Lecture 2-5: Computer Network Overview

Yuan Xue

Suppose you want to build a computer network. What technologies would serve as the underlying building blocks, what kind of software architecture would you design to integrate these building blocks into an effective communication service. Answering these questions is the overall goal of this lecture.

I. DIRECT LINK NETWORKS

![Diagram of Direct Link Networks]

Fig. 1. Direct Link Network.

Starting from the simplest, we study the network in which all the hosts are directly connected by some physical medium, as shown in Fig. 1, either in a point-to-point manner or in multiple-access manner. There are a list of problems that must be addressed before the hosts can successfully communicate with each other.

- **Encoding.** The first step to establish communication between two hosts is to turn binary data into the signals that the links are able to carry, and then to transform the signal back into the corresponding binary data at the receiving node. Let us ignore the details of modulation, and assume that we are working with two discrete signals: high and low. As shown in Fig. 2, different encoding mechanisms can encode bits into signals.

![Encoding Schemes]

Fig. 2. Encoding Schemes.

- **Framing.** With encoding, we know how to transmit a sequence of bits over a point-to-point link. Framing then delineates the sequence of bits transmitted over the link into complete messages that
can be delivered. There are several approaches to address the framing problem, including the sentinel approach (e.g., PPP) and the byte-counting approach for byte-oriented protocols; and high-level data link control (HDLC) for bit-oriented protocols.

- **Error detection.** Bit errors can be introduced into frames due to electrical interference or thermal noise. Error detection detects transmission errors. Several commonly techniques for error detection include cyclic redundancy check (CRC) (e.g., Fig. 3 (a)), two-dimensional parity (e.g., Fig. 3 (b)), and Internet checksum. Some error codes can only detect the errors, some codes are strong enough to correct errors.

![Fig. 3. Error Detection Schemes.](image)

- **Reliable delivery.** When errors are detected, and cannot be corrected, the corrupted frames must be discarded. Reliability delivery is sometimes implemented in point-to-point link network to recover from these lost frames.

- **Media access control.** When the link is shared by multiple hosts, their accesses to the link need mediation. Take Ethernet as an example. CSMA/CD (Carrier Sense Multiple Access / Collision Detection) is used to provide media access control. Essentially, participating hosts monitor the traffic on the link. If no transmission is taking place at the time, the particular host can transmit. If two hosts attempt to transmit simultaneously, this causes a collision, which is detected by all participating hosts. After a random time interval, the hosts that collided attempt to transmit again. If another collision occurs, the time intervals from which the random waiting time is selected are increased step by step. This is known as exponential back off.

Nearly all the networking functionalities described above are implemented in the network adaptor (as shown in Fig. 4): encoding, framing, error detection, and the media access control. In Ethernet, each adaptor has a unique Ethernet address, which is also the MAC address of the corresponding host. Each frame transmitted on an Ethernet is received by every adaptor connected to that Ethernet. Each adaptor recognizes those frames addressed to its own address, and passes only those frames to the host. An adaptor can also be programmed to run in *promiscuous* mode, in which case it delivers all received frames to the host.
II. INTERNETWORKING

So far, we have seen how to build a simple network using direct links. Direct link network can only connect a limited number of hosts and cover a small geographical area. To establish a network that has the potential to grow to global scale, we need to connect the direct-link networks. There are two important problems that must be addressed when connecting networks: heterogeneity and scale. As shown in Fig. 5, internetwork connects networks (e.g., Network 1 – Network 4) with different technologies with routers (e.g., R1, R2, R3). This presents the problem of heterogeneity. To understand the problem of scale, let’s consider the growth of the Internet, which has roughly doubled in size each year for 20 years. Additional functions are required to build scalable, heterogenous internetworks.

A. Addressing

Addressing is the task of providing suitable identifiers for all these hosts in internetworks. To address the problem of scalability, hierarchical address scheme should be used.

B. Fragmentation and Reassembly

One of the problems to provide a uniform host-to-host service over a heterogeneous collection of networks is that each network technology has its own definition of packet size. We need a means by which packets can be fragmented and reassembled when they are too big to cover a given network. As shown in Fig. 6, each fragment is a self-contained datagram that is transmitted, independent of the other fragments; Each datagram is re-encapsulated for each physical network over which it travels.
C. Routing and Forwarding

Now let’s look at the basic mechanism by which packets are delivered among different networks: *forwarding* is the process of a router to take a packet from an input and send it out on the appropriate output based on its forwarding table; *routing* is the process of building up the forwarding table.

**Fig. 7.** Forwarding table of R2 in the network shown in Fig. 5.

Forwarding datagrams can be handled in the following way. A datagram that carries the address of the destination host, is sent from a source host, possibly passing through several routers along the way, and arrives at its destination. Any node (a host or a router) that handles the datagram, examine the destination address with the network part of its addresses (routers may have more than one addresses, as they have multiple network interfaces that are connected to two or more networks). If a match occurs, then that means the destination lies on the same physical network, and the packet can be directly delivered over that network. If the node is not connected to the same physical network as the destination node, then it needs to send the datagram to a router based on its forwarding table, which is know as the *next hop* router. The forwarding table of Fig. 5 is shown in Fig. 7.

**Fig. 8.** Network as a graph.

Now the question is how do routers acquire the information in their forwarding table? This is addressed by the routing process. Routing is essentially a problem of graph theory. If we view a network as a graph,
as in Fig. 8, the nodes of the graph may be either hosts, routers, or networks; the edges of the graph correspond to the network links. Each edge has an associated cost, which gives some indication of the desirability of sending traffic over that link. The basic problem of routing is to find the lowest-cost path between any two nodes. In more practical networks, routing is achieved by running routing protocols among the nodes. The protocols need to disseminate network topology information to every node in the network, so that every node will have enough knowledge to find the optimal path to any node in the network and build the forwarding table.

The networking functionalities described above are usually implemented in the operating system kernel on end hosts and in all routers in the Internet.

III. END-TO-END PROTOCOL

We have established an internetwork connecting together a collection of hosts. The next problem is to turn this host-to-host packet delivery service into a communication channel between application processes. This is sometimes called the end-to-end protocol. Two end-to-end protocols of Internet are UDP and TCP.

Since many processes may run on the same hosts, the protocols need to add a level of de-multiplexing. To identify the processes, an abstract locator, often called a port is assigned to each process, as shown in Fig. 9.

Besides the de-multiplexing functionality, which is also provided by UDP, TCP provides a more sophisticated connection-oriented, reliable network service. Connection-oriented network service means that the participants in a TCP session must first build a connection - via the 3-way handshake procedure as shown in Fig. 10. TCP also guarantees the reliable, in-order delivery of a stream of bytes via sequences and acknowledgements. Finally TCP provides flow control/congestion control to determine an appropriate sending rate of the byte stream.

The end-to-end communication protocols are only implemented at end hosts, usually as an OS kernel process.
Fig. 10. 3-way handshake in TCP.

Fig. 11. Network Architecture.

IV. LAYERED ARCHITECTURE

To organize all the above networking protocols in a systematic way, an architecture needs to be defined. Open Systems Interconnection (OSI) (Fig. 11 (a)) architecture is one of the first defined architectures for connecting computers. The Internet architecture, which is also called the TCP/IP architecture (Fig. 11 (b)), is the most widely adopted architecture.