

IEEE Standard for Data Delivery and Logical Channels for IEEE 1284 Interfaces

Sponsor

Microprocessor and Microcomputer Standards Committee
of the
IEEE Computer Society

Approved 21 June 2000

IEEE-SA Standards Board

Abstract: A device to carry on multiple, concurrent exchanges of data and/or control information with another device across a single point-to-point link allowed by the packet protocol is described in this standard. The protocol is not a device control language. The protocol provides basic transport-level flow control and multiplexing services. The multiplexed information exchanges are independent, and blocking of one has no effect on any other. The protocol shall operate over interfaces such as described in IEEE Std 1284-2000.

Keywords: channelized protocol, IEEE 1284, logical channels, multiple logical channels (MLC), transport protocol

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 15 December 2000. Printed in the United States of America.

Print: ISBN 0-7381-2480-X SH94849
PDF: ISBN 0-7381-2481-8 SS94849

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

IEEE is the sole entity that may authorize the use of certification marks, trademarks, or other designations to indicate compliance with the materials set forth herein.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 1284.4-2000, IEEE Standard for Data Delivery and Logical Channels for IEEE 1284 Interfaces.)

This standard was formally started as an IEEE project in January 1996. Members of the IEEE P1284.4 Working Group would like to thank the Hewlett-Packard Company, and the IEEE 1284.1 and IEEE 1284.3 committees for their efforts in establishing the foundation for this standard and allowing this committee to include their work as the basis for much of this standard.

At the time this standard was completed, the key contributors to the IEEE P1284.4 Working Group were as follows:

	Larry A. Stein, <i>Chair</i>	
	Mark T. Edmead, <i>Secretary</i>	
	Bob McComiskie, <i>Secretary</i>	
	Brian Batchelder, <i>Editor</i>	
	Walt Scheiderich, <i>Editor</i>	
Jim Anderson	Paul Lindemuth	Bill Stanley
Lee Farrel	Raymond Lutz	Ron Talwalker
Robert Gross	Mike Moldovan	Randy Turner
Joe Halpern	Ron Norton	William Wagner
Reed Hinkel	Rick Pennington	Andrew Warner
Monte Johnson	Greg Shue	Forrest D. Wright
Laurie Lasslo	Lance Spaulding	Atsushi Yuki
Greg LeClair		Steve Zilles

The following members of the balloting committee voted on this standard:

John L. Cole	Donald N. Heirman	David Rockwell
Dante Del Corso	Thomas M. Kurihara	Marius Seritan
Stephen L. Diamond	Laurie Lasslo	Kerry Shore
Sourav K. Dutta	Franck Leprevost	Howard Stanley
Mark T. Edmead	Joseph R. Marshall	Larry A. Stein
Charlotte Ersmarker	Mike Moldovan	Forrest D. Wright
Julio Gonzalez-Sanz	Jeff Rackowitz	Janusz Zalewski

When the IEEE-SA Standards Board approved this standard on 21 June 2000, it had the following membership:

Donald N. Heirman, *Chair*
James T. Carlo, *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Mark D. Bowman
Gary R. Engmann
Harold E. Epstein
H. Landis Floyd
Jay Forster*
Howard M. Frazier
Ruben D. Garzon

James H. Gurney
Richard J. Holleman
Lowell G. Johnson
Robert J. Kennelly
Joseph L. Koepfinger*
Peter H. Lips
L. Bruce McClung
Daleep C. Mohla

James W. Moore
Robert F. Munzner
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Donald W. Zipse

*Member Emeritus

Also included is the following nonvoting IEEE-SA Standards Board liaison:

Alan Cookson, *NIST Representative*
Donald R. Volzka, *TAB Representative*

Greg Kohn
IEEE Standards Project Editor

Contents

1. Overview.....	1
1.1 Scope.....	1
1.2 Purpose.....	1
2. Definitions.....	1
2.1 General terminology	1
2.2 IEEE 1284.4-specific terminology.....	2
3. Features and compliance.....	4
3.1 Overview.....	4
3.2 Features.....	4
3.3 Compliance criteria.....	5
3.4 Multiple logical channels (MLC) compatibility criteria.....	5
4. Theory of operation of the IEEE 1284.4 protocol	5
4.1 Overview.....	5
4.2 General packet structure	5
4.3 Communication procedures	6
4.4 Service discovery	9
4.5 Data transfer and flow control	9
5. IEEE 1284.4 transactions.....	16
5.1 Overview.....	16
5.2 Transaction summary.....	16
5.3 Conversation control transactions.....	18
5.4 Connection control transactions.....	23
6. Data link service requirements.....	36
6.1 Overview.....	36
6.2 Required services	36
Annex A (informative) Bibliography.....	38
Annex B (normative) Service names registry	39
Annex C (informative) IEEE 1284.4 architecture	41
Annex D (informative) Example IEEE 1284.4 application programming interface (API).....	42
Annex E (informative) Implementation issues	50

IEEE Standard for Data Delivery and Logical Channels for IEEE 1284 Interfaces

1. Overview

1.1 Scope

The packet protocol described by this standard allows a device to carry on multiple, concurrent exchanges of data and/or control information with another device across a single point-to-point link. The protocol is not a device control language. The protocol provides basic transport-level flow control and multiplexing services. The multiplexed information exchanges are independent, and blocking of one has no effect on any other. The protocol shall operate over IEEE Std 1284-2000 [B2]¹ interfaces and may operate over other interfaces, such as RS-232, Universal Serial Bus (USB), and that of IEEE Std 1394-1995 [B5].

1.2 Purpose

IEEE Std 1284-2000 [B2] defines and describes an updated personal computer (PC) parallel interface by adding multiple modes of operation that provide for higher speed, bidirectional communication between devices. IEEE Std 1284-2000 [B2] does not provide anything beyond a physical protocol. IEEE Std 1284.4-2000 specifies a point-to-point protocol with one or more layers above the physical interface and below the application. These layers take on the functions and characteristics of the transport and session layers of the open systems interconnect (OSI) model.

2. Definitions

2.1 General terminology

The following terms and acronyms are used in this standard. The definitions are not intended to be absolute, but they do reflect the use of the terms in this standard.

2.1.1 big-endian: A data format where the most significant byte of a multibyte object is at the lowest address and the least-significant byte is at the highest address.

¹The numbers in brackets correspond to those of the bibliography in Annex A.

2.1.2 byte: An 8 bit value.

NOTE—In this standard, bits are read left to right with the most significant bit (MSB) labeled 7 to least significant bit (LSB) labeled 0 as shown in Figure 1.

Bit 7—MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0—LSB
-----------	-------	-------	-------	-------	-------	-------	-----------

Figure 1—Byte layout

2.1.3 connection: A persistent communications path between two endpoints.

2.1.4 device control language: A language used to monitor and/or control the state of a device.

2.1.5 endpoint: A consumer or producer of data on a communication link.

2.1.6 flow control: The function performed by a receiving entity to limit the amount of data that is sent by a transmitting entity.

2.1.7 logical channel: One connection over a single physical communication link. There may be multiple logical channels over a single physical communication link.

2.1.8 open systems interconnect (OSI) model: A seven-layer network communications model developed by an International Organization for Standardization (ISO) subcommittee that governs communications interchange between systems. The model is an internationally accepted framework of standards for intersystem communications.

2.1.9 packet: A group of bytes, including address, data, and control elements.

2.1.10 point-to-point communications: Communications that take place between exactly two devices.

2.1.11 service discovery: The function of providing transport clients with the ability to dynamically query service availability within a peer transport entity.

2.2 IEEE 1284.4-specific terminology

2.2.1 application: A software component that uses the IEEE 1284.4 transport.

2.2.2 blocked channel: A channel with at least one blocked endpoint.

2.2.3 blocked endpoint: An endpoint that is unable to send data on a connection due to lack of credit. A blocked endpoint becomes unblocked when it is granted credit to send data on that connection.

2.2.4 channel: An independently flow-controlled connection between a socket on the primary device and a socket on the secondary device. It provides a logical conduit for moving data between the two endpoints.

2.2.5 client: Name used to refer to the software component on one device that uses the services provided by a server on another device. The communication between the client and the server is managed by the protocol defined in this standard.

2.2.6 command: The instruction sent from an initiator to a target directing the target to execute a specified process.

2.2.7 conversation: Canonical name given to the interactions between two IEEE 1284.4 peers during the period of time bracketed between corresponding **Init** and **Exit** transactions.

2.2.8 credit: A value issued by an IEEE 1284.4 implementation on behalf of an endpoint indicating the number of packets that may be received by that endpoint.

2.2.9 end-of-message packet: The last packet of a message in the data stream.

2.2.10 initiator: The IEEE 1284.4 implementation that initiated a transaction by sending a command.

2.2.11 message: A set of ordered data (possibly empty) that includes a message boundary indication. Message data may span multiple packets. A packet shall not hold data from more than one message.

2.2.12 multiple outstanding transactions: A state where more than one transaction has been issued and is pending completion.

2.2.13 out-of-band packet: A packet of exceptional data in the data stream. This type of packet may contain special information relating to the data at this position in the data stream, for example, “end of job.” Information contained in the out-of-band packet is outside of the scope of IEEE Std 1284.4-2000.

2.2.14 piggyback credit: Used to describe the inclusion of credit inside an IEEE 1284.4 packet header. By using piggyback credit, IEEE 1284.4 can reduce the overhead of **Credit** transactions when processing a channel on which data is flowing bidirectionally.

2.2.15 primary device: The device that issued the **Init** transaction that started the current IEEE 1284.4 conversation. There is no priority implied by the assignment of primary and secondary devices once the conversation has been established. The assignment of primary status is used only to identify the device that contains the primary sockets.

2.2.16 primary socket identifier (PSID): A socket number identifying a particular endpoint on the primary device.

2.2.17 reply: The response sent from a target to an initiator indicating that the target has successfully or unsuccessfully executed the process specified by the command originally sent from the initiator to the target.

2.2.18 reserved: Any protocol elements identified as “reserved” are intended for future standardization. Reserved elements shall not be used. Reserved fields or bits shall be set to 0 and shall not be checked. Values not defined in IEEE Std 1284.4-2000 are reserved.

2.2.19 secondary device: The device that was the target of the **Init** transaction that started the current IEEE 1284.4 conversation. There is no priority implied by the assignment of primary and secondary devices once the conversation has been established. The assignment of secondary status is used only to identify the device that contains the secondary sockets.

2.2.20 secondary socket identifier (SSID): A socket number identifying a particular endpoint on the secondary device.

2.2.21 server: The software component on one device that provides services for use by clients on the same or another device. The transfer of data between the client and the server is managed by the protocol defined in IEEE Std 1284.4-2000.

2.2.22 service name: The canonical name of a particular service available for use by an IEEE 1284.4 client.

2.2.23 socket: Canonical name given to an IEEE 1284.4 endpoint to which, or from which, data is transmitted.

2.2.24 socket identifier (socket ID): A byte used to uniquely identify the socket. A socket ID can be well known or it can be dynamically assigned.

2.2.25 target: The peer IEEE 1284.4 implementation to which the initiator's command is being sent.

2.2.26 transaction: A IEEE 1284.4 command exchange, including both the command and the reply.

2.2.27 transaction channel: The channel used for IEEE 1284.4 commands and replies. The primary socket identifier and the secondary socket identifier for the transaction channel are both 0x00.

3. Features and compliance

3.1 Overview

It is the intention of this standard to complement various types of bidirectional, point-to-point links, such as those specified in IEEE Std 1284-2000 [B2], by providing a simple, flexible, and scaleable transport protocol. For the purposes of this standard, a point-to-point link may be a virtual connection over other physical topologies. It is assumed that this point-to-point link is error free.

A device that implements IEEE 1284.4 functionality provides the ability to carry on multiple, concurrent, independent exchanges of information, whether control or data, over a single point-to-point link. This service is provided by use of the packet-based, nonblocking, flow control system defined in this standard.

3.2 Features

IEEE Std 1284.4-2000 specifies a packet-based, bidirectional, connection-oriented, peer-to-peer communication protocol for a point-to-point link encompassing the following features:

- a) Multiple, concurrent, logical channels: nonblocking, independent information exchange
- b) Flow control to keep the channels independent
 - 1) Negotiation of packet sizes
 - 2) Transmission rate control (also known as pacing)
 - 3) Specification of minimum and maximum packet sizes
- c) Service discovery
 - 1) Dynamic query and mapping of channel address space
 - 2) Service names maintained in a third-party, open, persistent registry (see Annex B)
- d) Low system overhead: a small, fixed header allowing easy encoding and decoding of packets
- e) Extensibility
 - 1) Control byte provides simple out-of-band and end-of-message signaling
 - 2) Within a conversation, services can be assigned to previously unallocated sockets
- f) Information exchange independence: the transport protocol is not dependent upon the data being exchanged
- g) Physical link independence: the protocol can be used in conjunction with any physical link, provided a data link layer of sufficient functionality is available (see Clause 6)

3.3 Compliance criteria

An IEEE 1284.4-compliant device shall provide all commands and functionality in the following clauses:

- Clause 4: Theory of operation of the IEEE 1284.4 protocol
- Clause 5: IEEE 1284.4 transactions

A compliant device shall use a data link that meets the requirements enumerated in Clause 6.

IEEE 1284.4-compliant devices are not required to implement multiple outstanding transactions (see 4.5.6).

3.4 Multiple logical channels (MLC) compatibility criteria

IEEE 1284.4 is based upon the MLC protocol developed by Hewlett-Packard Company. MLC devices are not necessarily compliant with IEEE 1284.4. For details on the MLC protocol, refer to the bibliography of Annex A.

4. Theory of operation of the IEEE 1284.4 protocol

4.1 Overview

IEEE 1284.4 is a credit-based transport protocol. It provides a mechanism to establish, configure, and control communication between two endpoints. It provides this functionality through a well-defined packet structure and transaction protocol. IEEE 1284.4 transactions are symmetrical: each command has a corresponding reply.

IEEE 1284.4 transports client data on logical channels. These logical channels are multiplexed on a single physical link. Client data is exchanged over data channels. IEEE 1284.4 transactions are exchanged over the IEEE 1284.4 transaction channel. All client data and IEEE 1284.4 transactions are encapsulated in packets. Packet transfer is managed using a credit-based flow control process to pace the exchange.

IEEE 1284.4 also provides a service discovery mechanism to dynamically map service names to socket numbers.

4.2 General packet structure

All IEEE 1284.4 packets follow the general structure illustrated in Table 1. Each packet shall contain the 6-byte header and zero or more data bytes. The packet header contains the five fields detailed in Table 2: primary socket identifier (PSID), secondary socket identifier (SSID), Length, Credit, and Control. To eliminate ambiguity, all numbers in the following descriptions that are in hexadecimal are designated beginning with “0x” and are in big-endian byte order. Packet fields are listed in the order they are transmitted reading from left to right. The length includes the header. The maximum amount of data that may be transmitted in the payload field is 65 529 bytes (0xFFFF9). The offset and length values specified in Table 2 are unsigned integers.

If PSID and SSID indicate any other channel, the payload field contains application data for that channel.

Table 1—IEEE 1284.4 packet structure

Header					Payload
PSID 1 byte	SSID 1 byte	Length 2 bytes	Credit 1 byte	Control 1 byte	0–0xFFF9 bytes

Table 2—IEEE 1284.4 packet field descriptions

Name	Offset	Length	Description								
PSID	0x00	1 byte	An unsigned byte indicating the primary socket ID of the packet.								
SSID	0x01	1 byte	An unsigned byte indicating the secondary socket ID of the packet.								
Length	0x02	2 bytes	An unsigned word that specifies the length, in bytes, of the entire packet. The length includes the packet header. Valid values range from 0x0006 to 0xFFFF.								
Credit	0x04	1 byte	A piggyback credit field that specifies the number of additional packet credits the sender of the packet is issuing to the receiver of the packet. These credits are for data to be sent in the future by the receiver of this packet on the channel indicated by the PSID and SSID fields.								
Control	0x05	1 byte	A bit field containing information for controlling operation of the IEEE 1284.4 protocol. It is a bit-encoded field with the following definitions: <table><thead><tr><th>Bit(s)</th><th>Definition</th></tr></thead><tbody><tr><td>7-2</td><td>Reserved</td></tr><tr><td>1</td><td>0 = Normal Packet 1 = End-of-message packet</td></tr><tr><td>0</td><td>0 = Normal packet 1 = Out-of-band packet</td></tr></tbody></table>	Bit(s)	Definition	7-2	Reserved	1	0 = Normal Packet 1 = End-of-message packet	0	0 = Normal packet 1 = Out-of-band packet
Bit(s)	Definition										
7-2	Reserved										
1	0 = Normal Packet 1 = End-of-message packet										
0	0 = Normal packet 1 = Out-of-band packet										
Payload	0x06	0–65 529 bytes	If PSID and SSID indicate the IEEE 1284.4 transaction channel, the payload field contains an IEEE 1284.4 command or reply.								

4.3 Communication procedures

4.3.1 IEEE 1284.4 conversation management

A conversation is the set of interactions between two IEEE 1284.4 peers during the period of time bracketed between corresponding **Init** and **Exit** transactions.

The following subclauses describe how IEEE 1284.4 conversation management is handled, including the IEEE 1284.4 transactions that are used to establish and terminate the conversation. Once the conversation is

established, 4.3.2 explains how the channels are opened, data is transferred, and channels are closed. Details of the transactions are discussed in Clause 5.

4.3.1.1 Conversation initiation

When one IEEE 1284.4 peer wishes to establish an IEEE 1284.4 conversation, it does so by initiating an **Init** transaction on the IEEE 1284.4 transaction processor channel. The target of the **Init** transaction will complete the transaction.

The device that generates the successful **Init** transaction becomes the primary device for the duration of the IEEE 1284.4 conversation. The device that completes the **Init** transaction becomes the secondary device for the duration of the IEEE 1284.4 conversation.

During this exchange each side of the link indicates which revision of the protocol it supports. All IEEE 1284.4 implementations shall support the 0x20 revision of the protocol defined by this standard.

4.3.1.2 Conversation termination

The **Exit** transaction is used as a companion to the **Init** transaction. It is used to terminate IEEE 1284.4 conversations. The normal termination process is to first complete all outstanding transactions and close all open channels, and then to initiate an **Exit** transaction.

4.3.2 Channels

A channel is defined as a logical connection between two socket endpoints. The minimum number of supported channels is one: the IEEE 1284.4 transaction channel. The maximum number of channels is 65 026 $((255 \times 255) + 1)$. Implementations may support fewer channels.

4.3.2.1 Socket IDs

A PSID and SSID are used to identify the endpoints of a specific channel on which communication occurs. Socket IDs range from 0x00 to 0xFF. Communication between IEEE 1284.4 peers occurs on the IEEE 1284.4 transaction channel. The PSID and SSID for the transaction channel are both 0x00.

Other sockets are dynamically allocated by IEEE 1284.4 peers and assigned to applications wishing to communicate with services on the other side of the link. Service discovery (see 4.4) provides a mechanism used to locate those services. Unlike other channels, the transaction channel is assumed to be open when IEEE 1284.4 is initialized. No open or close of the transaction channel is allowed. See 4.3.2.2 and 4.5.5 for details on the transaction channel.

4.3.2.2 Packet size

The maximum size of packets in both primary-to-secondary and secondary-to-primary directions on the IEEE 1284.4 transaction channel is fixed at 64 bytes (6 byte header plus up to 58 bytes for the commands and replies). Fewer than 64 bytes may be sent in an IEEE 1284.4 transaction packet.

The maximum packet size on all other channels is limited by the two-byte length field to a maximum of 65 535 bytes (6 byte header plus up to 65 529 bytes of data). The actual maximum packet sizes supported will be implementation specific. When a channel is opened, the maximum sizes of both primary-to-secondary and secondary-to-primary packets are negotiated and can be different. Packets smaller than the negotiated maximum size may be sent.

The minimum packet size is 6 bytes (a 6 byte header plus 0 bytes of data).

4.3.2.3 Opening a channel

Once the IEEE 1284.4 conversation has started, channels may be established between endpoints on the IEEE 1284.4 peers. Channels are established using the **OpenChannel** transaction. Data packets shall only be transmitted on a channel while that channel is open.

4.3.2.4 Channel credit

Credit is the fundamental mechanism by which the flow of data on IEEE 1284.4 channels is controlled (see 4.5). A receiving IEEE 1284.4 peer, having knowledge of how many packets it can accept, uses credit to control the number of packets the sender can transmit across each open channel. Using credit to control the flow of data allows IEEE 1284.4 to avoid the overhead of interlocked send-receive-acknowledge transmissions. This provides a simple mechanism for the receiver to prevent the sender from overwhelming its resources with packet transmissions. Credit is granted by **OpenChannel**, **Credit**, or **CreditRequest** transactions, or by piggyback credit in a packet.

4.3.2.5 Channel data exchange

Once a channel is open and credit has been granted, data packets may be freely exchanged between the two endpoints as long as the sender has credit to do so. Credit may be piggybacked in the data packets. If the data stream is bidirectional (i.e., data is flowing in a primary-to-secondary direction as well as a secondary-to-primary direction), piggybacking credit reduces the crediting traffic.

4.3.2.6 Closing a channel

When either IEEE 1284.4 peer determines the channel is no longer necessary, it closes the channel by initiating a **CloseChannel** transaction.

4.3.3 Special data packets

4.3.3.1 End-of-message packets

A message is a set of ordered data (possibly empty) that includes a message boundary indication. Message data may span multiple packets. A packet shall not hold data from more than one message. Setting the “end-of-message packet” flag in the packet header indicates the message boundary. A packet with the end-of-message flag set may or may not contain data. The message boundary is after the last byte of data in the packet, or after the last byte of data previously transferred if the flagged packet has no data. The sending client indicates which packets are to be marked end-of-message. The IEEE 1284.4 peer transport shall pass the end-of-message indication to the receiving client. All data within each channel shall be delivered in order. All other processing of end-of-message data shall be in the same manner as non end-of-message data. End-of-message packets are not supported on the IEEE 1284.4 transaction channel.

4.3.3.2 Out-of-band packets

Packets can be marked as out-of-band by setting the “out-of-band packet” flag in the packet header. The sending client indicates which packets are to be marked out-of-band. The IEEE 1284.4 peer transport will pass the out-of-band indication to the receiving client. Out-of-band data shall not be split across multiple packets. Out-of-band data shall not be combined with in-band data. All data within each channel shall be delivered in order, regardless if it is in-band or out-of-band. All other processing of out-of-band data shall be in the same manner as non out-of-band data. Out-of-band packets are not supported on the IEEE 1284.4 transaction channel.

4.3.4 Errors

4.3.4.1 Error indications

Two mechanisms are provided for communicating errors to the IEEE 1284.4 peer. Errors detected within a transaction can be communicated in the transaction reply. Other errors can be communicated using the **Error** packet.

4.3.4.2 Response time

There are no response time requirements specified by the IEEE 1284.4 protocol. Implementers should consider using timers to avoid potential deadlock situations.

4.3.4.3 Other error conditions

Packets other than the **Init** command received outside of a conversation shall be ignored. The implementer shall handle any other unspecified error conditions.

4.4 Service discovery

4.4.1 Overview

Service discovery provides a standard method for IEEE 1284.4 client applications to locate a particular service application. It provides the ability to dynamically determine service name to socket ID mapping.

4.4.2 Service names

Canonical names of services are maintained by an administrative authority. Names may be registered by vendors through the administrative authority, along with a description of the service functionality. IEEE 1284.4 service names shall be expressed within the 7-bit ASCII character set with a maximum length of 40 characters. Service names are limited to the set of uppercase letters, digits, and the punctuation character hyphen. They must start with a letter, and end with a letter or digit. The registry process is defined in Annex B.

4.4.3 Service discovery protocols

Service discovery is supported through the **GetSocketID** and **GetServiceName** transactions. These complementary transactions allow name-to-socket and socket-to-name conversions.

4.5 Data transfer and flow control

4.5.1 Introduction

Within the context of IEEE 1284.4, flow control ensures that a sender only sends data when the receiver is ready to accept the data. This prevents a channel from blocking other channels. It also allows the transport layer to avoid retransmitting data. To provide flow control in IEEE 1284.4, the receiver informs the sender how much data can be sent. The sender then limits the amount it sends to this value.

The maximum amount of data that may be sent is communicated in two steps. First, the maximum size of the packets to be exchanged is negotiated. Second, during data transfer, the receiver of the data communicates to the sender the number of packets that can be sent (credits). The product of the maximum packet size and the number of credits specifies the maximum amount of data that can be sent by the sender before additional

credit is granted. The maximum size of packets remains constant while the channel is open. The receiver enables further data transfer by granting more credits to the sender.

4.5.2 Packet size negotiation

Packet size negotiation sets the maximum size of packets, per channel, in both the primary-to-secondary and secondary-to-primary directions. This is done when the connection is established, using the **OpenChannel** transaction. The packet size can differ with direction. A zero-length packet size indicates that no data packets will be sent in that direction. The maximum size of packets remains constant while the channel is open. Packet size negotiation allows optimization of data transfers within the current IEEE 1284.4 conversation.

4.5.3 Credit

The receiver grants credit to the sender to indicate that the receiver is prepared to accept a specific number of data packets. Credit is associated with packets on a one-for-one basis; i.e., each credit grants the sender permission to send one packet. An endpoint may only send data across a channel if it has the credit to do so.

Credit used for IEEE 1284.4 communications is accumulative. When a sender is granted credit, it adds the additional credit to any unused credit it has already been granted. As the sender sends packets, it subtracts the corresponding credits from its available credit. Once a sender's credit reaches zero, it shall not send any more packets until it is granted more credit for that channel.

Initial credit can be granted in the **OpenChannel** transaction. Once a channel is open, credit can be granted via a **Credit** transaction or requested via a **CreditRequest** transaction. To avoid the overhead of these explicit credit commands, credit can also be granted by piggybacking credit in IEEE 1284.4 packets. All outstanding credit for a given channel shall be discarded upon the completion of a **CloseChannel** transaction.

Flow control of the IEEE 1284.4 conversation is controlled by IEEE 1284.4 peers managing the credit by using the modes detailed in 4.5.4 and 4.5.5, and preventing deadlock as described in 4.5.7.

4.5.4 IEEE 1284.4 data flow control

The sender controls a parameter to manage the receiver's granting of credit. The sender sets this parameter using a field in the **OpenChannel** and **CreditRequest** commands. The receiver grants credit via the **OpenChannelReply**, **Credit**, or **CreditRequestReply** commands, or by piggybacking credit in a data packet.

The credit control parameter is called **MaximumOutstandingCredit**. **MaximumOutstandingCredit** is the maximum amount of credit that the sender will need at any given time. The receiver shall attempt to maintain the sender's credit balance equal to the value of this parameter.

The initial value of the **MaximumOutstandingCredit** parameter is set by the **OpenChannel** transaction. The sender may subsequently change the credit parameter at any time by sending a **CreditRequest** command. Changing the **MaximumOutstandingCredit** parameter does not change the available credit that the sender has already accumulated.

The **MaximumOutstandingCredit** parameter defines the following modes:

- a) *"No credit" mode*: When the **MaximumOutstandingCredit** parameter is set to 0x0000 the sender is asking the receiver to send no credit.

This may be used at the start of communications by an **OpenChannel** command when the sender wants to open a channel prior to having data ready to send and does not want to burden the receiver with having to supply credit until it is needed. It may also be used at the end of a channel packet exchange; e.g., the sender wishes to keep the channel open but has no more data to send at that time.

- b) *“Unlimited credit” mode*: When the MaximumOutstandingCredit parameter is set to 0xFFFF, the sender is asking the receiver to send as much credit as possible.

This may be used when the sender has a large or unknown number of packets to send. As it approaches the end of job, the sender may monitor the number of packets left to be sent so that it can switch to “no credit” mode.

- c) *“Maintain credit” mode*: When the MaximumOutstandingCredit parameter is in the range 0x0001 to 0xFFFF, the receiver will try to keep the sender’s outstanding credit at the amount specified by the MaximumOutstandingCredit parameter.

This mode may be used when the sender will be sending bursts of packets, where the burst size will never exceed the MaximumOutstandingCredit parameter. Since the maximum number of credits needed at any time is known, the receiver can efficiently manage its buffers.

The sender is not required to use all of the above modes. For example, the sender may use unlimited credit mode for every channel and allow the receiver to clean up resources only as channels are closed. The sender may also start in “no credit” mode, switch to “maintain credit” mode to send bursts of packets, and finish by switching back to “no credit” mode. The three modes provide the sender with flexibility in managing its need for credit and not burden the receiver with unnecessary requests or crediting.

The credit modes are hints for the receiver. The receiver may grant more than the requested number of credits, even in “no credit” mode.

Table 3 and Table 4 are examples of flow control of IEEE 1284.4 data packets associated with opening a channel between primary and secondary IEEE 1284.4 peers, granting initial credit, sending data, granting additional credit, and closing the channel. Time increases with each row as one moves down the tables from top to bottom. In addition, only the sending of packets is shown. The reception of packets is implied (e.g., an **Init** command is sent by the primary IEEE 1284.4 peer but the tables do not show the secondary IEEE 1284.4 peer receiving it). Table 3 and Table 4 do not show the transactions required to establish the IEEE 1284.4 conversation. Transaction channel flow control is detailed in 4.5.5.

Table 3 is an example of a unidirectional channel. Unidirectional channels are used when data flows in only one direction across the link. Common uses of unidirectional channels include print data and scan data.

Table 4 is an example of a bidirectional channel. Bidirectional channels are used when data flows in both directions across the link. Common uses of bidirectional channels include control, status, and other request-reply service.

4.5.5 IEEE 1284.4 transaction channel flow control

Flow control for the transaction channel is similar to flow control for data channels, with the following exceptions:

- a) IEEE 1284.4 command and reply packets have a maximum size of 64 bytes.
- b) The transaction channel is always open. Neither **OpenChannel** nor **CloseChannel** transactions of the transaction channel are permitted.
- c) After a successful **Init**, there is credit on the transaction channel for one command in each direction, regardless of the previous credit values. Commands contain piggyback credit of exactly one (1), which is used for the reply to that command. Replies contain piggyback credit of exactly one (1), which is used for the next command.
- d) The transaction channel defaults to “maintain credit” mode, with a MaximumOutstandingCredit of one (1) (see 4.5.4).

- e) Transaction channel credit shall be used to complete outstanding transactions before it is used to issue new transactions. Only when there is more credit than there are pending replies can a new transaction be issued.
- f) **Error** and **Init** transactions can always be sent. Credit balances are unaffected by the **Error** command. The **Init** transaction resets the credit balance to one (1).

Table 3—Flow control of IEEE 1284.4 data packets (unidirectional panel)

Time	Primary IEEE 1284.4 peer	Secondary IEEE 1284.4 peer	Primary peer data credit	Secondary peer data credit
0			0	0
1	OpenChannel (unlimited credit mode)	0	0	
2		OpenChannelReply grant 5 credits	$0 + 5 = 5$	0
3	Primary to secondary Data Packet 1		$5 - 1 = 4$	0
4	Primary to secondary Data Packet 2		$4 - 1 = 3$	0
5	Primary to secondary Data Packet 3		$3 - 1 = 2$	0
6		Credit grant 2 credits	2	0
7	CreditReply		$2 + 2 = 4$	0
8	Primary to secondary Data Packet 4		$4 - 1 = 3$	0
9	Primary to secondary Data Packet 5		$3 - 1 = 2$	0
10	CloseChannel		2	0
11		CloseChannelReply	0	0