48-bit Absolute Internet and Ethernet Host Numbers

by Yogen K. Dalal and Robert S. Printis

Abstract: Xerox internets and Ethernet local computer networks use 48-bit absolute host numbers. This is a radical departure from practices currently in use in internetwork systems and local networks. This paper describes how the host numbering scheme was designed in the context of an overall internetwork and distributed systems architecture.

CR Categories: 3.81.

Key words and phrases: names, addresses, absolute host numbers, internetwork, multicast, Ethernet, local computer network, network management.

To be presented at the *Seventh Data Communications Symposium*, Mexico City, October 1981.

1. Introduction

The Ethernet local computer network is a multi-access, packet-switched communications system for carrying digital data among locally distributed computing systems [Metcalfe76, Crane80, Shoch80, Ethernet80, Shoch81]. The shared communications channel in the Ethernet system is a coaxial cable—a passive broadcast medium with no central control. Access to the channel by stations or *hosts* wishing to transmit is coordinated in a distributed fashion by the hosts themselves, using a statistical arbitration scheme called *carrier sense multiple access with collision detection* (CSMA/CD). Packet address recognition in each host is used to take packets from the channel.

Ethernet packets include both a source and a destination *host number*, that is, the "address" of the transmitter and intended recipient(s), respectively. Ethernet host numbers are 48 bits long [Ethernet80]. 48 bits can uniquely identify 281,474,977 million different hosts! Since the Ethernet specification permits only 1024 hosts per Ethernet system, the question that is often asked is: "why use 48 bits when 10, or 11, or *at most* 16 will suffice?" This paper answers this question, and describes the benefits of using large absolute host numbers.

We view the Ethernet local network as one component in a store-and-forward datagram *internetwork* system that provides communications services to many diverse devices connected to different networks (see, for example, [Boggs80, Cerf78]). Our host numbering scheme was designed in the context of an overall network and distributed system architecture to take into account:

- the use of host numbers by higher-level software,

- the identification of a host or a logical group of hosts within the internetwork,

- the addressing of a host or a logical group of hosts on the Ethernet channel, and

- the management of distributed systems as they grow, evolve and are reconfigured.

Sections 2, 3, and 4 of this paper describe the pros and cons of various host numbering schemes in inter- and intra-network systems, and describe the properties and advantages of our host numbering scheme. Section 5 discusses our host numbers in the context of "names" and "addresses" in network systems. Sections 6 and 7 describe the reasons for choosing 48 bits, and the mechanisms for managing this space.

2. Addressing Alternatives

The *address* of a host specifies its location. A network design may adopt either of two basic addressing structures: *network-specific host addresses*, or *unique host addresses* [Shoch78]. In the first case, a host is assigned an address which must be unique on its network, but which may be the same as an address held by a host on another network. Such addresses are sometimes called *network-relative addresses*, since they depend upon the particular network to which the host is attached. In

the second case, each host is assigned an address which is unique over all space and time. Such addresses are known as *absolute* or *universal addresses*, drawn from a *flat address space*. Both network-specific and absolute host addresses can have any internal structure. For the purposes of this paper, we will treat them as "numbers" and will use host addresses and host numbers interchangeably.

To permit internetwork communication, the network-specific address of a host usually must be combined with a unique network number in order to produce an unambiguous *internet address* at the next level of protocol. Such internet addresses are often called *hierarchical internet addresses*. On the other hand, there is no need to combine an absolute host number with a unique network number to produce an unambiguous internet address. Such internet addresses are often called *flat internet addresses*. However, internetwork systems using flat internet addresses, containing only the absolute host number, will require very large routing tables indexed by the host number. To solve this problem, a unique network number, or other routing information is often included in the internet address as a "very strong hint" to the internetwork routing machinery; the routing information has been separated from host identification.

We anticipate that there will be a large number of hosts and many (local) networks in an internetwork, thus requiring a large internet address space. For example, the Pup-based internetwork system [Boggs80] currently in use within Xerox, as a research network, includes 5 different types of networks, has over 1200 hosts, 35 Experimental Ethernet local networks, and 30 internet routers (often called internetwork gateways). Figure 1 illustrates the topology of the internet in the San Francisco Bay Area.

If network-specific addressing is used, then the host number need only be large enough to accommodate the maximum number of hosts that might be connected to the network. Suitable installation-specific administrative procedures are also needed for assigning numbers to hosts on a network. If a host is moved from one network to another it may be necessary to change its host number if its former number is in use on the new network. This is easier said than done, as each network must have an administrator who must record the continuously changing state of the system (often on a piece of paper tacked to the wall!). It is anticipated that in future office environments, host locations will be changed as often as telephones are changed in present-day offices. In addition, a local network may be shared by uncooperative organizations, often leading to inconsistent management of the network. Duplication of addresses may lead to misdelivered or undeliverable data. Thus, the overall management of network-specific host numbers can represent a severe problem.

The use of absolute host numbers in an internetwork provides for reliable and manageable operation as the system grows, as machines move, and as the overall topology changes, if the (local) network can directly support these large host numbers. This is true, because the host is given one number or

identity when it is first built, and this number is never modified when the network configuration changes. A distributed system can be effectively managed if the special purpose parameterizing of the hardware can be reduced to a minimum. The absolute host number space should be large enough to ensure uniqueness and provide adequate room for growth.

Since an absolute host number is a property of a host rather than its location in the network to which it is connected, the number should not be associated with, nor based on, a particular network interface or controller. A host connected to zero or more networks has only one identity, which should not be "hard wired" into a particular interface or controller, but should be setable from the station (see Section 5). The address of this host on all connected networks that directly support absolute host numbers will, in general, be the same as the host's identity (see Sections 3 and 5). A host connected to a network that does not directly support absolute host numbers will, in addition, have an address relative to that network.

Such host numbers can be used by operating systems software to generate unique numbers for use by the file system, resource manager, etc. [Redell80, Abraham80]. By decoupling the host's number from the network to which it is connected, a uniform mechanism can be applied to networked and stand-alone workstations so that they may interact at higher levels. For example, both stand-alone and networked Pilot-based [Redell80] workstations may generate files that are identified by unique numbers and then exchange them by copying them onto removable storage media such as floppy disks.

Xerox internetwork systems will use flat internet addresses containing 48-bit host numbers, and a unique network number as a very strong routing hint. The internet address(es) for an object or resource in the internetwork is obtained from a distributed agent called the *clearinghouse*; it serves a function similar to the telephone system's "white" and "yellow" pages [Oppen81]. The user of the resource does not compute or determine the network number after discovering the host number of the resource—the network number is included in the binding information returned from the clearinghouse. We believe that our host number space is large enough for the foreseeable future (see Section 6). We expect that these internetworks will be built primarily from Ethernet local networks and thus directly support 48-bit absolute host numbers on the Ethernet channel. An internet packet is still encapsulated in an Ethernet packet when it is transmitted on the Ethernet channel.

48-bit host numbers lead to large Ethernet and internet packets. We believe that this will not pose a problem as both local and public data networks continue to offer higher bandwidths at reasonable costs, and the memory and logic costs associated with storing and processing these numbers continue to become lower with the advances in LSI technology.

We further justify our choice of absolute host numbers in the next section by comparing

internetwork routing techniques that use hierarchical and flat internet addresses. We show that routing based on flat internet addresses is very general, and especially efficient if the constituent (local) networks directly support the absolute host number.

3. Internetwork Delivery

In this section, we illustrate the pros and cons of using hierarchical and flat internet addresses for internetwork delivery by comparing the techniques prescribed by the Arpa Internet Protocols [IP80] and the Pup Protocols [Boggs80], with those prescribed by the new Xerox internetwork protocols.

A host is identified in an internetwork by its internet address. In general, a host may have many internet addresses, but an internet address can identify only one host.

Hierarchical internet addresses have a routing decision implicitly encoded in them because they specify the network through which a packet must be delivered to the host. This is not necessarily true for flat internet addresses. Flat internet addresses may contain routing information hints in them, and in such cases a sophisticated routing mechanism is free to use or ignore these hints.

The delivery of *internet packets* involves routing the packet from source host to destination host, through zero or more *internet routers* based on the internet address of the destination host. The internet packet usually must be encapsulated for transmission through the various communication networks, on its way from source host to destination host. The encapsulation specifies addressing and delivery mechanisms specific to that communication network. Each communication network may have a different form of internal addressing. When an internetwork packet is to be transported over a communication medium, the *immediate destination* of the packet must be determined and specified in the encapsulation. The immediate destination is determined directly from the internet address if it is the *final destination*, or through a routing table if it is an *intermediate destination*. We do not discuss mechanisms and metrics for managing routing tables in this paper.

The structure of the internet address influences the algorithms used for determining immediate destinations during encapsulation.

Consider flat internet addresses first: the absolute host number in a flat internet address may have no relation to any of the internal addressing schemes used by the communication networks. Hence, during encapsulation, as far as each of the communication networks is concerned, the absolute host number is a *name* that must be translated to an *address* on the network. This involves consulting some form of a *translation table*, possibly in conjunction with the routing table (we assume that the routing table supplies the absolute host number of the next internet router rather than its network-specific address, so that internet routers know one anothers' internet addresses should they wish to directly communicate, for the purpose of exchanging routing information or statistics, etc.). In a very general internetwork, the overhead of performing an absolute host number to internal address

translation can be large both in space and time. and also requires the maintenance of translation tables in all hosts. Xerox internetworks will consist primarily of Ethernets. Since absolute host numbers have many other advantages, we chose the internal addressing on an Ethernet system to be identical to the absolute host number to avoid translation. Therefore, as far as Ethernet systems are concerned, the absolute host number is indeed an address and not a name. When Xerox internet packets traverse other communication networks that do not support our absolute host numbers. like the Bell Telephone DDD network, Telenet, or other public or private data networks, translation tables will have to exist in the necessary hosts and internet routers to perform translation from absolute host numbers to internal addresses. We feel that this will not cause many operational problems, other than setting up and maintaining these translation tables at appropriate (and limited) hosts and internet routers. Flat internet addresses are not in widespread use because the designers of internetworks have had little or no control over the design of the constituent communication networks, and thus, have been forced to use hierarchical internet addresses, rather than flat internet address containing routing information or hints.

Flat internet addresses provide a vehicle for solving many of the hard internetwork routing problems in situations like network partitioning. multihoming, mobile hosts, etc. But they create others! These situations are described in greater detail in [Sunshine81].

A host in an internetwork that has hierarchical internet addresses has as many internet addresses as the number of networks to which it is connected. It is the encoding of the network-specific host number itself that distinguishes various schemes in this category. There are two cases, one represented by the Arpa Internet Protocols and the other by the Pup Protocols.

The Arpa Internet Protocols specify that the internet address is an 8-bit network number followed by a 24-bit host number. The host number is encoded such that it is synonymous with the internal addressing scheme of the communication network to which the host is connected. For example, a host connected to the Bay Area Packet Radio Network has a network-relative internal address of 16 bits, and therefore the host number in its internet address will contain these 16 bits in the "least significant" positions. During encapsulation, if the immediate destination is the final destination then it is equal to the host number in the destination internet address, and if the immediate destination is an intermediate destination then it is determined from the routing tables and has the right format. For such a scheme to work, the space reserved for the host number must be as large as the largest internal addressing scheme expected in any communication network. In the case of the Arpa Internet Protocols, this is already too small since it cannot encode new Ethernet host numbers!

The Pup protocols encode the host number in the internet address with only 8 bits, and so cannot be used to encode the various network-specific host numbers. The Pup Protocols were designed to be used in an internetwork environment consisting mainly of interconnected Experimental Ethernet

systems which have 8-bit internal addresses, and that is why the host number in the internet address is 8 bits long. Hence, even though the Pup Protocols use network-specific host numbers, when packets are transmitted through non-Experimental Ethernets a translation table is needed just as for absolute host numbers. For example, when Pup internet packets traverse the Bay Area Packet Radio Network, the 8-bit host number of the internet routers must be translated into the 16-bit ID used within the radio network [Shoch79].

Here is another way to look at internet addresses: whether the host number is absolute or network-specific, if it does not encode the communication network's internal addresses, then it may be necessary to translate from the internet host number to the communication network's internal address whenever the packet is to be transmitted over the network.

4. Multicast

In addition to identifying a single host, our absolute host numbering scheme provides several enhanced addressing modes. *Multicast* addressing is a mechanism by which packets may be targeted for more than one destination. This kind of service is particularly valuable in certain kinds of distributed applications, such as the access and update of distributed data bases, teleconferencing, and the distributed algorithms which are used to manage the network (and the internetwork). Multicast is supported by allowing the destination host numbers to specify either a physical or "logical" host number. A logical host number is called a *multicast ID* and identifies a group of hosts. Since the space of multicast IDs is large, hosts must filter out mulitcast IDs that are not of interest. We anticipate wide growth in the use of multicast and all implementations should, therefore, minimize the system load required to filter unwanted multicast IDs.

Broadcast is a special case of multicast; a packet is intended for *all* hosts. The distinguished host number consisting of all ones is defined to be the broadcast address. This specialized form of multicast should be used with discretion, however, since all nodes incur the overhead of processing such packets.

By generalizing the host number to encompass both physical and logical host numbers, and by supporting this absolute host number within the Ethernet system (which is inherently broadcast in nature) we have made it possible to implement multicast efficiently. For example, perfect multicast filtering can be performed in hardware and/or microcode associated with the Ethernet controller. Since logical host numbers are permitted in flat internet addresses we also have the capability for *internetwork multicast*. This is, however, easier said than done as the multicast ID may span many networks. Internetwork multicast and reliable multicast are subjects we are currently researching; an appreciation of the problems can be found in [Dalal78 and Boggs81].

5. Names and Addresses

The words "name" and "address" are used in many different ways when describing components of a computer system. The question that we often get asked is: "is a 48-bit number the name or the address of a host computer?" In the area of computer-communications we have tried to develop a usage that is consistent with that found elsewhere, and an excellent expose of the issues may be found in [Shoch79]. An important result of this paper is that a mode of identification (whether it be a number or a string of characters) is treated as a name or address depending on the context in which it is viewed.

From an internetworking point of view, the 48-bit number asigned to a host is its identity, and never changes. Thus, the identity could be thought of as the "name" (in the very broadest sense) of the host in the internetwork. According to Shoch's taxonomy, this identity could also be thought of as a flat address, as it is recognizable by all elements of the internetwork.

The Ethernet local network is a component of an internet, and was designed to support 48-bit host numbers. One could view this design decision as "supporting host name recognition directly on the Ethernet channel" (since broadcast routing is used to deliver a packet). This would be true if a host was connected to an Ethernet at only one point—a policy decision we made for the Xerox internetwork architecture. However, this is not a requirement of the Ethernet design, and it is possible for a host to be connected to many points on a single Ethernet channel, each one potentially responding to a different 48-bit number. In this situation the 48-bit number does in fact become an address in the classical sense as it indicates "where" the host is located on the channel. One of these 48-bit numbers could also be the host's internet identity; the mapping from internet address to local network address is now more cumbersome.

6. Market Projections

We have described our reasons for choosing absolute host numbers in internet addresses, and for using them as station addresses on the Ethernet channel. The host number space should be large enough to allow the Xerox internet architecture to have a life span well into the twenty-first century. 48 bits allow for 140,737,488 million physical hosts and mulitcast IDs each. We chose this size based on marketing projections for computers and computer-based products, and to permit easy management of the host number space.

An estimate of the number of computer systems that will be built in the 1980s varies, but it is quite clear that this number will be very large and will continue to increase in the decades that follow. The U.S. Department of Commerce, Bureau of Census estimates that in 1979 there were 165 manufacturers of general-purpose computers, producing about 635,000 units valued at $6,439,000,000 [USCensus79]. There were also about 992,000 terminals and about 1,925,000 standard typewriters

built! International Data Corporation estimates that during 1980-1984 there will be about 3.5 million general purpose mini, small business, and desktop computers built in the United States [IDC80]. Gnostics Concepts Inc. estimates that during 1980-1988 about 63 million central processing units (cpus) of different sizes with minimum memory will be built in the United States alone [Gnostics80].

We expect that the production of microcomputer chips will increase in the decades that follow, and there will be microprocessors in typewriters, cars, telephones, kitchen appliances, games, etc. While all these processors will not be in constant communication with one another it is likely that every now and then they will communicate in a network of processors. For example, when a car containing a microprocessor chip needs repairs, it might be plugged into a diagnostics system thereby putting the car on a communications system. During the time it is hooked into the communication network it would be very convenient if it behaved like all other computers hooked into the system.

We believe that 32 bits, providing over 2,147,483,648,000 physical host numbers and multicast IDs, is probably enough. However, when this large space is carved up among the many computer manufacturers participating in this network architecture, there are bound to be many thousands of unused numbers. It is for this reason that we increased the size to 48 bits. The next section discusses the problems of managing this space.

7. Management and Assignment Procedures

In order that an absolute host numbering scheme work, management policies are needed for the distribution and assignment of both physical and logical host numbers. The major requirement is to generate host numbers in such a way that the probability of the same number being assigned elsewhere is less than the probability that the hardware used to store the number will fail in an undetected manner. There are two ways to manage the host number space:

1) Partition the host number space into blocks and assign blocks to manufacturers or users on demand. The assignment of numbers within a block to machines is the responsibility of each manufacturer or user.

2) Formulate an appropriate algorithm for generating host numbers in a decentralized manner. For example, use a random number generator that reduces the probability of address collisions to a very small acceptable value.

Both options require the existence of an administrative procedure, and perhaps an agency supported by the user community which will have the overall reponsibility of ensuring the uniqueness of host number assignments.

The second option has a great deal of academic appeal, but nevertheless requires an administrative agency that must control the way the random number generator is used to ensure that users do not

initialize it with the same *seed.* One way to accomplish this is to assign unique seeds. This is not very different from assigning unique blocks of numbers! Another way is to provide a thermal noise device on the host to generate a seed or the random host number itself. From a technical standpoint this solution is superior to using software-implemented random number generators, but administrative procedures are still necessary. An agency must certify the "correctness" of the component, i.e., it must guarantee that the component is drawing its numbers from a uniform distribution. In addition to these technical issues, the problem of controlling the assignment of multicast IDs does not lend itself to a random number assignment procedure.

The first option was selected because of its simplicity and ease of administration and control. Xerox Corporataion will manage the assignment of blocks to manufacturers. An in-house database system is being used to assign numbers and produce summaries and reports. This is very similar to the way *uniform product codes* are assigned [UPC78]. The 48-bit host number space is partitioned into 8,388,608 (2^{23}) blocks, each containing 16,777,216 (2^{24}) physical and 16,777,216 (2^{24}) logical host numbers. The partitioning is strictly syntatctic, that is, the "block number" has no semantics, and does not identify a manufacturer.

The owner of a block of host numbers should use all of them before requesting another block. That is, the host numbers within a block should be used "densely", and should not encode the part number, batch number, etc. Mechanisms by which physical host numbers within a block are assigned to machines is manufacturer dependent. Typically, a large-volume manufacturer would make PROMs containing the host number, and then perform quality control tests to ensure that there weren't any duplicates.

Multicast ID assignment is a higher-level, system-wide function, and is a subject we are investigating.

With either assignment option it is possible that two machines inadvertantly received the same host number. Suitable techniques for discovering such anomalies will have to be developed by installations, as part of their network management strategy.

The continued advances in LSI development will make it possible to manufacture an inexpensive "Ethernet chip." Even though host numbers are associated with the host and not a particular network interface, it might be useful to have a unique host number built into each chip and allow the host to read it. The host can then choose whether or not to return this number to the chip as its host number; a host connected to many Ethernet systems can read a unique number from one of the chips and set the physical host number filter to this value in all of them.

The 48-bit host number is represented as a sequence of six 8-bit bytes A, B, C, D, E, F. The bytes are transmitted on the Ethernet channel in the order A, B, C, D, E, F with the least significant bit of each byte transmitted first. The least significant bit of byte A is the *multicast bit*, identifying whether the 48-bit number is a physical or logical host number. Figure 2 illustrates how the bytes

of a 48-bit host number are laid out in an Ethernet packet.

Although the destination address in an internet or intranet packet may specify either a physical host number or a multicast ID, the source address in a packet is generally the physical host number of the host which sent the packet. Knowing the source address is important for error control, diagnostic tests, and maintenance. A host which receives a multicast packet is also free to use that same multicast ID (the destination) in order to transmit an answer "back" to the multicast group.

8. Summary and Conclusions

We believe that all hosts should have a unique physical host number independent of the type or number of networks to which they are physically connected. With the continuing decline in the cost of computing and communications, we expect that internetworks will be very large. Many of the problems in managing the internetwork can be simplified by directly supporting the large absolute host number in the constituent networks, such as the Ethernet. Thus, addresses in the Ethernet system seem to be very generous, well beyond the number of hosts that might be connected to one local network.

The architecture of the Xerox internetwork communication system has been designed to have a life span well into the twenty-first century. We expect that it will receive wide acceptance as a style of internetworking, and therefore chose the host number to be 48 bits long. As a policy decision our internetwork architecture legislates that a host (mulitiply) connected to one or more Ethernet local networks has the same physical host number on each one.

In summary, absolute host numbers have the following properties:

- they permit hosts to be added to, or removed from networks in the internetwork with minimum adminstrative overhead.

- they permit mapping internet addresses to network addresses during encapsulation without translation.

- they permit the separation of routing from addressing, which is especially useful in internetworks with multihomed or mobile hosts.

- they provide the basis for unique identification of files, programs and other objects on stand-alone and networked hosts.

- they support multicast, or the delivery of data to a group of recepients rather than only to a single physical host.

Although a host has the same number for use by operating system software, both within the internetwork and on an Ethernet system, none of the principles of layered protocol design have been violated. Things have simply been conveniently arranged to be optimal in the most common

configurations.

We encourage designers of other local computer networks and distributed systems to use absolute host numbers from our 48-bit address space.

Acknowledgements

Our decision to support an absolute host numbering scheme in internetwork and Ethernet systems was based on many years of experience with the Pup internetwork and the Experimental Ethernet system; David Boggs, John Shoch, Ed Taft, Bob Metcalfe and Hal Murray have helped refine our ideas to their current state. Alan Kotok, Bill Strecker and others at Digital Equipment Corporation provided many recommendations on managing the host number space while we were developing the Ethernet specification.

References

[Abraham80]
   Abraham, S. M., and Dalal, Y. K., "Techniques for Decentralized Management of Distributed Systems," *20th IEEE Computer Society International Conference (Compcon)*, February 1980, pp. 430-436.

[Boggs80]
   Boggs, D. R., Shoch, J. F., Taft, E. A., and Metcalfe, R. M., "PUP: An internetwork architecture," *IEEE Transactions on Communications*, com-28:4, April 1980, pp. 612-624.

[Boggs81]
   Boggs, D. R., "Internet Broadcasting," Ph.D. Thesis, Stanford University, 1981, in preparation, (will be available from Xerox Palo Alto Research Center).

[Cerf78]
   Cerf, V. G., and Kirstein, P. K., "Issues in Packet-Network Interconnection," *Proceedings of the IEEE*, vol 66, no 11, November 1978, pp. 1386-1408.

[Crane80]
   Crane, R. C., and Taft, E. A., "Practical considerations in Ethernet local network design," *Proc. of the 13th Hawaii International Conference on Systems Sciences*, January 1980, pp. 166-174.

[Dalal78]
   Dalal, Y. K., and Metcalfe, R. M., "Reverse Path Forwarding of Broadcast Packets," *Communications of the ACM*, 21:12, December 1978, pp. 1040-1048.

[Ethernet80]
   Intel, Digital Equipment and Xerox Corporations, *The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications*, Version 1.0, September 30, 1980.

[Gnostics80]
   Gnostic Concepts, Inc., *Computer Industry Econometric Service, 1980, Volume 1.*

[IDC80]
   International Data Corporation, Corporate Planning Service, *Processor Data Book 1980.*

[IP80]
Postel, J., ed., *DoD Standard Internet Protocol*, January 1980, NTIS No. ADA079730, also in *ACM Computer Communication Review*, vol 10, no 4, October 80, pp. 2-51.

[Metcalfe76]
Metcalfe, R. M., and Boggs, D. R., "Ethernet: Distributed packet switching for local computer networks," *Communications of the ACM*, 19:7, July 1976, pp. 395-404.

[Oppen81]
Oppen, D. C., and Dalal, Y. K., "The Clearinghouse: A Decentralized Agent for Locating Named Objects in a Distributed Environment," in preparation.

[Redell80]
Redell, D. D., Dalal, Y. K., Horsely, T. R., Lauer, H. C., Lynch, W. C. McJones, P. J., Murray, H. G., and Purcell, S. C., "Pilot: An Operating System for a Personal Computer," *Communications of the ACM*, 23:2, February 1980, pp. 81-92.

[Shoch78]
Shoch, J. F., "Internetwork Naming, Addressing, and Routing," *17th IEEE Computer Society International Conference (Compcon)*, September 1978, pp. 430-437.

[Shoch79]
Shoch, J. F., and Stewart, L., "Interconnecting Local Networks via the Packet Radio Network," *Sixth Data Communications Symposium*, November 1979, pp. 153-158.

[Shoch80]
Shoch, J. F., and Hupp, J. A. , "Measured performance of an Ethernet local network," *Communications of the ACM*, 23:12, December 1980, pp. 711-721.

[Shoch81]
Shoch, J. F., *Local Computer Networks*, McGraw-Hill, in preparation.

[Sunshine81]
Sunshine, C., "Addressing Problems in Multi-Network Systems," in preparation.

[UPC78]
UPC Guidelines Manual, January 1978. Available from Uniform Product Code Council, Inc., 7061 Corporate Way, Suite 106, Dayton Ohio.

[USCensus79]
U.S. Department of Commerce, Bureau of Census, "Computers and Office Accounting Machines," Current Industrial Reports, 1979.
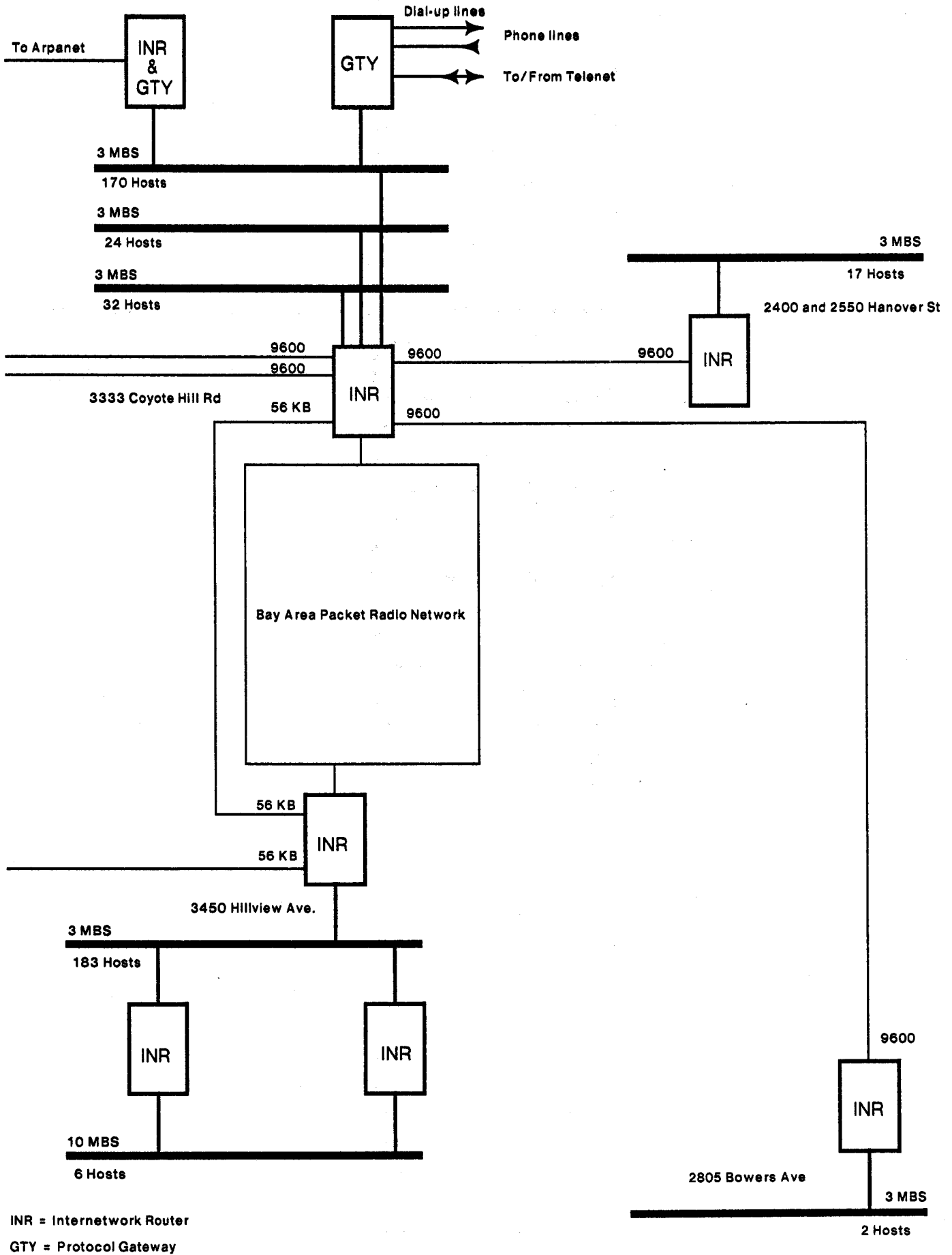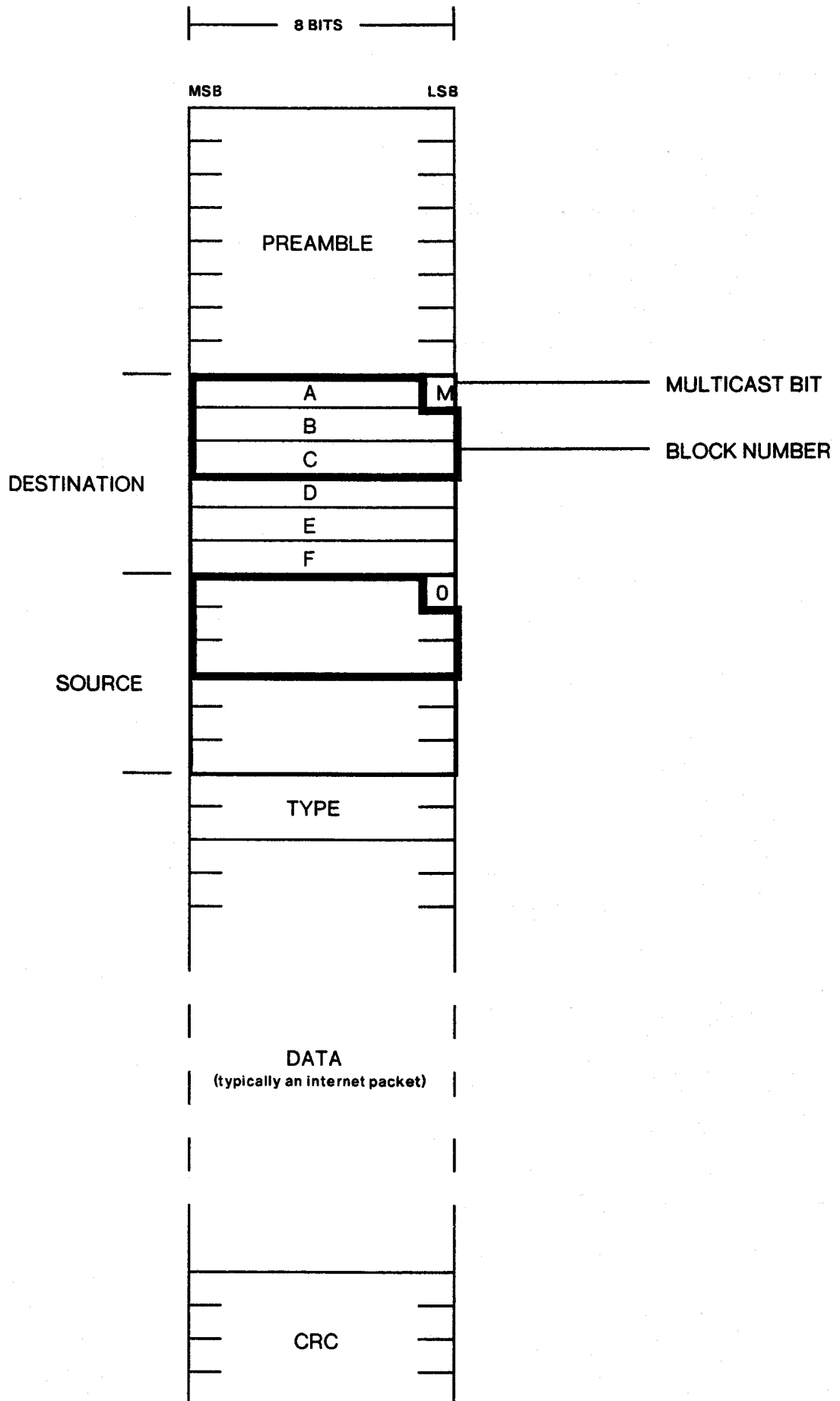
Figure 1. The Xerox Pup-based Experimental Internetwork in the Bay Area

Figure 2. Ethernet Packet and Host Number Format