Preliminary

- Course Syllabus
- Project 1

Introduction

Networking Today:

- Powerful computers are cheap.
- Networks are everywhere. Indeed, the network *is* the computer!
- blurred line between data processing (computers) and data communications (transmission and switching) equipment.
- data, voice and video communications have fewer differences—quality of service considerations.
- blurred lines between single-processors, multi-processors, ad hoc, sensor, local networks, metropolitan networks and long-haul networks.

Definition

Definition of a computer network: an *interconnected* collection of *autonomous* computers.

interconnected—capable of exchanging messages.

autonomous—do not control one another.

There are two aspects to computer networks:

Hardware "physically" connects machines to one another (allows signals to be sent).

Protocols specify the services the network provides. Protocols make the hardware usable by programmers and applications software.

Which is more important? They are both important. However, software is more important because it is the software that programmers and end users see and use. In addition, by careful design, the hardware can be hidden from the user, making the hardware even less important. Of course, software will have a difficult time overcoming hardware deficiencies.

Analogy: what's more important, a machine's hardware architecture or its operating system? A machines specific instruction set or its compilers?

Different scales and different kinds of network traffic.

How do networks fit into computer systems?

Ways in which networks are used by computers.

Distributed Operating System: User sees a single large *virtual computer* system. The system hides the details of the network and the existence of multiple machines from the user, providing the abstraction of a single virtual computer system. The presence of the network is integrated into the system, and services can be accessed *transparently*, that is, without the user knowing that the service does not reside on the local machine. There are few true distributed operating systems in existence, and none are products.

Autonomous System: Standalone machines run fine *even* when disconnected from network. Users invoke special commands to use the network. That is, explicit commands are needed to access files on remote machines, for example. The Unix/Linux r*/s* commands (*rsh/ssh, rcp/scp,* etc.) are examples of such utilities.

Network File System: Mostly autonomous machines share file systems located on remote file servers. The network is not totally integrated into the system (all processes run on the local machine, for instance), but remote files are accessed transparently. These are the most common systems today. Sun's NFS protocol is the de facto standard.

Internet Information Systems: Public access systems exemplified by the World Wide Web.

Distributed Applications/Services: Collection of server machines supporting a common service. Electronic mail is a prime example.

Network Video/Audio: Video-on-demand application or a teleconferencing application. Different requirements for each application—loose vs. tight timing requirements.

Sensor Networks: Low-power devices communicating collected data.

Each of these works with the client-server paradigm (see Fig 1-2). You will gain experience with this approach in this class.

Why Networks?

They make a more cost-effective use of hardware and software, as well as create new benefits.

Resource Sharing: One printer (or other special hardware) can be shared by many machines instead of requiring each machine have its own printer. Other expensive resources include plotters, color laser printers, terminals, storage devices, special machine architectures, etc.

Information Sharing: Electronic mail, or e-mail, has become ubiquitous for exchanging ideas quickly. Expect information to be there on the Web.

Improve Reliability: Eliminate single points of failure through replication. Note however, the following (alternate) definition of a distributed system: A system in which the failure of a machine I am not using prevents my machine from operating correctly.

Reduced Cost: One obtains more "bang for the buck" by buying many PCs and workstations than a single mainframe machine—Grid computing.

Effect on Society

"Information Superhighway"

Networks are changing not only computer science but all of society. Typical uses of networks include:

- Through electronic mail, or e-mail, it is common to hold "electronic conversations" with someone thousands of miles away. Even more immediate with *instant messaging*.
- bulletin boards
- World Wide Web
- Storing files on a file server, giving users access to their files regardless of what machine they actually use.
- Instant sharing of information, including public domain software, "important" discoveries—Internet "flash points."
- Electronic commerce. Business online.
- Anonymous FTP sites. Users place files (software, data, etc.) in a special account where anyone with network access can copy the files using FTP to their own machines.

Has become indispensable part of one's life.

Network Structure

A *host* or *end-system* refers to a computer that a user interacts with to do their work. Although a host is attached to a network, it is not usually part of the network itself.

The *communications subnet* refers to the everything between hosts. The *subnet* is responsible for transporting data from one host to another. Look at Figure 1-9.

Although many subnet designs exist, they generally fall into two broad classes (or a combination of the two):

Broadcast: Many (3–1000) machines connect to a common "wire". When one machine speaks, all other machines hear what is said. Each packet contains an address that specifies who the intended recipient is. Broadcast (also called *multi-point*) is similar to a person standing up in a room and shouting out an announcement. Everyone hears it, but only those interested in the announcement act on (or even listen to it!). Fig 1-7. Now more typically have switched broadcast.

Point-To-Point: Two machines, one at each end of a "wire".

Of course, a single pt-to-pt link isn't all the useful. Usually, a machine will connect to several links, resulting in a graph topology. Each machine (called a *packet-switch*) "switches" packets from one line to another. Fig 1-6.

Broadcast networks have the advantage that they efficiently support *multicasting* sending a single packet to every machine on the network. This contrasts with non-broadcast networks, which must send the same packet over and over again to each machine on the network to achieve the same outcome.

Note that *broadcasting* is a special case of multicasting, in which *all* machines receive the packet.

Unicasting refers to the normal case where packet is intended for a single recipient.

Have done some research work with multicasting and load sharing.

Types of Networks

See Stallings Fig 1-5.

Primary types of networks:

- Local area networks (LANs) cover small geographic regions (e.g., building(s) or campus). Data rates are high (10-10 Gbps and up), cost low (thousands of dollars). Typically uses broadcast technology.
- Metropolitan Area Networks (MANs) cover medium-size geographic regions (e.g., entire cities). Local cable system as an example.

  A standard exists IEEE 802.6—Distributed Queue Dual Bus (DQDB). Uses two broadcast buses, one for each direction.
- Wide area network (WANs) cover larger geographic distance (e.g. entire countries). Data rates typically much lower (56 kbps - 1.5 Mbps (T1), bundle T1 links to get higher rates), because cost is much higher (tens or hundreds of thousands of dollars per year).
- Wireless/Mobil. Notebook computers and portable digital assistants (PDAs). Provide a way to connect back to a base computing environment. Many applications. Wireless connections allow mobile computing, although mobile computing and wireless networks need not be used together (e.g. a wireless network may be used to connect a room of workstations).

  Look at Figure 1-5.

  Options:

  - Bluetooth—short range wireless network without cables—digital cameras, headsets, and other devices.
  - IEEE 802.11–wireless LAN (WiFi)
- Home Networks. Network home devices. Ease of use and cost become much more important.
- Internetworking. The connection of different types of networks. Internet is a specific worldwide connection of networks. internet is a generic term.

Layering

Building networks is a complex problem. Where do we start?

Use divide-and-conquer: Break the problem into smaller problems (subroutines or objects), solve the individual problems, and combine them into a final solution.

Networking extends this concept into *layering*. Starting with the hardware, we build successive layers on top of the lower layer. The idea is to build increasingly sophisticated services, using only the services provided by the layer immediately underneath.

Conceptually, layer N communicates with its remote *peer* (counterpart layer) on a remote machine. In reality, only the lowest layer physically transmits data. See Fig. 1-13.

Each layer defines a *protocol* that describes the *messages* exchanged with its peer on the remote machine.

A crucial point in achieving good layering is *abstraction* — providing a well-defined service. When using the services of layer N, we are not concerned with *how* the service is actually implemented. We only need to know how to *invoke* it. Thus, the service better be one that is easy to use and allows the layer above it to build a better service.

Each layer provides an *interface* that specifies how its services are accessed by the layers above (and below) it. The interface should never change — changing it affects the layer above it. The actual implementation of the service, however, can change.

How easily can we change the underlying network from point-to-point to broadcast? If the layers are designed properly, the change will be transparent to the upper layers.

Network Architecture

The layers and protocols form what is called a *network architecture*. Look at Fig. 1-15. When discussing layering in network protocols, two concepts are fundamental *messages* and *encapsulation*. Each layer deals with *messages*. Message properties:

1. Are (generally) limited to a maximum size. For example, Ethernet frames carry only 1500 bytes of data. Maximum sizes vary from a few hundred to thousands of bytes.
2. Contain *control* or *header* information used to synchronize with the remote peer. The header contains "instructions" that tell the remote peer what to do with the message.
3. Include a *data* portion that contains arbitrary data. The data portion contains whatever information users want to share, and is of no interest to that particular protocol layer.

When layer *N* accepts data from layer *N+1* (above it), it *encapsulates* the entire layer *N+1* message in the data portion of the layer *N* packet. It should never look inside the data portion of the message!

When the remote peer receives a message, it strips of the header information and passes only the data to the next higher layer.

Also may need to *fragment* packets.

OSI Reference Model

As a first step in standardization, the International Standards Organization (ISO) developed a seven-layer model known as the ISO Open Systems Interconnection (OSI) reference model.

Treat as a *framework* for organization of functions.

1. The lowest layer, the *physical layer*, is concerned with transmitting raw bits over a communication channel. It is concerned with insuring that when one side sends a "1" bit, the other side receives a "1" bit.

   The physical layer is usually the focus of an electrical engineer and deals with such questions as:

   - How many volts represents a "1", how many a "0".
   - How long does a bit time last?
   - How many pins does the connector have?
   - How many wires does the transmission media have?
   - Are pulses electrical or optical?

2. The *data link layer* transforms the raw transmission facility of the physical layer into an error free channel. It deals with communication between two machines sharing a common physical channel. It:

   - Divides the bit stream of the physical layer into *frames*, messages containing data and control information.
   - Handles lost, damaged, and duplicate frames.
   - Handles slowing down a fast transmitter (known as *flow-control).*

3. The *network layer* controls operation of the subnet (communication between hosts). It:

   - Routes packets from source to destination host (but may not guarantee that all packets are delivered).
   - Worries about congestion — hosts sending data into the network faster than the network can handle.
   - Deals with addressing: how do we specify *which* machine data should be delivered to?

4. The *transport layer* makes sure data gets delivered to a specific process on a specific machine. It:
    - Is an *end-to-end* protocol because it deals with the ultimate endpoints of communication, the sending and receiving application. That is, the two end points of communication.
    - Deals with retransmitting data if the network layer fails to deliver it.
    - Deals with suppressing duplicates. If it retransmits messages, it may introduce a duplicate, if the retransmission was unnecessary.
    - Also deals with addressing. Which process on a particular machine?
5. The *session layer* allows users to establish sessions between them. It:
    - Provides a cleaner interface to the transport layer. For example, one that is not operating system specific (sockets in BSD Unix, or TLI in System V streams).
    - Provides synchronization, such as recovering from transport layer failure.
6. The *presentation layer* performs services that are requested often enough to warrant development of a general solution. For instance:
    - Encoding data in a standard format, so that (for example) ASCII systems can communicate with EBCDIC systems.
    - Compressing data to reduce communication costs.
    - Encrypting data for privacy.
7. The *application layer* refers to the user programs themselves. Look at Figure 1-20.

TCP/IP Reference Model

Came out of work on ARPANET (predecessor to Internet). Not so much a model, but a base set of protocols. Uses a *connectionless* network layer. Look at Figure 1-22. Will explore all levels in more detail.

Why TCP/IP Succeeded

- Protocols came first, where in OSI reference model, the model came first. Model provides a nice basis to talk about networks. TCP/IP provides better protocols for using them.

  OSI got "crushed" because TCP/IP was already a working protocol. See Fig. 1-23. Dave Clark offers an observation dubbed "apocalypse of the two elephants":

  1. When a subject is first discovered, a burst of activity occurs in terms of discussions, papers, and meetings.
  2. After a while, the discussion subsides and corporations discover the subject, investing billions of dollars in the subject.
  3. When should standards be written? It is essential that standards be written in the "trough" between the two "elephants". If the standard is written too soon, it may be bad or technically inferior. If written too late, companies producing products will ignore the standard.

- Growth of the Internet, particularly the Web has cemented the victory of TCP/IP over OSI.

Hybrid Model

Tanenbaum offers critiques of each model. Settles on a "model" of leaving out session and presentation layers and organizing material around other layers.

Service Styles

There are two primary styles of network communication:

Connection oriented: User first *opens* a connection, then sends and receives data on the connection, and finally, *terminates* the connection. OSI model, TCP protocol. Primary example is the telephone system. Connection oriented service typically delivers *all* messages, and delivers them in the order in which they were sent.

Connectionless: Each message is a standalone entity and is routed independently through the subnet. Each message contains sufficient information to deliver it to its destination (e.g., source and destination addresses). Connectionless service is analogous to the postal service, where each item (letter, package, etc.) is delivered independent of all others. Each letter (packet) contains its own destination and return address.

Connectionless service is often called *unreliable datagram* (or just *datagram*) service because there is no guarantee that messages will be delivered correctly. Indeed, the main point of the "connection" is to insure that the network is able to guarantee that packets will be delivered in a known order. The term unreliable refers to delivery not being acknowledged, and hence, there is no way to know for sure that the data has been delivered. IP protocol is an example.

.

Services vs. Protocols

When thinking about network protocols, it is important to distinguish between *services* and *protocols*. A *service* is a high-level, abstract description of the functionality provided, while a *protocol* is the specific set of *rules* that specify the meaning of messages exchanged by peer entities. A protocol implements a service.

Network Standardization

A *standard* is a formal specification that specifies in great detail an entity's physical attributes as well as its response to specific stimuli.

At the hardware level, standards define such things as the number of pins on a connector, the spacing between pins, etc.

For software, standards define interfaces to protocol layers and message formats.

In networking standards are needed because:

- They allow dissimilar computers (e.g., architectures, vendor operating systems, etc.) to communicate with one another.
- They allow computers in different countries to communicate.
- They promote economy of scale. It's cheaper (per unit) to manufacture 10 million Ethernet LAN interfaces, then 1 million LAN interfaces for each of 10 different products.

Ideally, every computer manufacturer should adopt the same set of standards. Why doesn't this happen? Standardization increases competition, which often reduces profits. For example,

- Vendors can lock customers into their product line by using standards that only their products support. Vendors rarely encourage customers to buy another vendor's products.
- Adopting another vendor's standards may put one company at the other's mercy. In particular, the other vendor already has the advantage of having products that adhere to the standard. In addition, suppose the other company then changed the standard? They could then update their product line without letting their competitors in on the change.

Types of Standards

There are two types of standards:

1. *De facto*: those that just happen. They become adopted through market forces, e.g., IBM PC/Windows, TCP/IP, etc.
2. *De jure*: those adopted by a standards group, e.g. OSI reference model, Ada for DoD.

Standards Makers

- ISO, International Standards Organization
- IEEE, American Institution of Electrical and Electronic Engineers
- ITU-T, International Telecommunications Uniton—Telecommunications Sector. Formerly CCITT, International Telegraph and Telephone Consultative Committee.

ISO/IEEE produce standards for computer manufacturers. ITU-T defines standards for connecting equipment to the different types of national and international public networks. Increasing overlap between computer and telecommunications industries.

Network Performance/Quality of Service

Network performance and quality of service (QOS) are increasingly important areas of network work as the number of Internet users grows with the WWW and applications such as real-time video have more stringent performance demands (vs. electronic mail).

Terms:

- Bandwidth: On an analog channel, the frequency range of the channel measured in Hertz (Hz). A voice grade channel has a bandwidth of 3000Hz.

  For a digital link, we refer to the number of bits/sec. that can be transmitted. For example, a 10Mbps or 100Mpbs Ethernet.

  Also talk about *available* bandwidth as the capacity that is available for transmission.

- Throughput: Measured performance of a system. What is the *effective* transmission rate through a channel. Can also call it *goodput*.

- Bandwidth requirements: of an application. What are its demands? Are those demands a constant bit rate (CBR) or a variable bit rate (VBR)?

- Latency: How long for a bit to propagate from one end of a network to the other. Measured in terms of time. It is a function of

  1. distance/speed of light (e.g. at 2.0×10^8 meters/sec in a fiber, it takes 24ms to travel 3000 miles across the continental U.S.),
  2. time to transmit a unit of data (a bit), and
  3. queueing delays in the network.

- Round Trip Time (RTT): Round trip latency for a single bit from source to destination and back. 100ms across country is a reasonable estimate by increasing the 48ms minimum to account for switch delays and lack of a straight line path.

- DelayXBandwidth Product: How many bits can fit into the "pipe." Draw a series of bandwidth/delay pipes showing interconnections. Impacts throughput Want to keep pipe full for efficiency.

- Jitter: variation in latency. Negative impact on real-time communication. Relates to *elastic* vs. *inelastic* applications.

- Packet loss: Can applications tolerate packet loss? Should/Does underlying network/protocol recover from packet loss?

Example Networks

Tanenbaum finishes Intro chapter with examples of various networks.