CHAPTER 1

The Internet

The Internet grew from a small experiment in the late 1960s to a network that connects a billion users and has become society’s main communication system. This phenomenal growth is rooted in the architecture of the Internet that makes it scalable, flexible, and extensible, and provides remarkable economies of scale. In this chapter we explain how the Internet works.

1.1 BASIC OPERATIONS

The Internet delivers information by first arranging it into packets. This section explains how packets get to their destination, how the network corrects errors and controls congestion.

1.1.1 HOSTS, ROUTERS, LINKS

The Internet consists of hosts and routers attached to each other with links. The hosts are sources or sinks of information. The name ‘hosts’ indicates that these devices host the applications that generate or use the information they exchange over the network. Hosts include computers, printers, servers, web cams, etc. The routers receive information and forward it to the next router or host. A link transports bits between two routers or hosts. Links are either optical, wired (including cable), or wireless. Some links are more complex and involve switches, as we study later. Figure 1.1 shows a few hosts and routers attached with links. The clouds represent other sets of routers and links.

1.1.2 PACKET SWITCHING

The original motivation for the Internet was to build a network that would be robust against attacks on some of its parts. The initial idea was that, should part of the network get disabled, routers would reroute information automatically along alternate paths. This flexible routing is based on packet switching. Using packet switching, the network transports bits grouped in packets. A packet is a string of bits arranged according to a specified format. An Internet packet contains its source and destination addresses. Figure 1.1 shows a packet with its source address A and destination address B. Switching refers to the selection of the set of links that a packet follows from its source to its destination. Packet switching means that the routers make this selection individually.
CHAPTER 1. THE INTERNET

Figure 1.1: Hosts, routers, and links. The hosts have a distinct location-based 32-bit IP address. The packet contains the source and destination addresses and an error detection field (CRC). The routers maintain routing tables that specify the output for the longest prefix match of the destination address.

for each packet. In contrast, the telephone network uses circuit switching where it selects the set of links only once for a complete telephone conversation.

1.1.3 ADDRESSING
Every computer or other host attached to the Internet has a unique address specified by a 32-bit string called its IP address, for Internet Protocol Address. The addresses are conventionally written in the form \(a.b.c.d\) where \(a, b, c, d\) are the decimal value of the four bytes. For instance, 169.229.60.32 corresponds to the four bytes 10101001.11100101.00111100.00100000.

1.1.4 ROUTING
Each router determines the next hop for the packet from the destination address. The packets essentially follow the shortest path to their destination. The routers regularly compute these shortest paths and record them as routing tables. A routing table specifies the next hop for each destination address, as sketched in Figure 1.2.
To simplify the routing tables, the network administrators assign IP addresses to hosts based on their location. For instance, router $R_1$ in Figure 1.1 sends all the packets with a destination address whose first byte has decimal value 18 to router $R_2$ and all the packets with a destination address whose first byte has decimal value 64 to router $R_3$. Instead of having one entry for every possible destination address, a router has one entry for a set of addresses with a common initial bit string, or prefix. If one could assign addresses so that all the destinations that share the same initial five bits were reachable from the same port of a 32-port router, then the routing table of the router would need only 32 entries of 5 bits: each entry would specify the initial five bits that correspond to each port. In practice, the assignment of addresses is not perfectly regular, but it nevertheless reduces considerably the size of routing tables. This arrangement is quite similar to the organization of telephone numbers into [country code, area code, zone, number]. For instance, the number 1 510 642 1529 corresponds to a telephone set in the US (1), in Berkeley (510), the zone of the Berkeley campus (642).

The general approach to exploit location-based addressing is to find the longest common initial string of bits (called prefix) in the addresses that are reached through the same next hop. This scheme, called Classless Interdomain Routing (or CIDR), enables to arrange the addresses into subgroups identified by prefixes in a flexible way. The main difference with the telephone numbering scheme is that, in CIDR, the length of the prefix is not determined, thus providing more flexibility.

As an illustration of longest prefix match routing, consider how router $R_2$ in Figure 1.1 selects where to send packets. A destination address that starts with the bits 000010101 matches the first 9 bits of the prefix $18.128 = 00010010'10000000$ of
CHAPTER 1. THE INTERNET

output link L2 but only the first 8 bits of the prefix 18.64 = 00010010'01000000 of output link L1. Consequently, a packet with destination address 18.128.33.11 leaves R2 via link L2. Similarly, a packet with destination address 18.7.25.81 leaves R2 via link L1. Summarizing, the router finds the prefix in its routing table that has the longest match with the initial bits of the destination address of a packet. That prefix determines the output port of the packet.

1.1.5 ERROR DETECTION

A node sends the bits of a packet to the next node by first converting them into electrical or optical signals. The receiving node converts the signals back into bits. This process is subject to errors caused by random fluctuations in the signals. Thus, it occasionally happens that some bits in a packet get corrupted, which corresponds to a transmission error.

A simple scheme to detect errors is for the source to add one bit, called parity bit, to the packet so that the number of ones is even. For instance, if the packet is 00100101, the sending node adds a parity bit equal to 1 so that the packet becomes 001001011 and has an even number of ones. If the receiver gets a packet with an odd number of ones, say 001101011, it knows that a transmission error occurred. This simple parity bit scheme cannot detect if two or any even number of bits were modified during the transmission. This is why the Internet uses a more sophisticated error detection code, the CRC, for cyclic redundancy code. When using the CRC, the sending node calculates the CRC bits (typically 32 bits) from the previous bits in the packet. The receiving node performs the same calculation and compares the result with the error detection bits in the packet; if they differ, the receiving node knows that some error occurred and it discards the corrupted packet.

1.1.6 RETRANSMISSION OF ERRONEOUS PACKETS

In addition to dropping packets corrupted by transmission errors, a router along the path may discard arriving packets when it runs out of memory to temporarily store them before forwarding. This event occurs when packets momentarily arrive at a router faster than it can forward them to their next hop. Such packet losses are said to be due to congestion, as opposed to transmission errors.

To implement a reliable delivery, the source and destination use a mechanism that guarantees that the source retransmits the packets that do not reach the destination without errors. The basic idea of this mechanism is that the destination acknowledges all the correct packets it receives and the source retransmits packets that the destination did not acknowledge within a specific amount of time. Say that the source sends the packets 1, 2, 3, 4 and that the destination does not get packet 2. After a while, the source
notices that the destination did not acknowledge packet 2 and retransmits a copy of that packet. We discuss the specific implementation of this mechanism in Internet in the chapter on the transport protocol. Note that the source and destination hosts arrange for retransmissions, not the routers.

1.1.7 CONGESTION CONTROL
Imagine that many hosts send packets that go through a common link in the network. If the hosts send packets too fast, the link cannot handle them all and the router with that outgoing link must discard some packets. To prevent an excessive number of discarded packets, the hosts slow down when they miss acknowledgments. That is, when a host has to retransmit a packet whose acknowledgment failed to arrive, it assumes that a congestion loss occurred and slows down the rate at which it sends packets.

Eventually congestion subsides and losses stop. As long as the hosts get their acknowledgments in a timely manner, they slowly increase their packet transmission rate. This scheme, called congestion control, automatically adjusts the transmission of packets so that the network links are fully utilized while limiting the congestion losses.

1.1.8 FLOW CONTROL
If a fast device sends packets very rapidly to a slow device, the latter may be overwhelmed. To prevent this phenomenon, the receiving device indicates, in each acknowledgment it sends back to the source, the amount of free buffer space it has to receive additional bits. The source stops transmitting when this available space is not larger than the number of bits the source has already sent and the receiver has not yet acknowledged.

The source combines the flow control scheme with the congestion control scheme discussed earlier. Note that flow control prevents overflowing the destination buffer whereas congestion control prevents overflowing router buffers.

1.2 DNS, HTTP & WWW

1.2.1 DNS
The hosts attached to the Internet have a name in addition to an IP address. The names are easier to remember (e.g., google.com). To send packets to a host, the source needs to know the IP address of that host. The Internet has an automatic directory service called the Domain Name Service, or DNS, that translates the name into an IP address. DNS is a distributed directory service. The Internet is decomposed into zones and a separate DNS server maintains the addresses of the hosts in each zone. For instance,
CHAPTER 1. THE INTERNET

the department of EECS at Berkeley maintains the directory server for the hosts in the eecs.berkeley.edu zone of the network. The DNS server for that zone answers requests for the IP address of hosts in that zone. Consequently, if one adds a host on the network of our department, one needs to update only that DNS server.

1.2.2 HTTP & WWW

The World Wide Web is arranged as a collection of hyperlinked resources such as web pages, video streams, and music files. The resources are identified by a Uniform Resource Locator or URL that specifies a computer and a file in that computer together with the protocol that should deliver the file.

For instance, the URL http://www.eecs.berkeley.edu/~wlr.html specifies a home page in a computer with name www.eecs.berkeley.edu and the protocol HTTP.

HTTP, the Hyper Text Transfer Protocol, specifies the rules for getting the file from a server to a client. Essentially, the protocol sets up a connection between the server to the client, then requests the specific file, and finally closes the connection when the transfer is complete.

1.3 SUMMARY

- The Internet consists of hosts that send and/or receive information, routers, and links.

- Each host has a 32-bit IP address and a name. DNS is a distributed directory service that translates the name into an IP address.

- The hosts arrange the information into packets that are groups of bits with a specified format. A packet includes its source and destination addresses and error detection bits.

- The routers calculate the shortest paths to destinations and store them in routing tables. The IP addresses are based on the location of the hosts to reduce the size of routing tables using longest prefix match.

- A source adjusts its transmissions to avoid overflowing the destination buffer (flow control) and the buffers of the routers (congestion control).

- Hosts remedy transmission and congestion losses by using acknowledgments, timeouts, and retransmissions.
1.4 PROBLEMS

P1.1 How many hosts can one have on the Internet if each one needs a distinct IP address?

P1.2 If the addresses were allocated arbitrarily, how many entries should a routing table have?

P1.3 Imagine that all routers have 16 ports. In the best allocation of addresses, what is the size of the routing table required in each router?

P1.4 Assume that a host $A$ in Berkeley sends packets with a bit rate of 100Mbps to a host $B$ in Boston. Assume also that it takes 130ms for the first acknowledgment to come back after $A$ sends the first packet. Say that $A$ sends one packet of 1KByte and then waits for an acknowledgment before sending the next packet, and so on. What is the long-term average bit rate of the connection? Assume now that $A$ sends $N$ packets before it waits for the first acknowledgment. Express the long-term average bit rate of the connection as a function of $N$. [Note: 1Mbps = $10^6$ bits per second; 1ms = 1 millisecond = $10^{-3}$ second.]

P1.5 Assume that a host $A$ in Berkeley sends 1-KByte packets with a bit rate of 100Mbps to a host $B$ in Boston. However, $B$ reads the bits only at 1 Mbps. Assume also that the device in Boston uses a buffer that can store 10 packets. Explore the flow control mechanism and provide a time line of the transmissions.

1.5 REFERENCES

Packet switching was independently invented in the early 1960s by Paul Baran (5), Donald Davies, and Leonard Kleinrock who observed, in his MIT Thesis (25), that packet-switched networks can be analyzed using queuing theory. Bob Kahn and Vint Cerf invented the basic structure of TCP/IP in 1973 (26). The congestion control of TCP was corrected by Van Jacobson in 1988 (22) based on the analysis by Chiu and Jain (11) and (12). Paul Mockapetris invented DNS in 1983. CIDR is described in (67). Tim Berners-Lee invented the WWW in 1989.