IEN-68

27 June 1978
Jon Postel
ISI
27 June 1978

TCP Meeting Notes - 15 & 16 June 1978

Introductory Remarks - Clark

Dave told us about the local arrangements.

Objectives of the Meeting - Cerf

Vint presented his goals for the meeting:

The format of the TCP and INTERNET headers is to be firmly decided at this meeting.

The schedule for implementation of version 4 is to be established.

The schedule for Telnet and FTP running on TCP is to be established.

The remaing inconsistencies in the TCP-4 specification should be identified and corrected. The application or user level interface to TCP should be clarified.

The structure and interpretation of an internet address should be established in both bit form and symbolic form.

The textual form of the internet address should be specified. There is currently some discussion of the user representation of addresses in the ARPANET brought about by the change to the 96 bit host/imp interface leader.

Vint indicated that the whole ARPANET community should expect to move to using TCP. That the internet environment may be much more important as the ARPANET ages. And eventually the ARPANET may be replaced by some other network.

Vint then lead in revising the agenda as follows:

Status Reports

1. BBN: TENEX & TOPS-20 - Plummer
2. UCLA: 360/91 - Braden
3. SRI: MOS/ELF - Mathis
4. CCA: RSX-11 - Kuo-Mei
5. MIT: Multics - Clark
6. FORD: KSOS - Biba
7. BBN: Unix - Bressler
8. DTI: Unix - Grossman
9. BBN: Unix - Wingfield

10. LLL: Timer Protocol - Watson
TCP/Internet DG Format - Postel
Symbolic Addressing - Cerf
Telnet and FIP Interface - Postel
Test Schedules - Cerf
Working Group Formation - Cerf
Friday, 16 June
Working Groups
(a) TCP Specification - Sunshine
(b) Internet Addressing - Cohen
(c) Higher Level Protocols - Postel
(d) TCP Experiments - Clark
Reports from Working Groups
Agenda for Next Meeting - Cerf

Ed Cain reminded us that there was an Access Control action item from the Internet Meeting about "TCP Reconnection" that should be on the agenda. Vint reviewed the problem, and suggested that the problem could be solved without dynamic reconnection.

Status Reports

1. BBN: TENEX & TOPS-20 - Plummer

Bill reported that TCP 2.5 is now up on BBNC and can support two (2) connections. More connections could be supported with more memory allocated to the TCP. TCP 2.5 will be brought up on the following machines in this order: BBNC BBND SRI-KA ISIC ISIA. Tops20 release 3 will cause some problems in the plan of development.

Vint is very concerned about testing under fully loaded conditions. Also would like TCP 4 to be up by 1 October 78. That is a TCP 4 running on Tops20r3 by October to be demonstrated, with a real Telnet!

2. UCLA: 360/91 - Braden

Bob was unable to attend but supplied the following written report.

We have a module capable of sending and receiving Internet packets. Using an internet test driver under TSO, we have successfully sent packets to ourselves, with proper reassembly of odd fragments. This loopback is through the local IMP.

The original implementation plan had been to do simple unit

testing of modules under TSO (where we have a DDT-like debugger) and then move them into the NCP for integration and system testing. However, we have had considerable difficulty with debugging within the NCP environment, and in order to speed up development decided to move all testing out into TSO. A significant factor in this decision was the great difficulty in getting the necessary system test time during one of the year's heaviest periods of use.

To implement this decision, we have now built a "raw packet" interface to the NCP, allowing any process on the 360/91 to send and receive raw pakcets. This would be useful for future NMC-like facilities, for example. It allows us to build and debug almost all parts of the TCP/Internet mechanism within the safe environment of TSO, and to have the use of the TSO TEST facility.

This change in strategy has also affected our current direction. We had originally expected to move from Internet protocol implementation to TCP implementation, leaving for last the problems of interfacing to the existing system-call environment and user-level protocols within the NCP. It now appears that a more productive line of attack is to put together this user-level interface before we implement TCP. To that end, we are curently building a (simple) datagram interface between the user-level protocols and the Internet layer. This code will be executed under TSO along with a copy of the Telnet access method which normally executes entirely within the NCP. In short, we will have a User TELNET and Datagram protocol program running under TSO, using the raw-packet interafce for ARPANET access. This should only take a few weeks.

I wonder whether anyone else will have a Datagram Telnet?

3. SRI: MOS/ELF - Mathis

Jim said that the concersion of the existing TCP 2.5 to TCP 4 will begin in a few weeks and should be completed a few weeks after that. Hopes to get it done in August. There is a problem in testing due to an IMP port crunch.

Ray Tomlinson mentioned some measurement results that were obtained from the TCP 11 MOS 11/40 mini gateway. The path TCP->minigateway->TCP could send 200 packets per second. The TCP ultimate thruput was 65KB. This was with EOL processing on. For comparison it was said that the ultimate thruput for an NCP on Tenex is 83KB.

4. CCA: RSX-11 - Kuo-Mei

Kou-Mei and David Low reported that CCA has a version of FTP that runs on NCP on RSX11, and that they are now starting to work on TCP for RSX11. They think that TCP 4 may be up by the end of September.

5. MIT: Multics - Clark

Dave said that the Multics work on TCP has been in a hold state waiting for TCP specifications to settle down. Code was written for TCP 3. October still seems a reasonable goal to get Multics up with TCP 4. The difficulties are on the server side for telnet and ftp, and are primarily administrative policy issues.

6. FORD: KSOS - Biba

Ken indicated that Ford will be building a secure thing using TCP using languages C and Euclid.

7. BBN: Unix - Haverty

Jack reported that his group has had a TCP 2.5 running for 4 months on RCCNET. The core is TCP11 from Jim Mathis stuck into an environment for their Unix. They are working on improving performance by fixing the Unix/TCP enviromment interface.

At this point there was a diversion to discuss measurements, and the following item were named as being useful information to have about every implementation:

Philosophical Remarks
Machine
Operating System, Version
Implementation Language
Code Size
Buffer Space Used
Number of Connections Supported
Cost per Additional Connection
Delay per Packet
Bandwidth in Packets/Second and Bits/Second
CPU Utilization

8. BBN: Unix - Wingfield

Mike said his project is working on higher level protocols in the EON at DCEC. They are tracking TCP and will use TCP 4 plus the precedence and security features for AUTODIN II. THP (a version of Telnet) and FTP (a spec being written by BBN, available in January 1979) will also be implemented. Everything is being done in C. Project is to do extensive testing of the three protocols.

9. DTI: Unix - Grossman

Gary distributed a written note on the status of TCP work at DTI.

Background

DTI is constructing an IOC network front end (INFE) to interface WWMCCS hosts and terminals to the AUTODIN II network. The INFE uses a PDP-11/70 as its hardware base. DTI has modified the Bell System UNIX[tm] operating system to serve as the software base for the INFE. The modifications permit protocol interpreters to be implemented as processes running at the user level. Programs can access the protocol interpreter processes by using the standard UNIX I/O system primitives.

As a part of the INFE effort, DTI has developed a version 3 TCP. The DTI INFE TCP is operational; it awaits system integration for final debugging and performance measurement.

Development

Level of Effort: < 1 full-time person.

Schedule:

1 Dec 77	Study begun.
1 Jan 78	Version 3 study and design begun.
20 Feb 78	Coding begun.
8 Mar 78	Coding Completed.
22 May 78	Debugging to level sufficient to support the THP implementation.

Version

Internet: The DTI INFE TCP uses the internet header as defined in the IEN 26 of 14 February 1978 (Internet Notebook Section 2.3.2.1). It does not perform any internet reassembly. Four

octets have been added to the internet header to simulate AUTODIN II security and precedence functions.

TCP: The DTI INFE TCP implements TCP v/3 as of the January 1978 specification with the segment header as defined in the IEN 27 of 14 February 1978 (Internet Notebook Section 2.4.2.1). It discards data destined for a user process that has initiated a close. It forces an end of letter at the end of urgent processing in order to maintain consistent buffer management. This is not a general soultion to the problems posed by the interaction of the urgent mechanism and the buffering mechanism.

.Size

Source
Lines of C (actual code and data structures) : 5K
Lines of C (including commentary): 10K
Object
Fixed
Words of code: 9.6K
Words of tables and net buffers: 4.0K

Total: 13.6K
Per-Connection
Words of tables, etc.: 100
Words of buffer space: 250

Performance

Only very informal measurements have been performed to date. More careful and more extensive measurements will be performed once the complete INFE system has been integrated.

The measurements were performedon a single connection between a source process and a sink process through the DTI H516 IMP. The source and sink processes respectively wrote and read 512 octet buffers. ARPANET type 0 subtype 3 messages with extended Host/IMP leaders were used to carry the internet packets. Subtype 3 messages are limited to 135 octets, of which we use only 134. These octets are distributed among the functions as follows:

octets	function	relative size
12	ARPANET extended leader	09%
28	internet header	21%
16	segment header	12%
78	user data	58%
134	total	100%

Thus 42% of each packet is overhead.

Under these conditions the total throughput measured was 24K bits/sec (12K bits/sec of user data in each direction). This corresponds to a total packet processing rate of about 76 packets/sec (38 packets/sec in each direction).

It is not clear at this time to what extent the packet processing rate is a function of ACK delay and to what extent it is a function of CPU statuartion. In any event, the throughput in bits can prpbably be significantly increased by using larger internet packets (and thus reducing the leader and header overhead). This requires using ARPANRT subtype 0 messages instead of subtype 3 messages. The greater delay involved in using multi-packet messages could reduce the gains in throughput which increasing the message size would otherwise produce.

In addition Gary mentioned that using 1000 octet packets the DTI system has achieved a thruput of 46 KB in each direction.

10. LLL: Timer Protocol - Watson

Dick gave a brief overview of the LLL computer environment, with something like 40 computers interconnected with 50KB to 50MB technology, and gateways to other networks. They are trying to develop a new network approach to rationalize all the interconnections. They need a host-to-host protocol to solve relibility problems (losses, duplicates), but need to do transaction type services. They have developed a timer based protocol, that requies a bound on the maximum packet lifetime. They have now modified the "timer protocol" to also optionally use the three way handshake. They also need a very long address field in some cases since it is desirable to include a capability and its password in the address. They have a provision for using next level protocol information in reassembling fragments, so they don't need a fragment id as such.

TCP/Internet DG Format - Postel

Jon distributed the draft specifications of TCP 4 and Internet 4. He presented the new header formats and discussed the meaning of each field.

Much unrecorded discussion followed.

The result seemed to be to limit the address fields to a maximum of 15 octets (120 bits) and to delete the Pointer field. A source routing option is to be defined.

There was a little discussion of how to do multidestination or broadcast addressing, and the suggestion was put forward that one value of network should mean "group" and then the rest of the address should be a group name.

Symbolic Addressing - Cerf

Vint started this discussion by indicating the concern over user entered field oriented addresses in the ARPANET brought on by the advent of use of the 96 bit host/imp leaders. The TIP notation of HOST[/IMP] seems to some to be backwards, they feel things should go from general to specific in the normal left to right reading order. Most of the discussion suggested alternatives to even knowing about addresses, and using names instead. The general conclusion was that TIP notation for hosts in the ARPANET is decoupled from our concerns about internet names, addresses, and routes.

Telnet and FTP Interface - Postel

It was noted that a better discussion of transmission into a zero window would be helpful.

Gary Grossman made a presentation about some difficulties with the Urgent Mechanism.

A very large amount of discussion ensued.

At the end of the day it was decided that Vint and Gary would resolve the problem in the evening and present the solution Friday morning.

In the morning Vint began a presentation of the rules for Urgent indicating it would take about 5 minutes, an hour and a half later...

Working Group Formation

At this point it was decided to have two parallel working groups, one on Urgent, and the other on Addressing.

Urgent Mechanism Group - Cerf

Vint prepared the following notes summarizing the lengthy discussion:

The objective of the TCP URGENT mechanism is to allow the sending user to stimulate the receiving user to accept some urgent data and to permit the receiving TCP to indicate to the receiving user which octet in the received data is the last of the currently known urgent data.

The assumption made in providing this service is that the higher level will always transmit new data when URGENT is to be asserted. Typically, the higher level protocol may employ a special method to distinguish the URGENT data from ordinary data, e.g., by special format or coding conventions, but this need not be necessarily be the case.

An alternative to the URGENT mechanism was considered, namely to signal "interrupt" in a reliable fashion so that there would be a one-to-one correspondence between the number of interrupts sent and received by the using processes.

To achieve this signalling would have required a TCP like connection mechanism to deal with data loss, duplication and disordering and this consideration led us to try to economize on the amount of mechanism required to implement TCP.

The basic URGENT service can be described as follows:

When the user hands a buffer of data to the tcp to be sent, and asserts that it is urgent, the TCP assumes that the last octet of the urgent data coincides with the last octet of the buffer. Successive transmission of new urgent data causes the "end of urgent data" to extend farther into the data stream.

If the sending user asserts EOL when sending the urgent data, then the receiving tcp will attempt to deliver the data to the receiving user even if the buffer into which data is being assembled is not full. This is not unique to urgent data since EOL is the mechanism for the user to assert to the receiving TCP "deliver this without waiting for more data".

In any case, the receiving TCP will indicate to the receiving user precisely which octet of data is the last of the urgent octets. This is accomplished by associating with newly delivered data a pointer to the "end of urgent data".

The URGENT mechanism provides an out-of-hand signal which the sending user can employ to alert the receiving user to enter an "urgent" state. No semantics are assumed for this signal. Furthermore, there is no intent that every URGENT data transmission result in an URGENT signal to the receiving user. Instead, it is guaranteed that the receiving user will be signalled at least once when he should enter the urgent state and will later be told when he has received that last known (to the receiving tcp) urgent data.

The precise form of the "URGENT" signal is an implementation decision but it must be "out of band" with respect to delivery of normal data.

It is assumed that the user always provides new data to send when asserting URGENT. However, the TCP may not always be able to accept any new data to transmit (which is one reason for trying to assert URGENT). Then sending TCP will attempt to signal "URGENT" to the receiving TCP even if it cannot actually accept new data for transmission. To be consistent with the design of the URGENT mechanism, users which have attempted to send urgent data must continue to attempt to send this data until it is accepted or the connection is otherwise closed or aborted.

A consequence of the TCP's attempt to signal URGENT even when it cannot accept the new data for transmission is the receiving user may enter and leave the urgent state more than once before the desired urgent data is actually delivered.

The group also commented on Zero Windows and Close:

> The sending TCP must be prepared to accept and send at least one octet of new data even if the send window is zero. This is essential to guarantee that when either TCP has a zero window the re-opening of the window will be reliably reported to the other.
>
> Users must keep reading connections they close for sending until the TCP says no more data.

Addressing Working Group - Cohen

This group reviewed the use and meaning of the words broadcast and multidestination, and of the primary applications of these features - conferencing and mailing lists.

It was suggested that since the internet protocol is a very unreliable protocol and that since the applications that will make use of multidestination (and perhaps broadcast) will expect a high level of reliability, it was unreasonable to invest a lot of effort in defining a multidestination capability at the internet level until a more specific case could be made for the necessity of such capability.

The group then reverted to discussing regular addressing.

Jim Mathis suggested that one could view the header as a lot of little headers with a variable number of address headers or layers between the internet fragmentation on top and the host-to-host protocol in the middle, then more address layers and finally the application on the bottom.

Larry Stewart suggested that address elements be typed data, for example type codes could be net, host, port.

Ray Tomlinson suggested that it was useful to separate addressing from protocol, that to some degree one could use multiplexing procedure X with protocol Y, for some set of X and Y.

It was pointed out that one might want to use TCP with some other lower level environment, and in that case some other protocol would be necessary to carry the address information, and in some sense that made TCP incomplete.

It was decided (declared ?) that Port Id's (for the ARPA community) will be 16 bits.

Two votes were taken:

"Shall the Port be part of the Internet Header?"

Result: NO.

"Shall the Port be part of the TCP Header?"

Result: NO.

It was suggested that if the TCP and Internet protocols used the same checksum size and algorithm nice things would happen.

It was requested that all the fields that the TCP depends on should be checked by the TCP checksum, even if the fields are in the internet header, for example the addresses.

The format of the headers was decided. [See "Latest Header Formats" IEN 44.]

Agenda for Next Meeting - Cerf

1. Implementation Status of TCPs

2. Implementation Status of Telnets and FTPs

3. An N x M test of TCPs and Telnets.

The next meeting will be a TCP testing session 18&19 September at SRI.

Action Items:

1. TCP implementors are to document and distribute the user interface to their TCPs.

2. TCP implementors are to send a point of contact for TCP testing to Postel. Should include Name, network address, and phone number.

3. TCP implementors are to document and dustribute perforamace of their TCPs.

4. Wingfield is to distribute the BBN EDN FTP specification to this group when it is ready (estimated January 79).

Memos Distributed:

Postel: Draft Specification of TCP-4 [IEN-40]

Postel: Draft Specification of Internet-4 [IEN-41]

Braden: UCLA Status Report

Grossman: DTI Status Report

Attendees:

Name	Affiliation	15th	16th	Mailbox
Vint Cerf	ARPA	x	x	Cerf@ISIA
Jack Haverty	BBN	x	x	JHaverty@BBND
Tony Lake	BBN	x		ALake@BBNE
Bill Plummer	BBN	x	x	Plummer@BBNA
Ray Tomlinson	BBN	x	x	Tomlinson@BBN
Mike Wingfield	BBN	x	x	Wingfield@BBNE
Kou-Mei Chuang	CCA	x	x	Kou-Mei@CCA
David Lou	CCA	x		Lou@CCA
Ed Cain	DCEC	x	x	DCEC-R850@BBNB
Ray McFarland	DOD	x	x	McFarland@ISIA
Gary Grossman	DTI	x	x	grg@DTI
Ken Biba	Ford	x	x	Biba@SRI-KL
Danny Cohen	ISI	x	x	Cohen@ISIB
Jon Postel	ISI	x	x	Postel@ISIB
Dick Watson	LLL	x		DWatson@BBNB
Dave Clark	MIT	x	x	Clark@MIT-Multics
Karen Sollins	MIT	x		Karen%MIT-ML
Dave Reed	MIT	x		DPR@MIT-ML
Jim Mathis	SRI	x	x	Mathis@SRI-KL
Andy Poggio	SRI	x	x	Poggio@SRI-KL
John Shoch	XEROX		x	Shoch@PARC
Larry Stewart	XEROX	x	x	LStewart@PARC