL be 0x00-00-00 (3 bytes) indicating that the following two-bytes is an ethertype.

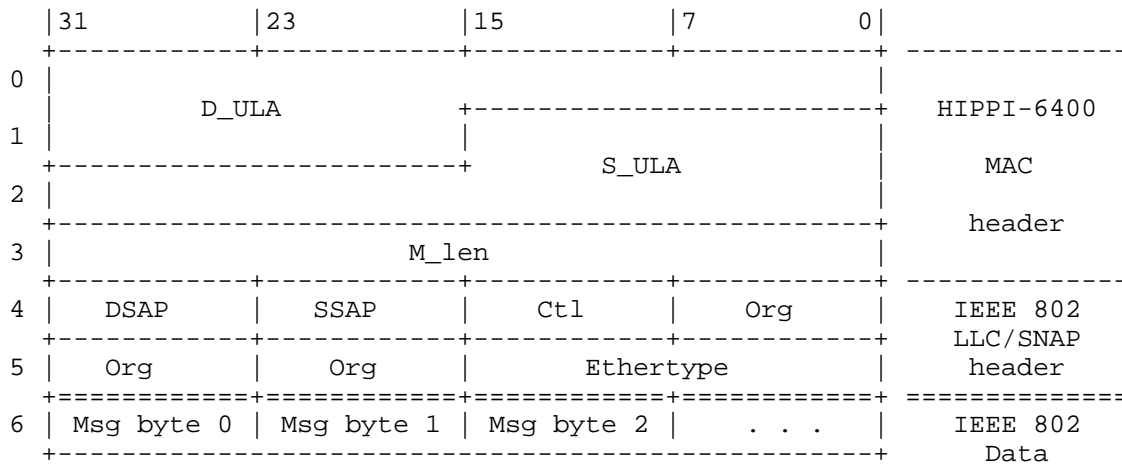
The Ethertype value SHALL be set as defined in Assigned Numbers:

```
IP           0x0800  2048  (16 bits)
HARP = ARP = 0x0806  2054  (16 bits)
```

The total size of the LLC/SNAP header is fixed at 8-bytes.

#### 4.1.3 HIPPI-6400 802 Packet diagrams

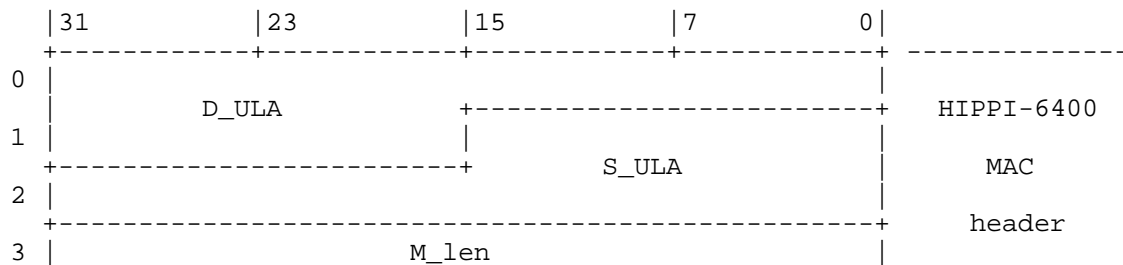
The following diagram shows a generic IEEE 802 packet.



Generic 802.1 data packet diagram

All IP (v4) packets will always span two or more micropackets. The first micropacket has a TYPE = header. The second and any further micropackets have a TYPE = Data (see [ ] for further information).

The following diagram shows an IP datagram of length n with the FILL bytes ( value: 0x0 ) marked as such. "<><>" indicates the micropacket separation. A HIPPI-6400-PH [1] micropacket is 32 bytes long.







Maximum IP packet size (MTU)	65280 bytes
	-----
Total	65536 bytes (64K)

## 5. HIPPI Address Resolution Protocol - HARP

Address resolution within the HIPPI-6400 LIS SHALL make use of the HIPPI Address Resolution Protocol (HARP) and the Inverse HIPPI Address Resolution Protocol (InHARP) . HARP provides the same functionality as the Internet Address Resolution Protocol (ARP).

HARP is based on ARP which is defined in RFC-826 [15] except the HIPPI-6400 specific packet format. Knowing the Internet address, conventional networks use ARP to discover another node's hardware address. HARP presented in this section further specifies the combination of the original protocol definitions to form a coherent address resolution service that is independent of the hardware's broadcast capability. InHARP is the same protocol as the original Inverse ARP (InARP) protocol presented in [5] except the HIPPI-6400 specific packet format. Knowing its hardware address, InARP is used to discover the other party's Internet address.

This memo further REQUIRES the PIBES (see section 7 below) extension to the HARP protocol, guaranteeing broadcast service to upper layer protocols like IP.

Internet addresses are assigned independent of ULAs. Before using HARP, each node MUST know its IP and its HW addresses. The ULA is optional but is RECOMMENDED if interoperability with conventional networks is desired.

If not all switches in the LIS support broadcast then there will be a HARP server providing the address resolution service and it will be the source of the replies. If on the other hand all switches support broadcast then the source address of a reply will be the target's source address.

### 5.1 HARP Algorithm

This section defines the behavior and requirements for HARP implementations on both broadcast and non-broadcast capable HIPPI-6400-SC networks. HARP creates a table in each node which maps remote nodes' IP addresses to ULAs, so that when an application requests a connection to a remote node by its IP address, the remote ULA can be determined, a correct HIPPI-6400-PH header can be built, and a connection to the node can be established using the correct ULA.

HARP is a two phase protocol. The first phase is the registration

phase and the second phase is the operational phase. In the registration phase the node detects if it is connected to broadcast hardware or not. The InHARP protocol is used in the registration phase. In case of non-broadcast capable hardware, the InHARP Protocol will register and establish a table entry with the server. The operational phase works much like conventional ARP with the exception of the message format.

#### 5.1.1 Selecting the authoritative HARP service

Within the HIPPI LIS, there SHALL be an authoritative HARP service. To select the authoritative HARP service, each node needs to determine if it is connected to a broadcast network. At each point in time there is only one authoritative HARP service.

The node SHALL send an InHARP\_REQUEST to the first address in the HRAL (FF:FF:FF:FF:FF:FF). If the node sees its own InHARP\_REQUEST, then it is connected to a broadcast capable network. In this case, the rest of the HRAL is ignored and the authoritative HARP service is the broadcast entry.

If the node is connected to a non broadcast capable network, then the node SHALL send the InHARP\_REQUEST to all of the remaining entries in the HRAL. Every address which sends an InHARP\_REPLY is considered to be a responsive HARP server. The authoritative HARP service SHALL be the HARP server which appears first in the HRAL.

The sequence of the HRAL is only important for deciding which address will be the authoritative one. On A non-broadcast network, the node is REQUIRED to keep "registered" with all HARP server addresses in the HRAL (NOTE: not the broadcast address since it is not a HARP server address). If for instance the authoritative HARP service is non-responsive, then the node will consider the next address in the HRAL as a candidate for the selected address and send an InHARP\_REQUEST.

The authoritative HARP server SHOULD BE considered non-responsive when it has failed to reply to one or more registration requests by the client (see section 5.1.2 and 5.2), any two HARP\_REQUESTs in the last 120 seconds or if an external agent has detected failure of the authoritative HARP server. The details of such an external agent and its interaction with the HARP client are beyond the scope of this document. Should a authoritative HARP server become non-responsive, then the registration process should be restarted. Alternative methods for choosing a authoritative HARP service are not prohibited.

#### 5.1.2 HARP registration phase

HARP clients SHALL initiate the registration phase by sending an

InHARP\_REQUEST message using all addresses in the HRAL. The client SHALL terminate the registration phase and transition into the operational phase, either when it receives its own InHARP\_REQUEST or when it receives an InHARP\_REPLY from at least one of the HARP servers and when it has determined the authoritative HARP service as described in section 5.1.1.

When nodes are initiated they send an InHARP\_REQUEST to the selected address as described in section 5.1.2. The first address to be tried will be the broadcast address "FF:FF:FF:FF:FF:FF". There are two outcomes:

1. The node sees its own InHARP\_REQUEST: then the node is connected to a broadcast capable network. The first address becomes and remains the selected address for the HARP service.
2. The node does not receive its InHARP\_REQUEST: then the node is connected to a non-broadcast capable network.

In the second case, the node SHALL choose the next address in the HRAL as a candidate for a selected address and send an InHARP\_REQUEST to that address: (0x07000FE0 00:00:00:00:00:00).

If the node receives its own message, then the node itself is the HARP server and the node is REQUIRED to provide broadcast services using the PIBES (see section 7).

If on the other hand, the node receives an InARP\_REPLY, then it is a HARP client and not a HARP server. In both cases, the current candidate address becomes the authoritative HARP service address.

If the client determines it is connected to a non-broadcast capable network then the client SHALL continue to retry each non-broadcast HARP server address in the HRAL at least once every 5 seconds until one of these two termination criteria are met for each address.

InHARP is an application of the InARP protocol for a purpose not originally intended. The purpose is to accomplish registration of node IP address mappings with a HARP server if one exists or detect hardware broadcast capability.

If the HIPPI-6400-SC LAN supports broadcast, then the client will see its own InHARP\_REQUEST message and SHALL complete the registration phase. The client SHOULD further note that it is connected to a broadcast capable network and use this information for aging the HARP server entry and for IP broadcast emulation as specified in sections 5.4 and 5.6 respectively.

If the client doesn't see its own InHARP\_REQUEST it SHALL await an InHARP\_REPLY before completing the registration phase. This will also provide the client with the protocol address by which the HARP server is addressable. This will be the case when the client happens to be connected to a non-broadcast capable HIPPI-6400-SC network.

### 5.1.3 HARP operational phase

Once a HARP client has completed its registration phase it enters the operational phase. In this phase of the protocol, the HARP client SHALL gain and refresh its own HARP table information about other IP members through the sending of HARP\_REQUESTS to the selected address in the HRAL and the reception of HARP\_REPLYS. The client is fully operational during the operational phase.

In this phase, the client's behavior for requesting HARP resolution is the same for broadcast or non-broadcast HIPPI-6400-SC switched networks.

The target of an address resolution request updates its address mapping tables with any new information it can find in the request. If it is the target node it SHALL formulate and send a reply message. A node is the target of an address resolution request if at least ONE of the following statements is true of the request:

1. The node's IP address is in the target protocol address field (ar\$tpa) of the HARP message.
2. The node's ULA, is in the ULA part of the Target Hardware Address field (ar\$tha) of the message.
3. The node is a HARP server.

NOTE: It is REQUIRED to have a HARP server run on a node that has a non-zero ULA.

### 5.2 HARP Client Operational Requirements

The HARP client is responsible for contacting the HARP server(s) to have its own HARP information registered and to gain and refresh its own HARP entry/information about other IP members. This means, as noted above, that HARP clients MUST be configured with the hardware address of the HARP server(s) in the HRAL.

HARP clients MUST:

1. When an interface is enabled (e.g. "ifconfig <interface> up") or assigned an IP alias, the client SHALL initiate the registration

phase.

2. In the operational phase the client MUST respond to HARP\_REQUEST and InHARP\_REQUEST messages, if it is the target node. If an interface has multiple IP addresses (e.g., IP aliases) then the client MUST cycle through all the IP addresses and generate an InHARP\_REPLY for each such address. In that case an InHARP\_REQUEST can have multiple replies. (Refer to Section 7, "Protocol Operation" in RFC-1293 [5].)
3. React to address resolution reply messages appropriately to build/refresh its own client HARP table entries. All (solicited and unsolicited) HARP\_REPLYS from the selected HARP server SHALL be used to update and refresh its own client HARP table entries.

Explanation: This allows the HARP server to update the clients when one of server's mappings change, similar to what is accomplished on Ethernet with gratuitous ARP.

4. Generate and transmit InHARP\_REQUEST messages as needed and process InHARP\_REPLY messages appropriately (see section 5.1.3 and 5.6). All InHARP\_REPLY messages SHALL be used to build/refresh its client HARP table entries. (Refer to Section 7, "Protocol Operation" in [5].)

If the registration phase showed that the hardware does not support broadcast, then the client MUST refresh its own entry for the HARP server, created during the registration phase, at least once every 15 minutes. This can be accomplished either through the exchange of a HARP request/reply with the HARP server or by repeating step 1. To decrease the redundant network traffic, this timeout SHOULD be reset after each HARP\_REQUEST/HARP\_REPLY exchange.

Explanation: The HARP\_REQUEST shows the HARP server that the client is still alive. Receiving a HARP\_REPLY indicates to the client that the server must have seen the HARP\_REQUEST.

If the registration phase showed that the underlying network supports broadcast, then the operation is NOT REQUIRED.

### 5.3 Receiving Unknown HARP Messages

If a HARP client receives a HARP message with an operation code (ar\$op) that it is not coded to support, it MUST gracefully discard the message and continue normal operation. A HARP client is NOT REQUIRED to return any message to the sender of the undefined message.

#### 5.4 HARP Server Operational Requirements

A HARP server accepts HIPPI-6400 connections from other HIPPI-6400 nodes. The HARP server expects an InARP\_REQUEST as the first message from the client. A server examines the IP address, the hardware address of the InARP\_REQUEST and adds or updates its HARP table entry <IP address(es), ULA> as well as the time stamp.

A HARP server replies to HARP\_REQUESTs and InARP\_REQUESTs based on the information which it has in its table. The HARP server replies SHALL contain the hardware type and corresponding format of the request (see also sec. 6).

The following table shows all possible source address combinations on an incoming message and the actions to be taken. "linked" indicates that an existing "IP entry" is linked to a "hardware entry". It is possible to have an existing "IP entry" and to have an existing "hardware entry" but neither is linked to the other.

#	IP entry	HW entry	misc	Action
1	exists	exists	linked	*
2	exists	exists	not linked	*, a, b, e, f
3	exists	new	not linked	*, a, b, d, e, f
4	new	exists	not linked	*, c, e, f
5	new	new	not linked	*, c, d, e, f

#### Actions:

\*: update timeout value  
a: break the existing IP -> hardware (HW) -old link  
b: delete HW(old) -> IP link and decr HW(old) refcount, if refcount = 0, delete HW(old)  
c: create new IP entry  
d: create new HW entry  
e: add new IP -> HW link to IP entry  
f: add new HW -> IP link to HW entry

Examples of when this could happen:

1: supplemental message

Just update timer.

2: move an IP alias to an existing interface

If the InHARP\_REQUEST requester's IP address duplicates a table entry IP address (e.g. IPa <-> HWa) and the InHARP\_REQUEST hardware address matches a hardware address entry (e. g. HWb <-> IPb), but they are not linked together, then:

- HWa entry needs to have its reference to the current IPa address removed.
- HWb needs to have a new reference to IPa added
- IPa needs to be linked to HWb

### 3: move IP address to a new interface

If the InHARP\_REQUEST requester's IP address duplicates a table entry IP address and the InHARP\_REQUEST hardware address does not match the table entry hardware address, then a new HW entry SHALL be created and the IP entry SHALL be updated.

### 4: add IP alias to table

If the InHARP\_REQUEST requester's hardware address duplicates a hardware address entry, but there is no IP entry matching the received IP address, then IP address SHALL be added to the hardware entries previous IP address(es). (E.g. adding an IP alias).

### 5: fresh entry, add it

Standard case, create both entries and link them.

A server MUST update the HARP table entry's timeout for each HARP\_REQUEST. Explanation: if the client is sending HARP requests to the server, then the server should note that the client is still "alive" by updating the timeout on the client's HARP table entry.

A HARP server SHOULD use the PIBES (see sect. 7) to send out HARP\_REPLYs to all hardware addresses in its table when the HARP server table changes mappings. This feature decreases the time of stale entries in the clients.

If there are multiple addresses in the HRAL, then a server needs to act as a client to the other servers.

## 5.5 HARP and Permanent ARP Table Entries

An IP station MUST have a mechanism (e.g. manual configuration) for determining what permanent entries it has. The details of the mechanism are beyond the scope of this memo. The permanent entries

allow interoperability with legacy HIPPI adapters which do not yet implement dynamic HARP and use a table based static ARP. Permanent entries are not aged.

The HARP server SHOULD use the static entries to resolve incoming HARP\_REQUESTs from the clients. This feature eliminates the need for maintaining a static HARP table on the client nodes.

Dynamic information overrides static HARP information, e.g. when a HARP\_REPLY from the HARP server indicates that the client's mapping needs to be updated, then the client SHALL update the entry and note that the entry is not permanent any more.

### 5.6 HARP Table Aging

HARP table aging MUST be supported since IP addresses, especially IP aliases and also interfaces (with their ULA), are likely to move. When so doing the mapping in the clients own HARP table/cache becomes invalid and stale.

- o When a client's HARP table entry ages beyond 15 minutes, a HARP client MUST invalidate the table entry.
- o When a server's HARP table entry ages beyond 20 minutes, the HARP server MUST delete the table entry.

NOTE: the client SHOULD revalidate a HARP table entry before it ages, thus restarting the aging time when the table entry is successfully revalidated. The client MAY continue sending traffic to the node referred to by this entry while revalidation is in progress, as long as the table entry is not invalidated. The client MUST revalidate the invalidated entry prior to transmitting any non address resolution traffic to the node referred to by this entry.

The client revalidates the entry by querying the HARP server. If a valid reply is received (e.g. HARP\_REPLY), the entry is updated. If the address resolution service cannot resolve the entry (e.g. HARP\_NAK, "host not found"), the associated table entry is removed. If the address resolution service is not available (i.e. "server failure") the client MUST attempt to revalidate the entry by transmitting an InHARP\_REQUEST to the hardware address of the entry in question and updating the entry on receipt of an InHARP\_REPLY. If the InHARP\_REQUEST attempt fails to return an InHARP\_REPLY, the associated table entry is removed.

### 6. HARP Message Encoding

The HARP message is another type of IEEE 802 payload as described in

section 4.1.3 above. The HIPPI-6400 HARP SHALL support two packet formats, both the generic Ethernet ARP packet and the HIPPI-800 HARP packet format defined in [13]. HARP messages SHALL be transmitted with a hardware type code of 28 on non-broadcast capable hardware or 1 in either case.

The ar\$hrd field SHALL be used to differentiate between the two packet formats. The reply SHALL be in the format of the request.

### 6.1 Generic IEEE 802 ARP Message Format

This is the ARP packet format used by conventional IEEE 802 networks (i.e. Ethernet etc). The packet format is described in RFC 826 [15] and is given here only for completeness purpose.

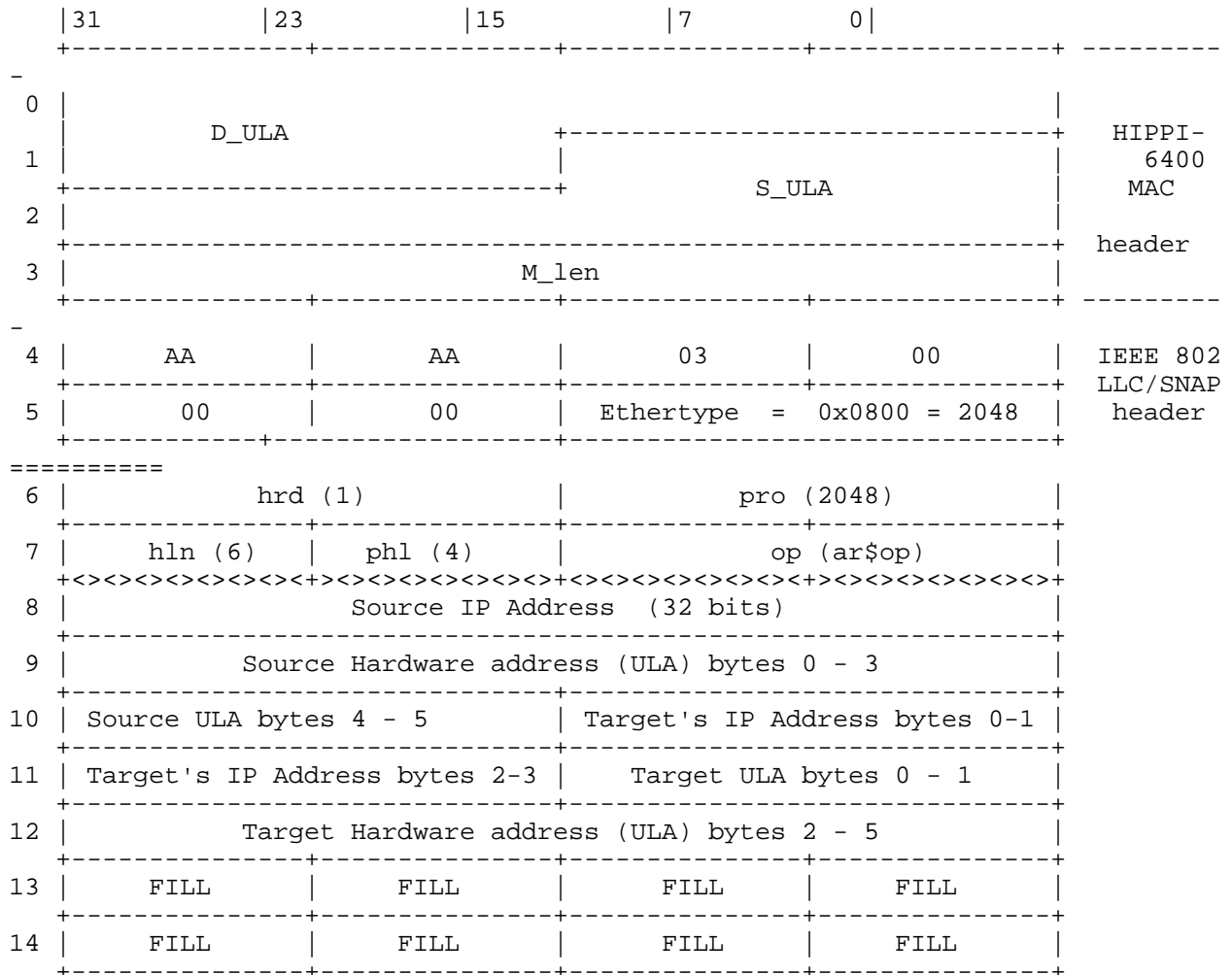
ar\$hrd	16 bits	Hardware type
ar\$pro	16 bits	Protocol type of the protocol fields below
ar\$hln	8 bits	byte length of each hardware address
ar\$pln	8 bits	byte length of each protocol address
ar\$op	16 bits	opcode (ares_op\$REQUEST   ares_op\$REPLY)
ar\$sha	48 bits	Hardware address of sender of this packet
ar\$spa	32 bits	Protocol address of sender of this packet
ar\$tha	48 bits	Hardware address of target of this
ar\$tpa	32 bits	Protocol address of target.

Where:

ar\$hrd	- SHALL contain 1. (Ethernet)
ar\$pro	- SHALL contain the IP protocol code 2048 (decimal).
ar\$hln	- SHALL contain 6.
ar\$pln	- SHALL contain 4.
ar\$op	- SHALL contain the operational value (decimal): 1 for HARP_REQUESTs 2 for HARP_REPLYs 8 for InHARP_REQUESTs 9 for InHARP_REPLYs 10 for HARP_NAK
ar\$rpa	- in requests and NAKs it SHALL contain the requester's IP address if known, otherwise zero. In other replies it SHALL contain the target node's IP address.
ar\$sha	- in requests and NAKs it SHALL contain the requester's ULA In replies it SHALL contain the target node's ULA.

- ar\$spa - in requests and NAKs it SHALL contain the requester's IP address if known, otherwise zero.  
In other replies it SHALL contain the target node's IP address.
- ar\$tha - in requests and NAKs it SHALL contain the target's ULA if known, otherwise zero.  
In other replies it SHALL contain the requester's ULA.
- ar\$tpa - in requests and NAKs it SHALL contain the target's IP address if known, otherwise zero.  
In other replies it SHALL contain the requester's IP address.

Payload Format for IEEE HARP/InHARP packet:







```

12 |                               Target ULA bytes 1 - 4                               |
   +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
13 | Tgt ULA oct 5 |
   +-----+-----+

```

#### HARP - InHARP Message

##### 6.2.1 Example Message encodings:

Assume for the following example that the HARP server is in the HIPPI-6400 side and the clients, X and Y are on the HIPPI-800 side of the non-broadcast capable network.

##### HARP\_REQUEST message

```

HARP ar$op   = 8 (InHARP_REQUEST)
HARP ar$rpa  = Ipy                               HARP ar$tpa  = 0 **
HARP ar$rha  = SWy ULay                           HARP ar$tha  = SWa ULAs
** is what we would like to find out

```

##### HARP\_REPLY message format

```

HARP ar$op   = 9 (InHARP_REPLY)
HARP ar$rpa  = IPs *                               HARP ar$tpa  = IPy
HARP ar$rha  = SWa ULAs                           HARP ar$tha  = SWy ULay
* answer we were looking for

```

##### InHARP\_REQUEST message format

```

HARP ar$op   = 8 (InHARP_REQUEST)
HARP ar$rpa  = Ipy                               HARP ar$tpa  = 0 **
HARP ar$rha  = SWy ULay                           HARP ar$tha  = SWa ULAs
** is what we would like to find out

```

##### InHARP\_REPLY message format

```

HARP ar$op   = 9 (InHARP_REPLY)
HARP ar$rpa  = IPs *                               HARP ar$tpa  = IPy
HARP ar$rha  = SWa ULAs                           HARP ar$tha  = SWy ULay
* answer we were looking for

```

##### 6.2.2 HARP\_NAK message format

The HARP\_NAK message format is the same as the received HARP\_REQUEST message format with the operation code set to HARP\_NAK; i.e. the HARP\_REQUEST message data is copied for transmission with the HARP\_REQUEST operation code changed to the HARP\_NAK value. HARP makes use of an additional operation code for HARP\_NAK and MUST be implemented.

## 7 Broadcast and Multicast

HIPPI-6400-SC requires compliant systems to support broadcast. The switch part of the system though MAY not implement broadcast but defer that operation to a broadcast server. It is likely therefore that broadcast support will be absent from initial HIPPI-6400 switches. However, a centralized HARP server architecture solves two of the three major duties of a broadcast server.

A central entity serving the whole LIS solves the coordination problem of a distributed approach. The registration requirement solves the second problem of determining which addresses make up the set loosely called "everyone". The last duty of a broadcast server is to replicate an incoming packet and send it to "everyone".

During its registration phase, every node, including HARP server(s), discover if the underlying medium is capable of broadcast (see section 5.1.1). Should this not be the case, then the HARP server(s) MUST emulate broadcast through an IP broadcast emulation server.

A HIPPI IP broadcast server (PIBES) is an extension to the HARP server and only makes sense when the LIS does not inherently support broadcast. The PIBES allows standard networking protocols to access IP LIS broadcast.

#### 7.1 Protocol for an IP Broadcast Emulation Server - PIBES

To emulate broadcast within an LIS, a PIBES SHALL use the currently valid HARP table of the HARP server as a list of addresses called the target list. The broadcast server SHALL validate that all incoming messages have a source address which corresponds to an address in the target list. Only messages addressed to the IP LIS broadcast address (255.255.255.255- independent of the ULA!) are considered valid messages for broadcasting. Invalid messages MUST be dropped. All valid incoming messages shall be forwarded to all addresses in the target list.

It is RECOMMENDED that the broadcast server run on the same node as the HARP server since this memo does not define the protocol of exchanging the valid HARP table.

#### 7.2 IP Broadcast Address

This memo only defines IP broadcast. It is independent of the underlying hardware addressing and broadcast capabilities. Any node can differentiate between IP traffic directed to itself and a broadcast message sent to it through looking at the IP address. All IP broadcast messages SHALL use the IP LIS broadcast address (255.255.255.255).

It is RECOMMENDED that the PIPES run on the same node as the HARP server. In that case, the PIBES SHALL use the same address as the HARP server.

### 7.3 IP Multicast Address

HIPPI-6400 does not directly support multicast address, therefore there are no mappings available from IP multicast addresses to HIPPI multicast services. Current IP multicast implementations (i.e. MBONE and IP tunneling, see [7]) will continue to operate over HIPPI-based logical IP subnets if all IP multicast addresses are mapped to the IP broadcast address (255.255.255.255).

### 7.4 A Note on Broadcast Emulation Performance

It is obvious that a broadcast emulation service (as defined in section 7.1) has an inherent performance limit. In an LIS with  $n$  nodes, the upper bound on the bandwidth that such a service can broadcast is:

$$(\text{total bandwidth})/(n+1)$$

since each message must first enter the broadcast server, accounting for the additional 1, and then be sent to all  $n$  nodes. The broadcast server could forward the message destined to the node on which it runs internally, thus reducing  $(n+1)$  to  $(n)$  in a first optimization. The point is that such a service is adequate for the standard networking protocols such as RIP, OSPF, NIS, etc. since they usually use a small fraction of the network bandwidth for broadcast. The broadcast emulation server as defined in this memo allows the HIPPI-6400 network to look similar to an Ethernet network to the higher layers.

It is further obvious that such an emulation cannot be used to broadcast high bandwidth traffic. For such a solution, hardware support for broadcast is required.

## 8 HARP for Scheduled Transfer [22]

This RFC also applies for resolving addresses used with Scheduled Transfer (ST) over HIPPI-800 instead of IP. This RFC's message types and algorithms can be used for ST (since ST uses Internet Addresses) as long as there is also an IP over HIPPI implementation on all the nodes.

## 9 Security

Not all of the security issues relating to ARP over HIPPI-6400 are

clearly understood at this time.

There are known security issues relating to node impersonation via the address resolution protocols used in the Internet [6]. No special security mechanisms have been added to the address resolution mechanism defined here for use with networks using HARP.

## 10 Open Issues

## 11 HARP Examples

Assume a HIPPI-6400-SC switch is installed with three connected nodes: X, Y, and a. Each node has a unique hardware address that consists unique ULA (ULAx, ULAY and ULAA, respectively). There is a HARP server connected to a switch port that is mapped to the address HWa,

this address is the selected HIPPI hardware address in the HRAL (HARP Request Address List).

The HARP server's table is empty. Nodes X and Y each know their own hardware address. Eventually they want to talk to each other; each knows the other's IP address (from the node database) but neither knows the other's ULA. Both nodes X and Y have their interfaces configured DOWN.

Note: The LLC, SNAP, Ethertype, ar\$hrd, ar\$pro, ar\$pln fields are left out from the examples below since they are constant. As well as ar\$rh1 = ar\$th1 = 6 since these are all HIPPI-6400 examples.

### 11.1 Registration Phase of Client Y on Non-broadcast Hardware

Node Y starts: its HARP table entry state for the server: PENDING

1. Node Y initiates its interface and sends an InHARP\_REQUEST to the HWa after starting a table entry for the HWa.

```

HIPPI-6400-PH D_ULA           = ULAA
HIPPI-6400-PH S_ULA           = ULAY
HARP ar$op                     = 8 (InHARP_REQUEST)
HARP ar$rpa                     = IPY
HARP ar$tpa                     = 0 **
HARP ar$rha                     = ULAY
HARP ar$tha                     = ULAA

```

\*\* is what we would like to find out

2. HARP server receives Y's InHARP\_REQUEST, it examines the source addresses and scans its tables for a match. Since this is the first time Y connects to this server there is no entry and

one will be created and time stamped with the information from the InHARP\_REQUEST. The HARP server will then send a InHARP\_REPLY including its IP address.

```
HIPPI-6400-PH D_ULA      = ULAY
HIPPI-6400-PH S_ULA      = ULAA
HARP ar$op               = 9 (InHARP_REPLY)
HARP ar$rpa              = IPs *
HARP ar$tpa              = IPy
HARP ar$rha              = ULAA
HARP ar$tha              = ULAY
* answer we were looking for
```

3. Node Y examines the incoming InHARP\_REPLY, completes its table entry for the HARP server. The client's HARP table entry for the server now passes into the VALID state and is usable for regular HARP traffic. Receiving this reply ensures that the HARP server has properly registered the client.

#### 11.2 Registration Phase of Client Y on Broadcast Capable Hardware

If there is a broadcast capable network then the authoritative address is the broadcast address, HWb = SWb, ULAb (FF.FF.FF.FF.FF.FF).

Node Y starts: its HARP table entry state for HWa: PENDING

1. Node Y initiates its interface and sends an InHARP\_REQUEST to HWa, in this example the broadcast address, after starting a table entry.

```
HIPPI-6400-PH D_ULA      = ULAb
HIPPI-6400-PH S_ULA      = ULAY
HARP ar$op               = 8 (InHARP_REQUEST)
HARP ar$rpa              = IPy
HARP ar$tpa              = 0 **
HARP ar$rha              = ULAY
HARP ar$tha              = ULAb
** is what we would like to find out
```

2. Since the network is a broadcast network, client Y will see an InHARP\_REQUEST, it examines the source addresses. Since they are the same as what Y filled in the InHARP\_REQUEST, Y can deduce that it is connected to a broadcast medium. Node Y completes its table entry for HWa. This entry will not timeout since it is considered less than likely for a particular underlying hardware type to lose its quality of being able to do broadcast and therefore this mapping will never change.

### 11.3 Operational Phase (phase II)

The Operational Phase of the HARP protocol as specified in this memo is the same for both possibilities of a broadcast and non-broadcast capable HIPPI-6400 hardware. The selected address in the HRAL for this example section will be HWa: <SWa, ULaa> and IPs for simplicity reasons.

#### 11.3.1 Successful HARP\_Resolve example

Assume the same process (steps 1-3 of section 11.1) happened for node X. Then the state of X and Y's tables is: the HARP server table entry is in the VALID state. So lets look at the message traffic when X tries to send a message to Y. Since X doesn't have an entry for Y,

1. Node X connects to the authoritative address of the HRAL and sends a HARP\_REQUEST for Y's hardware address:

```
HIPPI-6400-PH D_ULA      = ULaa
HIPPI-6400-PH S_ULA      = ULax
HARP ar$op                = 1  (HARP_REQUEST)
HARP ar$rpa               = IPx
HARP ar$tpa               = IPy
HARP ar$rha               = ULax
HARP ar$tha               = 0  **
** is what we would like to find out
```

2. The HARP server receives the HARP request and updates its entry for X if necessary. It then generates a HARP\_REPLY with Y's hardware address information.

```
HIPPI-6400-PH D_ULA      = ULax
HIPPI-6400-PH S_ULA      = ULaa
HARP ar$op                = 2  (HARP_Reply)
HARP ar$rpa               = IPy
HARP ar$tpa               = IPx
HARP ar$rha               = ULay *
HARP ar$tha               = ULax
* answer we were looking for
```

3. Node X connects to node Y and transmits an IP message with the following information in the HIPPI-LE header:

```
HIPPI-6400-PH D_ULA      = ULay
HIPPI-6400-PH S_ULA      = ULax
<data>
```

If there had been a broadcast capable HIPPI network, the target nodes

would themselves have received the HARP\_REQUEST of step 2 above and responded to them in the same way the HARP server did.

### 11.3.2 Non-successful HARP\_Resolve example

As in 11.3.1, assume that X and Y are fully registered with the HARP server. Then the state of X and Y's HARP server table entry is: VALID. So lets look at the message traffic when X tries to send a message to Q. Further assume that interface Q is NOT configured UP, i.e. it is DOWN. Since X doesn't have an entry for Q,

1. node X connects to the HARP server switch address and sends a HARP\_REQUEST for Q's hardware address:

```
HIPPI-6400-PH D_ULA      = ULAA
HIPPI-6400-PH S_ULA      = ULAX
HARP ar$op               = 1  (HARP_REQUEST)
HARP ar$rpa              = IPx
HARP ar$tpa              = IPq
HARP ar$rha              = ULAX
HARP ar$tha              = 0  **
```

\*\* is what we would like to find out

2. The HARP server receives the HARP request and updates its entry for X if necessary. It then looks up IPq in its tables and doesn't find it. The HARP server then generates a HARP\_NAK reply message.

```
HIPPI-6400-PH D_ULA      = ULAX
HIPPI-6400-PH S_ULA      = ULAA
HARP ar$op               = 10 (HARP_NAK)
HARP ar$rpa              = IPx
HARP ar$tpa              = IPq
HARP ar$rha              = ULAX
HARP ar$tha              = 0  ***
```

\*\*\* No Answer, and notice that the fields do not get swapped, i.e. the HARP message is the same as the HARP\_REQUEST except for the operation code.

If there had been a broadcast capable HIPPI network, then there would not have been any reply.

## 12 References

- [1] ANSI NCITS 323-1998, High-Performance Parallel Interface - 6400 Mbit/s Physical Layer (HIPPI-6400-PH), draft Rev 2.3

- [2] ANSI NCITS 324-1999 High-Performance Parallel Interface - 6400 Mbit/s Physical Switch Control (HIPPI-6400-SC), draft Rev 2.5
- [3] ANSI NCITS Project Number - 1249-D, High-Performance Parallel Interface - 6400 Mbit/s Optical Specification (HIPPI-6400-OPT), draft Rev 0.7
- [4] Braden, R., "Requirements for Internet Hosts -- Communication Layers", RFC-1122, USC/Information Sciences Institute, October 1989.
- [5] Bradely, T., and Brown, C., "Inverse Address Resolution Protocol", RFC-1293, USC/Information Sciences Institute, January 1992.
- [6] Bellovin, Steven M., "Security Problems in the TCP/IP Protocol Suite", ACM Computer Communications Review, Vol. 19, Issue 2, pp. 32-48, 1989.
- [7] Deering, S, "Host Extensions for IP Multicasting", RFC-1112, USC/Information Sciences Institute, August 1989.
- [8] Chesson, Greg, "HIPPI-6400 Overview", IEEE Hot Interconnects 1996, Stanford University
- [10] IEEE, "IEEE Standards for Local Area Networks: Logical Link Control", IEEE, New York, New York, 1985.
- [11] Laubach, Mark., "Classical IP and ARP over ATM", RFC-1577, Hewlett-Packard Laboratories, January 1994
- [12] Mogul, J.C., and Deering, S.E., "Path MTU Discovery", RFC-1191, Stanford University, November, 1990.
- [13] Pittet, J.-M., "ARP and IP Broadcast over HIPPI-800", ID, Silicon Graphics Inc., March 1998
- [14]
- [15] Plummer, D., "An Ethernet Address Resolution Protocol - or - Converting Network Addresses to 48-bit Ethernet Address for Transmission on Ethernet Hardware", RFC-826, MIT, November 1982.
- [16] Postel, J., "Internet Protocol", STD 5, RFC-791, USC/Information Sciences Institute, September 1981.
- [17] Renwick, J., Nicholson, A., "IP and ARP on HIPPI", RFC-1374, Cray Research, Inc., October 1992.

[18] Renwick, J., "IP over HIPPI", RFC-2067, NetStar, Inc., January 1997.

[19] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC-1700, USC/Information Sciences Institute, October 1994.

### 13 Acknowledgments

This memo could not have come into being without the critical review from Greg Chesson, Carlin Otto, the High performance interconnect group of Silicon Graphics (specifically Jim Pinkerton, Brad Strand and Jeff Young) and the expertise of the ANSI T11.1 Task Group responsible for the HIPPI standards work.

This memo is based on the second part of [17], written by John Renwick. ARP [15] written by Dave Plummer and Inverse ARP [7] written by Terry Bradley and Caralyn Brown provide the fundamental algorithms of HARP as presented in this memo. Further, the HARP server is based on concepts and models presented in [13], written by Mark Laubach who laid the structural groundwork for the HARP server.

### 14 Author's Address

Jean-Michel Pittet  
Silicon Graphics Inc  
2011 N. Shoreline Ave  
Mountain View, CA 94040

Phone: 650-933-6149  
Fax: 650-933-3542  
EMail: jmp@sgi.com