

INWG NOTE #28

RE: INWG PLENARY MEETING June 7 and 8

We have rooms Chambord 14 and 15 set aside in the Americana Hotel (across the street from the New York Hilton) for our exclusive use during the 7th and 8th of June. We will meet at 1000 a.m. June 7th in Chambord 14 to consolidate our agenda, and get a progress report out of the way. The reports on interconnection and host-to-host protocol are in preparation, and I hope to have them in your hands before you leave for the meeting. I will bring copies in case someone doesn't get a copy in time.

Housing arrangements have been left up to you, as I was unsure of everyone's plans for attending NCC.

See you there!



Vint Cerf
Assistant Professor of
Computer Science and
Electrical Engineering

VC:c

415-321-3300x365

CURRENT STATE OF THE INWG

1. Membership

The following countries and organizations are represented (See mailing list for details of addresses, etc. Proposed liaison agent has been marked with "(L)".)

- a. Austria
 - IBM Laboratory
 - H. Zemanek (L)
- b. Brazil
 - University of Rio de Janiero
 - C. Lucena (L)
- c. Canada
 - Bell Northern Research
 - W. Clipsham
 - O. Riml (L)
 - A. Curran (chairman of IFIP TC-6)
 - Communication Research Centre
 - C. D. Shephard (L)
 - University of British Columbia
 - V. Detwiler (L)
 - University of Waterloo
 - D. Cowan (L)
- d. Finland
 - Helsinki University of Technology
 - K. Rahko (L)
- e. France
 - CII
 - M. Elie (L)
 - CNET (French PTT) - CCETT
 - R. Despres (L)
 - Electricite de France
 - P. Martin (L)
 - IRIA
 - L. Pouzin (L)
- f. Germany
 - Siemens, A.G.
 - G. Bergmann (L)
 - Consultant
 - G. Färber (L)
- g. Israel
 - Ministry of Communications
 - J. Kella (L)
- h. Italy
 - Italcable
 - G. Puccioni (L)
 - Olivetti
 - M. Ballabeni (L)

i. Japan

Mitsubishi Research Institute
T. Utsumi (L)
Musashino Electrical Communication Labs.
M. Kato (L)
Nippon Telephone and Telegraph
F. Watanabe (L)
Tohoku University
J. Oizumi (L)

j. Norway

Norwegian Defense Research Establishment
Y. Lundh (L)

k. Sweden

Stockholm University, Royal Institute of Tech.
K. Samuelson (L)

l. United Kingdom

Government Post Office
R. Belton (L)
B. Sexton
K. Sandum
COST-11 Project
D. L. A. Barber (L)
Logica, Inc.
J. Tucker (L)
National Physical Laboratory
D. Davies (L)
R. Scantlebury
University of London
P. Kirstein (L)

m. United States

Advanced Research Projects Agency
R. Kahn (L)
L. Roberts
S. Crocker
Bolt, Beranek and Newman, Inc.
W. Barker
J. Burchfiel (L)
F. Heart
A. McKenzie (L)
D. Walden
MERIT
E. Aupperle (L)
Packet Communications, Inc.
R. Alter (L)
Stanford Research Institute/NIC
J. North (L)
Stanford University
V. Cerf (L)
University of California, Los Angeles
L. Kleinrock
J. Postel (L)

- m. United States (continued)
 - University of Hawaii
 - F. Kuo (L)
 - University of Southern California/ISI
 - K. Uncapher (L)
 - University of Utah
 - B. Wessler (L)
 - XEROX, Palo Alto Research Center
 - R. Metcalfe (L)

2. IFIP Affiliation

- a. INWG is now officially working group 6.1 under IFIP Technical Committee 6 (Data communications). TC6 is chaired by Mr. Alex Curran of Bell-Northern Research, Canada.
- b. In addition to lending to INWG considerably more official stature as an accredited working body of a well known international organization, IFIP can also be instrumental in providing INWG with recognized access to ISO, CCITT, and other standards making bodies which INWG should seek to influence.
- c. Our charter (see attachment to agenda) also permits us to host seminars, workshops, and symposia, set up and perform experiments, and generally proceed in a rapid fashion towards the fulfillment of our objectives.

3. Possible Experiments

Thus far, INWG has worked on paper to explore the issues of interconnecting networks. Many questions related to protocols and interfacing standards may only be resolved by experiment. To this end, the following possibilities are offered:

- 1. The University of London will be connected to the ARPA network by a TIP in mid-July. A 370/195 will be connected to the TIP via a PDP-9 which will act as gateway.
- 2. The British Post Office expects to have an operational model of the EPSS system available for experimentation by late 1974. Considerable attention has been given by the National Physical Laboratory and the British Post Office to the performance of a internetwork experiment.

3. Possible Experiments (continued)

3. Both the CYCLADES project at IRIA and the RCP network at CNET are rapidly approaching functional status. By mid-summer, both networks should be running with multiple nodes. One hopes that sufficient stability within each net will be reached to consider an interconnection experiment by the end of 1973.

7 June 1973

SECOND INWG PLENARY SESSION
AMERICANA HOTEL, CHAMBORD ROOMS 14 and 15.

PROPOSED AGENDA

Thursday, June 7

- 1000 CURRENT STATE OF THE INWG - V. Cerf
1. Membership
 2. IFIP Affiliation
 3. Experiments
- 1030 ADMINISTRATIVE ISSUES
1. Liaison mailing list
 2. INWG Meeting Schedule
 - a. Semi-annual Plenary Sessions
 - b. Locations?
 - c. Next IFIP General Assembly is October 18, Toronto
- 1100 TECHNICAL AGENDA
- Elements of a HOST/HOST Protocol and implied functions of the Gateway - V. Cerf
- 1200 - 1400 LUNCH BREAK
- 1400 An Interconnection Strategy Based on the ARPA Very-Distant-Host Protocol - R. Kahn
- 1500 Split into small informal groups (HOST-HOST groups in Chambord 14, Subnet interconnection groups in Chambord 15).

Friday, June 8

- 1000 Continue informal working group discussions
- 1200 - 1400 LUNCH BREAK
- 1400 Progress Report to INWG
1. Issues raised and/or solved
 2. Hot topics
- 1500 Official Adjournment

Peter K will call me
~6:30

Proposed Terms of Reference

W.G. 6-1 - International Network Working Group

1. Name

The group shall be called Working Group 6-1 - International Network Working Group, abbreviated INWG.

2. Sponsorship

INWG 1 will be sponsored by and will exist as a working group of technical Committee 6 (Data Communications) of the International Federation for Information Processing.

3. Membership

INWG activities will be co-ordinated by a chairman appointed by TC6. The chairman will provide regular reports of INWG activities to the Chairman of TC6.

The initial list of members will be approved by TC6. The chairman of INWG may appoint additional members who have demonstrated their ability and willingness to contribute to the work programme of INWG. The Chairman of INWG will obtain approval from the Chairman of TC6 for members added under this clause.

4. Purpose

INWG will study the problems associated with the interworking of computer networks planned or under construction in various countries. The ultimate goal is to define the technical characteristics of facilities and operating procedures which will make it possible and attractive to interconnect computer networks.

In pursuit of this goal INWG will, from time to time, define and publish guidelines for the interconnection of participating networks. Where necessary and possible INWG should test the guidelines on experimental interconnections between those networks on which meaningful experiments can be done.

5. Organization

Working Subgroups of a temporary nature may be created by the Chairman of INWG to study specific problems related to INWG's ultimate goal.

INWG members may act on more than one subgroup. Two suggested subgroups are:

- a) Communications System Requirements
- b) Host-Host Protocol Requirements

6. Meetings

The meetings of INWG will be scheduled and publicized through the IFIP Bulletin. The Chairman of TC6 will be advised of the calendar year's schedule and location of meetings, and of any budget requests by 15 September of the preceding year.

7. Communications

The Chairman of INWG will distribute to the INWG membership all reports, announcements and other material which he deems to be relevant.

8. Changes to Charter

The Charter may be amended by majority consensus of INWG membership with the approval of the current TC6 Chairman.

9. Dissolution

INWG will be dissolved by decision of the Chairman of TC6.

10. Workshops and Symposia

INWG, through TC6, may sponsor any workshops and symposia on computer communications.

March 1973

SUBNET ISSUES

1. Physical interconnection standards?
Modems, Satellites, ground lines, signalling method,
minimum bandwidth, synchronous or asynchronous, etc.
2. Minimum required packet length
3. Current minimum packet/maximum packet lengths in existing nets?
ARPA 8000 bits
EPSS 2000 bits (255 bytes)
IATA 2000 bits (255 bytes)
CYCLADES ?
NPL ?
CANUNET ?
CNET ?
...
4. How should routing to the gateway be accomplished?
As a practical matter, this may be different in
each network, but the gateway must understand
how to direct messages to other gateways if this is
necessary.
5. How many gateways should interconnect pairs of networks?
6. What kind of status messages might a subnet want to pass back
to source of a message?
e.g. destination dead, message lost, gateway
unavailable, invalid address, etc.
7. Should the subnet do any accounting to reflect service to
traffic originating outside the net?

GATEWAY Issues

1. Is a gateway a host or is it a state of mind of two packet
switches residing in different networks?
2. If we had a network of gateways, who would own and operate them?
Could each network own its own gateways?
3. Should messages be sequenced across gateways? Reassembled?
4. Can gateways avoid having to keep information about open connections?
5. What kind of accounting might a gateway be expected to do?

A SELECTION OF HOST PROTOCOL ISSUES

1. Message Switched Protocol versus Line Switched Protocol
2. Flow Control - Window Notion
Should Flow Control be exercised between NCP's (Network Control Programs) or should it control flow per connection?
3. Retransmission and Duplicate Detection Strategy
Is sequencing of messages/packets essential?
4. Positive Host-Host Acknowledgements
5. Negative Acknowledgement versus Timeout
6. Message Format
Key problem is breaking up of packets as they cross the gateway. If they are not reassembled, how can we preserve the format of the delivered packets?
7. Addressing
Should it be hierarchical?
Should there be a short form for established connections?
What to do about multiple NCP's in the same Host?
Process/Ports model of interprocess communication.
8. Minimum required packet length (Header+Chunk)
9. Error feedback from destination/source NCP and subnet(?)
e.g. message received on unopened connection,
illegal window size or chunk size definition,
invalid control message received, etc.
10. Control message semantics and syntax
11. Connection set-up and release
Full duplex, Simplex, Half-duplex?
12. Control message connection
13. Provision of status reporting and performance statistics collection
from NCP's.
14. Accounting

25
This is a draft by Cert, with Walden's comments,
of the protocol designed at the INWG meeting
in NYC on June 7, 8 1973. It was never
published

Walden's comments

A Partial Specification of an International Transmission Protocol

TO: E. Aupperle, R. Kahn, A. McKenzie, R. Metcalfe,
R. Scantlebury, D. Walden, and H. Zimmerman

FROM: V. Cerf

SUBJECT: First pass draft of International Transmission Protocol

Please feel free to make comments, revisions, additions, and deletions from the corpus. I have tried to faithfully record my understanding of the protocol. From time to time, I have departed from the proposed terminology, but I believe I have done this only where it seems appropriate.

Your early response is critical. The next meeting is most likely to be scheduled for September 16 in Sussex, just after the NATO Advanced Institute, and this protocol will be a major topic for discussion.

Send replies to me at:

ERL 407
Stanford University
Stanford, California 94305
U. S. A.

Respectfully,

Vint Cerf

Introduction

Let us begin with the assumption that we want to interconnect several, distinct, ~~resource sharing~~ computer networks. * Each of these networks connects together HOST computers whose resources can be shared among the users of the network. If we are to achieve a similar ability between HOSTS residing in different networks, we must find a way for a HOST in one network to reproduce, without alteration, a stream of messages originating from a HOST in another network.

This is a primitive but essential necessity, and the mechanism we devise to do this will be called the International Transmission Protocol. ^(ITP) This HOST level protocol will be implemented through a HOST program called the Transmission Control Program (TCP).

Consider a typical computer communication system. It is generally made up of nodes (packet switches, line switches, concentrators, multiplexors), and some communication medium connecting the nodes (telephone lines, wave guides, laser beams, radio or microwaves, satellite transponder, etc.).

HOST computers are connected to the nodes in some fashion, and messages from one HOST to another are passed through the network.

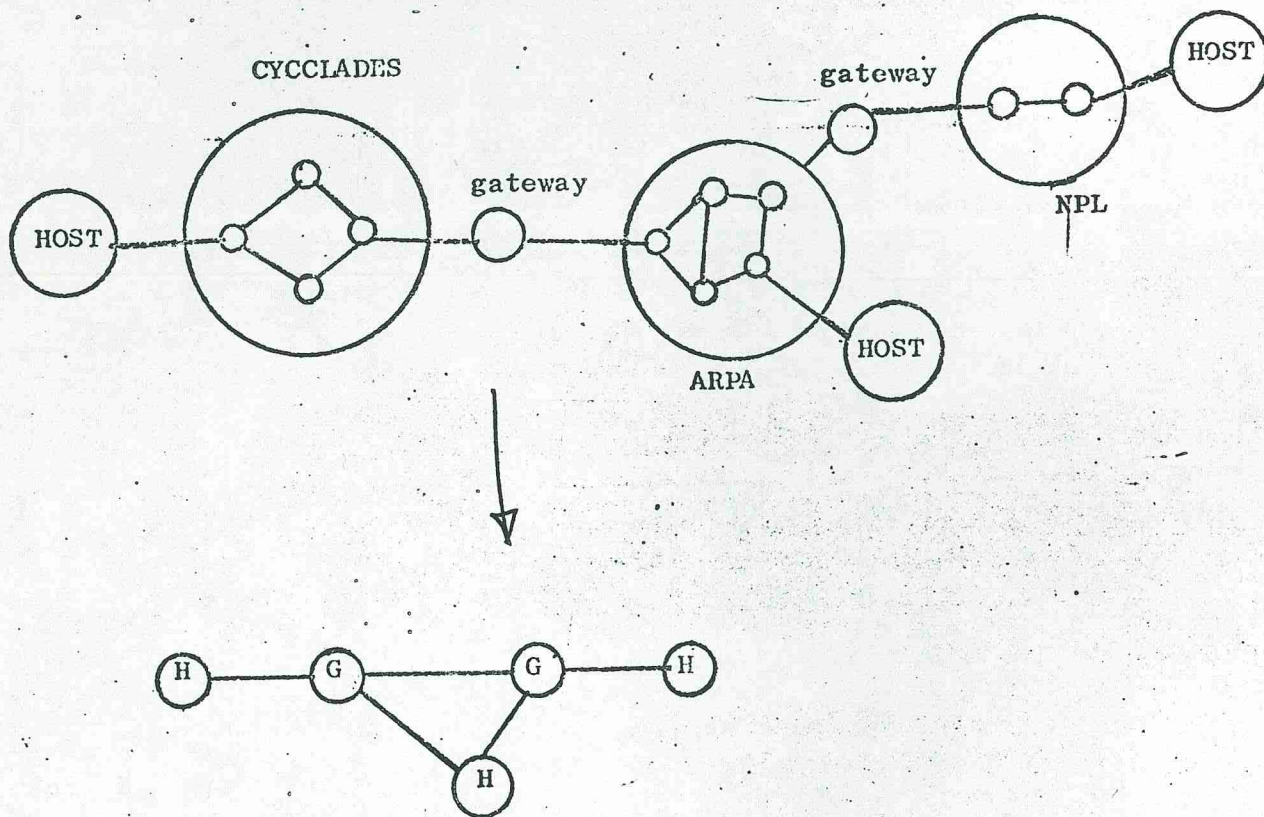
The interconnection of two networks can be achieved through an interface or gateway which is connected to the nodes of two or more distinct networks (conceptually, it might be enough if two nodes were connected directly, but had software making each node regard the other as a HOST on its own network).

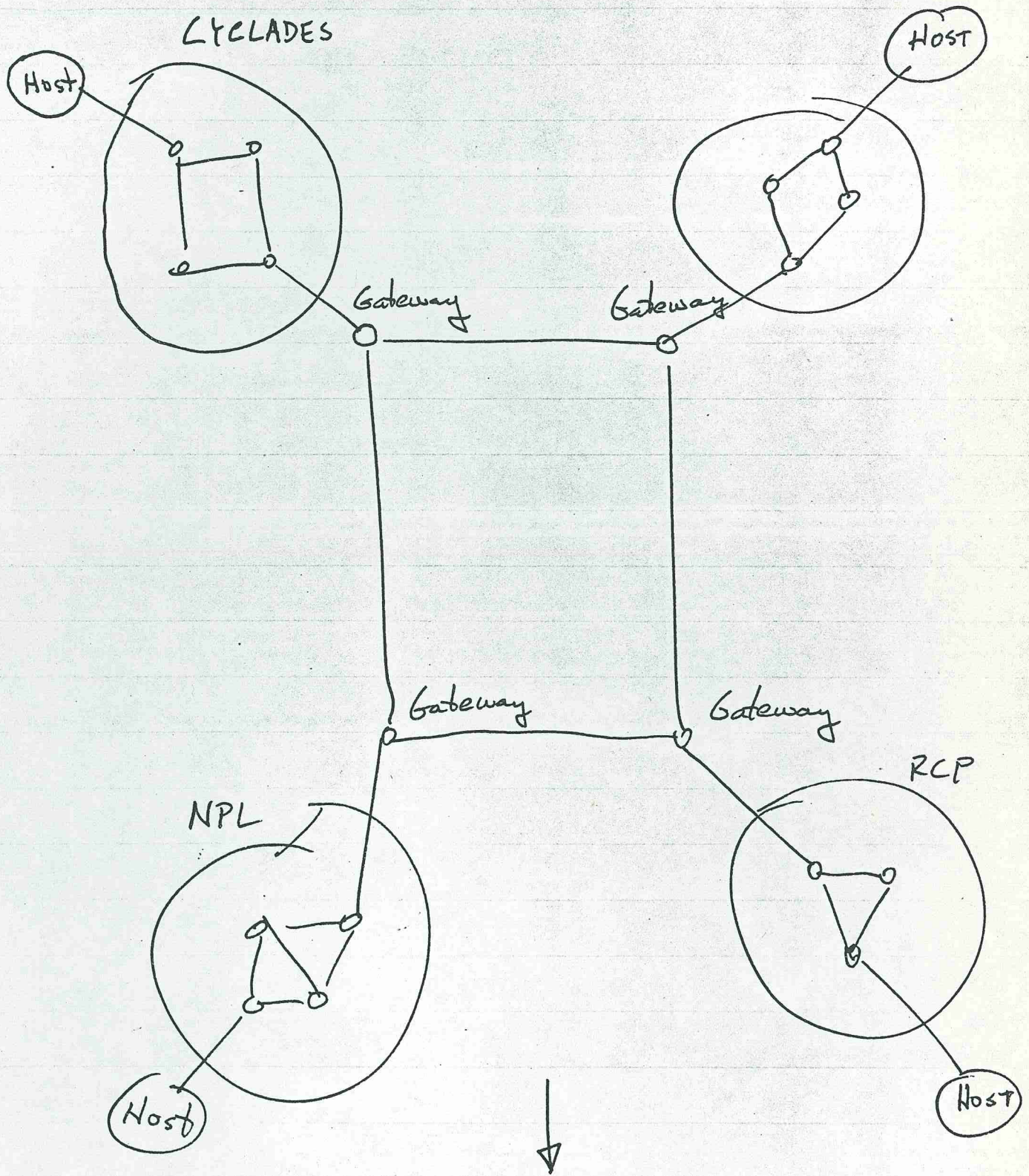
The gateway must accept messages from one network and pass them to another, reformatting the message as necessary for transmission in the new network.

* We don't require these to be "resource-sharing" networks as the PTT networks will probably only provide communications.

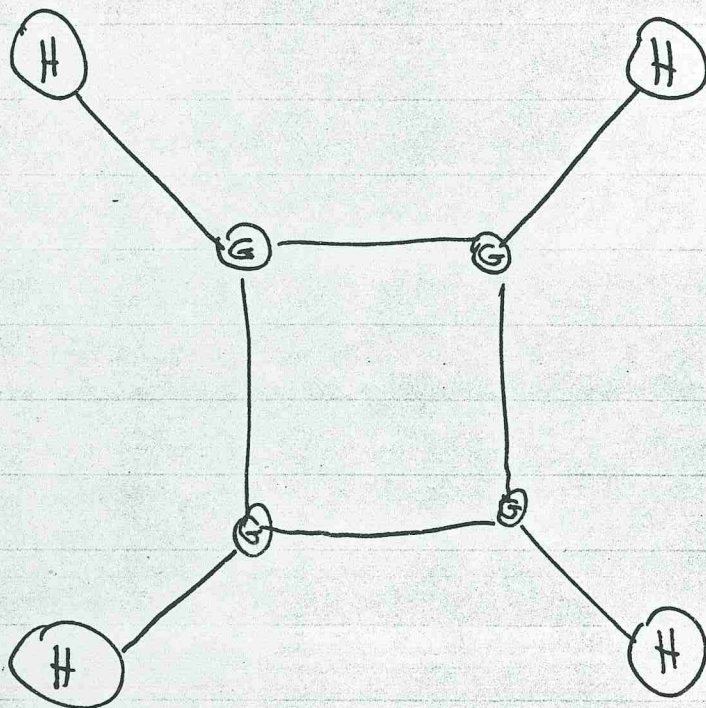
2

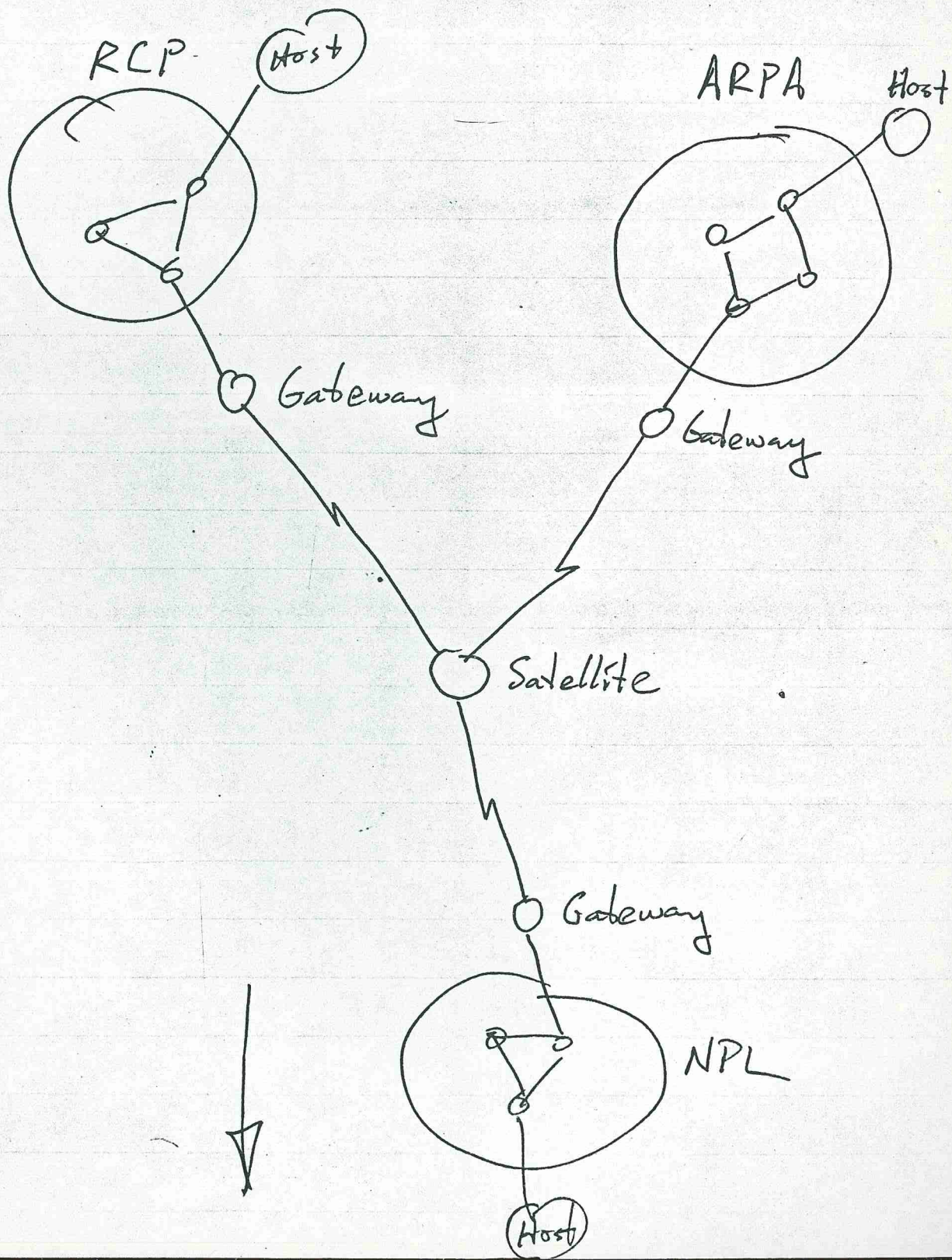
If we treat the computer communication system of each network as a complicated communication line joining HOSTs and Gateway, then the Gateways appear to be international nodes, joined by two or more networks, facilitating communication between HOSTs. *Some examples follows:*



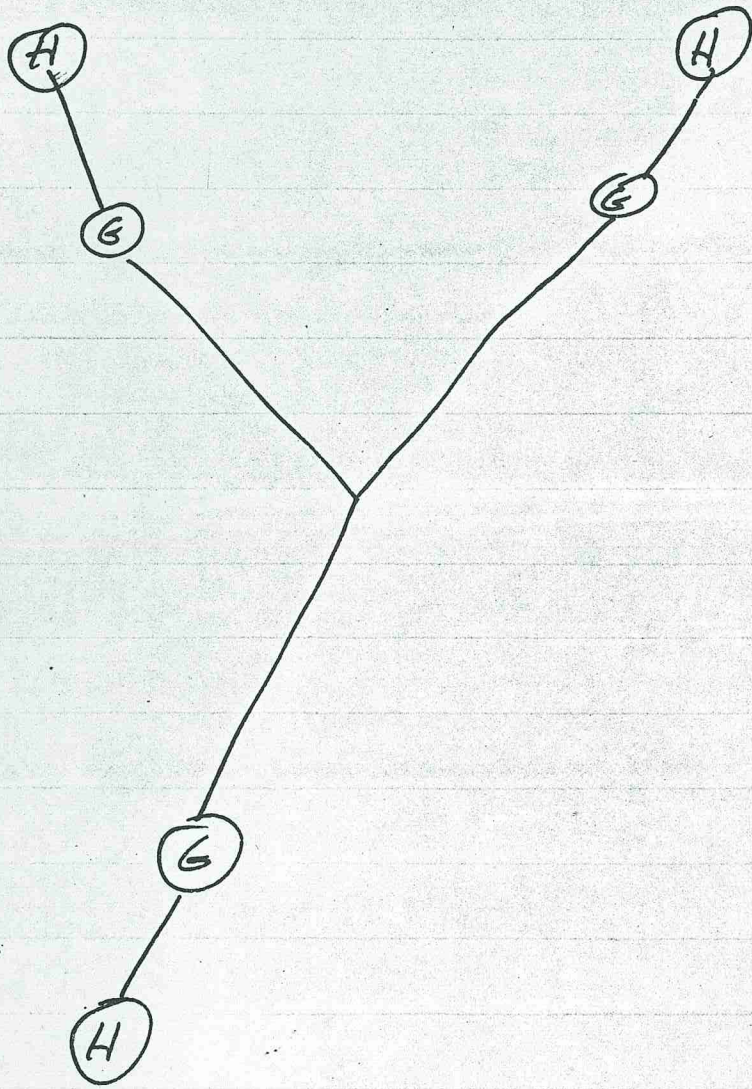


4





6



Combinations of the ~~the~~ above examples also will be possible.

9)

The international transmission protocol (ITP) described herein is intended to:

1. Be resistant to failures in intermediate constituent networks and gateways.
2. Be unaffected by the maximum message sizes permitted in constituent networks and by intra- and inter-network timing idiosyncracies.
3. Be capable of exactly reconstituting a message stream originating in a foreign HOST.
4. Provide for high bandwidth and/or low delay transmission.
5. Require ^{minimal} ~~no~~ status information in gateways (i.e., no per connexion status info)
6. Be relatively easy to implement.
7. Permit the receiving HOST to control the flow from the sending HOST.

Key Issues and Gratuitous Comments

A fundamental assumption behind the ITP is that message sizes are not uniform throughout all networks which are likely to be connected. It is tempting to try to pick a message size which all networks can transmit without alteration (e. g. 255 bytes), but our guess is that such an attempt is unlikely to succeed.

The inherent variability in message size accounts for much of the rigidity of the ITP. In another, as yet unpublished, position paper, R. Kahn proposes a scheme which appears more flexible. Kahn's design is based on a Simple Message Switched Protocol (reference Walden and Bressler), and solves the variable message size problem with a different message identification scheme than the one proposed for ITP. It is still not clear which of the two strategies, ITP or SMSP, is simplest.

It must be emphasized that the ITP described here is not a Host/Host protocol in the ARPA Network sense; a number of Host/Host protocols may be built upon the ITP. Rather, the ITP is more analogous to the ARPA Network IMP/Host protocol, but between ~~virtual~~ virtual IMPs and Hosts; the ITP is the protocol whereby destination addresses are specified, destination to source acknowledgements are sent, etc. The gateways and the Host/TCPs can be considered the virtual IMPs; the constituent networks or actual lines connecting the gateways and Hosts can be considered virtual lines.

↑

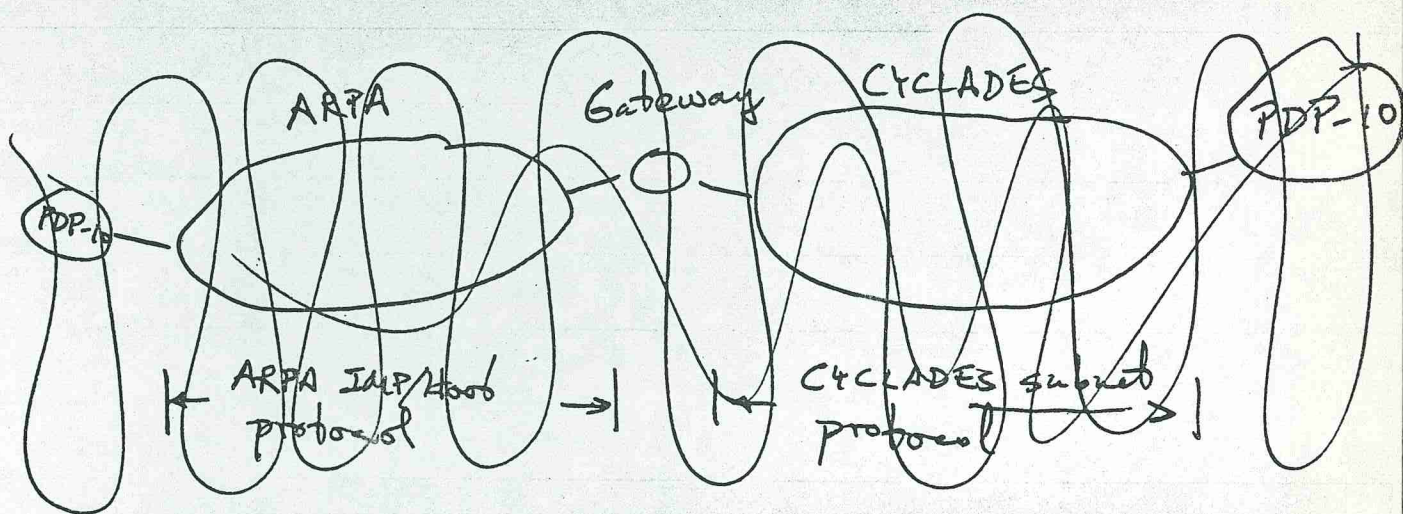
(The above can undoubtedly be said better, but I ~~do~~ do think such reemphasis is important)

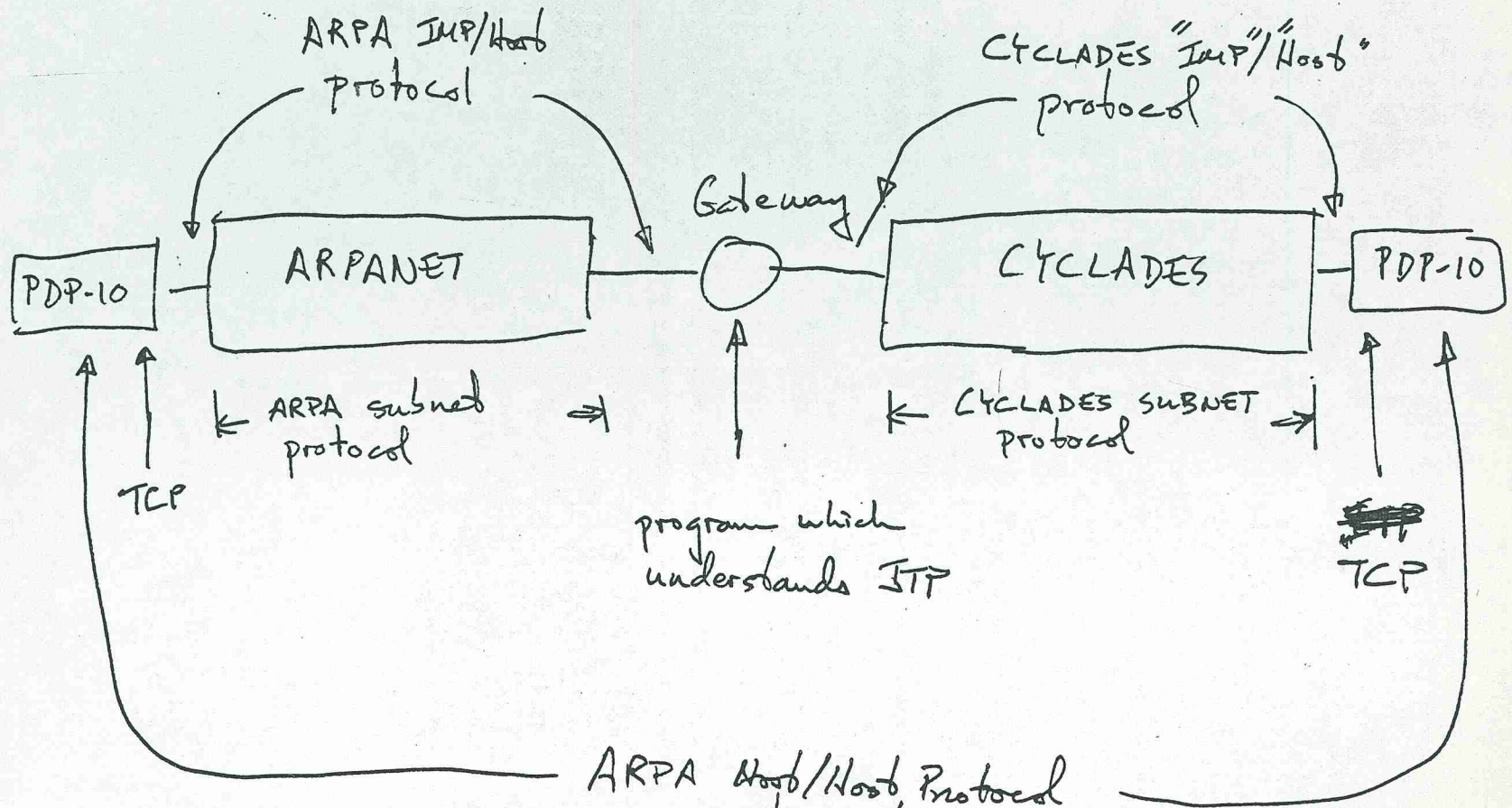
It must further be emphasized that the ITP is expected to be common between all Hosts in all ~~networks~~ constituent networks for the purpose of internetwork Host communication.

~~It~~ In many of the constituent networks, intranetwork Host communication will undoubtedly be carried out by a parallel mechanism (e.g., the ARPA network Host/Host protocol), although it is to be hoped that the intranetwork protocols eventually wither away in favor of the ~~internetwork~~ internetwork ITP.

Interestingly, once the ITP is "up" ~~in~~ on Hosts in two constituent networks, the intranetwork Host/Host protocol of either network might immediately be used as the internetwork Host/Host protocol between machines for which implementations of the protocol exist. For example,

assume the existence of a PDP-10 system in the CYCLADES network and in the ARPA network. Further, assume ~~the ITP is implemented for a PDP-10 and for a gateway~~ between CYCLADES and ARPA. The ARPA Host/Host protocol could then be put up on both PDP-10s. Perhaps the following figure will help.





ARPA Host/Host Protocol
on an interim basis;
International Host/Host
Protocols eventually..

QC.

The preceding example was not meant to promote the ARPANET Host/Host protocol as the international Host/Host protocol; it was meant only to indicate the type of interim shortcuts that may be available and to demonstrate the distinction between the ITP and Host/Host protocols.

2. TERMS OF REFERENCE

Connexion

It is imagined that between pairs of HOST computers, a single logical connexion is established upon which is multiplexed all communications between these two HOSTs. A connexion is a full-duplex channel, but it will be convenient to think of it as a pair of simplex, one-directional channels, each with a source and a destination.

Element

An element is a fixed length string of bits out of which ~~messages (see below) of a communication~~ between HOSTs are built.

The size of an element is 8 bits by default unless it is changed explicitly ^{by agreement of} the sender ^{and receiver.}

Gateway

A gateway is the interface between networks. It may, in fact, not be a real entity (e.g., computer), but may merely be realized in software which lies on either side of a physical (optical, electromagnetic, acoustic, etc.) connection between the ~~packet switches~~ (or ^{constituent} nodes) of two distinct networks.

HOST

A HOST is typically a computer which is attached to one or more ^{nodes} (packet switches) ~~(or nodes)~~ in a constituent network. The HOST is a source and destination for messages. It may be the case that the HOST

11

is coexistent with a packet switch (e.g. the TIP in the ARPA network), or that it is in fact a terminal (e.g. RCP). In the sequel, it will be necessary to assume that a terminal cannot be a HOST unless it is capable of performing according to the specified protocol, and furthermore, the name (identification) of the terminal must be known to the packet switching subnet in the same way a serving HOST's identification is known. Note that a HOST may be connected to more than one packet switch (in different networks or the same network).

Message

A message is a well defined, formatted string of bits which has the property that it can be inserted into a constituent network at one packet switch where it will leave the network in the same form it entered. If the message is broken up into packets by the network this is invisible and the packets are reassembled before they leave the network. In the ARPA network, a message is at most about 8095 bits long. In the NPL network, this length is more like 257 bytes, and in RCP it is 64 bytes.

Thus, a message is the largest bit string which can be entered into the constituent network and retrieved from it without apparent change, breakage, etc.

For some networks, messages and packets (see below) are identical, except possibly for some control information which is tacked onto the message

while it is in transit through the network. *Message, in general, will have to be split into smaller messages as they traverse the constituent networks from source to destination host.*

Node

A node is the elemental unit of a packet switching system. The term node and packet switch are used interchangeably in this paper, but a node might also be construed to be an addressable entity which participates

in the movement of packets within the network. This permits us to consider loop systems in which a node forwards but does not store packets which pass through it, on line switched systems.

Packet Switch

A packet switch is a computer which stores and forwards packets to achieve communication between HOSTS in a packet switching network. For purposes of this discussion, a gateway may be looked upon as a HOST by a packet switch.

leave one of these out - it's confusing

Transmission Control Program (TCP)

A program in a HOST computer which realizes the International Transmission protocol enabling a collection of messages from processes and terminals associated with one HOST to be multiplexed along a connexion to another HOST in the same or foreign network. The TCP does not address itself to the problem of interprocess or terminal/process or terminal/terminal communication. The job of the TCP is merely to take a stream of messages produced by one HOST and reproduce the stream at a foreign receiving HOST without change.

Of course a version of this program runs in the gateways, or at least a program which understands the format of strings of elements in the ~~ITP~~ ITP.

3. The International Transmission Protocol

Message Format

Messages are made up of a fixed length header, containing control and addressing information, an integral number of fixed length elements containing the text of the message, and space for an optional checksum (see figure 2).

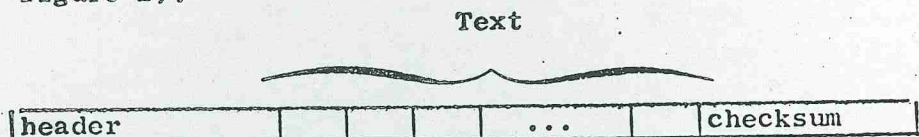


Figure 2

Message Format

We will take up the issue of variable element size later, but for the present we will assume a default size of 8 bits/element.

The ITP is resistant to failures in intermediate nodes and gateways. This is accomplished by arranging for positive acknowledgement of receipt of message by the receiving HOST. Furthermore, sending HOSTs are expected to time-out and retransmit messages which have not been acknowledged.

Any time-out and retransmission scheme requires that the receiver have the ability to detect duplicate transmissions. *of data and acknowledgements;* The ITP is no exception. Each message header carries a unique (short term) identifier which will be duplicated if the message is retransmitted.

The form that the unique identifier must take is determined in part by the fact that messages crossing through a gateway may require breaking into more than one message which may not be reassembled before delivery to a HOST. How can we uniquely identify the pieces of the broken message?

provide for the unique labeling of

One solution to this problem is to ~~uniquely label~~ each element of the message, ^{and to require} ~~tacitly assuming~~ that no gateway will break up a message into pieces smaller than one element, nor will messages be broken other than at element boundaries. (see (B))

(the point here is that each element does not in fact include id bits)

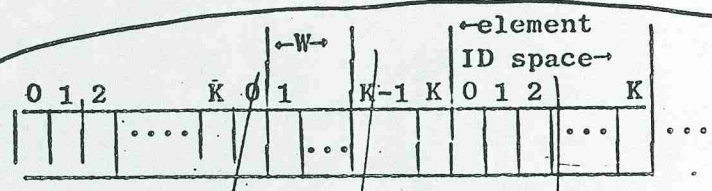


Figure 3

Bit stream, parsed into frames of $k+1$ elements.

W is the open window size.

In figure 3, we see a bit stream which has been parsed into a uniform, unbounded sequence of elements. Each element has an associated

identifier which ranges from 0 to K . If we define frame size to be $K+1$ bits, then the ID of any given element is just its left bit ordinal

modulo the frame size. *The frame size $K+1$ will be a constant based on header field sizes.*

In order to permit a receiver to control the total traffic from a sender, the sender is constrained to have no more than W bits outstanding

(unacknowledged) at one time. Initially, $W = 2000$ * *adjusted at the request of* ~~$W = \lfloor K/2 \rfloor$~~ , but can be ~~reduced~~ *must not be allowed to*

by the receiver. Note that W ~~cannot~~ exceed $\lfloor K/2 \rfloor$ for reasons which will be apparent later. (see (A))

The following discussion may clarify things

* let's start with a finite limit and then up it if necessary.

(B)

Zimmerman argues that some minimum message size will be necessary (e.g., header, checksum, + element). Therefore, he ~~says~~ says, why not just call that (or something slightly larger if possible) the virtual message size a deal in messages instead of elements and disallow gateway fragmentation. Choose a size, say 512 bits including header, etc. and do all acknowledgements, etc. on these up to 512 bit chunks. If there is some network which can't handle a message this big, the network ~~must~~ must divide the message and correctly reassemble it on the other side, just as ~~it~~ would be necessary for the ~~or~~ network which couldn't handle to minimum size "element message". The major advantage of this approach states Zimmerman is that it decouples the gateways from the Host/Host stuff almost entirely, thus simplifying change and simplifying the ~~gate~~

(B2)

146

February

I'll ignore this for the rest of
my comments, but I do think
Zimmerman's argument is fairly compelling.

Problem area 3

Notice that in the international network there may be thousands of hosts (the TCPs) or even ten of thousands or more. One doesn't want to keep sequence numbers for all the hosts one isn't talking to; therefore, the following strategy is adopted. A TCP* which goes down must stay down for greater than ^{twice} the max message transit time, t , discussed in the previous pages, that is, for $2t$.

When a TCP wants to talk to another TCP and there is presently no connection, it sends a message with left edge ϕ . When the receiver gets a message with left edge ϕ it sets its left edge for received messages to ϕ or $\phi + \text{the received message length}$.

* Acknowledge ~~no~~ or send ~~no~~ network traffic.

(A)

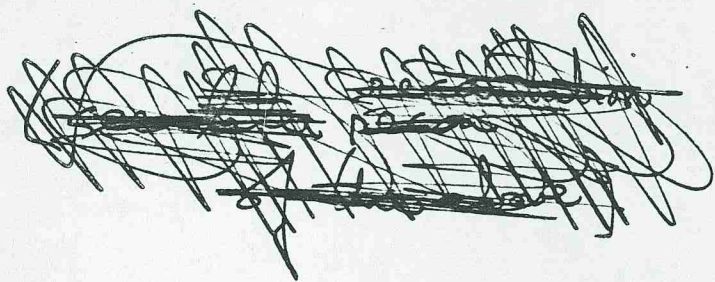
The $\lfloor K/2 \rfloor$ number seems almost irrelevant to me, for the following reasons:

1. The sequence number is used to maintain sequence.
2. The sequence number is used ~~to~~ as an identifier (for acknowledgements, etc.)
3. The sequence number is used for duplicate detection.

While the $\lfloor K/2 \rfloor$ consideration is valid for the first two, the third use of the sequence number requires such a big number that no Host will be able to think about a window size on the order of $\lfloor K/2 \rfloor$. That is, for purposes of duplicate detection one needs a sequence number space equal to the product of the number of elements (bits in the worst case) per second ^{of bandwidth} you want the network to be able to handle ~~to say~~ between a given ~~Host~~ pair of Hosts (say 500,000) and the total amount of time ~~can~~ duplicate

element can be wondering around the networks before finally popping out. The latter is probably 256 networks times 5 to 30 minutes per network minus something to account for the fact that probably not all 256 networks will delay the element in succession.

In summary, the emphasis should be on picking K . If you then want to state the $\lfloor K/2 \rfloor$ rule, OK!



A message can be made up of several sequential elements, and it is sufficient to identify the leftmost of these, while giving a count of the total number of bits in the message. Thus, the header format begins to look like Figure 5.

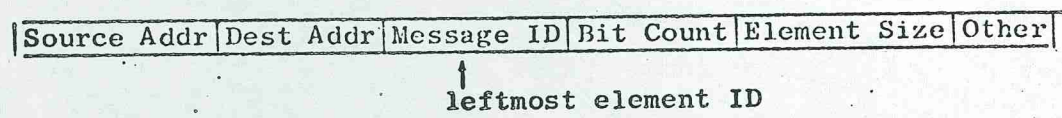
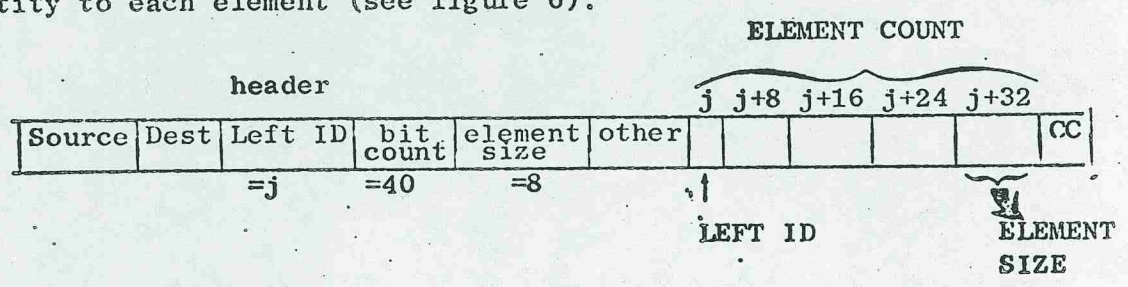
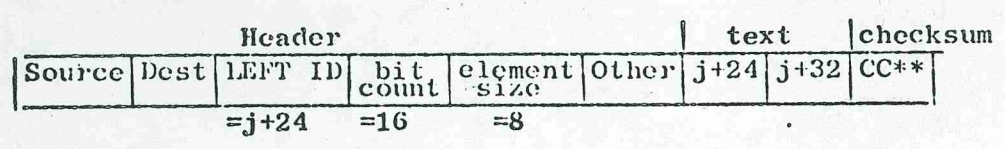
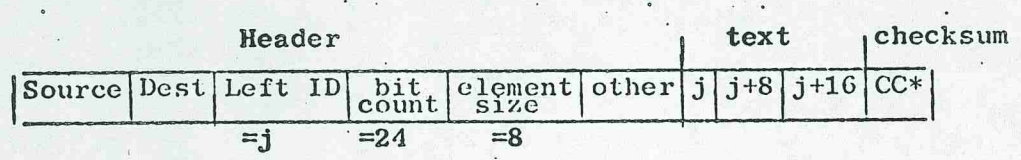


Figure 5
Header Format

If a message arrives at a gateway which finds that the message must be broken into pieces for transmission, each piece will have the same format. Since the text of each piece is constrained to consist of an integral number of elements, we can easily assign the correct unique identity to each element (see figure 6).



a. Original Message



b. Two new shorter messages

Figure 6
Message Breakage

Clearly, it is necessary that all networks have the capability of sending a message consisting of the header, checksum, and one element without breakage.

Note that the element size (in bits) is carried in the header to facilitate gateway message reformatting.

Retransmission, positive acknowledgment and duplicate detection

Earlier, we alluded to a window size, W , which represented the maximum number of bits the sender was allowed to have outstanding (unacknowledged) by the receiver. Initially both sender and receiver have the left edge of the window positioned on bit 0 and the right edge on bit $W-1$. ($W \leq \lfloor K/2 \rfloor$).

The receiver uses the window to detect duplicate transmissions. When a message arrives, its left element ID is compared with the left edge of the window. If these match, the window is advanced to the right as many bits as the bit count indicates. As an acknowledgment the receiver returns to the sender the ^{new} current left edge of the window. If the incoming message ID does not coincide with the present left edge of the window, the receiver determines whether the elements received lie inside or outside the window. If inside, the receiver can either ignore the elements, or mark them received, but not acknowledged. No acknowledgment will be sent for these elements until the missing elements to the left have all been successfully received. However, if the receiver sends as an acknowledgment the ^(unchanged) current left window edge, this may serve to stimulate retransmission of the missing elements. If the elements lie outside ("below") the window, then they are discarded, and an acknowledgment bearing the current left window edge ID is sent to the sender. Elements

can't arrive "above" the window because K is too large to allow that.

I think this is untrue - if a network chooses to pass an IP message between gateways (bit at a time), that's ok, it must just deliver the bits in order. In fact, the gateway must hold that much which means the id element size must be limited)

[Signature]

17

In fact, these acknowledgments are sent "piggyback" with other messages traveling to the source using a new field in the header labeled ACK (see figures 6 and 7). If no messages are awaiting transmission, then an empty message is sent. *Empty messages are, of course, not acknowledged themselves*

Source	Destination	Left Element ID	Bit Count	Element size	Left ID ACK	Other
--------	-------------	-----------------	-----------	--------------	-------------	-------

HEADER FORMAT

Figure 7

The sender transmits a group of elements as a message and starts a time-out. If the sender times out before an acknowledgment for the elements is received, the sender retransmits the elements. In the simplest implementation, ~~out of order acknowledgments are ignored (ultimately causing retransmission)~~. In the more complex implementation, they are accepted and the associated elements marked as successfully acknowledged. The sender does not advance the window to the right unless it receives acknowledgments for elements lying consecutively to the right of the left window edge.

One can easily be convinced that the sender can never advance the left edge of his window beyond the left edge of the receiver's window (flow control), and barring total disaster, all elements are eventually acknowledged, even if they require retransmission. Furthermore, the receiver has no trouble distinguishing between duplicate and original elements *or acknowledgements*.

There are several important points to notice. First, the receiver can actually advance his window until its left edge is W bits ahead of the sender (all elements received, but all acknowledgments lost). Since

When a duplicate ack is received (ie not necessarily non-decreasing) it is discarded (they are discarded)

13 14

the sender will ultimately retransmit to prompt the receiver again for acknowledgments, it is important that the receiver not confuse a re-transmitted element for an original one. If $W > \lfloor K/2 \rfloor$, where $K+1$ is the frame size, then such confusion can occur, hence the restriction $W \leq \lfloor K/2 \rfloor$.

The second point to notice is that a range of implementations is possible, ranging from purely sequential receipt and acknowledgment to a more complex implementation allowing non-sequential receipt and acknowledgment.

See Alex's remark

The third point is that negative acknowledgments are accomplished implicitly ~~if the receiver acknowledges only the left window edge on receipt of elements within but not contiguous with the left window edge.~~ If the sender keeps pointers to current left edge, last sent, and right window edge elements, ^{Since} then each acknowledge which fails to advance the sender's left window edge can be treated as a negative acknowledgment.

Flow Control and Element Size.

Initially, the communication channel between TCP's is assumed to consist of 8 bit elements with a window of $W = \frac{2000}{\lfloor K/2 \rfloor}$ bits where $K+1$ is the frame size.

The receiver may want to change the window size (as may the sender), and either may want to change the element size.

Such changes must be done in a controlled manner, so we postulate the existence of control messages sent from one TCP to another. The message format, since it embeds control messages within the data stream, must signal the presence of a control message by a flag in the header. Figure 9 shows the message header format.

19

Source	Destination	Left Elem ID	bit count	Elem size	Ack	Type	Text	CC
--------	-------------	-----------------	--------------	--------------	-----	------	------	----

Message Header

Figure 9

For the moment, only three types of messages are proposed.

- a. Type 0 = regular message for a processor terminal
- b. Type 1 = control message
- c. Type 2 = status message

Suppose the receiver wants to tell the sender about a new window size W' . The receiver sends a control message "DO SETWINDOWSIZE W' ." The sender will not respond until the window can be either reduced or expanded on the right to size W' . When this occurs, the sender responds with "WILL SETWINDOWSIZE W' ." Notice that the window should not be expanded beyond $W = \lfloor K/2 \rfloor$ nor should it be reduced (right edge moved leftward) over previously acknowledged elements. *What about when the sender wants to ask the receiver to do this?*

Element Size

In an early design, it was assumed that all messages in the stream from one TCP to another would consist of messages with the same element size. This size could be changed by negotiation between TCP's. Once we realize that control messages must be embedded in the stream, it is made easier to imagine that every message carries information about element size, and is guaranteed to consist of an integral number of elements, but each message may have a different element size. This permits the multiplexing of messages between several processes with different length elements, but still gives a good flow control mechanism to the receiver. Since message ID's are at the bit level, out-of-order

message can still, with some work, be inserted in the correct place in the circular buffer whose length is W bits.

Since each message can have a different element size, there is no need to negotiate to change it.

(I very much more like the idea of fixed max length messages)

Checksum

The checksum is optionally computed over the header and message text. If not computed, this field is all 1's, otherwise, it contains a folded checksum. If the originating TCP fails to put a checksum in (i.e. field is all 1's), then at some point at or before the first gateway, this checksum should be computed and put in its proper place.

If a message is broken up, the new pieces should have checksums generated by the gateway, *after the checksum on the unbroken message is first checked.*

Messages which fail to pass the checksum test should be discarded (and negative acknowledge produced -- e.g. current and unmoved left edge of window).

Routing

Initially, the gateway destination should be selected by the TCP from a table of networks versus gateway addresses. The constituent network nodes may choose to re-route messages which are destined for highly congested areas.

(see (C))

Addressing

We take the position that only sufficient addressing need be provided to get a message to a particular TCP. The address field must

(C)

With fixed routing, a node can reroute only if it ~~knows~~ knows that ~~the~~ it won't cause a loop, it seems to me. I'm sure much study has been done of this and perhaps we should read something to learn how to do it. ~~One thing that there follows are~~ Perhaps there should be a bit indicating "send-it-on-an-alternative-path" in the header.

Also, the routing section should state that in the long term we expect to implement dynamic routing between gateways.

be broken into a subfield for network, and one for TCP identifier.
(see figure 10). Although it is not mandatory, most large networks
may prefer a hierarchical addressing structure

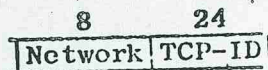


Figure 10
Source or Destination Address

in which the TCP-ID is broken into sections. This is purely up to
each network.

*(The ARPA network will be forced to the table
lookup addressing rather than bit-coded
addressing soon).*

PROBLEM AREAS

1. Setting time-outs for retransmission

2. Header field sizes:

a. element size - 8 bits

b. left ID - 16 bits

c. bit count - 16 bits

d. address - 32 bits

e. checksum - 16 bits

f. ACK - 16 bits

g. TYPE - 8 bits

h. various flags - 16 bits

← I'd guess 32 or more

3. Keeping status table sizes down (Walden's Link Table idea). *(see page 27)*

4. Note: I went to bit addressing instead of element addressing

because it became clear that I want to multiplex messages of many different
sizes along the same TCP-TCP connexion.

5. Status information obtainable by TCP from distant TCP?
(machine word size, window size, TCP performance data...).
6. Error messages?
7. How to propagate internal network status to HOSTs via ITP?
(e.g. HOST DEAD MSG NOT DELIVERED).
8. Gateway Accounting?
- 9.
- 10.

Problem area 1

I suggest an ~~initial~~ initial retransmission timeout of 1 minute with the understanding we'll adjust it ~~as~~ as necessary early in the implementation effort.

Problem area 2

~~Suggest~~ Propose a layout of the format for the header.

Returning to an old problem:

I now (this second) think that we must guarantee that messages make it from the source to the destination ~~and~~ "quickly" -- say ⁱⁿ under 2 minutes -- in the worst case. Such a ~~not~~ ^{relevant} guarantee would ~~make the network relevant again~~ ^{let} ~~the~~ the ack and sequence number fields be reasonable size; and most important, doesn't allow duplicates to come drifting in hours later which "complexifies" everything. ~~For~~ Since the source to destination transit time will usually (one hopes) be ~~much~~ much smaller than the guaranteed max (average ^{string} number of seconds), it is ~~impractical~~ ^{impractical} to try to make each of the n networks guarantee their max transit time is $\leq \text{overall max}/n$. This seems to ~~now~~ point to some kind of time stamping ~~throw~~ on ~~packets~~ messages. Most hosts will be able to keep time

somewhat accurately and could stamp
 messages going into the system with the
 time in seconds (or the hosts nearest
 approximation) since the beginning of the
 Greenwich ~~day~~^{year} mod ~~a couple of~~
~~hours~~. something on the order of a day.*
 Thus receivers could discard messages
 more than a few minutes old
 without processing them in anyway.
 The point is that for duplicate
 detection, we really want to measure
 time, ~~not~~ not number of messages ~~and~~ and
 K is a very poor thing to measure
 time with. Thus, I let's measure
 time with time and messages with
 message numbers without the
 need to maintain per connexion
~~numbers~~^{numbers} big enough for both, but
 rather a single time per
 TCP plus a small(er) sequence number
 per connexion which is $1 \text{ act} \cdot \text{time}$.

* or whatever is the time covered by K .

Successive messages from the sender to the receiver now are given appropriate left edge "counts". If ever greater than $2(t+\Delta)$ time passes without the receiver receiving a message from the sender, the receiver flushes its sequence number for the sender. If ever $t+\Delta$ time passes without the sender sending a message ~~to~~ the receiver, the sender flushes its sequence number; and the next time the sender has something to transmit to the receiver, it resets its sequence number to ϕ and ~~sender~~ sends that which causes the receiver to also reset to ϕ . If the receiver can't handle a new connection from the sender (its already loaded up with other connections), it merely doesn't acknowledge the original message from the sender with sequence number ϕ . Of course, there could also (probably should be)

a CAN'T HANDLE YOU command response which the receiver could send the sender when the receiver ~~can't~~ can't handle ~~to~~ the new connexion.

The two halves of the connexion are independent. That is, packets with only acknowledges (no data of either control or other type), do not require a sequence number and must not be given one ~~any other way~~ -- i.e., if the length field is zero, only the ACK field has meaning.

The ~~bounds~~ time bounds in the above can probably be tuned better.

Problem areas

4. OK -- but think about Zimmerman's comment.
5. Can't status be in a higher level "status protocol"? so the ITP is not too complicated?
6. Let's not have any error ~~ses~~ messages if possible in the ITP. The TCP_s should be written to assume transmission problems and recover from them -- there is no need to tell to the other TCPs. Of course, instrumentation for study purposes is always in order but should again be in another protocol.

V. Let's use safe

7. The first gateway which detects a destination is unavailable, should ~~copy this~~ do something like reversing the source and destination fields, flushing the data, and setting the DESTINATION DEAD bit. Similarly for other conditions ~~errors~~; e.g. ~~gateway discards message because it's too old~~ NO-PATH ~~to~~ TO-THERE.

8. Ugh? Let's not do this at this level!

9. You should mention that networks with long term storage should not put international traffic into that storage.

10. You should probably say somewhere that we think a very practical, pragmatic approach to ITP was the small group's consensus -- not necessarily optimum or fancy -- but doable soon.

A Partial Specification of an International Transmission Protocol

TO: E. Aupperle, R. Kahn, A. McKenzie, R. Metcalfe,
R. Scantlebury, D. Walden, and H. Zimmerman

FROM: V. Cerf

SUBJECT: First pass draft of International Transmission Protocol

Please feel free to make comments, revisions, additions, and deletions from the corpus. I have tried to faithfully record my understanding of the protocol. From time to time, I have departed from the proposed terminology, but I believe I have done this only where it seems appropriate.

Your early response is critical. The next meeting is most likely to be scheduled for September 16 in Sussex, just after the NATO Advanced Institute, and this protocol will be a major topic for discussion.

Send replies to me at:

ERL 407
Stanford University
Stanford, California 94305
U. S. A.

Respectfully,

Vint Cerf

- Vint, I basically agree with all of Walden's comments sent to you on ~~the~~ June 26. Enclosed are my own, perhaps orthogonal, comments.

Regards,
alex

TERMS OF REFERENCE

Connexion

It is imagined that between pairs of HOST computers, a single logical connexion is established upon which is multiplexed all communications between these two HOSTs. A connexion is a full-duplex channel, but it will be convenient to think of it as a pair of simplex, one-directional channels, each with a source and a destination.

Element

An element is a fixed length string of bits out of which messages (see below) of a communication between HOSTs are built. The size of an element is 8 bits by default unless it is changed explicitly by the sender.

Gateway

A gateway is the interface between networks. It may, in fact, not be a real entity (e.g., computer), but may merely be realized in software which lies on either side of a physical (optical, electromagnetic, acoustic, etc.) connection between the packet switches (or nodes) of two distinct networks.

HOST

A HOST is typically a computer which is attached to one or more packet switches (or nodes) in a constituent network. The HOST is a source and destination for messages. It may be the case that the HOST

is coexistent with a packet switch (e.g. the TIP in the ARPA network), or that it is in fact a terminal (e.g. RCP). In the sequel, it will be necessary to assume that a terminal cannot be a HOST unless it is capable of performing according to the specified protocol, and furthermore, the name (identification) of the terminal must be known to the packet switching subnet in the same way a serving HOST's identification is known. Note that a HOST may be connected to more than one packet switch (in different networks or the same network).

Message

A message is a well defined, formatted string of bits which has the property that it can be inserted into a constituent network at one packet switch where it will leave the network in the same form it entered. If the message is broken up into packets by the network this is invisible and the packets are reassembled before they leave the network. In the ARPA network, a message is at most about 8095 bits long. In the NPL network, this length is more like 257 bytes, and in RCP it is 64 bytes. Thus, a message is the largest bit string which can be entered into the network and retrieved from it without apparent change; breakage, etc. For some networks, messages and packets (see below) are identical, except possibly for some control information which is tacked onto the message while it is in transit through the net.

Node

A node is the elemental unit of a packet switching system. The term node and packet switch are used interchangeably in this paper, but a node might also be construed to be an addressable entity which participates

in the movement of packets within the network. This permits us to consider loop systems in which a node forwards but does not store packets which pass through it, on line switched systems.

Packet Switch

A packet switch is a computer which stores and forwards packets to achieve communication between HOSTS in a packet switching network. For purposes of this discussion, a gateway may be looked upon as a HOST by a packet switch.

Transmission Control Program (TCP)

A program in a HOST computer which realizes the International Transmission protocol enabling a collection of messages from processes and terminals associated with one HOST to be multiplexed along a connexion to another HOST in the same or foreign network. The TCP does not address itself to the problem of interprocess or terminal/process or terminal/terminal communication. The job of the TCP is merely to take a stream of messages produced by one HOST and reproduce the stream at a foreign receiving HOST without change.

Introduction

Let us begin with the assumption that we want to interconnect several distinct, resource-sharing computer networks. Each of these networks connects together HOST computers whose resources can be shared among the users of the network. If we are to achieve a similar ability between HOSTS residing in different networks, we must find a way for a HOST in one network to reproduce, without alteration, a stream of messages originating from a HOST in another network.

This is a primitive but essential necessity, and the mechanism we devise to do this will be called the International Transmission Protocol. This HOST level protocol will be implemented through a HOST program called the Transmission Control Program (TCP).

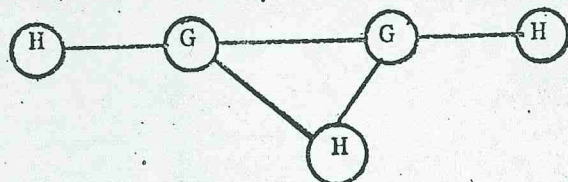
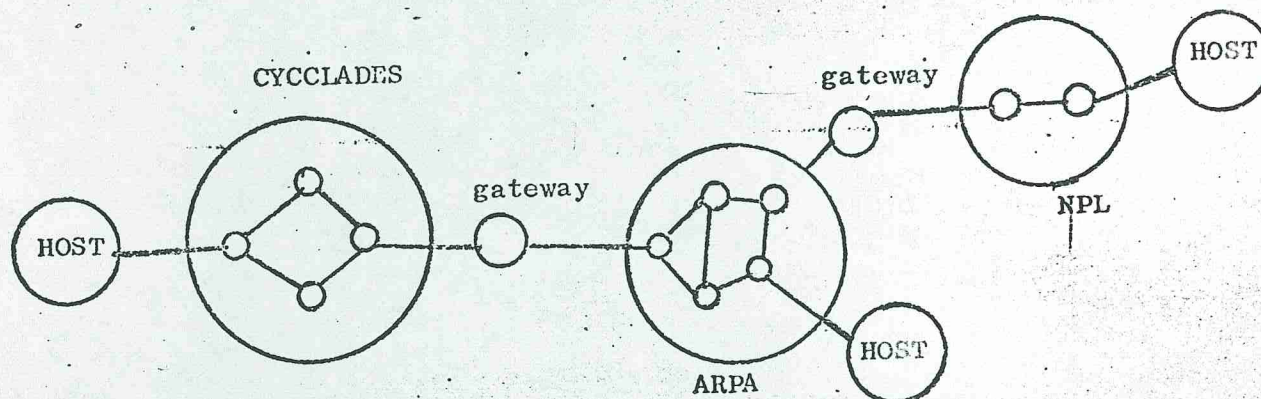
Consider a typical computer communication system. It is generally made up of nodes (packet switches, line switches, concentrators, multiplexors), and some communication medium connecting the nodes (telephone lines, wave guides, laser beams, radio or microwaves, satellite transponder, etc.).

HOST computers are connected to the nodes in some fashion, and messages from one HOST to another are passed through the network.

The interconnection of two networks can be achieved through an interface or gateway which is connected to the nodes of two or more distinct networks (conceptually, it might be enough if two nodes were connected directly, but had software making each node regard the other as a HOST on its own network).

The gateway must accept messages from one network and pass them to another, reformatting the message as necessary for transmission in the new network.

If we treat the computer communication system of each network as a complicated communication line joining HOSTs and Gateway, then the Gateways appear to be international nodes, joined by two or more networks, facilitating communication between HOSTs.



The international transmission protocol (ITP) described herein is intended to:

1. Be resistant to failures in intermediate constituent networks and gateways.
2. Be unaffected by the maximum message sizes permitted in constituent networks and by intra- and inter-network timing

idiosyncracies within some limits; ~~the~~ however, the protocol is not designed to ~~tolerate~~ work well with inter-network delivery delays of hours or days.

3. Be capable of exactly reconstituting a message stream originating in a foreign HOST.
4. Provide for high bandwidth and/or low delay transmission.
5. Require no status information in gateways.
6. Be relatively easy to implement.
7. Permit the receiving HOST to control the flow from the sending HOST.

Key Issues and Gratuitous Comments

A fundamental assumption behind the ITP is that message sizes are not uniform throughout all networks which are likely to be connected. It is tempting to try to pick a message size which all networks can transmit without alteration (e. g. 255 bytes), but our guess is that such an attempt is unlikely to succeed.

The inherent variability in message size accounts for much of the rigidity of the ITP. In another, as yet unpublished, position paper, R. Kahn proposes a scheme which appears more flexible. Kahn's design is based on a Simple Message Switched Protocol (reference Walden and Bressler), and solves the variable message size problem with a different message identification scheme than the one proposed for ITP. It is still not clear which of the two strategies, ITP or SMSP, is simplest.

↓
It seems to me that this "information" is not only gratuitous but also useless! Why not have the committee discuss Kahn's proposal only after it appears.

The International Transmission Protocol

Message Format

Messages are made up of a fixed length header, containing control and addressing information, an integral number of fixed length elements containing the text of the message, and space for an optional checksum (see figure 2).

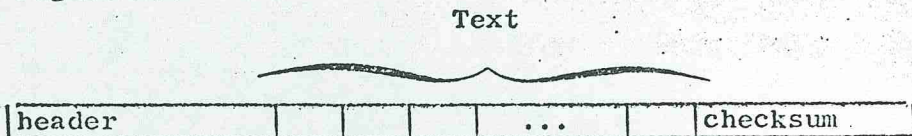


Figure 2

Message Format

We will take up the issue of variable element size later, but for the present we will assume a default size of 8 bits/element.

The ITP is resistant to failures in intermediate nodes and gateways. This is accomplished by arranging for positive acknowledgment of receipt of message by the receiving HOST. Furthermore, sending HOSTs are expected to time-out and retransmit messages which have not been acknowledged.

Any time-out and retransmission scheme requires that the receiver have the ability to detect duplicate transmissions. The ITP is no exception. Each message header carries a unique ^{in the} (short term) identifier which will be duplicated if the message is retransmitted.

The form that the unique identifier must take is determined in part by the fact that messages crossing through a gateway may require breaking into more than one message which may not be reassembled before delivery to a HOST. How can we uniquely identify the pieces of the broken message?

A message can be made up of several sequential elements, and it is sufficient to identify the leftmost of these, while giving a count of the total number of bits in the message. Thus, the header format begins to look like Figure 5.

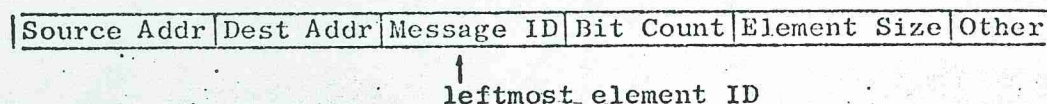


Figure 5

Header Format

If a message arrives at a gateway which finds that the message must be broken into pieces for transmission, each piece will have the same format. Since the text of each piece is constrained to consist of an integral number of elements, we can easily assign the correct unique identity to each element (see figure 6).

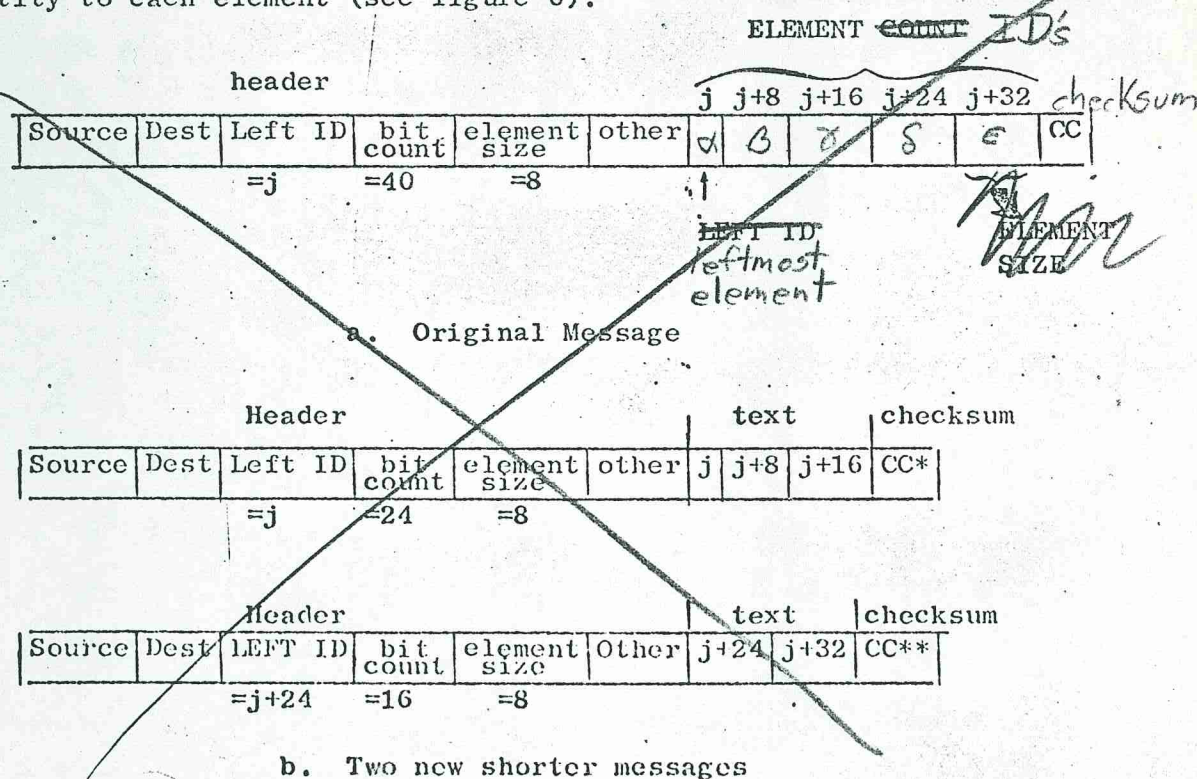
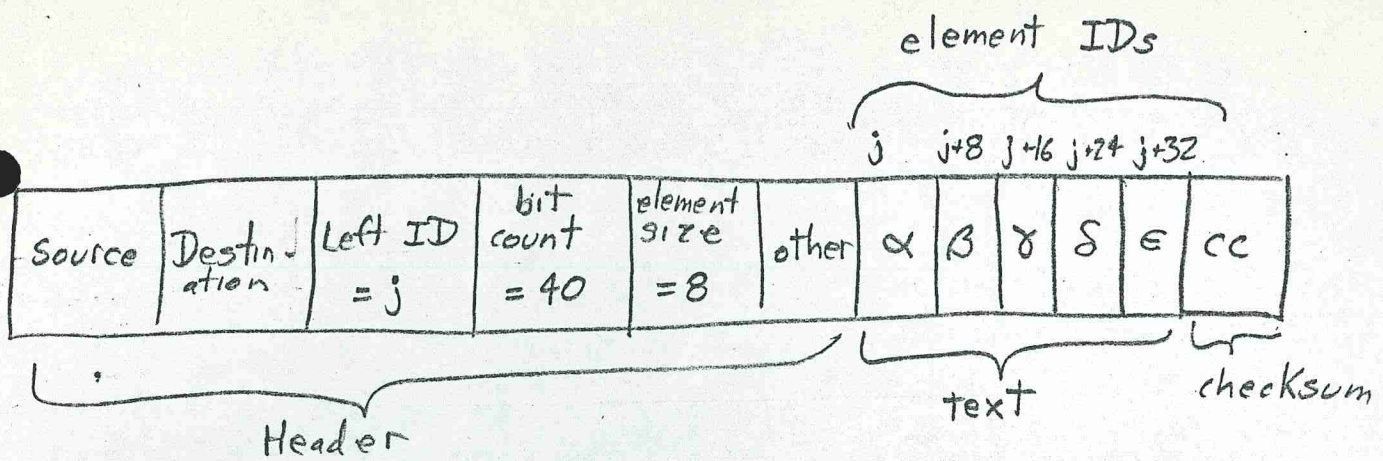
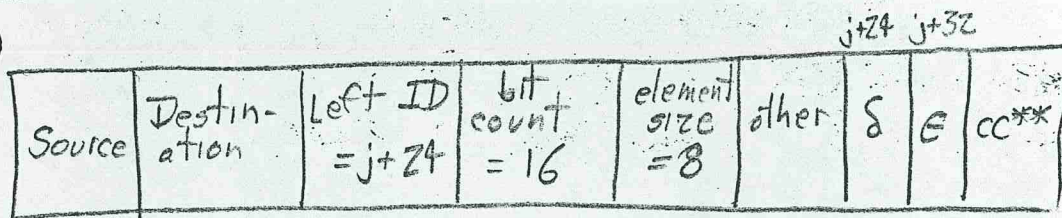
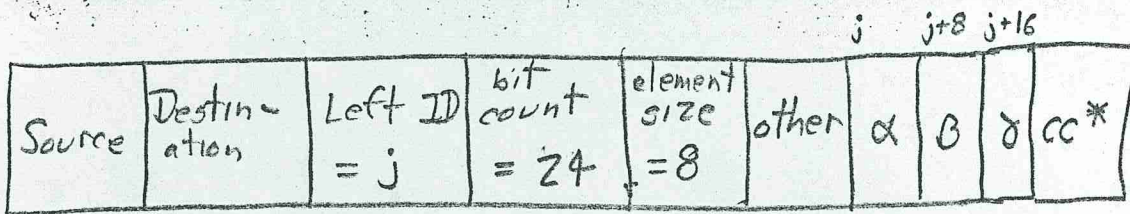


Figure 6



a. Original Message



b. Two Shorter messages
constructed by a gateway

Figure 6
Message Subdivision

Clearly, it is necessary that all networks have the capability of sending a message consisting of the header, checksum, and one element ~~without breakage~~. Note that the element size (in bits) is carried in the header to facilitate gateway message reformatting.

Retransmission, positive acknowledgment and duplicate detection

Earlier, we alluded to a window size, W , which represented the maximum number of bits the sender was allowed to have outstanding (unacknowledged) by the receiver. Initially both sender and receiver have the left edge of the window positioned on bit 0 and the right edge on bit $W-1$. ($W \leq \lfloor K/2 \rfloor$).

The receiver uses the window to detect duplicate transmissions. When a message arrives, its ^{ID (the} left element ID) is compared with the left edge of the window. If these match, the window is advanced to the right as many bits as the bit count indicates. As an acknowledgment, the receiver returns to the sender the ^{new value for} ~~current~~ left edge of the window. If the incoming message ID does not coincide with the present left edge of the window, the receiver determines whether the elements received lie inside or outside the window. If inside, the receiver can either ignore the elements, or mark them received, but not acknowledged. No acknowledgment will be sent for these elements until the missing elements to the left have all been successfully received. ~~However, if the receiver sends as an acknowledgment the current left window edge, this may serve to stimulate retransmission of the missing elements.~~ If the elements lie outside the window, then they are discarded, ~~and an acknowledgment bearing the current left window edge ID is sent to the sender.~~

In fact, ~~these~~ acknowledgments are sent "piggyback" with other messages traveling to the source using a ~~new~~ field in the header labeled as shown in

ACK ~~(see figures 6 and 7)~~. (If no messages are awaiting transmission, ^{when an acknowledgment}

ment is to be sent, then it should be transmitted in a message with no text.) ~~then an empty message is sent.~~

Since an acknowledgement is returned in every message, the source of a transmission must be prepared to receive several "ACKs" containing the

Source	Destination	Left Element ID	Bit Count	Element size	Left ID ACK	Other
--------	-------------	-----------------	-----------	--------------	-------------	-------

HEADER FORMAT

Figure 7

same message ID, and should not use this condition as the sole criterion for instituting retransmission procedures. In fact, the

The sender transmits a group of elements as a message and starts a time-out. If the sender times out before an acknowledgment for the elements is received, the sender retransmits the elements. In the

simplest implementation, out-of-order acknowledgments are ignored (ultimately causing retransmission). In the more complex implementation, they are accepted and the associated elements marked as successfully acknowledged. The sender does not advance the window to the right unless it receives acknowledgments for elements lying consecutively to the right of the left window edge.

One can easily be convinced that the sender can never advance the left edge of his window beyond the left edge of the receiver's window (flow control), and barring total disaster, all elements are eventually acknowledged, even if they ^(or their acknowledgements) require retransmission. Furthermore, the receiver has no trouble distinguishing between duplicate and original elements.

There are several important points to notice. First, the receiver can actually advance his window until its left edge is W bits ahead of the sender (all elements received, but all acknowledgments lost). Since

It is impossible to get an "at-of-order" ACK!!!

the sender will ultimately retransmit to prompt the receiver again for acknowledgments, it is important that the receiver not confuse a re-transmitted element for an original one. If $W > \lfloor K/2 \rfloor$, where $K+1$ is the frame size, then such confusion can occur, hence the restriction $W \leq \lfloor K/2 \rfloor$.

The second point to notice is that a range of implementations is possible, ranging from purely sequential receipt ^(discarding messages received out of order and failing to acknowledge them later) and acknowledgment to a more complex implementation allowing non-sequential receipt ^(although, of course, acknowledgments remain sequential) and acknowledgment.

The third point is that negative acknowledgments are accomplished implicitly [^] the receiver acknowledges only the left window edge on receipt of elements within but not contiguous with the left window edge. If the sender keeps pointers to current left edge, last sent, and right window edge elements, then each acknowledge which fails to advance the sender's left window edge can be treated as a negative acknowledgment.

Flow Control and Element Size.

Initially, the communication channel between TCP's is assumed to consist of 8 bit elements with a window of $W = \lfloor K/2 \rfloor$ bits where $K + 1$ is the frame size.

The receiver may want to change the window size (as may the sender), and either ^{also} may want to change the element size.

Such changes must be done in a controlled manner, so we postulate the existence of control messages sent from one TCP to another. The message format, since it embeds control messages within the data stream, must signal the presence of a control message by a flag in the header. Figure 9 shows the message header format.

He must! see Page 10 + my comment @ top of page 11.

Source	Destination	Left Elem ID	bit count	Elem size	Ack	Type	Text	CC
--------	-------------	-----------------	--------------	--------------	-----	------	------	----

Message Header

Figure 9

For the moment, only three types of messages are proposed.

- a. Type 0 = regular message for a processor terminal
- b. Type 1 = control message
- c. Type 2 = status message

Suppose the receiver wants to tell the sender about a new window size W' . The receiver sends a control message "DO SETWINDOWSIZE W' ." The sender will not respond until the window can be either reduced or expanded on the right to size W' . When this occurs, the sender responds with "WILL SETWINDOWSIZE W' ." Notice that the window should not be expanded beyond $W = \lfloor K/2 \rfloor$ nor should it be reduced (right edge moved leftward) over previously, acknowledged elements.

*these could stand
some amplification*

Element Size

In an early design, it was assumed that all messages in the stream from one TCP to another would consist of messages with the same element size. This size could be changed by negotiation between TCP's. Once we realize that control messages must be embedded in the stream, it is made easier to imagine that every message carries information about element size, and is guaranteed to consist of an integral number of elements, but each message may have a different element size. This permits the multiplexing of messages between several processes with different length elements, but still gives a good flow control mechanism to the receiver. Since message ID's are at the bit level, out-of-order

message^s can still, with some work, be inserted in the correct place in ~~the circular~~ buffer whose length is W bits, if the receiver chooses to accept out-of-order messages.

Since each message can have a different element size, there is no need to negotiate to change it at the ~~ITP~~ TCP level; thus, such negotiation is not ~~part~~ properly part of the ITP.

Checksum

The checksum is optionally computed over the header and message text. If not computed, this field is all 1's, otherwise, it contains a folded checksum. If the originating TCP fails to put a checksum in (i.e. field is all 1's), then at some point at or before the first gateway, this checksum should be computed and put in its proper place.

If a message is broken up, the new pieces should have checksums generated by the gateway.

Messages which fail to pass the checksum test, should be discarded ^{at any point (receiving Host or gateway)}.
~~(and negative acknowledge produced -- e.g. current and unmoved left edge of window).~~

Routing

Initially, the gateway destination should be selected by the TCP from a table of networks versus gateway addresses. The constituent network nodes may choose to re-route messages which are destined for highly congested areas.

Addressing

We take the position that only sufficient addressing need be provided to get a message to a particular TCP. The address field must

→ What does this mean? Is it best for Host efficiency?

be broken into a subfield for network, and one for TCP identifier.
(see figure 10). Although it is not mandatory, most large networks
may prefer a hierarchical addressing structure

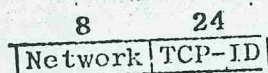


Figure 10

Source or Destination Address

in which the TCP-ID is broken into sections. This is purely up to
each network.

PROBLEM AREAS

1. Setting time-outs for retransmission
2. Header field sizes:
 - a. element size - 8 bits
 - b. left ID - 16 bits
 - c. bit count - 16 bits
 - d. address - 32 bits
 - e. checksum - 16 bits
 - f. ACK - 16 bits
 - g. TYPE - 8 bits
3. Keeping status table sizes down (Walden's Link Table idea).
4. Note: I went to bit addressing instead of element addressing
because it became clear that I want to multiplex messages of many different
sizes along the same TCP-TCP connexion.

5. Status information obtainable by TCP from distant TCP?

(machine word size, window size, TCP performance data,...).

6. Error messages?

7. How to propagate internal network status to HOSTs via ITP?

(e.g. HOST DEAD MSG NOT DELIVERED).

8. Gateway Accounting?

9. Value of K . Its a ^{partly} technical problem. In setting the field size for "left ID" you've implied a value for K , but someone needs to do technical work on this. Remember K is a function of the "time constants" of the constituent networks

10.