I. Introduction to Networks

Instructor: Patrick Tague **Date**: 7 January 2008

EE 565: Computer-Communication Networks I Winter Quarter 2008

1 Networks

The term **network** applies to any system of interacting objects (agents, users, etc.). Since we're focusing on computer and communication networks, we'll focus on interactions between **users** or **devices** using an underlying network structure. While there is no universally agreed-upon definition of a computer network, for the purposes of this course, we loosely define a computer network as *a collection of interconnected devices with the primary purposes of exchanging and sharing information and resources.* Put simply, a computer network provides a service to its users by allowing them to exchange messages (data, commands, etc.) or share resources (storage, printers, etc.). The following examples illustrate some of the important types of network architectures.

Many examples of computer networks behave according to the well-known **client-server model**, in which multiple **client** machines or users access one or more *server* machines, as illustrated in Fig. 1. A familiar example of a client-server system is an office with a large number of computers connected to a single printer and a single file server.

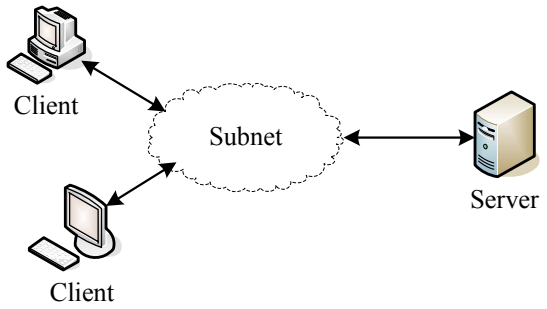

Figure 1: The client-server network model.

A modern alternative to a client-server network is a **peer-to-peer** (P2P) network in which users exchange information amongst themselves without relying on servers, as illustrated in Fig. 2. A familiar example of a P2P application is Napster, where users got information about the location of songs from a server but did not rely on the server to retrieve the songs themselves. Another example of a P2P network is a wireless ad hoc network.

Additional examples of computer and communication networks include the telephone

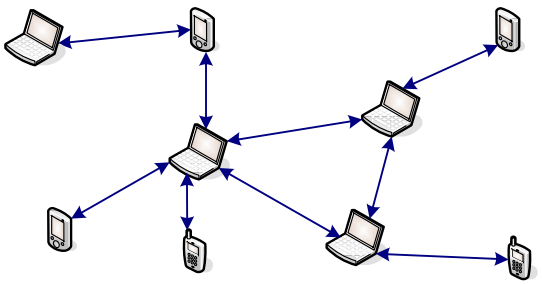

Figure 2: The peer-to-peer (P2P) network model.

system, the Internet (some argue this is actually a network of networks), electronic banking systems, inventory or point-of-sale systems, and personal networks.

To further understand the fundamentals of networking, we first investigate physical network properties in terms of network hardware and then discuss logical network properties in terms of network software.

2 Physical Network Properties

In what follows, we discuss two methods for classification of networks. The first classification is based on message transmission and the second is based on network scale.

2.1 Network Classification by Transmission Type

There are two types of message or packet transmission that are widely used, illustrated in Fig. 3. The first is **point-to-point** or **unicast** transmission, involving a single sender and a single receiver. The second is **broadcast** transmission, in which a single communication medium is shared by all users in the network. The sender must include information in a packet header to indicate which user is the desired recipient of the packet, and other users will simply ignore the transmission. Broadcasting can also be generalized to **multicast** transmission in which each packet is sent to a subset of the users in the network.

2.2 Network Classification by Scale

Networks can also be classified by scale, as illustrated in Fig. 4. The smallest networks, typically on the order of centimeters to 1 meter, are **personal area networks** (PANs). An example of a PAN is a Bluetooth network of personal devices such a mobile phone and

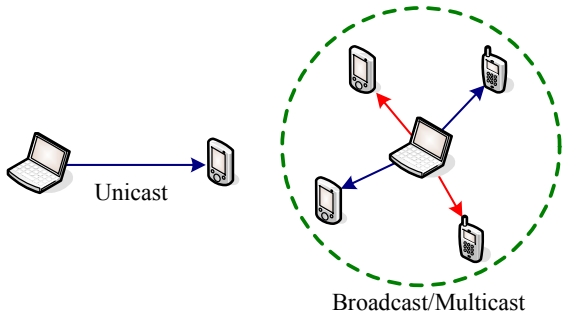

Figure 3: Classification of networks by transmission type.

hands-free device.

Local area networks (LANs), on the order of 10 meters to 1 km, tend to be privately-owned networks in a single room, building, or campus. An example of a LAN is the network of computers and printers in the EE building. LANs tend to be (relatively) small networks that are easy to manage, have high transmission rates (up to 10 Gbps), and usually use broadcast transmission.

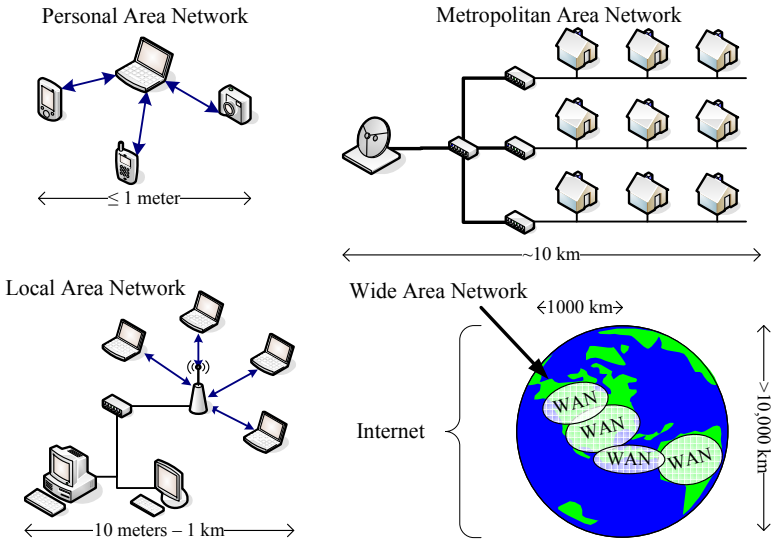

Figure 4: Classification of networks by scale.

Metropolitan area networks (MANs) span entire cities on the order of 10 km. Examples of MANs include the hybrid fiber coax network providing cable television and internet access to people throughout Seattle and networks using the new WiMAX technology.

Wide area networks (WANs) span countries or continents on the order of 100-1000 km. A WAN is typically made up of a **subnet** and a collection of host computers running user processes. The job of the WAN subnet (typically owned by a telephone company or service provider) is to carry messages between host machines. The subnet consists of

transmission lines to carry bits between machines and switching elements, or **routers**, to forward signals on to host machines. The design of WANs is greatly simplified by separating the communication aspects of the transmission lines and routers.

An **internet** is the largest type of network. An example of an internet is *the* Internet, spanning the entire planet Earth (>10000 km). An internet is essentially a collection of LANs connected by a WAN.

2.3 Delay

The scale of the network can have a significant impact on the delay in moving data between users. However, there are many places that delay is introduced in moving data across the network. When a node receives a packet, the packet must be processed in order for the node to determine what to do with the packet, causing a **processing delay**. Once the packet is processed, the packet must be buffered in a queue until the transmission medium is ready and there are no other packets earlier in the queue, causing a **queueing delay**. Since the transmission medium has a finite bandwidth, there is a maximum rate at which bits can be pushed onto the transmission medium (think of pouring water into a funnel), so there is a **transmission delay** equal to the ratio of the packet length L in bits to the link bandwidth or rate R. Similarly, there is a finite limit to the rate at which a transmitted bit can move across the transmission medium, computed as a function of the length of the link and the propagation rate over the transmission medium, introducing a **propagation delay**. The total delay in transmitting data over a link is often measured using the **round-trip time** (RTT), equal to the total delay in transmitting a short packet from a node A to a node B and back to A.

Delay in network protocols is often illustrated using space-time diagrams, such as that in Fig. 5. The horizontal dimension is space, representing the physical distance between nodes in a path. The vertical dimension is time, illustrating the delays experienced in moving data over the given path. Fig. 5 illustrates an example of the processing, queueing, transmission, and propagation delays experienced for each node and link along the path. The RTT can be approximated as twice the total delay experienced between the source and destination nodes in the path.

2.4 Switching

There are three primary techniques for switching in WANs: circuit switching, message switching, and packet switching. Each of the techniques is illustrated using a space-time diagram representing data transmission over a three-hop path.

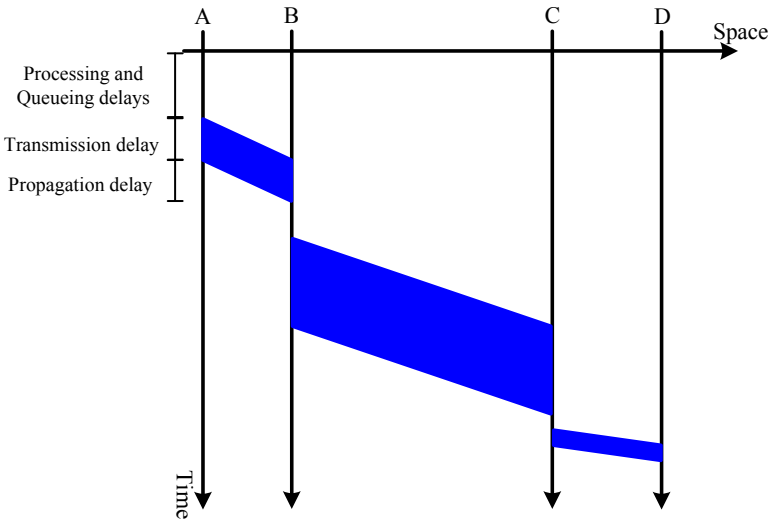

Figure 5: A space-time diagram illustrating processing, queueing, transmission, and propagation delays experienced.

In a network using **circuit switching**, such as the telephone network, an end-to-end connection between hosts is established once and resources are dedicated to the connection. Once the connection, or session, is established, the sender host can freely transmit to the receiver using the dedicated resources. When the transmission is complete, the sender can terminate the connection. Fig. 6 illustrates circuit switching.

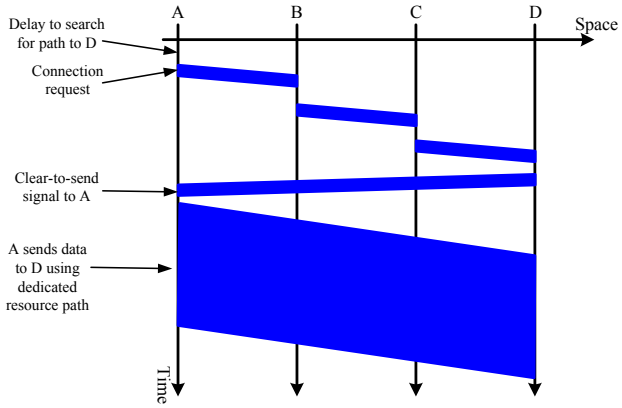

Figure 6: Circuit switching is illustrated in the space-time diagram.

In a network using **message switching**, known as a store-and-forward network, each router forwards the entire message to the next router at once, so there is no physical end-to-end connection between hosts. Fig. 7 illustrates message switching.

The idea of **packet switching** is similar to that of message switching, except the message is broken into small packets. Fig. 6 illustrates packet switching. Advantages of packet

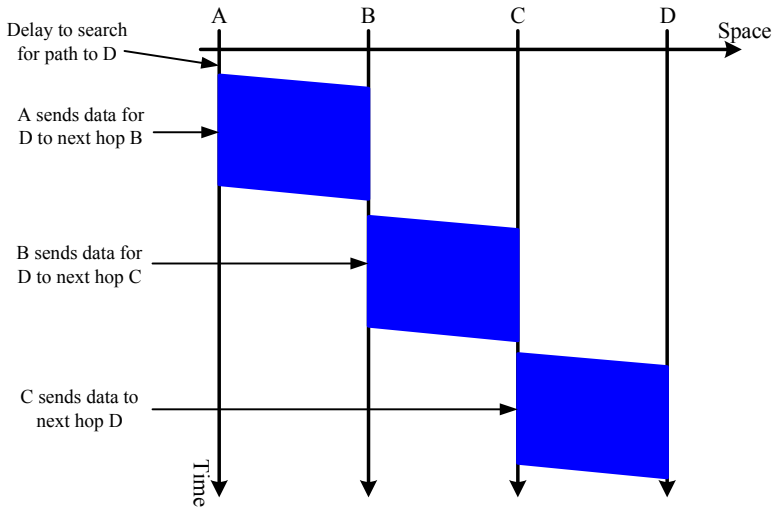

Figure 7: Message switching is illustrated in the space-time diagram.

switching include smaller buffer memories at intermediate routers, the inability for a single user to monopolize the transmission medium, and the ability to forward each packet before the next packet has been received, known as pipelining. Because of these advantages, message switching is almost never used. The use of circuit and packet switching in data networks is thus compared.

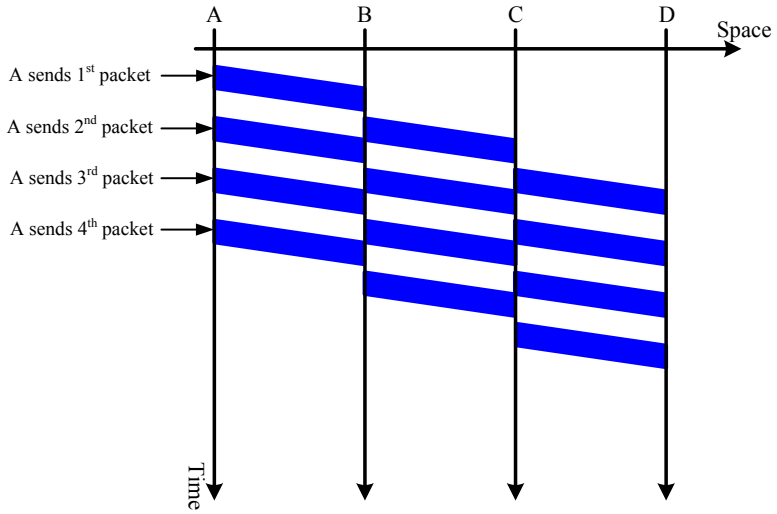

Figure 8: Packet switching is illustrated in the space-time diagram.

- Packet switching requires no advanced setup, so the first packet of a message can be sent as soon as it is available.
- A circuit switched connection guarantees in-order delivery of packets to the receiver.

Since packets in a packet switched network can traverse different paths, in-order delivery is not guaranteed.

- Packet switching is tolerant to faulty routers, as packets can be routed around a failed router. In circuit switching, a failed router causes all corresponding circuits to be terminated.
- Bandwidth is reserved in advance for circuit switching. This is advantageous because congestion cannot occur once the connection is established, while in packet switching, packets may be buffered until bandwidth becomes available. However, this can be disadvantageous. If a circuit is reserved for a user with no traffic, the bandwidth is wasted.
- ...

The above comparison indicates that various trade-offs exist between circuit and packet switching, suggesting that both techniques are useful.

3 Logical Network Properties - Layering

Since the purpose of a network is to provide service to users, the operation of the network is often more important than the hardware and physical network properties. Due to the complexity of networks, with many communication media, hosts, routers, applications, etc., network functionality is organized into independent **layers**, such that the layering is transparent to network users. Example of independent network functions to be organized into layers include finding, or addressing, data recipients in the network, routing data packets from source to destination through the network, ensuring the data received by the destination node is error-free, and ensuring that the network is not overly congested by the data communication. By breaking network functionality into layers, each function can be approached independently, effectively breaking the problem into smaller manageable pieces. The purpose of each layer is to provide a **service** to the higher layers and mask the details of the service implementation, preserving transparency. Each layer communicates through an **interface** with the next highest and lowest layers.

As an example of a layered architecture, consider the postal mail system. Suppose Alice wants to send a letter to Bob. Alice writes Bob's address on the letter and drops it in the mailbox, her only requirement being that the letter eventually arrives in Bob's mailbox. For the most part, Alice doesn't care about the underlying systems to get the letter to Bob. A postal carrier, the next lowest layer, takes the letter from Alice's mailbox and delivers it to the post office, providing the service of mail collection. At the post office, the next layer, mail is sorted and loaded into vehicles for transportation. The lowest layer in this examle is the

physical transportation of Alice's letter from her local post office to Bob's local post office (though this may take many steps). Once the letter reaches Bob's post office, the letter is again sorted and given to the postal carrier, the reverse service as that of Alice's post office. The postal carrier then delivers the message to Bob's mailbox, where it is received by Bob. As seen in this example, the complex task of delivering Alice's letter to Bob is broken down into simpler tasks, thus simplifying the entire system architecture.

One of the most common protocol stack architectures is the **Open Systems Interconnection (OSI) Reference Model** developed by the Internation Standards Organization (ISO). The OSI stack consists of seven layers, as illustrated in Fig. 9. The seven layers are described from the bottom up as follows.

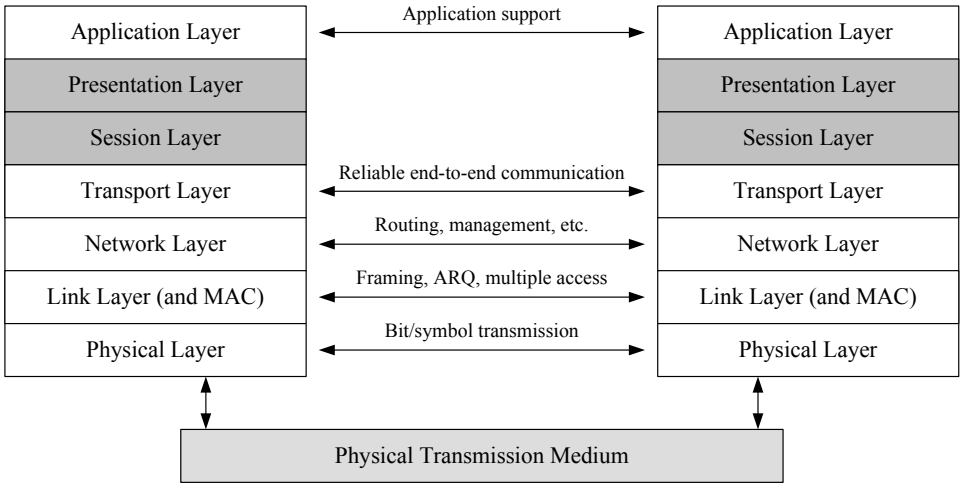

Figure 9: The protocol stack in the OSI Reference Model.

At the bottom of the protocol stack is the **physical layer**, responsible for converting bits (or symbols) to electrical signals (modulation) to transmit over the physical medium as well as the reverse operation (demodulation). The physical layer is responsible for synchronization, physical connection of devices, and hiding the physical medium from the higher layers in the protocol stack.

The **(data) link layer** is responsible for transfering data between neighboring nodes, receiving a sequence of bits over the interface from the physical layer and collects bits into **frames** to send to higher layers. The link layer is also responsible for acknowledging reception of frames from the sender and performing error detection. In broadcast networks, the link layer also contains the **medium access control** (MAC) sublayer which is responsible for controlling access to the shared channel. An example link layer protocol is Ethernet (IEEE 802.3).

The **network layer** is responsible for routing packets from source to destination nodes through the subnet. The network layer is also responsible for handling issues related to

quality of service (delay, etc.) and address and packet translation between internetworked LANs. An example network layer protocol is the Internet Protocol (IP).

The **transport layer** is responsible for host-to-host data transfer. In some cases, the transport layer is responsible for reliable transfer, guaranteeing that packets are delivered to higher layers without error and in the correct order. In other cases, the transport layer provices a best-effort services with no delivery guarantees. Example transport layer protocols are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). Since the transport layer performs end-to-end protocols, this layer is not present in the protocol stacks of routers and intermediate devices.

The **session layer** establishes and terminates sessions between users for the data to be sent and received. The **presentation layer** is responsible for tasks such as data compression, encryption, and standard representation. These two layers are often ignored, and we will do so.

The **application layer** provides support for network applications, using protocols such as HTTP (hyper-text transfer protocol) for web applications, SMTP (simple mail transfer protocol) for email, and FTP (file transfer protocol) for file exchange.

As mentioned, the session and presentation layers are often ignored. This is because there are simpler models that group their functionalities with the application layer. For example, the Internet Protocol Stack, or TCP/IP model, consists only of the other five layers.

For all of the benefits of layering, there are definately some drawbacks. For example, due to transparency of layers, applications may not be able to access information about the physical layer. In many cases, the hidden information may be useful, such as in the case of cross-layer design of wireless networks, where information from other layers can improve performance.