INTERNATIONAL STANDARD

ISO/IEC 11518-2

First edition
1996-06-15

Information technology — High-Performance Parallel Interface —

Part 2: Framing Protocol (HIPPI-FP)

Technologies de l'information — Interface parallèle à haute performance —

Partie 2: Protocole de trame (HIPPI-FP)

Reference number
ISO/IEC 11518-2:1996(E)

Contents

Page

Annexes

Tables

Figures

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 11518-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 25, *Interconnection of information technology equipment.*

ISO/IEC 11518 consists of the following parts, under the general title *Information technology – High-Performance Parallel Interface:*

- *Part 1: Mechanical, electrical, and signalling protocol specification (HIPPI-PH)*
- *Part 2: Framing Protocol (HIPPI-FP)*
- *Part 3: Encapsulation of ISO/IEC 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units (HIPPI-LE)*
- *Part 4: Mapping of HIPPI to IPI device generic command sets (HIPPI-IPI)*
- *Part 5: Memory Interface (HIPPI-MI)*
- *Part 6: Physical Switch Control (HIPPI-SC)*

Annexes A and B of this part of ISO/IEC 11518 are for information only.

Introduction

This part of ISO/IEC 11518 defines the data framing for an efficient simplex high-performance point-to-point interface.

Characteristics of HIPPI-FP include

- Large block data transfers with framing to split the data into smaller bursts.
- Separation of user control and data information, and early delivery of the control information.
- Identifiers for multiple upper-layer protocols (ULPs).
- Support for simplex topology.
- Support for ULP non-word-aligned and an arbitrary number of byte transfers.
- Error notifications, from the underlying physical layer, e.g., HIPPI-PH, are passed through this framing protocol to notify the upper layers of damaged data.
- Provides a connection-less data service.
- Best effort delivery of data, i.e., datagram.
- Connection control information, which may be used for physical layer switching, is supported.

Figure 1 shows the relationship of this part of ISO/IEC 11518 (in the solid rectangle) to the other entities shown.

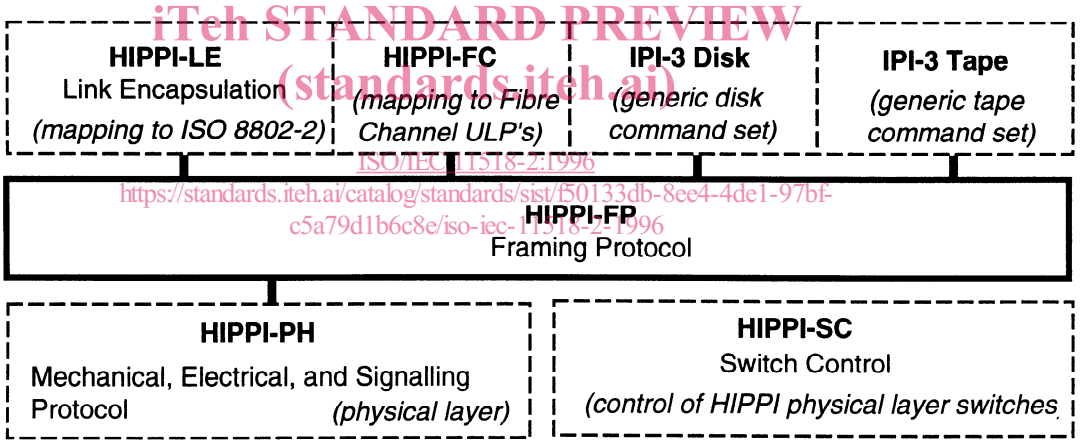

Figure 1 – HIPPI documents

Information technology – High-Performance Parallel Interface –

Part 2: Framing Protocol (HIPPI-FP)

1 Scope

This part of ISO/IEC 11518 provides the data framing for a high-performance point-to-point interface between data-processing equipment. This part of ISO/IEC 11518 does not protect against certain errors which might be introduced by intermediate devices interconnecting multiple HIPPI-PHs.

The purpose of this part of ISO/IEC 11518 is to facilitate the development and use of the HIPPI in computer systems by providing common data framing. It provides an efficient framing protocol for interconnections between computers, high-performance display systems, and high-performance, intelligent block-transfer peripherals.

2 Normative reference

The following standard contains provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 11518. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 11518 are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 11518-1:1995 - *Information technology - High-Performance Parallel Interface - Part 1: Mechanical, electrical, and signalling protocol specification (HIPPI-PH)* .

3 Definitions and conventions

3.1 Definitions

For the purposes of this part of ISO/IEC 11518, the definitions given in ISO/IEC 11518-1 and the following definitions apply.

3.1.1 burst: A group of words sent by the Source to the Destination. Bursts contain 1 to 256 words. Bursts that contain less than 256 words are called short bursts. On a 32-bit HIPPI-PH, bursts contain an even number of 32-bit words.

3.1.2 byte: A group of eight bits. Bytes are packed four per 32-bit word, or eight per 64-bit word.

3.1.3 connection: Condition of the HIPPI-PH when data transfers from Source to Destination are possible. [ISO/IEC 11518-1]

3.1.4 connection control information (CCI): A parameter sent as part of the sequence of operations establishing a connection from a Source to a Destination. (ISO/IEC 11518-6 includes examples of CCIs and topologies.)

3.1.5 Destination: The equipment at the end of the interface that receives the data. [ISO/IEC 11518-1]

3.1.6 optional: Features that are not required by this part of ISO/IEC 11518. However, if any optional feature defined by this part of ISO/IEC 11518 is implemented, it shall be implemented according to this part of ISO/IEC 11518. [ISO/IEC 11518-1]

3.1.7 packet: A data set sent from Source to Destination. A packet is composed of one or more bursts. The HIPPI specification does not limit the maximum packet size, but a maximum size may be imposed by a given HIPPI implementation, or by a ULP. A packet consists of a header, one or two optional ULP data sets, and optional fill.

3.1.8 service interface (SI): Connection points to the ULP.

3.1.9 Source: The equipment at the end of the interface that transmits the data.

3.1.10 state: The current condition of the interface, excluding transitions, as indicated by the control primitives.

3.1.11 station management (SMT): The supervisory entity that monitors and controls the HIPPI.

3.1.12 ULP data set: The data transferred between the ULP and the HIPPI-FP.

3.1.13 upper-layer protocol (ULP): A protocol immediately above the HIPPI-FP service interface.

3.1.14 word: A unit of information, consisting of 32 or 64 bits, matching the HIPPI-PH word size. Words contain an ordered set of four or eight bytes.

3.2 Editorial conventions

In this part of ISO/IEC 11518, certain terms that are proper names of signals or similar terms are printed in upper case to avoid possible confusion with other uses of the same words (e.g., CLOCK). Any lower case uses of these words have the normal technical English meaning.

A number of conditions, sequence parameters, events, states, or similar terms are printed with the first letter of each word in upper case and the rest lower case (e.g., In, Out, Enabled). Any lower case uses of these words have the normal technical English meaning.

4 HIPPI structure

4.1 Structure

The HIPPI-FP has been designed in a modular fashion to support simplex or dual simplex configuration requirements.

A compliant HIPPI network shall maintain packet and burst structures from the original Source to the final Destination.

Figure 2 shows the basic organization of the information on the HIPPI.

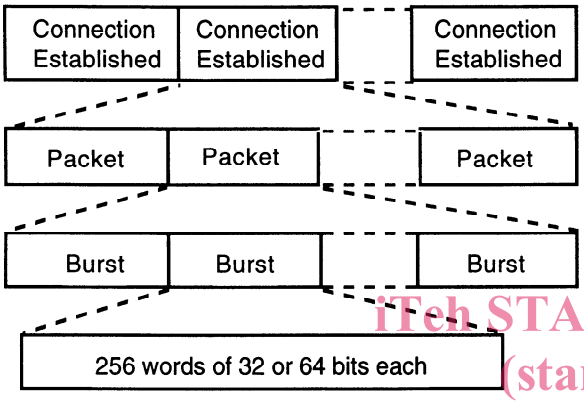

Figure 2 – Logical framing hierarchy

As specified in HIPPI-PH, once a connection is established a packet (or multiple packets) can be sent from the Source to the Destination. Each packet contains one or more bursts. Bursts contain 1 to 256 words. Words contain four or eight bytes. Bursts that contain less than 256 words are called short bursts. A packet contains no more than one short burst. A short burst may be either the first or last burst of a multiburst packet. For error detection HIPPI-PH uses byte parity and a parity-based checksum on each burst.

On a 32-bit HIPPI-PH, bursts shall contain an even number of 32-bit words. Words shall contain an ordered set of bytes as specified in 7.1.

4.2 Error detection mechanisms

4.2.1 Byte parity

The HIPPI physical layer (HIPPI-PH) uses bit-parallel word transfers, using 32-bit words for an 800 Mbit/s data rate and 64-bit words for a 1600 Mbit/s data rate. An odd-parity bit is also transmitted with each 8-bit byte of a word, i.e., four parity bits are transmitted with each 32-bit word. Hence an undetected error in a word would require a 2-bit error, with both bits being in the same byte.

4.2.2 LLRC

The Length-Longitudinal Redundancy Check (LLRC) implements even parity across the individual bits of multiple words in a burst. For example, bit 23 of the LLRC is the even parity of bit 23 of each word in the burst. A burst is nominally 256 words in length (1 Kbytes or 2 Kbytes), but short bursts may contain fewer words. Hence the LLRC would not detect errors where the same bit in an even number of words was incorrect.

In addition, the LLRC calculation includes the length of the burst. Hence, the LLRC would detect cases where a word was dropped or added, i.e., the length received was not the same as what was transmitted.

4.2.3 Packet length

A packet is composed of one or more bursts. In HIPPI-FP a length field specifying the number of bytes in the packet is specified. This length field provides a check for dropped or extra bursts. A special case where the packet length is not used is provided for such things as video data to a frame buffer, data collection from experimental equipment, etc.

4.3 Error detection limitations

The parity and LLRC will only fail on 4-bit errors in a rectangular pattern. That is, two bits in a byte must fail (undetected by the byte parity check) and the same two bits must fail in another word of the burst (undetected by the LLRC).

Use of the HIPPI-FP packet header length field permits the detection of lost bursts within a packet; however no mechanism of either HIPPI-FP or HIPPI-PH allows the detection of data corruption caused by the substitution of one burst, with good parity and LLRC, for another burst of the same length.

5 HIPPI-FP service interface to upper layers

This clause describes the services provided by HIPPI-FP. The intent is to provide the formalism necessary to relate this interface to other HIPPI interfaces. How many of the services described herein are chosen for a given implementation, and whether others may be required, is up to the implementer; however, a set of HIPPI-FP services must be supplied sufficient to satisfy the ULP(s) being used. The services as defined herein do not imply any particular implementation, or any interface.

In this part of ISO/IEC 11518 the ULP and station management protocol (SMT) are service users, and the HIPPI-FP is the service provider to the ULP and SMT. The interfaces consist of the ULP primitives, prefixed with FP_, and the SMT primitives, prefixed with FPSM_.

The HIPPI-FP is also the service user of the HIPPI-PH services, prefixed with PH_.

Figure 3 shows the relationship of the HIPPI-FP interfaces.

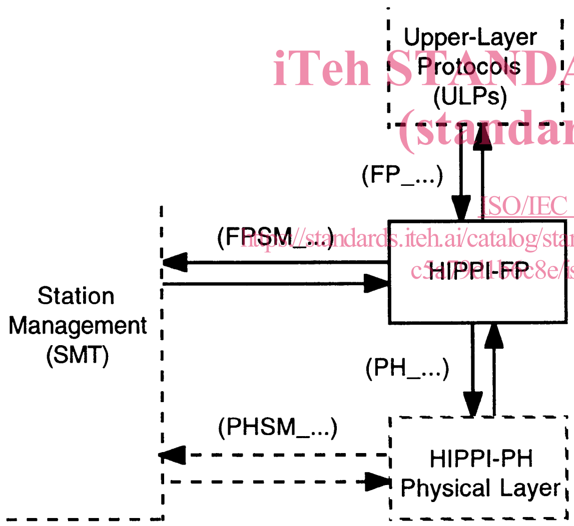

Figure 3 – HIPPI-FP service interface

5.1 Service primitives

All of the primitives and parameters are considered as required except where explicitly stated otherwise.

HIPPI service primitives are of four types.

– *Request primitives* are issued by a service user to initiate a service from the service provider. In this part of ISO/IEC 11518, a second Request primitive of the same name shall not be issued until the Confirm for the first request is received.

– *Confirm primitives* are issued by the service provider to acknowledge a Request.

– *Indicate primitives* are issued by the service provider to notify the service user of a local event. This primitive is similar in nature to an unsolicited interrupt. Note that the local event may have been caused by a service Request. In this part of ISO/IEC 11518, a second Indicate primitive of the same name shall not be issued until the Response for the first Indicate is received.

– *Response primitives* are issued by a service user to acknowledge an Indicate.

5.2 Sequences of primitives

The order of execution of service primitives is not arbitrary. Logical and time sequence relationships exist for all described service primitives. Time sequence diagrams, as in figure 4, are used to illustrate a valid sequence. Other valid sequences may exist. The sequence of events between peer users across the user/provider interface is illustrated. In the time sequence diagrams the HIPPI-FP users are depicted on either side of the vertical bars while the service provider is in the centre. A ULP or SMT implementation may present multiple requests for services, but the requests shall be serviced one at a time and in the order presented.

5.3 HIPPI-FP service primitive summary

ULP Data Transfer
FP_TRANSFER.Request (CCI, ULP-id, D1_Size, D1_Data_Set, D2_Size, D2_Data_Set, Keep_Connection, Start_D2_on_Burst_Boundary)
FP_TRANSFER.Confirm
FP_TRANSFER_D1.Indicate (ULP-id, CCI, Status, D2_Size, D2_Offset, D1_Area_Size, D1_Data_Set)
FP_TRANSFER_D2.Indicate (ULP-id, CCI, Status, D2_Size, D2_Offset, D2_Data_Set)
FP_TRANSFER.Response

Control Link
FPSM_CONTROL.Request (Command, Command_Parameter)
FPSM_CONTROL.Confirm (Status)

Link Status
FPSM_STATUS.Request
FPSM_STATUS.Confirm (Status)
FPSM_STATUS.Indicate
FPSM_STATUS.Response

5.4 ULP data transfer service primitives

These primitives, as illustrated in figure 4, shall be used to transfer ULP data from the Source ULP to the Destination ULP.

5.4.1 ULP Identifiers

The ULP-id of the HIPPI-FP header designates the Destination ULP to which the data set is to be delivered.

NOTE - Identifiers registered at the time this part of ISO/IEC 11518 was approved include the following (shown in binary notation). Later registrations will be added as an amendment to this part of ISO/IEC 11518.

Processing of packets with unlisted ULPs is undefined.

00000010 = Reserved
00000011 = Reserved
00000100 = ISO 8802.2 Link Encapsulation
00000110 = IPI-3 Slave, i.e., IPI-3 Master to Slave
00000111 = IPI-3 Master, i.e., IPI-3 Slave to Master
00001000 = IPI-3 Peer
00001010 = HIPPI-FC mapping to Fibre Channel ULPs
1xxxxxxx = Locally assigned

5.4.2 FP_TRANSFER.Request

Issued by the Source ULP to request a data transfer. If a connection to the Destination specified by the CCI does not currently exist, then a connection will be established. At the completion of the transfer the connection may be broken unless Keep_Connection was specified. The packet format is defined in 7.2, and shown in figure 9.

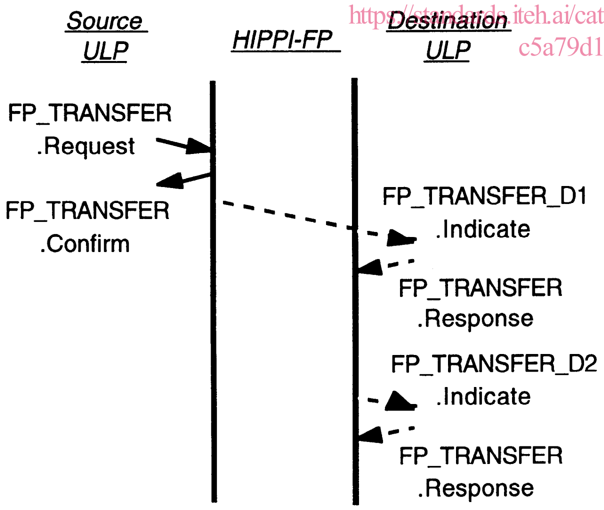

Figure 4 – Data transfer service primitives

Semantics – FP_TRANSFER.Request (CCI, ULP-id, D1_Size, D1_Data_Set, D2_Size, D2_Data_Set, Keep_Connection, Start_D2_on_Burst_Boundary)

The CCI is passed directly to the underlying HIPPI-PH.

The ULP-id identifies the Destination ULP. See 5.4.1.

D1_Size is the length, in bytes, of the ULP D1_Data_Set to be placed in the first burst of the packet. The maximum D1_Size shall be 1 016 bytes. A value of D1_Size equal to zero indicates a null D1_Data_Set and shall cause the FP header D1_Data_Set_Present bit to be set to 0. See 7.2.2.

D1_Data_Set is the ULP data set to be placed in the first burst, and delivered separately from the D2_Data_Set. The D1_Data_Set is intended for control information.

D2_Size is the length, in bytes, of the ULP data set to be placed in the remainder of the packet. The maximum determinate D2_Size is 4 294 967 294 bytes (2^{32} - 2). A D2_Size of hexadecimal FFFFFFFF shall mean that the length is indeterminate at the start of the transfer. Indeterminate length packets may be longer or shorter than the maximum determinate size. See B.1 for suggestions on transferring larger size, or indeterminate size, packets. The D2_Size shall be set to zero to indicate the absence of the D2_Data_Set.

D2_Data_Set is the ULP data set to be sent. Placement of the D2_Data_Set shall be governed by the Start_D2_on_Burst_Boundary parameter. The D2_Data_Set is intended for user data, or the whole data set if separate control information is not used. A ULP may deliver the D2_Data_Set to HIPPI-FP in multiple segments. To decrease latency and conserve buffers, implementations may start transmission before receiving all of these segments.

Keep_Connection true says that another ULP data set with the same routeing information is coming, and the physical HIPPI-PH connection should be maintained if possible. When Keep_Connection is false, the connection may be broken after this packet. Servicing FP_TRANSFER.Requests from other ULPs may also cause the connection to be broken, e.g., requests to different Destinations, or requests with Keep_Connection false. Keep_Connection is a local control parameter and is not passed to the Destination.

Start_D2_on_Burst_Boundary controls the starting location for the D2_Area. If true, then the D2_Area shall start at the beginning of the second HIPPI-PH burst. If false, then the D2_Area may start in the first burst.

Issued – The Source ULP issues this primitive to the Source HIPPI-FP to request the transfer of the ULP data set to the Destination.

Effect – The Source HIPPI-FP shall accept the ULP data set for transmission. The HIPPI-FP shall build an HIPPI-FP header, as specified in 7.2, and send the packet as a series of bursts to the Destination. If (1) the D1_Data_Set does not completely fill the first burst, and (2) Start_D2_on_Burst_Boundary = true, and (3) the underlying HIPPI-PH supports short first bursts, then this HIPPI-FP shall use a short first burst whose length is sufficient to completely contain the D1_Data_Set. If any of the above conditions are not met, then a 256-word first burst shall be used.

5.4.3 FP_TRANSFER.Confirm

This primitive acknowledges the FP_TRANSFER. Request from the Source ULP.

Semantics – FP_TRANSFER.Confirm (Status)

Status shall be

– Accept – the HIPPI-PH has completed the connection and accepted the packet for transmission.

– Reject – the Destination has rejected the connection request, no bursts were transmitted.

– Timeout – the Destination did not respond to the connection request within the timeout period. No bursts were transmitted. See A.4.7

Issued – The HIPPI-FP shall issue this primitive to the Source ULP to acknowledge the FP_TRANSFER.Request.

Effect – Unspecified

5.4.4 FP_TRANSFER.Indicate

These primitives indicate to the Destination ULP that the D1_Data_Set, or D2_Data_Set, of a packet, addressed to this particular ULP has been received from the Source.

Semantics –

FP_TRANSFER_D1.Indicate (ULP-id, CCI, Status, D2_Size, D2_Offset, D1_Area_Size, D1_Area)
FP_TRANSFER_D2.Indicate (ULP-id, CCI, Status, D2_Size, D2_Offset, D2_Data_Set)

ULP-id is the ULP to receive the data. See 5.4.1.

CCI is the CCI for the current connection, i.e., received with the PH_RING.Indicate connection request.

Status denotes whether the data set being delivered was received with errors. Status includes, but is not limited to, errors in the packet.

D2_Size is the length of the D2_Data_Set, in bytes, as received in the FP_Header. If D2_Size equals hexadecimal FFFFFFFF, then it is up to the ULP to determine the validity and actual length of the D2_Data_Set.

D2_Offset is the number of unused bytes from the start of the D2_Area to the first byte of the D2_Data_Set. The D2_Offset is used by the Source and Destination to keep proper word alignment on the D2_Data_Set so as to avoid shifting and copying the data to achieve alignment at the Destination. The D2_Offset allows the Source memory image of the D2_Data_Set, even if it does not start on a 64-bit word boundary, to be reproduced at the Destination.

D1_Area_Size is the size of the D1_Area being passed to the ULP. The actual size of the D1_Data_Set is self defining within the D1_Area.

D1_Area contains the D1_Data_Set. It is up to the Destination ULP to determine the size of the D1_Data_Set and extract it from the D1_Area. See 7.2.2.

The D2_Data_Set is the D2 ULP data being delivered to the ULP.

Issued – The Destination HIPPI-FP shall issue this primitive to the Destination ULP when a ULP data set has been received. A packet containing both the D1_Data_Set and the D2_Data_Set shall generate primitives for both the D1_Data_Set and the D2_Data_Set.

Effect – Unspecified

5.4.5 FP_TRANSFER.Response

This primitive acknowledges a FP_TRANSFER.Indicate for either the D1_Data_Set or the D2_Data_Set.

Semantics – FP_TRANSFER.Response

Issued – The Destination ULP issues this primitive to acknowledge receipt of the FP_TRANSFER.Indicate.

Effect – The Destination HIPPI-FP is enabled to issue another FP_TRANSFER. Indicate.

5.5 Control service primitives

These primitives, as illustrated in figure 5, shall be used to set parameters and control the interface. Note that a Control primitive can be initiated from either the Source or Destination.

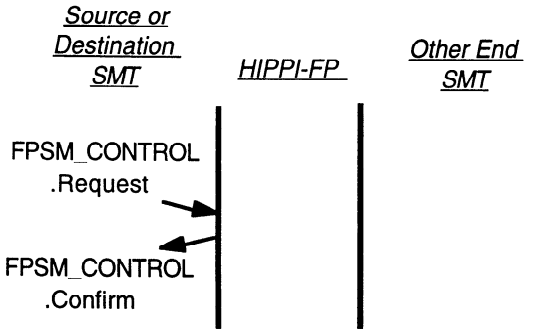

Figure 5 – Control service primitives

5.5.1 FPSM_CONTROL.Request

Issued by either the Source SMT or Destination SMT to set parameters, or otherwise control the local HIPPI-FP. Several functions are specified and others are left to specific implementations.

Semantics – FPSM_CONTROL.Request (Command, Command_Parameter)

The Command specifies the function to be performed. The parameters are specific to each function.

The Commands and Command_Parameters for the Source side include but are not limited to

Reset
Break Connection
Indicate Enable/Disable

Reset resets the HIPPI-FP, breaks any existing connections, and cancels any pending .Request primitives.

Break Connection breaks any existing connections.

Indicate Enable/Disable allows/disallows issuance of FPSM_ STATUS.Indicate primitives.

The Commands and Command_Parameters for the Destination side include but are not limited to

Reset
Break Connection
Allow/Disallow/Reject Connection
Indicate Enable/Disable

Reset resets the HIPPI-FP, breaks any existing connections, and cancels any pending .Request primitives.

Break Connection breaks any existing connections.

Allow/Disallow/Reject Connection. Sets Connection _Enable. Allow enables the Destination to make a connection. Disallow instructs the Destination to ignore connection requests. Reject instructs the Destination to respond to connection requests with rejected connection sequences.

Indicate Enable/Disable allows/disallows the HIPPI-FP to issue FPSM_ STATUS.Indicate primitives.

Issued – The Source or Destination SMT issues this primitive to perform some control function over the interface as a whole.

Effect – The HIPPI-FP shall perform the function specified.

5.5.2 FPSM_CONTROL.Confirm

This primitive acknowledges the FPSM_CONTROL.Request to the issuing SMT.

Semantics – FPSM_CONTROL.Confirm (Status)

Status reports the success or failure of the FPSM_CONTROL.Request commands.

Issued – The HIPPI-FP shall issue this primitive to the SMT when the command specified in the FPSM_ CONTROL.Request has been accepted.

Effect – Unspecified

5.6 Status service primitives

These primitives, as illustrated in figure 6, shall be used to obtain status information from the local HIPPI-FP. Note that a Status primitive can be initiated from either the Source or Destination, and shall only affect the local end of the interface.

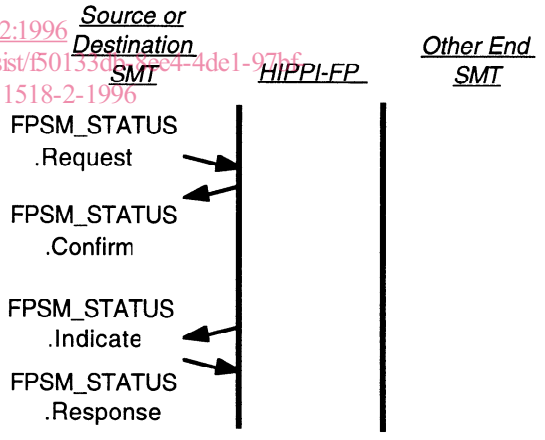

Figure 6 – Status service primitives

5.6.1 FPSM_STATUS.Request

Issued by either the Source SMT or Destination SMT to request a status report.

Semantics – FPSM_STATUS.Request

Issued – The SMT issues this primitive when it wishes to obtain the status of the HIPPI-FP.

Effect – The HIPPI-FP shall respond with a FPSM_STATUS.Confirm.

5.6.2 FPSM_STATUS.Confirm

This primitive replies to the previous FPSM_STATUS .Request with status information.

Semantics – FPSM_STATUS.Confirm (Status)

The Source side Status shall contain, but is not limited to

Errors
Current state of HIPPI connection
FPSM_STATUS.Indicates Enabled/Disabled

The Destination side Status shall contain, but is not limited to

Errors
Current state of HIPPI connection
Last CCI Received
Connections Allowed/Disallowed/Rejected
FPSM_STATUS.Indicates Enabled/Disabled

Issued – The HIPPI-FP shall issue this primitive to the SMT in response to a FPSM_STATUS.Request.

Effect – Unspecified

5.6.3 FPSM_STATUS.Indicate

This primitive informs the SMT entity that a major event has occurred that affects the operation of the HIPPI-FP.

Semantics – FPSM_STATUS.Indicate

Issued – The HIPPI-FP, when enabled, shall issue this primitive to the SMT whenever a major event is detected. Major events include but are not limited to

Detection of an illegal state transition

NOTE – If a FPSM_CONTROL.Request was accepted successfully but not completed, then an FPSM_STATUS .Indicate could be used to indicate completion.

Effect – Unspecified

NOTE – Upon receipt of this primitive the local SMT entity should issue a FPSM_STATUS.Request to read status and determine which event occurred.

5.6.4 FPSM_STATUS.Response

This primitive acknowledges the FPSM_STATUS.Indicate.

Semantics – FPSM_STATUS.Response

Issued – The SMT issues this primitive to acknowledge receipt of the FPSM_STATUS.Indicate.

Effect – The HIPPI-FP, if enabled, is allowed to issue another FPSM_STATUS.Indicate.

6 HIPPI-PH to HIPPI-FP services

A summary of the primitives used to connect the HIPPI-FP to the HIPPI-PH is included here. The complete specification of the primitives is contained in the HIPPI Mechanical, Electrical, and Signalling Protocol Specification (HIPPI-PH) document.

Initiate a Connection
- PH_RING.Request (CCI)
- PH_RING.Confirm
- PH_RING.Indicate (CCI)
- PH_RING.Response

Complete the Connection
- PH_ANSWER.Request (Accept/Reject)
- PH_ANSWER.Confirm
- PH_ANSWER.Indicate (Accept/Reject)
- PH_ANSWER.Response

Packet Control
- PH_PACKET.Request (Begin/End)
- PH_PACKET.Confirm (Accept/Reject)
- PH_PACKET.Indicate (Begin/End,Status)
- PH_PACKET.Response

Burst Transfer
- PH_TRANSFER.Request (Length,Burst)
- PH_TRANSFER.Confirm (Accept/Reject)
- PH_TRANSFER.Indicate (Status,Length,Burst)
- PH_TRANSFER.Response

Terminate the Connection
- PH_HANGUP.Request
- PH_HANGUP.Confirm
- PH_HANGUP.Indicate
- PH_HANGUP.Response