INTERNETTING

BEYOND NCP

Danny Cohen
Computer Science Department
CALTECH, Pasadena, California 91125

and

USC/Information Sciences Institute
Marina del Rey, California 90291
INTERNETTING
or
BEYOND NCP

NCP in particular and the ARPANet in general have fulfilled a most important (shall we say "historic") role in networking.

Facing today's reality of many diverse networks, several communication requirements, and equipped with the experience gathered so far, one could probably develop a new model for packet communication.

The TCP paper [Cerf and Kahn, May '74] is clearly a step in the right direction. The intentions behind it are more significant than the first attempts to implement it. In this working paper I would like to pursue my understanding of the ideas behind the TCP original paper, without paying attention to "details" imposed by reality on all implementations. This working paper benefits and suffers from my total ignorance regarding TCP implementation, and the latest developments regarding it. On the other hand, it is also influenced by my misfortune of having to use the ARPANet for traffic of a kind never anticipated by its designers (real time speech).

This mismatch suggests that not all data traffic is the same.

The basic notion we would like to promote here is the concept of the CP (communication-profile), which is similar to "type-of-service" (TOS), like telegrams, first-class letters, parcels, etc.

We would like to see a communication system where the end users can specify (describe) the communication service expected, and have the networks (note: in plural) underneath do the best they can in order to provide it, as requested.

This notion of customers telling which kind of service they want for their money, and having the right to expect it, is the basis of healthy economic systems.

In this system, each piece of communication has a control portion and a data portion. The control portion is all the information about the expected (desired) type of communication. It will be used by the communication systems. The data portion is the payload, the information to be given to the remote end user. Refering to the postal service analogy, the control is what is on the outside of the envelope: address, "airmail", stamps, return-address, etc. and the data is the content of the envelope. For simplicity, and with no loss of generality, let's consider the control portion to be a HEADER which is added in front of the data portion.
The front end (or the "interface") of any network, should be able to recognize the meaning of this CP, and know which action should be taken to meet (in some optimal way) the required service. However, even when the information is travelling along the network it still maintains its full identity, both data and control portions, both the letter and its envelope. Should an internetwork transfer take place in order to reach the final destination, the same process of converting the desired service as defined by the control portion, the CP, into the services available in the consequent network should take place again. Note that within each network a LNH (Local Network Header) may be appended to the travelling data. However this LNH should not REPLACE the CP (meaning here both Communication Profile and Control Portion) but be ADDED to it, just as the mail-bag does not replace the envelopes of the individual letters within it.

This is a most important point. Local Network Headers do not replace the CPs, but are added to them, even at the cost of a possible redundancy.

We will continue with this concept of CP later, but first let us explore another point. Several important practical reasons (e.g., flow control) caused us to revert to circuit switching philosophy in the packet switching environment. NCP with its CONNECTIONS is one example. The way large messages are communicated between IMPs is another one. The "circuit switching" philosophy here is expressed in the way that CONNECTIONs are built, the ICP handshake, etc. in the case of NCP CONNECTIONs, and in the way multi-packet messages are handled in the IMP-to-IMP protocol. TCP has ASSOCIATIONs which might prove to be just like NCP's CONNECTIONs, if so implemented.

We propose to examine the CONNECTION/ASSOCIATION concept and to clear the requirements which necessitate it. Whenever these requirements do not exist a CONNECTION-free communication could take place and probably should.

Let's continue with the CP concept. Which kinds of traffic do we know? Some of them can be exemplified by TELNET, FTP, Real-Time Speech, Interactive Graphics (it will probably prove to be different than the above), Interprocessing Communication and more.

TELNET is an example of requirement for high-speed (low delay) and high reliability, FTP of high bandwidth and high reliability, and Real Time Speech of high speed and high bandwidth. Is it possible to typify all CPs by specifying them along these three dimensions: SPEED (or delay), BANDWIDTH, and RELIABILITY? Probably not, at least COST should be associated to reflect on the relative merits of these dimensions for each CP. Is SECURITY another dimension? Are there more dimensions?

A possible way to specify a CP is by specifying a value along each of these dimensions. However, this turns out not to be sufficient to cover all the possible CPs.
An alternate approach is to address specifically the issues involved. Here is an attempt to do so:

**SORTING:**
- Sort/Don't Sort the fragments of each segment.
- Sort/Don't Sort the segments of each letter.
- Sort/Don't Sort the letters.

**ACKNOWLEDGMENT:**
- The destination should/shouldn't acknowledge this segment/letter.
- The source TCP should/shouldn't forward the acknowledgment to the user.

**RETRANSMISSION:**
- Unless acknowledgment has been received within time T1, this segment/letter should/shouldn't be retransmitted. This is to be repeated N-times/for-time-T2.

**SIZE:**
- This segment/letter is of size N.

**RELIABILITY:**
- The minimum required level of reliability is R (What does it mean?).

**PRIORITY:**
- The priority level is P, or how much one is willing to add to the delivery cost in order to speed up the deliver (By how much?).

**ERROR-CONTROL:**
- Something could be said here.

**SECURITY:**
- Something could be said here, too.

**ROUTING:**
- Some information may be supplied here, but let's ignore it for the time being.

Obviously all of these issues are not independent. For example, one cannot expect a communication system to operate reasonably without the ACKNOWLEDGMENT feature enabled and with the retransmission feature enabled.
Note that all of the above issues, and others which escape our attention, can be associated with each individual piece of communication (letter or message). Other issues are associated with the CONNECTION/ASSOCIATION rather than with the individual letters.

Probably the most important of them are the flow control and rate issues.

The basic question is: should the CP be decided upon when the communication system is designed or should it be bound as per user preference at run-time. Obviously, we are biased in favor of the latter. Can it be implemented, within a single network, and across network boundaries? We believe that it can and we believe that each single network implementation could be performed as part of the more general internetworking environment. Here is our modest proposals. We find it easier to start the explanation from the middle. Each travelling piece of information (packet, fragment, etc.) should look like

   LNH-FI-CP-DATA......

The LNH is the Local Network Header.

The FI is the information unique to this fragment/packet (but not to the entire letter/message) like its size, its relative position within the complete letter/message, numbered according to the granularity of fragmentation (e.g., L/S/F for letter number, segment number, and fragment number), and the EL and ES (end-of-) bits,

The CP is obviously the Communication Profile including the full addresses (network designation included) of both the originator and the final receiver, CONNECTION/ASSOCIATION identification (we'll come back to this one), etc.

The DATA is some portion of the data to be communicated.

The FI is supposed to be very much along the lines of the TCP-headers. Having the CP be a "standard" portion of each unit of transmission, and of the procedure used to enter the data into the communication system, allows the service-type to be bound at run-time in the most dynamic way. This obviously can support internetworking provided the gateways are implemented right. We'll continue with this subject soon.

It is important to distinguish between two levels which are so far considered together as the TCP level, namely the SD (Source to Destination) level and the TC (Transmission Control) level. Sorting and flow control are at the SD level, priority and addressing are at the TC level.

Back to CONNECTION/ASSOCIATION. The function of the CONNECTION is mainly to allow an end to end flow control. There is an initial investment associated with setting it up but once done it allows a more efficient communication (probably at higher bandwidth) due to the
presence of the flow control mechanism. Setting up a CONNECTION is just like establishing a virtual circuit between two points. The main advantage of pure packet switching, PPS, compared with virtual circuit switching, VCS (which might be implemented by packet switching technology for the well known reasons), is the fast switching time (i.e., short set up time). Its disadvantage is obviously the overhead associated. Hence, for each communication system there is some size breakeven point under which PPS is better and over which VCS is better.

Back to our postal service analogy. There is no need to alert B that A is about to mail him a postcard. However, if it is a parcel, B better be home when it is delivered. What makes the postcard different? The assurance that there is room in B's mailbox for a postcard, but not necessarily for the parcel. Exactly the same in bit communication! Isn't it amazing how far the postal analogy goes?

Maybe the entire CONNECTION concept should be divorced from the user, and be introduced only when needed, in some automatic way, by the communication system (namely, the source TCP) according to the quantity of the data to be communicated (as measured by SIZE and RATE).

Back to our mainstream. Consider the following diagram describing Real Time Speech communication, using the Network Voice Protocol (NVP):

```
U-S
 . Y-S
 . . NVP-S
 . . . SD-S
 . . . . TC-S
 . . . . . GA-S
 . . . . . . A-S
 . . . . . . . A-D
 . . . . . . . . GA-D
 . . . . . . . . . TC-I
 . . . . . . . . . . GQ-S
 . . . . . . . . . . . Q-S
 . . . . . . . . . . . . Q-I
 . . . . . . . . . . . . . Q-D
 . . . . . . . . . . . . . . GQ-D
 . . . . . . . . . . . . . . . TC-I
 . . . . . . . . . . . . . . . . GP-S
 . . . . . . . . . . . . . . . . . P-S
 . . . . . . . . . . . . . . . . . . P-D
 . . . . . . . . . . . . . . . . . . . TP-D
 . . . . . . . . . . . . . . . . . . . . SD-D
 . . . . . . . . . . . . . . . . . . . . . NVP-D
 . . . . . . . . . . . . . . . . . . . . . . Y-D
 U-D
```

The Source User (Talker)

Source Vocoder

Source NVP

Source TCP (1)

Source TCP (2)

Gateway into the ARPANet

A node of the ARPANet

A node of the ARPANet

Gateway from the ARPANet

Addressing, reliability, etc.

Gateway into the QNet

A node of the QNet

A node of the QNet

A node of the QNet

Gateway from the QNet

Addressing, reliability, etc.

Gateway into the PRNet

A node of the PRNet

A node of the PRNet

Gateway from the PRNet

Destination TCP (2)

Destination TCP (1)

Destination NVP

Destination Vocoder

The Destination User (Listener)
Needless to say this diagram is not complete and several additional layers of protocol do exist. For example, between the nodes of each network there are protocols such as IMPS/IMP and MODEM/MODEM. The additional levels may also exist in other levels like ENCRYPTION which would probably be initiated between the V-S and the NVP-S, and terminated between the NVP-D and the V-D.

Note that in the above diagram, for every source process, X-S, there is a corresponding destination inverse (in the mathematical sense!) process, X-D. The intermediate processes, such as Q-I, do not have to be inversed since they do not perform any information transformation. The role of U-S and V-S are obvious. The NVP-S "messagizes" the data into network messages, and prefixes them with the required Communication Profile, the CP, which contains the addresses of the final destination and the originating source (including their network designation). The TCP-S performs the function required for controlling the network transmission, defines "letters" and "segments", etc.

The GA-S, the source gateway into the ARPANet adds an ARPANet header which reflects the desired CP, in the best possible way. The ARPANet nodes (using their own protocols and headers) move the fragments to the GA-D, the destination ARPANet gateway, which discards the ARPANet headers, and undoes everything done by the source gateway, like breaking a fragments to smaller packets, as required by the particular network, sorting these packets, etc. Then the GA-D gives the fragments to the source gateway of the next network.

At this time, the headers of the fragments are the FI and the CP (which together constitute the TCP header!). The GQ-D translates the CP into the LNH of the QNet, which is the best approximation to the requested CP.

This process is repeated for each network on the way, until finally the fragments reach the destination TCP, which delivers them to the destination user process, according to the mode specified in the CP. In the Real Time Speech application the CP is probably set to indicate a request for fragment and segment sorting, but not for inter-letter sorting.

If only one network, say the ARPANet, is used between the source and the destination, one could bypass the GA-S and the GA-D, and connect the TCP-S directly to the A-S, and the TCP-D to the A-D. We believe that it is better not to do so, but to have the gateways between the TCPs and the network nodes (IMP-s here) even if no internetcitng is currently used. Doing so allows one to implement TCPs in a more general way, leaving all the knowledge of the specifics and the idiosyncrasies of the particular network in use to the gateway, which obviously must be tailored to the network.
SUMMARY

- The TCP header should be divided into two parts, the CP and the FI. The former should contain information about the desired communication profile for the entire letter, and the latter about THIS fragment only.
- The CP should constitute a dynamic specification of the desired service.
- Even within a single network, TCP gateways should be used.
- ASSOCIATIONS should be used only when the TCP, rather than the user, decides there is a need for them, based on the quantity of the data to be communicated.

Danny Cohen.
Here are several comments regarding the above note (with trivial formatting modifications):

3 7 Feb  cas at RAND-UNIX  Comments on PRIN 213 (internetting or beyond NCP) (4222 chrs)
6 8 FEB  CERF at USC-ISI  Re: Comments on PRIN 213 (internetting or beyond NCP) (2848 chrs)
7 8 FEB  KAHN at USC-ISI  Re: Comments on PRIN 213 (internetting or beyond NCP) (1253 chrs)
10 16 Feb  POSTEL at USC-ISID  PRIN 213 & Sunshine Comments (1247 chrs)
12 17 Feb  Cas at Rand-Unix  internet protocols (1329 chrs)
Mail from RAND-UNIX rcvd at 7-FEB-77 1728-PST
Date: 7 Feb 1977 at 1712-PST
From: cas at RAND-UNIX
Subject: Comments on PRTN 213 (internetting or beyond NCP)
To: cohen at isib
cc: sunshine, cerf at isi, su-dsl at isi, tomlinson at bbn,
    postel at sri-ai, kahn at isic

Danny,

The following are some comments on PRTN 213, "Internetting or Beyond TCP."

The key idea seems to be that the user (or his application
such as NVP) should be able to request various types of
communication service. This indeed seems highly desirable, and
to the extent that TCP is designed to provide only a particular
type of service, it falls short of the generality desired. TCP
was in fact designed to provide high reliability and potentially
high bandwidth CONNECTIONS, at the cost of significant overhead,
setup, and complexity. Hence it is not ideal for all types of
traffic. In particular, it may not be very good for network speech which would benefit from less reliable delivery at less
cost.

Your idea of identifying the components of the customer
profile seems good, and you have covered the important ones, but
I strongly disagree with your idea of a connection involving
primarily flow control. I believe that a connection involves 1)
error detection, 2) error correction (usually by retransmission),
3) sequencing, and 4) flow control in "increasing" order. That is
the latter require the earlier, except that flow control is
pretty much independent. Error correction requires uniquely
identifying each message at source and destination to avoid
missing or duplicating messages. Sequencing requires the unique
identifiers to be ordered. Both of these require synchronized
control information at either end of a connection. Similarly,
flow control requires window or credit information at each side.
I believe the definition of "connection" is the requirement for
this synchronized control information to exist beyond the
boundary of an individual message or packet. The "connection"
exists when and only when the synchronized information exists.

Therefore, I believe the main reason for connections is
reliability, not efficiency (by means of flow control). If the
control information for reliability is done away with, then the
only overhead (inefficiency) of message mode is that the full
address must accompany every message. This overhead can be
reduced by short addressing if a connection is set up, but this
requires routing tables to reflect the short address-to-route
correspondence all along the route, which is a big price to pay.
The reliability difference between message and connection mode is
clearcut, however. Hence I disagree with your suggestion that
data quantity is the main determinant of whether to use
connection (VCS) or message (PPS) mode, and suggest that instead
it is reliability.
Your suggestion that the communication profile (CP) should vary from letter to letter does not make sense if connection type service is desired. The existence of the connection implies that a certain type of CP maintains (for all letters). Of course you could agree to "turn off" sequencing, for example, for a while, and then turn it back on, but the position of the unsequenced letters relative to the sequenced letters would not be defined—in a sense they would not be part of the connection.

The essence of your suggestion, that the user of transport should be able to specify various levels of service (CP) from the local nets traversed, is highly desirable I agree. Thus TCP's want to create packets that ask for minimum local net services (in most cases), while NVP might wish for partial connection services (just error detection and ordering) in a different end-to-end protocol which requests ordering from the local nets traversed. But I suspect that this CP should remain constant for all exchanges of a particular session between end-users, although it varies among different sorts of sessions. Current networks do not seem very inclined to provide various levels of service, but rather to decide what one level is good for all its users and impose that.

Carl Sunshine

-----
Mail from USC- ISI rcvd at 8-FEB-77 1309-PST
Date: 8 FEB 1977 1248-PST
From: CERF at USC-ISI
Subject: Re: Comments on PRTN 213 (internetting or beyond NCP)
To: cas at RAND-UNIX, cohen at ISIB
cc: sunshine at RAND-UNIX, su-dsl, tomlinson at BBN,
cc: postel at SRI-Al, kahn at ISIC, CERF

In response to the message sent 7 Feb 1977 at 1712-PST from cas at RAND-UNIX

Danny,

Carl Sunshine's comments are very much aligned with my own.
The purpose behind connections was to achieve high
reliability which requires retransmission, duplicate
detection, sequencing (or at least, sequence numbering to
provide unique identification), fragmentation (and
reassembly). High bandwidth was an objective and achieved
through allowing multiple packets to be in transit at once.
Flow control is also a desirable capability.

I agree that creation of a connection should not be based on
the quantity of data to be sent but rather on the importance
of reliable delivery (without duplication).

As for the various CP options, fragments have to be sorted
and reassembled into segments or you won't be able to
compute the checksum, maybe you don't want a checksum??
Sorting the segments into a letter mayor may not be
important, users are not supposed to have control over
segment size, only letter size. Getting segments out of
order could be troublesome if the user cannot tell what to
make of the data in the segment - he cannot even parse it if
the segment breaks occur at random places in the letter.
Sending short letters might be a better strategy.

I think acknowledgements are essential for reliability, but
I can well imagine a user who doesn't want to hear about the
ACK because he doesn't care about the reliability.

I don't know what a TCP would do with the reliability
parameter you propose. Priority would more likely be used
to improve the probability that a packet would not be
dropped rather than improve its delay characteristics.
Priority probably matters most when the net is congested and
some traffic is going to be turned away or even thrown away.
So far, our philosophy has been to build internet raw
packets with the minimum information needed to get the
packets across the network boundary. If CP information is
largely addressing and mode information, I think a slightly
better packet format is LNH-CP-FI-DATA. FI might correspond
to TCP while CP might correspond to internet addressing with
some facilities information as well (i.e. mode of packet
transmission).

So, to summarize, the basic function of the connection is to
supply reliability; flow control is a feature for
convenience. (may turn out to be essential to maintaining controlled network operation as well.

Looking forward to the discussions on this and other topics.

3/12/77.

Vint
------
Mail from USC-ISI rcvd at 8-FEB-77 1456-PST
Date:  8 FEB 1977 1445-PST
From: KAHN at USC-ISI
Subject: Re: Comments on PRTN 213 (internetting or beyond NCP)
To:    CERF, cas at RAND-UNIX, cohen at ISIB
cc:    sunshine at RAND-UNIX, su-dsl, tomlinson at BBN,
cc:    postel at SRI-AI, kahn at ISIC, KAHN

In response to the message sent 8 FEB 1977 1249-PST from CERF at USC-ISI

VINT AND CARL,

This is probably not the forum to delve deeply into what TCP is or is not or what is was intended for or not for. However, after seeing the flurry of msgs about what connections are for I would like to point out that the concept of associations was introduced to avoid the unnecessary connotations usually attached to the word connection. To my way of thinking the TCP is equally at home at handling conventional connections as it is at potentially handling more fragile associations. The early implementations have emphasized the standard connection notion (and wisely so given the context) but little time has been spent on developing the other attributes of a tcp. We are actively thinking about broadcast TCP implementations which are extensions of the initial implementations but are TCP's to me nevertheless.

Regards, Bob

--------
Mail from USC-ISID rcvd at 16-FEB-77 2240-PST
Date: 16 Feb 1977 2232-PST
From: POSTEL at USC-ISID
Subject: PRTRN 213 & Sunshine Comments
To: Cohen at ISIB, sunshine at ISI
cc: postel, crocker at ISIB

Danny & Carl:

I am very interested in the issue that is discussed in your notes. The project I am joining at ISI is chartered (as I understand it) to investigate the protocol needs for interning. Danny brings up a point of concern to me: how to build internet interfaces that don't restrict the options on the kinds of service otherwise obtainable. Danny points out several possibly independent measures of a kind of service, e.g. delay, throughput, reliability. TCP makes one choice for the user: moderate delay, moderate throughput, high reliability. There are applications for which this choice is a disaster: is there a way to interface networks such that the kind of service is still an option? That is one of the questions I hope to investigate in the coming months. Some of Carl's points also will be subject of investigation. What are the protocol features or activities necessary in an internet interface and which are there for convenience. (Perhaps the convenience of some and the vexation of others.)

-- jon.

-------
In response to Jon's comments (16 Feb 1977), I agree that the TCP makes a certain choice of service at the end-to-end level, and that other types of service are more desirable for some applications. In my paper on network interconnection (in Computer Networks Journal, #3) I tried to make the distinction between end-to-end service levels and local net service levels. They should indeed be independently specifiable, since different E-E services are best accomplished by using different local net service levels. Even the same E-E service may be realizable by using different combinations of end controls and intermediate net services (e.g. a virtual circuit E-E service may be realized by datagrams on each local net and a fancy E-E protocol, or by virtual circuit services on each local net plugged into each other with nothing added at the ends.)

I look forward to being able to discuss all these things further in the near future.

Carl