NETWORK OWNERSHIP, SERVICE PARADIGM, AND PERFORMANCE

Network Ownership

- Private networks
- Public networks
- Virtual Private networks

Private Networks

- Network hardware and software are owned by a company or an individual
- Most of the private networks are LANs
- A large corporation may have a private WAN to connect computers at multiple sites; in this case, some of the connections may be leased from public carriers
- Advantages
  - Restricted to the owner's use
  - The network owner has a complete control over the network, e.g., set policies, isolate the network form computers outside the organization
- Disadvantage
  - Installation and operation costs may be expensive

Public Networks

- Network hardware and software are owned by common carriers
- Almost all public networks are WANs
- Does not have a broadcast address
- Offers private communication when two computers exchange messages
- "*Public*" refers to availability of the service, not the data transferred
- Subscription is needed; Anyone (individuals or corporations) who can afford the service is allowed to subscribe
- Advantages
  - As a subscriber, there is no need for a staff to install or to operate the network
  - Gives subscribers the ability to use the state-of-the-art networking without maintaining technical expertise
  - Arbitrary subscribers at arbitrary locations can connect to the network
- Disadvantage
  - Subscription fee

Virtual Private Network (VPN)

- A combination between private and public networks
- Allows a company with multiple sites to have a private network, but use a public network as a carrier
- VPN technology restricts traffic so that
  - Packets can travel only between the company's sites
  - Even if outsiders accidentally receive a copy of a packet, they will not be able to understand the contents (data encryption)
- A special hardware and a software system must be installed in each site
- The network manager at each site must configure the routing process

Network Services

- Computer networks can offer two different types of services
  - Connection-oriented service
  - Connectionless service

Connection-oriented Service

- Modeled after the telephone system: to talk to someone, you pick up the phone, dial the number, talk, and then hang up
- Similarly, a user of the service
  - Requests a "*connection*" to another user
  - Waits for the network to form the connection
  - Keeps the connection in place, while sending data
  - Terminates the connection when no longer needed
- Note that: After establishing a connection, there is no need for the destination address anymore; instead, a "*connection identifier*", which is much shorter than the full destination address, is used
- Changes in a routing table inside a switch does not affect the path from source to destination
- The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them in the same order at the other end

- A connection from one computer to another through a connection-oriented service is called Virtual Circuit, or Virtual Channel, (VC);
  The Term "*virtual*" arises because the circuit is achieved by placing routes in routing tables, not by establishing physical wires
- Advantages
  - Easy to price: charges per connection time, not by the amount of transfered data
  - Easy to inform communicating computers immediately when a connection breaks
  - shorter header per packet
- Disadvantages
  - Static routing, even if a failure happened
  - Setup and cleanup overhead are needed
- Example: ATM network

Connectionless Service

- Modeled after the postal system
- Each packet carries the full destination address
- Each packet is routed through the system independent of the other packets
- In a connectionless service, a packet is called "*datagram*"
- Advantage
  - Less setup overhead: there is no need to establish a connection and there is no cleanup after sending data
  - No dedicated path
  - Broadcast and/or multicast are allowed
- Disadvantages
  - Full size header
  - Routing overhead per packet
  - There is no guarantee that all messages will be received in order
- Examples: Ethernet and Token Ring

Permanent Virtual Circuit (PVC)

- A dedicated connection between a pair of computers
- In early computer networks
  - A permanent connection was achieved by installing a dedicated wires between a pair of computers
- In modern networks
  - A permanent connection is achieved by configuring the network to form a dedicated communication path electronically (virtual circuit)
  - Once it has been configured, the configuration is stored in a non-volatile memory, e.g., on a disk.
  - The connection can be reestablished after a power failure or a reboot.

Switched Virtual Circuit (SVC)

- A connection is formed, used, and then terminated
- Although it is possible to keep a switched connection in place for a long time, most switched connections persist for a short duration
- If a computer crashes while using a switch connection, the computer must reestablish the connection after rebooting
- A switched network needs only one physical connection per computer, i.e., there is no need to install, or change, wiring when a computer decides to communicate with another computer

Network Performance Characteristics

- Quantitative metric measures are needed to compare any two networks
- There are two primary performance measures
  - Delay
  - Throughput
- Delay
  - Definition: The time required for one bit to travel from source to destination through the network
  - Sources of delay (delay components) include
    - Fixed delays (nearly constant)
      - Propagation delay
      - Switching delay
    - Variable delays
      - Queuing delay (rises as the load on the network increases)
      - Access delay (depends on throughput)

- Throughput
  - Definition: A measure of the rate at which data can be sent through the network
  - Throughput is usually specified in bits per second
  - Note that: throughput is a measure of capacity, not a speed
  - The term "*throughput capability of the underlying hardware*" is called "*hardware capacity*" or "*bandwidth*"
  - The "*effective throughput*" (i.e., the rate at which a computer can send data) is less than the hardware bandwidth, this is because in most technologies, each frame contains a header;
    A header is considered as an overhead, since it is not part of the data
  - It is impossible for a user to send data faster than the rate at which the hardware can transfer bits, i.e., the hardware bandwidth gives an upper bound on throughput
  - If a network operates at a close to 100% of its throughput capability, it will experiences a severe delay
  - The hardware bandwidth is often used as an approximation of the network's throughput

- There are some other network performance measures are derived from delay, throughput, and bandwidth; these measures include
  - Utilization
    * The ratio of throughput to hardware capacity (i.e., bandwidth)

$$U = \frac{throughput}{bandwidth} \qquad (0 \leq U \leq 1)$$

  - Delay-throughput product
    * The quantity of data "*in transit*"

$$delay \times throughput$$

  - Delay-bandwidth product
    * The upper bound on the quantity of data "*in transit*"

$$delay \times bandwidth$$

  - Delay-utilization relationship

$$expected\ delay = \frac{the\ delay\ when\ the\ network\ is\ idle}{1 - U}$$

    * The delay when the network is idle consists of propagation and switching delays

DELIVERY CONTROL, FLOW CONTROL, AND CONGESTION CONTROL

Delivery Control

- Due to the nature of connectionless network service, packets might be
  - Delivered out of order
  - Lost
  - Duplicated
- Sender/receiver must solve the following problem:
  - Handling out-of-order delivery
  - Retransmitting lost/damaged packets
  - Eliminating duplicate packets

Handling Out-of-order Delivery

- A connectionless network may deliver packets out of order, due to the fact that routing table can be changed
- To deal with this problem
  - The sender
    - Attaches a sequence number to each packet
  - The receiver
    - Maintains the sequence number of the next expected packet
    - Maintains a list of additional packets that arrived out of order
    - Examines the sequence number of an arrived packet
    - If the packet is the next expected packet, i.e., it arrived in order,
      - This packet is delivered for processing
      - The out-of-order list is checked to see whether, or not, additional packets can be considered the next expected packet(s), and hence delivered for processing
      - Update the sequence number of the next expected packet
    - Otherwise, i.e., the arrived packet is out of order, it will be added to the out-of-order list

Retransmitting Lost/Damaged Packets

- Packet loss is a fundamental problem in the connectionless network service
- When a receiver detects a transmission error, it discards the frame; *Why?*
- To handle this problem, an acknowledgment mechanizm is needed
  - Whenever an undamaged-frame arrives, the receiver sends a small message back to report this successful reception
  - The sender takes responsibility for ensuring that each packet is successfully transferred
    - Whenever a sender sends a packet, it starts a timer
    - If the acknowledgment arrives before the timer expires, it cancels the timer
    - If the timer expires before the arrival of the acknowledgment, the sender
      - Re-sends another copy of the packet
      - Re-starts the timer again

Eliminating Duplicate Packets

- Malfunctioning hardware can cause packets to be duplicated
- Sequencing solves this problem
  - When a packet arrives, the receiver examines the sequence number
  - If the sequence indicates a previously received packet, the receiver discards the new copy

Flow Control

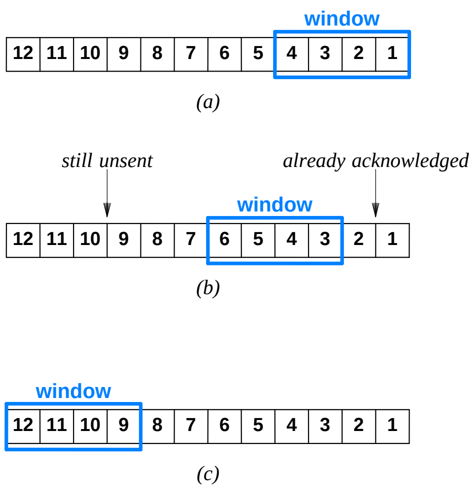
- Computers do not all operate at the same speed
- When a computer sends data across a network faster than the destination speed, i.e., the destination can not absorb these data in time, data overrun occurs, i.e., data will be lost
- This is a buffering problem
- Several techniques are available to handle data overrun, including
  - Stop-and-go
  - Sliding window

Stop-and-go

- The simplest form of flow control
  - A sender waits after transmitting each packet
  - When the receiver is ready to receive another packet, it sends a control message, usually a form of acknowledgment
- A half-duplex physical channel would be sufficient for this scheme
- This protocol can prevent data overrun, however, it can cause extremely inefficient use of the network bandwidth

Sliding Window

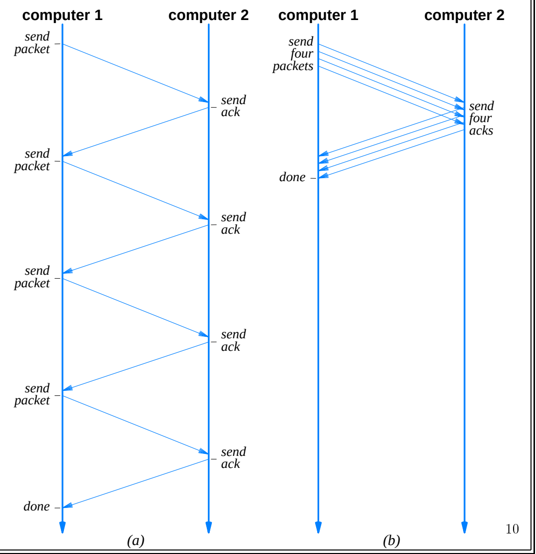
- Receiving side
  - Establishes multiple buffers
  - Informs the sender with the size of these buffers
  - When a packet is successfully arrived in order, the receiving side
    - * Sends an acknowledgment signal to the sender
  - Forward a packet to the application,
  - Slide the window
  - Re-advertise the buffer size
- Sending side
  - Transmits packets to fill all available buffers
  - When getting acknowledgment
    - * Slide the window
    - * Transmits more packets



(a) When transmission begins; (b) after two packets have been acknowledged; (b) after eight packets have been acknowledged

Stop-and-go Versus Sliding Window

Time proceeds down the page; Each arrow shows one message sent from one computer to the other



- A sliding window scheme aggressively uses more of the underlying network bandwidth to improve the network utilization
- Throughput$_{sliding\ window}$ is equal the minimum of
  - The underlying hardware bandwidth
  - The throughput$_{stop-and-go}$ $\times$ window size

Congestion Control

- Congestion is a fundamental problem in packet switching networks
- Caused by traffic, not hardware failure
- Analogous to the congestion on a highway
- When more packets arrive than can be sent, the queue grows and the effective delay increases (principle cause of delay)
- If congestion persists, a packet switch will run out of memory and begin discarding packets
- Although retransmission can be used to recover lost packets, retransmission takes time
- In a sever congestion cases, the network becomes unusable (*congestion collapse*)
- Network software attempts to avoid this situation, by monitoring the network and reacting quickly once congestion starts by reshaping the traffic

- There are two approaches to avoid network congestion:
  - Let packet-switches to inform senders when congestion starts to occur
  - Use packet loss as an estimate of congestion
- The appropriate response to congestion is to reduce the rate at which packets are being transmitted
- Sliding window scheme can achieve the same effect by temporarily reducing the window size
- Note that: a congestion control mechanism tries to avoid the network collapse by doing the opposite of what sliding window does; hence a balance between the two schemes is needed

NETWORK PROTOCOLS AND LAYERING

Why Protocols?

- Basic communication hardware consists of mechanisms that can transfer bits from one point to another
- During communications, many problems can occur, e.g.,
  - Bits corrupted or destroyed
  - Entire packet lost
  - Packet duplicated
  - Packets delivered out of order
- It is not that easy to just rely only on raw hardware to solve such problems
- Using raw hardware to communicate is analogous to programming by machine language, i.e., 0's and 1's
- To aid programmers, computers attached to a network use complex software that provides a convenient, high-level interface, for applications
- This software handles most of the low-level communication details and problems

- All parties involved in the communication process must agree on a set of rules to be used when exchanging messages, e.g.,
  - Messages format
  - Messages meaning
  - Procedures for handling problems
- Diplomats call such an agreement a *protocol*
- The term is applied to computer communication as well, i.e., a set of rules that specify the messages format and the actions required for each message is known as a *network protocol* or a *computer communication protocol*
- The software that implements such rules is called *protocol software*
- Most application programs rely on the network protocols software to communicate; they do not interact with network hardware directly

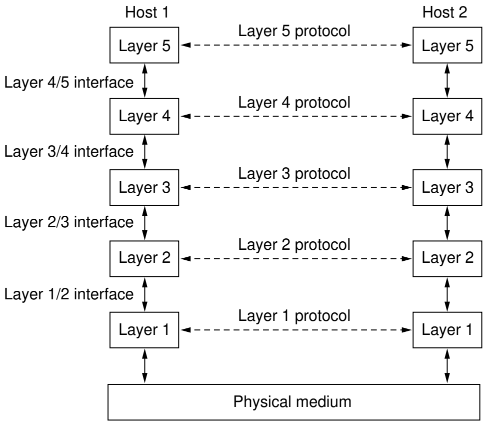
Protocol Design

- Instead of having a single giant protocol that specifies complete details for all possible forms of communications, designers have chosen to
  - Divide communication problems into subgroups
  - Design a separate protocol for each subgroup
- Doing so makes each protocol easier to
  - Design/analyze
  - Implement/test
- Within a set of protocols (a.k.a. a *protocol family*, a *protocol suite*, or a *protocol stack*)
  - Each protocol should handle a part of the communication problems which is not handled by other protocols
  - Protocols should work/interact together efficiently
  - The combination of all protocols should handle all possible hardware failures or other exceptional conditions

Layering Model

- *Layering model* is a conceptual framework which is used to explain the interaction among a set of protocols
- A layer is built upon the one below it
- The number of layers, the name, the contents, and the function of each layer may differ from network to network
- The purpose of each layer is to
  - Offer a certain service to the higher layers
  - Shield those layers from the details of how the offered services are actually implemented
- Layer n on one machine carries on a conversation with layer n on another machine; the rules used in this conversation are collectively known as the layer n protocol
- In reality, no data are directly transferred from layer n on one machine to layer n on another machine

- Instead, each layer passes data and control information to the layer immediately below it, through layer interface, until the lower layer is reached where the actual communication can be made



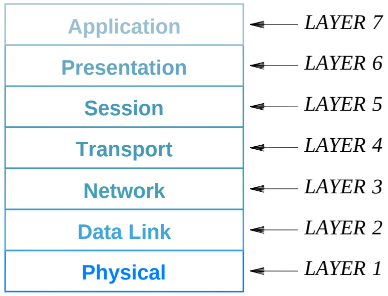
- Examples: Telephone and translator conversations

Network Architecture

- A set of layers and protocols is called a *network architecture*
- The specification of an architecture must contain enough information to allow an implementer to write the program, or to build the hardware for each layer, so that it will correctly obey the appropriate protocol
- The layer interfaces are not part of the architecture; it is not even necessary that the interfaces on all machines in a network to be the same

ISO OSI 7-Layers Reference Model

- The ISO OSI 7-layers reference Model was the first step toward having an international standard for network protocols



The ISO OSI 7-layer reference model

ISO stands for International Standard Organization
OSI stands for Open System Interconnection

- Layer 1: Physical layer
  - Basic network hardware
- Layer 2: Data link (media access) layer
  - Organizing/preparing frames for transmission over the network
- Layer 3: Network layer
  - Addressing and forwarding/routing
- Layer 4: Transport layer
  - Assuring reliable transfer
- Layer 5: Session layer
  - Authentication
- Layer 6: Presentation layer
  - Data representation
- Layer 7: Application layer
  - Individual application programs

Layer 1: Physical Layer

- The physical layer is concerned with transmitting raw bits over a communication channel
- The physical layer protocol makes sure that when one side sends a "1", it is received by the other side as a "1", not as a "0"
- Typical questions to be answered in this layer include:
  - How many volts should be used to represent a "1" and a "0"?
  - How many microseconds should a bit last?
  - Whether transmission may proceed simultaneously in both directions, or not?
  - How is the initial connection established?
  - How is it turn off when both sides are finished?
  - How many pins does a network connector have?
  - What is each pin used for?
- In general, this layer largely deals with physical, electrical, and procedural interfaces issues

Layer 2: Data Link (Media Access) Layer

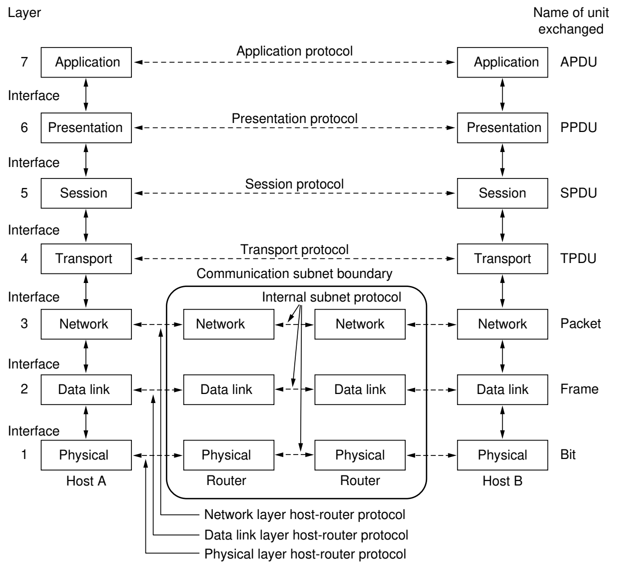
- The main tasks of this layer is to
  - Divide the input data into frames
  - Solve the problem of how to control access to the channel (in case of shared channel)
  - Transmit frames sequentially
  - Process acknowledgment frames sent back by the receiver
  - Detect damaged packets (CRC checking)
  - Solve the problems caused by damaged, lost, and duplicate frames
- Note that: since the physical layer basically accepts and transmits a stream of bits without any regard to meaning or structure, it is up to the data link layer to create and recognize frame boundaries (byte stuffing)

Layer 3: Network Layer

- The network layer is concerned with controlling the operation of the network
  - Determining how packets are routed from source to destination
  - Controlling congestion

Layer 4: Transport Layer

- The basic function of the transport layer is to
  - Accept data from the session layer
  - Split it up into smaller units if need be
  - Pass these units of data to the network layer
  - Ensure that all the units of data arrive correctly at the other end
- The transport layer is a true end-to-end layer, from source machine to destination machine, not from network node to another network node



Layer 5: Session Layer

- Allows users on different machines to establish sessions between them
- Provide authentication
- Provide synchronization

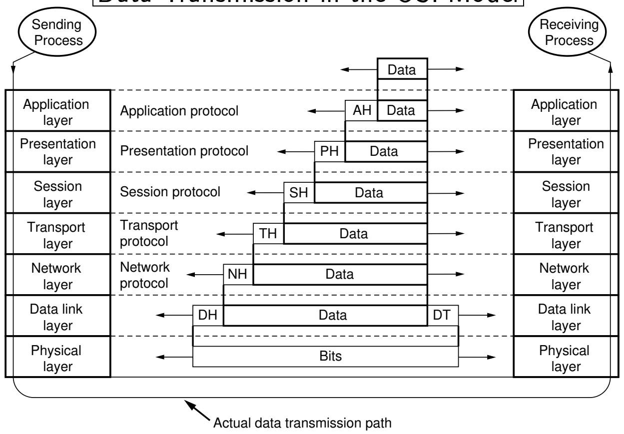
Layer 6: Presentation Layer

- The main function of the presentation layer is to convert data from a representation on one computer to another representation suitable for another computer;
  Note that different computers may have different codes for representing data
- Unlike all other lower layers, which are just interested in moving bits reliably from here to there, the presentation layer is concerned with the syntax and semantics of the information transmitted

Layer 7: Application Layer

- The application layer contains a variety of protocols that are commonly needed, for example:
  - Ftp
  - Email
  - Editor

Data Transmission in the OSI Model

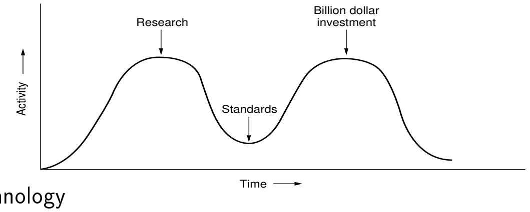


An example of how the OSI model is used

- Note that: although actual data transmission is vertical, each layer is programmed as if it were horizontal

Why the ISO Model Never Happened

- bad timing



- Bad technology
  - 7 is not the right choice for the number of layers, but the IBM SNA (Systems Network Architecture) was 7 layers!!!
    - ∗ The session layer has little use in most applications
    - ∗ The presentation layer is nearly empty
  - The data link layer is full
- Bad implementation
- Bad politics