0 TCP/IP overview

From these assumptions comes the fundamental structure of the Internet: a packet switched communications facility in which a number of distinguishable networks are connected together using packet communications processors called gateways which implement a store and forward packet forwarding algorithm.

David D. Clark

0.1 The Internet

The Internet is a global information system consisting of millions of computer networks around the world. Users of the Internet can exchange email, access to the resources on a remote computer, browse web pages, stream live video or audio, and publish information for other users. With the evolution of *e-commerce*, many companies are providing services over the Internet, such as on-line banking, financial transactions, shopping, and on-line auctions. In parallel with the expansion in services provided, there has been an exponential increase in the size of the Internet. In addition, various types of electronic devices are being connected to the Internet, such as cell phones, personal digital assistants (PDA), and even TVs and refrigerators.

Today's Internet evolved from the ARPANET sponsored by the Advanced Research Projects Agency (ARPA) in the late 1960s with only four nodes. The Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite, first proposed by Cerf and Kahn in [1], was adopted for the ARPANET in 1983. In 1984, NSF funded a TCP/IP based backbone network, called NSFNET, which became the successor of the ARPANET. The Internet became completely commercial in 1995. The term "Internet" is now used to refer to the global computer network loosely connected together using packet switching technology and based on the TCP/IP protocol suite.

The Internet is administered by a number of groups. These groups control the TCP/IP protocols, develop and approve new standards, and assign Internet addresses and other resources. Some of the groups are listed here.

- Internet Society (ISOC). This is a professional membership organization of Internet experts that comments on policies and practices, and oversees a number of other boards and task forces dealing with network policy issues.
- Internet Architecture Board (IAB). The IAB is responsible for defining the overall architecture of the Internet, providing guidance and broad direction to the IETF (see below).
- Internet Engineering Task Force (IETF). The IETF is responsible for protocol engineering and development.
- Internet Research Task Force (IRTF). The IRTF is responsible for focused, long-term research.
- Internet Corporation for Assigned Names and Numbers (ICANN). The ICANN has responsibility for Internet Protocol (IP) address space allocation, protocol identifier assignment, generic and country code Top-Level Domain name system management, and root server system management functions. These services were originally performed by the Internet Assigned Numbers Authority (IANA) and other entities. ICANN now performs the IANA function.
- Internet Network Information Center (InterNIC). The InterNIC is operated by ICANN to provide information regarding Internet domain name registration services.

The Internet standards are published as *Request for Comments* (RFC), in order to emphasize the point that "the basic ground rules were that anyone could say anything and that nothing was official" [2]. All RFCs are available at the IETF's website `http://www.ietf.org/`. Usually, a new technology is first proposed as an *Internet Draft*, which expires in six months. If the Internet Draft gains continuous interest and support from ISOC or the industry, it will be promoted to a RFC, then to a *Proposed Standard*, and then a *Draft Standard*. Finally, if the proposal passes all the tests, it will be published as an *Internet Standard* by IAB.

0.2 TCP/IP protocols

The task of information exchange between computers consists of various functions and has tremendous complexity. It is impractical, if not

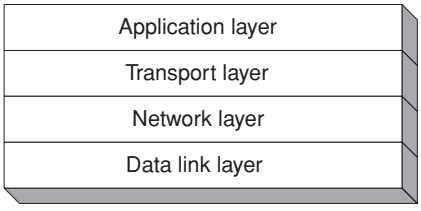

Figure 0.1. The TCP/IP protocol stack.

impossible, to implement all these functions in a single module. Instead, a *divide-and-conquer* approach was adopted. The communication task is broken up into subtasks and organized in a hierarchical way according to their dependencies to each other. More specifically, the subtasks, each of which is responsible for a facet of communication, are organized into different layers. Each higher layer uses the service provided by its lower layers, and provides service to the layers above it. The service is provided to the higher layer transparently, while heterogeneity and details are hidden from the higher layers. A protocol is used for communication between entities in different systems, which typically defines the operation of a subtask within a layer.

TCP/IP protocols, also known more formally as the *Internet Protocol Suite*, facilitates communications across interconnected, heterogeneous computer networks. It is a combination of different protocols, which are normally organized into four layers as shown in Fig. 0.1. The responsibility and relevant protocols at each layer are now given.

- The *application layer* consists of a wide variety of applications, among which are the following.
  - Hypertext Transfer Protocol (HTTP). Provides the World Wide Web (WWW) service.
  - Telnet. Used for remote access to a computer.
  - Domain Name System (DNS). Distributed service that translates between domain names and IP addresses.
  - Simple Network Management Protocol (SNMP). A protocol used for managing network devices, locally or remotely.
  - Dynamic Host Configuration Protocol (DHCP). A protocol automating the configuration of network interfaces.
- The *transport layer* provides data transport for the application layer, including the following.
  - Transmission Control Protocol (TCP). Provides *reliable* data transmission by means of *connection-oriented* data delivery over an IP network.

  - User Datagram Protocol (UDP). A *connectionless* protocol, which is simpler than TCP and does not guarantee reliability.
- The *network layer* handles routing of packets across the networks, including the following.
  - Internet Protocol (IP). The "workhorse" of the TCP/IP protocol stack, which provides *unreliable* and *connectionless* service.
  - Internet Control Message Protocol (ICMP). Used for error and control messages.
  - Internet Group Management Protocol (IGMP). Used for *multicast* membership management.
- The *link layer* handles all the hardware details to provide data transmission for the network layer. Network layer protocols can be supported by various link layer technologies, such as those listed here.
  - Ethernet. A popular *multiple access* local area network protocol.
  - Wireless LAN. A wireless multiple access local area network based the IEEE 802.11 standards.
  - Point to Point Protocol (PPP). A *point-to-point* protocol connecting pairs of hosts.
  - Address Resolution Protocol (ARP). Responsible for resolving network layer addresses.

Figure 0.2 shows the relationship among protocols in different layers. We will discuss these protocols in more detail in later chapters.

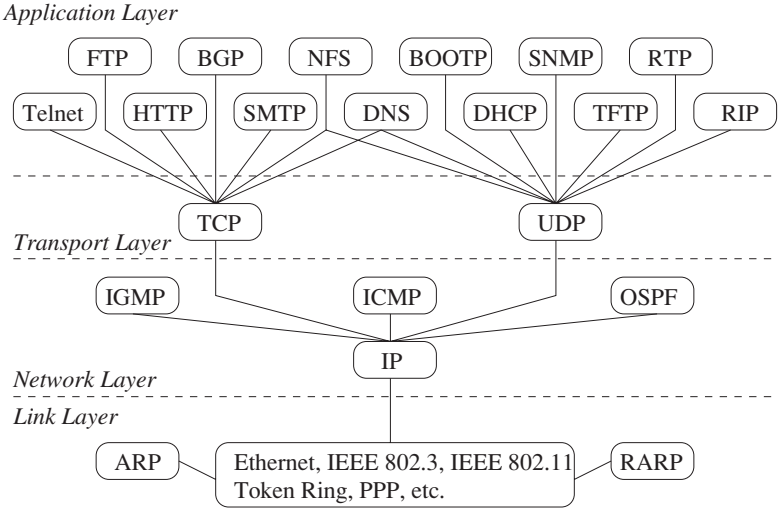

Figure 0.2. The TCP/IP protocols.

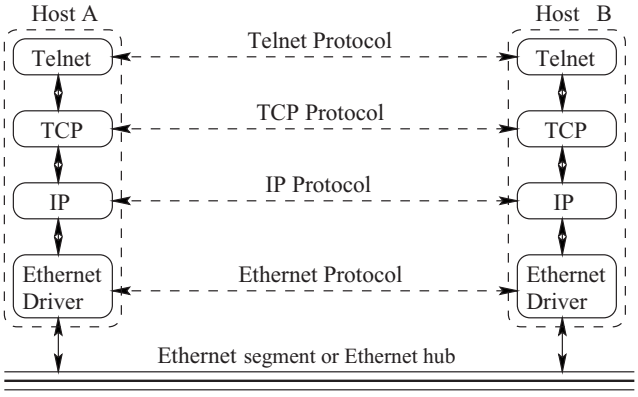

Figure 0.3. An illustration of the layers involved when two hosts communicate over the same Ethernet segment or over an Ethernet hub.

0.3 Internetworking devices

The Internet is a collection of computers connected by internetworking devices. According to their functionality and the layers at which they are operating, such devices can be classified as *hubs*, *bridges*, *switches*, and *routers*.

Hubs are physical layer devices, used to connect multiple hosts. A hub simply copies frames received from a port to all other ports, thus emulating a broadcast medium. Bridges, sometimes called *layer two switches*,[1] are link layer devices. They do not examine upper layer information, and can therefore forward traffic rapidly. Bridges can be used to connect distant stations and thus extend the effective size of a network. Bridges are further discussed in Chapter 3.

Routers, also called *layer three switches*, are network layer devices incorporating the routing function. Each router maintains a *routing table*, each entry of which contains a destination address and a next-hop address. None of the routers has information for the complete route to a destination. When a packet arrives, the router checks its routing table for an entry that matches the destination address, and then forwards the packet to the next-hop address. Routing is further discussed in Chapter 4.

Figure 0.3 shows the layers involved in communication between two hosts when they are connected by an Ethernet hub. The hosts can directly

[1] The industry, confusingly, also uses the term *smart hubs* for switches.

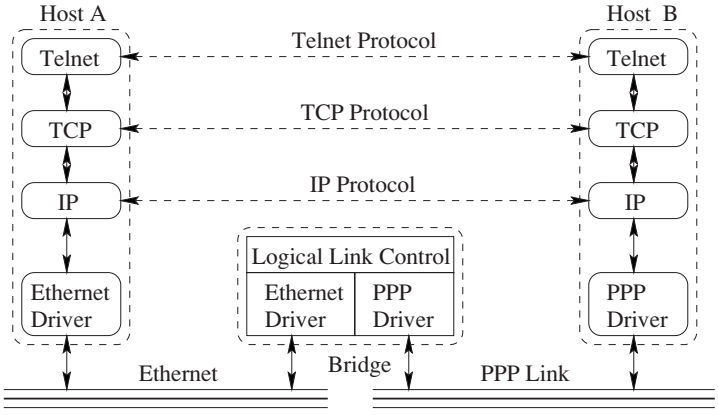

Figure 0.4. An illustration of the layers involved when two hosts communicate through a bridge.

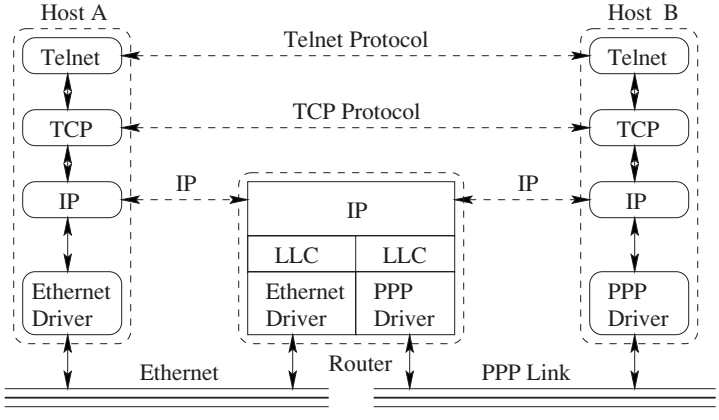

Figure 0.5. An illustration of the layers involved when two hosts communicate through a router.

communicate with each other since the same link layer protocol is used. Figure 0.4 shows how two different network segments using different link layer technologies are interconnected using a bridge, which interfaces between the link layer protocols and performs frame forwarding. Figure 0.5 shows how two networks are interconnected by a router, which not only performs the layer two functions as in Fig. 0.4, but also handles routing and packet forwarding, which are the major functions of the network layer.

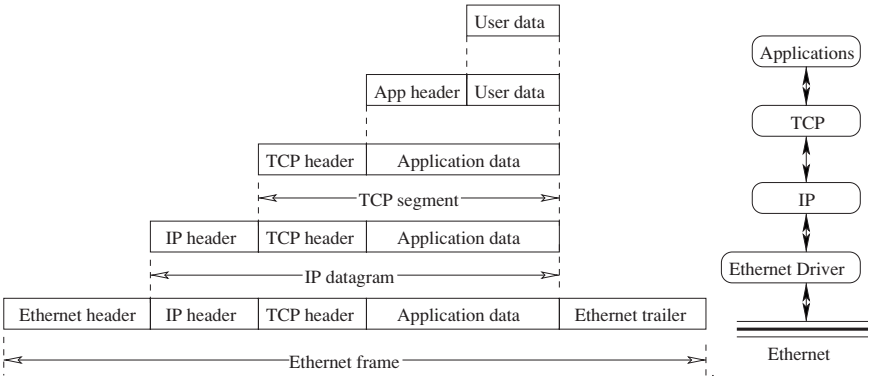

Figure 0.6. Encapsulation of user data through the layers.

As shown in the examples above, a single network segment is formed using hubs. A number of network segments are interconnected by bridges and switches to construct an extended local area network associated with typically a corporate or other institutional networks. Wide Area Networks (WAN) are constructed by connecting the routers of different enterprise networks using high-speed, point-to-point connections. These connections are usually set up over an SDH/SONET circuit-switched network.

0.4 Encapsulation and multiplexing

In a source host, the application data is sent down through the layers in the protocol stack, where each layer adds a header (and maybe a trailer) to the data received from its higher layer (called the *protocol data unit* (PDU)). The header contains information used for the control functions that are defined and implemented in this layer. This *encapsulation* process is shown in Fig. 0.6. When the packet arrives at the destination, it is sent up through the same protocol stack. At each layer, the corresponding header and/or trailer are stripped and processed. Then, the recovered higher layer data is delivered to the upper layer.

As explained in Section 0.2, one of the advantages of the layered structure is the great flexibility it provides for network design and management. For example, different higher layer protocols can use the service provided by the same lower layer protocol, and the same higher layer protocol can use the service provided by different lower layer protocols. In the first

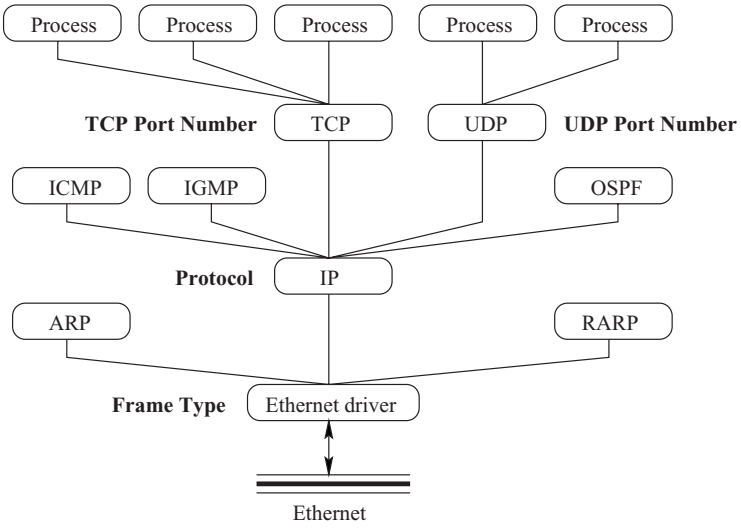

Figure 0.7. Multiplexing/demultiplexing in the layers.

case, each packet sent down to the lower layer should have an identifier indicating which higher layer module it belongs to. As is shown in Fig. 0.7, multiplexing and demultiplexing is performed at different layers using the information carried in the packet headers. For example, a communication process running in a host is assigned a unique *port number*, which is carried by all the packets generated by or destined to this process. Transport layer protocols such as TCP or UDP determine whether a packet is destined for this process by checking the port number field in the transport layer header. In the IP case, each protocol using IP is assigned a unique *protocol number*, which is carried in the `Protocol` IP header field in every packet generated by the protocol. By examining the value of this field of an incoming IP datagram, the type of payload can be determined. A field called *Frame Type* in the Ethernet header is used for multiplexing and demultiplexing at this level.

0.5 Naming and addressing

In order to enable the processes in different computers to communicate with each other, naming and addressing is used to uniquely identify them. As discussed in the previous section, a process running in a host can be

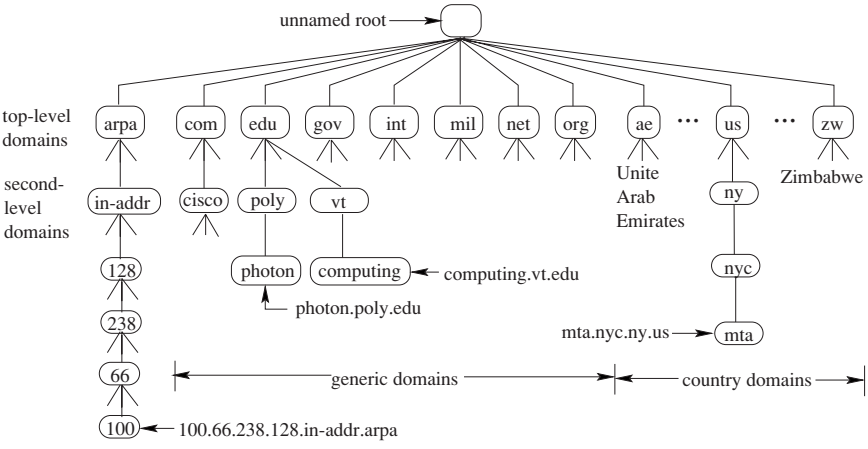

Figure 0.8. The organization of the domain name space.

identified by its port number. Furthermore, a host is identified by a *domain name*, while each network interface is assigned a unique IP address and a *physical*, or MAC, address.

0.5.1 Domain name

In the application layer, an alphanumeric *domain name* is used to identify a host. Since this layer directly interacts with users, a domain name is more user friendly than numeric addressing schemes, i.e., it is easier to remember and less prone to errors in typing.

Domain names are hierarchically organized, as shown in Fig. 0.8. In the tree structure, the root node has a null label, while each nonroot node has a label of up to 63 characters. As shown in Fig. 0.8, there are three types of domains. The `arpa` domain is mainly used for mapping an IP address to the corresponding domain name. The following seven domains are called *generic* domains with three-character labels, one for each of these special type of organization. The classification of the generic domains are given in Table 0.1. The remaining domains are two-character labeled *country* domains, one for each country, e.g., `ca` for Canada and `us` for the United States of America. The domain name of a node is the list of labels written as a text string, starting at the node and ending at the root node. Examples of domain names are `photon.poly.edu` and `mta.nyc.ny.us`, as shown in Fig. 0.8. In addition to the domain names shown in Fig. 0.8, seven new

Table 0.1. *Classification of the generic domains*

domain	Description
com	Commercial organizations
edu	Educational institutions
gov	other US government institutions
int	International organizations
mil	U.S. military groups
net	Major network support centers
org	Other organizations

top-level domains, .aero, .biz, .coop, .info, .museum, .name, and .pro, were added to the Internet's domain name system by ICANN in 2000.

Since the TCP/IP programs only recognize numbers, the domain name system (DNS) is used to resolve, i.e., translate, a domain name to the corresponding IP address. Then the resolved IP address, rather than the domain name, is used in the TCP/IP kernel. DNS is a client/server type of service. Since the entire database of domain names and IP addresses is too large for any single server, it is implemented as distributed databases maintained by a large number of DNS servers (usually host computers running the DNS server program). Thus each DNS server only maintains a portion of the domain name database shown in Fig. 0.8. A host can query the DNS servers for the IP address associated with a domain name, or for the domain name associated with an IP address. If the DNS server being queried does not have the target entry in its database, it may contact other DNS servers for assistance. Or, it may returns a list of other DNS servers that may contain the information. Thus the client can query these servers *iteratively*.

It is inefficient to perform name resolution for the same domain name every time its IP address is requested. Instead, DNS servers and clients use *name caching* to reduce the number of such queries. A DNS server or client maintains a cache for the names and corresponding IP addresses which have been recently resolved. If the requested domain name is in the cache, then there is no need to send a DNS query to resolve it. In addition, each cached entry is associated with a Time-to-Live timer. The value of this timer, which is usually set to the number of seconds in two days when the entry is first cached, is determined by the server that returns the DNS reply. The entry will be removed from the cache when the timer expires.