

TCP/IP: The Sub-Application Layer Protocols

by Eddie Konczal
CSC-200: Networking Technologies
Professor Jeff Spector
Fall 1996

TCP/IP: The Sub-Application Layer Protocols

Introduction

TCP/IP is a set, or suite, of over 100 protocols governing communications between computer systems and allowing dissimilar systems to be networked; it has been aptly described as “the glue holding ...the Internet...together.” (Hahn, 1996, p. 20). The genesis of TCP/IP occurred in the late 1960s, when the Department of Defense realized the need for computer systems of varying architectures to communicate over similarly diverse types of transmission media. By 1979, the protocols of the TCP/IP suite began to assume their current form, having been developed over the course of the decade by the Defense Advanced Research Projects Agency (DARPA) for use on its research network, ARPANET (Comer, p. 5). In 1983, DARPA decreed that all computer systems needing to communicate with ARPANET had to implement the TCP/IP protocol suite; to promote this implementation, DARPA funded TCP/IP research and development on universities across the country (Comer, p. 6). In this manner the foundation for the modern-day Internet was established. Today, the TCP/IP suite allows communication not only over the worldwide Internet, but also over corporate intranetworks or “intranets.” The UNIX platform, with native TCP/IP, had long dominated the intranet market; to compete, vendors of PC-based network operating systems (such as Microsoft’s Windows NT and Novell’s IntraNetWare) have included TCP/IP in their products (Milne, p. 76).

TCP/IP derives its name from the two most important protocols in the suite, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). There are many other protocols that comprise TCP/IP, including application-level or “process” protocols which perform key functions such as file transfer (FTP), electronic mail delivery (SMTP) and terminal emulation (Telnet). The focus of this paper is on TCP, IP, and related sub-application level protocols, and how they enable communications to occur throughout the Internet and similar networks.

Overview of TCP/IP Layering Model

TCP/IP can be visualized as a stack of protocols divided into layers, similar to the International Organization for Standards' Open Systems Interconnection (OSI) reference model (See Appendix A, "TCP/IP Layering Model"). The OSI model outlines "an ideal computer network system in which communication...occurs between processes at discrete and identifiable layers." (Novell TCP/IP Guide, p. B-1). Data at the source computer is passed downward through seven functional layers: Application, Presentation, Session, Transport, Network, Data-Link, and Physical. At each of these layers, protocols attach control information, in the form of headers, to the data. At the destination computer, the corresponding protocols analyze the headers, discard them, and pass the data upward through the protocol stack until the data can be processed by the computer.

The TCP/IP model is not an exact implementation of the OSI model. This should not be surprising, as development of the TCP/IP suite of protocols was well underway when the OSI model was introduced in 1979. Conceptually, the TCP/IP model contains four functional layers to the OSI model's seven. At the top of the stack is the *Application* layer, which combines the functionality of the OSI model's Application, Presentation, and Session layers. Protocols at the Application layer of TCP/IP are considered "process protocols" and perform functions that are apparent to the user, such as file transfer, mail transfer, and terminal emulation. These protocols pass data bytes downward to the *Transport Layer*, where TCP is responsible for dividing the stream of bytes into segments or *transport control packets* (Comer, p. 111). These packets are sequenced so that they can be reassembled in correct order at the destination. TCP then passes the packets down to the *Internet* layer, where IP attaches a header including the 32-bit Internet addresses of the source and destination computers. IP then passes the packet to the *Network Interface* layer, which consists not of TCP/IP protocols but rather of the device drivers needed to configure the computer's network interface card. At the Network Interface layer, packets are converted to network frames and begin their physical journey throughout the Internet. Once they reach their destination, the IP and TCP headers are checked for errors and discarded at the respective layers. If the data is transmitted accurately, it is passed to the appropriate application of the destination computer, "just as if it were directly connected to the application on the source computer." (Novell TCP/IP Guide, p. B-5)

The above description is an illustrative oversimplification of the process by which TCP/IP enables communications throughout a computer network. The following sections describe in greater detail the functions and services provided by the TCP/IP suite at the Transport and Internet layers.

Internet Layer Protocols

The protocols that form the Internet Layer of TCP/IP govern the actual routing of data packets throughout the Internet or an intranet. They are so essential to the functionality of TCP/IP that the protocol suite is commonly called “the Internet protocol suite.”

ARP. The Address Resolution Protocol (ARP) is designed to solve the “address resolution problem,” which goes as follows: Actual data transmission on the Internet occurs between computers’ physical addresses, or MAC addresses (for Media Access Control), which are usually hard-coded into a computer’s network interface card. However, TCP/IP based applications need to communicate based on 32-bit Internet (or IP) addresses. Before communication occurs, a source computer will know the IP address of the recipient, but not the recipient’s physical address.

ARP solves the address resolution dilemma through a process known as *dynamic binding*, which allows a computer to match IP addresses to MAC addresses as needed, without resorting to a centralized database. (Comer, p. 51). When a computer needs to transmit a packet to another computer with a known IP address and an unknown MAC address, an ARP request will be generated. The ARP request will contain the source computer’s IP and MAC addresses, the destination computer’s IP address, and a blank field for the unknown physical address (*Target Hardware Address*). The ARP request is broadcast to all computers on the network; only the computer whose IP address is contained in the request will respond. This computer will fill in the target hardware address field with its own physical address, convert the request to a reply, and send the ARP message back to the source computer. Once the source computer knows the IP - to - MAC address binding of the recipient, it stores the binding in an ARP cache (Comer, p. 52) for future reference. The source can then transmit data to the destination. Future ARP requests to the destination are unnecessary, as the source can obtain the address binding from the ARP cache before transmitting additional packets.

RARP. Some networked computers may be diskless workstations, with no storage space to keep their own IP address. RARP (Reverse Address Resolution Protocol), is designed to allow such machines to obtain their IP address from another computer. When a diskless machine boots, it generates a RARP request. The format is nearly identical to that of an ARP request; the key difference is that the sender of the request identifies itself as the target machine and supplies its physical address in the Target Hardware Address field. (Comer, p. 60) The RARP request is sent to all machines on the sender's physical network; for the request to be filled, a *RARP server* capable of supplying other machines with IP addresses must reside on the physical network. The RARP server responds to the sender by filling in the Target Internet Address field of the RARP broadcast and returning the RARP message to the sender. It is possible to design a network with multiple RARP servers. The additional servers could be designated as secondary RARP servers, programmed to respond only if network conditions indicate that the primary server is down or overloaded with requests (Comer, p. 63).

IP. The Internet Protocol (IP) lies at the heart of Internet data transmissions, as its name might suggest. It is best described as an “unreliable, best effort, connectionless, packet-delivery system.” (Comer, p. 67). It is considered “unreliable” because it includes no way of guaranteeing delivery of data. IP is a “best-effort” service because it always makes an “earnest” attempt to deliver packets, and only fails when unforeseen network errors occur (Comer, p. 67). IP delivery is “connectionless” because it routes packets (or *IP datagrams*, as they are called at the Internet layer) independently of any part they play in an ongoing conversation between two networked machines. IP datagrams that are part of the same conversation may take different paths through the Internet, based on changing variables such as network traffic (Schatt, p. 104). In addition to routing datagrams, IP is capable of subdividing them into smaller units when it is known that the datagram will pass through an intermediary computer (*router*) that cannot handle the entire datagram, which may be up to 65,356 bytes in size (Brewster, p. 110).

IP receives packets from the Transport layer protocols and converts them to IP datagrams by attaching standardized headers. Various fields in the IP header determine the manner in which the packet will be routed through the Internet (see Appendix B, “The IP Header”). Some of the field names are self-

explanatory, such as *Version*, *Header Length* (measured in 32-bit units), *Total Length of Datagram* (measured in bytes), and *Source IP Address* and *Destination IP Address* (both 32-bit numbers when IP version 4 is used). The *Type of Service* field provides guidelines for routing datagrams, based on criteria such as desired precedence, delay, throughput, and reliability. (Comer, p. 70) Three fields oversee fragmentation of datagrams: *Identification*, a unique value assigned to each datagram that allows the destination computer to identify fragments; *Flags*, which contains bits determining whether a datagram can be fragmented, and whether there are more fragments to follow; and *Fragment Offset*, which numbers the fragments so they can be reassembled correctly into the original datagram. The *Time to Live* field contains an estimate, in seconds, of how long the datagram should take to reach its destination. This value decreases each time the datagram passes through a router; when it reaches zero the packet is discarded. The *Protocol* field identifies which Transport-layer protocol is being serviced by IP (TCP or UDP). The *Checksum* field contains a value that is a function of the data in the header. It is computed at the source, at each router, and at the destination; if any computation does not match what is in the field (accounting for the decreasing *Time to live* value), an error is indicated and the datagram is discarded (Brewster, p. 110)..The *Options* field is primarily for diagnostic purposes, and allows the path of the datagram throughout the network to be tracked (record routing) or specified in advance (source routing). The *Padding* field is filled with enough zeros to make the header a multiple of 32 bits.

In late 1994, IP Version 6, or IPng (for IP Next Generation), was finalized. It is currently found mainly in developmental environments; there has been reluctance to embrace v6 as the new standard due to the software and hardware costs needed to implement it. Among the changes are a simplified header, and expanded fields to allow for 128-bit Source and Destination addresses. The new address size will allow encapsulation of MAC addresses within IP addresses, making it easier for LAN administrators to assimilate or *autoconfigure* new PCs (Strauss, p. 72). Another improvement in IP v6 is a *Flow Label* field in the header, enabling IP routing to simulate virtual circuits and thereby allocate bandwidth for multimedia and videoconferencing applications (Strauss, p. 72). Further development of the TCP/IP suite will be necessary in order to take advantage of the flow label feature.

ICMP. The Internet Control Message Protocol (ICMP) assists IP routing by providing information on unexpected network conditions, such as equipment failure or network traffic. Generally, ICMP messages are sent by routers to source computers when the source's data transmission failed for some reason. ICMP messages travel as data within IP datagrams; unlike usual datagrams, however, the message data is not generated or processed by a higher level protocol or application, but by the Internet layer of the communicating machines (Comer, p. 90).

ICMP messages can take various forms, depending on the type of message being generated (Comer, p. 91). The first four bytes of each message always contain the same three fields: *Type* (of ICMP message); *Code* (a clarification of the type field); and *Checksum* (for error detection). ICMP message types received by source machines from routers include: *Source Quench*, a request for the source to slow its rate of transmission; *Time Exceeded*, sent by a router that has discarded a datagram whose Time to live has expired; *Redirect*, sent to inform the source that its routes are not optimum; *Destination Unreachable*, sent by routers unable to deliver a datagram for various reasons (network unreachable, host unreachable, route failed, etc.); and *Parameter Problem*, sent when the router has computed an error in the IP header. ICMP messages can also be sent by sources to routers or destinations. These include *Echo Request*, sent in order to test if a destination is reachable; *Timestamp Request*, sent to compute delay times in transmission; *Information Request*, an alternative to RARP; and *Address Mask Request*, sent to learn which part of the source's IP address corresponds to the physical network. The third type of ICMP message takes the form of a reply to one of the four request types mentioned above.

Transport Layer Protocols

The TCP/IP Transport layer provides an interface between the Internet layer's packet delivery system and the application layer protocols. While IP is responsible for routing datagrams to the correct destination, the Transport layer protocols take the delivery a step further by routing the data to the correct application on the destination computer (Brewster, p. 110).

TCP. Of the two core Transport Layer protocols, TCP (Transmission Control Protocol) is by far the more powerful. TCP relies on IP for the actual datagram delivery, but adds something to the

protocol suite that IP cannot alone provide: reliability. TCP achieves reliability by simulating a permanent hardware connection between two communicating computers. The connection is established using a three-way acknowledgment or *handshake* allowing the source and destination to agree on the terms of transmission (Brewster, p. 112). In addition to establishing and terminating connections, TCP is responsible for tracking the progress of data transmissions and recovering from errors by initiating retransmission of lost packets (Schatt, p. 105).

At the source computer, TCP receives a stream of data from a particular application. The data are subdivided by TCP into segments of up to 64 kilobytes (Brewster, p. 112). Before passing the segments on to the Internet layer for delivery as datagrams, TCP attaches its own standard header to each (see Appendix C, “The TCP Header”). Within the header, fields for *Source Port* and *Destination Port* identify the application layer protocols sending and receiving the data, respectively. Examples of port numbers include 21 for File Transfer Protocol (FTP), 23 for Telnet, and 25 for Simple Mail Transfer Protocol (SMTP) (Comer, p. 149). The *Sequence number* field allows segments to be reassembled correctly into a stream of data at the destination. *Acknowledgment number* is used by the source to track successfully delivered segments; unacknowledged segments are retransmitted after a time-out interval has expired (McConnell, p. 183). *Data offset* indicates the length of the TCP header in 32-bit units. The *Code* field contains bits set to indicate whether the *Urgent pointer* and *Acknowledgment number* fields are valid, whether to reset the connection, and whether the sender is at the end of a transmission. The *Window* field provides for flow control by allowing communicating devices to agree on the number of bytes in transmitted segments. As with IP, a *Checksum* field is computed for error detection. The *Urgent pointer* field identifies the location of priority data (such as a keyboard interrupt signal) within the segment. The *options* field allows the sender TCP to communicate with the destination TCP on variables such as maximum segment size. Finally, *padding* is provided (if necessary) in the form of zeros to bring the total header size to a multiple of 32 bits.

UDP. The User Datagram Protocol (UDP) is a simplified Transport level protocol, used when the power of TCP is not required. If TCP can be compared to Federal Express delivery, UDP analogous

to the US Postal service. UDP is a connectionless, unreliable service, and is used to transmit individual segments of data that do not need to be assembled into a larger conversation between applications. (Schatt, p. 107). The UDP header is a mere 64 bits in size, and contains only four fields: *Source Port*, *Destination Port*, *Length* (of UDP datagram in bytes), and *UDP Checksum*. (Comer, p. 121). The port number fields are independent of their counterparts in the TCP header; however, in most cases UDP ports have been standardized to match those used by TCP (Comer, p. 149).

Conclusions

Developed by the Department of Defense for use on the Internet, the TCP/IP suite of protocols provides a powerful software platform for internetworking computer devices of various types over different kinds of transmission media. IP provides efficient routing of data packets over the network, while TCP guarantees reliability of transmission through simulation of connection-based services. Other protocols, such as ARP, RARP, ICMP, and UDP, provide additional services at the Transport and Internet layers of the protocol stack. The adaptability of TCP/IP to many physical network types has made it the protocol suite of choice for many corporate intranets. The future of TCP/IP will include the possibility for autoconfiguration of local area networks, as well as the ability to run multimedia and videoconferencing applications over IP-based networks. Such capabilities must wait, however, for the implementation of IP Version 6.

Bibliography

- Brewster, R.L. **Data Communications and Networks 3**. Institution of Electrical Engineers (Trowbridge, Wiltshire), 1994.
- Chernicoff, David P. "A TCP/IP map of the data superhighway." *PC Week*, vol. 10 no. 51 (Dec. 27, 1993).
- Comer, Douglas. **Internetworking with TCP/IP: Principles, Protocols, and Architecture**. Prentice Hall (Englewood Cliffs, NJ), 1988.
- Hahn, Harley. **The Internet: Complete Reference**. Osbourne/McGraw-Hill (New York), 1996.
- McConnell, John. **Internetworking Computer Systems**. Prentice Hall (Englewood Cliffs, NJ), 1988.
- Milne, Jay. "NOS: Stepping Up to Big Challenges." *Networking Computing*, vol. 7 no. 19 (December 1, 1996).
- NOVELL NetWare v3.11 TCP/IP Transport Supervisor's Guide**. Novell, Inc. (Provo, UT), 1991.
- Schatt, Stan. **Understanding Network Management: Strategies and Solutions..** Windcrest/McGraw Hill (Blue Ridge Summit, PA), 1993.
- Strauss, Paul. "Just When You THOUGHT IP Was Safe..." *Datamation*, vol. 40 no. 21 (Nov. 1, 1994).