1987

Some Experiments with TCP/IP Protocols over an X.25 Network

John T. Korb
Purdue University, jtk@cs.purdue.edu

Craig E. Wills

Report Number:

87-664

Some Experiments with TCP/IP Protocols over an X.25 Network*

CSD-TR-664

John T. Korb
Department of Computer Science
Purdue University

March 1987

Abstract

This report describes some experiments to test the viability of high-speed (56K bps access line) X.25 network service as a carrier for IP datagrams. The conclusion, based on results with the ACCUNET service provided by AT&T, is that X.25 can provide a reasonable carrier, but that TCP requires careful tuning to take advantage of the reliable, but long delay, service.

1 Background

The Defense Advanced Research Projects Agency (DARPA) protocol suite, including the Internet Protocol (IP) [Pos81a] and the Transmission Control Protocol (TCP) [Pos81b], have become widely used protocol standards for both local-area and long-haul networks. Their availability for many computers and operating systems, as well as their applicability to a wide range of physical networks, have made them the protocols of choice at many universities and research centers.

But, the TCP/IP protocols are just that—protocols. In order for remote sites to communicate, a long-haul carrier between them is necessary. One such choice is the ARPANET [DAR81], but the ARPANET is not available

*This work was supported by the National Science Foundation under grant number ASC-8412035.

to all sites. Another choice is CYPRESS [CNY86], which provides 9600 bps connections between sites. An intermediate option is to use one of the several "public networks" such as GTE Telenet or AT&T ACCUNET. Since the public networks offer only the CCITT X.25 protocols [CCI80], TCP/IP communication is not directly available.

To be able to use the publically-available packet-switched networks with the TCP/IP protocols a software interface is necessary to encapsulate IP datagrams and send them over X.25 virtual circuits. A mechanism for doing just that was developed in 1983 at Purdue [CK83a] for use with CSNET. The techniques were standardized [Kor83] and adopted by others, including the BBN VAN gateway and the University College London gateway.

The basic idea of the IP-to-X.25 interface is to open an X.25 virtual circuit to each site for which outbound IP datagrams are queued. The interface software opens circuits on demand, monitors their use, and closes them when traffic to the site has stopped. Each IP datagram is sent as an X.25 "complete packet sequence".[1]

Experience with the IP-to-X.25 interface developed for CSNET demonstrated some problems with X.25 service as a carrier for IP datagrams. The main problems are caused by (1) the 9600 bps access lines and (2) the long network delay coupled with a small, two-packet window available under existing implementations of X.25. The latter problem, while solvable in the long term as network vendors provide lower delay service and larger windows, was relieved somewhat in the short term through the technique of opening multiple X.25 virtual circuits as necessary to destination sites [CK83b].

The other problem–slow speed access lines to the X.25 service–is the subject of this report. With support from the National Science Foundation,[2] a set of experiments were devised and conducted to determine the feasibility of using high-speed access lines to publically-available X.25 network service as a means for connecting TCP/IP-based sites. The results of this study are the subject of this report.

[1] The software makes no attempts to map TCP virtual circuits to X.25 virtual circuits. Attempts to do this kind of protocol conversion have had little success [DC79].

[2] "Feasibility Studies of High-Performance Communication Over Public Packet-Switched Networks", grant number ASC-8412035, J. T. Korb and Douglas E. Comer

2 Experimental Apparatus

The experiments used the ACCUNET X.25 service provided by AT&T during the months of July–September, 1986. There were two connections to ACCUNET, both using 56K bps serial lines.

One connection was to a VAX 11/750 at CSNET headquarters (Bolt Beranek and Newman, Incorporated) in Cambridge, Massachusetts. The other connection was to a VAX 11/780 at Purdue University in West Lafayette, Indiana.[3]

Both machines were running the same version of the UNIX operating system (4.2BSD).

3 Experiments

This section documents several of the experiments performed using the apparatus described above.

3.1 Responsiveness Test

To simulate interactive response, the ping program [Mil83] was used to send packets at one second intervals between the two hosts. The ping program reports the round trip delay experience by each packet. Two tests were done: one with small packets (64 bytes) and one with large packets (512 bytes). The tests ran for an 11 day period (July 1 through July 11, 1986), and are summarized in Table 1.

The tests were done by sending 100 64-byte packets, followed by 100 512-byte packets, every 30 minutes (400 packets per hour). Unfortunately, at the start of each test there were no open X.25 virtual circuits, so the first of the 64-byte packets were delayed while a circuit was opened. This delay—as much 20–30 seconds—affected the average delay reported by ping in a way that does not accurately reflect expected response time. Consequently, the results summarized in the table include entries for packets with sequence numbers between 50 and 100, under the assumption that by the time the 50th packet has made it through the network the circuit has been opened and the backlog of packets have been delivered.

Finally, since a few packets with extremely long delays can severely affect the computation of the average delay, the table also shows the median delay.

[3] The connection to Purdue was made to the Central Office in Chicago, Illinois.

size (bytes)	*statistic*	*sequence*	*delay (ms)*
64	average	1–100	1992
64	average	50–100	561
64	median	1–100	340
64	median	50–100	330
512	average	1–100	1246
512	average	50–100	1270
512	median	1–100	880
512	median	50–100	880

Table 1: Summary of Round-Trip Packet Times

For comparison, small packets on a local area network take around 10 ms. A small packet ping across the ARPANET showed delays ranging from 210 ms to 3558 ms, with an average around 1000 ms (ARPANET measurements done in October of 1986).

The test period lasted 235 hours. During that time, the VAXes being used for the tests were unavailable 15 hours (7%). The network was unavailable 45 hours (19%).

3.2 Time of Day Tests

The data from the ping tests also provide a measure of how response varies by time of day. The results of these tests showed no consistent variation by time of day. Figure 1 shows the minimum, maximum, and average 64-byte round-trip delay for one day during the test. Figure 2 shows the same data for 512-byte packets.

Delays summarized by time of day over nine days during the test period (omitting the first and last partial days) are given in the next two figures. Figure 3 shows the *average* packet delay for both 64-byte and 512-byte packets. Figure 4 shows the *median* packet delay.

3.3 File Transfer Test

Using FTP, a 175,426 byte file was transferred between the two machines. Retrieving a file (using the FTP command `get`) from CSNET to Purdue was consistently faster than storing one (using the FTP command put).

Results for five tests (as reported by FTP) are given in Table 2.

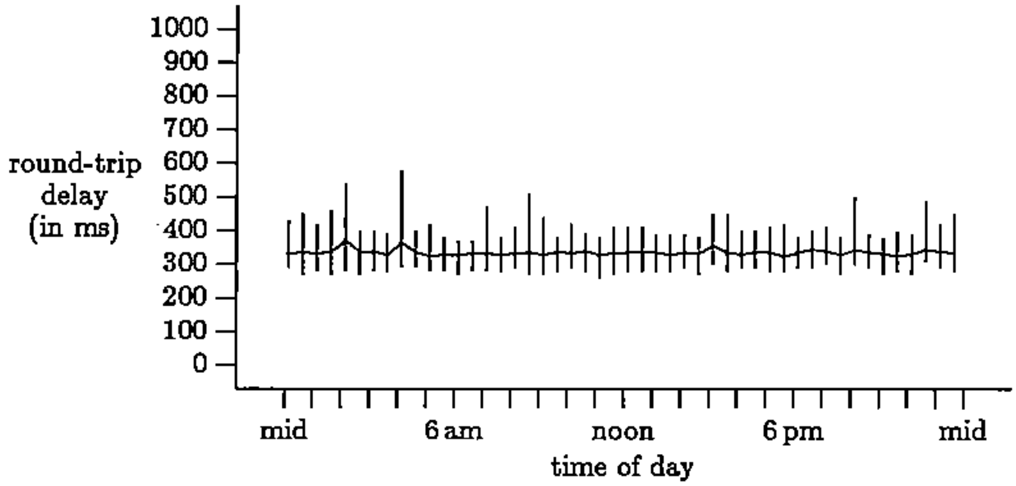

Figure 1: Average Delay of 64-Byte Ping Packets on Day 10

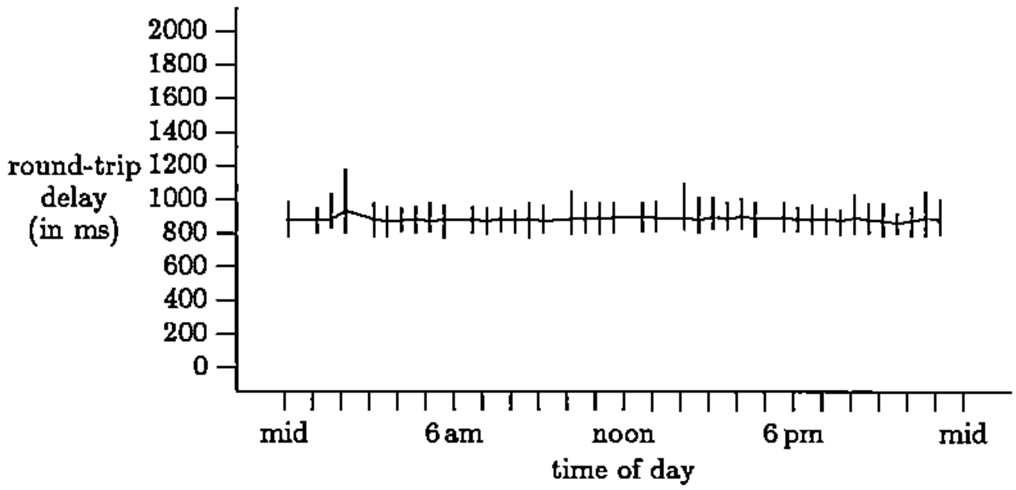

Figure 2: Average Delay of 512-Byte Ping Packets on Day 10

direction	*reported transfer rate*
put	15,012 bps
get	19,997 bps

Table 2: Summary of File Transfer Tests

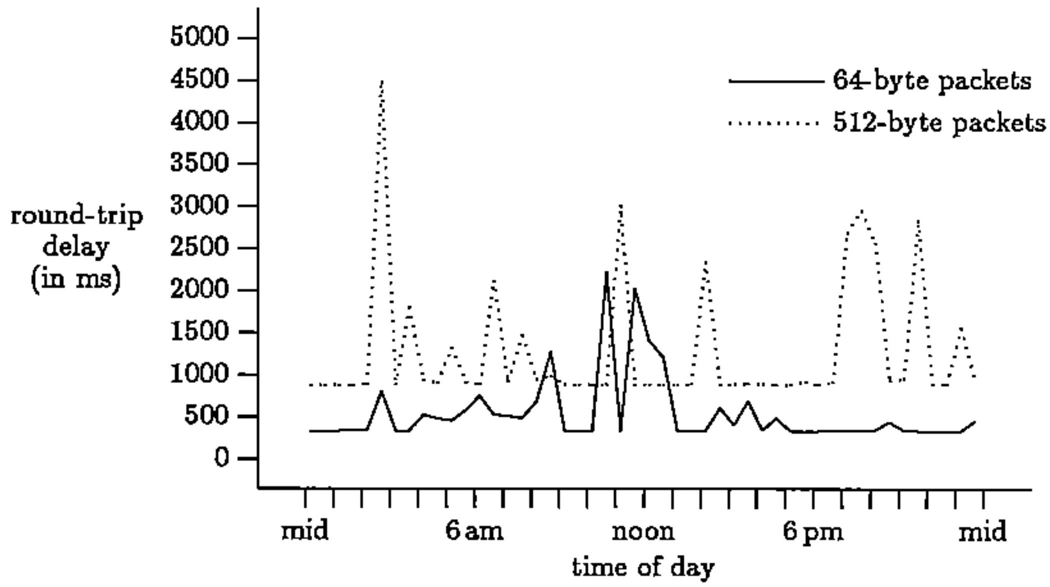

Figure 3: Average Packet Delay over Nine Day Period

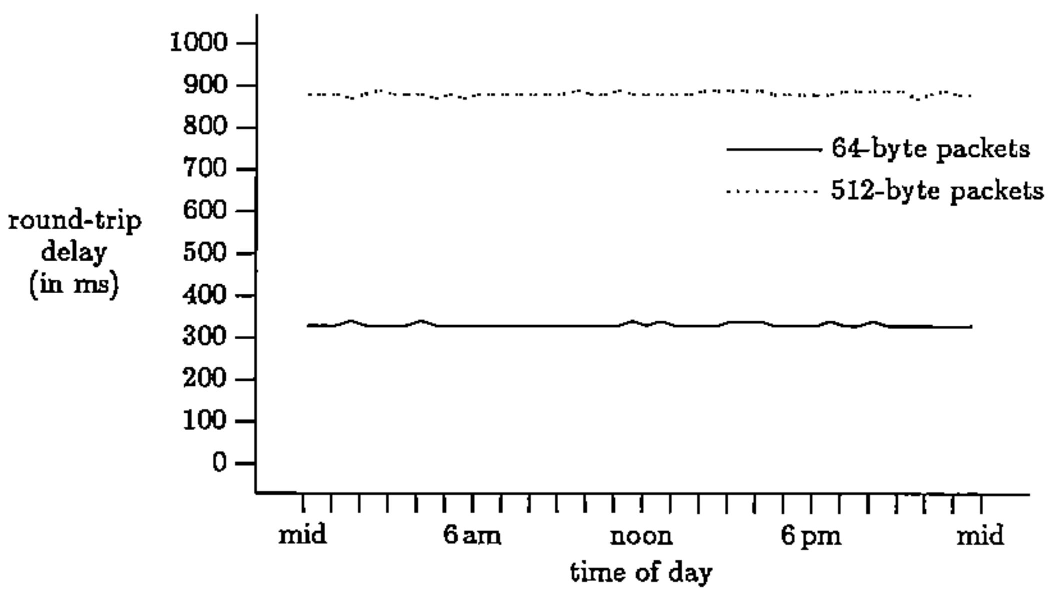

Figure 4: Median Packet Delay over Nine Day Period

3.4 Multi-Channel Tests

The transfers in the previous test were run at a time when the code that automatically opens multiple channels as load increases was not working properly. This code allows establishment of multiple X.25 virtual circuits between the two hosts during a single file transfer. With the correct code installed, the tests were repeated with a limit of first two, then four, channels between the sites. The results are summarized in Table 3.

number of channels	*reported transfer rate (for put)*
2	12,604 bps
4	19,254 bps

Table 3: Summary of Multi-Channel FTP Tests

3.5 Multi-Put Tests

The major limitation on the effective transfer rate appears to be caused by the interaction of the relatively long delay over ACCUNET (compared to a local area network) and the relatively few number of outstanding packets allowed by the current FTP/TCP implementation. Thus, despite the fact that we are using 56K bps connections, the line was idle much of the time while the higher level protocols waited for acknowledgements. To overcome this problem, I did a series of tests transferring several files at the same time. In this way, the load offered to ACCUNET would be greater than what it was willing to accept. Examination of the line with a data monitor showed that there was, in fact, more activity on the line in this experiment compared to the last one.

The problem with this experiment, however, is in interpreting the results. I transferred five 100K byte file simultaneously. The results reported by each FTP process were quite slow—since each process was only getting 1/5th of the communication line. The results in Table 4 give the time in seconds of the longest file transfer, which, since all transfers started at the same time, are approximate upper bounds on the data rate for transferring $5 \times 100\text{K} = 500\text{K}$ bytes.

Tests marked with an asterisk (*) reported only four of the five transfers completed successfully. Ignoring these tests, the reported maximum times increase dramatically at first, but level off with six or more open channels.

number of channels	*maximum time (seconds)*	*estimated rate (bps)*
1	3645	1,097
2	177*	—
3	244	16,393
4	111*	—
5	127	31,496
6	119	33,613
7	119	33,613
8	114	35,088
9	116	34,483
10	84*	—

Table 4: Summary of Multiple Simultaneous File Transfers

This observation, coupled with the fact that the monitoring systems show packets being queued waiting for the network to accept them, suggest that the network is applying flow control to prohibit saturation.

Assuming that the maximum reported times are an upper bound on the time required to transfer 500K bytes, the times above suggest a maximum transfer rate of

approximate maximum transfer rate
35,000 bps

This transfer rate should be achievable for a single file transfer by modifying TCP/FTP to use larger TCP segments so that more data can be presented to the ACCUNET interface.

3.6 Saturation Test

To test the carrying capacity of the network and how it reacts under heavy load, the network was monitored during the multi-file put test described above. The monitoring included observing at regular intervals (every minute) the following counts:

- the number of output packets presented to the X.25 Network Interface (XNI) by the IP layer,
- the number of output packets presented to the INcard (X.25 hardware interface) by XNI, and

- the number of bits actually sent on the network (excluding X.25 overhead, but including TCP/IP overhead).

The results are shown in Figure 5. This test monitored six transfers over an 80-minute period. Each transfer can be seen as a peak in the graph. The typical *raw* data rate was approximately 40K bps, with peaks to over 50K bps. Estimating TCP/IP overhead at 10%, these figures give an estimated achievable transfer rate of 36K bps, which agrees with the analysis in the previous section.

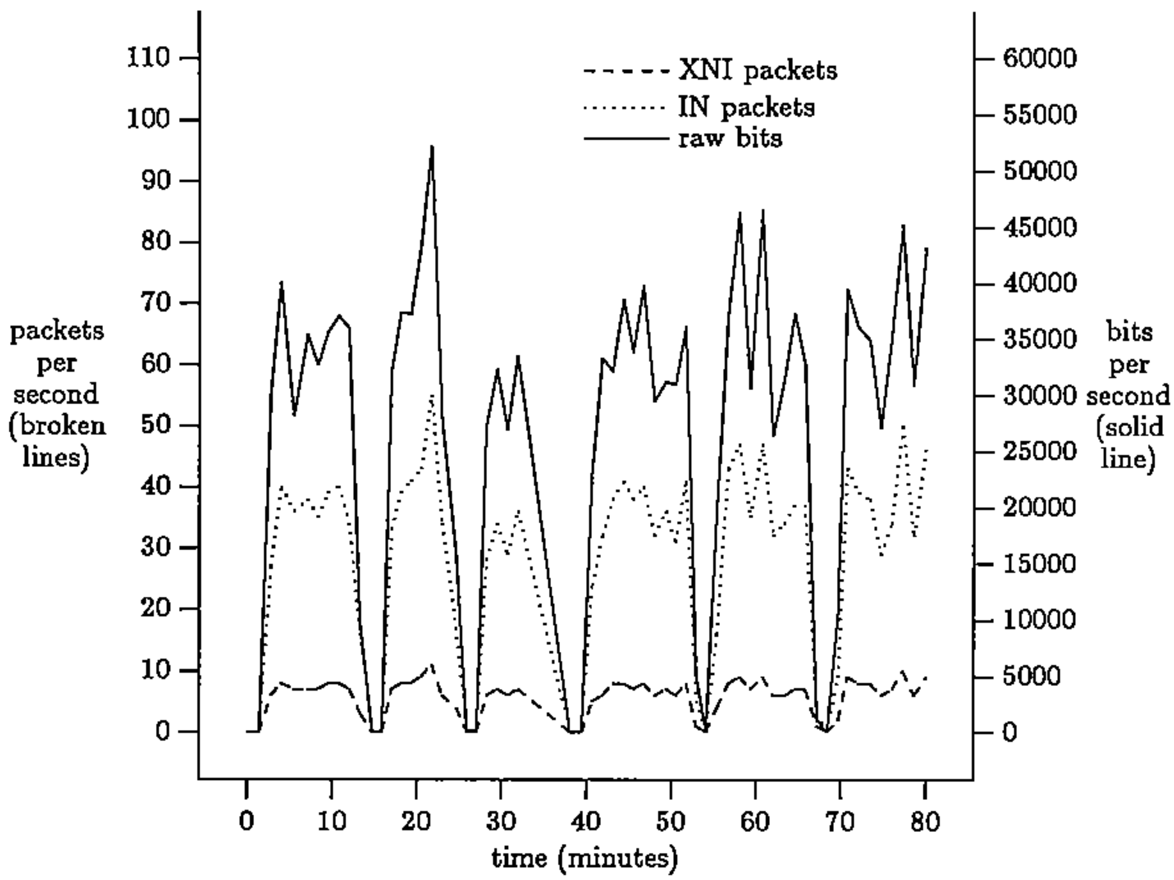

Figure 5: Traffic Monitor During Six Multi-File Transfers

4 Conclusions and Summary

There are several points worth mentioning concerning the use of TCP/IP over an X.25 network as well as comments specific to the ACCUNET X.25 service provided by AT&T.

- The ACCUNET service evaluated at the time of these experiments (Fall 1986) was capable of handling 63% of the 56K bps access line capacity.

- Response to small ping packets indicated that the network worked well for interactive sessions. In fact, during the tests, the ARPANET performance was signicantly worse—I used ACCUNET for all interactive sessions between Purdue and BBN.

- The time for ACCUNET to open an X.25 circuit seemed large (20–30 seconds).

- The ACCUNET network availability and reliability were erratic during the testing period. These problems may have been solved (the network was just getting underway at the time of the tests), but should be investigated before contracting for additional service.

- Multiple X.25 circuits must be opened to improve FTP throughput. The X.25 window size is so small and the delays large enough that only a few packets can be outstanding at a time. The right number of circuits requires further study, and depends heavily on the timeout and retransmission algorithm used by TCP.

- During most tests, packets (IP datagrams) were rarely lost. The long delay present when the network was saturated, however, appeared to cause many—unnecessary—TCP retransmissions. This point indicates that the TCP timeout period must be carefully adjusted for X.25 carriers. (The timeout cannot be ignored, since IP datagrams may still be lost or discarded at other protocol levels.)

- The results from the multiple simultaneous transfer tests indicate that the network can handle more data than is being offered by a single file transfer. TCP and FTP need to be able to choose a larger segment size and/or sending window to keep the network busy when several X.25 circuits are open.

The conclusion is that X.25 networks—including the AT&T ACCUNET service—can provide reasonable transport of IP datagrams, responsive TCP connections (TELNET sessions), and adequate FTP throughput, if multiple X.25 virtual circuits are used and TCP segment sizes and transmission timers are adjusted appropriately.

References

[CCI80] CCITT. *Data Communications Networks: Recommendations X.1–X.29.* Volume VIII, International Telecommunications Union, November 1980. Aka, *The Yellow Book.*

[CK83a] Douglas Comer and John T. Korb. CSNET protocol software: the IP-to-X.25 interface. In *SIGCOMM Symposium on Communications Architectures and Protocols,* March 1983.

[CK83b] Douglas Comer and John T. Korb. *The Proposed DARPA IP-to-X.25 Interface Standard: Performance Optimization with Multiple Circuits.* Technical Report CSD-TR-473, Department of Computer Science, Purdue University, October 1983. Revised May 1984.

[CNY86] Douglas Comer, Thomas Narten, and Raj Yavatkar. *The Cypress Network.* Technical Report CSD-TR-653, Department of Computer Science, Purdue University, December 8, 1986.

[DAR81] DARPA. *A History of the ARPANET: The First Decade.* Technical Report, Defense Advanced Research Projects Agency, April 1981.

[DC79] S. Das and R. Cole. *Protocol Converter, Transport Level Specifications.* INDRA Note 806, Department of Statistics and Computer Science, University College London, 1979.

[Kor83] John T. Korb. *A Standard for the Transmission of IP Datagrams Over Public Data Networks.* RFC 877, Department of Computer Science, Purdue University, September 1983.

[Mil83] D. L. Mills. *Internet Delay Experiments.* RFC 889, Linkabit Corporation, December 1983.

[Pos81a] Jonathon B. Postel. *The DoD Standard Internet Protocol.* RFC 791, Information Sciences Institute, September 1981.

[Pos81b] Jonathon B. Postel. *The DoD Standard Transmission Control Protocol.* RFC 793, Information Sciences Institute, September 1981.