

Network Working Group
Request for Comments: 1613
Category: Informational

J. Forster
G. Satz
G. Glick
cisco Systems, Inc.
R. Day
JANET
May 1994

cisco Systems X.25 over TCP (XOT)

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.....	1
2. Conventions.....	2
3. Relationship Between XOT and X.25.....	2
4. Overall Packet Format.....	3
4.1 XOT Header.....	4
5. TCP Connection, Port Number, and Logical Channel Numbers (LCNs).....	4
6. XOT Packets.....	5
6.1 Virtual Circuit Setup and Clearing.....	5
6.2 Data and Flow Control.....	6
6.3 Interrupt, and Reset Packets.....	8
6.4 Restart, DTE Reject, Diagnostics, and Registration.....	8
6.5 PVC Setup.....	8
7. Acknowledgments.....	12
8. Security Considerations.....	12
9. References.....	12
10. Authors' Addresses.....	13

1. Introduction

It is sometimes desirable to transport X.25 over IP internets. The X.25 Packet Level requires a reliable link level below it and normally uses LAPB. This memo documents a method of sending X.25 packets over IP internets by encapsulating the X.25 Packet Level in TCP packets.

TCP provides a reliable byte stream. X.25 requires that the layer below it provide message semantics, in particular the boundary between packets. To provide this, a small (4-byte) XOT header is used between TCP and X.25. The primary content of this header is a

length field, which is used to separate the X.25 packets within the TCP stream.

In general, the normal X.25 protocol packet formats and state transition rules apply to the X.25 layer in XOT. Exceptions to this are noted.

2. Conventions

The following language conventions are used in the items of specification in this document:

- o MUST, SHALL, or MANDATORY -- This item is an absolute requirement of the specification.
- o SHOULD or RECOMMEND -- This item should generally be followed for all but exceptional circumstances.
- o MAY or OPTIONAL -- This item is truly optional and may be followed or ignored according to the needs of the implementor.

In some places in this document, there is parenthetical material labeled "DISCUSSION". This material is intended to give clarification and explanation of the preceding text.

3. Relationship Between XOT and X.25

When a networking device (a host, router, etc.) has an X.25 engine (i.e., protocol implementation), that engine may be connected to interface(s) running LAPB, and/or to logical interface(s) running LLC or XOT/TCP/IP. In general, the XOT layer itself is not at all sensitive to what kind of packets the X.25 engine passes to it. However, to improve interoperability between separate implementations, this document in some cases does specify behavior of the X.25 engine.

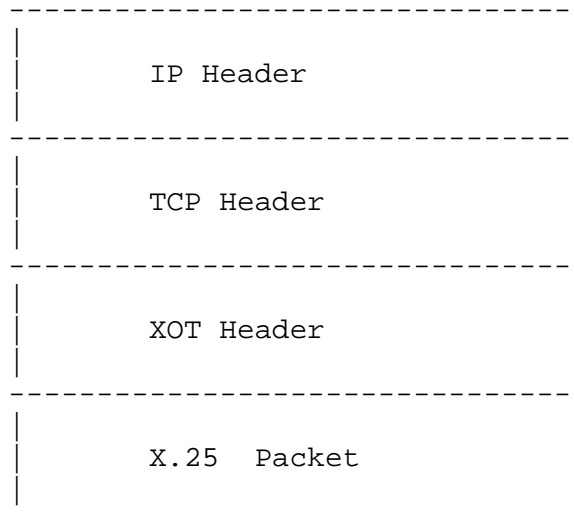
While this document primarily discusses XOT from the perspective of switching X.25 traffic (i.e., connecting an X.25 Virtual Circuit between the local X.25 interfaces of two networking devices), this should not prevent a host from offering X.25 connectivity using XOT.

The various X.25 standards may call a given packet type by a different name according to the assigned DTE/DCE role of the interface that originated the packet. XOT is intended to be insensitive to the DTE/DCE role of the local interfaces at either end of an XOT TCP connection, so, for this document, the following terms are interchangeable unless stated otherwise:

- o Call, Call Request and Incoming Call
- o Call Confirm, Call Accepted and Call Connected
- o Clear, Clear Request and Clear Indication
- o Clear Confirm, DTE Clear Confirmation and DCE Clear Confirmation
- o Data, DTE Data and DCE Data
- o Interrupt, DTE Interrupt and DCE Interrupt
- o Interrupt Confirm, DTE Interrupt Confirmation and DCE Interrupt Confirmation
- o RR, DTE RR and DCE RR
- o RNR, DTE RNR and DCE RNR
- o REJ, Reject and DTE REJ
- o Reset, Reset Request and Reset Indication
- o Reset Confirm, DTE Reset Confirmation and DCE Reset Confirmation
- o Restart, Restart Request and Restart Indication
- o Restart Confirm, DTE Restart Confirmation and DCE Restart Confirmation

4. Overall Packet Format

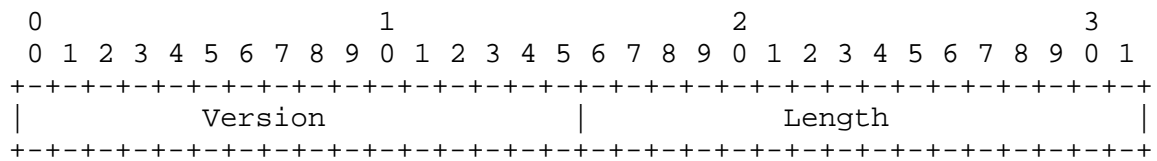
The entire encapsulated packet has the following format:



RFC convention is that a packet format is represented graphically with the data sent first above the data sent later. This convention is followed in this document, and therefore, while we refer to X.25 being transported over TCP, we draw the packet format with the X.25 portion of the packet lower on the page than the TCP portion.

4.1 XOT Header

The XOT header has the format:



Version (2 octets)

The version number of the XOT protocol is encoded in the first two octets. The version number **MUST** be 0. Other values of this field are reserved for future use. If a value other than 0 is received, then the TCP connection **MUST** be closed.

Length (2 octets)

The length of the X.25 packet is encoded in the second two octets. Values must be legal X.25 packet lengths. If the Length field has an illegal value, then the TCP connection **MUST** be closed.

5. TCP Connection, Port Number, and Logical Channel Numbers (LCNs)

A separate TCP connection **MUST** be used for each X.25 virtual circuit. All connections **MUST** be made to TCP port number 1998. This port number is an IANA Registered Port Number registered by cisco Systems; cisco has designated it for use by XOT.

The TCP connection **MUST** be created before the virtual circuit can be established. The TCP connection **MAY** be maintained after the virtual circuit has been cleared. Data **MUST NOT** be passed along with the TCP SYN packet.

The Logical Channel Number (LCN) field in the X.25 header has no significance and has arbitrary values. A corollary of this is that there is no assignment of one side of the connection to be DTE and another to be DCE.

DISCUSSION

Consider three devices A, B and C, where A and B both conduct XOT sessions to C. It's possible that C could receive two calls with the same LCN and, unless the X.25 engine could tell that they were received on different logical (XOT) interfaces, here would a danger of call collision (indeed a valid LCN on one interface may

not even be valid on a different interface). It is therefore necessary for C's X.25 engine to distinguish between the two streams, but the LCN field is not sufficient to do this. The XOT protocol design decision was to expect the XOT layer to communicate the stream identification to the X.25 layer.

6. XOT Packets

For each X.25 packet received from the TCP connection to be sent out a local interface, an XOT implementation **MUST** set the packet's logical channel number to that used on the outgoing interface. For the purposes of this RFC, a logical channel number is the 12 bit field confusingly defined by the X.25 Recommendations as the high-order 4 bit "logical channel group number" and low-order 8 bit "logical channel number", where the latter phrase is used to refer to both the aggregated 12 bits and the low-order 8 bits.

An XOT implementation **SHOULD NOT** modify the X.25 packet header information received on a local interface to be transmitted over the TCP connection.

An XOT implementation **MUST** modify the X.25 packet header information as required for proper X.25 protocol operation for packets received on a TCP connection to be transmitted over a local interface.

An XOT implementation **MAY** support connection between interfaces that use different flow control modulus. If this feature is supported, XOT **MUST** modify the packet General Format Identifier on all packets received over the TCP connection to set the proper modulus identifier.

6.1 Virtual Circuit Setup and Clearing

Once a TCP connection has been established, the X.25 Call packet is sent by the XOT that initiated the TCP connection. Eventually a Call Confirm or Clear packet is received, or the X.25 T11/T21 timeout occurs or the XOT TCP connection is closed. The usual X.25 state transitions are followed.

Any legal X.25 facilities from the family of X.25 protocols (including but not limited to the 1980, 1984 and 1988 CCITT X.25 Recommendations) **MAY** be included in the Call, Call Confirm and Clear packets. Receipt of an unknown or unsupported X.25 facility received from the TCP connection **SHOULD** be ignored (i.e., not presented in the packet sent out the local interface) or treated as an error as defined by the X.25 standard implemented.

To simplify end-to-end flow control, the packet size and window size are always sent explicitly as facilities in the Call packet. The Call packet MUST contain both Packet Size and Window Size facilities. The Call Confirm packet MAY contain these facilities. The handling of a Call received over a TCP connection that does not encode one or both of the flow control facilities is a local matter--if the XOT accepts such a Call, it MUST encode the missing flow control facility values that apply to the connection in the returned Call Confirm packet.

DISCUSSION

X.25 interfaces normally have a concept of network default values for packet size and window size. It was thought that when connecting diverse sites over a TCP/IP network this concept would be difficult to achieve in practice. If there is no network default, then each call must state the packet size and window size. This is the reason for requiring the packet size and window size facilities. It is expected that this can be achieved either by the XOT layer itself, or by configuring the X.25 engine such that there no network default on this interface.

After sending a Clear the TCP connection MAY be closed immediately without waiting for the Clear Confirm. A Clear Confirm received on the TCP connection MAY be silently discarded.

A packet with an invalid X.25 Packet Type Identifier (PTI) received over the TCP connection before a Call has been received (i.e., while in the P1 state) MUST be silently discarded.

6.2 Data and Flow Control

DISCUSSION

The implementation of X.25 flow control is a local matter, but different implementation choices affect XOT behavior.

An XOT implementation may implement either end-to-end flow control, where DATA, RR and RNR packets are sent over the TCP connection as received over the local interface, or local flow control, where flow control packets (RR, RNR and, if supported, REJ) are sent on a VC according to local criteria, a complete packet sequence of DATA packets may be fragmented or combined, and data packet numbering normally has only local DTE-DCE significance.

Existing implementations of XOT perform end-to-end flow control. Data and flow control packets are simply transferred between the

two local interfaces via the TCP connection, adjusting the X.25 header data as necessary for mixed modulo operation. This does not preclude an XOT implementation that performs local flow control, but interoperability requires that a local flow control implementation conduct the XOT session such that a connecting end-to-end flow control implementation receives Data packets of the proper size and flow control fields with appropriate P(S) and P(R) values.

An X.25 implementation that performs local flow control similarly may set up a Call between two local interfaces where each logical channel has its own packet and window sizes and Data packets must be fragmented or collected between the interfaces and each interface manages distinct packet sequence numbers; XOT operation is simply an extension to this operation as a VC is connected between the local interface and an XOT/TCP virtual interface, each of which have distinct window and packet sizes.

An XOT that implements local flow control **MUST** send data packet acknowledgements across the TCP connection for the DATA packets it receives from the TCP connection, using the received packet numbers, and **MUST** observe the maximum packet sizes agreed to across the TCP connection.

An XOT implementation **MUST NOT** assume that an RNR sent across the TCP connection will stop the flow of DATA packets in the other direction. An RNR packet received from the TCP connection **MAY** cause an RNR packet to be sent across the local interface; end-to-end flow control implementations **MAY** communicate the P(R) in an RNR packet received from the TCP connection by sending an RR packet on the local interface.

An XOT implementation that allows mixed-modulo connections and implements end-to-end flow control **MUST** intervene in the window size negotiation process when a modulo 128 Call Request proposes a window size of 8 or larger to an XOT connection that serves a modulo 8 interface. The intervention **MUST** either refuse the connection or lower the too-large window size(s) to a value valid for the interface and indicate the final result of the window size negotiation process in the Call Confirm packet returned over the TCP connection.

For any type of flow control implementation that supports mixed modulo connections, both cooperating XOTs **MUST** interpret the the P(S) and P(R) values received from the TCP connection and perform any flow control operation appropriate for correct X.25 operation of the local interface. End-to-end flow control implementations **MUST** translate between the two modulos and construct the analogous X.25 header P(S) and P(R) fields for DATA, RR and RNR packets.

An XOT implementation MAY support connecting two XOT TCP sessions to each other. If this feature is supported, XOT MUST simply connect the two TCP sessions without modifying the data passed.

6.3 Interrupt, and Reset Packets

Interrupt, Interrupt Confirm, Reset and Reset Confirm packets are sent over the TCP connection using the normal X.25 packet formats and state transitions. The end-to-end nature of both the Interrupt and Reset services MUST be maintained for correct X.25 operation.

6.4 Restart, DTE Reject, Diagnostics, and Registration

X.25 packets that have only a local DTE/DCE interface significance (Restart, Restart Confirm, DTE Reject, Diagnostic, Registration Request and Registration Confirmation) MUST NOT be sent over the TCP connection. If one of these packets is received, then it MUST be silently discarded.

6.5 PVC Setup

An XOT implementation MAY support connecting a PVC via XOT.

DISCUSSION

X.25 PVCs are Virtual Circuits that are presumed to be available when the X.25 service is available (i.e., in the R1 state). Connecting a PVC via XOT is complicated because no Call, Call Confirm, Clear or Clear Confirm packets are transferred (or allowed) across the X.25 interface--PVCs are simply available because they have been provisioned by the network provider as contracted for by the network users.

Supporting a PVC using XOT requires a data exchange between the XOT entities that is outside the scope of the X.25 standards, and must provide for a number of error conditions.

The setup of a PVC between two XOT entities is performed by exchanging a non-standard X.25 packet type (encapsulated in an XOT Header); the PVC setup exchange takes place immediately after a new TCP XOT connection has been established. The XOT implementation that initiated the TCP connection is the initiator; the other XOT is the responder.

The PVC Setup packet includes the X.25 General Format Identifier, LCN and Packet Type Identifier fields followed by additional data. This non-standard packet type takes the form:

```

+---+---+---+---+---+---+---+---+
| X.25 GFI | X.25 LCN |
+---+---+---+---+
|
+---+---+---+---+---+---+---+---+
|           X.25 PTI           | PVC setup PTI (= 0xF5)
+---+---+---+---+---+---+---+---+
|                               | version (= 0x81)
+---+---+---+---+---+---+---+---+
|                               | status
+---+---+---+---+---+---+---+---+
|                               | initiator interface name length (N)
+---+---+---+---+---+---+---+---+
|                               | initiator LCN (high octet)
+---+---+---+---+---+---+---+---+
|                               | initiator LCN (low octet)
+---+---+---+---+---+---+---+---+
|                               | responder interface name length (M)
+---+---+---+---+---+---+---+---+
|                               | responder LCN (high octet)
+---+---+---+---+---+---+---+---+
|                               | responder LCN (low octet)
+---+---+---+---+---+---+---+---+
|                               | sender incoming window
+---+---+---+---+---+---+---+---+
|                               | sender outgoing window
+---+---+---+---+---+---+---+---+
|                               | sender incoming max. packet size
+---+---+---+---+---+---+---+---+
|                               | sender outgoing max. packet size
+---+---+---+---+---+---+---+---+
|                               | initiator interface name (N octets)
+---+---+---+---+---+---+---+---+
|                               | responder interface name (M octets)
+---+---+---+---+---+---+---+---+

```

DISCUSSION

The PVC setup packet was designed so that the responder could simply modify a few fields of the received packet and send it back to the initiator.

The Packet Type Identifier was chosen from the unused X.25 PTI values so it is distinct from the standard X.25 Packet Type Identifiers.

The PVC setup version value was chosen to prevent connections with prior experimental implementations.

The PVC status field has the following values defined:

Status	Meaning
-----	-----
0x00	Waiting to connect
0x08	Destination disconnected
0x09	PVC/TCP connection refused
0x0A	PVC/TCP routing error
0x0B	PVC/TCP connect timed out
0x10	Trying to connect via TCP
0x11	Awaiting PVC-SETUP reply
0x12	Connected
0x13	No such destination interface
0x14	Destination interface is not up
0x15	Non-X.25 destination interface
0x16	No such destination PVC
0x17	Destination PVC configuration mismatch
0x18	Mismatched flow control values
0x19	Can't support flow control values
0x1A	PVC setup protocol error

DISCUSSION

Not all of the PVC status values are appropriate for a PVC setup packet; these values represent a particular implementation that chose to assign values in three groups that correspond to a short timer for a connect attempt (0x00 through 0x07), a long timer for a connect attempt (0x08 through 0x0F) and no attempt to connect (greater than 0x0F). The values that are appropriate for a PVC setup packet are 0x00 and those values greater than 0x12.

Most of the PVC status error values that may be found in a setup message are self-explanatory, with a few exceptions. The value 0x17, "Destination PVC configuration mismatch" may returned in the case that the targeted PVC already has an XOT PVC connection active. The value 0x19, "Can't support flow control values", may be returned when the flow control values match but, for instance, a modulo 8 interface is requested to set up a PVC with a window size greater than 7 or an interface is requested to set up a PVC

with a maximum packet size that is too large for its data link layer to transfer.

An XOT MAY retry a failed PVC setup; if implemented the XOT SHOULD wait between attempts (5 minutes is suggested).

Each XOT PVC is configured with the identity of the other XOT (i.e., IP address), the name of the interface to connect to, the Logical Channel Number on that interface and the flow control values to use. These data are present in the PVC setup packets and the responding XOT verifies the configurations are compatible.

The interface name fields are the ASCII names of the two interfaces involved. These names SHOULD be case-insensitive. There MUST NOT be any padding or trailing zero octets between or after the interface names.

The flow control values are the values from the perspective of the local interface of the XOT implementation that sent the PVC setup packet. The maximum packet size values are encoded as they are in the packet size facility, (i.e., the base-2 log of the size in octets, so 7 represents a maximum packet size of 128 octets). If the responding XOT implements end-to-end flow control, it will require that the configured flow control values be complimentary, so a returned status of 0x18 will indicate the values required by the responding XOT (note that the incoming value of one local interface corresponds to the outgoing value of the connecting local interface, and vice-versa).

After establishing the TCP connection the initiator sends a PVC setup packet, the status value MUST be 0x00; the responder will reply with its own PVC setup packet or by closing the TCP connection. An XOT PVC setup is successful if the responder returns a status of 0x00. Once the XOT PVC connection is successfully established, each XOT MUST complete a Reset procedure on the local interface, so if each local interface LCI is in state D1, a Reset packet would be generated both to the local interface and the XOT TCP connection.

An XOT PVC connection is broken by simply closing the TCP connection; X.25 packets that are not legal for PVCs MUST NOT be transferred across an XOT PVC connection. When a local interface undergoes the Restart procedure, the XOT PVC connections MUST be either perform a Reset (which is appropriate if the interface remains in state R1) or close the XOT PVC connection.

DISCUSSION

An XOT implementation SHOULD also consider how a PVC setup collision will be handled. Receipt of an XOT PVC setup for a PVC that is itself attempting to setup an XOT connection could either accept a (valid) setup attempt and, if two TCP XOT connections result, simply use one connection to send XOT data (XOT MUST NOT send traffic over both) and accept XOT data on either, or it can close the incoming attempt and, if no connections result, retry the connection after waiting for a random interval. If two connections are allowed for a PVC, closure of one SHOULD result in the closure of the other.

7. Acknowledgments

Greg Satz is the original designer and implementor of X.25 over TCP. Aviva Garrett of cisco Systems reviewed the specification and made many editorial corrections.

8. Security Considerations

Security issues are not discussed in this memo.

9. References

- [1] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.
- [2] CCITT, Blue Book Volume VIII--Fascicle VIII.2, "Data Communication Networks: Services and Facilities, Interfaces"; Recommendation X.25, "Interface Between Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit", 1989, Geneva.

10. Authors' Addresses

James R. Forster
Engineering Dept.
cisco Systems
1525 O'Brien Dr.
Menlo Park. CA. 94025

Phone: 1.415.688.8245
Fax: 1.415.688.8282
EMail: forster@cisco.com

Greg Satz
Engineering Dept.
cisco Systems
1525 O'Brien Dr.
Menlo Park. CA. 94025

Phone: 1.415.688.8245
Fax: 1.415.688.8282
EMail: satz@cisco.com

Gilbert Glick
Engineering Dept.
cisco Systems
1525 O'Brien Dr.
Menlo Park. CA. 94025

Phone: 1.415.688.8245
Fax: 1.415.688.8282
EMail: gglick@cisco.com

Bob Day
Joint Network Team
c/o Rutherford Appleton Laboratory
Chilton
Didcot
Oxfordshire OX11 0QX
United Kingdom

Phone: 44.235.44.5163
Fax: 44.235.44.6251
EMail: R.Day@jnt.ac.uk

