

REPORT NO. 1822

# **INTERFACE MESSAGE PROCESSOR**

Specifications for the  
Interconnection  
of a Host and an IMP

Developed for  
the Advanced Research Projects Agency  
by Bolt Beranek and Newman Inc.



SPECIFICATIONS FOR THE INTERCONNECTION OF A HOST AND AN IMP

(JANUARY 1976 REVISION)

NOTE: This Report is periodically updated. Please report change of name or address for updates to:

Network Control Center  
Bolt Beranek and Newman Inc.  
50 Moulton Street  
Cambridge, Mass. 02138  
Attn. Alexander McKenzie

Sponsored by:

Advanced Research Projects Agency  
ARPA Order No. 2351  
Contract No. F 08606-73-C-0032

Submitted to:

IMP Program Manager  
Range Measurements Lab.  
Building 981  
Patrick Air Force Base  
Cocoa Beach, Florida 32925

UPDATE STATUS

The pages of this report should be dated as follows:

Title page, this page, pp. iii to vii and H-1 to H-27: 1/76

All other pages: 12/75

IMPORTANT NOTE

January 1976

\* \* \* \* \*

\* Sections 3 and 5 of this revision \*

\* describe a new protocol which is \*

\* scheduled for implementation very \*

\* early this year. Before attempting \*

\* to use this protocol please confirm \*

\* its existence by calling the Network \*

\* Control Center at (617) 661-0100. \*

\* \* \* \* \*

## TABLE OF CONTENTS

1. INTRODUCTION	1-1
2. GENERAL REQUIREMENTS	2-1
2.1 Physical Configuration	2-1
2.2 Description of Equipment	2-9
2.3 Interfacing	2-12
3. SYSTEM OPERATION	3-1
3.1 Messages and Message-ids	3-1
3.2 Establishing and Breaking Host/IMP Communications	3-6
3.3 Host-to-IMP Leader Format	3-12
3.4 IMP-to-Host Leader Format	3-21
3.5 Word Length Mismatch and Message Boundaries	3-31
3.6 Two Debugging Aids	3-33
3.7 Uncontrolled Packets	3-35
3.8 Non-Blocking Host Interface	3-36
4. HARDWARE REQUIREMENTS AND DESCRIPTION	4-1
4.1 Structure of the Standard Host/IMP Interface	4-2
4.2 IMP/Host Handshaking	4-4
4.3 End-of-Message Indication	4-10
4.4 Master Ready Lines	4-10
4.5 Host Cable Connections	4-13
4.5.1 Connection to a Local Host	4-17
4.5.2 Connection to a Distant Host	4-22

5.	IMP BACKGROUND PROGRAMS	5-1
5.1	TTY	5-3
5.2	DEBUG	5-4
5.3	PARAMETER-CHANGE	5-5
5.4	DISCARD	5-9
5.5	TRACE	5-9
5.6	STATISTICS	5-14
5.6.1	Snapshots	5-16
5.6.2	Cumulative Statistics	5-19
5.6.3	Message Generator	5-22
APPENDIX A	Old-Style Leader Formats	A-1
A.1	Host-to-IMP Leader Format	A-2
A.2	IMP-to-Host Leader Format	A-4
B	Recommendations for Host Implementation of the Host/IMP Interface	B-1
B.1	Ready Line Philosophy	B-2
B.2	Programming the I/O Routines	B-6
B.3	Host Ready Line Implementation	B-8
B.4	Summary of Ready Line Controls	B-9
C	Local Host Connection Electrical Characteristics	C-1
D	Driver Receiver for Distant Host	D-1
D.1	Differential Receiver PAC Model CC-124	D-2
D.1.1	Circuit Description	D-2
D.1.2	Terminating Network	D-2
D.1.3	Specifications	D-2

D.2	Differential Line Driver PAC, Model CC-125	D-4
D.2.1	Circuit Function	D-4
D.2.2	Terminating Network	D-4
D.2.3	Specifications	D-5
E	ASCII Codes	E-1
F	Very Distant Host Interface	F-1
F.1	Philosophy of the Very Distant Host Interface	F-2
F.2	The Reliable Transmission Package	F-5
F.3	The Error Detecting Special Host Interface	F-14
F.3.1	Message Formatting	F-14
F.3.2	Character Codes	F-19
F.3.3	The Cyclic Redundancy Check	F-20
F.3.4	Connection to a Modem	F-24
G	IMP Power Wiring Convention	G-1
H	Interfacing a Private Line Interface (PLI) to an IMP and a Host to a PLI	H-1
H.1	Philosophy of the Private Line Interface (PLI)	H-2
H.2	Secure PLI Functional Specification	H-5
H.3	Bitstream PLI Functional Specification	H-9
H.4	Secure PLI Physical Characteristics	H-12

H.5	Secure PLI Cable Entry and Conduit	H-14
H.5.1	AC Power	H-14
H.5.2	IMP Connection	H-16
H.5.3	Red Host or Data Connection	H-18
H.5.4	Key Generator Connections	H-20
H.6	Bitstream PLI Physical Configuration	H-22
H.7	Software Interfaces to the PLI	H-24

## LIST OF FIGURES

Figure 1-1	A Typical Section of the ARPANET	1-2
1-2	The Model 516 IMP, Modem Cabinet, and IMP Teletype*	1-3
1-3	The Model 316 IMP and IMP Teletype*	1-4
1-4	The Terminal IMP and IMP Teletype*	1-5
1-5	The Pluribus IMP and IMP Terminal*	1-6
2-1	IMP Equipment	2-2
2-2	Minimum Floor Area Required for 516 IMP	2-5
2-3	Minimum Floor Area Required for 316 IMP	2-6
2-4	Minimum Floor Area Required for 316 TIP	2-7
2-5	Minimum Floor Area Required for Pluribus IMP (per rack)	2-8
2-6	Host/IMP Interface	2-12
3-1	Host-to-IMP Leader Format	3-12
3-2	IMP-to-Host Leader Format	3-21
4-1	Simplified Illustration of the Host/IMP Interface	4-3
4-2	Simplified Control Logic for Host/IMP Handshaking	4-5
4-3	IMP Ready Test and IMP Master Ready Lines	4-11
4-4	516 Host Cable Signals (Local Host)	4-19

\*Photographs by Hutchins Photography, Inc., Belmont, Mass.



Figure 4-5	316 Host Cable Signals (Local Host)	4-20
4-6	Pluribus IMP Host Cable Signals (Local Host)	4-21
4-7	Host Cable Signals (Distant Host)	4-23
5-1	Trace Message Format	5-12
5-2	Trace Block Format	5-13
5-3	Snapshot Message Format	5-17
5-4	Cumulative Statistics Message Format	5-20
A-1	Old-style Host-to-IMP Leader Format	A-2
A-2	Old-style IMP-to-Host Leader Format	A-5
B-1	Host-to-IMP (Host's Special Interface)	B-3
B-2	IMP to Host (Host's Special Interface)	B-4
D-1	Differential Receiver PAC Model CC-124 Schematic Diagram and Logic Symbol (Shown as Connected in IMP)	D-3
D-2	Differential Line Driver PAC Model CC-125 Schematic Diagram and Logic Symbol (Shown as Connected in IMP)	D-6
F-1	Normal IMP/Host Connection	F-2
F-2	IMP/Host Connection for Very Distant Host	F-4
F-3	Packet Format	F-5
F-4	Control Word Format	F-6
F-5	Segmentation of a Message into Packets for the Very Distant Host Interface	F-11

Figure F-6	Adaptation of Special Host Interface	F-14
F-7	Packet Format on Line	F-15
F-8	Output Check Register	F-21
G-1	Twistlock Plug (Viewed from Pin Side)	G-3
G-2	Twistlock Receptacle (Viewed from Socket Side)	G-3
H-1	Example of PLI and Network Configuration	H-3
H-2	Secure PLI Configuration	H-6
H-3	Secure PLI Outline Dimensions	H-13
H-4	Possible Secure PLI Interface Configurations	H-17



## 1. INTRODUCTION

The ARPANET provides a capability for geographically separated computers, called *Hosts*, to communicate with each other. The Host computers typically differ from one another in type, speed, word length, operating system, etc. Each Host computer is connected into the network through a local small computer, called an *Interface Message Processor (IMP)*, that is located on its premises; a typical network section is shown in Figure 1-1. The complete network is formed by interconnecting these IMPs through wideband communication lines supplied by common carriers. Each IMP is then programmed to store and forward messages to the neighboring IMPs in the network. During a typical operation, a Host passes a message to its IMP; this message is then passed from IMP to IMP through the network until it finally arrives at the destination IMP, which in turn passes it along to the destination Host.

Several models of IMPs are currently available. All perform the basic function of a store and forward mode, but they have different physical configurations and data handling rates. The Model 516 (see Figure 1-2) is the original IMP and is no longer normally installed. The Model 316 (see Figure 1-3) is a less expensive and somewhat slower version of the original IMP. The *Terminal IMP* or *TIP* (see Figure 1-4) is a Model 316 IMP mounted in a double hi-boy rack along with a BBN *Multi-Line Controller (MLC)*. The Terminal IMP is designed to connect both Hosts and up to 64 terminals to the network; the terminals are given access to the network directly, without an intervening Host. The Pluribus IMP (see Figure 1-5), the most recent addition to the IMP family, is based on a flexible multiprocessor design and is housed in from one to several racks, depending on precise speed and capacity.

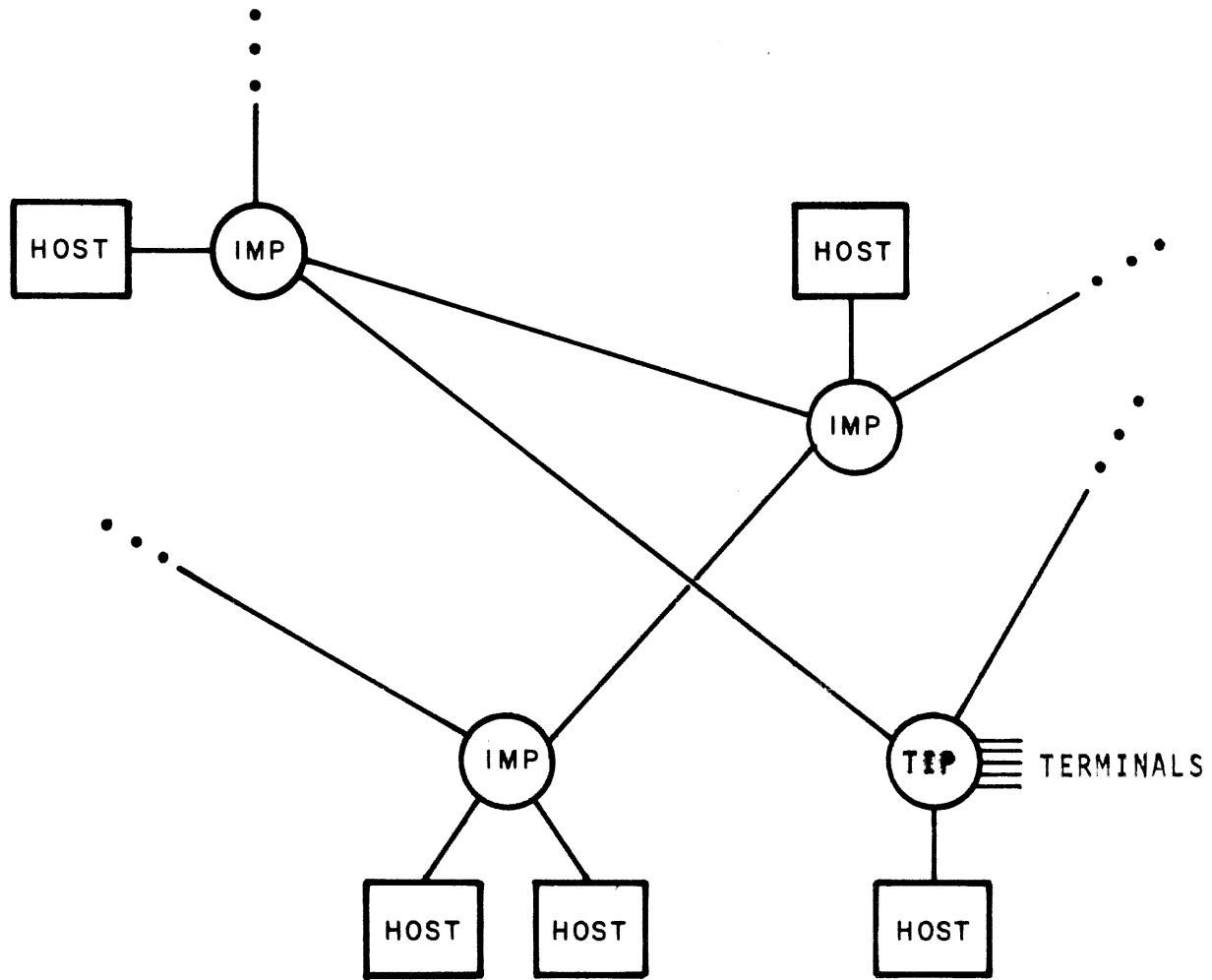


FIG.1-1 A TYPICAL SECTION OF THE ARPANET

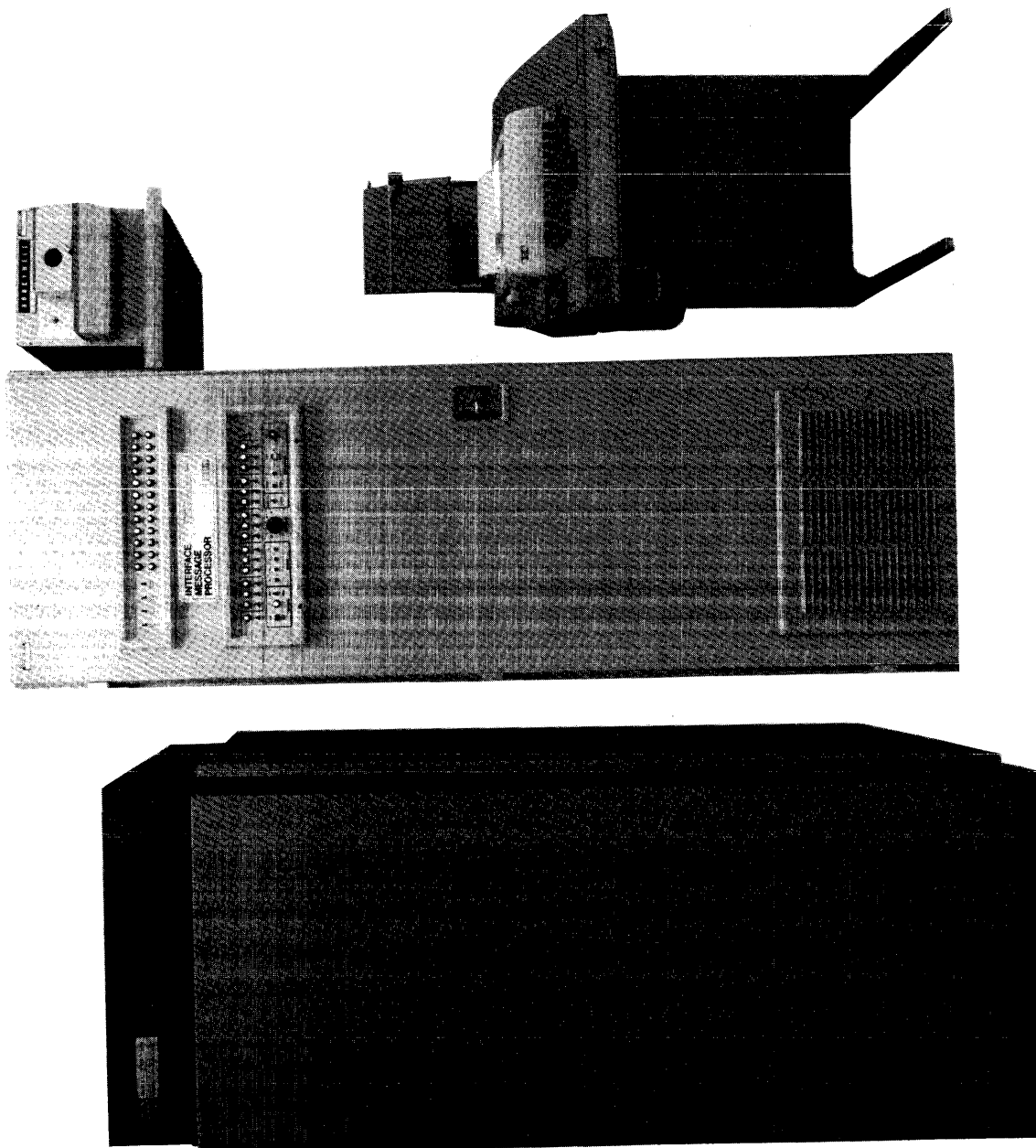


FIG. 1-2 THE MODEL 516 IMP, MODEM CABINET, AND IMP TELETYPE

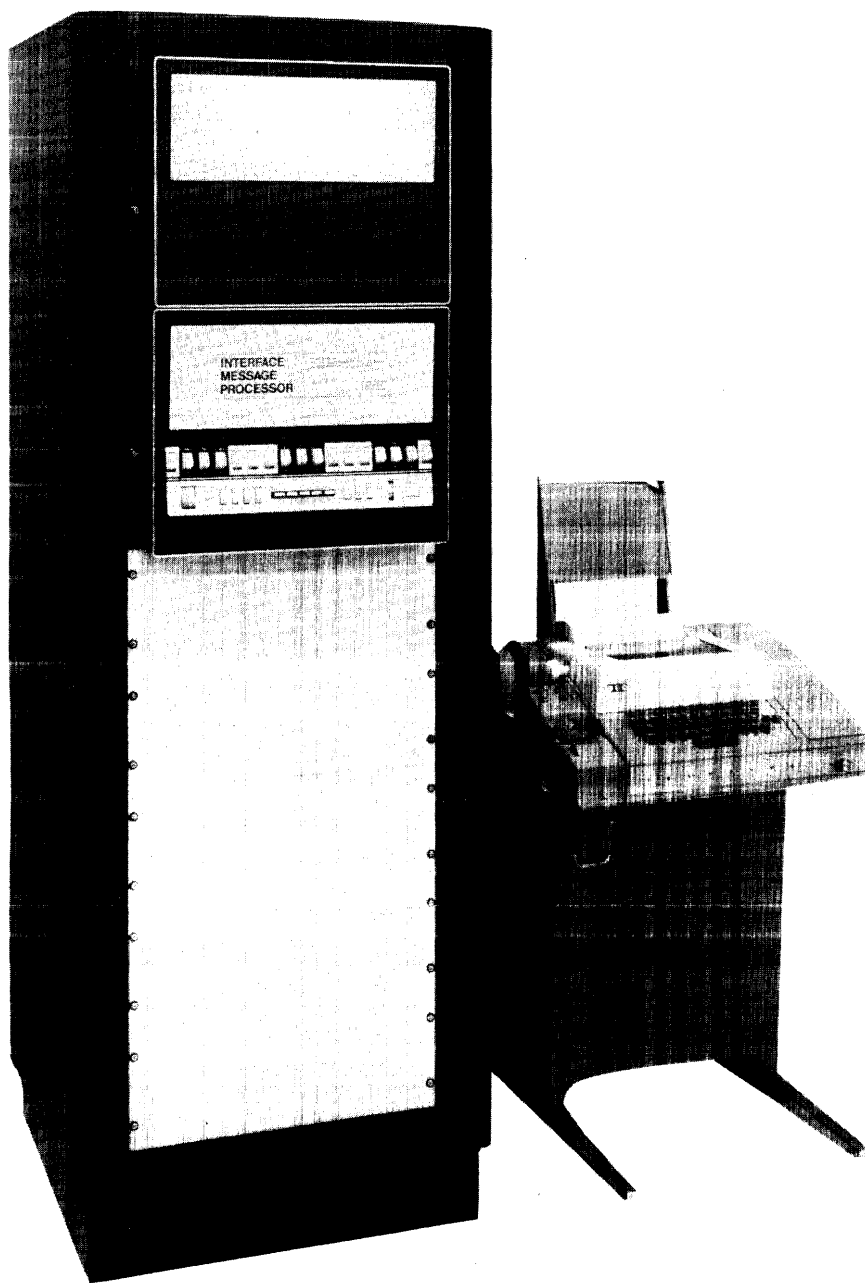


FIG. 1-3 THE MODEL 316 IMP AND IMP TELETYPE.

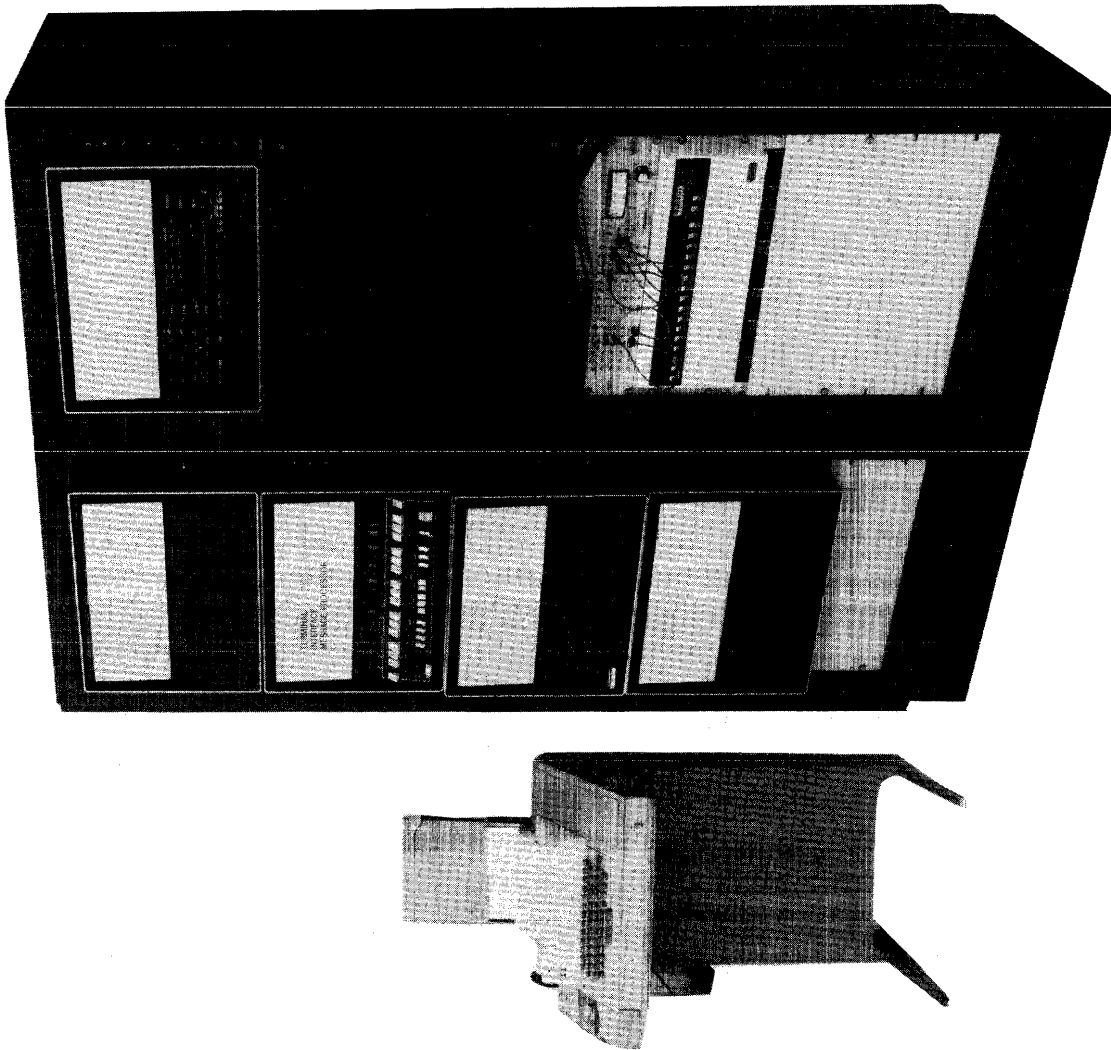


FIG. 1-4 THE TERMINAL IMP AND IMP TELETYPE





FIG. 1-5 THE PLURIBUS IMP  
AND IMP TERMINAL

This document contains the specifications for interconnecting a Host and an IMP and may be subject to change. The interconnection of a Host and an IMP is a joint effort that requires the Host personnel to provide interfacing hardware and software. Although we have tried to provide sufficient information to assist the Host personnel in the design of the interface, problems and questions that we have not anticipated will undoubtedly arise. These questions should be addressed to:

Network Control Center  
Bolt Beranek and Newman Inc.  
50 Moulton Street  
Cambridge, Massachusetts 02138

We strongly recommend that the personnel responsible for the design of the Host hardware and software interfaces visit in Cambridge with the technical staff of Bolt Beranek and Newman Inc. for a thorough review of the designs prior to implementation. We feel that this procedure will help to minimize the difficulties that will be encountered in connecting the Host and the IMP.



## 2. GENERAL REQUIREMENTS

In this section, we describe the physical configuration of the IMP, the space and power requirements, the equipment necessary to interconnect the IMP and Host, and the facilities that must be provided by the IMP site to assist with installation and maintenance of the IMP.

### 2.1 Physical Configuration

As shown in Figure 2-1, four pieces of equipment are provided: the IMP itself, which is a modified Honeywell H-516R, Honeywell H-316, or BBN Pluribus computer; an ASR-33 Teletype or Infoton Vistar\*; a high-speed paper tape reader (optional); and a cabinet, approximately the same size as the Model 516R, that contains *up to four modems* connecting the IMP to the communication lines. The telephone company will supply modems only for the communication lines actually installed. In addition, the telephone company usually supplies auxiliary equipment that may vary from site to site and need not be located near the modem cabinet or the IMP.

A Host is connected to an IMP by a Host cable.\*\* The particular cabling scheme is determined by the distance between the Host and the IMP. A *local Host* (one close to the IMP) is connected by a 30-foot cable\*\*\* *that is supplied with the IMP*. This cable connects a standard Host/IMP interface unit built into the IMP to a *special* interface provided by the Host.

---

\*The Vistar is a keyboard/display-type terminal used with the Pluribus. It performs the same functions as the ASR-33 Teletype.

\*\*The cables in Figure 2-1 are drawn only schematically rather than in their actual positions.

\*\*\*The length of this cable is limited by the characteristics of the cable drivers in the IMP.

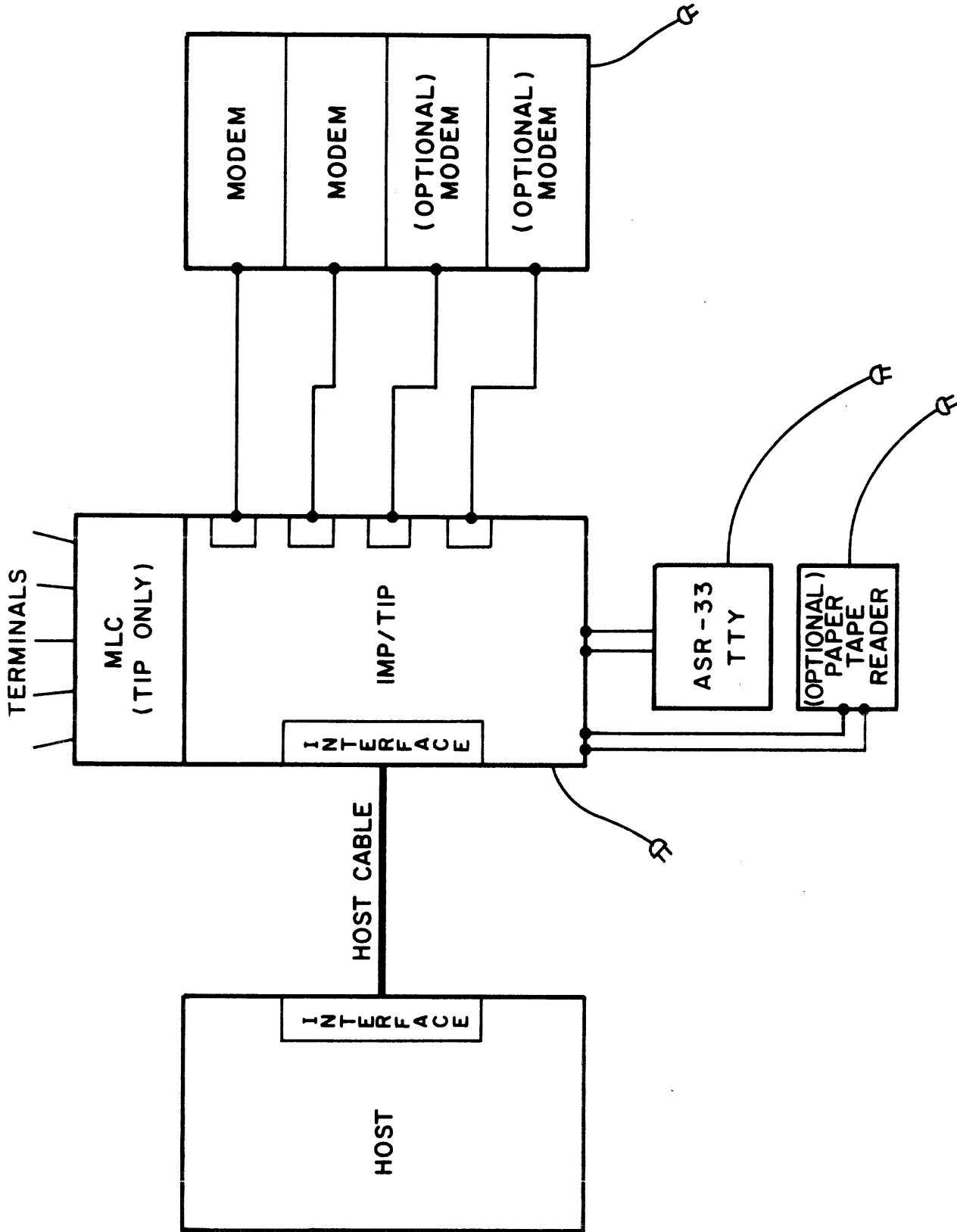


FIG.2-1 IMP EQUIPMENT

A *distant Host*\* may be located up to 2000 feet from the IMP, but an addition to the standard Host/IMP interface is required to modify the line-driving scheme. The Host personnel must design a special interface that is compatible and *must supply the connecting cable* as specified in Sec. 4.5.2. Since additional IMP hardware must be supplied, the decision to connect a distant Host must be made known well in advance. A distant Host will usually be connected to an IMP which has one or more local Hosts.

A *very distant Host* may be located even farther from the IMP, using an entirely different interface arrangement which is described in Appendix F. Basically, the very distant Host interface is designed for use over communication circuits with speeds up to 230.4 kilobits/second and up to tens (perhaps hundreds) of miles long. The communication protocol used with this interface includes a 24-bit cyclic redundancy check and a positive acknowledgment scheme.

A separate 30-foot cable is provided with the IMP for the connection to each modem. In addition, cables are provided for connecting the terminal (Teletype or Vistar) and paper tape reader (if supplied) to the IMP. For the H-516R and H-316 IMPs, cables exit from the IMP through the bottom of the rear panel. Cables will exit from the modem unit *through the bottom* of the modem cabinet; if a site does not have a false floor, other modem cable arrangements are easily provided. Cables are connected to the Pluribus IMP via a fantail panel located at the rear of the machine.

---

\*Not available with the Pluribus IMP.

Figures 2-2, 2-3, 2-4, and 2-5 depict the floor space requirements for the 516 IMP, the 316 IMP, the (maximum size) 316 TIP, and the (minimum size) Pluribus IMP respectively. Some configurations of the 316 TIP may only require the same floor space as a 316 IMP, and some Pluribus IMPs may require several racks side by side; the Network Control Center can furnish details for each installation.

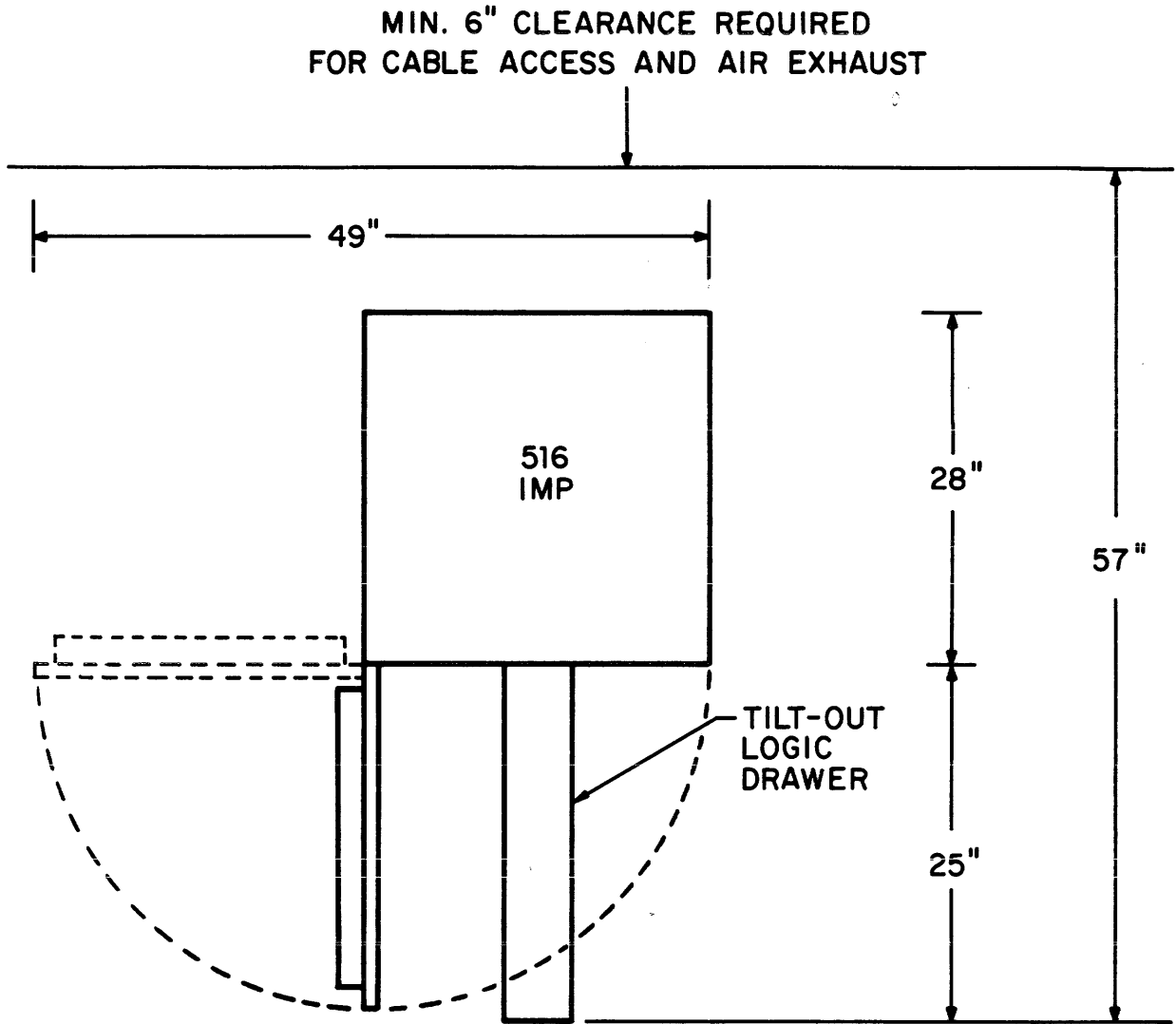
With the Honeywell machines, provision should be made to place the ASR-33 Teletype close to the IMP. The ASR-33 occupies approximately 2' x 2' of floor space. (The optional paper tape reader must be placed nearby if it is supplied.\* Its dimensions are 11x11x23 inches (WIDTHxHEIGHTxDEPTH). A convenient location is the top of the IMP cabinet, if overhead space permits.) With the Pluribus machine, table space should be provided nearby for the Infoton Vistar. Its dimensions are 20x13x24 inches. (Again, the optional paper tape reader must be placed nearby if it is supplied.\* Its dimensions are 20x8x22 inches. It can be located on top of the IMP cabinet if overhead space permits.)

A small lockable cabinet is needed on the Host premises for the storage of IMP-related materials (e.g., manuals, test tapes, scope, tool box, etc.). Finally, a telephone should be located within reach of both the terminal and the operating panel of the IMP for use during diagnosis and debugging. (Pluribus IMPs may be supplied without an operating panel.)

The locations of the IMP, modem cabinet, paper tape reader, and Teletype are to be selected by the Host personnel. These pieces of equipment should be placed within approximately eight

---

\*To determine whether a paper tape reader will be supplied, a site may contact the Network Control Center.



NOTE: DIMENSIONS ARE TO NEAREST INCH

TOP VIEW

FIG.2-2 MINIMUM FLOOR AREA REQUIRED FOR 516 IMP



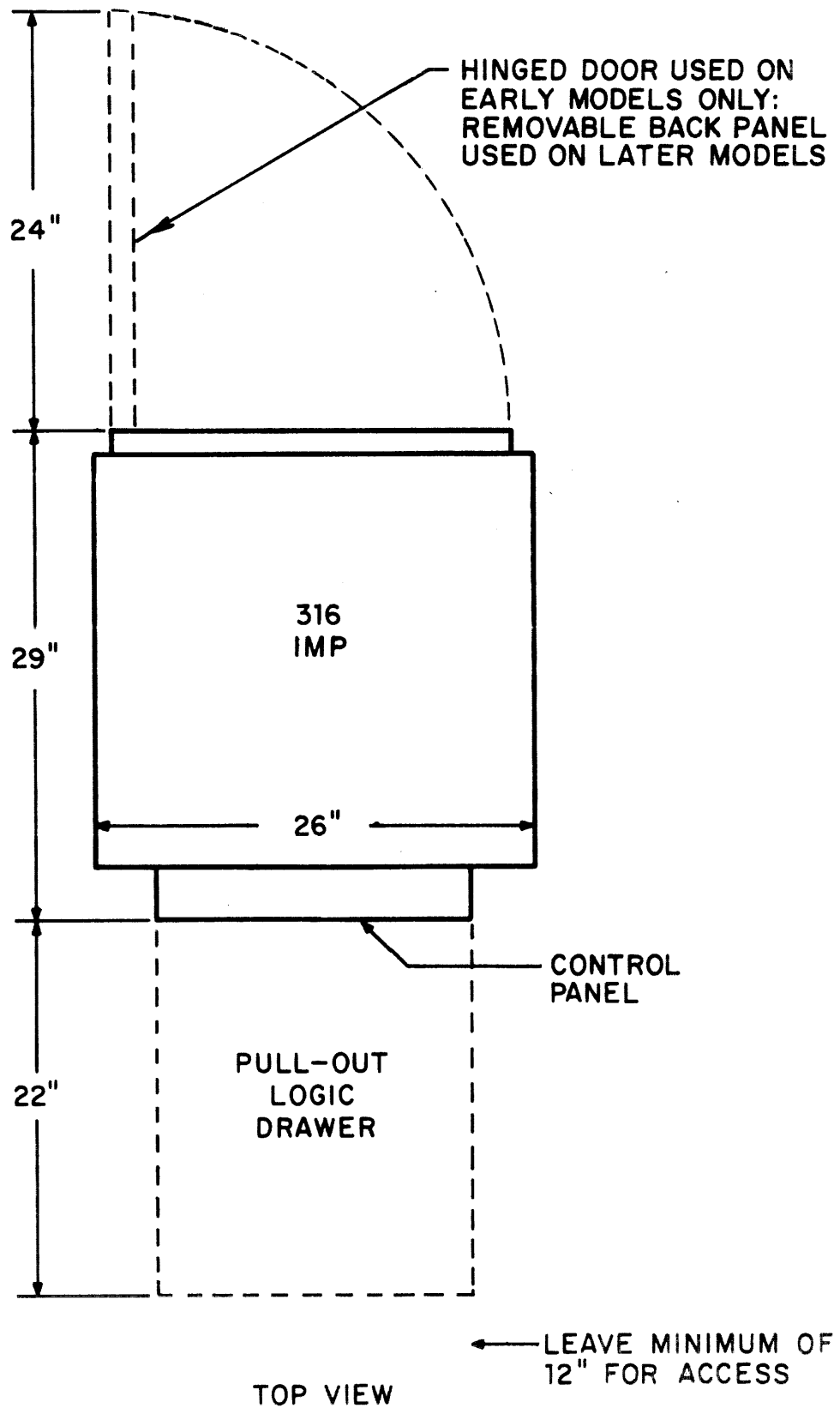


FIG.2-3 MINIMUM FLOOR AREA REQUIRED FOR 316 IMP

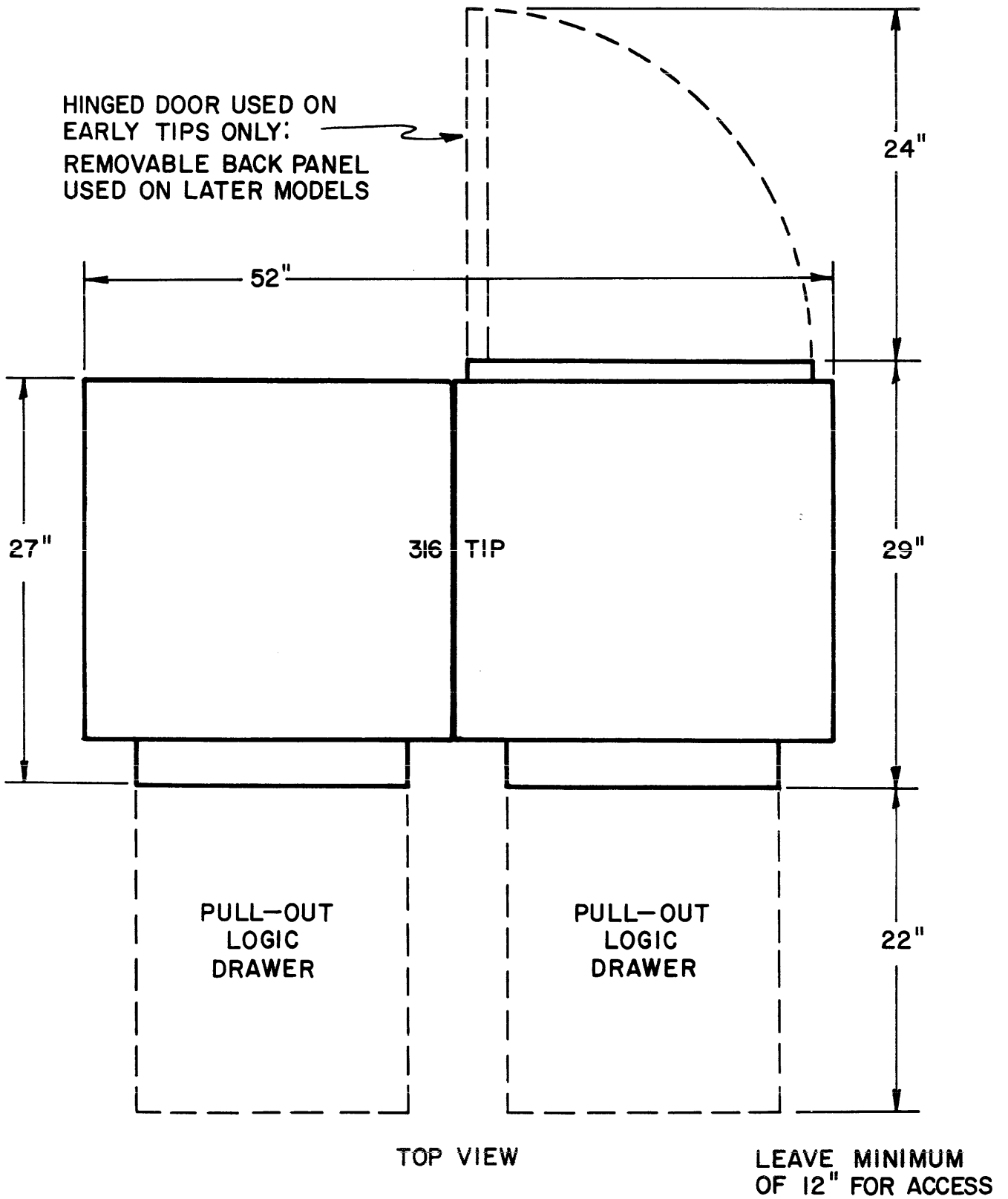
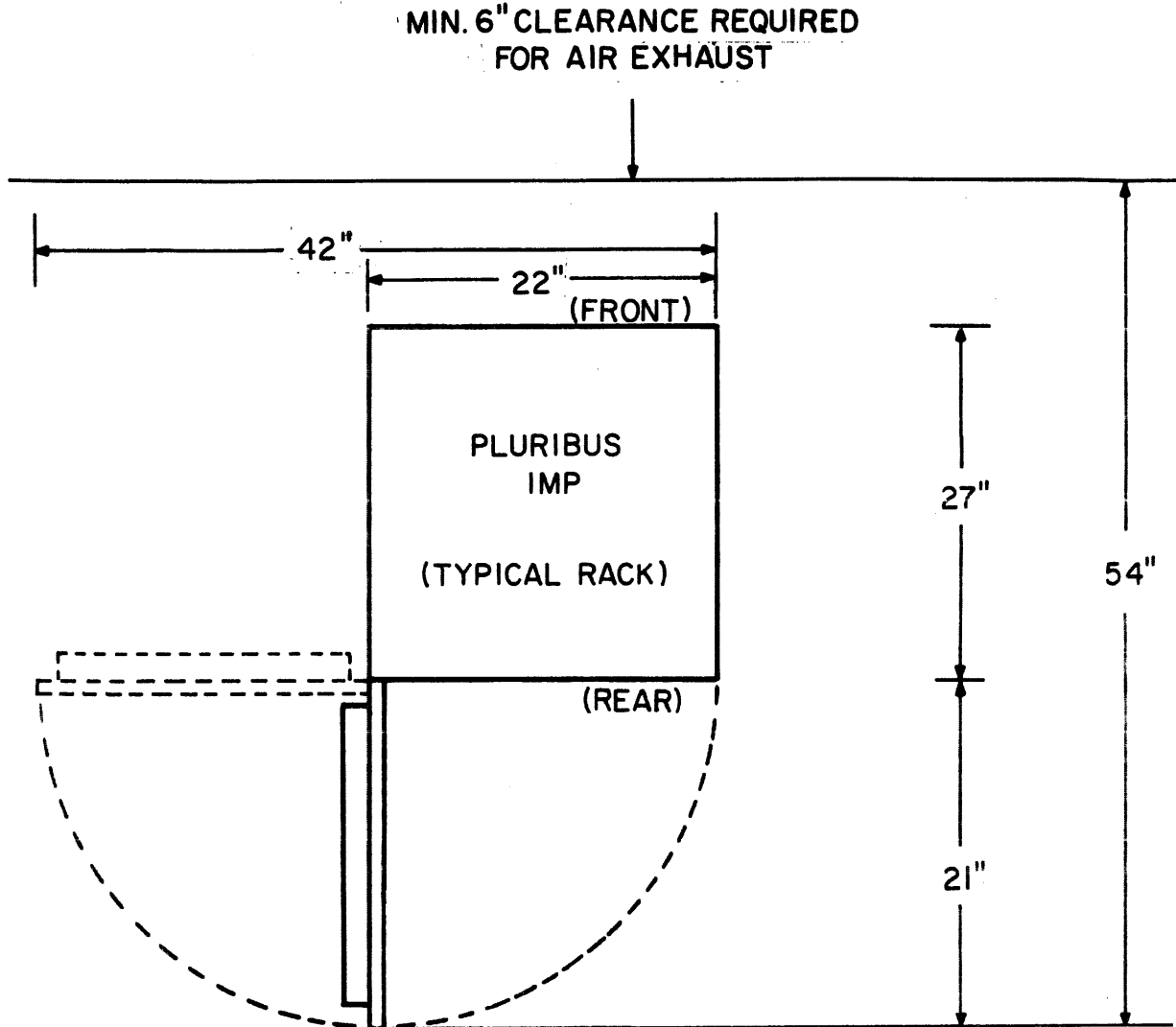


FIG. 2-4 MINIMUM FLOOR AREA REQUIRED FOR 316 TIP



NOTES: DIMENSIONS ARE TO NEAREST INCH;  
CABLES CONNECT ON REAR (DOOR) SIDE

**TOP VIEW**

FIG. 2-5 MINIMUM FLOOR AREA REQUIRED  
FOR PLURIBUS IMP (PER RACK)

feet of one another. A minimum of thirty square feet of floor space is required for the equipment, and additional space must be available for accessing the machine during maintenance and debugging. Access to the Model 516 IMP is via a full-length front door, which is hinged on the left side. Access to the 316 IMPs is via drawers which slide to the front. Access to the Pluribus IMP is via full-length rear doors and removable front panels. Access to the modem cabinet is via a removable front panel.

In addition to the modem cabinet, the telephone company may provide another cabinet to contain the auxiliary equipment. It is recommended that this auxiliary equipment be placed in an *inconspicuous* location on the Host premises, such as in a telephone company equipment room, since immediate access to this equipment is not necessary.

## 2.2 Description of Equipment

External dimensions, approximate weights, and power requirements of the various IMP models are given in Table 2-1. The paper tape reader weighs approximately 25 pounds, the ASR-33 Teletype weighs approximately 56 pounds, and the Infoton Vistar weighs approximately 55 pounds.

The Model 516 IMP is a ruggedized unit with E.M.I. protection. All IMPs will operate in an ambient environment from 0 to 45°C (Pluribus IMPs should not be operated at temperatures over 30°C unless special provisions are made) and up to 95% humidity. However, these features have been included for reliability and, in general, an environment suitable for most digital computing equipment should be provided; i.e., air-conditioned and free from excessive dust and moisture.

TABLE 2-1

<u>Model</u>	<u>Size (inches)</u>			<u>Weight (lbs)</u>	<u>Power (watts)</u>
	<u>Height</u>	<u>Width</u>	<u>Depth</u>		
516R IMP	74	24	28	990	2100
316 IMP	73	26	28	525	750
316 TIP	73	52	28	920	2200
expansion cabinet	39	25	28	100	0
Pluribus IMP (per rack)	68	22	26	550	3000 (approx)

The power requirements for the Honeywell IMP equipment are as follows:

- a) *IMP*: 115 VAC  $\pm$  10%; 60 Hz  $\pm$  5%, single phase. The line cord is 15 ft. long and contains 3-wire cable terminated by a 30-amp Hubbell 3331G twistlock connector (for wiring convention, see Appendix G).
- b) *High-speed reader* (optional): 115 VAC  $\pm$  10%; 60 Hz, single-phase at 125 watts. (The line must withstand 10-amp surges at 125 VAC.) The line cord is 6 ft. long and is terminated in a standard 3-wire grounded plug.
- c) *ASR-33*: 115 VAC  $\pm$  10%; 60 Hz  $\pm$  0.45 Hz, single phase at 230 watts. The line cord is 8 ft. long and is terminated in a standard 3-wire grounded plug.

Power for the Pluribus equipment is supplied via one 3-phase 208/110 volt wye 60 Hz connection *per rack*. Each power cord is 20 feet long and is terminated by a Hubbell 45215 twistlock

connector. Each circuit must supply 30 amps per leg. Sufficient convenience outlets for debugging equipment, the Infoton Vistar, and paper tape reader are provided on the Pluribus itself.

The Host must provide an appropriate power receptacle (located within 15 feet) for the IMP power plug and it is recommended that a separate fuse or circuit breaker be provided on the IMP's power line. (The Honeywell IMP normally draws about 20 amps, but the line must be capable of supplying up to 30 amps.) The IMP's chassis is connected to the ground (third) lead of the power plug, which is completely isolated from the signal return (i.e., "signal ground"). *If at all feasible*, the power to the IMP should be provided from the same transformer that delivers power to the Host in order to insure a common ground. For Honeywell equipment, three 115-VAC wall sockets (located within 5 feet of the IMP) are required to power the Teletype, paper tape reader, and an IMP debugging oscilloscope used during installation and maintenance. The line for these sockets should be fused for 20 amps and should be powered from the same transformer as the IMP, if feasible.

The modem cabinet dimensions are 68-1/8" x 28" x 28"; it weighs up to 750 lbs and requires up to 15 amps of standard 115 VAC power. The modem operates in an ambient environment of 40° to 120°F and up to 95% humidity. The Host must provide power for the modem from the same transformer that delivers power to the IMP. A standard 3-connector non-locking, non-twist plug is normally provided with the modem. The telephone company also recommends that a separate fuse or circuit breaker be provided on the power line to the modem. (The auxiliary equipment is a non-standard item that will vary from site to site; the size is generally no larger than the size of the modem cabinet and may be as small as a 2' x 3' wall mounting. A separate power outlet will also be needed for this equipment.)

In all, the Honeywell equipment requires six receptacles, and Pluribus machines require one receptacle per rack plus one for the modem cabinet. The site should plan to provide the power necessary for the phone company equipment after preliminary discussions with the local telephone company representatives and before the circuit installation date.

### 2.3 Interfacing

The Host/IMP interface is subdivided into two separate units, as illustrated in Figure 2-6.

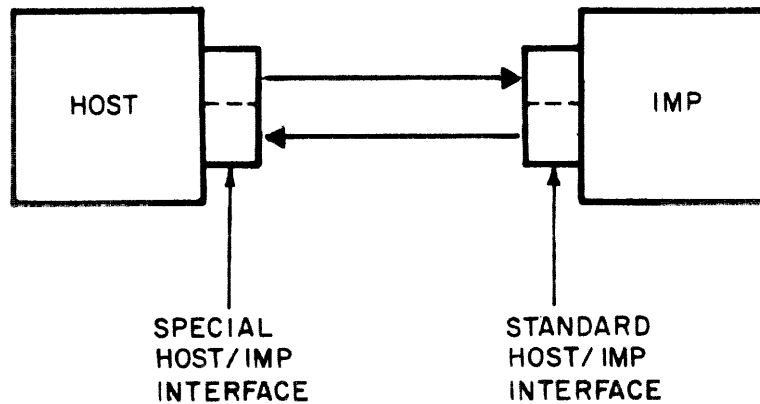


FIG. 2-6 HOST/IMP INTERFACE

The right-hand (standard) unit is built into the IMP and contains logic that is standard for all Host/IMP interfaces. The left-hand unit contains the special equipment for interfacing directly to the particular Host. *An addition to the standard Host/IMP interface is required for a distant Host.* Standard signals pass on the Host cable between these two halves; all special logic and signal adjustments (which vary from Host to Host) are handled in the left-hand portion. *Each participating Host will be responsible for the design and construction of its own special unit to mate to the standard Host/IMP interface unit.* The logical operation of this unit will be the same, regardless of whether a Host is local or distant; however, a different electrical signaling scheme is required to handle a distant Host. A detailed description of the requirements for the special unit is given in Section 4. The very distant Host interface follows the same general philosophy of a standard interface unit at the IMP end and a special interface unit at the Host end, but uses a completely different signaling scheme as described in Appendix F. Still another Host interfacing scheme, making use of the Private Line Interface (PLI), is described in Appendix H.

The Host computer and the IMP communicate by transmitting messages over the Host cable. The format for this communication has been established and is described in Section 3. Each Host is responsible for providing the necessary Network Control Program in the Host computer.

An IMP test program is available for use during installation and testing. In addition to checking various functions in the IMP, this program provides a mechanism for checkout of the Host's special interface. The program repeatedly transmits a



message to the Host, a copy of which it expects the Host to return with any Host padding, or data (Section 3.5). *The Host should plan to provide an appropriate test program to operate in conjunction with this IMP test program.*

### 3. SYSTEM OPERATION

#### 3.1 Messages and Message-ids

Hosts communicate with each other via *regular messages*. A regular message may vary in length from 96 up to 8159 bits, the first 96 of which are control bits called the *leader*. The leader is also used for sending control messages between the Host and its IMP. The remainder of the message is the *data*, or the *text*.

For each regular message, the Host specifies a *destination*, consisting of IMP, Host, and *handling type*. These three parameters uniquely specify a *connection* between source and destination Hosts. The handling type gives the connection specific characteristics, such as priority or non-priority transmission (see below). Additional leader space has been reserved for a fourth parameter, to be used in future inter-network addressing. For each connection, messages are delivered to the destination in the same order that they were transmitted by the source.

For each regular message, the Host also specifies a 12-bit identifier, the *message-id\**. The message-id, together with the destination of the message, is used as the "name" of the message. The IMP will use this name to inform the Host of the disposition of the message. Therefore, if the Host refrains from re-using a particular message-id value (to a given destination) until the IMP has responded about that message-id, messages will remain uniquely identified and the Host can retransmit them in the event of a failure within the network.

---

\* Until mid-1973 the first eight bits of the message-id field were called the "link".

After receiving a regular message from a Host connected to it, an IMP breaks the message into several packets (currently the maximum data bits/packet is 1008) and passes these through the network in the direction of the destination. Eventually, when all packets arrive at the destination, they are reassembled to form the original message and passed to the destination Host. The destination IMP returns a positive acknowledgment for receipt of the message to the source IMP, which in turn passes this acknowledgment to the source Host. This acknowledgment is called a *Ready for Next Message (RFNM)* and identifies the message being acknowledged by name. In some relatively rare cases, however, the message may be lost in the network due to an IMP failure; in such cases an *Incomplete Transmission* message will be returned to the source Host instead of a RFNM. Again, in this case, the message which was incompletely transmitted is identified by name.

If a response from the destination IMP (either *RFNM* or *Incomplete Transmission*) is itself lost in the network, this condition will be detected by the source IMP, which will automatically inquire of the destination IMP whether the original message was correctly transmitted or not, and repeat the inquiry until a response is received from the destination IMP. This inquiry mechanism is timeout-driven, and each timeout period may be as little as 30 or as much as 45 seconds in length.

When a message arrives at its destination, the leader is modified to indicate the source Host, but the message-id field is passed through unchanged. Thus, in addition to providing message identification between a Host and its local IMP, the message-id can provide a means for Hosts to identify messages between themselves. For example, the message-id can be used for

multiplexing several independent data streams, or for keeping track of the portions of a single data stream being sent "in parallel" through the network.

If the *priority* bit of the handling type is set, the message will be expedited through the network by being placed at the front of the various transmission queues it will encounter along the way. This can be useful for transactions requiring minimal delay (e.g., remote echoing or the exchange of control information) but should be used judiciously, since the more it is used the less effect each further use will have.

In order to prevent various types of deadlocks within the network, a source IMP must guarantee that the destination IMP will have enough storage to accept the message it is about to send. This is done by preceding each message with a short "request for buffer space" message. When the destination has enough buffer space to receive another message, it returns an "allocation" to the source IMP, which can then send the message it has been holding.

There are several situations in which an IMP may temporarily block\* the transmission of a message from the source Host to the source IMP. In general, any such blockage will last for only a few milliseconds, but in some cases the blockage may be indefinite. In at least one such case the IMP will be unable to accept the remainder of a message from its Host until it frees buffer space by delivering some message to the Host (it is for this reason that half-duplex Host-IMP interfaces are prohibited). In all such cases, in order to prevent permanently hanging up transmission between the Host and the IMP, the source IMP will

---

\* By failing to provide *Ready-for-Next-Bit*, see Section 4.1.

discard the message after a wait of about fifteen seconds and return a type 9 (sub-type 4) message (see Section 3.4) to the Host, thus limiting the length of time that the interface will be blocked. Similarly, once a Host has begun to send the IMP a message, it must be prepared to deliver the entirety of that message to the IMP promptly. In particular, the IMP will discard any message that is not completely received from its Host in fifteen seconds and return a type 9 (sub-type 2) message to the Host (see Section 3.4).

One situation under which interface blocking will occur is when the source IMP must wait to receive an allocation from the destination IMP. Since a Host cannot send other messages into the network while its interface is blocked, it is desirable to expedite the "allocation" mechanism, and this is done in two different ways depending upon message length. For one-packet messages, the message itself is sent as its own request. Thus, if space is available, the message is immediately accepted and no additional delay is incurred. For multi-packet messages, when the destination IMP is about to return a RFNM it reserves storage in *anticipation* of the source Host's next message, and returns the allocation along with the acknowledgment. Thus, when the source IMP eventually sends its Host the RFNM, it is also implicitly informing it of the allocation now being available.\* If the Host responds promptly with another message on that same connection (message-id is irrelevant), the message can be forwarded immediately, avoiding any set-up delay waiting for an allocation. If this allocation remains unused for about

---

\* In some (rare) cases the destination is unable to reserve storage immediately, and returns a RFNM without the reservation. Currently, the destination waits 1/2 second, attempting to reserve storage, before returning the RFNM without an accompanying reservation.

125 ms, it is returned, unused, to the destination. Note that this mechanism applies only for messages longer than one packet (about 1103 bits, including leader).

The message processing (reassembly of packets into messages, allocation of buffer space, detection of lost messages, etc.) requires the IMP to perform a certain amount of bookkeeping on the flow of messages between each pair of communicating Hosts. In order to keep the amount of required table space within manageable bounds, the following two restrictions are imposed.

- 1) The maximum number of messages which a Host is permitted to have "in transit" on any connection is eight. In other words, if a Host attempts to transmit nine messages on any connection, the interface will be blocked by the IMP during transmission of the ninth message until a RFNM (or *Incomplete Transmission*) is returned for the first message. However, this rule does not prohibit one Host from having eight messages in transit to Host "A", eight more in transit to Host "B", etc., simultaneously.
- 2) When a Host wishes to establish a new connection with another Host, both source and destination IMPs must acquire a block of table space from a pool of such blocks shared by all the Hosts local to each IMP. The source IMP must notify the destination of the need for the new connection, and the destination must reply with a confirmation that it has also acquired the table space. This action may result in a small additional delay before Host communication can begin. The pool will be sufficiently large to seldom interfere with a

pair of Hosts wishing to communicate. In no case will Hosts be prevented from communicating because of lack of these resources. In the event that the Hosts on an IMP desire to simultaneously communicate with so many other Hosts that the pool would be exhausted, the space in the pool is quickly multiplexed in time among all the desired Host/Host conversations so that none is stopped although all are possibly slowed.

Section 3.8 describes an optional mechanism available to Hosts that wish to keep interface blocking to a minimum.

### 3.2 Establishing and Breaking Host/IMP Communications

Each IMP and Host interface has its own hardware Ready indicator. The Ready indicator in the standard Host/IMP interface will be on whenever the IMP is powered on and both the IMP program and the IMP hardware are determined to be working properly. The Ready indicator in the special Host interface should be on whenever the Host is powered on, the hardware is working properly, and the Host's Network Control Program (NCP) is running. If the Host temporarily neglects communications with the IMP, the Host's hardware Ready indicator should not go off. An off indication should mean only that something is broken or that communications have been willfully cut off for an extended period (cable removed, power shut off, routine maintenance programs running, batch processing with no network program running etc.).

In addition to the Ready indicator, the standard interface has a flip-flop, called the *Error* flip-flop, which remembers a not-ready indication from the Host or the IMP. This flip-flop is used to detect any momentary off condition on either the

the Host's ready line or the IMP's ready line. The flip-flop is cleared by the IMP program each time the program enables (i.e., prepares to receive) a new input from the Host and is tested by the program when the input is completed. The input is discarded if the Error flip-flop is turned on.

To establish communication, a Host should simply send its message to the IMP. The operational IMP program will process any message transmitted from the Host. The Host must always send at least three *NOP* messages\* to the IMP whenever either the Host or the IMP Ready line is turned on, for the reasons described below.

One reason is that the Host-to-IMP *NOP* message contains information as to how much leader padding is to be contained in regular Host-to-IMP and IMP-to-Host messages. Also, until old-style leader formats (Appendix A) are no longer used, this *NOP* informs the IMP of the style of leader the Host is using.

Another reason is that in general, when the Host Ready indicator goes off, the IMP program will be either receiving or waiting (in an input command) to receive a message from the Host. Upon resumption of transmission by the Host, the IMP will unwittingly append the new information to the unfinished input. Upon completion of the message, the IMP program will note that the Error flip-flop is on and thus discard the entire message. To guarantee that a useful new message is not thereby discarded, the first message sent by the Host after its Ready indicator comes on should be a discardable *NOP* message. The special interface should have a similar Error flip-flop, and the Host's Network Control Program should be designed to use this flip-flop in a similar manner.

---

\* See Section 3.3



When the Host Ready indicator comes on, it will generally alternate a few times between on and off (due to relay contact bounce -- see Section 4.4) before setting solidly on. The Host should delay an appropriate period to permit its ready indicator to stabilize before starting output or preparing for input. Failure to do so may cause incorrect data to be taken from or sent to the IMP.

A Host may go down, thus halting network traffic to itself from other Hosts, in either of two ways: by turning off its ready indicator (hard down), or by failing to accept messages from the IMP (tardy down). In either case, the IMP will mark the Host as dead and see to it that any attempt to communicate with the Host results in a Destination Dead response.

The IMP program tests the Host Ready indicator (not the Error flip-flop) every half-second. If the program ever finds this ready indicator off, the Host will be marked dead (hard down) and the IMP will discard old messages for transmission to the Host and will set up 3 *NOP* messages followed by a type 10 message for transmission to the Host. Both the IMP and the Host must discard any *NOP* messages that are recognized as such. (A *NOP* message that is appended to an unfinished message may not be recognized, but it will be discarded as discussed above.)

The IMP follows the above procedures when the Host Ready indicator is off momentarily or for an extended period. The following steps are taken by the IMP when its own indicator has gone off.

- 1) The Error flip-flop is turned on. This action will cause the first incoming message from the Host to be discarded.

- 2) Old messages for transmission to the Host are discarded.
- 3) The IMP Ready indicator is turned on.
- 4) Sufficient *NOP* messages are placed on the output queue to the Host to cover the period of relay bounce and insure correct transmission of at least one *NOP*.
- 5) A Type 10 message is placed on the output queue to the Host.

The Host should employ a similar procedure whenever its own Ready indicator has gone off, except that old messages for transmission to the IMP need not necessarily be discarded.

In order to not tie up network resources for an inordinate amount of time, Hosts must be prepared to accept messages from the network promptly. In particular, any given message will be discarded if it resides on a queue to the Host for more than thirty seconds. (With the current IMP system, this requires that the Host must read its interface at the rate of about 1,500 bits/second, averaged across about twenty seconds.) If the Host does not meet this constraint, the IMP will:

- 1) Declare the Host to be "tardy down".
- 2) Discard all messages pending on the queues to the Host.
- 3) Momentarily drop its ready line (thus setting the error flip-flop). This is done because a component failure in the interface may have caused the handshaking procedure (see Sec. 4.2) to get out of step, which would have the same effect as the Host merely being tardy. "Flapping" the ready line insures that the interfaces are synchronized.
- 4) Place some *NOP*'s and a type 10 message on the queue to the Host.

The Host will be declared up the next time that it sends a message to the IMP or accepts a message from the IMP. The Host must send at least three NOP messages to the IMP if it is aware that it has been declared tardy, since the error flip-flop will cause the first Host-to-IMP message to be discarded. (alternatively, the Host could bring down its own ready line; the IMP would then proceed as though the Host were in a hard down, rather than continuing to treat the Host as though it were in a tardy down.)

If the Host has advance warning that it will be going down, it may use the *Host Going Down* message (see Section 3.3) to inform the IMP of its status (i.e., the reason for and duration of the down). Transmission of this message from the Host to the IMP will not cause the IMP to declare the Host down; the IMP will store the status information for use during the next Host down. When the Host comes up again, the status information stored in the IMP will be discarded.

The set of events described above is summarized in Table 3-1. Suggestions for Host use of the Ready indicators are contained in Appendix B.

EVENT: STATE	READY LINE UP	READY LINE DOWN	SEND OR RECEIVE HOST MESSAGE	RECEIVE HOST GOING DOWN	HOST TARDY
UP:		HARD DOWN		SET STATUS	TARDY DOWN
HARD DOWN:	UP, CLEAR STATUS		UP, CLEAR STATUS*	UP, SET STATUS*	
TARDY DOWN:		HARD DOWN	UP, CLEAR STATUS	UP, SET STATUS	

\*This can't, of course, happen without the Ready line being up, but the IMP might detect the input or output before detecting the change in Ready line status.

TABLE 3-1 TRANSITIONS BETWEEN HOST STATES

3.3 Host-to-IMP Leader Format

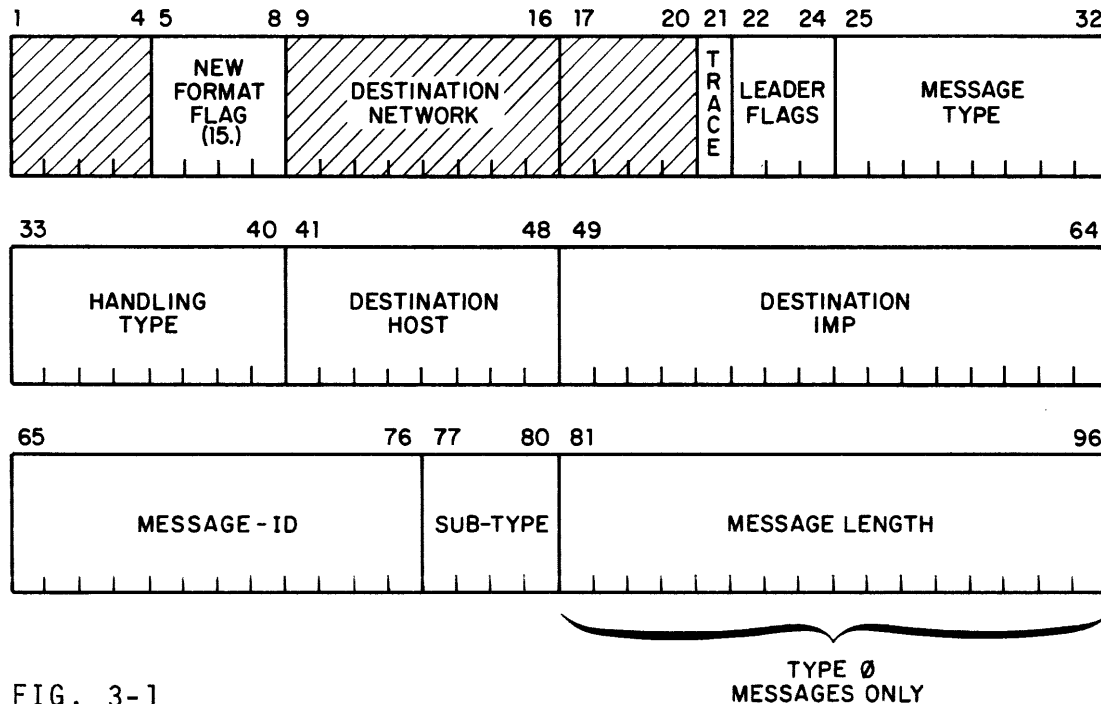


FIG. 3-1  
HOST-TO-IMP LEADER FORMAT

Bits 1 - 4 Unassigned -  
Must be zero.

Bits 5 - 8 New Format Flag -  
These bits are always set to the value 15. This permits the IMP to distinguish between new-style and old-style (Appendix A) leaders.

Bits 9 - 16 Destination Network -  
For future use, these bits must always be zero.

Bits 17 - 20 Unassigned -  
Must be zero.

**Bit 21 Trace -**

If equal to one, the message is designated for tracing as it proceeds through the network so that reports of this message's transit through the network may be sent to a trace destination (see Section 5.5).

**Bits 22 - 24 Leader Flags -**

Bits 23 and 24 are currently unassigned but are reserved for future network use and must be zero. Bit 22 is available as a destination Host flag, its meaning, if any, being assigned by that Host. The only Host with a pre-assigned meaning is the IMP Teletype Fake Host. If the bit is one, the message will be printed on the Teletype as a sequence of octal numbers, each representing one 16-bit IMP word. If equal to zero, then the message will be printed as a sequence of ASCII characters.\*

**Bits 25 - 32 Message Type -**

0. *Regular Message* - All Host-to-Host communication occurs via regular messages. Sub-types (bits 77-80):
  0. *Standard, Non-Refusable*. Interface blocking will occur if any resource needed to send the message is not immediately available.
  1. *Refusable* (see Section 3.8).\*\* Used to minimize the number of times the interface may be blocked. If any resource needed to send the message is not available, the message is discarded, and the Host is notified via a type 11, 12, or 13 Host-to-IMP control message. In the case of a type 12 (*Refused, will notify*) response, the IMP is committed to also sending a type 14 (*Ready*) when the resource does become available.

---

\* The IMP's internal ASCII character set is listed in Appendix E.

\*\* The non-blocking Host interface (see Section 3.8) is not yet implemented.

2. *Get Ready* (see Section 3.8).<sup>\*</sup> Similar to *Refusable* (above), except only the leader, rather than the full message, is sent in to the IMP. If all necessary resources are immediately available, the Host is notified via a Type 14 message.
3. *Uncontrolled* - (see Section 3.7). The IMP will perform no message-control functions for this type of message.
- 4 - 15. Unassigned.
1. *Error Without Message Identification* - The Host program detected an error in a previous IMP-to-Host message and had to assume that the leader was garbled. Sub-types:
  0. Host's error flip-flop was set during transmission of the message.
  1. Host received a message less than 80 bits.
  2. Host received a message of an unassigned type (3, 15-255).
  - 3 - 15. Unassigned.
2. *Host Going Down* - It is assumed that as the time for the Host to (voluntarily) go down approaches, the Host itself will send warning messages to its network users. Just before going down, the Host should send the Host-Going-Down message to its IMP. The Host should then (if it can) continue to accept messages from the IMP for a period of 5 or 10 seconds, to allow messages already in the network to reach it. The IMP will store the Host-Going-Down message and return it to any source

---

\* The non-blocking Host interface (see Section 3.8) is not yet implemented.

Host along with Destination (Host) Dead messages. The IMP will try to preserve this message over IMP reloads where appropriate. The NCC will be able to manually update the stored copy of this message in response to a phone call from the Host site in the event the Host is going to be down longer than it said or if it did not have time to give warning before going down.

Bits 65-76 (the message-id field) of the Host-Going-Down message give the time of the Host's coming back up, bit-coded as follows:

Bits 65-67: the day of the week the Host is coming back up. Monday is day 0 and Sunday is day 6.

Bits 68-72: the hour of the day, from hour 0 to hour 23, that the Host is coming back up.

Bits 73-76: the five minute interval, from 0 to 11, in the hour that the Host is coming back up.

All three of the above are to be specified in Universal Time (i.e., G.M.T.). The Host may indicate that it will be coming back up more than a week away by setting bits 65-76 all to ones. Setting all bits 65-75 to one and bit 76 to zero means it is unknown when the Host is coming back up.

Bits 77-80 (the sub-type field) of the Host-Going-Down message should be used by the Host to specify



the reason it is going down. These bits are coded as follows:

<u>Value</u>	<u>Meaning</u>
0-4	Reserved for IMP use
5	Scheduled P.M.
6	Scheduled Hardware Work
7	Scheduled Software Work
8	Emergency Restart
9	Power Outage
10	Software Breakpoint
11	Hardware Failure
12	Not scheduled up
13-15	Currently Unused

3. *Unassigned.*
4. *NOP* - The IMP will discard this message, which is intended for use during initialization of IMP/Host communication. Bits 77-80 (the sub-type field) contain the number of 16-bit words of padding (9 max.) that the Host wishes to send and receive on type 0 messages. This padding occurs immediately after the leader (starting at bit 97) and is provided as a convenience for Hosts for which the combined Host/IMP (IMP/Host) and Host/Host leaders would otherwise not be an integral number of memory words. A simple rule for the Host to follow is to send three *NOP* messages whenever the Host or the IMP has been down either voluntarily or involuntarily.
5. *Unassigned.*
6. *Unassigned.*
7. *Unassigned.*

8. *Error with Message Identification* - The Host detected an error in a previous IMP-to-Host message after the leader was correctly received; e.g., the message was too long, or the IMP Error flip-flop was set after transmission of the first packet of a multiple packet message but before the end of the message. A message of this type will have a leader whose assigned bits are identical to the assigned bits in the leader of the message in error except that the message type bits will be changed to have value 8.

9-255. *Unassigned.*

**Bits 33 - 40 Handling Type -**

This field is bit-coded to indicate the transmission characteristics of the connection desired by the Host.

Bit 33: *Priority* - Most messages should have this bit set to zero; messages with this bit set to one will be treated as priority messages (see Section 3.1).

Bits 34-37: Currently unassigned, must be zero.

Bits 38-40: *Maximum Message Size\** - The maximum size (in packets) of any message the Host expects to send on the connection ( $\# \text{packets} = (\# \text{bits in message} - 96) / 1008$ ). This number is expressed as (maximum # of packets - 1) and ranges from  $\emptyset$  (1 packet max) to 7 (8 packets max). It is to the advantage of the Host to specify this quantity as accurately as possible, since it enables the destination IMP to make the most efficient allocation of reassembly space. On the other hand, messages that must remain in strict sequence must all have the same handling type. Multiple connections between two Hosts, each with a different maximum message size, should be used only when there are large differences in the maxima and strict sequencing is not required. A message whose length exceeds the specified maximum will be discarded and type 9, subtype 1 will be returned to the Host.

---

\* Until this is implemented by the IMP, a default value of 7 (8 packets max) will be used.

**Bits 41 - 48 Destination Host -**

Identify the particular Host at an IMP site. Host numbers 252-255 are reserved for use by the IMP's "fake" Hosts (see Section 5).

**Bits 49 - 64 Destination IMP -**

Identify the IMP site

**Bits 65 - 76 Message-id -**

Host-specified identification supplied in all type 0 and 8 messages. Also used in type 2 (Host-Going-Down) message.

**Bits 77 - 80 Sub-type -**

Used by message types 0, 2, 4, and 8.

**Bits 80 - 96 Message Length -**

This field is used for type 0 messages only and specifies the length (in bits) of the message, exclusive of leader, leader padding and hardware padding. The only use that the IMP makes of this field is the *Get Ready* (Sub-type 2) message where it is used to determine if the message is single or multi-packet. If a zero length is given in a *Get Ready* message, a multi-packet length is assumed.

The following table shows which non-constant fields are used by each valid message type.

<u>Fields</u>	<u>Message Type</u>				
	0	1	2	4	8
Trace	x				
Leader Flags	x				
Message Type	x	x	x	x	x
Handling Type	x				x
Destination Host	x				x
Destination IMP	x				x
Message-id	x		x		x
Sub-type	x		x	x	x
Message Length	x				

3.4 IMP-to-Host Leader Format

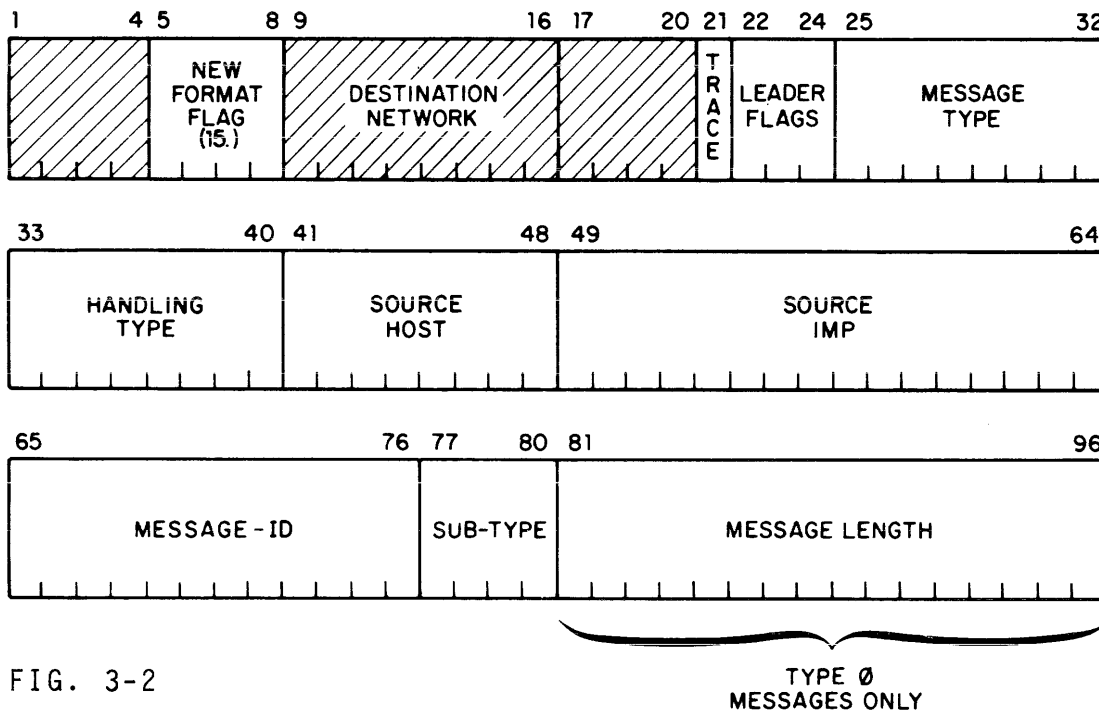


FIG. 3-2  
IMP-TO-HOST LEADER FORMAT

- Bits 1 - 4 Unassigned -  
Set to zero.
- Bits 5 - 8 New Format Flag -  
Set to 15.
- Bits 9 - 16 Source Network -  
Currently set to zero.
- Bits 17 - 20 Unassigned -  
Set to zero.
- Bit 21 Trace -  
If equal to one, source designated that message be traced (see Section 5.5). Used in type 0 messages only.
- Bits 22 - 24 Leader Flags -  
Bit 23 and 24 are currently unassigned and are set to zero. Bit 22 may be assigned a meaning by the destination Host, in which case it is used by the source Host to

signal some special meaning, e.g. octal printing for the Teletype Fake Host. Used in type 0 messages only.

Bits 25 - 32 Message type -

0. *Regular Message* - All Host-to-Host communication occurs via regular messages. The subtype field is the same as sent in the Host-to-IMP message; in particular a sub-type of 3 indicates an *uncontrolled* message (see Section 3.7).
1. *Error in Leader* - The IMP detected an error in a previous Host-to-IMP message and had to assume that the leader was garbled.

Sub-types:

0. IMP's Error flip-flop set during the first 96 bits of a message (see Section 3.2).
  1. IMP received a message of less than 32 bits.
  2. IMP received a message of an illegal Type.
2. *IMP Going Down* - The IMP will transmit this message to its Host before it voluntarily goes down. The Host should forward the information in the message to its users from the network ( and to its own users of the network).

Bits 65-80 of the message are coded as follows:

Bits 65-66: Why;

0. "last warning" or "panic restart": the IMP is going down in 30 seconds.

1. Scheduled hardware PM
2. Scheduled software reload
3. Emergency restart

Bits 27-70: How Soon; in 5 minute increments  
(zero implies immediately)

Bits 71-80: For How Long; in 5 minute increments  
(zero implies immediately)

3. *Unused.*
4. *NOP* - The Host should discard this message. It is used during initialization of IMP/Host communication. The Host and IMP fields will contain the local Host and IMP identification numbers, and the sub-type field will be zero. All other fields are unused.
5. *RFNM* - "Ready for Next Message". The named regular message was successfully delivered to the destination IMP, and the destination Host began to accept it. In addition, if the named message was longer than one packet (about 1103 bits including leader) space is reserved at the destination IMP for another transmission, but the space reservation will remain valid for only a short time (see Section 3.1). The subtype field will be 0 if the original message was non-refusable, and 1 if it was refusable.
6. *Dead Host Status* - Bits 65-76 (the message-id field) have the same meanings as bits 65-76 in the Host-to-



IMP type 2 (Host-Going-Down) message described in Section 3.3. Bits 77-80 (the sub-type field) have the following meanings:

<u>Value</u>	<u>Meaning</u>
0	Currently Unused
1	The destination Host is not communicating with the network -- it took its ready-line down without saying why.
2	The destination Host is not communicating with the network -- the Host was tardy in taking traffic from the network without saying why.
3	The destination Host does not exist to the knowledge of the NCC.
4	The IMP software is preventing communication with this Host; this usually indicates IMP software re-initialization at the destination.
5	The destination Host is down for scheduled P.M.
6	The destination Host is down for scheduled hardware work.
7	The destination Host is down for scheduled software work.
8	The destination Host is down for emergency restart.

- 9           The destination Host is down because of power outage.
- 10          The destination Host is stopped at a software breakpoint.
- 11          The destination Host is down because of a hardware failure.
- 12          The destination Host is not scheduled to be up.
- 13-14       Currently Unused.
- 15          The destination Host is in the process of coming up.

When the value of the sub-type field is 1, 2, 3, 4, or 15, the message-id field will have the "unknown" indication.

Bit 33 in this message will always be set to zero and Hosts receiving this message should discard (without reporting an error) type 6 messages with bit 33 set to 1. This will allow the later addition of similar status information on dead destination IMPs.

The Dead Host status message will be returned to the source Host shortly (immediately, if possible) after each Destination Host Dead (type 7) message. The Destination Host Dead message applies to a specific named message, although the information contained in the Destination Host Dead message should probably be

reported to all users connected to the dead Host. The Dead Host Status message does not apply to a specific named message and all users connected to the dead Host should be notified of the information contained in the Dead Host Status message.

7. *Destination Host or IMP Dead (or unknown)* - This message is sent in response to a message for a destination which the IMP cannot reach. The message to the "dead" destination is discarded.

Sub-types:

0. The destination IMP cannot be reached.
1. The destination Host is not up.
2. Currently unused.
3. Communication with the destination Host is administratively prohibited.
- 4-15. Currently unused.

8. *Error in Data* - The IMP's Error flip-flop was set after transmission of the leader of a message but before the end of the message.

9. *Incomplete Transmission* - The transmission of the named message was incomplete for some reason. An incomplete transmission message is similar to a RFNM, but is a failure indication rather than a success indication.

Sub-types:

0. Destination Host did not accept the message quickly enough.

1. Message was too long (in excess of maximum number of packets specified for connection).
  2. The message spent more than 15 sec. in transmission from the source Host to the IMP. This time is measured from the last bit of the leader through the last bit of the message.
  3. Message lost in the network due to IMP or circuit failures.
  4. Resources unavailable (see Section 3.1).
  5. Source IMP I/O failure during receipt of this message.
  - 6-15. Currently unused.
10. *Interface Reset* - The IMP's ready line has been dropped and pending output to the Host has been discarded (see Section 3.2). This probably indicates that the Host did not accept data from the IMP fast enough. Since dropping the ready line also sets the IMP's error flip-flop, the next message from the Host will be discarded and answered with a type 1 (sub-type 0) message. The sub-type field is unused.
11. *Refused, Try Again\** - A type 0, subtype 1 or 2 message was received from the Host but a certain "non-markable" resource needed for sending the message was not available. The message was discarded, and the Host should try to send it again when best able to do so.
- Sub-type:
0. IMP buffer was not available.
  1. Transmit block for connection was not available.
  - 2-15. Currently unused.

---

\* The non-blocking Host interface (see Section 3.8) is not yet implemented.

12. *Refused, Will Notify\** - A type 0, subtype 1 or 2 message was received from the Host but a certain "markable" resource needed for sending the message was not available. The message was discarded, and the Host will be notified via a type 14 (*Ready*) message when the resource becomes available.

Sub-types:

- 0-1. Currently unused.
- 2. Connection not available.
- 3. Reassembly space (for multi-packet message only) not available at destination.
- 4. Message number not available.
- 5. Transaction block for message not available.
- 6-15. Currently unused.

13. *Refused, Still Trying\** - A type 12 response is indicated, but a type 14 message has already been queued for some previous type 12 response. The message was discarded and no other response will be given. The subtype field is unused.

14. *Ready\** - The needed resource has become available for some previous type 0, subtype 1 or 2 message. The actual message is "named" by the message-id field.

15-255. *Unassigned.*

Messages of other than type 0 are sent to the Host prior to messages of type 0.

---

\* The non-blocking Host interface (see Section 3.8) is not yet implemented.

**Bits 33 - 40 Handling Type -**

The value assigned by the source Host, this field is used only in message types 0, 5-9, and 11-14.

**Bits 41 - 48 Source Host -**

See Source IMP, below.

**Bits 49 - 64 Source IMP -**

For type 0 messages, these fields identify the particular Host and IMP site that originated the message. For type 4 messages, these fields identify the local Host and IMP, and for message types 5-9 and 11-14, these fields identify the particular Host and IMP site to which a type 0 message was sent or will be sent. The fields are unused in all other message types.

**Bits 65 - 76 Message-id -**

For message types 0, 5, 7-9, and 11-14, this is the value assigned by the source Host to "name" the message. The field is also used by message types 2 and 6, and unused by all other message types.

**Bits 77 - 80 Sub-type -**

This field is used by message types 0-2, 4-7, 9, and 11-12.

**Bits 81 - 96 Message Length -**

This field is contained in type 0 messages only, and is the actual length in bits of the message (exclusive of leader, leader padding, and hardware padding) as computed by the

destination IMP using the end of message padding conventions. It should be noted that the IMP will not verify the length of the message if it is specified by the Host.

The following table shows which non-constant fields are used by each valid message type.

<u>Fields</u>	<u>Message Type</u>													
	Ø	1	2	4	5	6	7	8	9	10	11	12	13	14
Trace	x													
Leader Flags	x													
Message Type	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Handling Type	x				x	x	x	x	x		x	x	x	x
Source Host	x			x	x	x	x	x	x		x	x	x	x
Message-id	x	x		x	x	x	x	x	x		x	x	x	x
Sub-type	x	x	x	x	x	x	x	x	x		x	x		
Message Length	x													

### 3.5 Word Length Mismatch and Message Boundaries

There are two related aspects of word length mismatch: first, the obvious need for message formatting in order for Host computers having different word lengths to communicate; and, second, the need for locating the end of a message, since mismatched word lengths may lead to messages that end in the middle of words. The IMP design guarantees that between Hosts of identical word length, the natural word boundaries are preserved. Generally, however, reformatting is left to the Hosts. The problem of recognizing the end of a message at the receiving Host is solved in the following manner. As a message passes from the transmitting Host to its IMP, the standard Host/IMP interface appends a one to the bit string when it receives the end-of-message signal. This bit may fall in any position of an IMP word. The hardware then fills any remaining bits of this IMP word with trailing zeros. This process is called *IMP padding*. The transmitting Host may also specify the message length (in bits), which need not be the same as the physical length of the message.

As the message is serially shifted to the receiving Host, the last bit from the IMP will generally fall somewhere in the middle of the receiving Host's word. The remaining bits in this word are to be filled in with additional trailing zeroes from the Host's special interface hardware. (Note that a one is purposely omitted here.) Thus, the message appears in the receiving Host with a one immediately following the last data bit in the message and a string of zero or more trailing zeroes, that terminate at a Host word boundary, following the one. The last Host word in the received bit stream does not neces-



sarily contain the last data bit in the message; it may contain nothing but padding.

The maximum message that is shipped across the interface from the IMP to the destination Host contains 8160 bits (i.e., it includes the source IMP's padding). The destination Host's special interface unit will generally add padding of its own to round out the total number of bits going into the Host's memory to a multiple of the destination Host's word length. The destination Host should, therefore, be prepared to accept messages of at least 8160 bits. Not counting the destination Host's padding, messages of greater than 8160 bits in length should be discarded by the receiving Host.

It should be noted that Hosts may specify *leader padding* (see Section 3.3, NOP message). This padding is some integral number of 16-bit words which are transmitted and received immediately following the 96-bit leader of type 0 messages. This facility is designed to assist the Host in aligning some portion of the transmitted or received data with its own word boundaries. In particular, the Host may wish to make the sum of leader, leader padding, and other elements of Host-to-Host Leader equal to an integral number of Host words. This leader padding is not counted in the message length and exists only across the Host/IMP interface (i.e. not in the network).

### 3.6 Two Debugging Aids

It is occasionally useful, during debugging, for Host personnel to see the exact contents of messages which the IMP is receiving from a Host. The IMP provides two routines to aid this process; each routine can be enabled by the Network Control Center staff upon request.

The first such routine enables Host personnel to examine the exact bit pattern received by the IMP. When this routine is enabled, all messages sent from a particular local Host are returned to that Host. In the returned message the IMP and Host fields are those of the Host being tested; the rest of the message is unchanged. *Of course, IMP padding has been added.* This feature is intended to help find bits which are picked up or dropped by the Host/IMP interface.

The second debugging aid allows the IMP Teletype to simulate a Host. When this routine is enabled, the leaders of *all* messages to the IMP Teletype are printed in octal, one octal word per line, including RFNMs, incomplete transmission messages, etc. This feature is intended to help discover Host leader-generation errors, timing errors, and the like.

The following table should help the user to compose and to interpret leader codes.

TABLE 3-2  
LEADER WORDS

2nd leader word		
	4000	trace
	3400	leader flags
	2000	octal print
	377	message type (see Section 3.3 for Host-to-IMP, 3.4 for IMP-to-Host messages)
3rd leader word		
	177400	handling type
	100000	priority
	3400	max packets -1
	377	Host number
4th leader word		IMP number
5th leader word		
	177760	message-id
	17	sub-type
6th leader word		message length (type 0 only)

### 3.7 Uncontrolled Packets

For certain limited experiments which are being carried on using the network, it may be desirable for specified Hosts to be able to communicate without using the normal ordering and error-control mechanisms in the IMP. Communication of this type is possible using the Host-to-IMP and IMP-to-Host message type 0, sub-type 3. The rules governing IMP handling of these messages are:

- 1) Messages of type 0, subtype 3 are limited to the Host-to-IMP leader (96 bits) and not more than 991 additional data bits. Messages which exceed this length will be discarded without error notification.
- 2) At the destination IMP, these messages are put on the output queue for the destination Host in the order in which they are received; the messages are likely to be delivered in a different order from the order in which they were sent. Duplicate copies of some messages may be delivered.
- 3) There is no source-to-destination control of these messages. Lost messages will not be retransmitted. No RFNM, Incomplete Transmission, Destination Dead, etc., will be returned to the source.
- 4) The same bit-level error control applied to Regular messages will be applied to these messages passing between IMPs; i.e., type 0 subtype 3 messages are delivered with a very low probability of bit error.
- 5) If at any time there are insufficient resources in the network to handle one of these messages, it will be immediately discarded.

- 6) Use of these messages between two Hosts will not affect use of regular messages between these Hosts. Regular messages and subtype 3 messages may be inter-mixed over the Host/IMP interface.
- 7) Uncontrolled use of these messages will degrade the performance of the network for *all* users. Therefore, ability to use these messages will be regulated by the Network Control Center and will require prior arrangement for each experiment.

### 3.8 Non-Blocking Host Interface\*

As mentioned in Section 3.1, it is sometimes necessary for the source IMP to block the transmission of a message from the source Host. When this blocking occurs, all messages from that Host are held back, even though some of them might well be transmitted unimpeded if allowed into the IMP. Such might be the case, for example, if Host A is sending to Hosts B, C, and D, and the connection to Host B has eight messages in transit, the first (oldest) of which has become lost in the net. If a ninth message is sent to B, the interface will be blocked for the duration of the "incomplete" timeout (30-45 seconds), waiting for a message slot to become available on that connection. During this time, however, it would have been possible for A to send messages to C and D, had the interface not been blocked.

The non-blocking Host interface is a software mechanism which provides the source Host with the capability of keeping the interface unblocked for the vast majority of situations

---

\* Not yet implemented.

under which it might otherwise have become blocked. There will still be a few circumstances, associated with bandwidth and storage limitations of the source IMP, under which the interface may be blocked regardless of the mechanism used by the Host.

The non-blocking mechanism works by allowing the Host to flag some or all of its type 0 messages as "refusable", thus allowing the IMP to discard them if they would otherwise block the interface. In such a case, not only is the Host notified that the message was discarded, but it is also given guidance as to when the message should be retransmitted. In most cases, the particular resource that was missing is "markable", and the Host can be notified when the resource becomes available. In some cases, the resource is not "markable", and the Host must simply retransmit in accordance with its own requirements. The specific protocol for this mechanism is now described.

Host-to-IMP type 0 messages have four subtypes: *Non-refusable*, *Refusable*, *Get Ready*, and *Uncontrolled*. The *uncontrolled* subtype, described in Section 3.7, is never refused, and because it does not require most of the resources of "controlled" messages, is seldom blocked. The *Non-refusable* subtype is the standard mode of operation, which can cause interface blocking under the various circumstances described in Section 3.1. The *Refusable* subtype is treated identically to the *Non-refusable* subtype if blocking is not necessary. Under most circumstances where blocking would have been necessary, however, this message subtype is discarded, and one of three types of IMP-to-Host messages sent back to the Host. A *Refused, Try Again* (type 11) message indicates a "non-markable" resource was required, and the Host should merely retransmit at its convenience.

A *Refused, Will Notify* (type 12) message indicates a "markable" resource was required. The Host should wait for a fourth IMP-to-Host message type, *Ready* (type 14), before retransmitting. The IMP will send the *Ready* when the resource becomes available. A *Refused, Still Trying* (type 13) message indicates that the IMP has already given a *Refused, Will Notify* on that connection, but has not yet sent the *Ready* (it will only queue one such response at a time for any connection). There is no additional response after the *Refused, Still Trying*, and the Host should queue the message to be retransmitted after the one for which the *Ready* is expected.

The *Get Ready* subtype of the type 0 Host-to-IMP message is not a real message in the sense that it contains only the leader of an intended (future) message. It is provided so that the Host can determine whether or not a message could get through without blocking, without actually sending the data in the message through the interface. The possible responses to this subtype are identical to those of the *Refusable* subtype, except that in the normal case, when the *Refusable* message would have been transmitted to the destination without any interface blocking followed eventually by a RFNM, the IMP's response to the *Get Ready* is to send a *Ready* back to the Host.

Finally, it should be noted that a *Ready* does not guarantee that a retransmission will not be blocked, since no resources are actually reserved for some particular *message-id*, and in fact many are shared by all connections. The correct strategy for the Host willing to use the non-blocking feature is to make all messages *Refusable*, even when responding to a *Ready*.

#### 4. HARDWARE REQUIREMENTS AND DESCRIPTION

A local Host is connected to the IMP through a Host cable (provided with the IMP), which joins a *standard Host/IMP interface unit* in the IMP to a *special Host/IMP interface unit* in the Host. A distant Host is connected to an augmented standard Host/IMP interface through a cable provided by the Host. The structure of the standard Host/IMP interface, the IMP/Host handshaking procedure, the end-of-message indication, the Master Ready lines, and the signals on the Host cable are all described in detail below. A very distant Host is connected via communications circuits to a *modem interface unit* as described in Appendix F.

The special interface should be designed by the Host personnel to operate in conjunction with the standard Host/IMP interface or the augmented interface as the case may be. *We have not, however, attempted to specify the special Host/IMP interface in any detail.* We recommend that the special interface be modeled after the standard interface, and, in the remainder of this section, we assume that it will be. It should be noted that the special interface must be operated in a *full duplex* mode\*. A simplified schematic drawing of a special Host/IMP interface is included in Appendix B to assist Host personnel in the design of the special interface. The distant Host modification to the standard interface affects only the cable and the method of cable driving; it does not change the basic operation of the interface.

While the 316 and 516 type IMPs are available with either the local, the distant or the very distant Host connection, the Pluribus IMP will connect to the Host using the local or the very distant connection only.

---

\*Those few Hosts which originally implemented half duplex interfaces have had inordinate difficulties of various kinds. See, for example, Section 3.1.



#### 4.1 Structure of the Standard Host/IMP Interface

The standard Host/IMP interface is a full duplex bit-serial unit that is logically divided into a Host-to-IMP section and an IMP-to-Host section. Each section contains a 16-bit shift register (and control logic), one of which is for shifting bits to the Host and the other for receiving bits from the Host. A simplified picture of the Host/IMP interface is shown in Figure 4-1.

The technique of transferring information between the Host and the IMP is identical in each direction; we will, therefore, refer to the *sender* and the *receiver* without specifying the Host or IMP explicitly. In general, words are taken one by one from the sender's memory and transferred bit serially across the interface to the receiver, where they are reassembled into words of the appropriate (i.e., receiver's) length and stored into the receiver's memory. The transmission thus consists of a bit train containing no special indications of word boundaries but delayed occasionally while the sender fetches, or the receiver stores, a word. *The high-order bit of each word is transmitted first.*

Bit transfer is asynchronous, the transmission of each bit being controlled by a *Ready-For-Next-Bit, There's-Your-Bit* handshaking procedure. Each bit is transferred only when both sender and receiver indicate preparedness. This permits either the sender or the receiver to hold up the transmission between any two bits in order to take as much time as necessary to get a new word from memory, to tuck an assembled word into memory, or to activate an interrupt routine that sets up new input or output buffers. Neither the sender nor the receiver should expect transmission to take place at a pre-determined bit rate and each must be able to accept arbitrary delays introduced by the other at any point in the bit train.

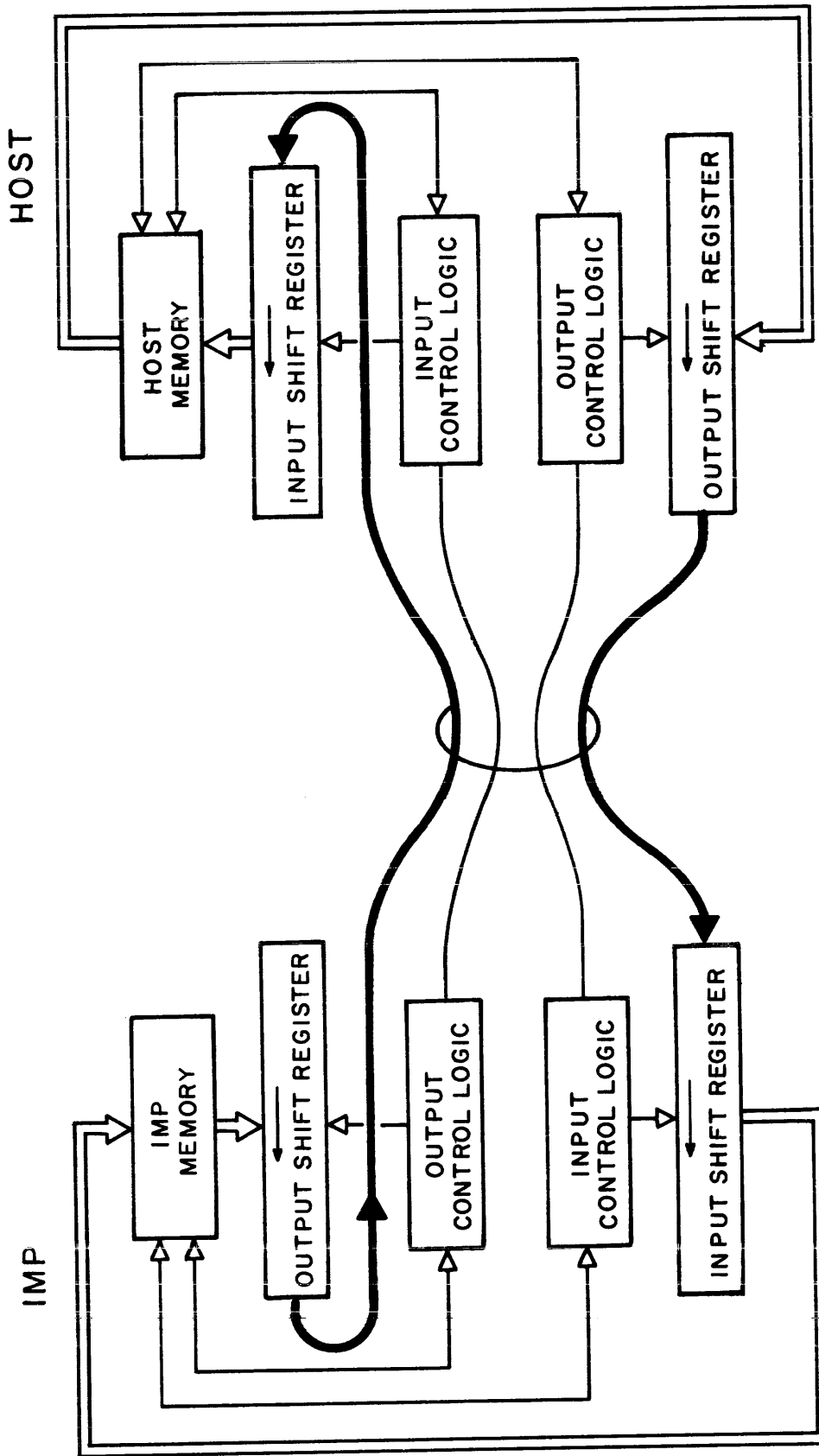


Figure 4-1. Simplified Illustration of Host/IMP Interface

The design of an asynchronous interface was selected for two reasons: first, because of the inherently asynchronous nature of the process by which words of one length are fetched from one machine and reformed into words of another length and stored in another machine; and, secondly, because such a design allows a variety of special Host/IMP interfaces to be designed independent of stringent timing specifications that may be difficult or impossible for certain Hosts to meet.

#### 4.2 IMP/Host Handshaking

Figure 4-2 shows a much simplified version of the control logic for the bit-by-bit handshaking procedure. When PG #1 (Pulse Generator) fires, it turns off the Bit Available flip-flop and a new data bit is shifted into position by the sender. The Bit Available flip-flop is then turned back on, and, if (or when) the receiver is ready to receive a bit, a *There's-Your-Bit* signal is sent to him. This triggers PG #2\*, which shifts in the new bit and shuts off the *Ready-For-Next-Bit* flip-flop. When this indicator goes off, the sender knows that the bit has been taken by the receiver. PG #1 then fires and shuts off the Bit Available flip-flop in preparation for getting the next bit ready for transmission. After the receiver has taken in the bit and is ready to accept a new one, it turns the Ready-For-Next-Bit flip-flop back on. The cycle then repeats.

---

\*The *on* (↑) transition of *There's-Your-Bit* triggers PG #2.  
The *off* (↓) transition of *Ready-For-Next-Bit* triggers PG #1.

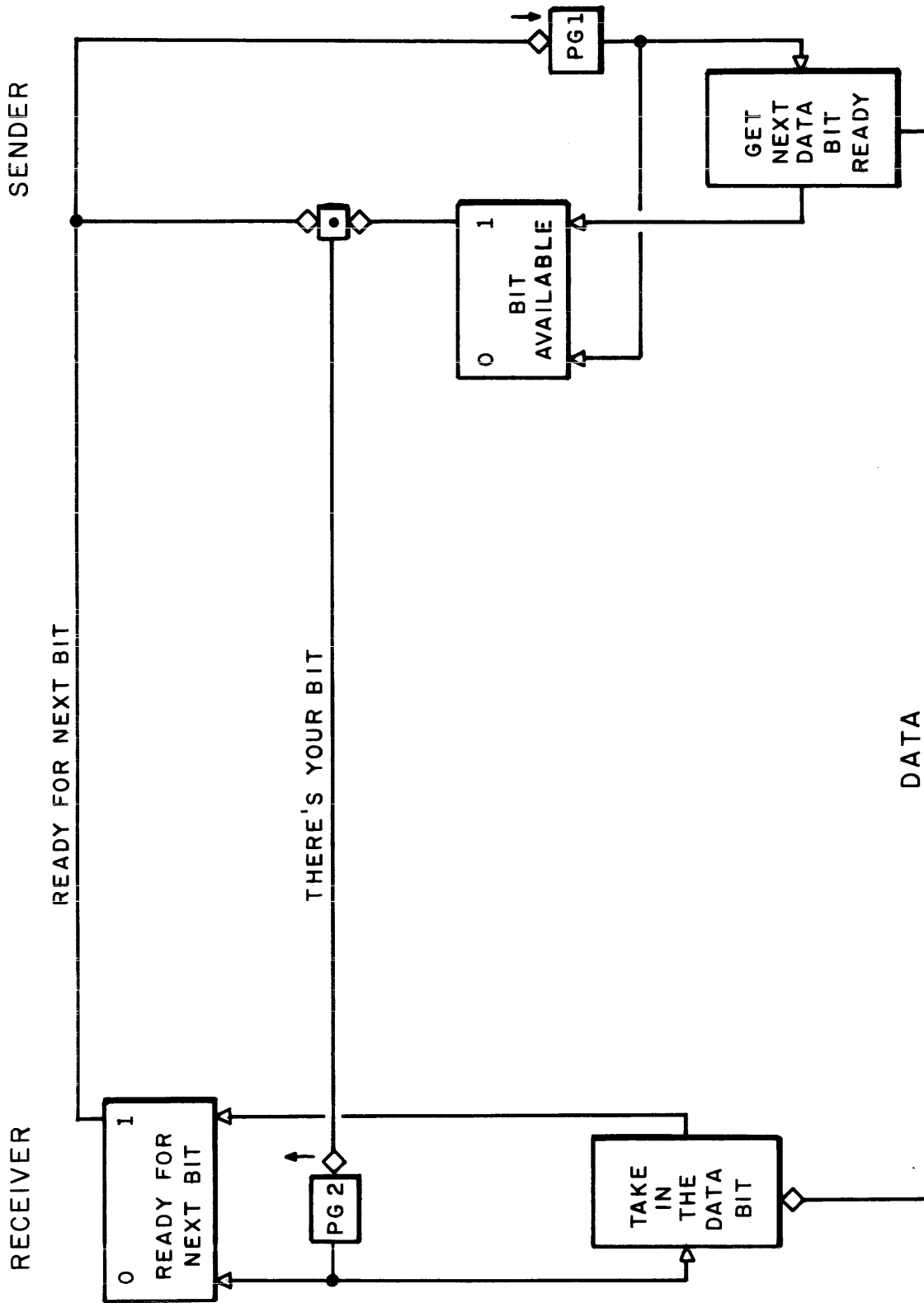


FIG. 4-2 SIMPLIFIED CONTROL LOGIC FOR HOST/IMP HANDSHAKING

Each time the sender is notified that a bit has been accepted (by the *off* transition of Ready-For-Next-Bit), a word length counter is checked to see whether a new word must be fetched from memory. Similarly when a bit is accepted at the receiver, it may be necessary to tuck an assembled word into the memory before registering readiness to receive another bit. In addition to these obvious requirements, the simplified picture contains critical race problems\*, which have been carefully resolved in the IMP's interface and must be similarly resolved in the Host's special interface.

The receiver may choose either of two methods of handshaking, a two-way or a four-way handshake. In the four-way handshake, the receiver awaits the dropping of There's-Your-Bit before raising Ready-For-Next-Bit. A full cycle of the four-way handshake works as follows: The sender readies the next data bit and the There's-Your-Bit signal is sent to the receiver (1<sup>st</sup> cable transit). The receiver takes in the bit and notifies the sender by dropping Ready-For-Next-Bit (2<sup>nd</sup> cable transit). The sender responds by dropping the There's-Your-Bit signal (3<sup>rd</sup> cable transit) and after the receiver has noted this, the Ready-For-Next-Bit signal can be turned back on (4<sup>th</sup> cable transit), registering preparedness for a new bit.

---

\*For example, the race in shutting off the Ready-For-Next-Bit flip-flop.

The two-way handshake works as follows: The sender reads the next data bit and the There's-Your-Bit signal is sent to the receiver (1<sup>st</sup> cable transit). The receiver takes in the bit and notifies the sender by dropping Ready-For-Next-Bit (2<sup>nd</sup> cable transit). Instead of waiting for this signal to propagate to the sender and the resultant dropping of There's-Your-Bit to return, the receiver holds Ready-For-Next-Bit off for a brief period and then turns it back on.

This method has two dangers that must be considered, both arising from the situation where Ready-For-Next-Bit is off for too short a time.

- 1) If Ready-For-Next-Bit is off for too short a period, the sender may never note that it went off and he will continue to wait for the bit to be taken. The IMP itself requires that the signal be off at the IMP end of the cable for at least 50 nanoseconds for local Hosts and for at least 1  $\mu$ sec for distant Hosts.\*
- 2) If the receiver turns Ready-For-Next-Bit back on before the There's-Your-Bit signal has been observed to go off at the receiver's end of the cable, then the receiver may mistakenly believe the new bit is ready to be taken in. This problem is avoided if the receiver

---

\*The 316 and 516 IMPs, in fact, always use the two-way procedure. They do not wait for the There's-Your-Host-Bit signal to go off but instead guarantee to hold the Ready-For-Next-Host-Bit signal off for at least 1  $\mu$ sec. The Pluribus IMP uses the four-way handshake.

maintains a *There's-Your-Next-Bit* flip-flop which is turned off when *Ready-For-Next-Bit* is turned off and is turned on only by the leading edge (*on* transition) of the *There's-Your-Bit* signal from the sender.

For local Hosts, where the cable delays are insignificant, either handshake procedure may be used. For distant Hosts, where cable delays may be significant, the two-way handshake procedure is recommended, in order to avoid placing an unnecessary restriction on the maximum bit rate.

The IMP introduces some deliberate delays into this control loop, both as a sender and as a receiver. Specifically, as a sender, the IMP introduces approximately 10  $\mu$ sec of delay\* between the time that the Host indicates that it has taken one bit and the time that the next bit is made available. As a receiver, the IMP shifts in the data bit and turns off the *Ready-For-Next-Bit* signal shortly after the *There's-Your-Bit* signal comes on. However, *Ready-For-Next-Bit* will not be turned on again until about 10  $\mu$ sec\* after *There's-Your-Bit* comes on. By introducing these deliberate delays, the IMP slows down the rate of information flow on the Host channels, thereby controlling

---

\*These are minimum times assuming no IMP memory reference is required. Where a memory fetch or store is required, the times will be increased by at least 4  $\mu$ sec.

the maximum amount of IMP memory bandwidth that the channels can consume. This control is essential to avoid usurping bandwidth required for the store-and-forward functioning of the IMP.

Because of the loop nature of the handshake procedure, the Host can also introduce delays. However, knowing that the IMP will limit the data rate, *the Host should, in general, not introduce further deliberate delays of its own.* The delays we have mentioned are adjustable and can be tuned so that the interface operates at much higher speeds. At the time of installation of a new IMP, the standard interface will be set to run the Host channels at a 10- $\mu$ sec-per-bit rate. Once the IMP is connected to the Host, both the input and the output channels will normally be tuned to operate at a maximum rate of 100 kilobits/second, thereby lumping together the delays in the IMP interface and the Host special interface.\*

---

\*Since the IMP as a receiver holds the Ready-For-Next-Bit signal off for 10  $\mu$ sec, there does not appear to be any real need for the Host to have a Bit-Available flip-flop go on and off on a per-bit basis. The There's-Your-Bit line will go off when Ready-For-Next-Bit goes off. However, the 10  $\mu$ sec delay is subject to shrinkage; therefore, the Host should not rely on this delay to provide time for the next bit to arrive — even if getting the bit amounts only to moving a shift register over one place.



### 4.3 End-of-Message Indication

A Host indicates the end of its message to the IMP by presenting a *Last-Host-Bit* signal to the IMP together with the last bit. This signal will generally occur somewhere in the middle of an IMP word, i.e., with the input shift register in the standard interface only partially loaded. Additional padding bits will then be shifted into the register, namely a single one followed by enough zeroes (perhaps none) to fill up the register. These additional bits are appended at the end of a Host message by the hardware in the input section of the standard interface. If the last data bit happens to just fill the shift register, an additional IMP word consisting of a single one followed by fifteen zeroes will be appended to the message. Alternatively, if the single one happens to just fill the shift register, the IMP padding will contain only this single one. At the destination, the IMP will indicate the end of the message to its Host by presenting a *Last-IMP-Bit* signal to the Host together with the last bit of the IMP padding. In general, this signal will occur somewhere in the middle of a Host word, i.e., with the input shift register in the special interface only partially loaded. The Host must shift enough additional zeroes (perhaps none) into this register to fill up the register.

### 4.4 Master Ready Lines

Whenever the IMP is ready, it holds closed a relay contact that connects two wires (the *IMP Master Ready* and the *IMP Ready Test* lines) in the Host cable. Figure 4-3 illustrates how the

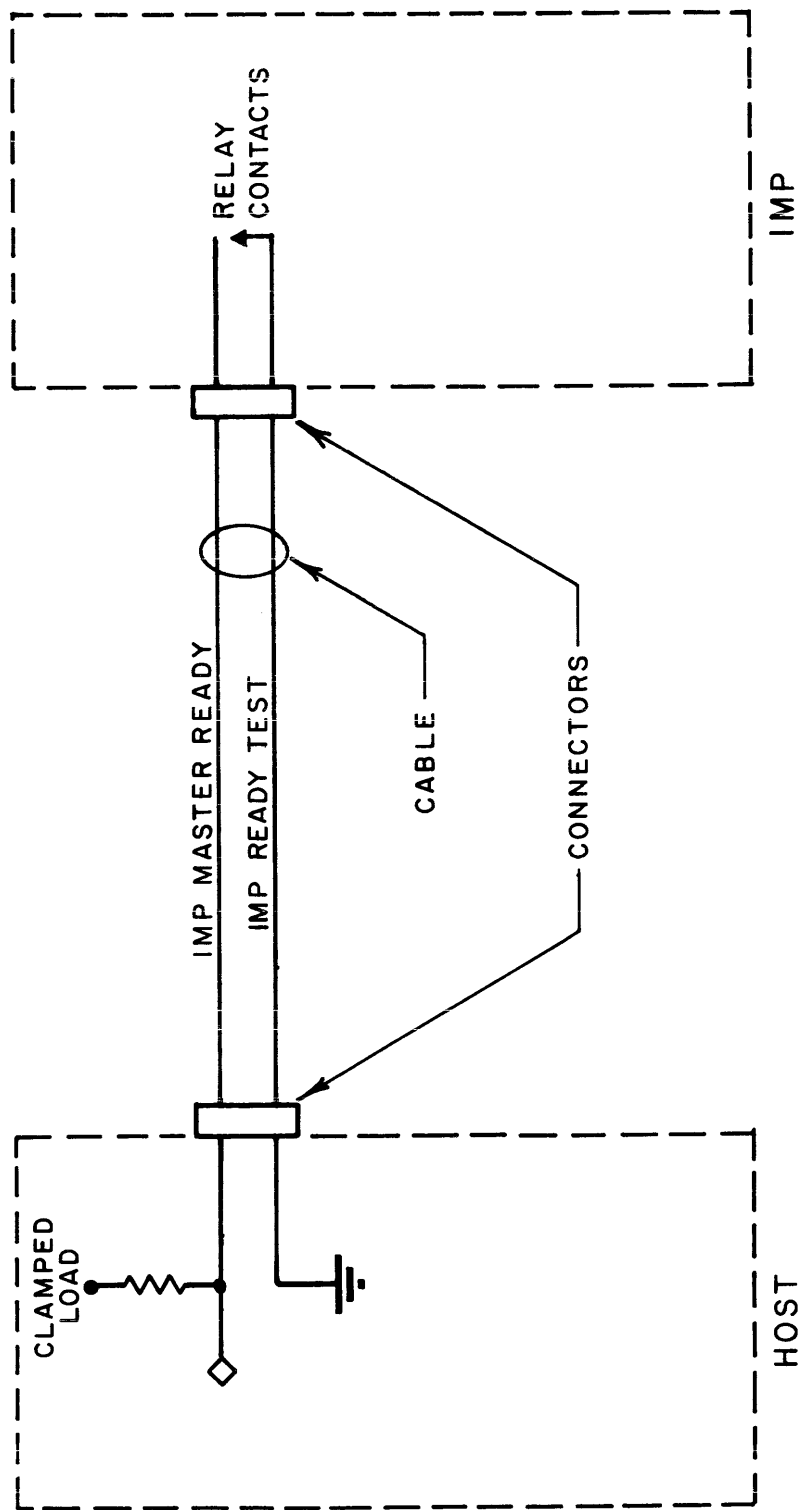


FIG. 4-3 IMP READY TEST AND IMP MASTER READY LINES

Host can employ this contact closure to ground a clamped logic line whose polarity indicates readiness of the IMP\*. Note that, if the cable is removed at either end, the IMP appears to the Host not to be ready. The relay contacts are a Normally-Open pair and thus, if the IMP's power goes off, the line indicates "not ready".

The relay closure is also controlled by the IMP program. If the IMP detects a serious program failure, it initiates an automatic recovery procedure. This same procedure is also initiated by the Network Control Center under certain conditions. Execution of the recovery procedure causes the relay to open; successful recovery will eventually cause the relay to close again.

Similarly, each Host must provide for its IMP a set of contacts, which open when Host power goes off or whenever the Host does not wish to communicate with the rest of the network for an *extended* period.\*\* The IMP will use this contact, in the specific manner suggested above, to pass a signal ground around to itself for testing Host readiness.

---

\*The choice of ground as the interrogation level is obviously arbitrary, and the Host may use any reasonable arrangement.

\*\*See Section 3.2 for a more complete discussion of the alternatives available to the Host for voluntarily stopping communication with the rest of the network.

The special Host interface should gate all incoming signals with the signal (or its inverse) on the IMP Master Ready line in order to avoid responding to meaningless transitions. Since the Master Ready signal passes through a relay, it will, in general, show contact bounce. When the IMP becomes ready (i.e., closes its relay), it executes a programmed delay before the There's-Your-IMP-Bit line becomes true. This delay covers the contact bounce period and thus the Host need not worry about bounce on the gated versions of this signal. (The IMP also executes a programmed delay before beginning a *new* input operation. Since there may however be errors in the current transmission to the IMP, the Host should always send at least one NOP message after seeing the IMP in a not-Ready state.)

The Host should provide similar protection by not permitting the There's-Your-Host-Bit signal to become true until after its relay contacts have solidly finished closing.

#### 4.5 Host Cable Connections

Following is a summary of the signals on the Host cable:

1. *IMP Master Ready* - The return for the IMP Ready Test signal through the IMP's relay contact.
2. *IMP Ready Test* - The test signal sent to the IMP to interrogate its ready status through the IMP's relay contacts. No more than 100 mamp should flow in this wire and the IMP Master Ready wire.
3. *Host Master Ready* - The return for the Host-Ready-Test signal through the Host's relay contact.

4. *Host Ready Test* — The ground signal sent to the Host to interrogate its ready status through the Host's relay contacts. No more than 100 mamp should flow in this wire and the Host Master Ready wire.
5. *Host-to-IMP Data Line* — The data from the Host should be changed for successive bits only after the IMP's Ready-For-Next-Host-Bit signal goes off indicating that the previous bit has been accepted.
6. *There's-Your-Host-Bit* — This signal should be presented to the IMP by the Host as soon as the Host has a bit available to transmit *and* the IMP is indicating that it is Ready For Next Host Bit. When the Ready-For-Next-Host-Bit signal goes off, the There's-Your-Host-Bit signal should be removed. This must be done in two ways, as shown in Figure 4-2 — first by the AND gate between the Bit Available flip-flop and the Ready-For-Next-Bit signal, and second by immediately turning off the Bit Available flip-flop itself.\*

---

\*At first glance this seems like duplication. However, when the next bit becomes available, the Bit Available flip-flop will be turned back on and yet the There's-Your-Bit signal should not be sent unless the Ready-For-Next-Bit signal is on. Thus, the need for the AND gate. Shutting off Bit Available is required to avoid confusing the receiver with an old bit when a new Ready-For-Next-Bit signal comes on.

7. *Ready-For-Next-Host-Bit* - This signal will be presented to the Host whenever the IMP is waiting for a transmission by the Host. Each time that the Host gives the IMP a bit (via *There's-Your-Host-Bit*), the *Ready-For-Next-Host-Bit* will go off after the bit has been taken in. It will go back on again within 10  $\mu$ sec unless a memory access is required (once every 16 bits). A much longer off period will result when an IMP memory buffer region fills, and an interrupt service routine must operate before the IMP is ready for another bit.
8. *Last-Host-Bit* - When the Host transmits the last bit of a message, the *Last-Host-Bit* signal should be sent to the IMP in conjunction with the *There's-Your-Host-Bit* signal. Specifically, the *Last-Host-Bit* signal must come on no later than the *There's-Your-Host-Bit* signal comes on, and should remain on at least until *Ready-For-Next-Host-Bit* goes off. The IMP will pad the message with a one followed by enough zeroes (perhaps none) to fill the current IMP word.
9. *IMP-to-Host Data Line* - The data for the Host will be changed for successive bits only after the Host's *Ready-For-Next-IMP-Bit* signal goes off, indicating that the previous bit has been accepted.
10. *There's-Your-IMP-Bit* - This signal will be presented to the Host by the IMP as soon as the IMP has a bit available to transmit and the Host presents the

Ready-For-Next-IMP-Bit signal. When the Ready-For-Next-IMP-Bit goes off, the There's-Your-IMP-Bit signal will be removed. It will not be renewed until a new Ready-For-Next-IMP-Bit signal arrives.

11. *Ready-For-Next-IMP-Bit* - This signal should be presented to the IMP whenever the Host is ready to receive information. Each time that the IMP gives the Host a bit (via the There's-Your-IMP-Bit line), the Ready-For-Next-IMP-Bit signal should go off after the bit has been taken in. This notifies the IMP that the bit has been taken and that a new bit can be moved into position and made available. Ready-For-Next-IMP-Bit should be off for at least 50 nanoseconds (1  $\mu$ sec for distant Hosts) as seen at the IMP before it goes back on again. It may, of course, be off for as long as it takes the Host to ready itself to receive the next bit.
12. *Last-IMP-Bit* - When the IMP transmits the last bit of the source IMP's padding, the Last-IMP-Bit signal will be sent to the Host in conjunction with the There's-Your-IMP-Bit signal. Specifically, the Last-IMP-Bit signal will come on no later than the There's-Your-IMP-Bit signal. Last-IMP-Bit will stay on for some *arbitrary* short time after There's-Your-IMP-Bit goes off. The Host's interface must not interrogate this line after the Ready-For-Next-IMP-Bit signal has been turned off. The Host's special interface should round out the last memory word with zeroes, as required.

The asynchronous (i.e., sequential) nature of the interface causes stress to be laid on the *order* in which operations occur rather than on their *timing*. Minimum on or off times for the circuits in the IMP are 50 nanoseconds for a local Host and 1  $\mu$ sec for a distant Host. Thus, for example, the Host's Ready-For-Next-IMP-Bit line must be visibly down at the IMP for at least this length of time before it is brought back up even if the Host takes the bit more quickly than that. Similarly, the Host's Bit Available flip-flop must appear off to the IMP (via the There's-Your-Host-Bit line) for at least 50 nanoseconds for local Hosts and 1  $\mu$ sec for distant Hosts. We expect that, in general, much slower circuitry will be used, so that these minimum times will present no problems. The IMP delays only a very short time from the arrival of There's-Your-Host-Bit before taking the Host's data bit or checking the Last-Host-Bit line. However, for IMP-to-Host transmission, the IMP will guarantee that the IMP data bit is on the line and the Last-IMP-Bit level is correct at least 500 nanoseconds before turning on the There's-Your-IMP-Bit signal. Thus, skews of under 500 nanoseconds in the signals for Host-to-IMP transmission at the IMP end of the cable will be removed by the IMP, but skews for IMP-to-Host transmission must be removed by the Host.

#### 4.5.1 Connection to a Local Host

The nominal asserted level for all logic lines (Data, Ready-For-Next-Bit, There's-Your-Bit, Last Bit) is +5 volts and the unasserted level is ground (these are with respect to the IMP signal ground). The driving and receiving circuits and specifications are shown in Appendix C.



The Host cable supplied with the 516 IMP and the Pluribus IMP is 30 feet long and contains 12 RG 174/U coaxial conductors with grounded shields. *Host personnel must provide an appropriate connector for the Host end of the cable.* The shield of each conductor is connected to signal ground at the IMP connector. Each cable is labelled with the IMP connector pin number corresponding to the center lead of the coaxial conductor. These wires are assigned as indicated in Figure 4-4 for the 516 IMP; that is, the number in the figure corresponds to the number on the label attached to each coaxial conductor. Pluribus IMP connections are pictured in Figure 4-6. All shields should be connected to signal ground in the Host. DC amplifiers are used for line driving and, by this means, we expect to couple the signal ground systems as tightly as possible.

The Host cable supplied with the 316 IMP is 30 feet long and contains 32 twisted pairs. The cable is terminated at the IMP end with a paddle card which plugs directly into the 316 Host interface. Each pair of the cable consists of a colored wire and a black wire numbered with the pin number of the paddle card to which the colored wire is connected. All black wires connect to the paddle card signal ground. *Host personnel must provide an appropriate connector for the Host end of the cable.* The wires are assigned as in Figure 4-5. All twisted pair grounds should be connected to signal ground in the Host. DC amplifiers are used for line drivers and the signal ground systems of the Host and IMP should be as highly coupled as possible.

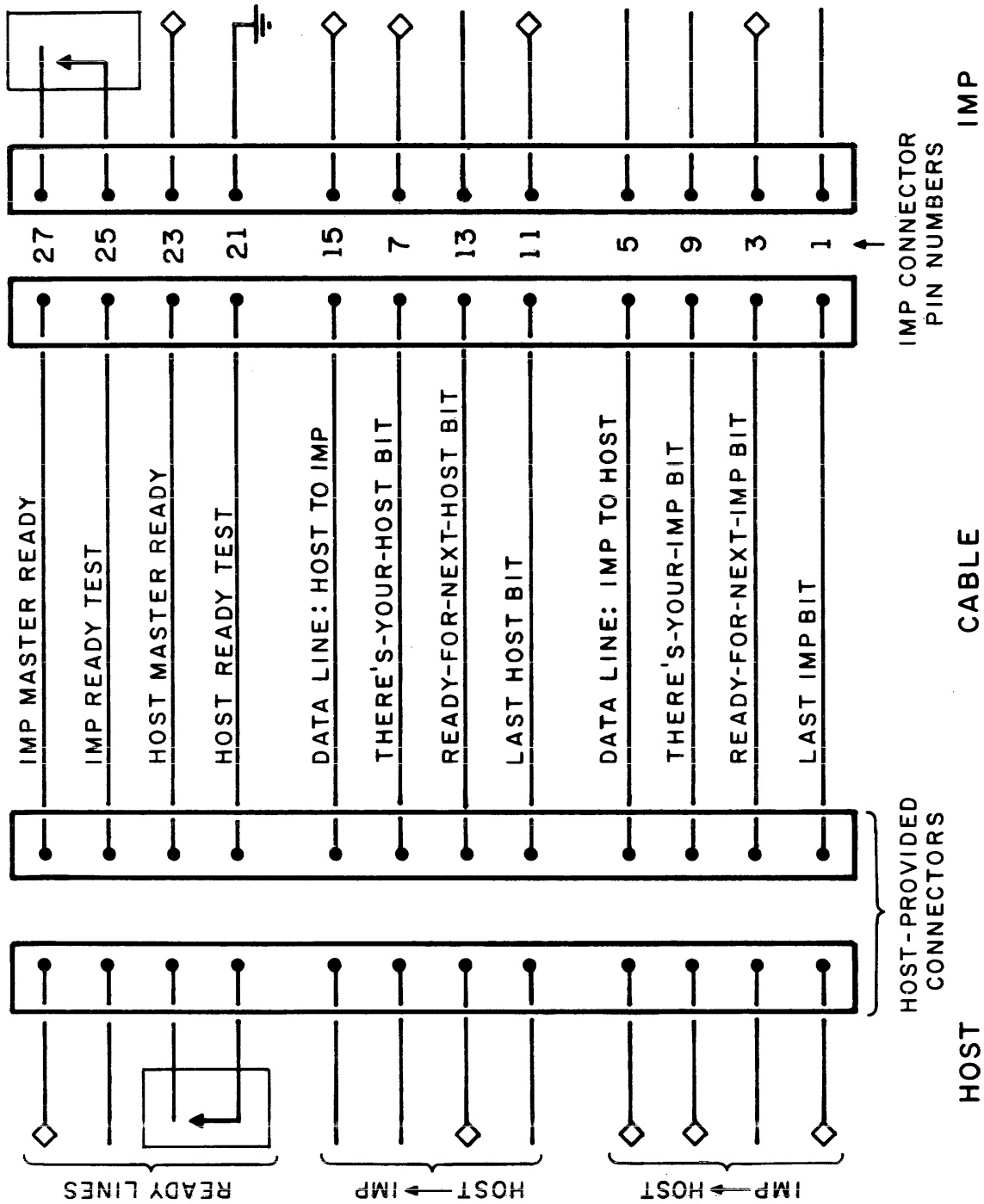


FIG. 4-4 516 HOST CABLE SIGNALS (LOCAL HOST)

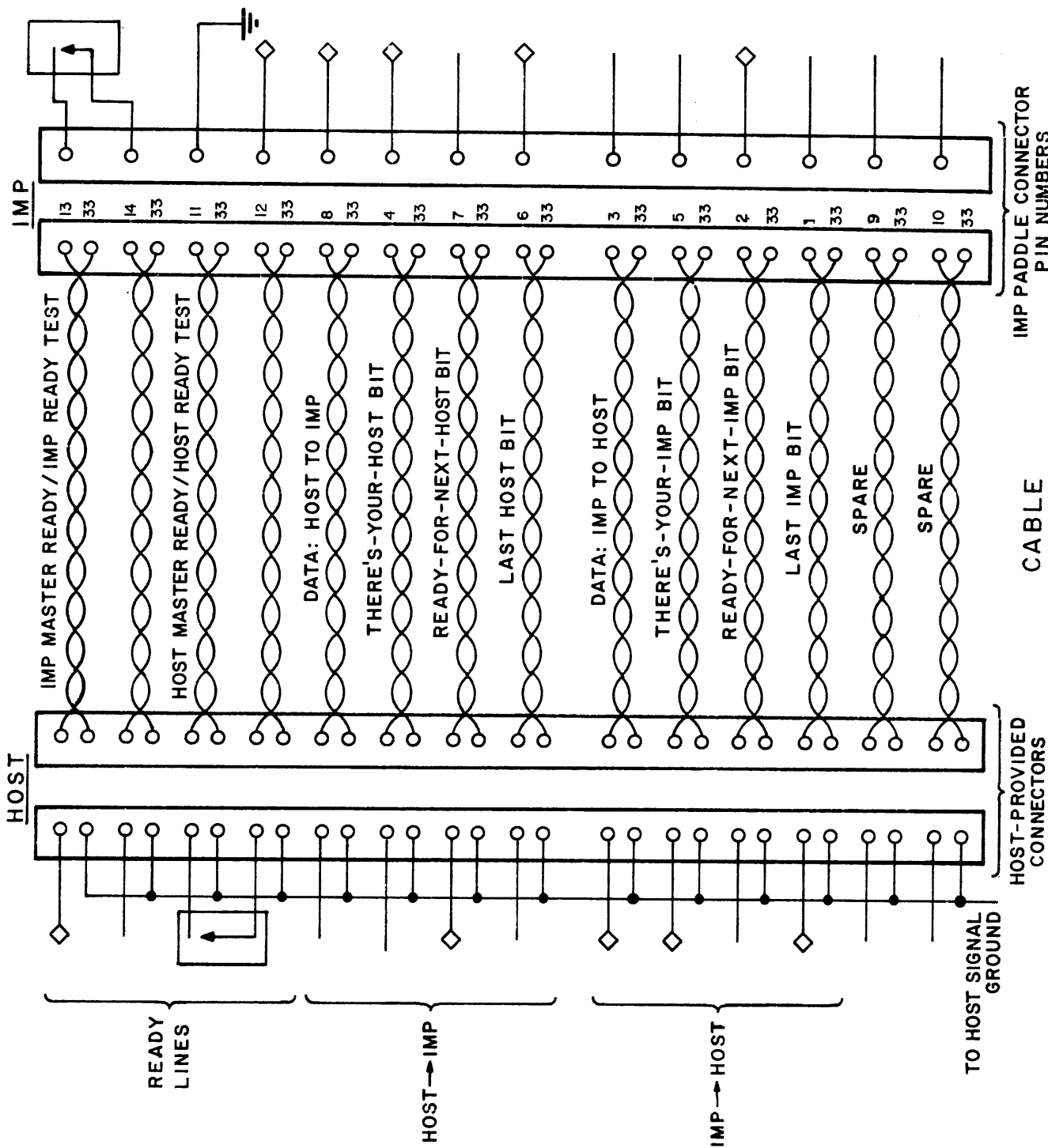


FIG. 4-5 316 HOST CABLE SIGNALS (LOCAL HOST)

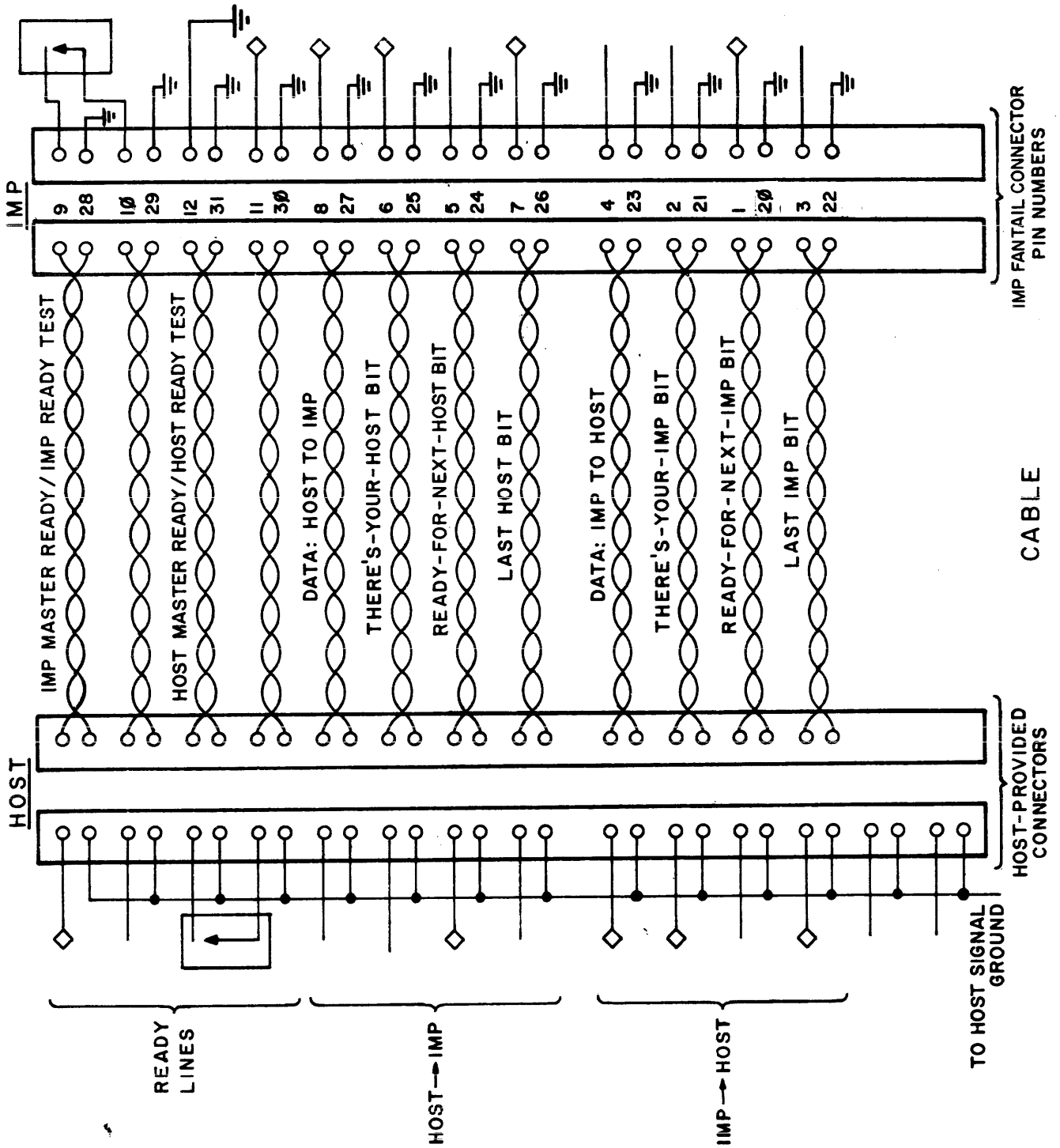


FIG. 4-6 PLURIBUS IMP HOST CABLE SIGNALS (LOCAL HOST)

#### 4.5.2 Connection to a Distant Host

Connection to a distant Host necessitates the use of balanced lines and requires that a Host pay careful attention to differentials in ground potential. The distant Host's special interface must provide balanced drivers and receivers. *Ground isolation is provided by the IMP.*

The Host must supply a shielded cable containing multiple twisted pairs of #20 (or heavier) gauge wire. The characteristic impedance ( $Z_0$ ) must be approximately 120 ohms. The wires may be either individually shielded or may have a single shield covering all pairs. The shield is used to carry the Host's ground reference and should have very low resistance. There must be at least 10 pairs in the cable, and we strongly recommend that at least two spare pairs be carried (see Figure 4-7). A suitable cable is Direct Burial Cable, REA Specification PE-23, 19AWG conductors, 12 pairs.

At the IMP side the cable must be terminated in a MS24266R18B31PN (Amphenol 48-16R18-31P) plug with a MS27291-5 clamp and MS24254-20P contacts. Pair and Pin assignments are shown in Figure 4-7. Note that the cable shield(s) should be connected to pin 31 and *not* to the connector shell.

The cable shields should be very solidly connected to the Host's signal ground, which should be connected to the third-wire power ground at the Host computer. DC isolation is done at the IMP end of the cable to prevent significant currents from

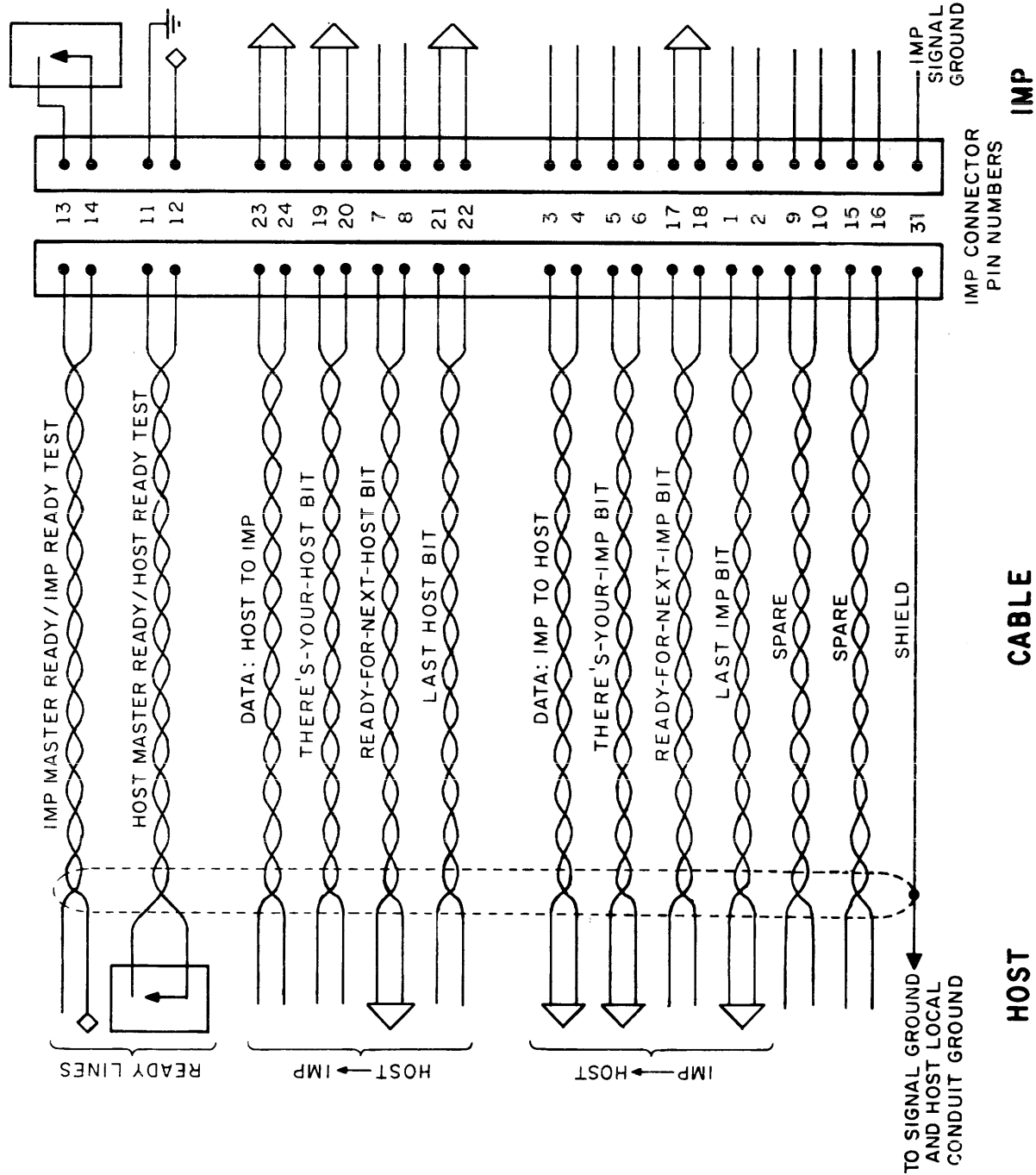


FIG. 4-7 HOST CABLE SIGNALS (DISTANT HOST)

flowing through the shields. This isolation is accomplished by transformer coupling the signals. All signals from the distant Host must therefore have transition times of less than 100 nanoseconds, and must remain in each state for at least 1  $\mu$ sec between transitions.

The logic signals on the pairs of the cable (Data, Ready-For-Next-Bit, There's-Your-Bit, Last-Bit) are balanced voltage signals with each side terminated at the driver in 62 ohms to ground. Thus, the terminating impedance is 124 ohms, and matches the cable impedance. The asserted logic signal drives the odd-numbered connector pin of each pair to +0.5 volts, and the other pin to -0.5 volts, producing a differential signal of 1.0 volt. The unasserted signal switches the polarity of this pair. There is no voltage drop across the cable since the receiver is unterminated. This produces a step reflection at the receiver which is absorbed at the transmitter. At the Host end, the transmitters should terminate the cable in 120 ohms across each signal pair.

Standard 6-volt IMP logic signals are converted to differential signals by the line drivers and from differential signals to 6-volt logical signals by the receivers. Drawings for the drivers and receivers used in the IMP are shown in Appendix D.

The Host should provide drivers and receivers similar to those used in the IMP. Use of these exact circuits is acceptable, as is use of any other circuits capable of driving and receiving a differential signal of 1.0 volts centered around ground. Care should be taken to preserve proper signal polarity in the cable.

## 5. IMP BACKGROUND PROGRAMS

In this section we describe the formats and procedures that a Host should use to communicate with one or more of the background programs that run with the operational IMP program. These programs are called TTY, DEBUG, PARAMETER-CHANGE, DISCARD, TRACE and STATISTICS. The procedure for communicating with an IMP Teletype is described in the first part of the section, and a typical Host will ordinarily have no reason to use any program other than TTY. The other sections, therefore, can generally be omitted from study. However, certain Hosts, such as the Network Measurement Center and the Network Control Center, will need to know the information contained in the latter sections.

Messages to or from an IMP background program are called IMP messages and are distinguished from regular Host messages by the Host field of the leader -- the four highest Host numbers are used.\* IMP messages are processed by the operational program as if they were Host messages, thus efficiently utilizing the existing program structure. The Host Source or Destination field of the leader identifies the source or destination of IMP messages, as indicated in Table 5-1.

TABLE 5-1

For IMP	Host field	From IMP
to IMP Teletype (TTY)	252	from IMP Teletype (TTY)
to DEBUG	253	from DEBUG
to PARAMETER-CHANGE	254	from TRACE
to DISCARD	255	from STATISTICS

\* See Sections 3.3 and 3.4



The background programs are sometimes referred to collectively as the *fake Hosts*. For each message to or from a fake Host, a RFNM is returned, just as with a regular Host-to-Host message.

A six-word leader must be provided for each IMP message. The source Host should include this leader at the beginning of each For IMP message in the usual fashion, specifying the particular background program with the Host field. When a message originates from an IMP background program, the IMP must have access to a leader to affix to the front of the message. A Host that activates the TRACE or STATISTICS programs is expected to provide (once only) a leader for the resulting messages, using PARAMETER-CHANGE. In the special case of DEBUG, messages are always returned to the Host that is using the DEBUG program. A Teletype user may specify the leader for TTY messages directly.\*

Messages to Host 252 will be typed on the destination IMP Teletype. Similarly, messages from Host 252 originate from the source IMP Teletype. The other entries in Table 5-1 can be interpreted straightforwardly. Messages to PARAMETER-CHANGE affect the value of selected IMP parameters; messages to DISCARD are thrown away; DEBUG messages are normally used in conjunction with the IMP Teletype for local debugging but may be used with any other IMP Teletype or Host program to aid in remote debugging; messages from TRACE and STATISTICS contain measurement information. TRACE and STATISTICS transmit messages but do not receive them; conversely, PARAMETER-CHANGE and DISCARD receive messages

---

\* The operation of the TTY and DEBUG facilities is described fully in the IMP Operating Manual, BBN Report No. 1877.

but do not send any. Both TTY and DEBUG can send and receive messages. The operational program restricts the use of DEBUG and PARAMETER-CHANGE to authorized users.

A From IMP message contains the 96-bit leader followed by text and padding. The text of a From IMP message is either pure binary text or ASCII characters. The binary From IMP messages are always multiples of 16 bits, ending with a 16-bit word containing padding, i.e., a one followed by 15 zeroes. The text of ASCII-character messages contains a multiple of 8-bit characters packed two to an IMP word. The first character is stored in the eight high-order bits, the following character in the eight low-order bits, and so forth. When the number of characters is odd, padding occurs in the second half of the last word; when the number of characters is even, the entire last word is devoted to padding.

## 5.1 TTY

It is possible to send messages from the IMP Teletype to any Host or to any IMP in the network; likewise, it is possible to send messages from any Host or IMP in the network to an IMP Teletype.\*

If bit 22 (one of the leader flags) in the leader is equal to zero, the TTY program interprets the text of messages to the IMP Teletype as 8-bit ASCII characters. These characters are printed literally, e.g., separate line feed and carriage return. If bit 22 is equal to one, the words are printed separately as octal numbers, one word per line.

---

\*The procedure for creating ASCII or binary messages from the IMP Teletype and for defining TTY leaders from the keyboard is described in the Operating Manual.

The text of messages from TTY either contain 8-bit ASCII characters (packed two to each word) as typed at the IMP Teletype, or binary text as is required, for example, by the PARAMETER-CHANGE program. Teletype-to-Teletype communication in general will not involve binary text, although bit 22 can be set in the leader of messages from TTY.

## 5.2 DEBUG

The debug program is primarily a tool for debugging the operational IMP program. It allows registers in core to be inspected while the system is running.

Any access to DEBUG other than from the Network Control Center requires that sense switch 4 be turned on at the site. *Sites must not change the setting of sense switch 4 without the express permission of the Network Control Center.* Pluribus IMPs do not have physical switches; a program switch is used to protect this feature. Coordination with the Network Control Center will be necessary for site use of DEBUG.

Network messages sent to DEBUG are interpreted as a sequence of ASCII characters that specify debug commands as if the commands had been typed at the local IMP Teletype. Network messages from DEBUG should be interpreted as an ASCII character sequence that constitutes DEBUG's response to a received command.\*

---

\* A list of the various debug commands and a description of their operation are given in the IMP Operating Manual.

### 5.3 PARAMETER-CHANGE

The parameter change routine allows a Host to specify the value of IMP parameters used in tracing and statistics gathering.\* These parameters are primarily leaders for trace and statistics messages and ON-OFF switches. The parameters that currently may be specified are listed below. Both the parameter number and its value must be specified to change any parameter. *The IMP is not vulnerable to the inadvertent mis-setting of these parameters.* However, some Hosts may be affected if a mis-setting causes unexpected, or unwanted, messages to be delivered to the wrong Host. Furthermore, the network performance can be seriously degraded by improper and excessive usage of the statistics messages. For these reasons access to PARAMETER-CHANGE is administratively controlled by the Network Control Center; a *destination dead* message will be returned to any Host which sends messages to PARAMETER-CHANGE without prior arrangement with the Network Control Center.

PARAMETER-CHANGE expects to receive a message whose text consists of consecutive pairs of IMP words, the first of which is a parameter number and the second of which is the parameter value. Within a single message to PARAMETER-CHANGE, the order of parameters is unimportant; however, parameters should be set in a logical order (e.g., leader contents before enabling flags). The twelve reserved parameters are used for reports to the Network Control Center and may not be changed. At initialization all other parameter values are reset to zero.

---

\* The IMP Teletype can be used to send messages to PARAMETER-CHANGE. The use of semicolon delimiters and colons to denote octal input are described in the IMP Operating Manual.

It is only necessary to specify four parameters for each leader, since the first word of any leader is a constant, and the sixth word of the leader (message length) will be set to 0.

Parameter  
Number  
(Octal)

0.	Trace Flag	Tracing is enabled when this flag is nonzero.
1.	Snapshot Flag	Snapshots are enabled when this flag is nonzero.
2.	Cumulative Statistics	Enabled when this flag is nonzero.
3.	Message Generator	Enabled when this flag is nonzero.
4.	Reserved	
5.	Reserved	
6.	Packet Trace Flag	Packet tracing is enabled when this flag is nonzero.
7.	Trace Leader	Fifth word
10.	Snapshot Leader	Fifth word
11.	Cumulative Statistics Leader	Fifth word
12.	Message Generator Leader	Fifth word
13.	Reserved	
14.	Reserved	
15.	Trace Leader	Fourth word
16.	Snapshot Leader	Fourth word

17.	Cumulative Statistics Leader	Fourth Word
20.	Message Generator Leader	Fourth Word
21.	Reserved	
22.	Reserved	
23.	Trace Leader	Third Word
24.	Snapshot Leader	Third Word
25.	Cumulative Statistics Leader	Third Word
26.	Message Generator Leader	Third Word
27.	Reserved	
30.	Reserved	
31.	Trace Leader	Second Word
32.	Snapshot Leader	Second Word
33.	Cumulative Statistics	Second Word
34.	Message Generator Leader	Second Word
35.	Reserved	
36.	Reserved	
37.	Autotrace Interval	If this value is equal to N, the trace bit will be set to one on every Nth message from the Hosts (including Fake Hosts). The trace bit will not be set if N=0. Parameter 0 may have any value. Trace messages are never autotraced.

40. Snapshot Frequency The minimum time between executions of the snapshot program, measured in 25.6 ms units (e.g., a value of  $40_8$  implies 0.82 seconds). The frequency must be a power of 2.
41. Cumulative Statistics Frequency The minimum time between executions of the cumulative statistics program, measured in 25.6 ms units (e.g., a value of  $1000_8$  implies 13 seconds). The frequency must be a power of 2.
42. Message Generator Frequency The minimum time between executions of the message generator program, measured in 25.6 ms units. If set to zero, the next message is sent immediately (without waiting for a RFNM). The frequency must be a power of 2.
43. Reserved
44. Reserved
45. Message Generator Length The message length is IMP words exclusive of leader and padding.
46. Packet Trace Interval If this value is equal to N, the trace bit will be set to one on every Nth packet passing through the IMP. The trace bit will not be set if N=0. Parameter 6 must be non-zero.
47. Round Trip Time Units Units for round trip time statistics -- value of from 0 to 16. Time is measured in 100 microseconds times 2 to the power of this parameter, e.g. 0:100 microseconds, 1:200 microseconds, 2:400 microseconds, etc.

50. Auto Trace  
Destination IMP  
If zero, messages to any destination are autotraced. Otherwise, only messages to the destination IMP equal to this parameter and destination Host equal to parameter 51 are autotraced.
51. Auto Trace  
Destination Host
52. Auto Trace Source IMP  
If zero, messages from any source are autotraced. Otherwise, only messages from the source IMP equal to this parameter and source Host equal to parameter 53 are autotraced.
53. Auto Trace Source  
Host

#### 5.4 DISCARD

This routine discards messages and returns a RFNM to the source. It is primarily useful in testing and performing measurements. DISCARD also happens to be the eventual repository for overlong messages and other undeliverable messages. In those cases, an *Incomplete Transmission* message is returned to the source Host.

#### 5.5 TRACE

The trace program forms a record that 1) identifies each traced packet and 2) contains a history of its processing through the IMP. This program is extremely useful in studying aspects of network activity such as routing, queueing, and IMP processing. The trace message is transmitted from fake Host 254.



A trace message is built by an IMP whenever the following three conditions hold: 1) the trace flag is non-zero, 2) a packet arrives with its Trace bit equal to one, and 3) a block of trace storage is available to record the trace information. *Tracing will not occur if the trace flag is off, even if the Trace bit is equal to one.* The trace flag is turned on by setting parameter 0 to a non-zero value using PARAMETER-CHANGE. The Trace bit in the message leader may be set to one either directly by the source Host or by the source IMP using the autotrace interval parameter, parameter 37 (octal). The use of autotracing can be made selective by the use of parameters 50 through 53 (octal). The Trace bit may also be set through the use of the packet trace feature. If parameter 6 is on, then the IMP will turn on the Trace bit in every Nth packet, where N is given by the Packet Trace interval, parameter 46 (octal).

Storage blocks are reserved in the IMP for recording the trace information. Occasionally, a packet may arrive with the Trace bit set when the trace blocks are momentarily all filled. In that case, an overflow indication is recorded and the packet is not traced by that IMP. A trace message is sent as soon as all the trace data are available.

A single trace message may contain several trace blocks, each block corresponding to a single packet. If, for example, a three-packet message is traced, the source IMP may generate three trace messages; the destination IMP, however, may generate only one trace message containing all three trace blocks.

The first six words of the message contain the trace message leader. The seventh word currently contains a one. The eighth word is an overflow register that will be non-zero whenever a

packet that should have been traced was not, due to a lack of trace storage. A message from the trace program has the format shown in Figure 5-1.

Figure 5-2 shows the format of a trace block. The first word of each trace block indicates the time at which the last bit of the packet arrived. The second word is the time at which the packet was processed on the task queue. The third word contains the time at which the IMP started to transmit the packet. For packets to the Host, the fourth word denotes the time that transmission to the Host ended. For packets sent out over a communication circuit, the fourth word denotes the time the acknowledgment was received for that packet. Five words taken from the *IMP header* of the traced packet occupy words 5 to 9. The last word contains the input channel on which the packet arrived, and the output channel onto which the packet was transmitted. For packets transmitted to the Host, the channel is equal to the Host number. Both lines and Hosts are numbered starting at zero. In this last word of each trace block, the high order 2 bits normally have the value  $2_8$ . A value of  $3_8$  indicates that the traced packet had to be rerouted onto another line and the channel number refers to the unsuccessful attempt; another trace block will eventually be generated with the new channel number and those bits equal to  $2_8$ . The next 6 bits contain the length in words of the packet, excluding 8 words of IMP overhead (acknowledges plus header).

The type of packet traced can be deduced from the source and destination of the traced packet and the source of the traced message. RFNMs are never traced. The trace times are measured in 100- $\mu$ sec units.

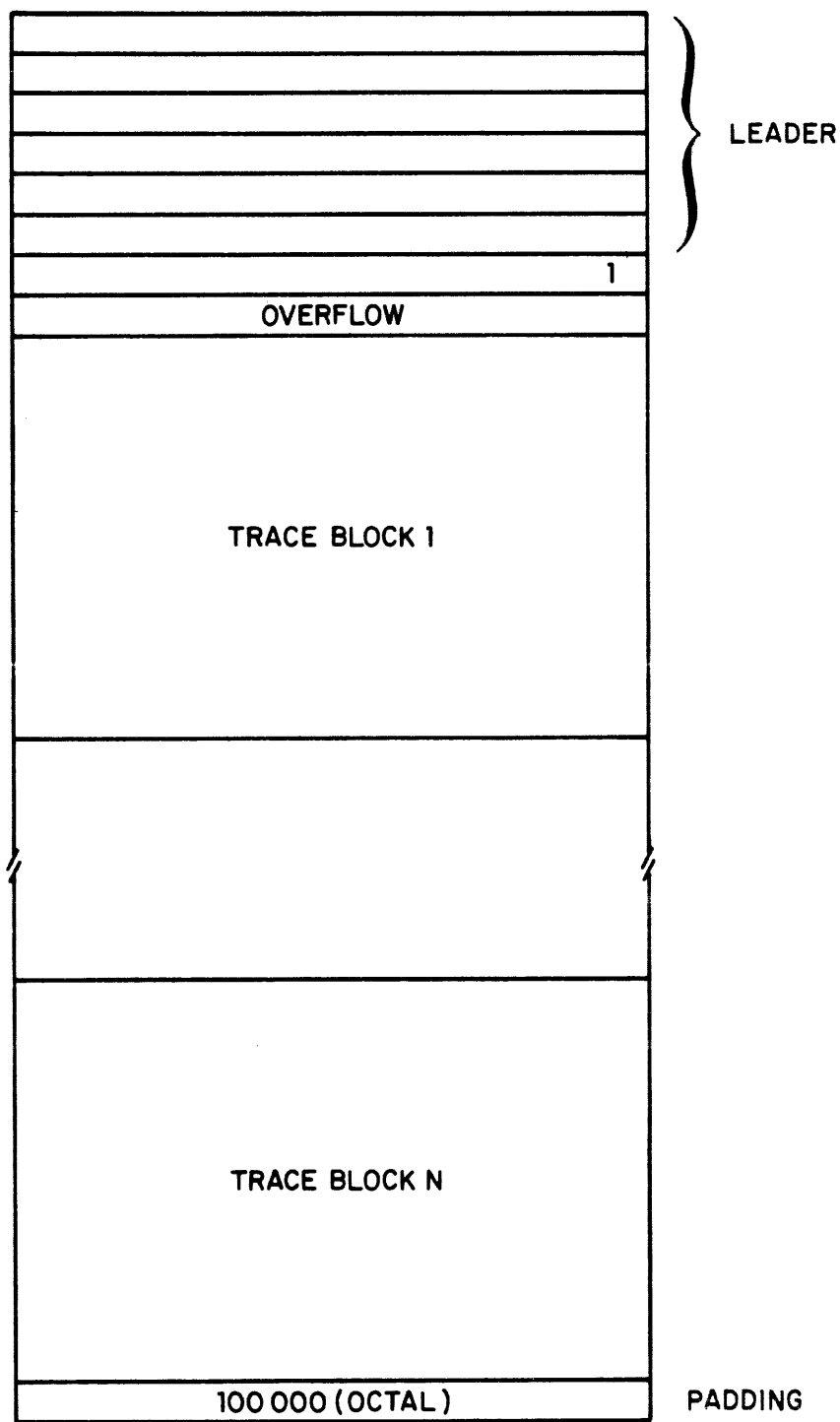


Fig. 5-1 TRACE MESSAGE FORMAT

ARRIVAL TIME						
QUEUE PLACEMENT TIME						
SENT TIME						
ACKNOWLEDGE TIME						
SOURCE IMP						
MESSAGE NUMBER				DESTINATION BLOCK NUMBER		
M <sub>LT</sub>	L <sub>ST</sub>	PRO <sub>ITY</sub>	T <sub>RACE</sub>	LEADER FLAGS	PACKET NUMBER	PACKET CODE
P <sub>KT</sub>	P <sub>KT</sub>					
DESTINATION IMP						
MESSAGE - ID					SUB-TYPE	
D O N E	R E R O U T E	DATA LENGTH		INPUT CHANNEL	OUTPUT CHANNEL	

Fig. 5-2 TRACE BLOCK FORMAT

## 5.6 STATISTICS

The statistics programs in the IMP can perform three different functions, each of which may be activated independently using the parameter-change routine. These functions are 1) taking snapshots, 2) recording cumulative statistics, and 3) generating artificial traffic. When one or more of these functions is activated, the IMP performs the designated activity and, except for the artificial traffic, transmits a separate message containing the recorded information to the selected destination.

The three message leaders may be independently selected using PARAMETER-CHANGE.

Statistics programs run in the background loop of the operational program and have lower priority than the handling of regular traffic. Should an IMP become compute-bound, the running of any statistics program is deferred until time is available. Each program is run to completion before another statistics program is executed.

A schedule of execution times for the statistics programs is determined by a synchronization procedure between IMPs which allows the snapshot program to run at approximately the same time in different IMPs. Further, it permits the cumulative statistics program in each IMP to be run in order of IMP number. Each IMP regulates its measurement interval according to its own IMP number relative to the total possible number of IMPs. For example, if the total # of IMPs is 64, and parameter 41 (octal) were set to 64 seconds in several IMPs at time T, IMP 1 would

run cumulative statistics at  $t=T+1$ ,  $T+65$ ,  $T+129$ , etc. IMP 5 would run cumulative statistics at  $t=T+5$ ,  $T+69$ ,  $T+133$ , and so on. Each statistics program is assigned a minimum time that must elapse between *scheduled* executions. This time is determined by the frequency assigned to that statistics program using PARAMETER-CHANGE. At the beginning of each pass through the background loop, a test is made to determine whether the minimum time has elapsed from the time the program was last scheduled to be run (it need not necessarily have been executed then). If the elapsed time has not exceeded the minimum time, the program is not run. During periods of sufficiently heavy traffic, the actual time between certain executions of a statistics program may be considerably above the minimum scheduled time.

Statistics messages are transmitted from fake Host 255. A buffer is normally always set up for transmitting statistics messages, but only one buffer at a time may be set up. Therefore, when more than one program is to be run, the statistics programs are executed in the following order. First the snapshot routine is run and its data is transmitted; then the cumulative statistics program is run and its data transmitted, the cumulative statistics having already been taken while the operational program was running; and, finally, the message generator is run.

In the current version of the system the number of Hosts, NH, is equal to 4; the number of fake Hosts, FH, is equal to 4; the number of channels, CH, is 5; and the number of IMPs, NIMP, is 64. The quantities NH, FH, CH and NIMP are incorporated as parameters in the IMP program and are set at assembly time.

At a minimum, these quantities should be assembly parameters of any program that decodes messages from the statistics program. Ideally, these parameters should be runtime parameters of the decoding program since they may eventually be site-dependent and are transmitted in the snapshot and cumulative statistics messages.

### 5.6.1 Snapshots

These messages contain information about the instantaneous state of the IMP, such as the length of queues, routing tables, etc. Other functions of the IMP required for processing packets are not inhibited while a snapshot is taken; consequently the snapshot is only a *close approximation* of an instantaneous picture. Snapshots taken at different IMPs in the network are approximately synchronized.

The leader is specified by parameters 32, 24, 16, and 10 (octal). The first word of data is a 5 and the second data word is the global time (in 25.6-msec units) that the message was transmitted. Data words 3-6 specify NH, FH, CH, and NIMP, respectively. The statistics and the padding word follow as shown in Figure 5-3.

The next  $2*(NH+FH)+4$  words contain queue lengths as listed in Table 5-2. The next  $2*NIMP$  words contain the current IMP routing table and the delay table in alternating words. The first word of each pair contains the current output line, i.e., the route, to IMP  $i$  in the high-order 8 bits (all ones if the IMP is down or unreachable). The second word contains the current delay/hop data, i.e., the route status, to IMP  $i$ ; the eleven low-order bits contain the delay in arbitrary units, and

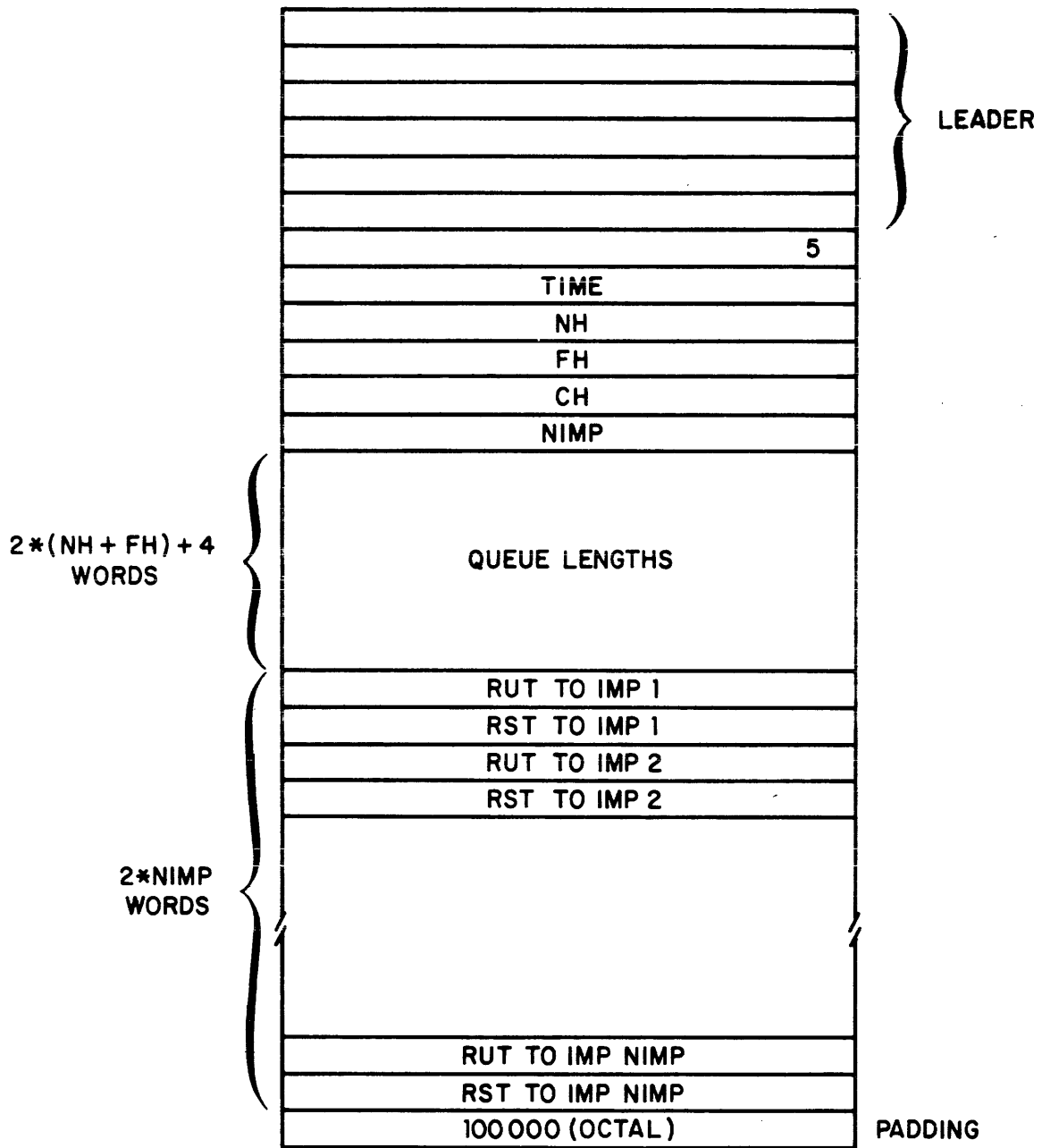


Fig. 5-3 SNAPSHOT MESSAGE FORMAT



TABLE 5-2

## Order of Queue Lengths in Snapshot Statistics

1	Regular queue to Host 0
.	
.	
NH	Regular queue to Host NH-1
NH+1	Regular queue to fake Host 0
.	
.	
NH+FH	Regular queue to fake Host FH-1
NH+FH+1	Priority queue to Host 0
.	
.	
2*NH+FH	Priority queue to Host NH-1
2*NH+FH+1	Priority queue to fake Host 0
.	
.	
2*(NH+FH)	Priority queue to fake Host FH-1
2*(NH+FH)+1	Free Storage queue
2*(NH+FH)+2	Store-and-Forward buffers in use
2*(NH+FH)+3	Reassembly buffers in use
2*(NH+FH)+4	Reassembly buffers allocated

the five high-order bits contain the minimum number of hops (i.e., the number of IMPs along the shortest path).

### 5.6.2 Cumulative Statistics

The leader for cumulative statistics is specified by parameters 33, 25, 17, and 11 (octal). As shown in Figure 5-4, the first word of data is 2; the second word is the time in 25.6 msec units; data words 3-6 specify NH, FH, CH, and NIMP, respectively; then follow  $44+2*(NH+FH)+2*NIMP+13*CH$  words of statistics and the padding word. The majority of Host statistics are combined into a single set of statistics. The individual behavior of Hosts 0, 1, 2, and 3 is thus unavailable.

For the statistics data, the first 15 words contain a histogram of message length (in packets) input to the IMP from the NH real Hosts. The first word of the histogram contains the number of two-packet messages; the second word contains the number of three-packet messages, etc.

The next 6 words contain a histogram of the number of (IMP) words of *data* for all last packets of messages input to the IMP from the NH real Hosts. The first word of the histogram is a count of the number of packets containing 0-1 data words and the remaining words of the histogram count (in order) the number of packets containing 2-3 words, 4-7 words, 8-15 words, 16-31 words, and 32-63 words. The sum of these 6 words minus the sum of the first 15 words gives the number of single-packet messages.

The next word contains a number which is the sum of the number of (IMP) words in the last packet of each message sent by each of the NH real Hosts during the measurement interval.

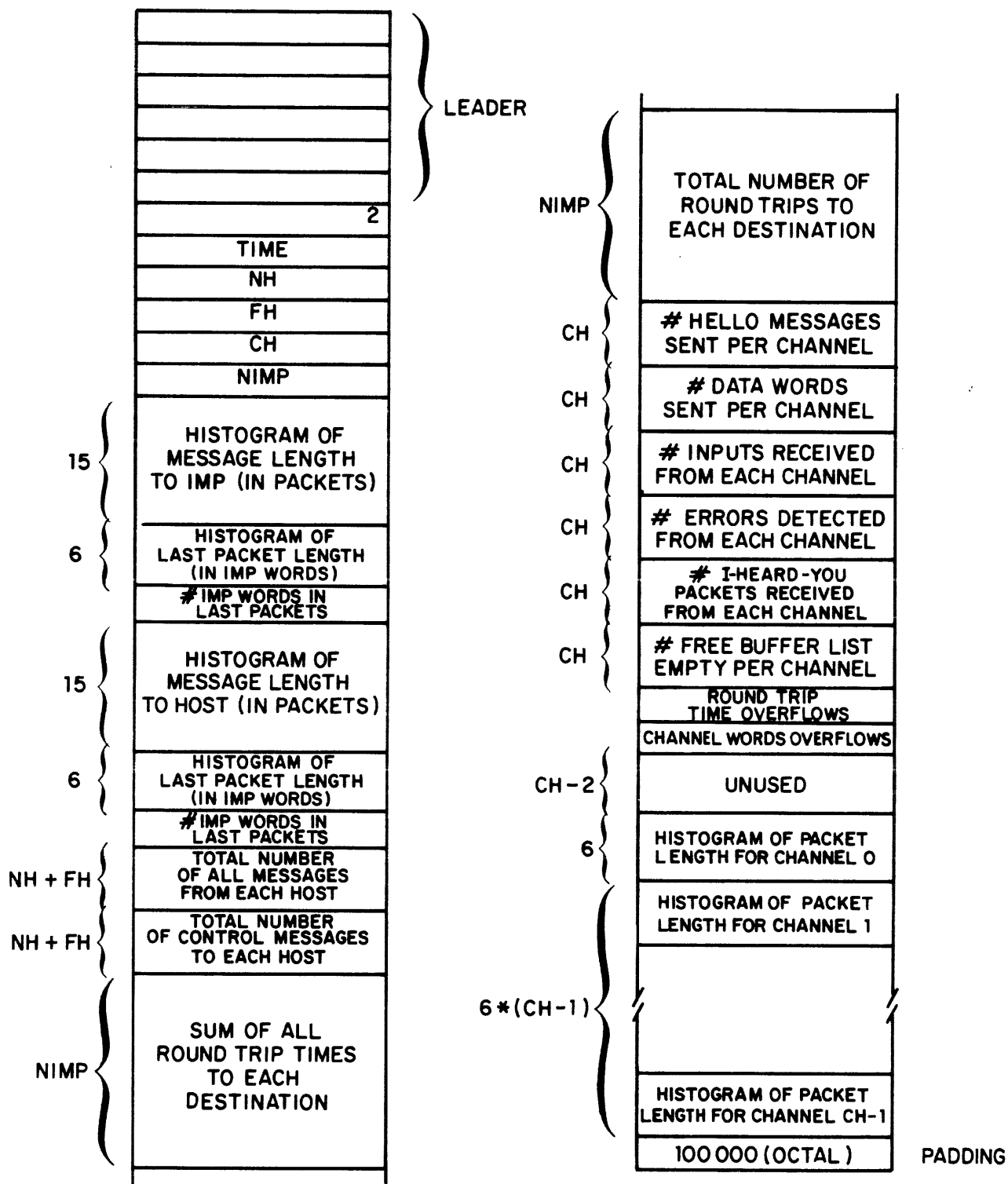


Fig. 5-4 CUMULATIVE STATISTICS MESSAGE FORMAT

The next 22 words contain statistics, analogous to the 22 words of statistics described above, for messages to the NH real Hosts.

The next NH+FH words contain a count of the total number of messages (including control messages) from each Host in the order real Host 0,1,...NH-1,Fake Host 0,1,...FH-1; these are followed by another NH+FH words containing a count of the total number of control messages to the Hosts, in the same order as above.

Next, there are two NIMP-word tables. The first table contains the sum of the times for all round trips of messages to each destination in 800-usec units. The second table contains the total number of round trips to each destination.

Then come seven CH-word blocks. In block 1 is contained the number of hello messages sent on each channel; in block 2 the number of *data* words sent; in block 3 the number of inputs received from the modems; in block 4 the number of errors detected on modem inputs; in block 5 the number of I-heard-you packets received; in block 6 the number of times the modem routine found the free buffer list empty. In block 7 the first word gives the number of times that any of the NIMP round trip time counters overflowed. The next word gives the number of times that any of the CH counters for the data words sent per channel overflows. The remaining CH-2 words (currently 3) are unused.

Finally, there are CH 6-word histograms of packet length transmitted on each modem circuit. The format is the same as for the lengths of the last packets of messages received from the Hosts, as described above.

### 5.6.3 Message Generator

The leader for the message generator is set up using parameters 34, 26, 20, and 12 (octal). The message generator is turned on using parameter 3. The number of IMP words in each message is specified by parameter 45 (octal) and does not include the leader and padding. The length may vary from zero, namely just a leader plus one word of padding, to the length of a full 8-packet message. If a length greater than 777 (octal) is specified, only the low-order bits will be used.

The generator interval is specified using parameter 42 (octal). This parameter indicates how many 25.6-msec intervals should occur between messages. If this value is zero, the next message will be sent immediately. The user will often wish to discard these messages at the destination IMP.

APPENDIX A  
OLD-STYLE LEADER FORMATS

This Appendix is included for completeness to retain a specification of a previous Host/IMP leader format. Hosts already using these formats will be supported into the indefinite future, but it is strongly recommended that new implementations use the current formats, and that other Hosts switch over to the current formats as soon as possible as it will be impossible to address the full range of Hosts on the network using the old formats.

It should be noted here that backward compatibility will also be maintained for Hosts using this older format and connected to the IMP as *Very Distant Hosts*. The VDH package in the IMP will be capable of correct operation with either the two or six-word leaders.

### A.1 Host-to-IMP Leader Format

For the most part, the various fields in this format correspond to, or are subsets of, fields in the Host-to-IMP leader format described in Section 3.3. Fields or subfields in Section 3.3 that are not explicitly contained in these formats are given a default value, as noted in the text.

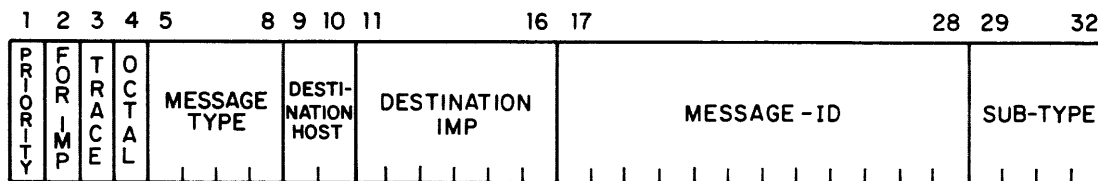


FIG. A-1 OLD-STYLE HOST-TO-IMP LEADER FORMAT

**Bit 1 Priority -**

Corresponds to bit 33, Section 3.3

**Bit 2 For IMP -**

Used to specify a Fake Host. If equal to one, causes the following mapping of values from bits 9-10 of this format to bits 41-48 of Section 3.3:

0	-	252
1	-	253
2	-	254
3	-	255

**Bit 3 Trace -**

Corresponds to bit 21, Section 3.3.

**Bit 4 Octal -**

Corresponds to bit 22, Section 3.3 and has a predefined meaning (octal print) for Host 252 (TTY) only.

**Bits 5-8 Message Type -**

Correspond to bits 25-32, Section 3.3, with the following exceptions:

- a) The subtype of type 0 (regular) messages will be ignored, and a subtype of 0 always used. Therefore subtypes 1 and 2 cannot be specified. Type 3 messages will be converted to type 0, subtype 3.
- b) The subtype of type 4 (NOP) messages will be ignored and a subtype of 0 always used.



Bits 9 - 10 Destination Host -

Correspond to bits 41-48, Section 3.3; also affected by bit 2 (For IMP).

Bits 11 - 16 Destination IMP -

Correspond to bits 49-64, Section 3.3

Bits 17 - 28 Message-id -

Correspond to bits 65-76, Section 3.3

Bits 29 - 32 Sub-type -

Correspond to bits 77-80, Section 3.3, with exceptions noted above.

The other fields of section 3.3 not specified above are given the following default values:

- a) Bits 23-24 (*leader flags*) -  $\emptyset$
- b) Bits 38-40 (*maximum message size*) - 7 (8 packet max).
- c) Bits 81-96 (*message length*) -  $\emptyset$  (information not needed).

A.2 IMP-to-Host Leader Format

As in Section A.1, there is a correspondence of bits in this format to those in section 3.4.

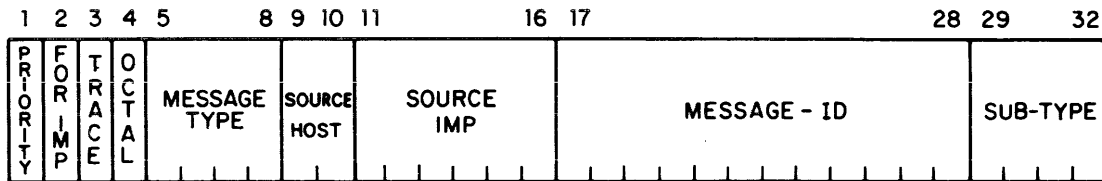


FIG. A-2 OLD-STYLE IMP-TO-HOST LEADER FORMAT

Bit 1 Priority -

Corresponds to bit 33, Section 3.4.

Bit 2 From IMP -

Used to specify a Fake Host. If equal to one, causes the following mapping of values from bits 9-10 of this format to bits 41-48 of Section 3.4:

- 0 - 252
- 1 - 253
- 2 - 254
- 3 - 255

Bit 3 Trace -

Corresponds to bit 21, Section 3.4.

Bit 4 Octal -

Corresponds to bit 22, Section 3.4.

**Bits 5 - 8 Message Type -**

Corresponds to bits 25-32, Section 3.4, with the following exceptions:

- a) Types 11, 12, 13, and 14 are never used.
- b) Type 0 subtype 3 (uncontrolled packets) messages, described in Section 3.7, are delivered to the Host as type 3 messages.

**Bits 9 - 10 Source Host -**

Correspond to bits 41-48, Section 3.4; also affected by bit 2 (From IMP).

**Bits 11 - 16 Source IMP -**

Correspond to bits 49-64, Section 3.4.

**Bits 17 - 28 Message-id -**

Correspond to bits 65-76, Section 3.4.

**Bits 29 - 32 Sub-type -**

Correspond to bits 77-80, Section 3.4.

APPENDIX B  
RECOMMENDATIONS FOR HOST IMPLEMENTATION  
OF THE HOST/IMP INTERFACE

This Appendix recommends a plan for Host implementation of the Host/IMP interface, both the hardware and the lowest-level Host software which drives the hardware. In particular, the software discussed here has the tasks of input and output, detection of errors, and management of the *Ready* lines. Figures B-1 and B-2 provide simplified schematic drawings of the Host interface hardware.

### B.1 Ready Line Philosophy

The actions which should be taken when transitions occur on the *Ready* line have the objective of reliably resynchronizing transmission after a temporary lapse of service, and possible loss of state information by either the IMP or the Host.

First, consider the IMP *Ready* line. When it drops, the IMP has suffered a possible loss of state, so the message in transit from the IMP to the Host (if any) is likely to be incomplete. Similarly, the message in transit from the Host to the IMP (if any) is likely to be incomplete. Both the Host and the IMP must recognize this explicitly by sending a message intended to be discarded (for example, a NOP) and discarding the message currently being received. (Note that the discardable message may be appended to some other message already partly received.)

The simplest arrangement for the Host's interface driver is a pair of processes, one sending messages and the other receiving messages. A drop of the IMP's ready line must be provided as an error status bit to each process. However, the two processes will need to clear this condition independently: the simplest implementation is an Input Error flop and an Output



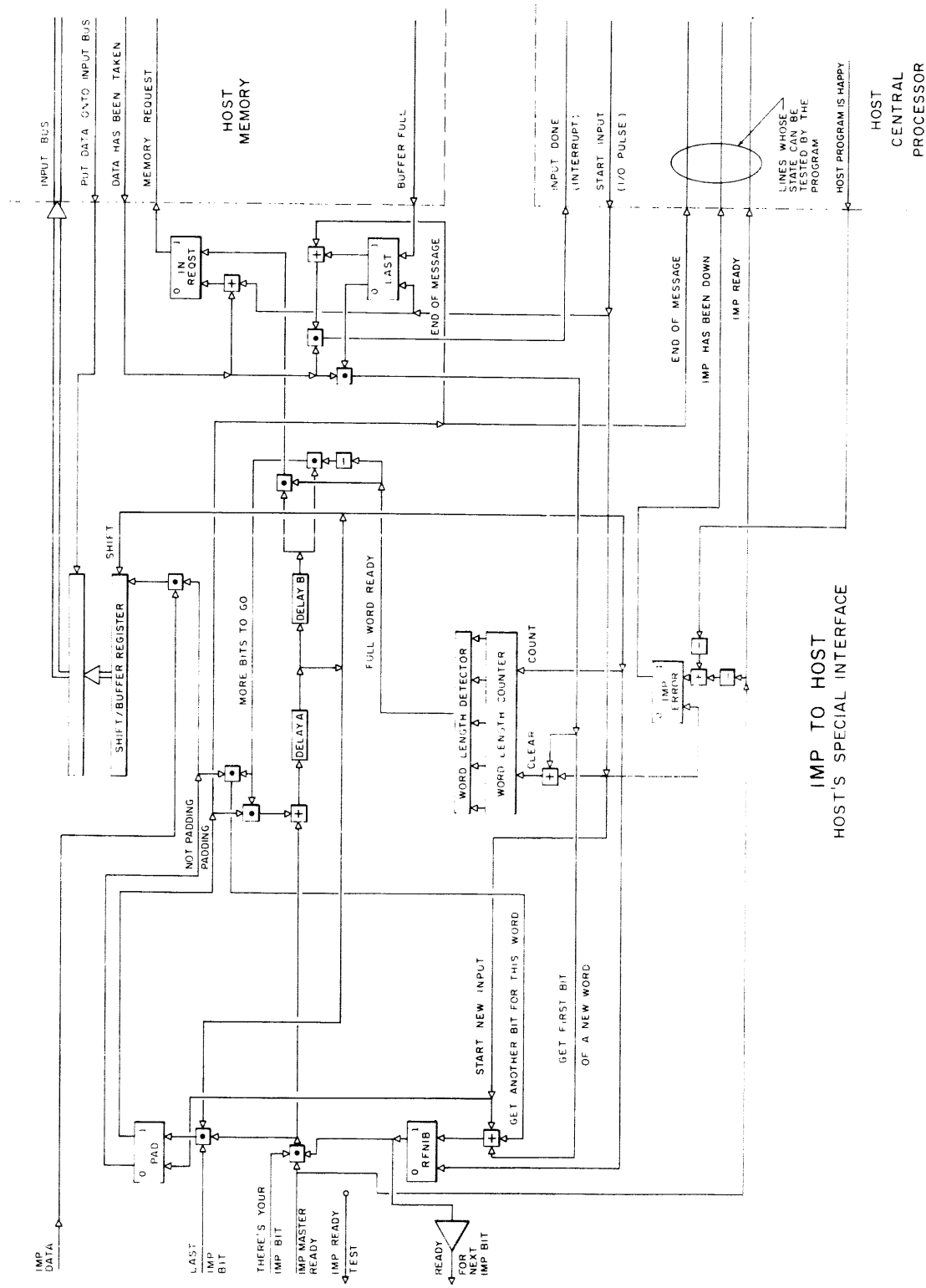


FIGURE B-2

Error flop. Both flops are set by a drop of the IMP's ready line, and they are cleared independently under program control.

When the IMP raises its ready line, each contact bounce will again set the Error flops in the Host's interface. To insure that messages are not flowing across the interface at this time, assertions of the signals There's-Your-IMP-Bit and Ready-For-Next-Host-Bit have been delayed sufficiently in the IMP to guarantee that the IMP ready line has stabilized.



## B.2 Programming the I/O Routines

System startup or restart requires the initialization step of clearing the Host Ready flop (driving the relay controlling the *Host Ready* line), waiting at least 1/2 second, and setting the Host Ready flop. Restarting the following two (asynchronous) interface driver processes will then properly resynchronize Host-IMP communication.

```
INPUT:  Wait until an input buffer is available
        Wait until IMP ready
        Start input
        Wait until input is complete
        IF Input Error
        THEN clear Input Error
           Comment:  Discard erroneous message; reuse
                    buffer
        ELSE queue message on input queue
        GOTO INPUT

OUTPUT: Wait until a message is present on output queue
        Wait until IMP ready
        Start output
        Wait until output is complete
        IF Output Error
        THEN clear Output Error
           Comment:  Save erroneous message for
                    retransmission
        ELSE remove message from output queue
        GOTO OUTPUT
```

The *IMP Ready* line and error flops should only affect the processes above; this interface resynchronization should be

invisible to both the process which interprets IMP-to-Host messages (it will be resynchronized by the IMP-to-Host type 10 message) and to any user software.

Actually, it is possible to share a single Error flop between the input and output processes by implementing Input Error and Output Error as software flags. A process testing for error must test both the Error flop and its own flag. An interlock is required (e.g., a mutual exclusion semaphore) to guarantee that only one process at a time is testing and modifying the flags. If the Error flop is set, the process must copy it into the other process' flag before clearing the flop and its own flag.

### B.3 Host Ready Line Implementation

When the Host drops and raises its ready line, the IMP behaves in a fashion symmetric to that outlined above. Of course, this drop indicates that the state of the Host's interface driver, as well as the current incoming and outgoing messages, are likely to be lost. The appropriate action is triggered by setting the Error flop or flops in the Host interface, and the processes specified above will correctly resynchronize message flow in both directions. Of course, to guarantee that messages are not flowing across the interface while the Host ready line is undergoing contact bounce, the Host must delay transmission until its ready line has stabilized. This may be done in two ways:

Hardware: an integrating one-shot driven by the Host ready line flop is ANDed with There's-Your-Host-Bit and Ready-For-Next-IMP-Bit to guarantee that message transfer will not start until the ready flop has been on for 1/2 second.

Software: the initialization program executes a 1/2 second wait after setting the Host ready flop before permitting input or output to begin.

#### B.4 Summary of Ready Line Controls

The following summarizes the specification of the Host's Ready Line control:

1. The special Host interface contains a Host ready flop which drives a relay closure in the Host Ready line. This flop is set and cleared under program control.
2. The special Host interface detects the IMP's ready signal and sets a program-readable status condition (not an "interrupt" condition).
3. The special Host interface contains one or two error flops set when either the Host Ready flop is off or the IMP ready signal is off. The flop (flops) is a program-readable and program-clearable status condition (but not an interrupt condition). These status flops must not be cleared by system initialization.
4. If hardware stabilization of the Host's Ready line is provided, it is a 1/2 second integrating one-shot driven by the Host Ready flop. This signal is ANDed with There's-Your-Host-Bit and Ready-For-Next-IMP-Bit.



APPENDIX C  
LOCAL HOST CONNECTION  
ELECTRICAL CHARACTERISTICS

## APPENDIX C

Local Host Connection  
Electrical Characteristics

All Host-IMP logic signals (Data, Ready-For-Next-Bit, There's-Your-Bit, Last Bit) are unbalanced, source-terminated lines with a nominal characteristic impedance of 68 ohms. The line is terminated at the driving end with the characteristic impedance. The receiver is ideally an open circuit; in practice, it is a single TTL gate. In this scheme a voltage step of half the nominal level is propagated from source to receiver. At the receiver, it is reflected by the high impedance termination, resulting in a full level step at the receiver and another half level step propagating back to the source, where it is absorbed by the termination.

Voltage levels for drivers and receivers used by the IMP are given below:

	Min	Typ	Max
<u>Pluribus</u>			
$V_{OH}$	4.1	5.0	
$V_{OL}$		0.07	0.1
$V_{IH}$		1.7	2.0
$V_{IL}$	0.6	0.9	
<u>316/516</u>			
$V_{OH}$	3.5	4.5	
$V_{OL}$		0.2	0.35
$V_{IH}$		1.55	2.5
$V_{IL}$	1.1	1.35	

12/75

C-2

The IMP will properly receive 5-volt logic signals; however, signals from the IMP may go to 6 volts. Therefore, the Host must provide a voltage divider, if these signals are to be received by normal 5-volt logic, to prevent destruction of the receiving circuit.





APPENDIX D

DRIVER RECEIVER FOR DISTANT HOST

Printed with permission of Honeywell, Inc.  
Computer Control Division, Framingham, Massachusetts  
Descriptions and schematic diagrams reflect use in the  
IMP.

## D.1 DIFFERENTIAL RECEIVER PAC MODEL CC-124

The Differential Receiver PAC, Model CC-124, contains three identical and independent circuits. Each circuit takes a bipolar differential signal and converts it to standard single-ended  $\mu$ -PAC logic levels. The schematic diagram (Figure D-1) reflects the use of this PAC in the IMP.

### D.1.1 Circuit Description

The circuit functions as both a Differential Amplifier and Discriminator.

When the "+A" input is more positive than the "-A", Q3 is cut off and the output is a logic "1". When the polarity of the input signal is reversed, or "-A" is made more positive than "+A", then Q3 is turned on and the output goes to logic "0".

### D.1.2 Terminating Network

The input to the CC-124 is unterminated. The terminating network in the PAC is not used.

### D.1.3 Specifications

Frequency of Operation: DC to 5 MC.

Input: Differential signal, 0.1V to 4.0V.

Input Impedance: 2.5K (min.)

Common Mode Rejection: Greater than  $\pm 2.5V$

Output Drive Capability: 8 unit loads and 70 pf stray capacitance each.

Circuit Delay: 30 nsec (max.).

Current Requirements (exclusive of terminators):

+6V: 60 ma (max.).

-6V: 30 ma (max.).

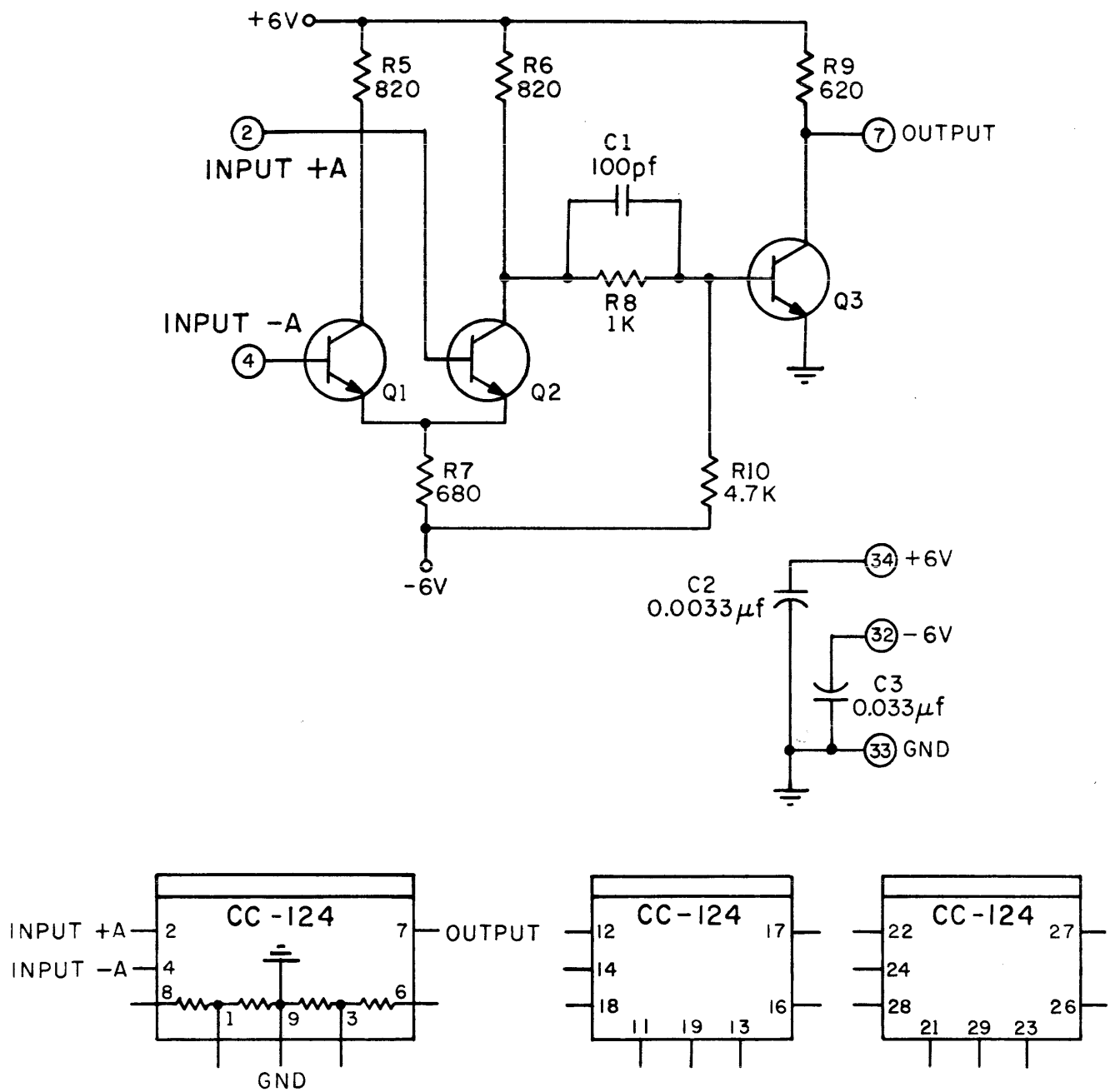


FIG.D-1 DIFFERENTIAL RECEIVER PAC MODEL CC-124. SCHEMATIC DIAGRAM AND LOGIC SYMBOL. (SHOWN AS CONNECTED IN IMP)

## D.2 DIFFERENTIAL LINE DRIVER PAC, MODEL CC-125

The Differential Line Driver PAC, model CC-125, contains three identical and independent circuits. Each circuit will switch approximately 18 ma into a balanced load when a standard  $\mu$ -PAC logic level of "1" is present at the input. When the input is at logic "0", the output is open circuited. The schematic diagram (Figure D-2) reflects use of this PAC in the IMP. Note that the circuit has been modified by the addition of externally (i.e., back panel) mounted resistors.

### D.2.1 Circuit Function

When the input is at ground or logic "0",  $Q_1$  is biased "on". With  $Q_1$  "on", the emitters of  $Q_2$  and  $Q_3$  are biased "off", and the output is effectively open-circuited.

When the input is open or at logic "1",  $Q_1$  is turned "off", which causes  $Q_2$  and  $Q_3$  to turn on, switching approximately 18 ma into the output.

### D.2.2 Terminating Network

The terminating network consists of resistors R7-R10 as well as the externally mounted resistors R101 and R102, and is designed for use with 100 to 140 ohm, balanced, twisted-pair transmission lines. With a logic "0" applied to the input of the transmitter, the terminating network establishes a 1.0V differential signal on the transmission line pair. When a logic "1" is applied to the input of the transmitter, the polarity of the 1-volt differential signal on the transmission line pair will be reversed.

### D.2.3 Specifications

Frequency of Operation: DC to 5 MC.

Input Loading: 1 unit load each.

Output Drive Capability: Approximately 18 ma into a  
balanced load.

Circuit Delay: 15 nsec (max.).

Current Requirements: Exclusive of terminators

+6V: 90 ma (max.).

-6V: 90 ma (max.).

The combination of the internal terminator network and the externally connected resistors will draw about 9 ma each when connected to +6V and -6V.

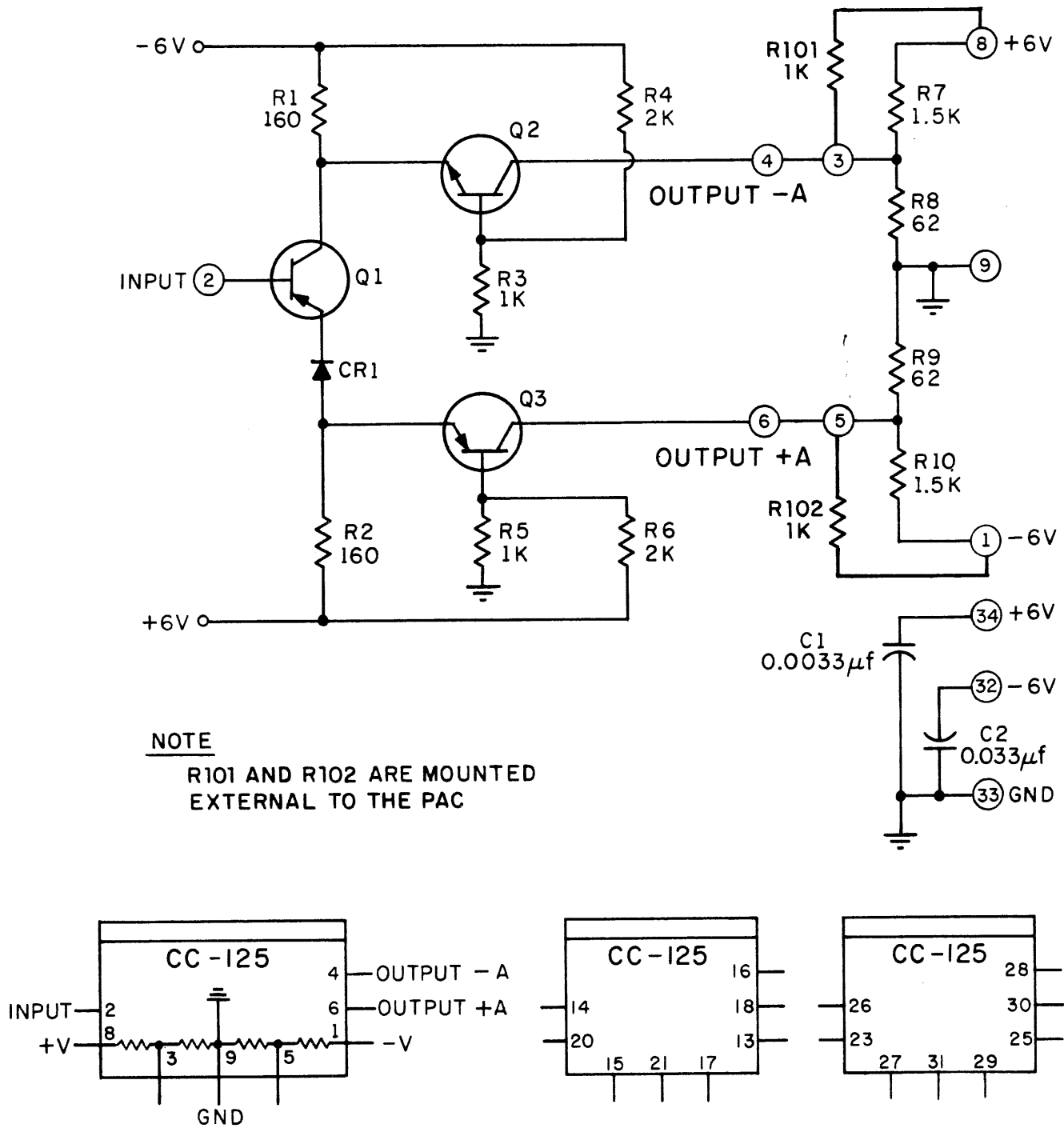


FIG.D-2 DIFFERENTIAL LINE DRIVER PAC MODEL CC-125. SCHEMATIC DIAGRAM AND LOGIC SYMBOL. (SHOWN AS CONNECTED IN IMP)

APPENDIX E  
ASCII CODES



ASCII CODES

ASCII		MNEMONIC	SYMBOL
OCT	HEX		
200	80	NUL	↑@
201	81	SOH	↑A
202	82	STX	↑B
203	83	ETX	↑C
204	84	EOT	↑D
205	85	ENQ	↑E
206	86	ACK	↑F
207	87	BEL	↑G
210	88	BS	↑H
211	89	HT	↑I
212	8A	LF	↑J
213	8B	VT	↑K
214	8C	FF	↑L
215	8D	CR	↑M
216	8E	SO	↑N
217	8F	SI	↑O
220	90	DLE	↑P
221	91	DC1	↑Q
222	92	DC2	↑R
223	93	DC3	↑S
224	94	DC4	↑T
225	95	NAK	↑U
226	96	SYN	↑V
227	97	ETB	↑W
230	98	CAN	↑X
231	99	EM	↑Y
232	9A	SUB	↑Z

ASCII		MNEMONIC	SYMBOL
OCT	HEX		
233	9B	ESC	↑[
234	9C	FS	↑\
235	9D	GS	↑]
236	9E	RS	↑↑
237	9F	US	↑←
240	A0	SP	Space
241	A1		!
242	A2		"
243	A3		#
244	A4		\$
245	A5		%
246	A6		&
247	A7		'
250	A8		(
251	A9		)
252	AA		*
253	AB		+
254	AC		,
255	AD		-
256	AE		.
257	AF		/
260	B0		0
261	B1		1
262	B2		2
263	B3		3
264	B4		4
265	B5		5
266	B6		6
267	B7		7
270	B8		8

ASCII		MNEMONIC	SYMBOL
OCT	HEX		
271	B9		9
272	BA		:
273	BB		;
274	BC		<
275	BD		=
276	BE		>
277	BF		?
300	C0		@
301	C1		A
302	C2		B
303	C3		C
304	C4		D
305	C5		E
306	C6		F
307	C7		G
310	C8		H
311	C9		I
312	CA		J
313	CB		K
314	CC		L
315	CD		M
316	CE		N
317	CF		O
320	D0		P
321	D1		Q
322	D2		R
323	D3		S
324	D4		T
325	D5		U
326	D6		V
327	D7		W

ASCII		MNEMONIC	SYMBOL
OCT	HEX		
330	D8		X
331	D9		Y
332	DA		Z
333	DB		[
334	DC		\
335	DD		]
336	DE		^ (↑)
337	DF		-
340	E0		`
341	E1		a
342	E2		b
343	E3		c
344	E4		d
345	E5		e
346	E6		f
347	E7		g
350	E8		h
351	E9		i
352	EA		j
353	EB		k
354	EC		l
355	ED		m
356	EE		n
357	EF		o
360	F0		p
361	F1		q
362	F2		r
363	F3		s
364	F4		t
365	F5		u

ASCII		MNEMONIC	SYMBOL
OCT	HEX		
366	F6		v
367	F7		w
370	F8		x
371	F9		y
372	FA		z
373	FB		{
374	FC		
375	FD		}
376	FE		~
377	FF	DEL	RUBOUT

---

The IMP uses 8-bit ASCII with the left-most bit set to one.

↑ = control

↑@ = shift control - P

APPENDIX F

VERY DISTANT HOST INTERFACE

F.1 PHILOSOPHY OF THE VERY DISTANT HOST INTERFACE

It is sometimes desirable to connect a Host and an IMP over a distance greater than 2000 feet, the maximum distance over which the distant Host interface can guarantee error-free transmission. Such a connection is possible with a relatively small change to the IMP/Host protocol and is made in the manner discussed in the following paragraphs. We call this kind of connection a *very distant host* connection.

Normally, connection between an IMP and any of its Hosts takes place as illustrated in Figure F-1.

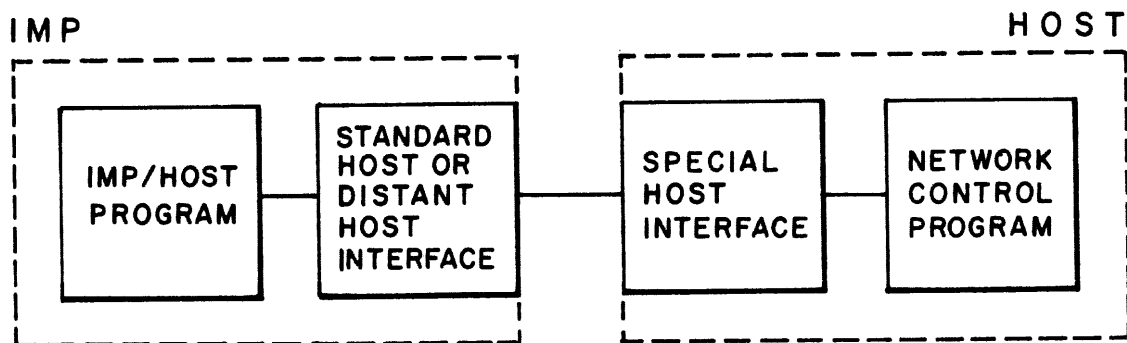


FIG. F-1 NORMAL IMP/HOST CONNECTION

The standard Host (or distant Host) interface and the special Host interface communicate according to the hardware specification set down in Section 4 of this document, and the IMP/Host program (in the IMP) and the Network Control Program (in the Host) communicate using the software protocol described in Section 3 of this document.

To minimize the disturbance to existing programs and specifications in both the IMPs and the Hosts, the very distant Host connection is implemented by adding a new level of protocol (which can be programmed in self-contained front end packages) and using the IMP's standard modem interface as shown in Figure F-2.

At the IMP end of the connection, the Host interface is replaced by a modem interface and a modem, and a software package which provides reliable packet transmission is added between the IMP/Host program and the hardware interface. At the Host end of the connection, a modem is added, along with some sort of hardware device which provides an interface between the modem and the Host. Also, between the hardware interface and the Network Control Program, a software package which provides reliable packet transmission is added. As before, the IMP/Host program (in the IMP) and the Network Control Program (in the Host) communicate according to the software specifications in Section 3. The new Reliable Transmission Packages in the IMP and the Host communicate as outlined in Section F.2 below; the modem interface in the IMP and the Error Detecting Special Host Interface communicate using the line protocol currently used between IMPs as discussed in Section F.3.

The new components that are required at the Host end of a very distant Host connection are a Reliable Transmission Package and a new piece of hardware, namely the Error Detecting Special Host Interface. The next two sections specify the functioning of the Reliable Transmission Package and the Error Detecting Special Host Interface.



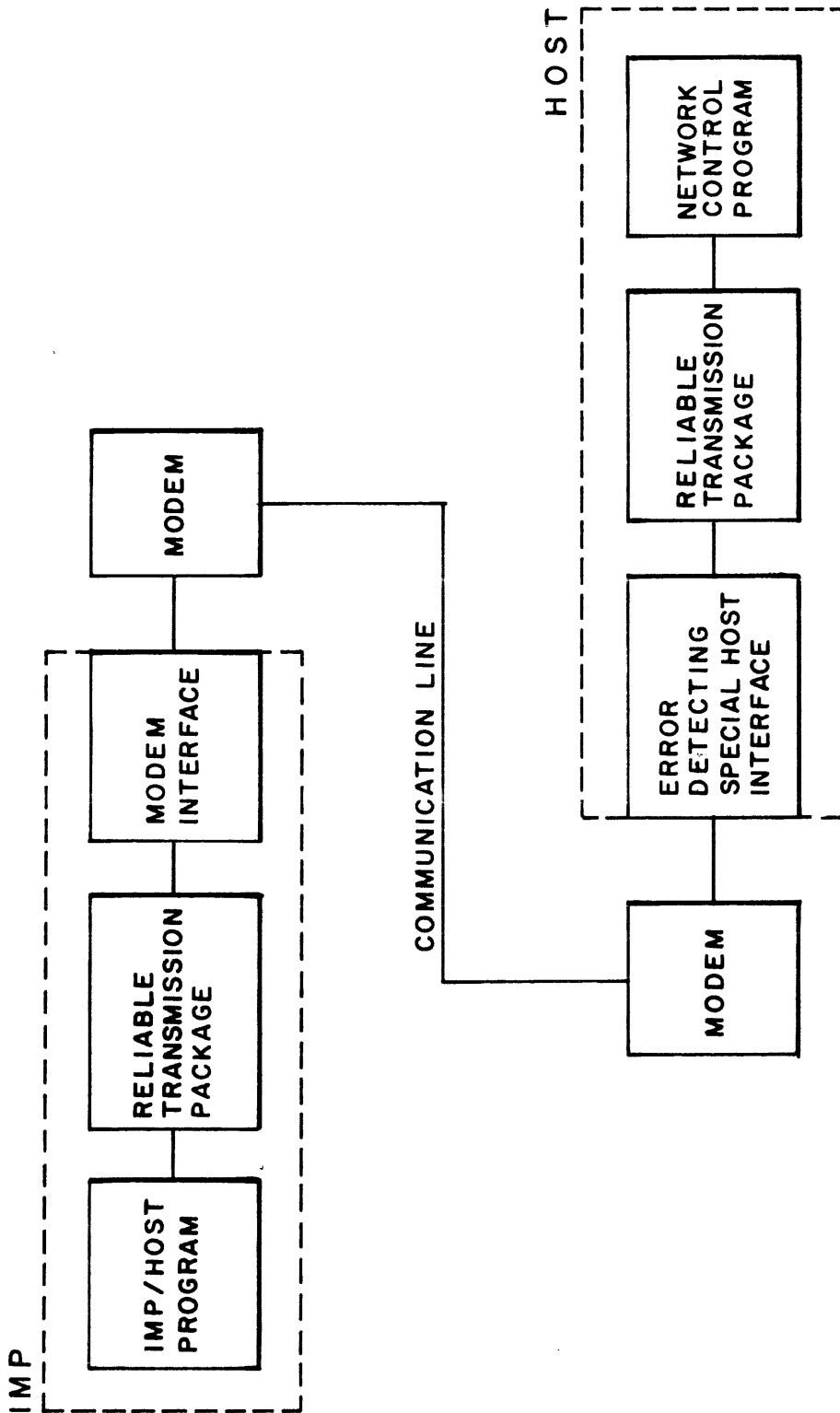


FIG. F-2 IMP/HOST CONNECTION FOR VERY DISTANT HOST

## F.2 THE RELIABLE TRANSMISSION PACKAGE

The Reliable Transmission Packages (RTPs) in the Host and IMP are functionally equivalent. Both send and receive packets of data which are multiples of 16 bits in length. Appended to the front of each packet is 16 bits of control information as shown in Fig. F-3.

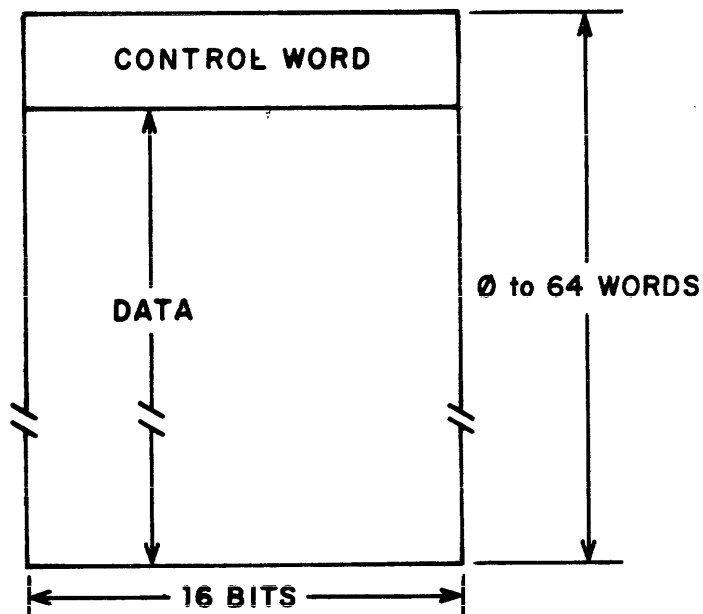


FIG. F-3 PACKET FORMAT

The 16-bit word of control information, as illustrated in Fig. F-4, includes a count giving the length (in 16-bit words) of the *data* in the packet, a bit which, when set, indicates the last packet of a message, an "odd/even" bit which is used to detect duplicate packet transmissions, a one-bit "channel number", a Host/IMP bit, and two acknowledge bits—one for channel zero and one for channel one.

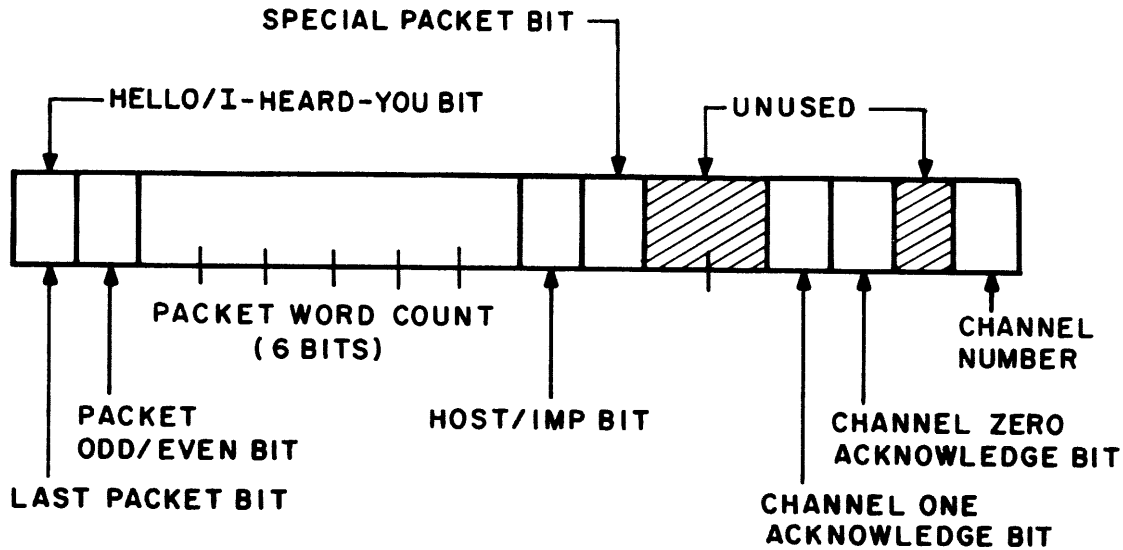


FIG. F-4 CONTROL WORD FORMAT

For efficiency, the RTPs must be able to handle two packets going in each direction (transmit and receive) simultaneously.\* At any time, each of the two packets going in one direction is associated with one of the two "channels" mentioned above. For each transmit channel the RTP keeps a used/unused bit and an odd/even bit (both initialized to zero). The used/unused bit indicates whether there is currently a packet associated with the channel. For each receive channel, an odd/even bit is kept (also initialized to zero). The transmit portion of the RTP cycles\*\* through its *used* channels (those with packets associated with them), transmitting the packets

\*Note that the control word format is laid out to enable easy expansion to four channels.

\*\*Packets must be retransmitted until acknowledged. IMP delay and transmission delay, however, may delay acknowledgment for more than one packet transmission time. Unnecessary retransmission may interfere with new transmissions, as well as placing an added burden on both transmitter and receiver. Therefore, we recommend a program delay before deciding to retransmit an unacknowledged packet; the amount of delay should be adjustable but we recommend an initial (trial) value of 100 msec.

along with the channel number and the associated odd/even bit. At the receive side of the RTP, if the odd/even bit of the received packet matches the odd/even bit associated with the appropriate receive channel, the receive odd/even bit is complemented. Otherwise the packet is a duplicate and is discarded. Acknowledgments of all packets correctly received at the receive side of the RTP, whether the acknowledgments are duplicate or not, are sent to the transmit side of the other RTP. This is done by copying the receive odd/even bits for both channels into the positions reserved for the two acknowledge bits in the control word of *every* packet transmitted. In the absence of other traffic, the acknowledges are returned in 16 bit "null packets". These have a word count of zero. When the transmit portion of the RTP receives a packet, it compares (bit by bit) the two acknowledge bits against the two transmit odd/even bits. For each *non-match* found, the corresponding channel is marked unused and the corresponding packet is discarded, and the odd/even bit is complemented. The transmit portion of the RTP must fill its channels in sequence\* (one to channel zero, one to channel one, one to channel zero, ...), waiting if necessary for any outstanding acknowledgments. The receive portion must pass on correctly received packets in sequence\*, waiting for the retransmission of any missed packet. To insure correct sequencing, the first channel filled or emptied after initialization must be channel zero. Null packets do not use a channel (nor a channel number) when sent and are not acknowledged when received.

---

\*Although packets may be transmitted, retransmitted, and received out of sequence.

The following algorithm is used to decide whether the circuit between an IMP and a very distant Host is dead or alive. We first define what we call a special packet — this is (logically) a one-word packet consisting of only the control word and having the SPECIAL PACKET bit set to one. All packets which are not special packets (i.e., which are regular data packets or null packets) have the SPECIAL PACKET bit set to zero. In a special packet, none of the control word fields or bits have their usual meanings; consequently, a special packet cannot be used to acknowledge data packets or send data. In a special packet, only two bits other than the SPECIAL PACKET bit have any meaning, the HELLO/I-HEARD-YOU bit and the Host/IMP bit.

Every  $r$  seconds both IMP and Host (independently) send a HELLO packet, a special packet with the HELLO/I-HEARD-YOU bit set to zero. When either IMP or Host receives a HELLO packet, it must promptly (with highest priority) send the other an I-HEARD-YOU packet, a special packet with the HELLO/I-HEARD-YOU bit set to one. In other words, the I-HEARD-YOU packet is an acknowledgment of the periodic HELLO packet, and an I-HEARD-YOU packet must only be sent as an acknowledgment for a HELLO packet. If either IMP or Host sends more than  $t$  HELLO packets without receiving an I-HEARD-YOU packet in acknowledgment, the IMP or Host declares the line dead. Once either IMP or Host declares the line dead, it must send or accept no packets (either special or regular) for  $2*t*r$  seconds to allow the other party also to declare the line dead. After waiting  $2*t*r$  seconds, an attempt is made to bring the line alive. This is done by sending HELLO packets (but no regular packets) every  $r$  seconds while noting received I-HEARD-YOU packets until

k HELLO packets in a row are acknowledged with I-HEARD-YOU packets. While doing this, received HELLO packets must be acknowledged with I-HEARD-YOU packets. Once acknowledgments for k HELLO packets have been received in a row (i.e., one acknowledgment every r seconds for k intervals\*), the line is declared alive and regular packets again may be sent, received, and acknowledged along with the periodic (every r seconds) HELLO packets. If a regular data packet is received while a party is trying to bring the line up (due perhaps to slight timing differences between the parties at the ends of the line), the data packet must not be acknowledged.

The odd/even bits, the used/unused bits, and the channel filling and emptying sequences must be initialized at start-up\*\* and reinitialized every time the line is declared dead. If either the IMP or Host decides the line is dead, the same action is taken that the IMP or Host normally takes when the other's ready line is down. The line being up causes the same action as is normally taken when the ready line is up. The value of r is currently 1.25 seconds, the value of t is currently 4, and the value of k is currently also 4. It is likely that the values of r, t, and k will be adjusted in the future; very distant Host programmers are advised to make it easy to change these parameters.

---

\*In particular, the IMP implementation requires the receipt of an acknowledgment within r seconds of the transmission of a HELLO packet in order to consider that the HELLO packet was successfully acknowledged.

\*\*At start-up, the line must be assumed to be dead and the procedure of waiting  $2*t*r$  seconds before sending HELLO packets, etc., must be used to bring the line alive initially.

Packets going from the IMP to the Host have the Host/IMP bit set to zero. Packets going from the Host to the IMP have the Host/IMP bit set to one. This enables the IMP or Host to discard packets which could cause errors when the telephone line is looped back on itself as occasionally happens. In fact the Host, the IMP, or the modem manufacturer may desire the ability to loop the connection for test purposes, and the RTP should probably be designed with this in mind.\*

The IMP requires that transmissions to and from a very distant Host be in packets which are multiples of 16 bits in length, up to a maximum of 1008 bits (not including the 16 bits of control information which is appended to the front of each packet). Thus, unlike a normal Host, a very distant Host must be aware of packets. Furthermore, the leader of a message is transmitted to and from a very distant Host in a separate packet which precedes the rest of the packets of the message. For instance, a message of length 1500 bits would be transmitted in three packets as shown in Fig. F-5.

As previously mentioned, the word count in the word of control information preceding each packet does not include the word of control information itself. Thus, packets which include data have a word count between one and sixty-three. A packet without any data, the "null packet" which is sent in the absence of traffic, has a word count of zero. A maximum length message is segmented into nine packets for transmission over a very distant Host line, eight packets for the data in the message and one packet for the leader.

---

\*In particular, if the Bell 303 modem is used it includes the capability of being looped back on the Host under Host program control. See the *Bell System Communications Technical Reference Manual on Wide Band Data Stations, 303 Type (PUB 41302)* for the specification of the signals needed to activate this feature.

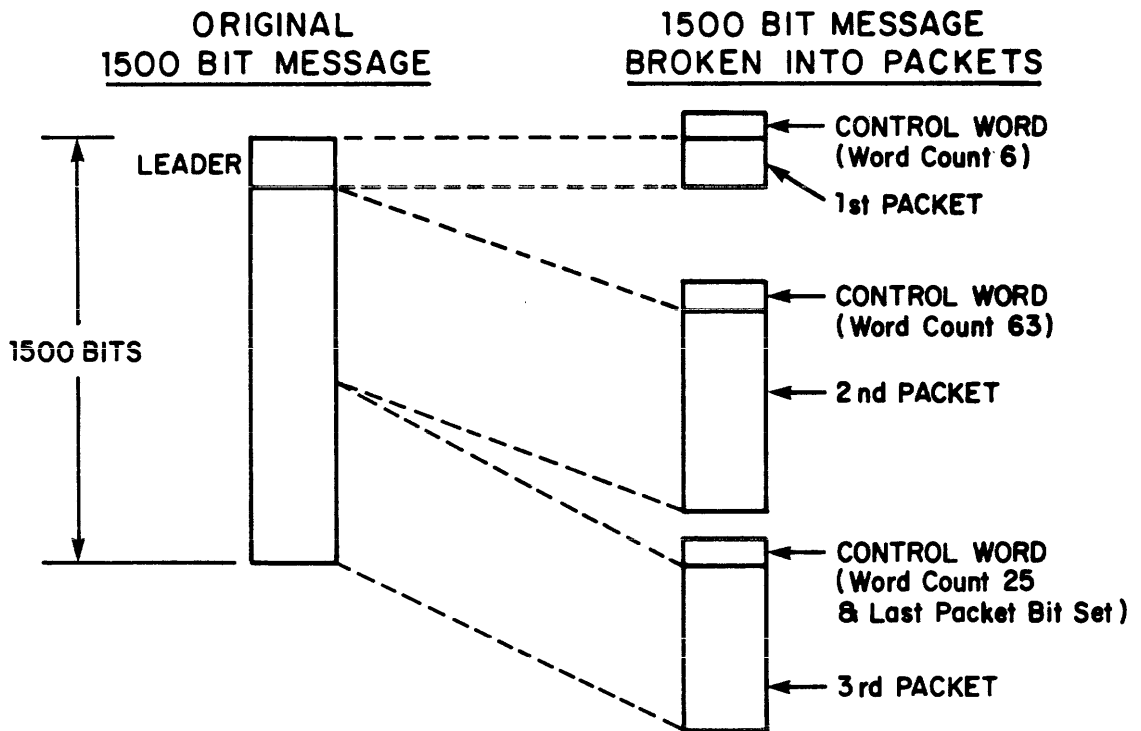


Fig. F-5 SEGMENTATION OF A MESSAGE INTO PACKETS FOR THE VERY DISTANT HOST INTERFACE



The "padding" convention (see Section 3.5) is not simulated for messages traversing a very distant Host interface. Thus, the maximum length message the IMP will allow to come from a Host over a very distant Host interface is 8160 bits (including leader), one bit longer than is permissible over a normal IMP/Host connection (again, see Section 3.5). Further, IMP/Host control messages only require 96 bits, since padding is not required.

Another important ramification of the use of multiples of 16-bit words over a very distant Host interface is that a word length mismatch problem may exist. For instance, it takes four 36-bit (PDP-10) words to make a multiple of 16 bits. The RTP in the IMP uses the following series of tests to determine message legality on the basis of message length:

1. Any packet which (including the control word) is physically longer than sixty-four 16-bit words (regardless of the packet's word count) is discarded.
2. Any packet which is less than one 16-bit word long is discarded.
3. A packet may have a word count of zero and such a packet is treated as a "null packet".
4. The first packet of a message must be physically at least 7 words long and must have a word count of exactly six.

5. The second through the ninth packets of a message may have a word count from one to sixty-three, but the physical packet length may be different. In the case of a packet physically shorter than the word count indicates, the IMP fills out the packet with garbage to the length specified by the word count. In the case of a packet physically longer than the word count indicates (but not longer than a total of 64 words) the IMP uses only the portion of the packet specified by the word count. The packet boundaries specified by the word counts from the very distant Host will be preserved by the IMP subnetwork; thus if the *destination* of the message is a Host connected to its own IMP via a very distant Host interface, the delivered message may consist of several packets, each of less than maximum length.

Because the actual packet length cannot be greater than 64 words, the Host's very distant Host interface must be able to discard any information remaining in the Host's output buffer (probably only part of one Host word) after 64 16-bit words have been sent to the IMP *or else* the Host must reconcile itself to being able to send a maximum length message which is somewhat shorter than normal. For instance, the maximum length message a PDP-10 might be able to send would be about 160 bits shorter than the normal 8160 bits.

### F.3 THE ERROR DETECTING SPECIAL HOST INTERFACE

We see several reasonable ways for the Host to build the Error Detecting Special Host Interface. For example:

1. Build the equivalent of the IMP's modem interface, a device which provides an interface between the modem and the Host.
2. Adapt the Special Host Interface so that it interfaces to a modem instead of to an IMP as shown in Fig. F-6.

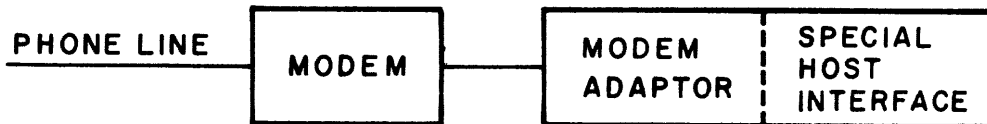


FIG. F-6 ADAPTATION OF SPECIAL HOST INTERFACE

3. Place a mini-computer between the Host and the modem (and program the RTP in the mini-computer).

Any of these methods is feasible; the method chosen will depend on what is comfortable at the Host site.

Whichever method is chosen, the interface must follow the same line protocol the IMPs now follow between themselves. This protocol is described in the following sections.

#### F.3.1 Message Formatting

Fig. F-7 shows the format of a packet *on the phone line*. This format is the realization of a particular Binary Synchronous Communication mechanism wherein a packet of data is enclosed

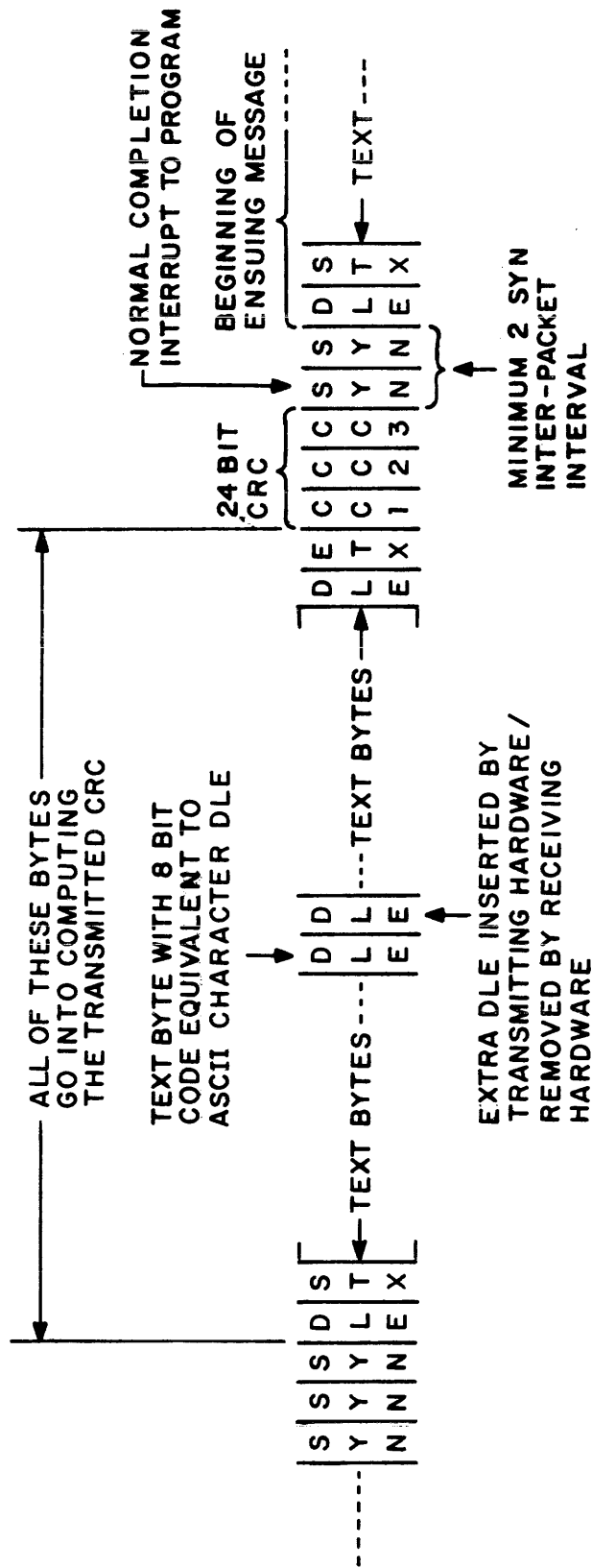


FIG. F-7 PACKET FORMAT ON LINE

within a framing structure provided by the hardware.

We will describe the structure for a unidirectional transmission although transmissions in either direction are of this same form.\* When the line is in the passive state (that is, when no information is being transmitted) the hardware generates a continuous stream of SYN characters (see Section F.3.3 for code definitions). In addition to this passive stream, the hardware guarantees that at least two SYN characters will always be inserted between successive packets. When the program has a packet ready for transmission, it notifies the hardware which, at the end of the current SYN, places first a DLE character and then a STX character on the line. Following this the text of the packet (including the control word) is transmitted. This text must consist of an even number of 8-bit bytes. Further, considering each pair of bytes as a 16-bit word, the less significant (right) byte is sent first. At the end of the text, the hardware appends another DLE character followed this time by an ETX character.

\*In particular, we describe a transmission from the point of view of the IMP in terms of the hardware available to the IMP. While a description of the transmission must be identical from the point of view of the Host, the method the Host uses to construct the transmission may be different. For instance, the Host may choose to implement in software much of what the IMP implements with hardware.

As the text of a packet is being transmitted, a 24-bit cyclic redundancy check (CRC) is computed by the hardware. The details of this process are described in Section F.3.2. After the final ETX of the packet has been put on the line, the 24-bit redundancy check is then transmitted in three 8-bit bytes. Following this at least two SYN characters will be put on the line. If another packet is ready to be sent, a new DLE/STX sequence will begin it. Otherwise the stream of SYN characters will continue.

One further sophistication is included. Packets are of variable length, thus the receiver must be provided with a method of uniquely identifying the end of a packet. The end is, of course, marked by the DLE/ETX sequence. However, inasmuch as the bytes of information within a packet may consist of transparent binary data, it is possible for a sequence of bytes within the text to look like the DLE/ETX which marks the end. In order to break up such sequences, the transmitting hardware scans the sequence of text bytes for DLE characters. If such a code is found, the hardware inserts an extra DLE between the DLE found and the next byte of the packet. In short, all DLE characters which appear within the text are doubled by the hardware. This, of course, excludes the DLEs of the DLE/STX and DLE/ETX pairs or any DLE that might chance to get constructed as part of the CRC.

The receiver uses the following set of rules to deal with this format: When the receiver is first turned on, it works in a SEARCH mode wherein it scans the line bit by bit with a moving window, looking for SYN characters. As soon as a SYN character is detected in the window, the receiver leaves the SEARCH mode and thereafter steps, 8 bits at a time, from byte to byte. Ensuing bytes should either be more SYNs or a DLE. (The arrival of any bytes other

than SYN or DLE puts the receiver back into SEARCH mode.) Once a DLE character has been received, the ensuing byte must be a STX or once again the receiver returns to the SEARCH mode. After the STX has arrived, the receiver enters a MESSAGE mode in which any set of bytes is acceptable.

Any errors prior to this point (e.g., front end framing errors such as the presence of a non-STX after the initial DLE) are not flagged to the receiver program. Beyond this point, however, data will be entered into the receiver memory and any errors which occur must be indicated to the program which processes the input buffer.

In the MESSAGE mode if a DLE character is found in any byte the receiver enters an ESCAPE mode in which it expects one of two things; either a second DLE, which it throws away, or an ETX which indicates the true end of the packet. (Arrival of any other character indicates an error and an error flag must be set for the receiver program. In such a case, the hardware returns to the SEARCH mode.) After the ETX character has been received, the hardware draws the three ensuing bytes through the CRC receiving logic. After the third byte has been shifted in, the check register will contain all zeros if the message was correctly received. A completion flag (interrupt) is set for the program at this point. If the check register does not contain zeros, the error flag is also set. The hardware expects ensuing bytes to be SYN characters, and the entire process of absorbing a message starts over again.

Note that the doubling of text DLE's results in an increase in the length of the packet as it occurs on the line. In the extraordinary case where a packet consisted entirely of DLE

characters, the length of the packet would be doubled on the line. Of course, as it arrived at the receiver, the removal of the extra DLEs would reduce the packet to its original length.

In the receiver, in order to avoid missing packets, the program must provide a new buffer between the time of the completion interrupt at the close of one packet and the point within the ensuing packet where the hardware must first enter data into the memory from the text of the new message. The completion interrupt should be provided to the program at the end of the third check byte. Assuming a minimum inter-packet gap of two SYN characters, the program must field the interrupt and be ready for a new input within the time required to transmit two SYN characters, the DLE/STX and the number of bytes which fill one of the receiver's words. If a new buffer has not been provided by that time, the new packet should be discarded by the hardware, which may then return to SEARCH mode. Although for reasons of efficiency the receiver should try to avoid missing packets, any missed packets will eventually be retransmitted by the RTP logic.

#### F.3.2 Character Codes

SYN	00010110
DLE	00010000
STX	00000010
ETX	10000011

All bytes (data bytes too) are transmitted *least significant (rightmost)* bit first.



### F.3.3 The Cyclic Redundancy Check

The CRC check register is cleared at the transmitting site at the beginning of the DLE character which signals the beginning of a packet. All of the bytes from (including) this opening DLE through the final ETX (including both DLEs of a doubled pair) go into the makeup of the 24-bit checksum. After the final ETX, the 24 bits of checksum are shifted onto the line. In the accompanying explanatory figures (F-8 and F-9), for the sake of simplicity, time is divided into periods called CKTIME and  $\overline{\text{CKTIME}}$ .  $\overline{\text{CKTIME}}$  consists of the period during which the checksum is being computed; that is, the time from the beginning of the initial DLE at the front end of the packet through the closing ETX at the end of the packet. CKTIME is the 24 bit times during which the checksum is transmitted or received.

Figures F-8 and F-9 show the output and input sections. In both cases shifting takes place to the right; clocking, which is not shown, is once per bit time. The numbered boxes are flip-flops whose logic true outputs appear at their tops and whose inputs appear at their bottom. Logical boxes are as follows:

+ = OR  
 • = AND  
 ⊕ = EXCLUSIVE OR

No attention is paid to electrical polarity — all expressions are in terms of logic true values. Transmission of data and check bytes is least significant (i.e., rightmost) bit first.

The output and the input sections compute in the same way during  $\overline{\text{CKTIME}}$ . The difference is that during CKTIME, the output

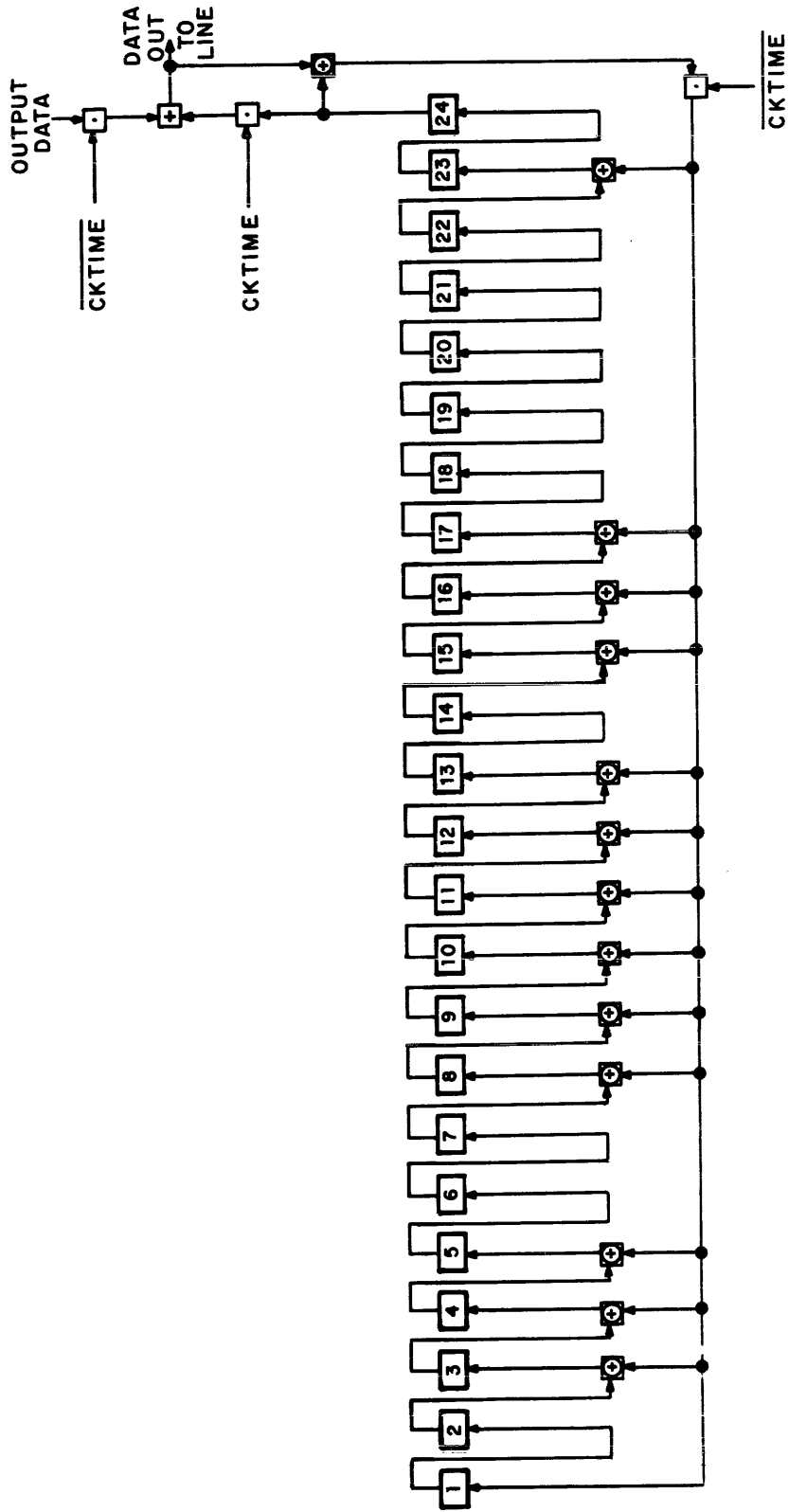


FIG. F-8 OUTPUT CHECK REGISTER

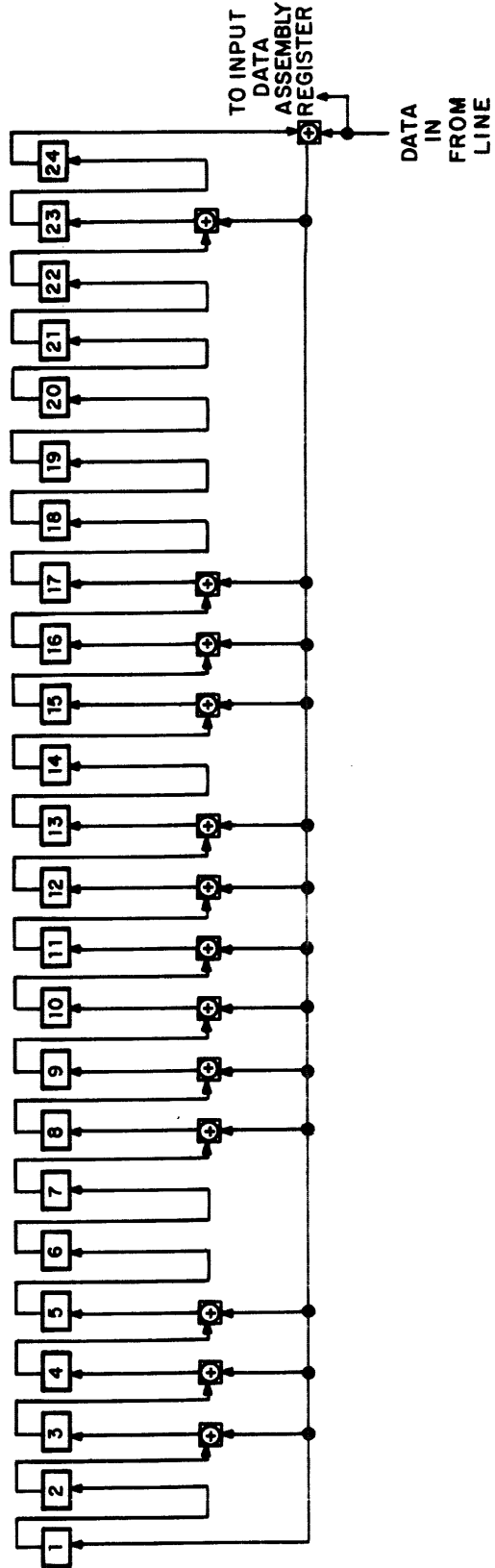


FIG. F-9 INPUT CHECK REGISTER

version stops computing and shifts what it has computed onto the line whereas on input shifting continues right through to the end of the CRC at which point the check register contains all zeros if the packet was received correctly.

Output works as follows: At the beginning of a transmission (i.e., just as the DLE is readied for transmission), the 24 flip-flops of the output check register are cleared. During  $\overline{\text{CKTIME}}$ , the bits of data which are transmitted are fed directly to the data out line. The data out line is EXCLUSIVE OR'd with the rightmost bit of the check register and the output of the EXCLUSIVE OR feeds back to EXCLUSIVE OR taps into the various bits of the check register as shown in Fig. F-8. Thus, the CRC is built as the data is shifted out.

After the last bit of data (i.e., the last bit of the closing ETX) has been shifted onto the line, the CRC device proceeds to the CKTIME stage. During CKTIME the output of bit 24 of the check register is gated to the line as the register continues to shift. The feedback path is effectively decoupled by the AND gate in the lower right hand corner of Fig. F-8, causing a logical zero to feed the bottom input of each of the EXCLUSIVE OR taps, thus turning the check register into a straightforward shift register.

Input, shown in Fig. F-9, is simpler. The input check register is cleared at the end of every arriving SYN character *before a packet* starts arriving. Starting with the opening DLE, the bits of the packet flow into an EXCLUSIVE OR together with the output of bit 24 of the check register. The output of this EXCLUSIVE OR in turn feeds EXCLUSIVE OR taps back into the shift

register just as in the case of output. This process continues through the bits of the incoming check characters at the end of the packet. After the last bit of checksum has thus been drawn in, the check register will contain a zero if no errors have occurred in transmission. (That is, no errors of the sort that this particular check detects.)

The characters are drawn through the check registers as they appear on the line (i.e., least significant bit of all bytes first, less significant byte of data byte pairs first).

#### F.3.4 Connection to a Modem

A very distant Host communicates with an IMP via communication circuits such as those provided by the phone company. Synchronous modems and dedicated full duplex lines are required. Either an EIA RS232C interface or the special Bell 303 interface can be used. Speeds up to 230.4 kilobits/second are permitted. Arrangements must be made ahead of time with BBN to allow for procurement of a proper IMP-modem cable. At the IMP end, the hardware interface between the modem and the IMP is *logically* identical to the interface which is used with the Bell 303/50 kilobit modem on the lines that interconnect IMPs, with the exception that the mark and space convention is inverted for characters sent to the modem (i.e., binary "one" equals high current) - the control level lines are not inverted.

At the Host site there will be a mating full-duplex modem. Arrangements must be made through BBN in order to guarantee that the proper strapping arrangements are provided for controlling that modem. The particular type of connector and definition of

signals on the pins of this connector will vary from one modem to another. The only signals which are of direct interest to the Host are transmit and receive data and clock signals. Since only dedicated lines may be used, other control signals are tied off either internally within the modem or via lines to the modem bearing fixed signals. The electrical specifications for drivers and receivers must be obtained from the modem manufacturer.



APPENDIX G  
IMP POWER WIRING CONVENTION



Wiring conventions used for twistlock connectors vary depending on application and locality. The convention used for 316 and 516 IMPs is illustrated in Figures G-1 and G-2. Wire colors shown for the Hubbell 3331G plug are those found on the IMP line cord. Colors shown for the 3330G receptacle are as found in most electrical installations. The terms HOT, NEUTRAL and GROUND refer to standard three-wire 110 volt a.c. power wiring, where HOT is the 110 volt a.c. lead, NEUTRAL is the 110 volt a.c. return, and GROUND is conduit ground. It is important that the IMP power receptacle be properly wired. Connecting an IMP to an improperly wired receptacle may damage the IMP or create a potential shock hazard, or both.

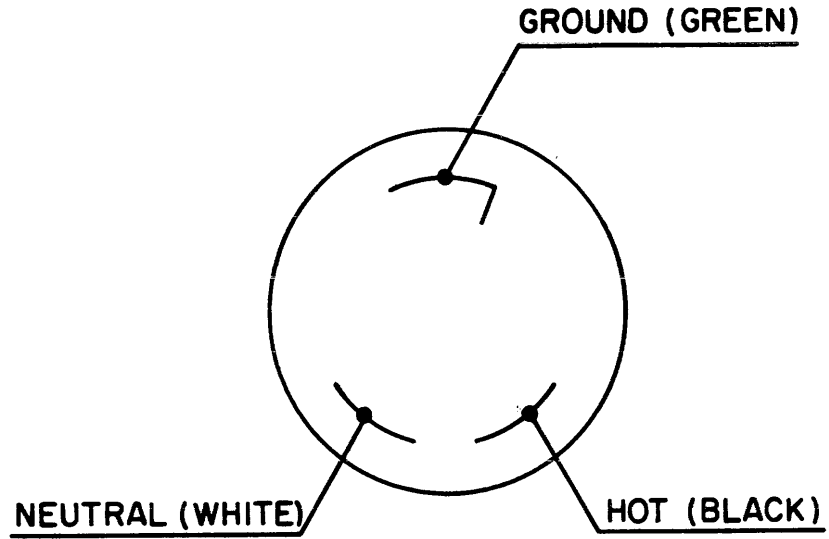


FIG. G-1 TWISTLOCK PLUG (VIEWED FROM PIN SIDE)

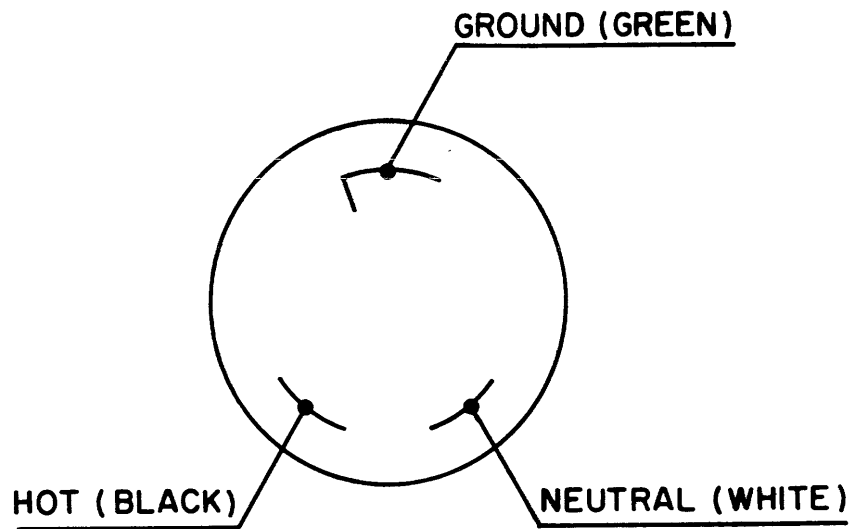


FIG. G-2 TWISTLOCK RECEPTACLE (VIEWED FROM SOCKET SIDE)



APPENDIX H  
INTERFACING A PRIVATE LINE INTERFACE (PLI)  
TO AN IMP AND A HOST TO A PLI

## H.1 PHILOSOPHY OF THE PRIVATE LINE INTERFACE (PLI)

The Private Line Interface or PLI provides alternative methods of connecting an IMP to a data source or sink (e.g., a Host). The PLI is required when the data must be transformed between the source/sink and the IMP.

Typical transformations include:

1. Encryption/decryption of data between a Red (secure) Host and a Black (unsecured) IMP.\*
2. Insertion/deletion of network protocol information between an IMP and a bit stream source/sink (e.g., a vocoder.)
3. Combinations of the above; e.g., encryption/decryption between a Red vocoder and a Black IMP.

In any case the PLI appears to the network (i.e., the IMP) to be a normal Host. In the first case the PLI appears to the application Host to be an IMP. In the latter cases the PLI appears to the application "Host" to be a communications circuit. Since the transformation that the PLI performs on data upon its entry to the network must be untransformed before exit from the network, the PLIs must be used in pairs, as is illustrated in Figure H-1.

The PLI is constructed using the Pluribus computer technology used also for Pluribus IMPs. Since the PLI also must appear as both a Host (to the IMP) and an IMP (to the actual Host), naturally many of the interfacing considerations for the PLI are very

---

\* In a "Black" environment, sensitive (i.e. classified) information has to be encrypted so that unauthorized users cannot make use of the data even if they obtain it.

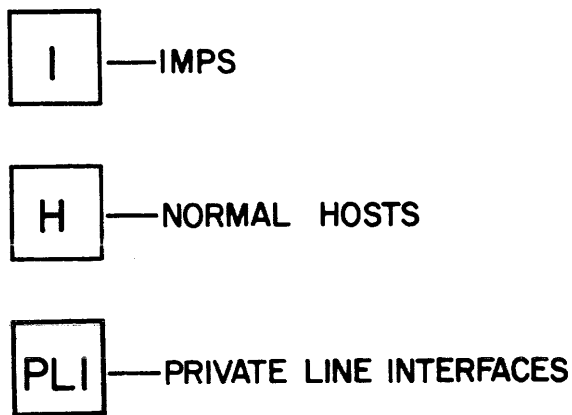
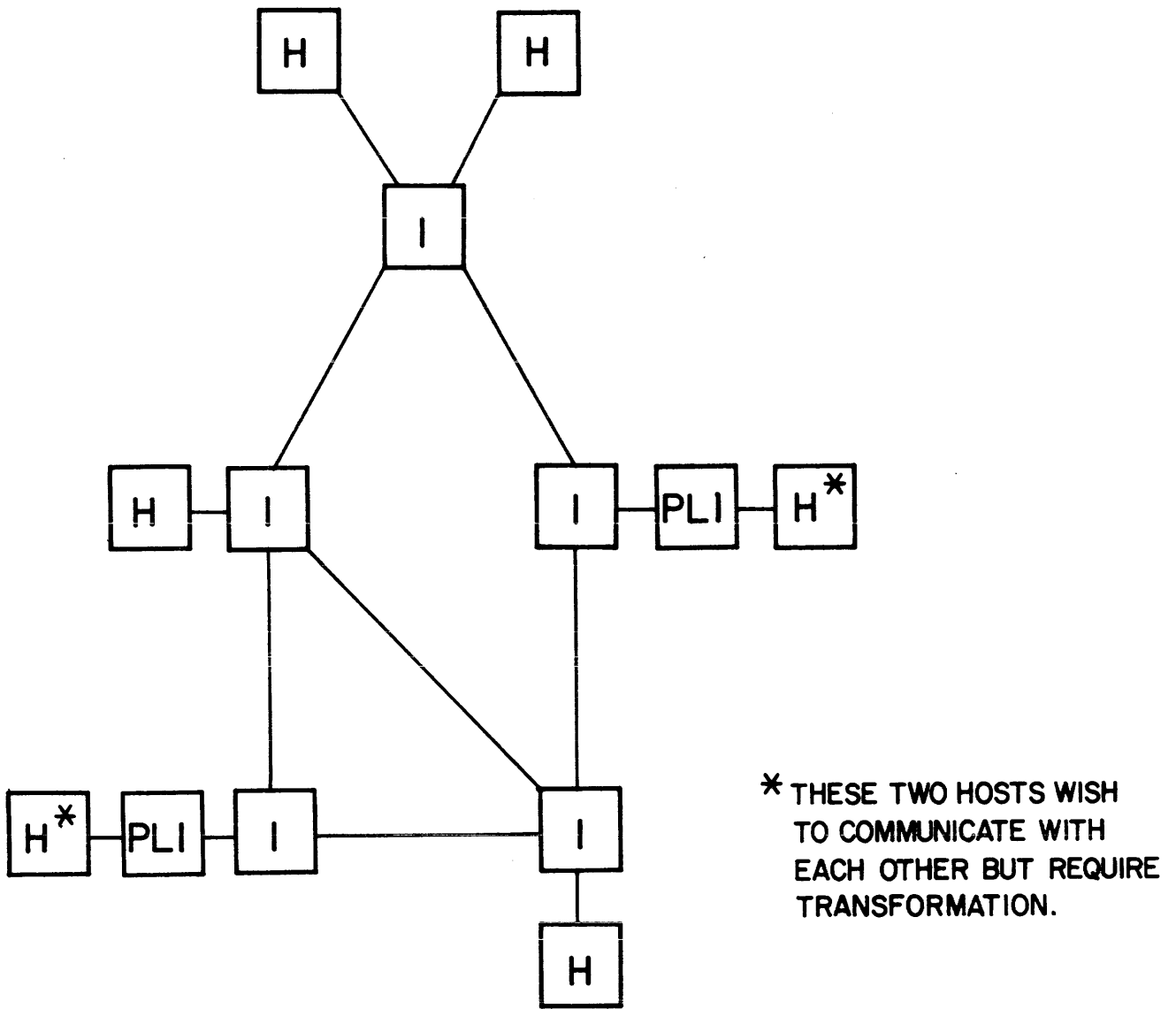


Figure H-1 Example PLI and Network Configuration

similar to the interfacing considerations for a Pluribus IMP, as discussed earlier in this document.

For convenience in referring to the two basic types of PLI, we have called them PLI/1 (or "secure PLI") and PLI/2 (or "bitstream PLI").

## H.2 SECURE PLI FUNCTIONAL SPECIFICATION

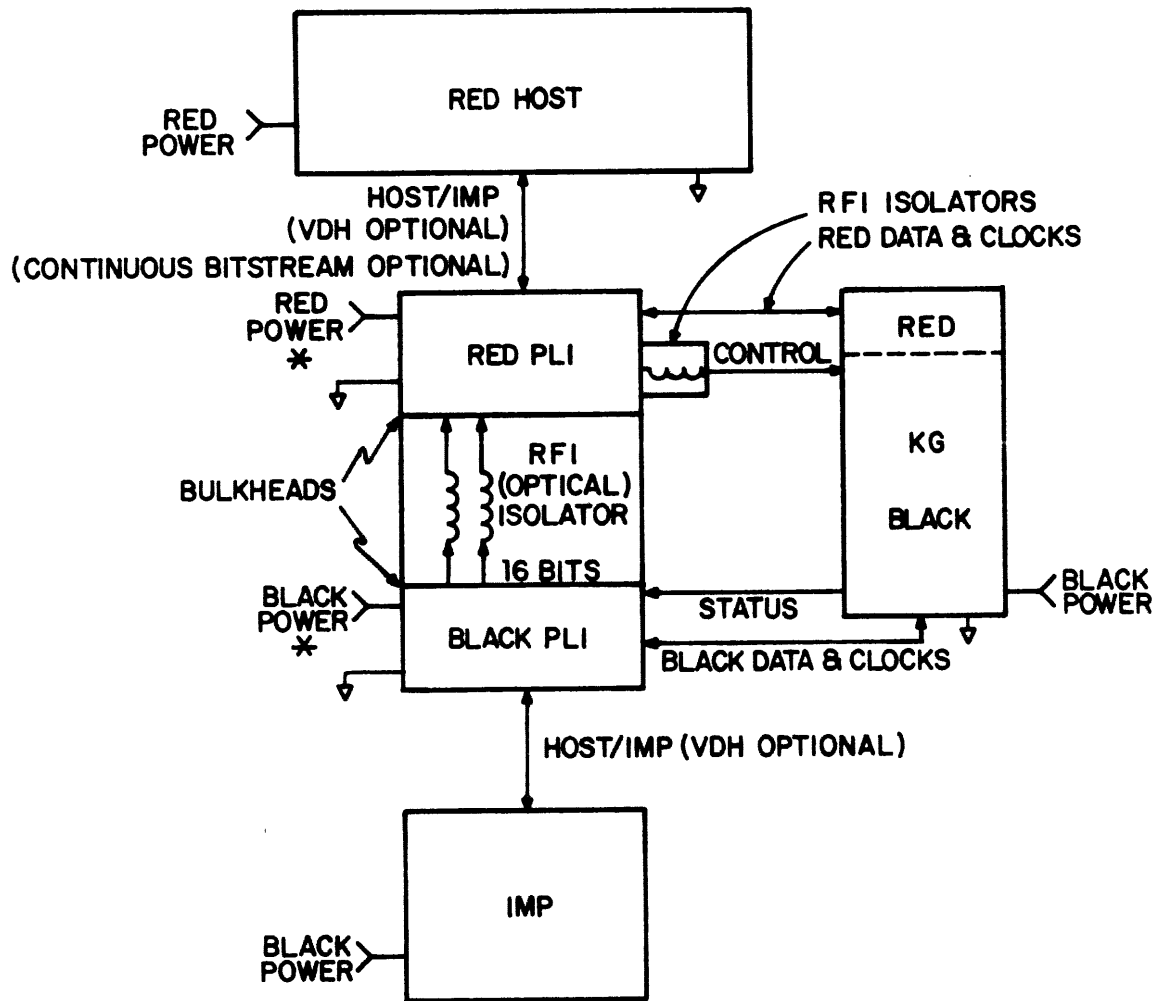
There are two main problems which had to be solved to make the PLI suitable for installation in a secure environment. These problems are: (1) whether or not the Key Generator (KG) has to be in series between the secure Host and the network, thus providing complete Red to Black (unencrypted to encrypted data) separation; and (2) the extent to which TEMPEST (leakage of secure data by radiation) considerations have to be handled within the PLI rather than counting on a shielded environment.

In both cases the most conservative approach has been taken: namely, the KG is in series providing complete Red to Black separation (although retaining a 16-bit Black to Red data path over which necessary control messages can be sent), and the PLI is self-contained with regard to TEMPEST considerations. Figure H-2 illustrates the conservative design adopted.

A PLI/1 may be used to transmit secure Host-Host traffic over the network. It requires several data interfaces: first, a Host-PLI interface which makes the PLI appear to the Host to be an IMP; second, separate interfaces to the Red and Black halves of a KG-34 key generator unit; third, a PLI-to-IMP interface which makes the PLI appear to the IMP to be a Host.

Specifications for the Host-PLI interface and the PLI-IMP interface are generally identical to the specification of Host-IMP communications. The standard Pluribus IMP-Host interface and the Very Distant Host interface specifications are used.





\* ONLY ONE REQUIRED. PLI CONTAINS ITS OWN POWER FILTER.  
 ALL CABLES AND CABINETS FULLY SHIELDED.

Figure H-2 Secure PLI Configuration

Each packet sent between PLIs includes a three-word leader which contains the packet length in bytes, a sub-routing word to allow for a number of PLI interfaces at each end, a word used to hold the original message identifier for the Host-PLI interface, and various control bits.

A PLI/1 actually comes in halves. The secure Host communicates with the Red-half PLI via a normal Host-PLI interface. The Red-half PLI signals the KG-34 unit to generate and send a key-sequence. The Black-half PLI supplies clocking to the KG-34. It scans the incoming data stream for a key-sequence, compresses the key, and stores it as the first few bytes of the message. Then the Red-half PLI signals the KG-34 unit to generate and send a key-sequence. The Black-half PLI supplies clocking to the KG-34. It scans the incoming data stream for a key-sequence, compresses the key, and stores it as the first few bytes of the message. Then the Red-half sends a message segment (initially somewhat less than 100 bits) through the KG unit. The segment is padded with SYN characters if needed to bring it to a fixed size. The Black-half adds this data block to the key, assigns a fixed destination, message identifier, and sub-routing code, then transmits it over the network. It also notifies the Red-half of the message identifier assigned via a uni-directional data and status link (from Black to Red). When a RFNM or other status indicator is received from the IMP the Red-half is notified over this same data link.

When the Black-half receives an encrypted message from the IMP, it preps the KG-34 receiver, expands the key-sequence in the message, then transmits the data through the KG unit to the Red-half. Since the Black-half provides a clocking signal,

the Red-half must be prepared to accept the decrypted data as fast as it is sent. There is no way for the Red-half to indicate overrun or other errors to the Black-half, although errors are reported to the Host. This design requires the Host-Host protocol to effect retransmission should errors of this type occur.

### H.3 BITSTREAM PLI FUNCTIONAL SPECIFICATION

Currently, it is sometimes difficult for certain existing systems, or some planned "simple-minded" systems, to take advantage of the ARPANET technology. For such installations, even the effort of integrating the relatively simple IMP/Host Protocol (described in this document) into their systems presents a considerable burden. One purpose of the bitstream PLI is to eliminate this problem and open the network to these potential users, who could then use it in lieu of a point-to-point communication circuit.

We have approached this problem by designing the PLI to appear to a source system as some standard modem which the system's software (and hardware) is already able to service. In particular, we designed an interface which will appear to be a standard, full duplex Bell System type-303 modem. We also provided a standard RS-232 or MIL-188C interface for the PLI.

In order to make more efficient use of the network, and hence decrease the resultant costs, the synchronous interface is designed to take advantage of the fact that many users of synchronous modems utilize SYN characters to fill the line when they have no data for it (i.e., to indicate an idling state). With code for the PLI to determine the actual message boundaries in the source's bit stream\* the PLI will be able to automatically elide all SYNs between messages, eliminating the expense of sending the inter-message padding through the network. Similarly

---

\* The feasibility of this is strongly dependent upon the line protocol the source is using. For example, it is simple if the source sends only messages of some fixed length.

on output, if a network delay should cause an interruption in the data stream, the PLI will "cover" the interruption by sending SYN's until the next message arrives.

Where SYN insertion is not permitted, the PLI will "stop the clock" when it either can accept no more input (because its internal buffers are full) or has no more output to send (because the next message has not yet arrived). Standard synchronous modems provide the data clock for both input and output. Thus, suspending the clock in one direction or the other presents few problems to the PLI, and our preliminary investigations indicate that usual user interfaces are immune to an occasional suspension of the clock.\*

The software in the PLI will drive the attached system's interface, breaking up input into messages for network transmission and concatenating received messages to reconstruct the bit stream for output. The PLI will also handle all of the IMP/Host protocol or, if necessary, the VDH protocol. Thus, a facility could use a pair of PLIs to replace an already existent private line and pair of modems with no other impact than a probable improvement in the line's apparent reliability and a decrease in communications costs. Of course, the transit delay would be increased and of variable length.

We have designed the bitstream PLI in constant awareness of the fact that one of the most important properties of the PLI should be its flexibility. With a relatively small hardware repertoire, a PLI will be able to appear to a system as almost

---

\* Indeed, the protection circuits in such interfaces appear to concentrate on preventing the modem from running too fast, rather than too slow.

any standard modem. The software, however, allows the flexibility to provide for a great deal more variety. There are a large number of potential options: The PLI is switchable to a VDH Network connection; the PLI could maintain two or more independent source bit streams over the single interface, and the bit streams could be directed to distinct destinations; the PLI can have various buffering strategies to match the attached systems' needs (e.g., the data could incur only a fixed delay, but portions might occasionally be lost; or the data transmission could be "guaranteed", but the delays it incurred would vary). Further, it is easy to enable and disable the various options, allowing the users of an attached system to experiment in order to determine the correct set to match local needs.

#### H.4 SECURE PLI PHYSICAL CHARACTERISTICS

The secure Private Line Interface is contained in a TEMPEST-approved rack, approx. 66H x 25W x 29D, as shown in Figure H-3. The total weight of the system is between 600 and 700 lbs. The top half of the rack contains the Red portion of the PLI and the bottom half of the rack contains the Black portion of the PLI along with a paper tape reader. The reader can be used to load programs into either half (with the rack doors open). A horizontal bulkhead separates the two halves of the rack; a filter box containing optical isolators and TEMPEST filters is provided in the bulkhead. Each half of the rack contains space to allow an additional Pluribus computer chassis. Consequently the rack is considerably larger than the minimum required size for current configurations of the PLI.

A sealed symmetrical powerline filter in the base of the enclosure allows the PLI to operate from a single power source, either Red or Black, at the convenience of the installing site. The enclosure has been designed, tested, and certified for installation in either a TEMPEST-Red or TEMPEST-Black environment, provided that the appropriate signals are contained in conduit as specified below.

The PLI is designed to interface with an externally located KG-34, which must have the following options:

- (1) 110 Volt AC power
- (2) Low Speed
- (3) Message Indicator; no A/S
- (4) Data transition on positive clock transitions. (See the KG-34 manuals for strap option on two KG cards.)
- (5) Eight bit MI pattern (two front panel switches).

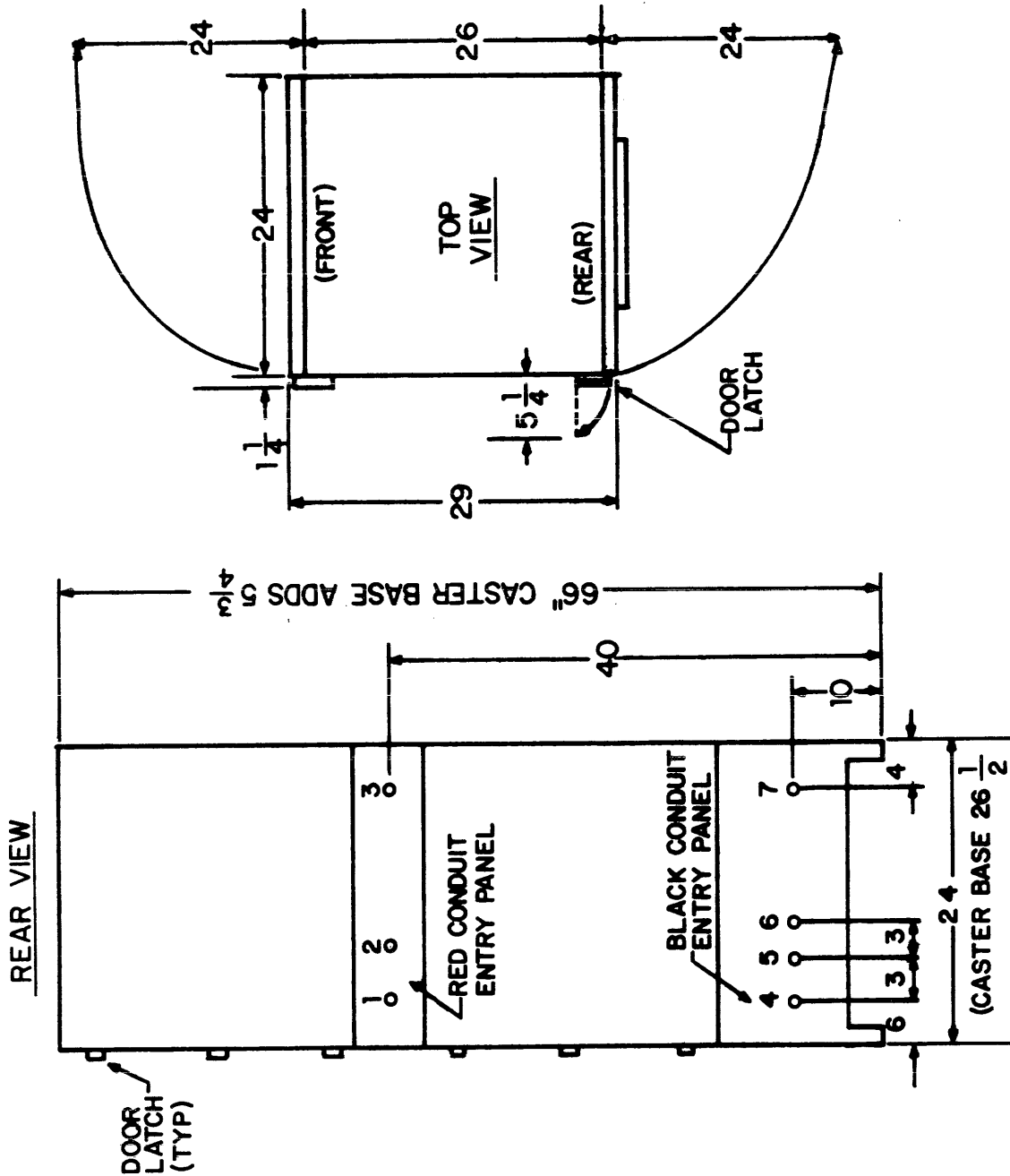


Figure H-3 Secure PLI Outline Dimensions



## H.5 SECURE PLI CABLE ENTRY AND CONDUIT

All connections to the PLI are made through one of two (one Red; one Black) plated conduit entry panels at the rear of the enclosure (see Figure H-3). The external conduits should be routed in such a way that they do not interfere with the rear doors. All conduit holes are 1 1/8 inch in diameter, as required for normal 3/4" ID conduit bulkhead fittings. The use of RF gasketing material is advisable to ensure TEMPEST integrity.

No conduit, fittings, or gaskets are supplied with the PLI. Unless prior arrangements have been made, the cables furnished with the PLI are of the lengths shown in Table H-1. Connectors for the PLI end are attached to the cables, and where applicable, connectors are furnished for the other end, to be attached after the cables have been pulled through the installed conduits. The drawings specifying these connections are listed in Table H-1. Although all cables to the PLI may be installed in conduit, in a TEMPEST Red environment, only the Black signals are required to be in conduit. Similarly in a TEMPEST Black installation, only the Red signals are required to be in conduit. Grommets or similar devices should be installed for physical protection of cables where conduit fittings are not used.

### H.5.1 AC Power

The AC cord shipped with the PLI is primarily for pre-installation checkout after unpacking. Prime power for the PLI should be directly wired from a dedicated 20A 120VAC circuit breaker to the appropriate AC outlet box within the PLI. Entry "3"

PURPOSE	Cable DWG #	Length Furnished	Destination	PLI Conduit No.	PLI Connector	PLI Interface Card
Prime Power if Red (temporary cable furnished for checkout)	--	10 ft	AC Socket	3	None	None
Prime Power if Black	--		AC Socket	7	None	None
PLI to 316 IMP - Local Host	FHSH	30 ft	IMP	4	Black J-7	HLC
PLI to 316 IMP - Distant Host	*		IMP	4	Black J-7	*
PLI to (IMP) Modem - Very Distant Host	FMLA	30 ft	303 Modem	4	Black J-7	MLX/MLR
PLI to Pluribus IMP - Differential Driver Host	*		Pluribus IMP	4	Black J-7	*
Red Host to PLI - Local Host	FHSA	30 ft	Host	1	Red J-7	HLC
Red Host to PLI - Distant Host	*		Host	1	Red J-7	*
Red Host to PLI - Very Distant Host	FMLA	30 ft	303 Modem	1	Red J-7	MLX/MLR
Red Host to PLI - Differential Driver Host	*		Host	1	Red J-7	*
Red Synchronous Data Source to PLI non-303	FEIH	50 ft	Synchronous Interface	1	Red J-7	SMS
Red Synchronous Data Source to PLI 303 Modem	FMLA	30 ft	303 Interface	1	Red J-7	SMS
KG-34 to PLI - Low Speed Signals	FKGA	30 ft	KG TB-4	5	Black J-6	KGB
KG-34 to PLI - Black Clock and Data	4 FBNC-B	30 ft	KG TB-1	6	Black J1-J4	KGB
KG-34 to PLI - Red Clock and Data	4 FBNC-B	30 ft	KG TB-3	2	RED J1-J4	KGR

\* Not yet available

Table H-1 Cable Tabulation - Secure PLI

should be used for RED power, or entry "7" should be used for BLACK power, at the convenience of the site. Only one feed should be used, as the in-line power filter in the PLI supplies power to its other half. Input line voltage to the PLI should be maintained above 115VAC to allow for the drop through the powerline filter. A separate convenience outlet should be provided near the PLI for an oscilloscope (not furnished) and the Teletype furnished for diagnostic use. The unused AC conduit entrance must be sealed off in TEMPEST approved fashion.

#### H.5.2 IMP Connection

The Black half of the PLI acts as a network Host and is connected to the IMP using a Local Host, Distant Host, or Very Distant Host interface. See Figure H-4. The external cable (see Table H-1) is connected to J-7 within the Black half of the PLI.

1. Local Host. The HLC interface card is used in the PLI when it is connected as a Local Host to its IMP. Although 30 feet has been specified as the maximum cable distance between Host (PLI) and IMP, distances of several hundred feet are made practical by the use of coax cables and special attention to installation ground connections. BBN engineering personnel should be consulted well in advance of site installation.

2. Distant Host. (Soon to be available.) Where grounding problems induce substantial common-mode voltages, or nearby equipment causes severe differential noise problems, the IMP should be provided with the Distant Host Interface, specified in this report, Section 4.5.2; the mate to it will be provided

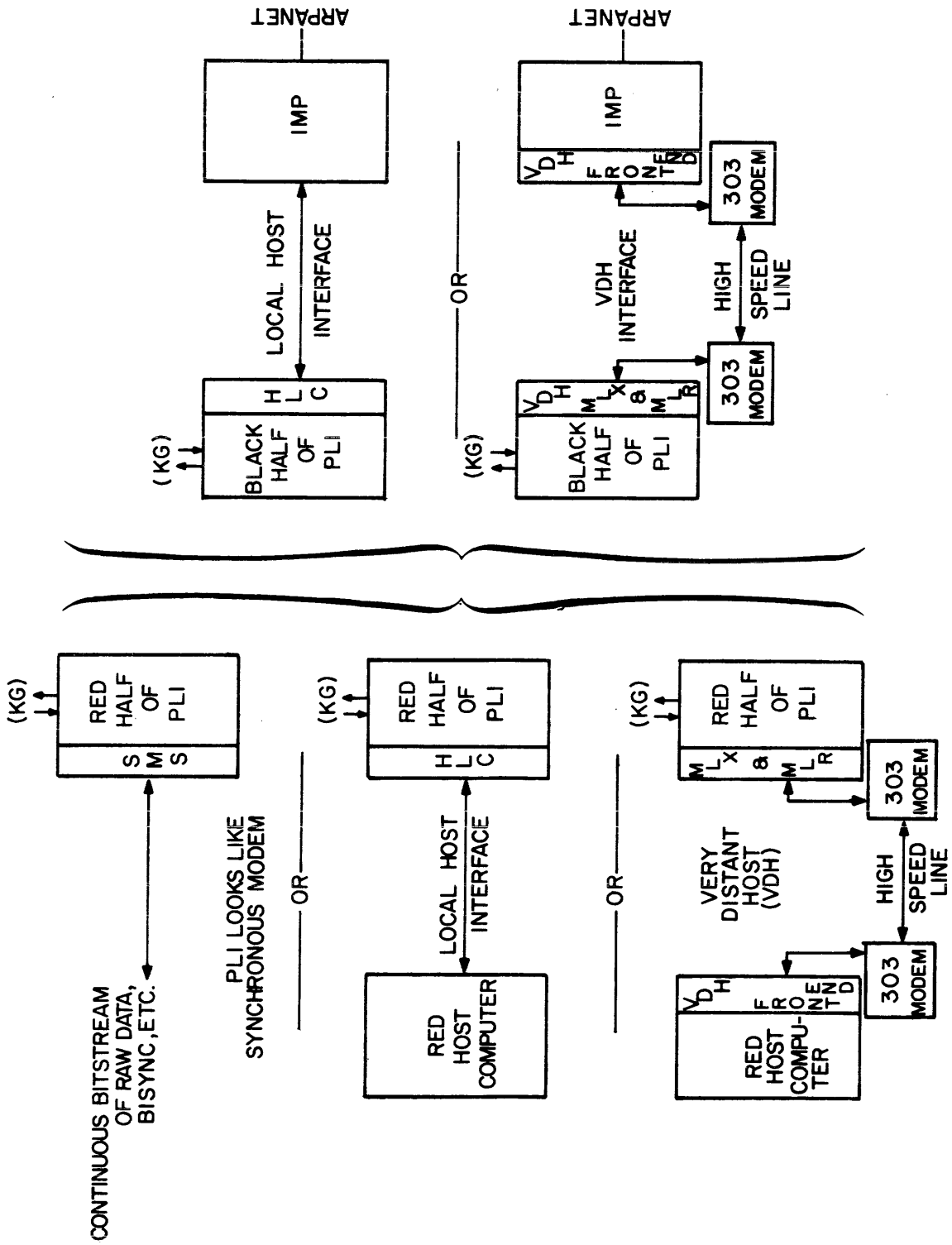


Figure H-4 Possible Secure PLI Interface Configurations

in the Black half of the PLI. Where possible, the Local Host interface should be specified to connect to a 316 IMP unless there is strong evidence that difficulties will be encountered. Connection to a Pluribus IMP should use the Differential Driver Host Interface at each end.

3. Very Distant Host (VDH). The VDH interface is a communication-line protocol using high-speed synchronous modems (typically Bell-303 or various commercially available modem eliminators). It is completely specified in Appendix F of this report, and should be used where the cable distance between IMP and PLI is sufficiently long that direct-wire connection (Local or Distant Host) is impossible. Note that proper TEMPEST precautions must be taken for both modems and their data lines if in a Red area.

If the 303 modem (or substitute) is less than 30 cable feet from the PLI, the cable will be provided with the coax inserts on the modem end so that it can be pulled through the conduit from the PLI, and then the inserts can be seated in the Burndy connector block (provided) without the use of special tools. Since a special tool is required for removing them, it is advisable to inform BBN as soon as the required cable length is known.

### H.5.3 Red Host or Data Connection

Several interfaces between the PLI and the Red data source are available. See Figure H-4. If the Red data source is an ARPANET Host, the Red half of the PLI looks to it like an IMP. In Sections 3 and 4 of this report the software and hardware

requirements for the Host are specified. Where raw synchronous data or a computer without Host protocol is to be interfaced, the Continuous Bitstream Interface is used. In all cases, connector J-7 and conduit hole 1 will be used for the cable.

1. Local Host. (identical to discussion in Section H.5.2.1)
2. Distant Host. (identical to discussion in Section H.5.2.2)
3. Very Distant Host. (identical to discussion in Section H.5.2.3) Note that both the hardware and software requirements imposed on the Red Host are considerably greater than with the other Host interfaces.
4. Continuous Bitstream Interface. The BBN Synchronous Modem Simulator (SMS) interface is provided for applications where data communication would otherwise be carried out with synchronous modems and a leased line. Such applications include seismic, acoustic, and digitized speech data.

The electrical interface may be one of the following:

RS-232 (EIA), CCITT, MIL-188, or Bell 303.

The desired option must be specified prior to delivery in order that the correct interconnecting cables can be furnished. The cable is connected to J-7 within the Red half of the PLI, and the pin and wiring assignments are specified in the drawings designated in Table H-1. Note that although the RS-232 interface is often used over several hundred feet, under worst case

conditions, 50 feet can be the practical limit, and special precautions should be discussed with BBN engineering personnel.

#### H.5.4 Key Generator Connections

Two types of signals (in three groups) connect the KG-34 to the PLI.

Type 1. All low-speed control and status (indication) signals.

These are contained in a single multiconductor cable (first group) attached to connector J-6 in the Black half of the PLI through conduit opening 5. Installations requiring cables longer than 200 feet may require the use of larger gauge wire in order to meet the KG-34 interface specifications. In general, total DC resistance of a single conductor should be kept below 5 ohms.

Type 2. High speed clock and data signals.

BNC connectors are provided in each half of the PLI for the four Red (second group) and four Black (third group) high speed signals (conduit entrances 6 and 2 respectively for J-1 through J-4 in each half). The RG-58 coax cables provided have mating BNC connectors at the PLI end but no connector for the KG since direct connection to it is usually made on terminal strips. The use of crimp-on ferrules is suggested.

Note that the KG-34 interface specifications point out that its output circuits are not designed to drive terminated coax cables. If cable distances of more than about 150 feet are

involved, special precautions may be required, or operation at reduced bandwidth may be necessary. BBN engineering personnel should be consulted during the planning of the installation. A certain amount of fine tuning may be necessary after the installation is completed if maximum KG bandwidth is required and long cables are involved.



## H.6 BITSTREAM PLI PHYSICAL CONFIGURATION

The Bitstream Private Line Interface is contained in a single equipment cabinet (approx. 52H x 24W x 28D). Also included is a pedestal-mounted (standalone) Teletype, used for debugging and changing parameters. Cable entrance is through the open bottom of the rack, although if desired the connector panel may be relocated to provide connection from outside the rack.

The cabinet contains a 24 slot Pluribus, a standalone power supply, and a paper tape reader. A rear door provides access to the connector panel ("fantail" panel), and removable bezels provide access to the logic cards. The side panels of the rack may be easily removed when necessary. The front panel contains a keyswitch used for turning power on and off. A duplex AC receptacle containing Hubbell No. 5362 (or equivalent) sockets should be provided at the point of cable entry. A 15 Amp 115 VAC circuit will handle the PLI's power requirements.

The perforated top of the rack should remain uncovered to allow air to exhaust. Intake air is drawn in through a filter in the front of the rack. If the cable entrance in the floor is a potential source of dust it should be sealed off after cables have been installed.

Connection to the IMP will be made as specified in the appropriate line of Table H-1. Connection to the data source will be made through the standard 25-pin EIA connector (DB-25S) in the fantail panel, and a 50 foot extension cable (similarly terminated) is provided with the PLI. The use of the pins is as stated in EIA specification RS-232-C. If the Bell Series 303

Modem interface is desired instead of RS232, this must be stated when the PLI is ordered. A 30 foot FMLA cable is then supplied, which is terminated in the Burndy coax connector appropriate to a 303 modem.

## H.7 SOFTWARE INTERFACES TO THE PLI

As with the PLI hardware, there are two areas of PLI software interface, the interface from the PLI to the IMP and the interface from the Host to the PLI.

The interface from the PLI to the IMP is of little interest to the Host except for general site planning considerations; i.e., where the IMP, PLI, and Host should be located with respect to one another. As has already been made clear in the PLI interface hardware section, the PLI can reside at a distance from the IMP which will require the PLI to appear to be either a local, distant, or very distant Host to the IMP. Naturally, the PLI software interface to the IMP also supports the full spectrum of PLI to IMP distance options.

The software interface from the Host to the PLI supports a number of variations: a) bitstream vs. IMP/Host message interface; b) secure vs. non-secure interface; and c) very distant vs. non-very distant.

If the PLI/Host interface is of the bitstream variety, the very distant vs. not very distant option is irrelevant and the secure vs. non-secure option is transparent to the Host. In other words, there is no software interface for the Host to implement if the bitstream option is chosen, other than the requirement that the Host deliver to the PLI a stream of bits.

If the PLI/Host interface is of the IMP/Host message variety, then the protocol described in Section 3 is followed to the greatest extent possible. That is, every effort is made to make the PLI appear transparent (as if it were the IMP itself) to the

Host. Thus, as is the case when a Host is connected directly to an IMP and not through the PLI, the Host may be connected optionally to the PLI using the Very Distant Host protocol described in Appendix F. In the case where the non-secure option is chosen, the IMP/Host message interface between the Host and the PLI should be completely transparent from the Host's point of view whether the Very Distant Host option is used or not.

In the case where the secure option is chosen, it is not possible for the PLI to be completely transparent from the point of view of the Host. Nonetheless, an attempt has been made to make the PLI as transparent as possible, even when used in the secure mode.

The few areas where the PLI operating in a secure mode is not completely transparent from the Host's point of view all stem from the fact that in the secure mode there is no un-encrypted communication path between the Host and the IMP through the PLI. Thus, for example, the PLI in several cases has no way to convey to the IMP network information which the Host is normally able to send. We list the specific instances in which the PLI is not transparent from the Host's point of view:

1. Both priority and non-priority traffic from the Host are sent identically across the network; there is no other choice for there is no way to convey the priority information across the Red/Black interface.

2. Host Going Down messages from the Host are ignored by the PLI; there is no way to pass this information on to the IMP across the Red/Black interface.

3. Type 3 messages (see Section 3) are treated the same as type 0 messages; there is no way to pass the distinction between these two types of messages across the Red/Black interface.

4. If the IMP accepts a message from the PLI and immediately goes down, the PLI waits 40 seconds, and then the PLI returns an Incomplete Transmission message to the Host. Also, if the Black side of the PLI goes down, the Red side returns an Incomplete Transmission message to the Host, but after 50 seconds. Either of these conditions represents a solid down of the communication path from which there is no recovery which is not noticeable; thus, extra buffering does not need to be provided to cover these delays.

5. The network address supplied by the Host will be specified by BBN at installation time. They will very likely not be the same as real network addresses, but will of course have the same format.

6. The bandwidth between the Host and the IMP through the PLI will very likely be limited to something less than the many hundreds of kilobits per second possible over a normal IMP/Host interface or the 50 kilobits per second possible over a Very Distant Host interface. Testing of the maximum possible bandwidth is not yet complete, but presently appears to be limited to something on the order of 20 to 30 Kbs. Limiting

factors include KG prep rate, limited buffering in the PLI, lack of flow control on the Black to Red transmission paths, and available network bandwidth for single packet messages.

7. The Host packet size is adjustable by BBN when the PLI is installed. The PLI expands each packet it receives by at least 100 bits to a fixed size, normally the standard ARPANET packet size. Each packet is sent over the network as a separate message and reassembled at the destination. This means there is no limit to the length of a Host message. If VDH is being used, knowing the packet length becomes important. If it is inconvenient to modify the Host VDH to send shorter than standard packets, the PLI can be modified to send two-packet messages over the network, as long as VDH is not also being used between IMP and PLI.

Despite the cases of non-transparency listed above, in most cases a Host tested by direct connection to an IMP should be able to later operate connected to the IMP through a PLI with only trivial (if any) software modifications.