

ISO Transport Class 2 Non-use of Explicit Flow Control over TCP
RFC1006 extension

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Table of Contents

1. Introduction - General recommendations.....	2
2. The protocol.....	3
2.1 TCP service as a Network Service - The Primitives.....	3
2.2 Connection Establishment.....	4
2.3 Data Transfer.....	5
2.4 Connection Release.....	6
3. Packet Format.....	6
4. DIGITAL DECnet over TCP/IP.....	8
Acknowledgements.....	9
References.....	9
Author's Address.....	9

1. Introduction - General recommendations

This document is an extension to STD35, RFC1006, a standard for the Internet community. The document does not duplicate the protocol definitions contained in RFC1006 and in International Standard ISO 8073. It supplements that information with the description of how to implement ISO Transport Class 2 Non-use of Explicit Flow Control on top of TCP.

The document should be used in conjunction with the RFC1006 and ISO 8073.

The RFC1006 standard defines how to implement ISO 8073 Transport Class 0 on top of TCP. This memo defines how to implement ISO 8073 Transport Class 2 Non-use of Explicit Flow Control on top of TCP. Like ISO Transport Class 0, Class 2 Non-use of Explicit Flow Control provides basic connection with minimal overhead.

A Transport protocol class is selected for a particular Transport connection based upon the characteristics of the lower layers and the

requirements of the upper layer. Use of class 2 Non-use of Explicit Flow Control is suitable when the use of separate virtual data channels for normal and expedited Data are desirable or when an explicit disconnection of the Transport connection is desirable.

Hosts which choose to implement this extension are expected to listen on the well-known TCP port number 399.

It is recommended that the well-known RFC1006 TCP port 102 not be used. This recommendation is done to minimise impact to an existing RFC1006 implementation.

The memo also describes the use of this extension within the DIGITAL Network Architecture (DNA).

2. The protocol

The protocol specified by this memo is fundamentally equivalent to the protocol ISO 8073 Transport Class 2 Non-use of Explicit Flow Control, with the following extensions:

- Expedited Data service is supported.
- Splitting and Recombining may be used for Expedited Data transmission.
- The Network Service used is provided by TCP.

The ISO 8073 Transport protocol Class 2 allows Multiplexing. It is recommended that this capability not be use for performance reasons.

The ISO 8073 Transport protocol exchanges information between peers in discrete units of information called transport protocol data units (TPDUs). The protocol defined in this memo encapsulates these TPDUs in discrete units called TPKTs. The structure of these TPKTs and their relationship to TPDUs are discussed in the next sections.

2.1 TCP service as a Network Service - The Primitives

The mapping between the TCP service primitives and the service primitives expected by ISO 8073 Transport when operation over Connection-oriented network service is straightforward.

Note: The following description of the mapping is a repeat from the RFC1006 standard.

network service	TCP
-----	---
CONNECTION ESTABLISHMENT	
N-CONNECT.REQUEST	open completes
N-CONNECT.INDICATION	listen (PASSIVE open) finishes
N-CONNECT.RESPONSE	listen completes
N-CONNECT.CONFIRMATION	open (ACTIVE open) finishes
DATA TRANSFER	
N-DATA.REQUEST	send data
N-DATA.INDICATION	data ready followed by read data
CONNECTION RELEASE	
N-DISCONNECT.REQUEST	close
N-DISCONNECT.INDICATION	connection closes or errors

Mapping parameters between the TCP service and the network service is also straightforward:

network service	TCP
-----	---
CONNECTION ESTABLISHMENT	
Called address	server's IP address (4 octets)
Calling address	client's IP address (4 octets)
all others	ignored
DATA TRANSFER	
NS-user data (NSDU)	data
CONNECTION RELEASE	
all parameters	ignored

2.2 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 the use of Expedited Data transfer using the negotiation mechanism specified in ISO 8073.
- Connection Request and Connection Confirmation TPDUs must not negotiate the Use of Explicit Flow Control.

To perform an N-CONNECT.REQUEST action, the TS-peer performs an active open to the desired IP address using the well know TCP port 399. When the TCP signals either success or failure, this results in an N-CONNECT.INDICATION action.

To await an N-CONNECT.INDICATION event, a server listens on the well know TCP port 399. When a client successfully connects to this port, the event occurs and an implicit N-CONNECT.RESPONSE action is performed.

2.3 Data Transfer

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

- Expedited Data may be supported (if negotiated during connection establishment).

In Non-Use of Explicit Flow Control Expedited Data requires no Expedited Data Acknowledgement.

- Splitting and Recombining may be used for Expedited Data transmission.

The procedure of Splitting and Recombining allows a transport connection to make use of multiple TCP connections. TCP connections created for Splitting purposes should also use the primitives described in 2.1.

It is recommended to only create a second TCP connection for Expedited Data when transmission of Expedited Data is requested.

Expedited Data must only be sent over an outgoing TCP connection. This second 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.

Implementors note: The procedure of Splitting and Recombining for Expedited Data transmission guaranties that a congested Normal Data TCP connection cannot block an Expedited Data TCP connection. It also ensures independence of the Normal Data TCP connection from the Expedited Data TCP connection.

To perform an N-DATA.REQUEST action, the TS-peer constructs the desired TPKT and uses the TCP send data primitive.

To trigger an N-DATA.INDICATION action, the TCP indicates that data is ready and a TPKT is read using the TCP read data primitive.

2.4 Connection Release

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

A connection can be terminated by the user in one of two ways:

- Abort Disconnect specifies that all messages at the source are not required to be sent to the destination before the connection is disconnected.
- Synchronous Disconnect specifies that all messages at the source must be sent to the destination, and that all messages at the destination must be delivered, before the connection is disconnected.

Disconnect Request and Disconnect Confirmation TPDUs are exchanged in both cases. The Disconnect Request TPDU carries a code indicating the reason for the disconnection.

In the case of a Synchronous Disconnect the Disconnect Request reason code is normal (80 hex). For an Abort Disconnect the Disconnect Request reason code is normal with additional information parameter value set to (c0 hex).

Upon receipt of a Disconnect Confirmation TPDU a N-DISCONNECT.REQUEST action is performed to close the TCP connection.

If the TCP connection fails for some other reason, this generates an N-DISCONNECT.INDICATION event.

3. Packet Format

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

The protocol described in RFC1006 uses a simple packetization scheme in order to delimit TPDU's. Each packet, termed a TPKT, consists of two parts: a packet-header and a TPDU.

We use the same scheme described in RFC1006 for this extension. There is no need to change the version number. The ISO transport TPDU sufficiently describes the transport protocol class being used.

The format of the packet-header described below is a repeat from RFC1006.

0																1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																						
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-																						
									vrsn																		reserved																		packet length																		
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-																						

where:

```
vrsn          8 bits
```

This field is always 3 for the version of the protocol described in this memo.

packet length	16 bits (min=7, max=65535)
---------------	----------------------------

The packet length is the length of the entire packet in octets, including packet-header.

The format of the ISO transport TPDU is defined in ISO 8073.

4. DIGITAL DECnet over TCP/IP

DECnet over TCP/IP is implemented using the DECnet Session Control layer over this RFC1006 extension protocol.

The informational RFC defined in this document provides the Transport Service functionality required by DECnet Applications while operating over TCP/IP.

The next paragraph is a brief summary of the role of the DECnet Session Control Layer. For further details, refer to the DIGITAL DNA Session Control Layer Specification.

The DECnet Session Control Layer makes a Transport Service available to End Users of a network. This layer is concerned with system-dependent functions related to creating, maintaining, and destroying Transport Connections. Separate virtual data channels, known as

"Normal" and "Expedited", are provided to End Users. DECnet Session Control must be guaranteed independence of these channels by the Transport Layer. Expedited Data transmission cannot be blocked by a congested normal data channel. DECnet Session Control requires that all data in transit be delivered before initiating the release of the Transport Connection.

DECnet, DNA, and the DIGITAL logo are trademarks of Digital Equipment Corporation.

Acknowledgements

Bill Duane, Jim Bound, David Sullivan, Mike Dyer, Matt Thomas, Dan Harrington and many other members of the DECnet engineering team.

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."
- [ISO8327] ISO. "International Standard 8327. Information Processing Systems -- Open Systems Interconnection: Session Protocol Specification."
- [RFC791] Postel, J., "Internet Protocol - DARPA Internet Program Protocol Specification", STD 5, RFC 791, USC/Information Sciences Institute, September 1981.
- [RFC793] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, USC/Information Sciences Institute, September 1981.
- [RFC1006] Rose, M., and D. Cass, "ISO Transport Services on Top of the TCP - Version: 3", STD 35, RFC 1006, Northrop Research and Technology Center, May 1987.

Security Considerations

Security issues are not discussed in this memo.

Author's Address

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 92-95-62-32
EMail: pouffary@taec.enet.dec.com

