

# The HIPPI Protocol

Jim Bell ( [bell@cse.ohio-state.edu](mailto:bell@cse.ohio-state.edu) )

---

## Abstract

The High-Performance Peripheral Interface (HIPPI) protocol was designed to facilitate high-speed communications between very high-performance computers (such as supercomputers), and thereby to attempt to meet their I/O requirements.

This paper describes the HIPPI protocol in some depth, then surveys a few topics of advancement and extension of HIPPI.

---

## Table Of Contents

- [Abstract](#)
- [Overview of HIPPI](#)
  - [HIPPI-PH \(Physical Layer\)](#)
  - [HIPPI-FP \(Framing Protocol\)](#)
  - [HIPPI-SC \(Switch Control\)](#)
  - [Mapping HIPPI to Other Protocols](#)
- [HIPPI's Origin](#)
- [Current Topics in HIPPI](#)
  - [Serial HIPPI](#)
  - [High Speed SONET Extensions](#)
  - [HIPPI Connection Management](#)
  - [Real-world Uses of HIPPI](#)
- [Annotated Bibliography](#)
- [Glossary](#)

## List Of Figures

- [Suite of HIPPI Protocol Documents](#)
- 

## Overview of HIPPI

HIPPI is a very high-speed data transfer protocol, with the following properties, features, and limitations:

- Data rates of 800 or 1600 Mb/s.
- Uses a 50- or 100-pair connection. (50-pair for 800 Mb/s data-rate, 100-pair for 1600 Mb/s data-rate.) The 100-pair connection is actually a set of two identical 50-pair cables.
- Useful for distances up to 25 meters. (Serial-HIPPI extensions are being proposed for operation up

to 10km.)

- Transfers 32 bits (for 800 Mb/s data-rate) or 64 bits (for 1600 Mb/s data-rate) in parallel. Packet format allows byte alignment.
- Connection-oriented protocol.
- Point-to-point connection.
- Simplex (i.e., one-way data transfer) operation.
- First standard in its class (data-transfer for high-performance computing environments).[1]
- Designed for ease of implementation: available options are very limited.

The HIPPI protocol is defined, at several layers, by a collection of [standards](#). The organization of these standards are shown below (from [1], [2]).

<b>HIPPI-LE</b> Link Encapsulation (Maps to IEEE 802.2)	<b>HIPPI-FC</b> (Maps to Fibre-Channel Protocols)	<b>HIPPI-IPI</b> (Maps to IPI-3 Command Sets)	<b>Others</b>
<b>HIPPI-FP</b> Framing Protocol (Packet format, packet headers)			
<b>HIPPI-PH</b> Mechanical, Electrical, Signalling (Physical Layer)		<b>HIPPI-SC</b> Switch Control (for Physical Layer)	
<b>Serial-HIPPI</b> Fiber-based HIPPI-PH 10km extender (Not proposed as an ANSI Standard)			
<b>Suite of HIPPI Protocol Documents [1,2]</b>			

The components of the HIPPI protocol are described below.

## HIPPI-PH (Physical Layer)

The HIPPI-PH standard defines the mechanical, electrical, and signaling of the HIPPI physical layer.

Note that since HIPPI is a simplex protocol, a full-duplex link would be achieved by another HIPPI connection in the opposite direction.

HIPPI's physical layer consists of a set of 50-twisted-pair copper cables (maximum length of 25 meters). Two options exist, based the desired data-rate: 800 or 1600 Mb/s. Since the latter is a logical extension of the former, this section will describe the former (800 Mb/s), then describe differences for the 1600 Mb/s channel.

## Conceptual Transmission Elements

To understand signaling, one must first understand how HIPPI transmits data. The conceptual elements of data transmission are the *connection*, the *packet*, the *burst*, and the *word*.

- One connection consists of any number of packets.
- One packet consists of any number of bursts.
- One burst consists of 1 to 256 words of data. (Maximum of 1 K-byte of data per burst.)
- Each data word is 32 bits long.

## Signal Lines

The HIPPI signals and timing are described below. Each line is dedicated to its own function. (Since HIPPI is simplex, this includes the data bus and parity bus.) The item in parentheses is the line direction: source is denoted by *S*, and destination by *D*. (Note that the source initiates the data connection. One may think of a common example of a computer talking to an external disk drive where the computer is the data source and the disk is the destination.)

- *REQUEST(S-->D)* -- The source raises this line to initiate a connection.
- *CONNECT(D-->S)* -- The destination raises this line if accepts the source's initiation. Once this line is raised, a connection is in progress.
- *READY(D-->S)* -- Once the connection has been established, the destination sends one ready pulse for each burst that it can receive (up to a maximum of 63). (These pulses are called "credits" by some authors [3].) Two things are accomplished by this means:
  1. The destination performs flow control by not sending ready pulses until it knows it can handle the resulting burst.
  2. These ready pulses can be sent at a much faster rate than the bursts themselves, and in parallel with bursts. This allows the source to have advanced notice of the amount of data it can send, and not interrupt the source's sending with these notifications.
- *PACKET(S-->D)* -- The source raises this line at the beginning of a packet transmission. It remains high as long as the packet is in progress.
- *BURST(S-->D)* -- The source raises this line at the beginning of a burst transmission. It remains high as long as data is being transmitted, then is dropped to transmit the *LLRC*.
- *DATA BUS(S-->D)* -- These 32 lines carry the data during the transmission burst. All data-bits are transmitted in parallel, with line (bit) 0 being the *LSB*.  
The DATA BUS also carries two other important items:
  - When the REQUEST line is raised, the *I-Field* (specified in the HIPPI-SC layer) is put onto the DATA BUS. This I-Field is used by physical layer switches as a destination address.
  - When the BURST line is dropped, the *LLRC* is put on the DATA BUS. This is a longitudinal, even check-sum of the entire burst data.
- *PARITY BUS(S-->D)* -- These 4 lines carry the parity bits (one for each word) along with data word.
- *CLOCK(S-->D)* -- Provided by the source, this is a continuous 25 MHz clock. (Timing issues are described below.)
- *INTERCONNECT(S<-->D)* -- This is a set of two lines, one in each direction, which inform each end that the other is powered up and connected.

## Timing

Though the HIPPI connection is considered asynchronous[4] (probably because connections can be requested asynchronously), it uses a 25 MHz (+/- 5%) clock, provided by the source, to transfer data synchronously within each burst. The clock signal is symmetrical (i.e., high and low pulses split the clock cycle evenly). During data transmission, one word is transmitted each clock cycle.

The CONNECT and READY signals are raised and dropped for a minimum of four clock cycles. Maximum signal times for most signals are not specified, to give both source and destination as much time as each needs during the times when the other is waiting on it.

The BURST and PACKET signals should be dropped for a minimum of two clock cycles, but the standard specifies that the destination should operate properly if BURST or PACKET are dropped for only one clock. This gives intermediate switches one clock cycle to synchronize.

Readers are referred to [2] for detailed timing descriptions.

## 1600 Mb/s Operation

For 1600 Mb/s operation, two identical cables are used instead of one. The doubled data-rate is achieved by doubling the width of the data-bus (64 bits instead of 32). The number of parity lines is also doubled (8 instead of 4), as is the number of interconnect lines in both directions (2 each, instead of 1). (This allows the hardware to make sure both cables are connected when required.)

## HIPPI-FP (Framing Protocol)

This standard describes the format and content (including header) of each packet of user information. Note that other layers may be implemented above this one. Note also that this is the layer which splits higher layer packets to the 1 or 2 K-byte packets required by the physical layer.

The contents of a HIPPI-FP frame are as follows:

- Header\_Area (64 bits)
  - D2\_Offset (word 0, bits 0 - 2) -- The offset into the D2\_Area buffer of the first byte of user information. This allows byte alignment of user data.
  - D1\_Area\_Size (word 0, bits 3 - 10) -- Defines the size of the D1\_Area.
  - Reserved (word 0, bits 11 - 21).
  - B (word 0, bit 22) -- This bit gives a hint to the destination: it is set to 1 if the D2\_Area will arrive in a subsequent burst (not the current one). This gives higher layer protocols a bit of advanced notice, if necessary, of the arrival of user data.
  - P (word 0, bit 23) -- Set to 1 if the D1\_Area is present. (Otherwise, the D1\_Area is zero bytes.)
  - ULP-id (word 0, bits 24-31) -- Designates the upper layer protocol to which the packet data is to be delivered. Options currently include IEEE 802.2, various IPI-3 options, or HIPPI-FC. Space is reserved for locally assigned ULP-id's. (The reader is referred to [2] for more information.)
  - D2\_Size (word 1) -- Contains the number of bytes to be found in D2\_Area .
- D1\_Area (0 - 1016 bytes) -- This area contains protocol data. (In other words, the data transmitted here is based on the particular higher layer protocol.)
- D2\_Area (0 - (2<sup>32</sup> - 2)) -- This area contains user data.
  - Padding (0 - 7 bytes) -- Up to the first 7 bytes may be unused, to allow byte alignment of the D2\_Data\_Set on the source and/or destination.
  - D2\_Data\_Set (0 - (2<sup>32</sup> - 2) bytes) -- Contains user data.
  - Padding (0 - 2047 bytes) -- May be unused.

## HIPPI-SC (Switch Control)

The most glaring limitation of HIPPI-PH is that it supports only a single point-to-point connection. Though this is necessary for practical purposes (to achieve the required data-rates), it goes against the dominant paradigm of networking: to allow many computers to share data. HIPPI-SC was developed as one workable solution to this quandary. It allows for a switching mechanism to be built which could allow multiple simultaneous point-to-point connections to occur (functionally similar to the way the phone company switches its point-to-point connections [i.e., phone calls][2]). Note that HIPPI-SC does *not* specify any switching hardware or technology, but only provides functional mechanisms for these things to be used.

HIPPI-SC is closely coupled with HIPPI-PH. As described above, when the HIPPI-PH source raises its *REQUEST* line, it also specifies an *I-Field* on the *DATA BUS*. HIPPI-SC documents the format of the *I-Field*, as follows[2]:

- *Routing Control*, bits (bits 0 - 23) -- Selects the switch destination of the HIPPI-PH packet. This is to be interpreted in the context of the other bits of this word.
- *C*, bit 24 -- The *camp-on* bit allows the source to wait for a destination to become available. The source "camps on" a busy connection, so that as soon as it becomes free it gets switched in.
- *PS*, bits 25 & 26 -- The *path-selection* bits allow the source to specify the type of routing information used by the switches to establish the connection. These bits may be modified by the switches, and their meanings are described in [2].
- *D*, bit 27 -- The *direction* bit allows destinations to reply to sources by toggling only this bit, when the switch set is properly configured.
- *W*, bit 28 -- The *double-wide* bit informs the switch and destination that two cables (i.e., the 64-bit data-bus) should be used.
- *VU*, bits 29 & 30 -- *Vendor\_Unique* bits allow vendor-specific commands to be sent to destination devices. HIPPI-SC switches must pass these bits through to the destination.
- *L*, bit 31 -- When set, allows HIPPI-SC to be bypassed completely by another (possibly proprietary) *I-Field* format. In other words, the rest of the *I-Field* is not translated.

It should be noted that although HIPPI-SC does help alleviate the the number of connections limitation, it doesn't provide the final solution. The number of switches required for interconnecting devices grows by the square of the number of devices: the cost of such a device also grows in proportion. As well, the size of HIPPI's 50-pair cable is large enough to cause switch-banks to grow at least linearly with the number of connections, and all connected computers still have to be within the 25m distance limitation. ( [Serial HIPPI](#) is providing better solutions in this area.)

## Mapping HIPPI to Other Protocols

Three other standards for HIPPI exist, which are built upon the HIPPI-FP layer. These are:

- *HIPPI-LE* (Link Encapsulation) provides mapping of IEEE 802.2 LLC headers to the *D1\_Area* and the beginning of the *D2\_Area*.
- *HIPPI-FC* (Fibre Channel) maps Fibre Channel products to the HIPPI-FP standard. This standard seems to be rather unimportant: very little literature exists on it, and Tolmie[1] mentions that a separate project, FC-FP, may allow Fibre Channel products to work on HIPPI's physical layer.
- *HIPPI-IPI* (Disk & Tape Commands) maps IPI-x standard command-sets into HIPPI-FP headers. This standard may also disappear: it may be incorporated into the IPI standards instead of retaining a set of separate HIPPI-IPI standards.

With the overwhelming popularity of the TCP/IP and related protocols, all built upon the LLC layer (described by IEEE 802.2), the HIPPI-LE standard is, of the above, by far the most important. The following lists the most important mappings of HIPPI-LE:

- Two 24-bit switch addresses (in the low 24 bits of the first two words of *D1\_Area*). These are used to select switches on a local switched HIPPI network.
- The 48-bit IEEE 802.2 source and destination address PDU's (also mapped into the *D1\_Area*). The mapping between 32-bit words in the *D1\_Area* and these 48-bit addresses in the Universal LAN MAC Address (ULA) is explicitly done by the protocol.
- The entire IEEE 802.2 LLC SNAP (Logical Link Control Sub Network Access Protocol) packet

header is mapped into the beginning of the D2\_Area .

Note that the above mappings into the D1\_Area (and, possibly, the D2\_Area ) are only one possible interpretation of these HIPPI-FP areas. Therefore, a virtually unlimited number of other mappings could be built upon the HIPPI-FP layer, and the HIPPI-FP layer (and the HIPPI-PH layer) would be completely independent of each of these.

---

## HIPPI's Origin

The HIPPI protocol standard suite was designed by ANSI Task Group X3T9.3, which was initiated and chaired by Don Tolmie of Los Alamos National Laboratory. Tolmie says that the project was first proposed in January, 1987, with products available by late 1988[1] (but van Praag, et. al.[6] date its beginning at 1989, and note that the standard is dated 1991).

HIPPI was designed to be a "fire hose" for moving data, as well as a very implementable standard. The following design guidelines were used[1]:

- Shoot for 800 Mb/s data rate, which allows "movies" to be transmitted steady-state in real-time: 1K x 1K pixels, 24 bits (of color) per pixel, and 30 frames per second require transfer rates in the neighborhood of 750 Mb/s.
- Make it as simple as possible, to maximize performance.
- Minimize the options available, to minimize complexity.
- Get the standard done quickly, to allow vendors to begin working on HIPPI products before the entire market lost interest.

Some important design decisions were made based on the above guidelines:

- Copper wire was chosen over fiber optics since suitable fiber optic components were not available (or mature enough to warrant inclusion). This imposed the 25 meter distance limitation, but allowed the standard to progress.
- A 64-bit CRC, proposed late in the development cycle, was not accepted. (It was rejected for several reasons, the most interesting one being this: the standards committee wanted to ward off perceptions in the potential vendor community that the standard was either flawed or unstable, either of which would have caused hesitancy to build HIPPI components.)
- Many other options were excluded from the standard to keep it simple and allow it to get to market quickly.

Franta and Hughes[7] consider HIPPI a "standardized variant of Cray's proprietary HSX channel, which [also] operates at 800 Mb/s." This indicates that Cray had a significant influence on the standard, though I did not find that elsewhere in the literature.

---

## Current Topics in HIPPI

This section reports some recent and current topics in HIPPI. Notably, most of these topics are attempts to overcome one of the two inherent limitations of HIPPI: the maximum distance between nodes and the

number of connections.

## Serial HIPPI

Serial HIPPI is an attempt to overcome the 25m distance limitation by the use of &HIPPI Modems& of sorts: devices that connect to HIPPI-PH via the standard 50-pair connection on one side, but which serialize the data and use another technology to transmit it longer distances to another complementary device, which converts it back to HIPPI-PH for transmission to the HIPPI destination.

Two flavors of such devices are discussed[2,3]:

- Giga-bit dark fiber optics devices, which could be ganged to achieve 1600 Mb/s operation. (If chip-sets are developed, these could be brought on-board the source and destination devices, thus allowing the 50-pair cables to be done away completely.)
- Copper co-axial cable devices.

These devices, though transparent to HIPPI-PH, inevitably increase the turn-around time of its signals. But due to the design of HIPPI-PH, latencies are well hidden, and only show up in making and breaking connections, which is typically reasonably infrequently.

## High Speed SONET Extensions

Hughes and Franta[3] propose a HIPPI extender device which will use SONET's STS-12c technology to transport HIPPI packets. They propose this device as a lower-cost alternative to other approaches using multiple slower-speed long-distance paths (OC3c).

Much like Serial HIPPI, they create a set of "HIPPI terminating devices" which serialize the data, then place it into an STS-12c payload for transmission. The down-link device then converts it back to HIPPI for transmission into the destination HIPPI link. They lay out a physical layer convergence protocol (PLCP) which maps HIPPI bursts into STS-12c rows.

Nowhere do they attempt to match HIPPI's 800 Mb/s data rate. STS-12c (over a OC-12 optical carrier) has a maximum data-rate of 622.08 Mb/s, and with their PLCP, the effective data-rate becomes 589.824 Mb/s. They do not directly address this discrepancy, but mention one customer benchmark in which supercomputers would sustain a 400 Mb/s data-rate, with or without their device in the data path. Apparently, the assumption is that steady-state data-rates of 800 Mb/s will not occur.

Their devices rely heavily on large RAM stores in the down-link extender (i.e., on the SONET receiving end). Their rough estimate is that as much as  $2 \times \text{band\_width} \times \text{delay\_bytes}$  storage is required in the down-link extender. For a 3,000-mile link, they estimate that over 1 Megabytes of data may be transmitted by the up-link extender before the HIPPI source can be stopped. This data must be buffered in the down-link extender.

The extenders they designed had the following features:

- HIPPI to SONET Interface implemented completely in hardware.
- 4 Megabytes of RAM for data buffering (on the down-link side).
- 64-bit CRC, transmitted with each STS-12c frame. Their devices carefully overlap error detection, generating the CRC before discarding the LLRC (and vice-versa). Their message is clear: expect the SONET link to allow bit errors. (The CRC is generated in hardware, using a specialized, DEC-

patented chip.)

- An i960 RISC-based microprocessor, with diskette drive, LCD display, and RS-232 and Ethernet ports, is included for supervisory functions and monitoring. Running the VxWorks real-time operating system, it monitors laser power consumption (to detect worn-out lasers), transmitted and received light levels, operating temperature, and SONET diagnostics. It has programmable thresholds for each monitored value. It also services SNMP commands.

## HIPPI Connection Management

As mentioned above, the HIPPI standards do not provide a complete, general-purpose interconnection solution. Therefore, work is being done to investigate methods for providing such a solution (without sacrificing HIPPI's data transfer speed or efficiency).

Chlamtac, Ganz and Kienzle[4] analyze the theoretical and simulated performance of various *connection management policies*. A connection management policy is a method by which connections (a network resource) are shared. Three such policies are considered:

1. *Centralized*-- One centralized, distinct management system, closely integrated with the switches, maintains all connection management information for the entire network. When a connection is established or broken, this system performs the next assignments which need to be made.
2. *Broadcast*-- A separate bus-type communications system allows nodes to broadcast information regarding the HIPPI links, and therefore manage connections by policies built upon this shared knowledge.
3. *Distributed*-- Each HIPPI node works with no explicit knowledge of any others. Management policies are then built upon random waiting and retrying methods.

They provide a detailed statistical analysis of each method (which is beyond the scope of this paper). They come to the following conclusions:

- The broadcast and centralized policies are modeled identically. The main difference between these two policies is that broadcast incurs an overhead penalty on each node, for processing the broadcast information.
- The centralized system is the most efficient, maximizing overall system throughput, minimizing average connection delay, and requiring no overhead penalty on any node. But this system is also the least scalable since it requires hardware whose complexity grows non-linearly with the number of connections.
- The broadcast and distributed systems are nearly equivalent, since each incurs an overhead penalty. In broadcast systems, as mentioned above, this is in processing broadcast information as it arrives. In distributed systems, the overhead penalty is incurred when a retry must take place. For different sets of simulation parameters, the distributed systems could be tuned to closely match the broadcast system's performance.
- Since broadcast systems require additional hardware and distributed systems don't, the authors propose the distributed approach as a reasonable solution.

## Real-World Uses of HIPPI

CERN, the European high energy physics laboratory has several HIPPI applications well underway[6].

Though they mention this in passing, it's interesting to note some of the standard HIPPI components available (as of 1992):

- A HIPPI chip set, manufactured by AMCC, which implement the HIPPI-PH level protocol.
- A HIPPI crossbar switch, manufactured by Network Systems Corp.
- HIPPI interfaces are provided by IBM (mainframes), NEC (mainframes), Sun Microsystems, and Silicon Graphics.
- A HIPPI VMEbus interface.

They are using the HIPPI VMEbus interface to build a VMEbus interface data collection unit, which will be used in the Large Hadron Collider among other things.

---

## Annotated Bibliography

1. D. Tolmie and J. Renwick, "HIPPI: Simplicity Yields Success", *IEEE Network*, Vol. 7 No. 1, January 1993, p 28-32. Describes basic operation of HIPPI, and origin of standards. Mentions goals of the standards committee.
2. D. Tolmie, "High-Performance Parallel Interface (HIPPI)", (from "High Performance Networks, Technology and Protocols," edited by Tantawy), Kluwer Academic Publishers, 1994. Describes operation of HIPPI in detail, at several levels. Describes packet formats in detail.
3. J. P. Hughes, W. R. Franta, "Geographic extension of HIPPI channels via high speed SONET", *IEEE Network*, Vol. 8 No. 3, May-June 1994. p 42-53. Describes a method for using SONET to carry HIPPI bursts. Also mentions the current state of Serial HIPPI.
4. I. Chlamtac, A. Ganz, M. G. Kienzle, "An HIPPI Interconnection System", *IEEE Transactions on Computers*, Vol. 42 No. 2, Feb 1993. p 138-150. Describes a proposed HIPPI interconnection system using a cross-bar switch network. Performs a reasonably extensive mathematical model of the performance of such a network.
5. I. Chlamtac, M. G. Kienzle, "Multitasking in high-speed interconnection systems.", *Computer Networks and ISDN Systems*, Vol. 25 No. 6, Jan 1993, p 701-716. Overlaps with [4], and also analyzes *access disciplines* (i.e., methods by which processes compete or share a single resource, such as a HIPPI port, which can only be connected to one destination at a time).
6. van Praag, A.; Anguelov, T.; Burckhart, D.; McLaren, R. A.; van der Bij, H. C.; Bovier, J.; Cristin, P.; Haben, M.; Jovanovic, P.; Kenyon, I.; Staley, R.; Cunningham, D.; Watson, G.; Green, B.; Strong, J., "HIPPI developments for CERN experiments.", *IEEE Transactions on Nuclear Science*, Vol. 39 No. 4, Aug 1992. p 880-885. Describes CERN's work with HIPPI devices. Describes a HIPPI VMEbus interface computer they are designing (from standard components).
7. W. R. Franta and J. P. Hughes, "Extended High Speed Networks Employing HIPPI Switches, High Speed WANs, and FDDI Rings", *Journal of High Speed Networks*, Vol. 2, 1992. p 167-192. This is an overview of many Wide Area Network technologies, with one brief overview section on HIPPI. Not directly on topic, this article makes several observations of networking in general, and use these to attempt to predict the future of networking. They consider HIPPI to be a significant part of the future of networking.
8. K. Hung-Chang, A. Nilsson, D. Winkelstein, L. Bottomley, "Traffic Measurements on HIPPI Links in a Super-computing Environment", (from "Asynchronous Transfer Mode Networks", edited by Y. Viniotis and R. O. Onvural), Plenum Press, New York, 1993. Heavy duty mathematical analysis of HIPPI performance.

### The HIPPI Standards:

- // NSI X3.183-1991, "High-Performance Parallel Interface - Mechanical, Electrical, and

*Signaling Protocol Specification (HIPPI-PH)"*

- *ANSI X3.210-1992, "High-Performance Parallel Interface - Framing Protocol (HIPPI-FP)"*
- *ANSI X3.218-1993, "High-Performance Parallel Interface - Encapsulation of ISO 8802-2 (IEEE Std 802.2), Logical Link Protocol Specification (HIPPI-LE)"*
- *ANSI X3.222-1993, "High-Performance Parallel Interface - Physical Switch Control Specification (HIPPI-SC)"*

## Related Standards & RFC's

- *ANSI/ISO 9318-3:1990, "Intelligent Peripheral Interface - Device Generic Command Set for Magnetic and Optical Disk Drives (IPI-3 Disk)"*
- *ANSI/ISO 9318-4:1990, "Intelligent Peripheral Interface - Device Generic Command Set for Magnetic Tape Drives (IPI-3 Tape)"*
- *IEEE 802.2, "Link Encapsulation" (also known as ISO 8802-2, "Logical Link Protocol Data Units")*
- *"Serial-HIPPI Specification, Revision 1.0, Serial-HIPPI Implementers Group" (Referenced as [ftp://nsco.network.com/hippi/serial\\_hippi\\_1.0.ps](ftp://nsco.network.com/hippi/serial_hippi_1.0.ps), but this site & file were not found.)*
- *IETF RFC 1374, "IP and ARP on HIPPI"*

---

## Glossary

### HIPPI

The High-Performance Peripheral Interface. This protocol, used to transfer data at extremely high rates, is the focus of this paper.

### PH

Physical Layer.

### FP

Framing Protocol.

### LE

Link Encapsulation.

### IPI, IPI-3, IPI-4

Intelligent Peripheral Interface. A set of disk and tape commands specified by the standards: ANSI/ISO 9318-3:1990 and ANSI/ISO 9318-4:1990.

### Mb/s

Megabits per second.

### km

Kilometers.

### MHz

Mega-hertz,  $10^6$  cycles per second.

### source

The data transmitting node (device, computer, etc.).

### destination

The data receiving node (device, computer, etc.).

### LLRC

Longitudinal Length Redundancy Check-word. This word, an even longitudinal parity word transmitted (by the physical layer) at the end of each burst, allows error detection by higher layers. All odd-bit and 2-bit errors, and 99.99% of 4-bit errors are detected.

### LSB

Least Significant Bit.

MSB

Most Significant Bit.

PDU

Protocol Data Unit. Defined by the IEEE 802.2 standard.

SONET

Synchronous Optical NETwork. A high-speed networking service provided by phone service providers.

---

[Other Reports on Recent Advances in Networking 1995](#)

[Back to Raj Jain's Home Page](#)