## INTERNETWORKING

The terms *internetworking* or *network interconnection* refer broadly to the techniques that enable computer systems on one network to communicate with systems on another network. The set of interconnected networks may be called an *internet.* (We shall use the proper name *Internet* for the particularly well known global internet that has come to dominate internetworking in the 1990s.)

A major challenge for internetworking is to allow different types of networks to participate. A variety of network technologies and products have been devised to provide efficient data communication through different media (twisted pair copper wires, optical fibers, coaxial cable) and over various distances, such as within a building, across a campus, and between widely separated locations. Recently, wireless data communications networks (ground and satellite based) have become more prevalent to support mobile users or remote locations. Providing for all types of networks to be interconnected so that users on one network can effectively communicate with users on other networks adds great value to the system.

However, each network technology comes with its own characteristics for speed, format, reliability, and *protocols* which define the format and procedures for data exchange (1). There are good technical and marketing reasons for these different solutions, so diversity in network technologies is likely to persist. This suggests that for a network interconnection strategy to succeed, it must accommodate the autonomy and differences of individual networks to the greatest extent possible. On the other hand, some commonality of services must be supported if communication between users on different networks is to succeed. These two requirements represent a tension, within which a variety of interconnection approaches has been devised (2–5).

Typically, some additional equipment is required to interconnect two different networks, by connecting to both networks through appropriate interfaces and implementing any necessary additional protocols (see Fig. 1). These intermediate devices that create the internet from its component networks may be called *gateways,* or *routers,* since one of their key functions is to forward incoming data in the proper direction to reach its ultimate destination, possibly many networks away. To accomplish this, a higher-level internet addressing scheme must be provided that can identify destinations across all of the networks in the internet. The routers must then determine from this internet address where to send the data next and how to package data in the local protocol used within the next individual network.

The basic operation of an internet is much like that of the postal service. The sender of a letter places it in an envelope with the address of the destination and drops it in the mail. The local postal service then reads the address and delivers the letter to an appropriate forwarding office, using whatever transport mechanism is most suitable (bicycles, trucks, planes). At the forwarding office, the letter is sorted and forwarded again, until it reaches the final post office, which can deliver it to the destination. The postal service is not concerned with the contents of the letter, although it must conform to certain maximum size and weight limits (which may vary in different postal systems). Thanks to certain international agreements, there is enough commonality in mail services and the languages used for addresses that the basic mail forwarding service can be provided successfully, even if the contents might not be understood.

Similarly, in an internet, the data to be sent are bundled into *packets,* with an "envelope" of header and trailer information including the source and destination internet addresses. Each network delivers these to an appropriate router, which uses the header information to determine how to forward the packet onward. In Fig. 1, the sending host *A* sends packets to destination *B* via local area network (LAN) *X* to router *R*, which in turn forwards the packet through WAN *Y* to router *S*, which finally forwards the packet via LAN *Z* to host *B*.

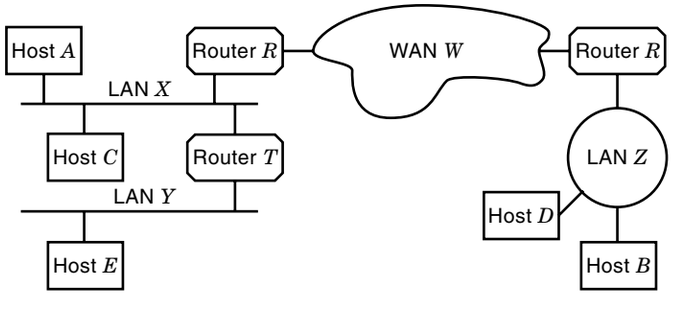


**Figure 1.** Routers interconnect networks to form an internet.

To support all types of data communication applications, the internet must be able to forward arbitrary data inside the packet, so long as the size is acceptable and the "envelope" information is properly formed. If the end host systems and the intermediate routers all implement a common internet protocol to handle the basic addressing and routing functions, the data can reach their destination anywhere in the system. In practice, additional issues, such as congestion control, fragmentation, and multiplexing, must also be dealt with (3).

We first summarize how network interconnection has developed historically. We then review the major technical problems of network interconnection, including stepwise versus endpoint services, level of interconnection, addressing, routing, fragmentation, and congestion control, ending with a summary of functions performed by a router. Next we present several important examples of internet systems that illustrate the technical alternatives, and we conclude with some directions for further research.

## HISTORY OF COMPUTER INTERNETWORKING

Computer networking as we know it today may be said to have gotten its start with the ARPANET development in the late 1960s and early 1970s under the sponsorship of the Advanced Research Projects Agency (ARPA) in the United States. Prior to that time there were computer vendor "networks" designed primarily to connect terminals and remote job entry stations to a mainframe. But the notion of networking between computers viewing each other as equal peers to achieve "resource sharing" was fundamental to the ARPANET design (6). The other strong emphasis of the ARPANET work was its reliance on the then novel technique of packet switching to share communication resources efficiently among users transmitting intermittent bursts of information, instead of the more traditional dedicated links or circuit switching which supported steady rate transmission well.

Although the term *network architecture* was not yet widely used, the initial ARPANET design did have a definite structure and introduced another key concept: protocol layering, or the idea that the total communications functions could be divided into several layers, each building on the services of the one below. The original design had three major layers, a network layer that included the network access and switch-to-switch (IMP-to-IMP) protocols, a host-to-host layer (the Network Control Protocol, or NCP), and a function-oriented protocol layer, where specific applications such as file transfer, mail, speech, and remote terminal support were provided (7).

Similar ideas were being pursued in several other research projects around the world, including the Cyclades network in France (5), The National Physical Laboratory Network in England (8), and the Ethernet system (9) at Xerox Palo Alto Research Center in the United States. Some of these projects focused more heavily on the potential for high-speed local networks such as the early 3 Mbps Ethernet. Satellite and radio channels for mobile users were also a topic of growing research interest.

### Creation of the Internet Protocol

By 1973 it was clear to the networking vanguard that another protocol layer needed to be inserted into the protocol hierarchy to accommodate the interconnection of diverse types of individual networks. Cerf and Kahn published their seminal paper describing such a scheme (10), and development of the new Internet Protocol (IP) and Transmission Control Protocol (TCP) to jointly replace the NCP began. Similar work was being pursued by other groups meeting in the newly formed International Federation of Information Processing (IFIP) Working Group 6.1, called the Internetwork Working Group (11).

The basis for the network interconnection approach developing in this community was to make use of a variety of individual networks, each providing only a simple "best effort" or *datagram* transmission service. Reliable virtual circuit services would then be provided on an end-to-end basis with the TCP (or similar protocol) in the hosts. ARPA sponsored an effort to form a national internet based on TCP/IP protocols and connecting research groups with local networks via the ARPANET. This sytem gradually grew international extensions and eventually became the Internet.

### Other Internetworking Approaches

During the same time period, public data networks (PDN) were emerging under the auspices of what is now the International Telecommunications Union (ITU), then known as CCITT. The newly defined X.25 protocol aimed at providing more traditional virtual circuit types of network service that guaranteed reliable end-to-end delivery (1). The PDNs devised an interconnection scheme based on concatenating virtual circuits across each network (12). The middle and late 1970s saw networking conferences dominated by heated debates over the relative merits of circuit versus packet switching and datagrams versus X.25 virtual circuits (13).

The mainframe computer vendors continued to offer their proprietary networks, gradually supporting the new X.25 service as links under their own protocols. Digital Equipment (DEC) was the notable exception, adopting the research community approach of peer-to-peer networking at an early date and coming out with its own new suite of protocols (DECNET).

By the late 1970s, a new major influence was emerging in the computer networking community. The computer manufacturers realized that multivendor systems could no longer be avoided and began to take action to satisfy the growing user demand for interoperability. Working through their traditional standards body, the International Standards Organization (ISO), a new group (Study Committee 16) was created to develop standards in the networking area. Their initial charter was to define an explicit architecture or "reference model" for Open Systems Interconnection (OSI) (1). They formalized the concept of protocol layering to facilitate the design of increasingly complex communications software. In a layered architecture, the communications functions in each system are partitioned into a set of layers, with each layer making use of the functions provided by the layer beneath. This allows modifying the protocol within a layer so long as the functions provided upward and used below are maintained.

### Interconnection of LANs

Another force contributing to the growth of internetworking was the introduction of personal computer networks, initially for business purposes. Both Apple Computer and Novell introduced networking software in the mid-1980s that allowed multiple LANs to be interconnected, with sharing of files and printers. The work on Ethernet at Xerox was extended to allow interconnection of LANs over long-distance links (14).

The breakup of the long-distance phone monopoly in the United States in 1984 provided competition and a rapid drop in prices for higher-speed links to interconnect the growing business LANs at various sites. Such links also provided greater bandwidth for interconnection of the growing number of TCP/IP networks at university and research sites. This led to formation of the first high-speed national TCP/IP network by the National Science Foundation and a further growth of TCP/IP systems to include commercial sites. The Internet Engineering Task Force (IETF) was formed to guide the further evolution of the TCP/IP internet, later known as the Internet.

Meanwhile, the CCITT and ISO camps aligned their efforts, with OSI adding an internet sublayer within the network layer to accomodate the datagram internetworking approach beside the virtual circuit approach. This new OSI protocol family functioned much like the TCP/IP suite. Many proponents of the OSI stack expected it to succeed the TCP/IP suite, and it enjoyed considerable acceptance in Europe and the Far East. The United States government mandated its inclusion in all network purchases through the Government Open Systems Interconnect Profile (GOSIP).

### Dominance of the Internet

In the mid-1990s, several factors contributed to the growing dominance of the TCP/IP system, which came to be called simply the Internet. Free software for the TCP/IP suite was widely available. The invention of hypertext browser software (the original was called Mosaic) made hypermedia information throughout the Internet easily accessable. With the tremendous growth in PCs and the discovery of the Internet for personal and general business use, demand for connectivity accelerated dramatically, and by the late 1990s there are millions of connections to the Internet. The protocol suite developed by the researchers in the 1970s is now an essential basis for a vast array of personal and enterprise information exchange. In the process of growth, some modifications to the original internet protocols have been proposed, but the fundamental principles remain valid.

## MAJOR TECHNICAL ISSUES

As noted previously, an internet must deal with basic issues common to any switching system, such as addressing, routing, congestion control, fragmentation, and multiplexing. The following sections focus on the extra concerns that are important at the internet level in each of these areas, along with a discussion of alternatives for the level at which to interconnect networks.

### Naming, Addressing, and Routing

To understand the problem of delivering data to the correct destination in an internet, a clear distinction must be drawn among names, addresses, and routes (15). Although these concepts are applicable at each protocol level, we shall be primarily concerned with the network level, where *hosts* or *end systems* and routers are the relevant objects. A *name* serves

to identify the host "logically," independent of its point(s) of attachment to the network(s). The same host may have several names to provide for convenient "nicknames" or aliases. An *address* identifies a point of attachement for purposes of delivering data to the host; since the same host may have multiple network interfaces, it may have multiple addresses. Finally, a *route* is the path taken from source to destination host (the sequence of intermediate nodes that the packet traverses), and there are typically multiple routes available to the same destination.

The process of sending data to a destination generally involves first determining its address from its name using a directory service, and then determining the best route to that address. In large systems, this name lookup function is typically implemented in a distributed fashion, with a hierarchical name space where subdirectories are responsible for their portion of the name space (16).

**Addressing.** As noted earlier, packets traversing an internet include a header specifying the internet address of the destination host. Internet addresses must provide a unique identifier for each network interface in the internet system. In small internet systems with broadcast media (such as Ethernet, or token rings) it may be sufficient to use a "flat" address format where the addresses provide no indication of the location of the host's interface.

In large internet systems, it is essential to introduce a hierarchical internet address format where an explicit "network" prefix is combined with a "local" suffix to form a complete address (Fig. 2). The network prefix identifies the destination network (or closely related set of networks), and the local suffix identifies the destination host interface within that network.

In the Internet, the original internet address format was 32 bits, with either 8, 16, or 24 bits allocated for the network prefix. This allowed for a small, medium, or large number of nets which each contained a large, medium, or small number of hosts, respectively. As the Internet expanded, a more flexible scheme was developed called *subnetting* (17) which allows a locally interconnected group of networks (such as LANs on a campus) to appear as a single network to the rest of the Internet. Local routers in the group then use the first few bits of the "local" address to properly distinguish between the different "internal" networks. While subnetting has allowed successful expansion of the Internet for many years, the newly adopted Internet Protocol Version 6 provides for longer 128-bit addresses to facilitate future growth.

Another addressing issue in forming successful internets is how to create the network level addresses needed to transmit a packet through each individual network along its path. As described later, routing tables in each host or router provide the internet address to which each incoming packet must be forwarded, but this must be translated to a network level address that can be "understood" by the next network. In some cases the local portion of the internet address can be

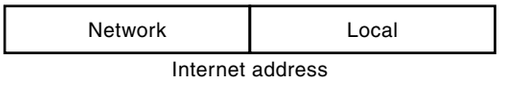


**Figure 2.** Hierarchical addresses simplify global addressing and routing.

| Dest. | Next Address | Port |
|---|---|---|
| *A* | *A* | *X* |
| *B* | *S* | *W* |
| *C* | *C* | *X* |
| *D* | *S* | *W* |
| *E* | *T* | *X* |
| ⋮ | | |

(**a**) Flat internet addressing

| Dest. Net | Next Address | Port |
|---|---|---|
| *X* | Local | *X* |
| *Y* | *T* | *X* |
| *Z* | *S* | *W* |
| ⋮ | | |

(**b**) Hierarchical internet addressing

**Figure 3.** Routing table for Router *R* in Fig. 1. Table is larger with flat internet addressing (a) and smaller with hierarchical addressing (b).

used or translated directly into a network address (e.g., the IMP and port numbers in the original ARPANET). In other cases, the local portion must be determined or "resolved" using tables created by an address resolution protocol (ARP) (17). The ARP dynamically discovers the network addresses (and internet addresses) of hosts and routers connected to a particular network, and maintains a table giving the correspondence between network and internet addresses on a given network.

**Routing.** As a packet traverses the internet, the source node and each router must take the destination internet address and determine the next place to send the packet. Normally this is done using a *routing table,* or data structure containing destination addresses and how to reach them. With a flat internet address space routers must maintain information on how to reach each destination individually, and hence have large routing tables. This approach is usable in smaller internet systems, such as interconnected LANs.

Figure 3(a) shows a routing table for Router *R* in Fig. 1, containing a list of destination addresses and the address of the next hop. The next hop specifies the internet address of the next router to use when forwarding the packet, which in turn determines which "port" or network interface to use. The routing subroutines usually index entries with a hash, tree, or other efficient lookup mechanism to speed the process and may contain additional information for each entry, like age and frequency of use.

Routing tables may be constructed according to many requirements, such as lowest delay or cost or highest availability. In some cases they are best created statically when the end system or router is configured. For example, host *A* in Fig. 1 need only create a single default route to Router *B*, since that is the only path into the internet.

Though static tables for singly connected end systems are commonplace, other routing tables are commonly altered dynamically to represent current link and router availability. Accomplishing this task in large networks or internets with high reliability, efficiency, and timeliness has been a very challenging problem that has led to the development of many routing information exchange protocols that balance complexity, optimality, and required processing speed (1,18).

For large internets, a hierarchical address is often used in the internet protocol, so that routing can be done in steps. First the gateways route packets to the final network (ignoring the local suffix), and then within the final network to the local address. With this approach the routing table contains an entry for each net, rather than for every destination host.

This reduces the size of routing tables, as shown in Fig. 3(b), with the potential for some loss in optimality.

Despite considerable recent progress in routing algorithms, computer and human errror conditions can still occur that create *routing loops,* a condition among a set of routing tables in which packets repeatedly traverse the same set of intermediate systems, never reaching their ultimate destination. To prevent these conditions from congesting the network indefinitely, internetwork protocols specify a *hop count* or *time to live* field that is decremented by each router. If the hop count ever reaches zero, the packet is discarded. Senders normally set this field equal to or greater than the longest normal path through the internet.

Another design choice concerns the frequency of routing decisions. For maximum robustness, each packet may cause a best route selection process to be carried out, as in the original ARPA internet (19). Other systems choose to perform the best route determination process only for the initial packet to a destination. This route is then remembered in the routers, and subsequent packets to the same destination follow the same route. Often this type of path set up is accompanied by an abbreviated addressing convention, where only the first packet must carry full destination address and subsequent packets carry only a shorter path identifier. The CCITT X.75 and the new IPv6 use this approach. A mechanism for timing out such routes and recovering from changes in the internet topology must be provided.

Yet another approach employs flooding to avoid the need for intelligence in packet forwarders. Since flooding is expensive of network resources, it is typically used only for control purposes or for initially establishing a path that later packets to the same destination will follow. Another method of routing called *source routing* allows the sender to avoid the need for intelligent routers or to force a specific path to be used by providing a route in the packets it sends (20).

### Congestion Control

The problems of congestion control in an internet system are much like those of individual networks. Speed mismatches are likely to be more severe between LANs and slower wide area networks (although recent advances in high-speed WAN service should reduce this). In some cases, the individual network procedures may be adequate [e.g., Asynchronous Transfer Mode (ATM) quality-of-service parameters]. In others, some form of explicit internet-level control may be needed.

Questions have been raised about the ability of connectionless systems to provide effective congestion control. This is a particular concern when connectionless or datagram internet service is used to support higher-level connection-oriented services. Several techniques have been proposed in this area, including input buffer limits, buffer classes, fair queuing, slow start, and choke packets (1,21). Once the sender has determined that congestion has occurred (by receiving an explicit signal from a host or router or by timing out waiting for an acknowledgment), it must reduce its transmission rate for a while and then try to increase it again. Various specific algorithms for this purpose have been proposed, and this is an active area of research.

### Fragmentation and Reassembly

When networks with differing maximum packet size limits are interconnected, the need to fragment large packets for traversal through networks with smaller size limits must be considered. The original packet is broken into two or more new packets, each small enough to transmit over the next network. These fragments can be reassembled at the exit from the individual small packet network or allowed to propagate all the way to the final destination.

Mechanisms to support such fragmentation typically include some sort of additional sequencing information in the packet header. The most robust mechanisms allow further fragmentation of already created fragments and proper reassembly of fragments at the final destination that may have followed different paths.

In general, fragmentation is undesirable because of the processing burden placed on routers and because of the possibility of inefficient link utilization. For example, a fragment that fills one network packet may have to be fragmented at a subsequent router into one large and one very small piece. The very small piece has a large "overhead" (ratio of data carried to data and header information), which uses resources inefficiently. To help alleviate this problem, the internet protocol suite may provide for an advisory message to be transmitted back to the source of large packets, indicating that they are too big for the router to forward without fragmentation.

### Level of Interconnection

The previous discussion has assumed that networks are interconnected at the network level of the protocol hierarchy, since this is the dominant approach in use today. However, other levels of interconnection may also be chosen, from the lowest (physical) level to the highest (application) level. In general, the lower the level of interconnection, the more similar the networks to be connected must be, while high-level interconnections support more specialized services.

When different networks and protocols are involved, the interconnection involves a conversion process between the services provided for comparable functions in each network (22). The complexity of this process and the quality of end-to-end services resulting are largely determined by the level of interconnection chosen. The following sections summarize the key features of each major alternative.

**Physical Level.** The physical level deals with serial transmission of bits over a physical medium. Interconnection devices operating at the physical level are generally called *repeaters.* They forward individual bits of the packet as they arrive, perhaps translating from one medium to another (e.g., baseband coaxial cable to optical fiber). The resulting interconnected system functions essentially as a single network at the data link level, and hence all networks to be so connected must have identical data rates and link protocols. This approach is typically used to interconnect several physically separate segments of a LAN system, perhaps separated by a point-to-point link. A disadvantage is that repeaters propagate noise and interference as well as valid data.

**Link Level.** The link level deals with transmission of frames over a link, which may be shared by multiple users. Interconnection devices operating at the link level receive entire frames from one link, examine the link level protocol header, and possibly forward the frame onto another link. They are

typically called *bridges* (20). As with repeaters, they may interconnect two or more local LAN segments or may interconnect remote segments over a long-distance link. Major motivations for their use are to interconnect LAN segments with different speeds and/or protocols or to increase network capacity by "filtering" incoming packets and forwarding only those whose link-level destination is on another segment. Hence bridges accommodate parallelism by permitting simultaneous use of both segments. Moreover, bridges transparently support systems with multiple network-level protocols in use.

**Network Level.** The network level deals with transmission of packets over a network that may include intermediate switches. Traditionally, interconnection at the network protocol level has been a WAN problem, where different networks had independently developed different protocol mechanisms for the variety of network-level functions, such as routing, congestion control, error handling, and segmenting. If the networks are identical, then the problem becomes largely one of routing as with the X.25/X.75 approach in public data networks. When the networks differ, the complexity of protocols at the network level (e.g., X.25 versus ARPANET 1822) makes a translation approach difficult. There has been some success in one vendor emulating another vendor's network behavior [e.g., IBM Systems Network Architecture (SNA) gateways].

The approach that has gained wide acceptance in the Internet places a common IP sublayer on top of the different network protocols. As noted previously, this has particular benefits for supporting the sophisticated routing procedures needed for large internet systems, and devices operating at this level are often called IP *routers*. Choosing this level for interconnection makes available the general-purpose services of the network level and allows the router implementor to take advantage of what is normally a well-documented interface with many implementations. It allows each network to function autonomously with its own procedures internally, while requiring some standard "internet" procedures to be used on top of the normal network access for individual networks.

**Transport Level.** The transport layer is intended to provide general-purpose data transfer between end users. In the OSI architecture, the transport service is supposed to be an end-to-end service, so transport-level gateways are, strictly speaking, a violation of the architecture. Nevertheless, they may be of practical benefit when common upper-level protocols are in use but different transport protocols are available. Early experiments with the competing protocol hierarchies demonstrated connections of this nature (for example, concatenating TCP and ISO TP4 connections to each other).

**Higher Level.** Many application-level gateways have been implemented to support specific services found at the application level. This type of gateway is essentially a "Janus host" that implements two (or more) full protocol suites. Common examples have been interconnecting terminal concentrators or CCITT packet assemblers/disassemblers (PAD) to provide an interactive terminal service, or electronic mail servers to form a mail forwarding service. Where only a specific application service is wanted and the desired application services on each net match closely, this type of gateway may be easy to set up with existing equipment. However, the service provided is clearly not general purpose, and the limitations imposed by providing only those service elements common to the interconnected systems are often more irksome than anticipated (23).

## MAJOR INTERNET EXAMPLES

The following sections illustrate the application of the technical issues discussed previously in several widely used internet systems.

### The Internet

One of the first major internet systems was developed by ARPA in the United States (24,25). This system included the original ARPANET, packet radio nets, satellite networks, and various LANs. The system was subsequently split into separate systems for research users and for operational military users and eventually evolved into the Internet.

Networks in the Internet are interconnected by routers that implement a connectionless or datagram IP (19,26,27) to provide maximum robustness and routing flexibility. The system originally employed dedicated router machines based on general-purpose 16-bit minicomputers, but special-purpose high-speed routers are now manufactured by a variety of vendors. Each datagram is analyzed by the routers and routed based on its destination address. The Internet uses hierarchical 32-bit addresses, with routers designed to route to the network portion of the address first, and then the local portion once the correct net is reached. As described earlier, subnetting has been introduced to allow more efficient and flexible use of address space. Host name to address lookup was initially supported by a single flat directory, but as the number of hosts grew, a hierarchical distributed directory service [the domain name system (DNS)] was adopted (17), which now can access millions of names throughout the world within a few seconds.

Most of the individual networks in the Internet provide connectionless service, although there is a provision for running IP over connection-oriented network services such as X.25 and ATM. The major transport service is connection oriented, implemented by a common protocol called the transmission control protocol (TCP), that must be present in the end systems (not in routers). IP also supports other types of transport protocols, including datagram and "stream" mode (for packetized voice or video).

The Internet IP provides for fragmentation at routers, with reassembly at the final destination so that individual fragments may follow different routes. A time-to-live or hop limit field is included to limit the maximum lifetime of packets in the system, providing an essential part of the overall routing system. Options are defined to allow inclusion of source routes, security markings, timestamps, and so on.

There is a separate Internet Control Message Protocol (ICMP) used for signaling errors and diagnostic information. This includes destination unreachable, congestion control (choke packets), packet too big, echo request/reply, and redirect indications (giving a better route for a specific destination).

Internet routing information exchange was originally handled by a gateway-to-gateway protocol that required interaction between all "neighboring" gateways. As the Internet grew, a more hierarchical scheme called the Exterior Gateway Protocol (EGP) (24) was developed to reduce the amount of routing traffic. In EGP, each autonomous system (typically a campus or corporate internet) elects one gateway to exchange routing data with a neighbor gateway in an adjacent autonomous system, and the systems then propagate the information to all their other gateways through an internal procedure. EGP evolved further to become the Interdomain Routing Protocol (IDRP) (28).

The version of IP developed in the 1970s (IPv4) has been adapted to work on a wide variety of subnetwork technologies and is used at very high speeds, but its deployment in large-scale networks has revealed opportunities for enhancement. In particular, a larger address space is needed. A new version, IPv6, which supports 128-bit addresses, eliminates fragmentation, and improves route lookup times, has been defined by the IETF and is being cautiously deployed.

### International Standards Organization

The ISO extended its original seven-layer OSI architecture to define three sublayers within the network layer. The topmost layer corresponds to the internet protocol, and the middle layer is intended to adapt ("converge") specific network services to those required by the internet sublayer. One example would be use of a connectionless internet protocol over a connection oriented network, requiring a *connection management* intermediate layer protocol to set up and terminate connections as needed in order to send internet-level datagrams.

ISO has defined a connectionless internet sublayer protocol (1,29,30) much like the Internet IP. Although the format of the packet header is different, most fields have a one-to-one correspondence with the Internet IP. However, the ISO IP does not include a field to specify the upper layer protocol being carried since this is viewed as part of the address information. The ISO IP includes an error reporting capability, while the Internet IP provides this through the separate ICMP protocol. The fragmentation (segmentation) fields are different, with the ISO IP including a field giving the total length of the original segment in each fragment to aid in assigning reassembly buffers.

The final major difference concerns the format of addresses at the network level, which is not part of the ISO IP itself but is covered in a separate document. The ISO format is a variable-length string that is intended to cover the requirements of both public and private, local and wide area networks for the foreseeable future. This involves a maximum of 16 octets of binary data, which could be alternatively coded as 40 binary-coded decimal digits. The first octet is an authority/format code meant to indicate what format the following data are in. Provision has been made to identify all the major address formats as alternatives (X.121, F.69 [telex], E.163 [telephone], E.164 [ISDN], ISO 6523). The address is assumed to be hierarchical, with each domain responsible for defining the meaning of the suffix portion of the address under its control.

### Appletalk

Appletalk was developed in the mid-1980s and is primarily used on Apple computers. It has several innovations that make it efficient and easy to configure. There are two header format options: a 5-byte short form for use on packets that do not exit a single LAN, and a 13-byte-long form for routed packets. The former is quite compact, containing 6 reserved bits, 10 length bits, the source and destination sockets, and the Datagram Delivery Protocol (DDP) type, indicating the application. The sockets identify the particular application to receive the data, as is usually done in the transport protocol; the packets do not have internetwork addresses because the link layer sends it to the correct destination. Zero to 586 data bytes follow the header.

The first two bytes of the extended header are the same as for the short, except for a 4- bit hop count field that limits maximum network diameter to 15. Bytes 3 and 4 are an optional header checksum. Following are two destination and two source network bytes for a total of over 65,000 allowed networks (addresses FF00 through FFFE are reserved.) Each network may have 254 nodes, as indicated in the following two bytes.

Addressing is handled in a "plug and play" fashion. End systems arbitrate (using a broadcast protocol) to obtain an unused node ID when they are initialized and learn the network number (if any) from their nearest router. This eliminates the need to configure end nodes with unique addresses. Routers are manually configured with network numbers.

### Novell IPX

Novell began selling its distributed system in the mid-1980s, and made rapid inroads in the office automation market. Novell's Internet Packet Exchange (IPX) protocol has a fixed 30-byte header. The first 2 bytes contain an optional checksum, followed by 2 bytes of length (excluding LAN overhead.) Next is a 1-byte time-to-live field that starts at 16, and a packet type that indicates which transport protocol is used. The destination and source node addresses have a 4-byte network part and a 6-byte node identifier that is the same as the physical address for Ethernet. Since the physical address is expected to be unique, this simplifies manual configuration and eliminates the need to discover physical addresses dynamically. The source and destination sockets identify particular applications, like transport sockets.

## FUTURE DIRECTIONS

The variety of individual network technologies is likely to continue increasing. Fortunately, by introducing standards at the internetwork level, it is possible to interconnect diverse networks while preserving their individual autonomy to a large degree. The success of the Internet in working with new network technologies such as FDDI and ATM indicates the validity of its basic architecture.

To cope with the tremendous growth in end systems, broader addressing and routing schemes are now emerging from IETF work (28). Research is also underway on improved methods of congestion control and routing protocols.

With the high packet rates now flowing in the Internet, some provisions for streamlining packet processing are needed. A new version of the Internet IP, IPv6, provides for 128-bit addresses, with a flow label in each packet to allow routers to cache routes and avoid a full destination lookup on each packet. These are timed out every few seconds to ensure

responsiveness to changing conditions. IPv6 allows fragmentation only at the source host, to streamline packet processing in intermediate routers.

Inclusion of high-latency links, such as satellite hops, and high-error-rate links (mobile users) also provides new challenges for the Internet. Greater demand for broadcast service (the same data going to multiple users), constant rate data (audio and video), and asymmetric rate links (fast data retrieval, slow requests, as provided in some broadband cable systems, and Asynchronous Digital Subscriber Line technology) are other directions for expansion.

## BIBLIOGRAPHY


1. C. Sunshine (ed.), *Computer network architectures and their protocols,* 2nd ed., New York: Plenum, 1989.
2. V. Cerf and P. Kirstein, Issues in packet network interconnection, *Proc. IEEE,* **66**: 1386–1408, 1978.
3. M. Gien and H. Zimmermann, Design principles for network interconnection, *Proc. 6th Data Commun. Symp.,* Pacific Grove, CA, 1979, ACM/IEEE, pp. 109–119.
4. J. Postel, Internetwork protocol approaches, *IEEE Trans. Comm.,* **COM-28**: 604–611, 1980.
5. L. Pouzin, A proposal for interconnecting packet switching networks, *Proc. Eurocomp,* 1974.
6. L. Roberts and B. Wessler, Computer network development to achieve resource sharing, *AFIPS Conf. Proc.,* (SJCC) **36**: 543–549, 1970.
7. V. Cerf, The DoD internet architecture model, *Comput. Netw.,* **7**: 307–318, 1983.
8. R. Scantlebury and P. Wilkinson, The national physical laboratory data communication network, *Proc. ICCC,* Stockholm, 1974.
9. R. Metcalfe and D. Boggs, ETHERNET: Distributed packet switching for local computer networks, *Commun. ACM,* **19**: 395–404, 1976.
10. V. Cerf and R. Kahn, A protocol for packet network intercommunication, *IEEE Trans. Commun.,* **COM-22**: 637–648, 1974.
11. V. Cerf et al., Proposal for an international end-to-end protocol, *Comput. Comm. Rev.,* **6**: 68–89, 1974.
12. A. Rybczynski, J. Palframan, and A. Thomas, Design of the Datapac X.75 internetworking capability, *Proc. 5th Int. Conf. Comput. Comm.,* 1980, pp. 735–740.
13. B. Meister, P. Janson, and L. Svobodova, Connection-oriented versus connectionless protocols: a performance study, *IEEE Trans. Comput.,* **C-34**: 1164–1173, 1985.
14. D. Boggs et al., PUP, An internetwork architecture, *IEEE Trans. Commun.,* **COM-28**: 612–624, 1980.
15. J. Shoch, Internetwork naming, addressing, and routing, *Proc. IEEE COMPCON,* 1978, pp. 72–79.
16. P. Mockapetris and K. Dunlap, Development of the domain name system, *Proc. ACM SIGCOMM Symp.,* 1988, pp. 123–133.
17. D. Comer, *Computer Networks and Internets,* Englewood Cliffs, NJ: Prentice-Hall, 1997.
18. C. Huitema, *Routing in the Internet,* Upper Saddle River, NJ: Prentice-Hall, 1995.
19. J. Postel, C. Sunshine, and D. Cohen, The ARPA internet protocol, *Comput. Netw.,* **5** (4): 261–271, 1981.
20. R. Dixon and D. Pitt, Addressing, bridging, and source routing, *IEEE Netw.,* **2** (1): 25–32, 1988.
21. V. Jacobson, Congestion avoidance and control, *Proc. ACM SIGCOMM Symp.,* 1988, pp. 314–329.
22. P. Green, Jr., Protocol conversion, *IEEE Trans. Comm.,* **COM-34**: 257–268, 1986.
23. M. Padlipsky, Gateways, architectures, and heffalumps, in *The Elements of Networking Style,* Englewood Cliffs, NJ: Prentice-Hall, 1985, pp. 167–176.
24. R. Hinden, J. Haverty, and A. Sheltzer, The DARPA internet: Interconnection of heterogeneous computer networks with gateways, *IEEE Comput.,* **16** (9): 38–48, 1983.
25. J. Postel, C. Sunshine, and D. Cohen, Recent developments in the DARPA internet program, *Proc. 6th Int. Conf. Comput. Comm.,* London, UK, 1982, pp. 975–979.
26. Department of Defense, *Internet protocol,* MIL-STD-1777, 1983.
27. D. Clark, The design philosophy of the DARPA internet protocols, *Proc. ACM SIGCOMM Symp.,* 1988, pp. 106–114.
28. S. Thomas, *IPng and the TCP/IP Protocols: Implementing the Next Generation Internet,* New York: Wiley, 1996.
29. R. Callon, Internetwork protocol, *Proc. IEEE,* **71**: 1388–1393, 1983.
30. International Standards Organization (ISO), *Protocol for providing the connectionless network service,* IS 8473, March 1986.


CARL A. SUNSHINE
Aerospace Corporation

**INTERPRETERS, PROGRAM.** See PROGRAM INTERPRETERS.

**INTERPROCESS COMMUNICATION.** See APPLICATION PROGRAM INTERFACES.