

FURTHER READING:

As a preview for further reading, the following reference has been provided from the pages of the book below:

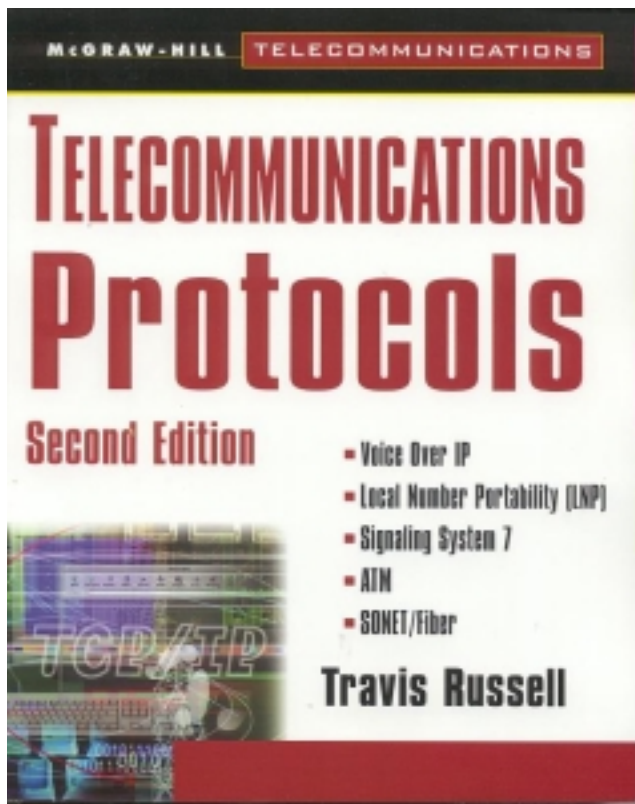
Title: Telecommunications Protocols

Author: Travis Russell

Publisher: McGraw-Hill



ISBN: 0071349154



CHAPTER

4

TCP/IP—Protocol of the Internet

4.1. Introduction

Transmission Control Protocol/Internet Protocol (TCP/IP) is a family of many protocols, each tailored to address specific applications within an internet. The fact that it has gained such popularity in the last few years can be attributed mostly to the success of the Internet. With so many corporate networks connecting to the Internet, vendors are busy developing products for these corporate users.

In addition to the explosive growth of the Internet, many corporations are seeing the value in the TCP/IP technology. TCP/IP offers a robust suite of services and applications, and it is standardized (which resolves a number of interoperability issues). No other technology offers the services of TCP/IP, the ability to interconnect with almost any vendors equipment, and the maturity of the TCP/IP protocols (TCP and IP have been standardized and deployed for over 12 years).

TCP/IP is not a single protocol but is a suite of over 100 protocols, each addressing a specific application within an internet. This is one of the factors which makes TCP/IP so flexible; each protocol can be used independently of the others, making them compatible with other transport technologies.

The fact that TCP/IP was developed in the 1960s does not make it obsolete. Development on TCP/IP protocols continues, as new demands are placed on internets. IP addressing has been strained to the point that new addresses are distributed on a limited basis. Old protocols have been updated and in some cases replaced by newer more robust protocols. There is no doubt that TCP/IP has a long future.

4.1.1. History of TCP/IP

During the 1950s, the U.S. defense network consisted of several main-frame computers linked by point-to-point transmission links. The concern at the time was the survivability of such a network during wartime. The Department of Defense (DOD) was looking for a replacement.

What was needed was a network which could heal itself. If any node, or worse, any section of the network were to suddenly become unreachable, the rest of the network must be able to continue operation. Such a technology did not exist at the time, so the Defense Communications Agency (DCA) began development in the 1960s on behalf of the DOD.

The result was TCP/IP. The first efforts concentrated on the network layer (IP) and the transport layer (TCP). Later, other protocols were added to provide additional functionality to the government internet. The Advanced Research Projects Agency (ARPA) network was first commissioned in 1968, but without TCP/IP. It was several more years before TCP/IP was commissioned in the defense network.

The group was disbanded in 1971, and work was resumed by the Defense Advanced Research Projects Agency (DARPA). DARPA began deploying TCP/IP in all ARPANET computers in 1983. The network was later split into two separate networks, the Military Network (MILNET) and the ARPANET. The network remained in use until the last original node was decommissioned in 1990.

Today, the ARPANET is known as the Internet, and it links millions of nodes that are connected via subnetworks located around the world. The Internet is no longer limited to military use, and the TCP/IP protocol suite continues to evolve through contributions from commercial, scientific, and educational institutions.

One of the smartest decisions made by the DOD was granting the right to distribute TCP/IP code to the University of California in Berkeley. It did not take long for the code to spread to other universities, which were already using the UNIX operating system. Universities began developing additional protocols, providing many of the applications we enjoy today on the Internet.

TCP/IP is a nonproprietary networking protocol, it works on any type platform, and it is flexible enough to use with any other type of technology (TCP/IP can be run over X.25, Frame Relay, or even the Integrated Services Digital Network, or ISDN). This is one of the reasons it has grown in popularity to become one of the most widely deployed networking technologies today.

4.1.2. Overview of Internets

Before we begin our discussion of TCP/IP, an overview of internets is in order. An internet is two or more networks linked together, forming one ubiquitous network. The internet should be transparent to the end user and should interoperate with all other subnetworks. This presents a challenge when mixing equipment from many different vendors. It is for this reason that standards are so important.

Throughout our discussions, we will use the term *internet* to identify any combination of networks interconnected with one another. The

term *Internet* (capitalized) is used to reference the worldwide Internet that is used by many corporations and individuals today to link their individual networks.

Subnetworks are the individual networks within a larger network deployed by different corporations, service providers, and other large network users. A subnet may have a small number of nodes, or it may consist of other subnets (as is the case with many service providers). As a result, hierarchically, an internet may be several layers deep.

It is important to understand the relationships various subnetworks have with one another to understand the inner workings of TCP/IP. The devices used to interconnect the various subnetworks are routers and gateways.

A router receives data packets and forwards them through a port to another network or another part of its own network. A gateway works the same way as a router (from an TCP/IP perspective) but provides access to another network. A gateway can be considered an entry/exit point from one network to another. In other chapters, we discuss other functions for gateways, which will differ from our discussions in this chapter.

4.1.2.1. Autonomous Systems Internets are grouped into autonomous systems. An autonomous system is a group of networks joined together and maintained by a single authority. Autonomous systems are then linked to other autonomous systems by gateways. This provides a hierarchical approach to internets and simplifies the task of routing within an internet.

We will discuss the concept of autonomous systems later when we discuss routing in an internet. There you will see the role of this concept and the devices which are involved. Hospitals and universities are typical examples of where autonomous systems can be found.

4.1.3. Description of TCP/IP

As mentioned earlier, TCP/IP is a suite of protocols. These protocols fall within various layers of the protocol stack. To understand the protocol stack, one should understand the interactions between layers.

Subnetworks are managed at the physical and data link layers. While these layers are not part of the TCP/IP protocol, they do interact with the protocol stack. For example, to reach an IP address, the IP address must first be translated (or resolved) into a Local Area Network (LAN) machine address. If the IP address is part of a subnetwork, the subnet

mask must be determined and the address resolution based on the results of the subnet mask.

Internetworking is managed by the IP protocol at the network layer. IP does not support error control, so it relies on another protocol for this function. The Internet Control Message Protocol (ICMP) provides error correction and flow control for IP. ICMP, although a user of IP, is still considered part of the network layer (ICMP information is encapsulated in IP packets, making it a user of the IP protocol).

TCP and the User Datagram Protocol (UDP) are considered service provider protocols and reside at the transport layer. TCP is a connection-oriented protocol, while UDP is a connectionless protocol. Both rely on the services of IP but do not require IP. For example, TCP and UDP can be transported over X.25 or Frame Relay services (both of which reside at the network layer).

The applications service layer consists of a number of protocols such as File Transfer Protocol (FTP), TELNET, and Network News Transport Protocol (NNTP). These are not really applications themselves but are protocols which interface to the various applications that are necessary to use these services. They provide the communications to remote devices but do not provide the user interface to interact with the various remote services.

All of these protocols encapsulate data into envelopes referred to as protocol data units (PDUs). There are many different labels used for these PDUs at various layers. In this chapter, *segment* will be used to describe PDUs from the transport layer (such as TCP) down to the network layer. In other words, when a protocol passes data from TCP to IP, the data unit is referred to as a segment.

A datagram is used to refer to PDUs passed from the network layer down to the data link layer (as in from IP to Ethernet). Datagrams sometimes refer to packets in connectionless protocols, such as UDP, but in this chapter they will refer to data units at the IP to data link layer.

Once a data unit has passed through the various layers and is sent to the physical layer, it is considered a frame. Once the data unit has been passed over the network, it is referred to as a packet. These labels are pretty much consistent with other technologies as well and will be used throughout this book as described here, unless otherwise noted.

To interact with other parts of a host (software modules), protocols must interact with interfaces provided by the operating system. The operating system provides ports as entries to applications. As a data unit is passed to the application layer, the operating system provides a connection to the application by way of a logical port. The connection is

established and maintained throughout the transaction period until data segments have been terminated.

A socket identifies an endpoint communications process. In order for communications to pass from one machine to another, a port must be connected and a socket defined. Internet ports are usually predefined (0 to 255) for well-known applications (such as FTP and NNTP). Undefined ports are provided as well, allowing operating systems to define their own ports when necessary.

Now that we understand some the terminology used with this technology, let us look at some of the advantages of TCP/IP. When data is sent from an application down to the transport layer, the data may be too big to fit into one data unit. TCP provides a service called fragmentation and reassembly, to handle this problem. The data is divided into evenly sized data units and is then passed to the network layer for further processing.

At the network layer, the individual data units may require further fragmenting. There is nothing wrong with this practice since protocols work within a peer-to-peer relationship. Data fragments created at the transport layer cannot be processed at the network layer; they must be passed to the transport layer before the fragments can be reassembled and processed. So fragmenting at various layers does not pose a problem.

When the data unit is passed over the network, it must pass through routers and gateways to reach its destination. It is possible that a data unit may pass through a network that will not accept the size of the IP fragments, and the data units will have to be fragmented further. This is not a problem for IP or TCP, which manages fragmented data at both the network and the transport layers.

When fragments are received by a host, the IP layer looks for the other parts of the data. Timers are used to determine when fragments are considered as lost, and when a time-out occurs, the received data is thrown out and an error message is sent to the source to generate retransmission of all of the fragments.

In addition to this handling of data fragments, multiple addressing conventions can be supported between various subnets. Different routing methods can be used within the various subnets with absolutely no effect on the end-to-end transmission itself. For example, a message may be passed to a subnet using source routing (the source defines the path to take to reach a destination), even though the source originated the message using nonsource routing (this is discussed in more detail in Sec. 4.3.4). In short, a message does not have to follow one method of routing all the way through the network. Each portion of the network can use

any routing mechanism it wants without affecting the delivery of the original data unit.

By the same token, various services can be provided from one subnet to the next. A data unit can be originated using the connection-oriented services of TCP, but along the way a subnet can use connectionless UDP to pass the message through its own network as long as TCP services are used at the destination.

There are many other advantages to using TCP/IP in an internet-network environment. We will examine a number of them as we discuss the other functionality of these protocols. In short, TCP/IP was designed to support data communications through a number of nonrelated networks, ensuring reliable delivery of data even when networks fail along the way.

4.2. TCP/IP Standards

TCP/IP is a national standard, developed and standardized in the United States. Standardization is important because it ensures that various vendors' equipment will be compatible within the same network (provided the vendors followed the standards). TCP/IP standards are a little harder to track than other standards, partly because there is no central authority responsible for developing and writing the standards [as in the American National Standards Institute (ANSI)] and partly because of the way standards evolve.

In this section we will discuss how TCP/IP standards are written and how they evolve from contributions to standards.

4.2.1. Standards Documentation

TCP/IP standards are submitted in the form of a Request for Comments (RFCs). An RFC can be submitted by anyone and does not become a standard right away. In fact, there are thousands of RFCs available on various subjects regarding TCP/IP, but not all of them are standards. Many of them have not been implemented. This makes it very difficult to determine which are approved standards and which are not.

The first step in the standards process is the submission of a preliminary draft RFC. The draft is made available to anyone wishing to add to or comment on it (which is why it is called a Request for Comments).

Drafts can be found on the Internet at ds.internic.net (an AT&T server) and can be freely downloaded. Many network providers also provide access to these RFCs through their own servers.

Once an RFC has been submitted, the Internet Engineering Task Force (IETF) reviews it and makes a recommendation to make the RFC a standard. The document then becomes a draft standard. It takes about 6 months to move from preliminary draft to draft standard.

After another 4 months of review, and actual implementation, the draft standard can be moved to a published standard. This is again decided by the IETF, as well as the Internet Advisory Board (IAB). These committees and their activities are discussed in more detail below.

The IAB also publishes the IAB Official Protocol Standards Document List, which provides the status (preliminary draft, draft standard, or standard) of all RFCs. This is the best source for tracking actual TCP/IP standards. Before implementing any RFC, check this document first, or you may be implementing a nonstandard technology.

This process is far quicker than those used by other standard organizations. The cycle of ANSI or the International Telecommunications Union (ITU) standards may take 4 to 10 years before a standard is published. This is due in part to the use of a committee, rather than the actual user community as in the Internet.

4.2.2. Standards Groups

The Internet provides an excellent proving ground for TCP/IP standards. With its many variations and millions of users, the Internet is the best source for testing and validating new technologies, as long as the preliminary implementation is isolated to a subnet and does not affect the entire Internet.

There are several organizations which oversee the activities of the Internet and the standards process used to implement new technologies in the Internet. These organizations look after all TCP/IP standards.

The IAB oversees the entire development process. Using the resources of the IETF, they determine which RFCs will actually become standards and oversee the activities of the IETF.

The IETF evaluates RFCs and provides technical expertise to the IAB. Consisting of engineers and networking professionals, the IETF is the technical arm of the IAB. They also design and implement new standards under the direction of the Internet Engineering Steering Committee (IESC).

The IESC consists of IETF leadership and provides direction for the IETF staff. It is chaired by committee leaders and works under the direct input of the IAB.

The Internet Research Task Force (IRTF) oversees long-term issues of TCP/IP network architecture. They work under the direction of another steering committee, the Internet Research Steering Group (IRSG).

One other organization which works apart from those mentioned above is the InterNIC. The InterNIC provides services to the Internet community, such as IP address administration. Domain names are also issued by the InterNIC. The InterNIC is sponsored by the National Science Foundation (NSF). They are active in governmental issues as well and represent the Internet community in regulation issues. Now that we understand the standards and the various organizations responsible for the standards, let us look at the protocols themselves.

4.3. Internet Protocol

The IP resides within layer 3 (network layer) of the OSI Model. It provides end-to-end transport of data units through internets using connectionless services. Being connectionless, IP does not provide reliable data transfer, but this is not an issue if the upper layers provide reliability and error control.

An IP host must encapsulate data into IP headers, which are then passed to the data link (such as Ethernet). The protocol at the data link layer then encapsulates the IP header with the data into its own data unit (the datagram). The datagram is then passed down to the physical layer, where it is passed over the network as a serial bit stream (with possible encapsulation again, depending on the technology used).

For data to leave the local network, it must be sent to a router. Routers are network layer devices and are capable of processing the Ethernet and the IP headers. If the data is to be passed to another network, the Ethernet (or data link header) is stripped from the data, and the IP header is then processed.

Before transmitting the data over a port to the next network, the router must create a new IP header and place the data (consisting of the TCP header, possibly application header, and user data) into the IP header. The datagram is then given to the data link layer (which may now be X.25, ISDN, Frame Relay, or even Switched 56), and the whole process is repeated.

IP does have its limitations, the biggest being the number of addresses available. As you will see when we discuss IP addresses, there is a severe limitation in the number of addresses that IP can support. This issue has brought about the need for a replacement to IP. Internet Protocol next generation (IPng) provides a 16-byte address rather than a 4-byte one.

The primary function of IP is to provide routing information for data being transported through internets. Any error control is provided by the Internet Control Message Protocol (ICMP), which resides at layer 3 as well. This protocol does not provide error control but merely reports errors to the originating hosts.

IP is not a requirement for TCP. The TCP protocol can use almost any network layer protocol for delivery as long as the protocol is capable of providing routing services and supports the interfaces between the two layers. Remember that the concept of layering was to allow various layers of a protocol to be changed without affecting the layers above or below it.

4.3.1. IP Header

Figure 4.1 depicts the IP header and its fields. The first field in the IP header is the *version* field. This is used to identify which version of IP was used to create the header. This is important in internets because not every network is running the same version of a protocol. If the IP header was created in a network using the latest version of IP, it may contain information not recognizable by an older version of IP.

When this occurs, the receiving network (running the older version of IP) knows to ignore unrecognizable fields because the version field indicates a version newer than its own. This is valuable information for an internet and can be found in a number of the TCP/IP protocols.

Figure 4.1
IP Header Format

IP Header			
Version	Length	Service Type	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source IP Address			
Destination IP Address			
IP Options (optional)			Padding
Data			