IEN 130

POSTEL

The enclosed analysis is submitted for your review and comment. It is part of a study investigating the use of commercially supported standard protocols for use in local network environments. Comments can be sent to:

McFarland @ ISI

CERF @ ISI

or mailed to International Computing directly.

INTERNATIONAL COMPUTING

SUBJECT	Comparison of X.25 and TCP Version 4 as Cable-bus Network Protocols	DATE	August 29, 1979
FROM	Jonathan Dreyer JND	COPIES	
TO	John Kenyon		

X.25 is a three-level protocol intended to be used between a host (AKA: DTE) and a network node (AKA: DCE, IMP). It also has some characteristics of an end-to-end protocol. TCP4, on the other hand, is strictly an end-to-end protocol and makes few assumptions about the lower protocol layers. It makes sense to compare the two only because they cannot conveniently coexist (at best a TCP can reside atop a datagram protocol built from X.25 level one or two, but especially in the first case there would be a lot of extra software and compatibility problems, and in the second case there would be duplication of function since TCP does not require the reliability provided by X.25 level two) so in any actual network a choice must be made between them.

RELIABILITY

Level two of X.25 implements error control, so transmissions between a host and a node are essentially error-free. Level three does no further error processing, so additional error control (if required) must be done at a lower level between nodes or by a higher-level protocol. The former technique is not immune to the (unlikely) possibility that a packet will be correctly read by a node but garbled before a checksum is computed for the next hop in its journey.

X.25 does not specify whether acknowledgements are host-to-host or host-to-node. In the latter case, a sending host cannot be guaranteed that a receiving host has received a packet. Such a failure might occur if the network fails to deliver a packet or if the receiver crashes.

TCP, on the other hand, uses end-to-end acknowledgement, error detection and retransmission, so the reliability of the intervening interfaces and processors is not crucial to the reliability of the system. The burden of checking rests upon each user host.

An end-to-end protocol like TCP is inherently more reliable than a point-to-point protocol like X.25 because point-to-point protocols are vulnerable between and outside of the protocols (e.g., an intermediate node might destroy a packet before computing a checksum to be transmitted, or it might crash after acknowledging a packet), but depending upon the reliability of the nodes and the node-to-node transfer the difference might be negligible.

FLOW CONTROL

X.25 uses a fixed window size (established a priori by the network administrator) with the additional capability of a receiver temporarily halting and restarting the output of a sender. There is also an optional facility for varying this on a per call basis. Since the left window edge comes as an acknowledgement the (intentional) ambiguity mentioned before exists in the X.25 specification, so whether the flow control information is end-to-end or just across the interface is up to the system manager. In the latter case, a stalled receiver can cause network congestion.

TCP uses a flexible window size - the receiver sends a window to the sender, implying an authorization to send information in that sequence number range. This offers more control than the simple "receive-not-ready" technique, especially since individual bytes (octets) are sequenced. It seems intuitively that this offers a performance advantage but it does increase the packet header size. More analysis is required to evaluate this.

TRANSPARENCY

Both protocols are capable of transmitting arbitrary bit streams and both have the ability to delimit arbitrary-length (multi-packet) bit strings ("more data" in X.25, "end-of-letter" in TCP). TCP does, however, restrict transmissions to an integral number of octets.

EFFICIENCY

A quick but possibly deceptive indicator of efficiency is the overhead per packet (in bits). For X.25, a packet has 24 non-data bits. This packet is enclosed in a frame that costs 48 extra bits.

A TCP segment header has 192 bits and, if the suggested internet datagram protocol is used, 96 additional bits of header are prepended to that.

This represents a difference of a factor of four in header sizes; with 1K bit data packet, X.25 transmits 7% overhead and TCP transmits 28% overhead.

ADAPTABILITY

Since the responsibility for acknowledgement and flow control is left unspecified by X.25, the network manager is given this choice, and this decision can be changed at any time. This change would probably necessitate a change in higher-level protocols. Also there are a few optional features that can be added at the manager's whim. Other changes to X.25 itself are unlikely since those would violate the standard. X.25 is restrictive in what higher-level protocols it supports; it provides a virtual circuit but, depending upon design choices, may require additional flow control or error control. A datagram service is not available.

TCP is not designed for any particular lower-level protocol, nor does it make many demands on the protocol, so just about any protocol can be used under it, though if the protocol provides much besides an unreliable datagram service there will be duplication of effort. Changes in lower-level protocols will be transparent to TCP users.

A true end-to-end protocol like TCP, and to a lesser extent the "end-to-end" version of X.25, seems more directly useable by application programs, since a sender will often require reassurance that its messages are received. For example, a file transfer program will certainly need such feedback, especially in the case where the local copy of a file is to be deleted after it is sent.

A netowrk offering TCP implemented with a datagram protocol might also offer the datagram service directly, giving the user a choice of type of service.

INTERNETTING

X.25 has no provision for internetting. A separate CCITT recommendation, X.75, provides for internet virtual calls between hosts in separate nets by concatenating intranet calls between the hosts and whatever gateways intervene.

TCP has no trouble with internetting, since only the two communicating hosts are involved and host addressing is done by a lower level protocol. If this protocol provides internetting, as does the recommended Internet Datagram Protocol, so does TCP.

Sending datagrams between networks is much simpler and more efficient that establishing multiple virtual circuits between them. In addition, the local flow control of X.75 could cause multi-network congestion.

STANDARDIZATION

X.25 has been recommended by the CCITT, accepted by a number of carriers in different countries and will probably be accepted by tne NBS. Western Digital willsoon release a chip that implements level two.

TCP is the DoD inter-process communication standard, but has not been accepted by any standards organizations.

X.25 is a network-host interface standard and attempts to completel specify the interface and, in so doing, the protocol. It does not specify a user interface.

TCP is merely a protocol standard with recommendations for the datagram interface at one end and the user interface at the other. This added flexibility may or may not be an advantage.

SIMPLICITY

Neither protocol is particularly simple to implement. Each works best on a machine with convenient access to bytes and (to a lesser extent) nibbles. In addition, TCP likes a machine what can operate on 32-bit words. Of course these requirements can be taken care of by software and slight additional cost. TCP also requires help from the operating system. In particular, the system should be able to communicate asynchronously with a running process, since the process is told of exception conditions asynchronously, and sophisticated implementations will allow multiple sends and receives to be pending with asynchronous notification of success or failure.

One very important distinction is that all three levels of X.25 must be implemented both on hosts and network nodes, whereas network nodes supporting TCP need only implement the underlying datagram protocol, since TCP is strictly a host-to-host protocol. This will result in much simpler software for the network than required for X.25.

ORDERING OF DATA

Both protocols provide correctly sequenced data. X.25 sequences packets and TCP sequences bytes, but the difference is in flow control and retransmission, not in ordering of data.

EXTRA FEATURES

X.25 provides a true interrupt facility; TCP allows higher-priority "urgent" data which is not a true interrupt.

TCP allows sockets to be multiplexed.

X.25 allows permanent circuits.

APPLICABILITY TO A CABLE-BUS NETWORK

In a local area cable-bus network, node-to-node reliability is high and delay is small when compared with typical long distance nets. Also packets can only be received out of sequence if they are sent out of sequence (e.g., a retransmission).

It seems that TCP's sequence space 2^{32} bytes) is far too large, since windows on such a reliable, well-sequence, low-delay network may be small. It may be that even X.25's small sequence space(2^3 packets) is larger than necessary, although it is already so small that there would probably be little advantage in shrinking it unless a window of size one could be used.

X.25 depends upon the reliability of its components for reliabilit This is quite convenient since the CSMA/CD protocol by necessity provides a reliable node-to-node protocol. However, two error checks are required by X.25: one on the source to source-node tran mission, and another on the destination-node to destination transmission. TCP requires only one end-to-end check. Also an end-to-end check is inherently more reliable, and is not very painful when delay is short. However, X.25's local error checking is more likely to be performed in hardware (and therefore faster) either in an X.25 level two chip or at least partially in an HDLC chip. It is unlikely that similar chips will be available for TCP, since it is at a higher level and not as widespread.

Since cable-bus networks are likely to be small with cost a major consideration, the simpler network nodes required for TCP may be a significant advantage.