3

Internetworking Concept
And Architectural Model

3.1 Introduction

So far we have looked at the low-level details of transmission across individual data networks, the foundation on which all computer communication is built. This chapter makes a giant conceptual leap by describing a scheme that allows us to collect the diverse network technologies into a coordinated whole. The primary goal is a system that hides the details of underlying network hardware while providing universal communication services. The primary result is a high-level abstraction that provides the framework for all design decisions. Succeeding chapters show how we use this abstraction to build the necessary layers of internet communication software and how the software hides the underlying physical transport mechanisms. Later chapters also show how applications use the resulting communication system.

3.2 Application-Level Interconnection

Designers have taken two different approaches to hiding network details, using application programs to handle heterogeneity or hiding details in the operating system. Early heterogeneous network interconnections provided uniformity through application-level programs called application gateways. In such systems, an application-level program, executing on each computer in the network, understands the details of the network connections for that computer, and interoperates across those connections with application programs on other computers. For example, some electronic mail systems
consist of mail programs that are each configured to forward a memo to a mail program on the next computer. The path from source to destination may involve many different networks, but that does not matter as long as the mail systems on all the machines cooperate by forwarding each message.

Using application programs to hide network details may seem natural at first, but such an approach results in limited, cumbersome communication. Adding new functionality to the system means building a new application program for each computer. Adding new network hardware means modifying existing programs (or creating new programs) for each possible application. On a given computer, each application program must understand the network connections for the computer, resulting in duplication of code.

Users who are experienced with networking understand that once the interconnections grow to hundreds or thousands of networks, no one can possibly build all the necessary application programs. Furthermore, success of the step-at-a-time communication scheme requires correctness of all application programs executing along the path. When an intermediate program fails, the source and destination remain unable to detect or control the problem. Thus, systems that use intermediate applications programs cannot guarantee reliable communication.

### 3.3 Network-Level Interconnection

The alternative to providing interconnection with application-level programs is a system based on network-level interconnection. A network-level interconnection provides a mechanism that delivers small packets of data from their original source to their ultimate destination without using intermediate application programs. Switching small units of data instead of files or large messages has several advantages. First, the scheme maps directly onto the underlying network hardware, making it extremely efficient. Second, network-level interconnection separates data communication activities from application programs, permitting intermediate computers to handle network traffic without understanding the applications that are sending or receiving it. Third, using network connections keeps the entire system flexible, making it possible to build general purpose communication facilities. Fourth, the scheme allows network managers to add new network technologies by modifying or adding a single piece of new network level software, while application programs remain unchanged.

The key to designing universal network-level interconnection can be found in an abstract communication system concept known as *internetworking*. The internetwork, or *internet*, concept is an extremely powerful one. It detaches the notions of communication from the details of network technologies and hides low-level details from the user. More important, it drives all software design decisions and explains how to handle physical addresses and routes. After reviewing basic motivations for internetworking, we will consider the properties of an internet in more detail.

We begin with two fundamental observations about the design of communication systems:
• No single network hardware technology can satisfy all constraints.
• Users desire universal interconnection.

The first observation is an economic as well as technical one. Inexpensive Local Area Networks that provide high speed communication only cover short distances; wide area networks that span long distances cannot supply local communication cheaply. Because no single network technology satisfies all needs, we are forced to consider multiple underlying hardware technologies.

The second observation is self-evident. Ultimately, users would like to be able to communicate between any two points. In particular, we desire a communication system that is not constrained by the boundaries of physical networks.

The goal is to build a unified, cooperative interconnection of networks that supports a universal communication service. Within each network, computers will use underlying technology-dependent communication facilities like those described in Chapter 2. New software, inserted between the technology-dependent communication mechanisms and application programs, will hide the low-level details and make the collection of networks appear to be a single large network. Such an interconnection scheme is called an internetwork or internet.

The idea of building an internet follows a standard pattern of system design: researchers imagine a high-level computing facility and work from available computing technology, adding layers of software until they have a system that efficiently implements the imagined high-level facility. The next section shows the first step of the design process by defining the goal more precisely.

3.4 Properties Of The Internet

The notion of universal service is important, but it alone does not capture all the ideas we have in mind for a unified internet because there can be many implementations of universal services. In our design, we want to hide the underlying internet architecture from the user. That is, we do not want to require users or application programs to understand the details of hardware interconnections to use the internet. We also do not want to mandate a network interconnection topology. In particular, adding a new network to the internet should not mean connecting to a centralized switching point, nor should it mean adding direct physical connections between the new network and all existing networks. We want to be able to send data across intermediate networks even though they are not directly connected to the source or destination computers. We want all computers in the internet to share a universal set of machine identifiers (which can be thought of as names or addresses).

Our notion of a unified internet also includes the idea of network independence in the user interface. That is, we want the set of operations used to establish communication or to transfer data to remain independent of the underlying network technologies and the destination computer. Certainly, a user should not have to understand the network interconnection topology when creating or using application programs that communicate.
3.5 Internet Architecture

We have seen how computers connect to individual networks. The question arises, "How are networks interconnected to form an internetwork?" The answer has two parts. Physically, two networks can only be connected by a computer that attaches to both of them. A physical attachment does not provide the interconnection we have in mind, however, because such a connection does not guarantee that the computer will cooperate with other machines that wish to communicate. To have a viable internet, we need special computers that are willing to transfer packets from one network to another. Computers that interconnect two networks and pass packets from one to the other are called internet gateways or internet routers.

Consider an example consisting of two physical networks shown in Figure 3.1. In the figure, router R connects to both network 1 and network 2. For R to act as a router, it must capture packets on network 1 that are bound for machines on network 2 and transfer them. Similarly, R must capture packets on network 2 that are destined for machines on network 1 and transfer them.

![Figure 3.1 Two physical networks interconnected by R, a router (IP gateway).](image)

In the figure, clouds are used to denote physical networks because the exact hardware is unimportant. Each network can be a LAN or a WAN, and each may have many computers attached or a few computers attached.

3.6 Interconnection Through IP Routers

Although it illustrates the basic connection strategy, Figure 3.1 is quite simplistic. In an actual internet that includes many networks and routers, each router needs to know about the topology of the internet beyond the networks to which it connects. For example, Figure 3.2 shows three networks interconnected by two routers.

---

†The original literature used the term IP gateway. However, vendors have adopted the term IP router—the two terms are used interchangeably throughout this text.
In this example, router $R_1$ must transfer from network 1 to network 2 all packets destined for computers on either network 2 or network 3. For a large internet composed of many networks, the router's task of making decisions about where to send packets becomes more complex.

The idea of a router seems simple, but it is important because it provides a way to interconnect networks, not just computers. In fact, we have already discovered the principle of interconnection used throughout an internet:

*In a TCP/IP internet, special computers called IP routers or IP gateways provide interconnections among physical networks.*

You might suspect that routers, which must each know how to forward packets toward their destination, are large machines with enough primary or secondary memory to hold information about every computer in the internet to which they attach. In fact, routers used with TCP/IP internets are usually small computers. They often have little disk storage and modest main memories. The trick to building a small internet router lies in the following concept:

*Routers use the destination network, not the destination computer, when forwarding a packet.*

If packet forwarding is based on networks, the amount of information that a router needs to keep is proportional to the number of networks in the internet, not the number of computers.

Because routers play a key role in internet communication, we will return to them in later chapters and discuss the details of how they operate and how they learn about routes. For now, we will assume that it is possible and practical to have correct routes for all networks in each router in the internet. We will also assume that only routers provide connections between physical networks in an internet.
3.7 The User's View

Remember that TCP/IP is designed to provide a universal interconnection among computers independent of the particular networks to which they attach. Thus, we want a user to view an internet as a single, virtual network to which all machines connect despite their physical connections. Figure 3.3a shows how thinking of an internet instead of constituent networks simplifies the details and makes it easy for the user to conceptualize communication. In addition to routers that interconnect physical networks, software is needed on each computer to allow application programs to use an internet as if it were a single, physical network.

The advantage of providing interconnection at the network level now becomes clear. Because application programs that communicate over the internet do not know the details of underlying connections, they can be run without change on any computer. Because the details of each machine's physical network connections are hidden in the internet software, only the internet software needs to change when new physical connections are added or existing connections are removed. In fact, it is possible to optimize the internal structure of the internet by altering physical connections while application programs are executing.

A second advantage of having communication at the network level is more subtle: users do not have to understand, remember, or specify how networks connect or what traffic they carry. Application programs can be written that communicate independent of underlying physical connectivity. In fact, network managers are free to change interior parts of the underlying internet architecture without changing application software in most of the computers attached to the internet (of course, network software must be reconfigured when a computer moves to a new network).

As Figure 3.3b shows, routers do not provide direct connections among all pairs of networks. It may be necessary for traffic traveling from one computer to another to pass through several routers as the traffic crosses intermediate networks. Thus, networks participating in an internet are analogous to highways in the U.S. interstate system: each net agrees to handle transit traffic in exchange for the right to send traffic throughout the internet. Typical users are unaffected and unaware of extra traffic on their local network.

3.8 All Networks Are Equal

Chapter 2 reviewed examples of the network hardware used to build TCP/IP internets, and illustrated the great diversity of technologies. We have described an internet as a collection of cooperative, interconnected networks. It is now important to understand a fundamental concept: from the internet point of view, any communication system capable of transferring packets counts as a single network, independent of its delay and throughput characteristics, maximum packet size, or geographic scale. In particular, Figure 3.3b uses the same small cloud shape to depict each physical network because TCP/IP treats them equally despite their differences. The point is:
The TCP/IP internet protocols treat all networks equally. A Local Area Network like an Ethernet, a Wide Area Network used as a backbone, or a point-to-point link between two computers each count as one network.

Readers unaccustomed to internet architecture may find it difficult to accept such a simplistic view of networks. In essence, TCP/IP defines an abstraction of "network" that hides the details of physical networks; we will learn that such abstractions help make TCP/IP extremely powerful.

![Diagram of internet architecture](image)

Figure 3.3 (a) The user's view of a TCP/IP internet in which each computer appears to attach to a single large network, and (b) the structure of physical networks and routers that provide interconnection.

3.9 The Unanswered Questions

Our sketch of internets leaves many unanswered questions. For example, you might wonder about the exact form of internet addresses assigned to computers or how such addresses relate to the Ethernet, FDDI, or ATM physical hardware addresses described in Chapter 2. The next three chapters confront these questions. They describe the format of IP addresses and illustrate how software on a computer maps between internet addresses and physical addresses. You might also want to know exactly what a packet looks like when it travels through an internet, or what happens when packets arrive too fast for some computer or router to handle. Chapter 7 answers these
questions. Finally, you might wonder how multiple application programs executing concurrently on a single computer can send and receive packets to multiple destinations without becoming entangled in each other’s transmissions or how internet routers learn about routes. All of these questions will be answered as well.

Although it may seem vague now, the direction we are following will let us learn about both the structure and use of internet protocol software. We will examine each part, looking at the concepts and principles as well as technical details. We began by describing the physical communication layer on which an internet is built. Each of the following chapters will explore one part of the internet software, until we understand how all the pieces fit together.

3.10 Summary

An internet is more than a collection of networks interconnected by computers. Internetworking implies that the interconnected systems agree to conventions that allow each computer to communicate with every other computer. In particular, an internet will allow two computers to communicate even if the communication path between them passes across a network to which neither connects directly. Such cooperation is only possible when computers agree on a set of universal identifiers and a set of procedures for moving data to its final destination.

In an internet, interconnections among networks are formed by computers called IP routers, or IP gateways, that attach to two or more networks. A router forwards packets between networks by receiving them from one network and sending them to another.

FOR FURTHER STUDY

Our model of an internetwork comes from Cerf and Cain [1983] and Cerf and Kahn [1974], which describe an internet as a set of networks interconnected by routers and sketch an internet protocol similar to that eventually developed for the TCP/IP protocol suite. More information on the connected Internet architecture can be found in Postel [1980]; Postel, Sunshine, and Chen [1981]; and in Hinden, Haverty, and Sheltzer [1983]. Shoch [1978] presents issues in internetwork naming and addressing. Boggs et al. [1980] describes the internet developed at Xerox PARC, an alternative to the TCP/IP internet we will examine. Cheriton [1983] describes internetworking as it relates to the V-system.