IEN 151

Vint Cerf
DARPA/IPTO
1 April 80

FINAL REPORT OF THE STANFORD UNIVERSITY TCP PROJECT

Vinton G. Cerf
1 April 1980

ABSTRACT:

This report provides a brief historical and technical summary of the Stanford University internet, host-to-host Transmission Control Protocol (TCP) project. Developed from 1973-1976 at Stanford, Bolt Beranek and Newman and University College London, the effort then continued at other research and development centers during 1977-1980. Originally designed as a monolithic internet protocol, the internet aspects were separated into a distinct protocol layer in early 1979 with the publication of version 4 of TCP. The resulting pair of protocols, TCP4 and Internet Protocol (IP) are now undergoing procedures for standardization within the DoD and Intelligence Community. The four most valuable results of the Stanford effort were the assessment of which functions could or should be implemented in the protocol and which should not, the implementation of an efficient, assembly-language version of TCP for an LSI-11 computer, the development of a "micro" operating system (MOS) for the same computer, and the specificiation of the TCP protocol.

1. Introduction:

The Stanford TCP project was supported by the Defense Advanced Research Projects Agency (DARPA) under contract No. DAHC 15-73-C-0363 and MDA903-76-C-0206, ARPA Order No. 2494 during the period 1 July 1973 to 30 September 1976. During the time that the project was at Stanford, two versions of TCP were developed, one in December 1974 [1] and another which was published in March 1977 [2], by DARPA. Since that time, two other versions have been published, versions 3 and 4 [3,4] in January 1978 and February 1979, respectively. Editing of these latter versions was the responsibility of J. Postel, Information Sciences Institute, University of Southern California. Intermediate versions were published in July 1976 [5], July 1978 [6], and August 1979 [7]. The July 76 version was developed for the Defense Communication Agency by J. Postel, L. Garlick and R. Ram of Stanford Research Institute for the Defense Communication Agency in connection with the AUTODIN II project. The final verions of TCP and Internet Protocol were published in January, 1980 [14,15] by DARPA on behalf of the Defense Communication Agency.

During the course of the DARPA Internetting program, which supported the TCP development, a great many people and groups were involved in or influenced the development of the TCP.

The initial impetus for the effort resulted from work by R. Kahn and V. Cerf, which was published in May 1974 [8], but whose basic roots were already known in September 1973 [9].

An initial design for TCP was worked out during 1973-74 at Stanford among V. Cerf, Y. Dalal, C. Sunshine and R. Karp, with the participation of R. Tomlinson [Bolt Beranek and Newman], W. Plummer [BBN], R. Metcalfe [Xerox PARC] and D. Boggs [Xerox PARC]. Implementation, testing and further development were carried out jointly at Stanford with J. Mathis and J. Estrin, Bolt Beranek and Newman, and University College London [F. Deignan, C. J. Bennett, A. J. Hinchley and M. Galland]. Visiting at Stanford during this initial development period were G. LeLann [University of Rennes, France] and D. Belsnes [University of Oslo] who provided additional philosophical leavening which influenced the design of the protocol.

By 1976, implementations had been written for the Tenex/PDP-10 [Tomlinson, Plummer - in BCPL], ELF/PDP-11 [Tomlinson, Plummer - in BCPL; Karp, Dalal, Cerf - BCPL], LSI-11 [Mathis - assembly language], PDP-9 [Deignan - BABBAGE] and some performance experience obtained [10].

Since then, implementations have been pursued for UNIX [M. Wingfield - "C"; J. Haverty - assembly language], OS 360 [B. Braden - assembly language], Multics [D. Clark - PL/1], TOPS-20 [W. Plummer - assembly language], and NORD-10 [A. Stensby].

2. TECHNICAL ISSUES

The initial concept behind TCP was very simple. Two processes wishing to communicate had only to know the other's "address". Data would be accounted for in 8-bit octets, and sequence numbers would be used to re-order the received data at the destination, if necessary. The first packet would have a special synchronization flag ["SYN"] which would alert the receiver that the sender's sequence numbers would start with the one associated with the "SYN" packet. All control information would be associated with data sequence numbers so that end to end acknowledgements for data could also be used to acknowledge control. If resynchronization were needed, the sender could simply send another "SYN" packet. There would be no used for a "connection set-up" in the conventional sense. Finally, packet formats would permit internet gateways to fragment packets into identically formatted, smaller packets if necessary, to get through a net with a smaller packet size. Reassembly of fragments would be done by the destination.

An initial implementation of the protocol by Tomlinson and Plummer was used to control a line printer which was supposed to spool output from a host computer onto paper. The first obvious problem which cropped up was an inability to distinguish an old duplicate SYN packet (created by retransmissions in the absence of an acknowledgement) from a new SYN packet which is trying to start a new sequence.

This problem was documented in [11-13] and the solution proposed was called the "three-way handshake" by its inventor, Ray S. Tomlinson. This addition to the original proposed protocol [8] was probably the most major change to its philosophy, since it introduced an explicit connection "set-up" exchange at the beginning to allow both ends to validate the connection sequence numbers. Basically, each end selects a starting sequence number, and receives both an acknowledgement and an indication of the start of the other end's sequence space. This validating exchange is intended to help each end distinguish old, duplicate SYNs from recent ones. For this to work correctly, some maximum packet lifetime in the internet system had to be assumed and a maximum throughput, so as to be able to select a suitably-sized sequence number space. In the end, a 32 bit sequence space was selected which would not wrap around for about 4 hours at 2 megabits/second.

An alternative to the three-way handshake was examined in 1978 by D. Reed at MIT [16], in which the initiator of a "connection" employs a unique, non-reusable "socket identifier" as a way of distinguishing new from old connection requests. These identifiers are needed in TCP also, to distinguish among multiple connections, but the socket numbers are allowed to be reused, leading to the need to use sequence numbers and the three-way handshake to distinguish old from new connection requests. The strategy proposed by Reed required some non-volatile supply of new socket identifiers which would either never recur, or recur at such long cycles (days, weeks...) that there would never be any confusion or duplication of use. Obvious, if the socket identifiers cycled too quickly, a particular identifier might present itself for use anew while it was still in use on some connection. This idea was considered for TCP in 1978, but was not pursued as it created problems of "dangling, half-open server connections" when things went wrong.

The second major issue had to do with the resynchronization of a connection after host failure. A variety of techniques were explored and designed in detail. In the final TCP design, a simple strategy for recovering from "half-open" connections was developed.

A hazard was found, however, based on violation of the assumption that sequence numbers would be consumed at no more than some maximum rate. If a connect stayed idle for a long enough period, the basic "initial sequence number" selection strategy outlined in [11-13] would encounter a hazard if the host failed just as the "next sequence number" to use fell into the so-called initial sequence number "forbidden zone". A complex set of tests were defined to catch this case, but required action on each packet transmitted to determine whether the hazardous "forbidden" zone had been entered. If so, the sequence numbers on the connection would be re-synchronized to skip around the danger area. The probability of the hazard actually occuring was judged to be quite small, and finally these tests were eliminated, vastly simplifying the TCP definition and implementation.

The third critical issue had to do with the treatment of packets which required fragmentation to pass through a network. Gateways between nets were postulated which could detect that an incoming packet was too large to fit encapsulated in the packet format of the next network.

A fragmentation strategy was developed which permitted an internet packet to be fragmented into smaller packets and marked in such a way that the resulting fragments could be routed independently of one aother and could still be reassembled at the final destination.

A major change to the TCP philosophy occurred when the basic internetting functions (addressing, fragmentation and type of service selection) were separated from the end to end functions of TCP (sequencing, retransmission, duplicate detection, flow control and port multiplexing). At this point, the fragmentation problem was substantially simplified since it applied only to the internet packets and limited knowledge of protocols to the IP layer in the gateway.

At the same time, this permitted other higher level protocols, adjacent to the TCP layer, to rely on the special services, including fragmentation and reassembly, of the IP protocol layer.

Network Security

The TCP concepts were applied to the ARPA network security program and an architecture was developed which accommodated the use of TCP as an end to end protocol, below which, end to end encryption could take place. This architecture became even simpler when the TCP was split from the internet protocol functions since security was provided at the IP level, allowing many different transport protocols, in addition to TCP, to be secured by the same basic system.

Implementation and Experimentation

Two TCPs were developed during this project:

1. BCPL for PDP-11 under ELF operating system.

2. Assembly Language for LSI-11 under MOS operating system.

The former was useful for high-level description of the TCP functionality but was never very efficient. Extensive performance tests and timing analyses revealed that 75-80% of the overhead for the BCPL TCP running under ELF was attributable to either the slow interprocess communication system in ELF or the inefficient "Reliable Transmission Protocol" for the Very Distant Host interface to ARPANET.

The RTP introduced a minimum round-trip time to the IMP and back of some 50 ms, limiting packet rates to about 20 packets/second.

Performance tests conducted with a Babbage implementation at University College London [10] were even less satisfactory due to buffer limitations in the London PDP-9 computer and the very lengthy cross-net delays imposed by a 9.6 kb/s satellite circuit connecting London to the U.S. by way of Norway. The best packet rate we could sustain was about 2-3 packets/second.

We found a significant improvement in performance to be possible upon using ARPANET "type 3" packets which are not limited to 8 outstanding messages as is normal ARPANET traffic. However, this mode of operation, if stressed, caused serious network congestion problems.

To determine whether the limited performance of TCP was a design flaw, we implemented an assembly language version in about 1200 words of LSI-11 memory running under a small, 800 word micro-operating system. This version, while compatible with our 12,000 word BCPL version on ELF, was one tenth the size and achieved 37 kb/s on the LSI-11 and 50 kb/s on the PDP-11/20.

The LSI-11 version was subsequently integrated in to the packet radio network at SRI as part of a Terminal Interface Unit.

Conclusions

This project successfully developed, implemented, documented and tested a reliable, internet work, host-to-host protocol, capable of operating in extremely hostile environments and able to recover from catastrophic failures in a graceful fashion.

The adoption of the protocol for DoD use in packet networking is an indication of the far-rearching impact of this research.

REFERENCES

1. Cerf, V., Y. K. Dalal, C. A. Sunshine, "Specificaation of Internet Transmission Control Program", INWG General Note 72, IFIP Working Group 6.1, December 1974.

2. Cerf, V., "Specification of Internet Transmission Control Program - TCP (Version 2)", March 1977.

3. Cerf, V., J. Postel (eds), "Specification of Internetwork Transmission Control Program - TCP (Version 3)", January 1978.

4. J. Postel (ed), "Transmission Control Protocol - TCP (Version 4)", February 1979.

5. J. B. Postel, L. L. Garlic, R. Rom, "Transmission Control Protocol Specification", Augmentation Research Center, Stanford Research Institute, Menlo Park, CA, 15 July 1976 [for the U.S. Defense Communications Agency in connection with AUTODIN II].

6. "Specification of Internetwork Transmission Control Program - TCP (Version 3.1)", July 1978.

7. DARPA, "Transmission Control Protocol, Version 4", August 1979 [edited by J. Postel].

8. Cerf, V. G. and R. E. Kahn, "A Protocol for Network Intercommunication", IEEE Transactions on Communications, Vol. Com-22, No. 5, May 1974.

9. Cerf, V. and R. E. Kahn, "Proposed Protocol for Internet Host-to-Host Communication", INWG Note No. 39, September 1973.

10. Bennett, C. J. and A. J. Hinchley, "Quantitative Measurements of the Transmission Control Protocol", Proceedings of the Int'l Conference on Protocols, Andre Danthine [ed].

11. Tomlinson, R. S., "Selecting Sequence Numbers", INWG Protocol Note No. 2, August 1974; Proceedings of the ACM SIGCOMM/SIGOPS Interprocess Communication Workshop, (Santa Monica, CA, March 24-25, 1975) and ACM Operating Systems Review, Vol. 9, No. 3, July 1975, Association for Computing Machinery, New York, 1975.

12. Dalal, Yogen K., "More on Selecting Sequence Numbers", INWG Protocol Note No. 4, IFIP Working Group 6.1, August 1974; also, in

Proceeding of the ACM SIGCOMM/SIGOPS Interprocess Communications Workshop, (Santa Monica, CA, March 24-25, 1975) and ACM Operating Systems Review, Vol. 9, No. 3, July 1975, Association for Computing Machinery, New York, 1975.

13. Dalal, Yogen K., "Establishing a Connection", INWG Protocol Note 14, IFIP Working Group 6.1, March 1975.

14. DARPA, "DoD Standard Transmission Control Protocol", (J. Postel, ed.), January 1980.

15. DARPA, "DoD Standard Internet Protocol", (J. Postel, ed.), January 1980.

16. Reed, David P., "Naming and Synchronization in a Decentralized Computer System", MIT Report LCS/TR-205, September, 1978.

17. Sunshine, C., "Interprocess Communication Protocols for Computer Networks", Ph.D. Dissertation, Stanford University, December 1975.