IEN-66

TCP Meeting Notes
13 & 14 October 1977

The following is a reconstruction from my handwritten sketchy notes and memories of the meeting after a lapse of about a week, so please feel free to offer corrections or additions if you think these notes misrepresent the situation.

Agenda

Discussion of TCP-3
Packet Radio Demonstration
Discussion of TCP-4

Discussion of TCP-3

Vint spoke first and outlined his goal for the meeting: "Define TCP-3". With the following guidelines: "Don't make major changes unless necessary, but do fix the problems in TCP-2". TCP-3 should have hooks for growth to TCP-4 and for provision of modes for speech or other types of service (unreliable modes). DSP & TCP -- MIT will explore some of the DSP ideas, MIT will run TCP on it's major servers in LCS net but will also persue DSP ideas locally. It may work out that DSP is a subset of TCP.

Vint presented a list of issues to be discussed, divided them into TCP-3 and TCP-4 issues, and left the rest of us to hash it out.

TCP-3
- Rubber EOL
- Resynchronization
- Three Way Handshake
- Control and Data
- Zero Window
- BOL and EOL

TCP-4
- Fragmentation
- Types of Service

TCP-3 Issues

Ray Tomlinson lead the discussion by reviewing the proposal he circulated just before the meeting (see - arpanet message number <[BBN-TENEXA]12-Oct-77 11:59:41.TOMLINSON> or the Appendix). The discussion of the points below is a summary of the meetings discussion and conclusions as captured by my rough notes, individual contributions are not identified.

1. Rubber EOL

The idea that an EOL signals the consumption of the rest of the space in the buffer and that the data sequence numbers reflect that. The exchange of buffer size and sequencing still done in octets. If a receiver states a buffer size, then the EOL is rubber; if no buffer size is stated, then assume the buffer size is 1 octet (equivalent to saying the EOL is not rubber). The receiver tells the sender the size of the buffer in a SYN packet that contains a 16 bit buffer size field in the TCP header, the presence of the field being signaled by a BSZ control bit (see item "y." below). If a letter starts at sequence number x and is n octets long and the users buffer is m octets long then if n is less than or equal m the next letter starts at x+m, otherwise the next letter starts at x+im where i is a positive integer such that im > n > (i-1)n.

2. Resynchronization

Resynchronization is replaced by a time out of one packet lifetime before transmission on host restart. No one was sure what the maximum packet lifetime was (even for the regular arpanet). It is recommended that the delay on startup be 2 minutes. It was noted that this strategy will only work in a benign environment, for example spoofing can cause problems.

3. Three Way Handshake

The three way handshake is retained, (and by implication the re-use of socket numbers).

4. Control and Data

In TCP-3 the flavor of control should be of the monotonic increasing variety, e.g. use the urgent pointer rather than interrupt bit, but separate channel for control and data is not to be in TCP-3.

Urgent Pointer

As soon as an urgent pointer is in advance of the left edge the TCP should tell the user to go into "read fast" mode, when left edge catches up to urgent pointer the TCP should tell user to go into "read normal" mode. If the urgent pointer is updated while the user is in "read fast" mode it will be invisible to the user.

The facility to count interrupts at the user level that is now used in Telnet when it uses the NCP will not be available in this style of operation of the urgent pointer, but it is felt that the urgent pointer can carry all the functionality of the current known uses of the NCP interrupt and is more reliable.

5. Zero Window

Mechanisms specific to opening zero windows seem to be at least as expensive as simply retransmitting into the zero window at a low retransmission rate. So no zero window mechanism is to be in TCP-3 other than retransmission.

6. BOL and EOL

BOL (begining of letter) is useful as well as EOL (end of letter) in the case of a mode of operation where the receiver acknowledges everything to keep retransmissions at a minimum to provide a peculiar type of service. In this mode TCP provides the user with complete letters but allows letters to be lost inbetween the ones actually delivered. For this mode the TCP must be able to find the beginning of a letter as well as the end. (Actually this could be done without a special BOL since the end of one letter is the beginning of the next, but a BOL allows a slight improvement in the probability of finding whole letters.)

x. Addressing

(Vint didn't tell us to talk about this but we did anyway.) Following Ray's proposal we adopt a variable length address, with two 4 bit fields to specify the address lengths in octets then that many octets of destination and source address respectively. The first octet is to be interpreted as the network number, some felt that this would soon prove to be too small a size for this field, but we left it at one octet. There was much discussion of Ray's proposed "proc" fields, but what it came down to was making explicit the notion that the tail of the address could be used inside the host for further multiplexing.

y. Packet Format

(Vint didn't explicitly tell us to talk about this either.) Besides the changes for the addressing changes above we accepted the format Ray suggested with a few other changes, the main one being that there is a Version field to say what the shape of the Internet portion is, and a Format field to say what the shape of the rest is (examples of format values are TCP-3, TCP-4, DSP).

Internet Section (IS)

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|version|  TOS  |    format     |         total length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS hdr lgth=16| DAL=5 | SAL=3 |          destination          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                destination continued          |     source    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    source continued           |      internet options         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           internet options continued          |    padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP Specific Section

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       sequence number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    acknowledgement number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|x x x x x x U B B E A B E R S F|                               |
|x x x x x x R S O O C O O S Y I|            window             |
|x x x x x x G Z S S K L L T N N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       destination port                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          source port                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| data offset |   reserved      |           checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     urgent pointer*           |         buffer size*          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       TCP options             |    padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            data                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

DAL = Destination Address Length. SAL = Source Address Length.

Fields marked with a asterisk may be omitted if the corresponding control flag is 0 and if all control flags for following fields are 0.

Total Length is the total length of the packet in octets including Internet Section, TCP Header, and Data.

Internet Section Header Length (IS hdr lgth) is the length of the Internet Section in octets, and thus points to the beginning of the TCP Header.

Data Offset is the length of the TCP Header in octets, and thus points to the beginning of the Data.

[EDITORS NOTE: i don't have any notes on this, and have supplied what i guess is intended, please correct me if you remember something different than what i have stated.]

Note that padding fields are used to ensure that the TCP Header and the Data begin on 32 bit word boundaries.

z. What is a Letter

To clarify what we mean by a letter Ray prepared the following statement.

A letter is a sequence of one or more successive octets on a TCP connection. The beginning of a letter is marked by a BOL control flag in a packet (segment). The end of a letter is marked by the appearance of an EOL control flag in a packet. A letter is the minimum unit of information which must be passed from a TCP to the receiving program. A TCP may pass less information to the receiving program, or it may pass more.

Generally, the locations of letter boundaries are not passed to the receiving program. The exception is for non-reliable transmission. In this case, when a section of data is missing, the data which follows must either begin on a letter boundary or an indication given that the data does not begin on a letter boundary.

The difference in sequence number of the first octet of data in any letters on a given connection is always equal zero modulo the receive buffer size. That is, whenever an EOL is transmitted, the sender advances his send sequence number by an amount (in the range 0 to buffersize-1) sufficient to consume all the unused space in the receiver's buffer. The amount of space consumed in this fashion is deducted from the send window just as is the space consumed by actual data.

We also agreed that an EOL functions to force the TCP to deliver all the data before that point to the receiving user, but that a receiving user can't expect to see all (or any) letter boundaries.

Packet Radio Demonstration

We first attended a briefing on the packet radio project given by Vint and Ron Kunzelman. They described the operation of the packet radio network and the facilities in place in the SF bay area.

Then we all squoze into the SRI-INTERNATIONAL "bread truck" for a mobile demonstration of the packet radio system in action. The system worked fairly well and displays showed the action in packets transmitted etc, with the effects of going under a bridge quite apparent. About 2/3 rds of the the way through the trip something failed in the computer support in the van, demonstrating that there is still work to be done.

Discussion of TCP-4

7. Fragmentation

The question of the role of TCP in fragmentation was raised (for some apparently for the first time). The idea of a separate internet protocol that routes TCP segments through the internet system is gaining favor, and if such an internet protocol is to exist perhaps it should be the site of fragmentation and reassembly. There are questions about the impact of such a decision on formats of headers.

The idea of a fragmentation protocol (or fragmentation function in an internet protocol was discussed. The idea presented added the following two fields to the Internet Header: unique-id, and location-in-this-id. A last fragment flag would be needed too. It then seems that BOS and EOS would not be needed in TCP.

This seemed like too big a change for the present, so the current TCP fragmentation description will be in TCP-3, but it may be removed in TCP-4.

8. Types of Service

At what levels of protocol are types of service indicators needed? (Internet, TCP, Both ?) What is the flavor of a type of service indicator? ("high priority" or "type 3" or "deliver this in 1 second or kill it"?)

Since almost all the stuff in the internet header gets passed to the next level one way or another we decided that TOS can be pased on that way too, and we are sure that we want everything in the path to see and act on the basis of the TOS so we want it on the outside of the envelope.

We discussed a range of possible experinents to try to get a handle of the types of service that might be called for.

point to point speech

real time ponter - e.g. a fox and hares game to focus on human factors delay effects

conferencing - network support by multidestination addressing

synchronization of multiple tcp connections

TCP-3 will be a reliable protocol, but the should be provision in TCP-4 for unreliable modes if necessary for low delay.

TCP-3 may have a problem with multi addressing if different receivers have different buffer sizes, then the rubber EOL causes problems.

Danny and Earl explaind some things about the speech conferencing models they have. The main point is to leave the the conference control at the user level.

TOS may have to be generic at the gateway or internet level and translated to a specific thing for each transit network.

Ideas for TOS values: posion packet (self destruct after x time in the system), relibility class, delay class, "bigger than a breadbox service" (letter gets delivered only if its whole, but may not be delivered at all).

Action Items

TCP-3 Specification: draft in two weeks, final two weeks after that - Postel.

Presentation on Conference Graphics: Next TCP Meeting - Cohen

TCP-3 Implementations:

Tenex	1 Jan 77
Small Tops 20	1 Jan 77
Packet Radio Station	1 Jan 77
Multics	1 Jan 77

CCN 360/91	1 Jun 77
Large Tops 20	1 Jul 77
MIT Unix	1 ??? 77

Higher Level Protocols

I am not too precise about this but it seemed that user telnet was to be available as soon as TCP-3 came up, and that the other higher level protocols about 3 months later. The other protocols are server telnet, user and server ftp.

One other constraint is that TCP-2 is to be made available on both Tenex and Tops-20 before TCP-3 is.

Next Meeting

The next meeting will be at ISI on 30 and 31 Jan 78.

Attendees

Danny Cohen	ISI	(213)822-1511	Cohen@ISIB
Earl Craighill	SRI	(415)326-6200	Craighill@SRI-KL
Raphael Rom	SRI	(415)326-6200	Rom@SRI-KL
Wm. W. Plummer	BBN	(617)491-1850	Plummer@BBNA
Bob Braden	UCLA	(213)825-7518	Braden@CCN
John Shoch	Xerox	(415)494-4000	Shoch@PARC
Yogen Dalal	Xerox	(415)494-4000	Dalal@PARC
Dave Clark	MIT	(617)253-6003	Clark@MIT-Multics
Steve Crocker	ISI	(213)822-1511	Crocker@ISIC
Jon Postel	ISI	(213)822-1511	Postel@ISIB
Ray McFarland	DOD	(301)688-7125	DEdwards@ISIA
Dave Kaufman	SDC	(213)829-7511	Weissman@ISIA
Vint Cerf	ARPA	(202)694-5037	Cerf@ISIA
Ray Tomlinson	BBN	(617)491-1850	Tomlinson@BBNA

Appendix

Proposal for TCP 3
12 October 1977

Raymond S. Tomlinson
Bolt Beranek and Newman
Cambridge, Ma.

This note sets forth proposals for the disposition of the unresolved issues concerning TCP and reiterates those decisions which have previously been made. In most cases, the proposals presented here are a consensus of the involved parties. However, in some cases, they represent what I feel are workable solutions to the issues which have been raised for which a consensus has not or cannot be reached.

There are a number of areas of concern. These are: addressing extensions, unique sockets vs. unique sequence numbers, interrupt/urgent details, reliable delivery of control information, and type of service provisions. Previously resolved areas consist of protocol simplification through removal of ARQ, RSN, and INT, improved buffer management capabilities (buffer size option), possible checksum changes, data delimiters (EOL etc.).

ARQ (acknowledgment request) has been eliminated. This control bit was introduced in an attempt to eliminate the hangups caused by inaccurate flow control information at the sender -- namely, when the packet opening a zero width window was lost. Instead of ARQ, the suggestion that the sender avoid indefinite blocking by transmitting into a zero-width window at a lower rate was accepted thus eliminating the need for ARQ. Related to this was the decision to adopt a buffer size option which permits a sender to predict by how much the window will decrease as a consequence of sending a letter mark (EOL). This option allows a sender to do a better job of tracking the receiver's available buffers and thus improve the accuracy of the sender's send window size.

RSN was eliminated by requiring that a TCP delay opening connections following a memory destroying crash sufficiently long that all packets transmitted prior to the crash have had time to die out. INT was replaced with a variation of the DSP urgent mechanism. Details were not resolved and a proposal for their resolution appears below.

The suggestion to adopt a continuous form of checksum was rejected since it required frequent resynchronization in the case of non-reliable transmission and did not provide any great advantages. At some time, it may be desirable to adopt a more powerful checksum algorithm. The packet format given below allows space for an additional 8 bits for this purpose.

The EOL interpretation was changed so as to permit the receiving TCP to discard letter boundary information. Higher level protocols are required to provide their own mechanism for parsing the data stream and cannot depend on the EOL mechanism. EOL also has the property that it consumes all the unused space in a buffer (as specified in the buffer size option).

The need for a way to mark the beginning of significant information was demonstrated previously, but the details of this were left unspecified. I propose that a beginning of letter (BOL) control bit be defined which serves to mark the beginning of a piece of information which, to some degree, can stand by itself. In non-reliable applications such as speech, data will be delivered from the TCP to the application program in segments beginning at BOL marks and ending at EOL marks. There may be intervening EOL/BOL marks.

The DSP urgent mechanism was adopted to replace INT. This mechanism permits a point in the data stream to be designated as the end of "urgent" information. Whenever this point is beyond the left window edge at the receiving TCP, that TCP so informs the application program so that program can switch into a mode of operation intended to scan through the data up to the urgent pointer in an attempt to extract the urgent information. The exact nature of this scan depends on the higher level protocol being employed, but would typically involve discarding information.

Two methods of implementing this feature have been suggested. The first employs a bit which marks all packets up through the urgent packet. Since the urgent packet may not have existed at the time an earlier packet was transmitted, those packets may have to be retransmitted with the urgent bit set. The drawback of this method is that the receiver can't tell were the urgent information will end and must constantly go into and out of urgent mode. The second method employs a pointer which is carried in a field of all packets transmitted while the urgent pointer exceeds the left window edge. The second method seems preferable. I propose a control bit be defined (URG) which indicates that the packet contains a 16-bit field which should be added to the packet sequence number to yield the urgent pointer. The absence of this bit indicates that the urgent pointer has not changed.

It should be mentioned that the urgent pointer should invariably fall at a letter boundary. If this is not the case, the sending TCP and receiving TCP will not be obligated to deliver the data preceding the urgent pointer to the receiving program.

A great deal of discussion has taken place on the subject of unique sockets vs. unique sequence numbers. The weight of that discussion seems to be that there are problems with supporting connections between two "well-known" sockets which have not yet been satisfactorily resolved and that there is no great benefit of unique sockets over unique sequence numbers except the connection passing off capability. I propose that TCP retain unique sequence numbers with three-way handshake and that resynchronization be eliminated as indicated above. Reconnection can be provided within the TCP framework without significant difficulties. One such mechanism has been designed and appears to have no major problems. It suffers from requiring an auxiliary reliable connection in order to coordinate the passing off of the connection. An alternative design is under development which substitutes an unreliable TCP to TCP exchange to effect the passing of the connection end. This second design still needs to be analyzed for difficulties in obscure cases. If time permits, drafts of each of these designs will be available.

The committee studying the reliable delivery of control information has published a report which seems to provide the capability for reliably delivering control information. At the moment, there is no pressing need for this capability. However, the mechanism described can be added on with no major disruption of TCP.

No concrete proposals have been made about changes to the addressing structure. In the absence of such a proposal, I propose the following. The packet format should be modified to permit the extension of the addressing capabilities. The basic requirement here is provide room for more bits of address without excessively burdening packets which don't require these longer addresses. For this purpose, two address length fields should be added to the packet header which specify the length (in octets) of each of the source and destination addresses. Gateways will parse the destination address in the same way as is currently done except the field boundaries are less rigid. In particular, the host-on-net field need not be 24 bits for networks having 16-bit host addresses (e.g., Packet Radio Net). Since variable length addresses imply variable length headers, the protocol-independent header is required to be padded to a 32-bit boundary (using the padding option) in order to avoid excessive processing inefficiency.

Note that extra bits may be added at the end of each address which the gateways will not attempt to interpret. These bits may specify which protocol process within a host is to receive the packet. Examples of such processes would be the TCP process, the XNET server process, the gateway control process, or the packet echoer process. There is also the possibility of placing another whole layer of addressing hierarchy in this position.

I propose that the following other minor alterations in packet format be made. Eliminate protocol version field. Expand protocol format field and interpret solely as information for gateways to determine how to process (fragment) the packet. Add type-of-service field. Change data length field to total length field. Change header length field to protocol-independent header length. Move protocol-independent options to follow protocol-independent header. Re-order TCP dependent fields so that unused fields can be omitted.

The following is a suggested packet header format. For concreteness, the destination address is assumed to be a 40 bit address and the source address is assumed to be 32 bits.

Protocol-independent Section

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     format     |       TOS      |          total length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| NP hdr lgth=16| DAL=5 | SAL=4 |   dest net    |   dest host     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       dest host continued      |   dest proc   |  source net    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           source host          |  source proc  |    padding     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP Specific Section

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        sequence number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    acknowledgement number*                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|x x x x x x U B B E A B E R S F|                                |
|x x x x x x R S O O C O O S Y I|             window             |
|x x x x x x G Z S S K L L T N N|                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   reserved    |                    dest port                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TCP hdr lgth  |                   source port                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            checksum            |        urgent pointer*         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          buffer size*          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

DAL = Destination Address Length. SAL = Source Address Length.

Fields marked with a asterisk may be omitted if the corresponding control flag is 0 and if all control flags for following fields are 0.