Network Working Group
Request for Comments: 2126
Category: Standards Track

Y. Pouffary
Digital Equipment Corporation
A. Young
ISODE Consortium

March 1997

ISO Transport Service on top of TCP (ITOT)

Status of the Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document is a revision to STD35, RFC1006 written by Marshall T. Rose and Dwight E. Cass. Since the release of RFC1006 in May 1987, much experience has been gained in using ISO transport services on top of TCP. This document refines the protocol and will eventually supersede RFC1006.

This document describes the mechanism to allow ISO Transport Services to run over TCP over IPv4 or IPv6. It also defines a number of new features, which are not provided in RFC1006.

The goal of this version is to minimise the number of changes to RFC1006 and ISO 8073 transport protocol definitions, while maximising performance, extending its applicability and protecting the installed base of RFC1006 users.

Table of Contents

1. Introduction, Motivation....................................2
2. The Model...3
2.1 ISO Transport Model..3
2.2 ISO Transport over TCP (ITOT) Model........................4
2.3 Overview of Protocol and Service...........................5
3 Service Definition...5
3.1 Transport Service Definition...............................5
3.1.1 Transport Service Definition Primitives..................6
3.2 Network Service Definition.................................7
3.2.1 ISO 8348 CONS primitives.................................7
3.2.2 TCP Service primitives...................................8
3.2.3 Mapping TCP as a Network Service Provider................8

3.2.3.1 Network Connection Establishment........................8
3.2.3.2 Network Data Transfer...................................9
3.2.3.3 Network Connection Release.............................10
4. Transport Protocol Specification............................10
4.1 Class 0 over TCP...10
4.1.1 Connection Establishment.................................11
4.1.2 Data Transfer..11
4.1.3 Connection Release.......................................11
4.2 Class 2 over TCP...12
4.2.1 Connection Establishment.................................12
4.2.2 Data Transfer..13
4.2.3 Connection Release.......................................15
4.3 TPKT Packet Format...15
5. Address representations.....................................16
5.1 String representation of ITOT access point addresses.......17
5.2 OSI Network Address encoding...............................17
6. Notes to Implementors.......................................17
6.1 TCP Connection Establishment...............................17
6.2 TCP Data transfer..17
6.3 Class negotiation..18
6.4 Default maximum TPDU size..................................18
6.5 Class 0 TPDU bit encoding..................................18
6.6 Class 2 Options..19
6.7 Class 2 Expedited Data Acknowledgement.....................21
6.8 Class 2 Normal Data and Expedited Data handling............21
6.9 Class 2 Forward Connection procedure.......................22
6.10 TPKT..22
7. Rationale - Interoperability with RFC1006...................22
8. Security Considerations.....................................23
Acknowledgements...23
References...23
Authors' Addresses...25

1. Introduction, Motivation

There are two basic approaches which can be taken when "porting" ISO applications to TCP/IP ([RFC793],[RFC791]) and IPv6 [IPV6] environments. One approach is to port each individual application separately, developing local protocols on top of TCP. A second approach is based on the notion of layering the ISO Transport Service over TCP/IP. This approach solves the problem for all applications which use the ISO Transport Service. This document describes the second approach.

The protocol described in this memo is based on the observation that both the Internet Protocol Suite and the ISO Protocol Suite are layered systems. A key aspect of the layering principle is that of layer-independence. The concept of layer-independence means that if one preserves the services offered by a particular layer (the Service-Provider) then the Service-User at that layer is completely unaffected by changes in the underlying layers or by the protocol used within the layer.

This document defines a Transport Service which appears to be identical to the Services and Interfaces offered by the ISO Transport Service Definition [ISO8072], but which will in fact implement the ISO Transport Protocol [ISO8073] on top of TCP/IP (IPv4 or IPv6), rather than the ISO Network Service [ISO8348].

The basis of this document is STD35, RFC1006 [RFC1006] written by Marshall T. Rose and Dwight E. Cass and it defines two transport classes of service. Transport Class 0 refines and supersedes the RFC1006 protocol and is aimed at preserving the RFC1006 installed base. Transport Class 2 defines a number of new features which are not provided in RFC1006, such as independence of Normal and Expedited Data channels and Explicit Transport Disconnection. These new features are largely based on RFC1859 [RFC1859] and extend the applicability of RFC1006 to new groups of applications.

This document specifies changes to the standards mentioned above and must be read in the context of the above mentioned standards. It will not be meaningful on its own.

The 'well known' TCP port 102 is reserved for hosts which implement the Protocol described in this document. Note that the Protocol does not mandate the use of TCP port 102 for all connections.

2. The Model

This section describes the differences between the model used by the ISO Transport and that described in this document.

2.1 ISO Transport Model

The ISO 8072 standard describes the ISO Transport Service Definition (TS). The ISO Transport Service Definition describes the services offered by the Transport Service Provider and the interfaces used to access these services.

The ISO 8073 standard describes the ISO Transport Protocol Specification (TP). The ISO Transport Protocol specifies common encoding rules and a number of classes of transport protocol procedure which can be used with different network Quality of Service.

The ISO 8348 standard describes the ISO Network Service Definition (NS). The ISO Network Service Definition describes the services offered by the network service Provider and the interfaces used to access these services.

The ISO Network Service specifies two type of service:

- Connection Oriented Network Service (CONS)

- ConnectionLess Network Service (CLNS)

The ISO Transport Protocol specifies five classes of procedures when operating over CONS and one class of procedure when operating over CLNS.

The relationship of these ISO standards is illustrated below:

```
         Transport Service User
           |
           |-ISO Transport Service Definition [ISO8072]
           |
      +---------------------------------------------------+
      |  Transport Service Provider                        |
      |  ISO Transport Protocol Specification [ISO8073]    |
      +---------------------------------------------------+
           |
           |-ISO Network Service Definition [ISO8348]
           |
```

2.2 ISO Transport over TCP (ITOT) Model

This document defines a model which provides ISO Transport Service, with minor extensions, running over TCP.

The ISO 8072 Transport Service is supported with minor modifications. See section 3.1.

The ISO 8073 Transport Protocol with some modifications is used to provide the modified Transport Service.

The Transmission Control Protocol is used in place of the ISO 8348 to provide a CONS like service. See section 4.

This document specifies a simple encapsulation mechanism between the modified ISO 8073 Transport Protocol and the TCP.

ISO 8073 Transport Protocol specifies five classes when operating over ISO 8348 CONS. This document specifies how to operate class 0 and 2 over TCP. This document does not prevent use of other classes from operating over TCP, but their specification is beyond the scope of this document.

The relationship of these standards is illustrated below:

```
         Transport Service User
           |
           |-ISO Transport Service (modified)
           |
      +---------------------------------------------------+
      |  Transport Service Provider                       |
      |  ISO Transport Protocol (modified) Specification |
      +---------------------------------------------------+
           |
           |-TCP as a Connection Oriented Network Service
           |
```

2.3 Overview of Protocol and Service

This document defines use of the ISO Transport Protocol (with some extensions) running over TCP. Two variants of the protocol are defined, "Class 0 over TCP" and "Class 2 over TCP", which are based closely on the ISO Transport Class 0 and 2 Protocol.

Section 3 defines the Service offered to the Transport User by this protocol, and shows the differences from the ISO Transport Service. The mapping between the Service primitives in the ISO Network Service and TCP are defined. Section 4 defines the Transport Protocol.

3 Service Definition

This section describes the Transport Service offered to the Transport User. It also defines the mapping between the Network Service Definition and the TCP Service Definition.

3.1 Transport Service Definition

ISO 8072 Transport Service is supported with the following extensions:

- Use of Quality of Service parameter is not defined

- Access to Non-disruptive Transport Disconnection

3.1.1 Transport Service Definition Primitives

Information is transferred to and from the TS-User in the Transport Service primitives listed below:

Actions

T-CONNECT.REQUEST
- a TS-User indicates that it wants to establish transport connection

T-CONNECT.RESPONSE
- a TS-User indicates that it will honour the request

T-DISCONNECT.REQUEST
- a TS-User indicates that the transport connection is to be closed

T-DATA.REQUEST
- a TS-User sends data

T-EXPEDITED DATA.REQUEST
- a TS-User sends "expedited" data

Events

T-CONNECT.INDICATION
- a TS-User is notified that a transport connection establishment is in progress

T-CONNECT.CONFIRMATION
- a TS-User is notified that the transport connection has been established

T-DISCONNECT.INDICATION
- a TS-User is notified that the transport connection is closed

T-DATA.INDICATION
- a TS-User is notified that data can be read from the transport connection

T-EXPEDITED_DATA.INDICATION
- a TS-User is notified that expedited data can be read from the transport connection

3.2 Network Service Definition

This section describes how TCP is used to provide ISO 8348 CONS.

3.2.1 ISO 8348 CONS primitives

Information is transferred to and from the NS-provider in the Network Service Primitives listed below:

Actions

N-CONNECT.REQUEST
- a NS-user indicates that it wants to establish a network connection

N-CONNECT.RESPONSE
- a NS-user indicates that it will honour the request

N-DISCONNECT.REQUEST
- a NS-user indicates that the network connection is to be closed

N-DATA.REQUEST
- a NS-user sends data

N-EXPEDITED_DATA.REQUEST
- a NS-user sends "expedited" data

Events

N-CONNECT.INDICATION
- a NS-user is notified that a network connection establishment is in progress

N-CONNECT.CONFIRMATION
- a NS-user is notified that the network connection has been established

N-DISCONNECT.INDICATION
- a NS-user is notified that the network connection is closed

N-DATA.INDICATION
- a NS-user is notified that data can be read from the network connection

N-EXPEDITED_DATA.INDICATION
- a NS-user is notified that expedited data can be read from the connection

3.2.2 TCP Service primitives

The mapping between, ISO 8348 CONS primitives and TCP Service primitives, defined in this document assumes that the TCP offers the following service primitives:

Actions

TCP-LISTEN_PORT
- PASSIVE open on given port

TCP-OPEN_PORT
- ACTIVE open to the given port

TCP-READ_DATA
- data is read from the connection

TCP-SEND_DATA
- data is sent on the connection

TCP-CLOSE
- the connection is closed (pending data is sent)

Events

TCP-CONNECTED
- open succeeded (either ACTIVE or PASSIVE)

TCP-CONNECT_FAIL
- ACTIVE open failed

TCP-DATA_READY
- Data can be read from the connection

TCP-ERRORED
- the connection has errored and is now closed

TCP-CLOSED
- an orderly disconnection has started

3.2.3 Mapping TCP as a Network Service Provider

3.2.3.1 Network Connection Establishment

In order to perform a N-CONNECT.REQUEST action, the TS-Provider performs a TCP-OPEN_PORT to the desired IPv4 or IPv6 address using the selected TCP port. When the TCP signals either success or failure, this results in an N-CONNECT.INDICATION action.

In order to await a N-CONNECT.INDICATION event, a server performs a TCP-LISTEN_PORT to the selected TCP port. When a client successfully connects to this port, the TCP-CONNECTED event occurs and an implicit N-CONNECT.RESPONSE action is performed.

Mapping parameters between the TCP service and the ISO 8348 CONS service is done as follow:

```
Network Service                TCP
---------------                ---
CONNECTION ESTABLISHMENT

        Called address         server's IPv4 or IPv6 address
                               and TCP port number.

        Calling address        client's IPv4 or IPv6 address

        all others parameters  ignored
```

Please also refer to 'Notes to Implementors' section 6.1.

TCP port 102 is reserved for implementations conforming to this specification. Use of any TCP port is conformant to this specification.

3.2.3.2 Network Data Transfer

In order perform a N-DATA.REQUEST action, the TS-provider constructs the desired transport protocol data unit (TPDU), encapsulates the TPDU in a discrete unit called TPKT and uses the TCP-SEND_DATA primitive. Please also refer to 'Notes to Implementors' section 6.2.

In order to trigger a N-DATA.INDICATION action, the TCP indicates that data is ready through TCP-DATA_READY event and a TPKT is read using the TCP-READ_DATA primitive.

Mapping parameters between the TCP service and the ISO 8348 CONS service is done as follow:

```
Network Service                TCP
---------------                ---
DATA TRANSFER

        NS User Data (NSDU)    DATA
```

3.2.3.3 Network Connection Release

In order to perform an N-DISCONNECT.REQUEST action, the TS-provider simply closes the TCP connection through TCP-CLOSE primitive.

In order to trigger a N-DISCONNECT.INDICATION, the TCP indicates that the connection has been closed through TCP-CLOSE event. If the TCP connection has failed the TCP indicates that the connection has been closed through TCP-ERRORED event, this trigger a N-DISCONNECT.INDICATION.

Mapping parameters between the TCP service and the ISO 8348 CONS service is done as follow:

```
Network Service               TCP
---------------               ---
CONNECTION RELEASE

        all parameters        ignored
```

4. Transport Protocol Specification

ISO 8073 Transport Protocol Classes 0 and 2 are supported with extensions as defined in each subsections below.

A Transport Protocol class is selected for a particular transport connection based on the requirements of the TS-User.

ISO 8073 Transport Protocol exchanges information between peers in discrete units of information called transport protocol data units (TPDU). The protocol defined in this document encapsulates these TPDUs in discrete units termed Packets (TPKT).

This document mandates the implementation of ISO 8073 Transport Protocol options negotiation (which includes class negotiation).

Please refer to ’Notes to Implementors’ section 6.3 with respect to Class negotiation and to the ’Rationale’ section 7. with respect to Interoperability with RFC1006.

4.1 Class 0 over TCP

Class 0 provides the functions needed for connection establishment with negotiation, data transfer with segmentation, and protocol error reporting. It provides Transport Connection with flow control based on that of the NS-provider (TCP). It provides Transport Disconnection based on the NS-provider Disconnection.

Class 0 is suitable for data transfer with no Explicit Transport Disconnection.

4.1.1 Connection Establishment

The principles used in connection establishment are based upon those described in ISO 8073, with the following extensions:

- Connect Data may be exchanged using the user data fields of Connect Request (CR) and Connect Confirm (CC) TPDUs

- Use of "Expedited Data Transfer Service" may be negotiated using the negotiation mechanism specified in ISO 8073. The default is to not use "Expedited Data Transfer Service".

- Non-standard TPDU size may be negotiated using the negotiation mechanism specified in ISO 8073. The maximum TPDU size is 65531 octets. The Default maximum TPDU size is 65531 octets. Please refer to 'Notes to Implementors' section 6.4.

4.1.2 Data Transfer

The elements of procedure used during transfer are based upon those presented in ISO 8073, with the following extension:

- Expedited Data may be supported (if negotiated during connection establishment) by sending the defined Expedited Data (ED) TPDU.

The ED TPDU is sent inband on the same TCP connection as all of the other TPDUs.

To support Expedited Data a non-standard TPDU is defined. The format used for the ED TPDU is nearly identical to the format for the Normal Data (DT) TPDU. The only difference between ED TPDU and DT TPDU is that the value used for the TPDU code is ED and not DT. The size of a Expedited Data user data field is 1 to 16 octets.

For TPDU bit encoding please refer to 'Notes to Implementors' section 6.5.

4.1.3 Connection Release

The elements of procedure used during a connection release are identical to those presented in ISO 8073.

Transport Disconnection is based on the NS-provider (TCP) Disconnection and is therefore disruptive.

4.2 Class 2 over TCP

Class 2 provides the functions needed for connection establishment with negotiation, data transfer with segmentation, and protocol error reporting. It provides Transport Connection with flow control based on that of the NS-provider (TCP). It provides Explicit Transport Disconnection.

Class 2 is suitable when independence of Normal and Expedited Data channels are required or when Explicit Transport Disconnection is needed.

4.2.1 Connection Establishment

The principles used in connection establishment are based upon those described in ISO 8073, with the following extensions:

- Connection Request and Connection Confirmation TPDUs may negotiate use of "Transport Expedited Data Transfer" service. "Transport Expedited Data Transfer" service is selected by setting bit 1 of the "Additional Option" parameter, and is negotiated using the mechanism specified in ISO 8073.

  The default is to not use "Transport Expedited Data Transfer Service".

- Connection Request and Connection Confirmation TPDUs may negotiate use of "Expedited Data Acknowledgement". "Expedited Data Acknowledgement" is selected by setting bit 6 of the "Additional Option" parameter, and is negotiated using the mechanism specified in ISO 8073.

  The default is to not use "Expedited Data Acknowledgement" for Expedited Data transfer.

- Connection Request and Connection Confirmation TPDUs may negotiate use of the "Non-blocking Expedited Data" service. "Non-blocking Expedited Data" is selected by setting bit 7 of the "Additional Option" parameter, and is negotiated using the mechanism specified in ISO 8073.

  The default is to not use the "Non-blocking Expedited Data" service.

- Connection Request and Connection Confirmation TPDUs may negotiate use of either "Forward Connection (Splitting and Recombining)" or "Reverse Connection" procedure for Expedited Data transfer.

Use of "Forward Connection" or use of "Reverse Connection" procedure is selected by setting bit 4 of the "Additional Option" parameter, and is negotiated using the mechanism specified in ISO 8073.

The default is to use "Forward Connection" procedure for Expedited Data transfer.

- Connection Request and Connection Confirmation TPDUs must not negotiate the use of "Explicit Flow Control".

- Non-standard TPDU size may be negotiated using the negotiation mechanism specified in ISO 8073. The maximum TPDU size is 65531 octets. The default maximum TPDU size is 65531 octets. Please refer to 'Notes to Implementors' section 6.4.

In the absence of a Flow Control policy, the use of ISO 8073 Multiplexing procedure lead to degradation of the quality of service. The Protocol defined in this document does not supported Multiplexing.

For the values of the "Additional Option" parameter please refer to 'Notes to Implementors' section 6.6.

For Class 2 options Profile please also refer to 'Notes to Implementors' section 6.6.

4.2.2 Data Transfer

The elements of procedure used during transfer are based upon those presented in ISO 8073, with the following extensions:

- Expedited Data may be supported (if negotiated during connection establishment) by sending Expedited Data (ED) TPDU.

- "Expedited Data Acknowledgement" may be supported (if negotiated during connection establishment) by sending Expedited Data Acknowledgement (EA) TPDU.

  When using "Expedited Data Acknowledgement", ED TPDUs require acknowledgement, and once an ED TPDU is transmitted no further DT/ED TPDUs may be sent until the outstanding ED TPDU has been acknowledged.

  When non-use of "Expedited Data Acknowledgement" has been negotiated, ED TPDUs require no acknowledgement, and further DT/ED TPDUs may be sent immediatly.

Please refer to 'Notes to Implementors' section 6.7 and section 6.8.

- "Non-blocking Expedited Data" service may be supported (if negotiated during connection establishment).

  When using "Non-blocking Expedited Data" service, the sender of an ED TPDU shall send the ED TPDU on both the Normal Data and Expedited Data TCP connections. Transmission of subsequent DT TPDU will not be interrupted. The receiver of ED TPDU counts how many ED TPDU it has seen on each TCP connection, and will only deliver to the TS-User the ED TPDU from the TCP connection with the higher count.

  When non-use of "Non-blocking Expedited Data" has been negotiated, ED TPDUs will not be duplicated.

  Please refer to 'Notes to Implementors' section 6.7 and section 6.8.

- For Expedited Data transfer, there are two possible procedures for the establishment and assignment of the Expedited Data TCP connection. Which one is used is negotiated during connection establishment.

  Both the "Forward Connection" procedure and "Reverse Connection" procedure guarantee independence of the Normal Data TCP connection from the Expedited Data TCP connection. They also ensure that a busy Normal Data TCP connection cannot block an Expedited Data TCP connection.

  The Expedited Data TCP connection created by either procedure must be between the same pair of hosts as the Normal Data TCP connection, must not be shared among Transport Connections, and must remain established until the Transport Connection is terminated, at which time it must be closed.

  TCP connections created for Expedited Data transfer should also use the TCP primitives defined in this document.

  The Forward Connection (Splitting and Recombining) procedure is defined in ISO 8073. This procedure allows a transport connection to make use of multiple TCP connections. Please refer to 'Notes to Implementors' section 6.9.

  The Reverse Connection procedure is not defined in ISO 8073. When using the Reverse Connection procedure the initiator of a Transport Connection creates a Normal Data TCP connection using an arbitrarily-chosen local TCP port ’x’ and a known remote TCP port (either the ITOT well-known port, or some other). The initiator listens for an incoming TCP connection on the TCP port ’x’. The responder of the Transport Connection must create a second TCP connection (to be used for Expedited Data) using an arbitrarily-chosen local TCP port ’y’ and the remote TCP port ’x’ , before it can issue a CC TPDU on the Normal Data TCP connection. The initiator need not listen for further TCP connections on port ’x’ after the Expedited Data TCP connection is established.

4.2.3 Connection Release

The elements of procedure used during a connection release are based upon those described in ISO 8073. A connection can be terminated by the TS-user in one of two ways:

- Disruptive Disconnect

- Non-Disruptive Disconnect

Disconnect Request (DR) and Disconnect Confirm (DC) TPDUs are exchanged in both cases. The DR TPDU carries a Reason code indicating the reason for the Disconnection.

Disruptive Disconnect specifies that all TPDUs still at the source are not required to be sent to the destination before the connection is disconnected. The DR Reason code is normal (80 hex).

Non-Disruptive Disconnect specifies that all TPDUs already given to the local TS-provider must be delivered to the remote TS-user, before the connection is disconnected. The DR Reason code is normal (80 hex) with Additional Information parameter value set to 80 hex.

4.3 TPKT Packet Format

A fundamental difference between the TCP and the ISO Network Service expected by ISO Transport is that the TCP manages a continuous stream of octets, with no explicit boundaries.

ISO Transport expects information to be sent and delivered in discrete objects termed Network Service Data Units (NSDU). Although ISO Transport allows combination of more than one TPDU inside a single NSDU for the purposes of discussion an NSDU is identical to a TPDU. Please refer to ISO 8073 for the valid set of concatenated TPDUs.

The protocol described by this memo uses a simple packetization scheme in order to delimit TPDU. Each packet (TPKT), is viewed as an object of variable length composed of an integral number of octets.

A TPKT consists of two part:

- a Packet Header

- a TPDU.

The format of the Packet Header is constant regardless of the type of TPDU. The format of the Packet Header is as follows:

```
+--------+--------+----------------+-----------....---------------+
|version |reserved| packet length  |             TPDU             |
+----------------------------------------------....---------------+
<8 bits> <8 bits> <   16 bits    > <       variable length       >
```

where:

- Protocol Version Number
  length: 8 bits
  Value: 3

- Reserved
  length: 8 bits
  Value: 0 - (See 'Notes to Implementors' section 6.10)

- Packet Length
  length: 16 bits
  Value: Length of the entire TPKT in octets, including Packet Header

- TPDU
  ISO Transport TPDU as defined in ISO 8073 and as defined in this document.

5. Address representations

It is desirable to be able to represent ITOT access point addresses as:

- Printable strings

- OSI Network Addresses (often known as NSAP addresses or simply NSAPAs)

This section defines the formats which MUST be used in each case.

5.1 String representation of ITOT access point addresses

RFC1278 [RFC1278] defines a general string representation for OSI Presentation Addresses, including specific reference to RFC1006 addresses which encapsulate IPv4 addresses. RFC1278 is also applicable to ITOT addresses which encapsulate IPv4 addresses.

This RFC is currently being updated to define a string representation for ITOT addresses which encapsulate IPv6 addresses.

ITOT access point address string representation specify an IP address (IPv4 or IPv6) and an optional TCP port number.

5.2 OSI Network Address encoding

RFC1277 [RFC1277] defines a general mechanism to encode addressing information within OSI Network Addresses (NSAPA), including specific reference to RFC1006 using IPv4. RFC1277 is also applicable to ITOT addresses using IPv4.

The RFC "IPv6 addresses inside an NSAPA" [IPv6] defines general mechanisms for the support of NSAP addressing in an IPv6 network. It also defines how to embed an IPv6 address inside a OSI NSAP address.

This RFC is applicable to ITOT addresses using IPv6. For ITOT addresses, the default selector of the NSAPA is defined to have the value ’10000000’B.

It should be noted that given that an IPv6 addresses can encode IPv4 addresses, this format can also encode ITOT addresses using IPv4.

6. Notes to Implementors

6.1 TCP Connection Establishment

Implementors should be aware that ISO transport protocols assume that they will be told by the network service provider (in this case TCP/IP) when the network connection being used to transmit their TPDUs is unexpectedly terminated. It is therefore strongly suggested that the TCP keep alive mechanism be selected, as this ensures reporting of network connection loss.

6.2 TCP Data transfer

For performance reason it is suggested that the Nagle algorithm [RFC 896] be disabled (using the TCP_NODELAY socket option). This feature allows TPKT data to be sent without delay.

6.3 Class negotiation

The principle used in Class negotiation is identical to those described in ISO 8073. Class and options are negotiated during Connection establishment. The choice made by the Transport will depend upon the TS-User requirements as expressed via T-CONNECT service primitives.

The initiator of the Transport Connection proposes a preferred class and may propose an alternative class.

The responder selects one class defined in the table below.

If the preferred class is not selected then on receipt of the connect confirm TPDU the initiator adjusts its operation according to the class selected.

Proposed in CR TPDU		CC TPDU
Preferred class	Alternative class	Response
class 0	none	class 0
class 2	class 0	class 2 or 0
class 2	none	class 2

6.4 Default maximum TPDU size

The default maximum TPDU size value specified in this document breaks ISO Transport negotiation rule which states that the maximum TPDU size specified or defaulted by the CC TPDU cannot be greater than the maximum TPDU size proposed by the CR TPDU.

To avoid the consequences of this, it is strongly recommended that the CC TPDU always specifies the maximum TPDU size value.

6.5 Class 0 TPDU bit encoding

This protocol no longer allows credit and TPDU-NR (bits 0 to 6) fields to be ignored on input, which is in line with ISO 8073 encoding rules. RFC1006 TPDU encoding defined inconsistent encoding rules.

6.6 Class 2 Options

Class 2 Additional Option parameter value

BIT	OPTION
8	Not applicable
7	= 1 Use of Non-blocking Expedited Data = 0 Non-use of Non-blocking Expedited Data (default)
(*) 6	= 1 Use of Expedited Data Acknowledgement = 0 non-use of Expedited Data Acknowledgement (default)
5	Not applicable
(*) 4	= 1 Use of Reverse Connection procedure = 0 Use of Forward Connection procedure (default)
3	Not applicable
2	Not applicable
1	= 1 Use of Transport Expedited Data Service = 0 Non-use of Transport Expedited Data Service (default)

(*) In ISO 8073, bit 4 is defined as use of "Network Expedited" and bit 6 is defined as "Request Acknowledgement".

Class 2 Options Profile

Bits 1 4 6 7	Service selected
0 x x x	Non-use of Transport Expedited Data Service
	Bits 4 6 7 are not applicable (*)
1 x x x	Use of Transport Expedited Data Service
1 0 x x	Use of Expedited Data Service with Forward Connection
1 0 1 0	Forward Connection with Expedited Data Acknowledgement
1 0 1 1	Forward Connection with Expedited Data Acknowledgement and use of Non-blocking Expedited Data (**)
1 0 0 0	Forward Connection with non-use of Expedited Data Acknowledgement (***)
1 0 0 1	Forward Connection with non-use of Expedited Data Acknowledgement and use of Non-blocking Expedited Data
1 1 x x	Use of Expedited Data Service with Reverse Connection
1 1 1 0	Reverse Connection with Expedited Data Acknowledgement
1 1 1 1	Reverse Connection with Expedited Data Acknowledgement and use of Non-blocking Expedited Data (**)
1 1 0 0	Reverse Connection with non-use of Expedited Data Acknowledgement (***)
1 1 0 1	Reverse Connection with non-use of Expedited Data Acknowledgement and use of Non-blocking Expedited Data

(*) Note the default (0000) provides an RFC1006-like service with Explicit Transport Disconnection.

(**) Note in this case use of Expedited Data Acknowledgement with use of Non-blocking Expedited Data is a wasted effort (See section 6.5)

(***) Note in this case Normal and Expedited Data TPDU are not synchronised. (See section 6.6)

6.7 Class 2 Expedited Data Acknowledgement

The Protocol specified in this document does not define any relationship between use of "Expedited Data Acknowledgement" option and use of "Non-blocking Expedited Data" service.

However please note that when using "Non-blocking Expedited Data" service it is a wasted effort to use "Expedited Data Acknowledgement", since ED TPDUs are duplicated and sent on both the Normal Data and Expedited Data TCP connections.

6.8 Class 2 Normal Data and Expedited Data handling

There exist two separate application requirements for using Expedited Data:

1- Synchronisation of the order of delivery between Normal and Expedited Data TPDU.

2- Independence of Normal and Expedited data channels. A busy Normal Data channel should not block an Expedited Data channel.

The protocol described in this document can accommodate both requirements, separately or in combination.

Synchronisation:

If synchronised order of delivery between Normal and Expedited Data TPDU is required then use of either "Expedited Data Acknowledgement" TPDU or use of the "Non-blocking Expedited Data" service must be negotiated during connection establishment.

If synchronised order of delivery between Normal and Expedited Data TPDU is not required then non-use of "Expedited Data Acknowledgement" need not be negotiated during connection establishment.

Independence:

If Independence of Normal and Expedited data channels is required then Forward or Reverse connection must be negotiated during connection establishment. Expedited data TPDU must be sent on the Expedited data channel.

If Independence of Normal and Expedited data channels is not required then Forward connection should be negotiated during connection establishment and the Expedited data channels should never be established. Expedited data TPDU is then sent inband on the Normal data channel.

Finally please note that independence of Normal and Expedited data channels without synchronisation relaxes the Transport Service definition of Expedited data and is not consistent with ISO 8072.

6.9 Class 2 Forward Connection procedure

As defined in ISO 8073, when "Forward Connection" (Splitting and Recombining) procedure is used for Expedited Data transmission, ED TPDU must only be sent over an outgoing NS-provider TCP connection.

As defined in ISO 8073, this document does not mandates use of the Splitting procedure for Expedited Data transmission. The Recombination procedure, which associates Data (normal and expedited) TPDUs arriving for a transport connection over two TCP connections must be handled.

It is legal to send Expedited Data TPDU inband on the Normal Data TCP connection.

Please note that the protocol specified in this document does not define when an Expedited Data TCP connection should be established. This is an implementation choice.

When using "Non-blocking Expedited Data" service it is recommended to not delay establishing Expedited Data TCP connection.

6.10 TPKT

This document specifies the value of the TPKT reserved field.

Implementation should not interpret and act upon any value in a reserved field. To avoid Interoperability issues with RFC1006, this field should be ignored on input.

7. Rationale - Interoperability with RFC1006

We have chosen to maintain the same TPKT protocol version in ITOT as in RFC1006 (version 3). The reason for this decision is that the changes in this document do not conflict with RFC1006. If we were to change the protocol version we would prevent existing RFC1006 implementations which mandate version 3 from interoperating with the protocol defined in this document.

One consequence of this decision relates to class negotiation. The protocol described in this document introduces Class 2 over TCP, and it therefore introduces the need to be able to perform class negotiation between Class 2 and Class 0. While all Transport implementations should be able to handle Class negotiation, we recognise that some RFC1006 implementations cannot. Therefore Implementors should be aware that Class 2 Connect Request (with no Alternative class) could be accepted with a Class 0 Connect Confirm, at which point the Connect Confirm should be rejected as specified in ISO 8073.

8. Security Considerations

Security issues are not specifically addressed in this document. Operation of this protocol is no more and no less secure than operation of TCP and ISO 8073 protocols. The reader is directed there for further reading.

Acknowledgements

The authors are pleased to acknowledge the suggestions and comments of Harald T. Alvestrand, Jim Bound, John Day, Mike Dyer, Peter Furniss, Dan Harrington, Steve Kille, Keith G. Knightson, Keith Sklower, Matt Thomas, Robert Watson and many other members of the IETF TOSI mailing list. The support of Allison Mankin of the IESG was essential.

References

[ISO8072] ISO. "International Standard 8072. Information Processing Systems - Open Systems Interconnection: Transport Service Definition."

[ISO8073] ISO. "International Standard 8073. Information Processing Systems - Open Systems Interconnection: Transport Protocol Specification." ISO 8073:1992 and 8073:1992/Amd.5:1995.

[ISO8348] ISO. "International Standard 8348. Information Processing Systems - Open Systems Interconnection: Network Service Definition."

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC896] Nagle, J., "Congestion Control in IP/TCP Inertnetworks", RFC 896, January 1984.

[RFC1006] Rose, M., and D. Cass, "ISO Transport Services on Top of the TCP Version 3", STD 35, RFC 1006, May 1987.

[RFC1277] Hardcastle-Kille, S., "Encoding Network Addresses to support operation over non-OSI lower layers", RFC 1277, November 1991.

[RFC1278] Hardcastle-Kille, S., "String encoding of Presentation Address", RFC 1278, November 1991.

A string encoding of Presentation Address update to RFC1278, Work in Progress.

[RFC1859] Pouffary, Y., "ISO Transport Class 2 Non-use of Explicit Flow Control over TCP - RFC1006 extension", RFC 1859, October 1995.

[IPV6] Deering, S., and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December 1995.

Hinden,, R., and S. Deeing, "IP Version 6 Addressing Architecture", RFC 1884, December 1995.

Bound, J., Carpenter, B., Harrington, D., Houldsworth, J., and A. Lloyd, "OSI NSAPs and IPv6", RFC 1888, August 1996.

Authors' Addresses

Yanick Pouffary
End Systems Networking
Digital Equipment Corporation
Centre Technique (Europe)
B.P. 027
950 Routes des colles
06901 Sophia antipolis, France

Phone: +33 92-95-62-85
Fax: +33 92-95-62-35
EMail: pouffary@taec.enet.dec.com

Alan Young
ISODE Consortium
The Dome
The Square
Richmond, UK

Phone: +44 181 332 9091
Fax: +44 181 332 9019
EMail: A.Young@isode.com