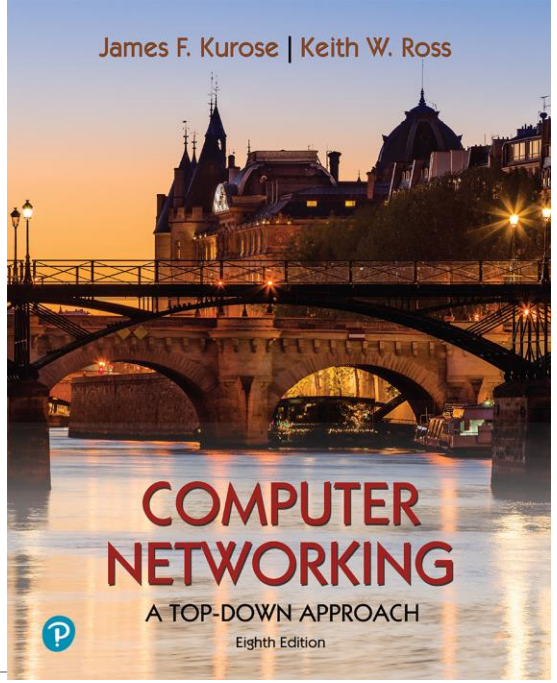


Unit-1

CS44 Data Communications and Networking

Dr. SSC & LM

Department of Computer Science and Engineering
RIT, Bangalore

Books

Text Books:

- Data Communications and Networking with TCP/IP Protocol Suite, Behrouz A.Forouzan, McGraw Hill, 6th Edition, 2021.
- James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach, 8th edition, Addison-Wesley, 2021.

Outline of Unit-1

- DATA COMMUNICATIONS
  - Components
  - Data Representation
  - Data Flow
- NETWORKS
  - Network Criteria
  - Physical Structures
- NETWORK TYPES
  - Local Area Network
  - Wide Area Network
  - Switching
  - The Internet
  - Accessing the Internet
- TCP/IP PROTOCOL SUITE
  - Layered Architecture
  - Layers in the TCP/IP Protocol Suite
  - Description of Each Layer
  - Encapsulation and Decapsulation
  - Addressing
  - Multiplexing and Demultiplexing
- THE OSI MODEL
  - OSI versus TCP/IP
  - Lack of OSI Model's Success

Outline of Unit-1

- Web and HTTP
  - Overview of HTTP
  - Non-Persistent and Persistent Connections
  - HTTP Message Format
  - User-Server Interaction-Cookies
  - Web Caching
  - The Conditional GET
- FTP
  - FTP Commands and Replies
- Electronic mail in the Internet
  - SMTP
  - Comparison with HTTP,
  - ail Access Protocols.
- DNS
  - The Internet's Directory Service: Services Provided by DNS
  - Overview of How DNS Works
  - DNS Records and Messages
- P2P applications: P2P File Distribution.

Telecom vs. Datacom

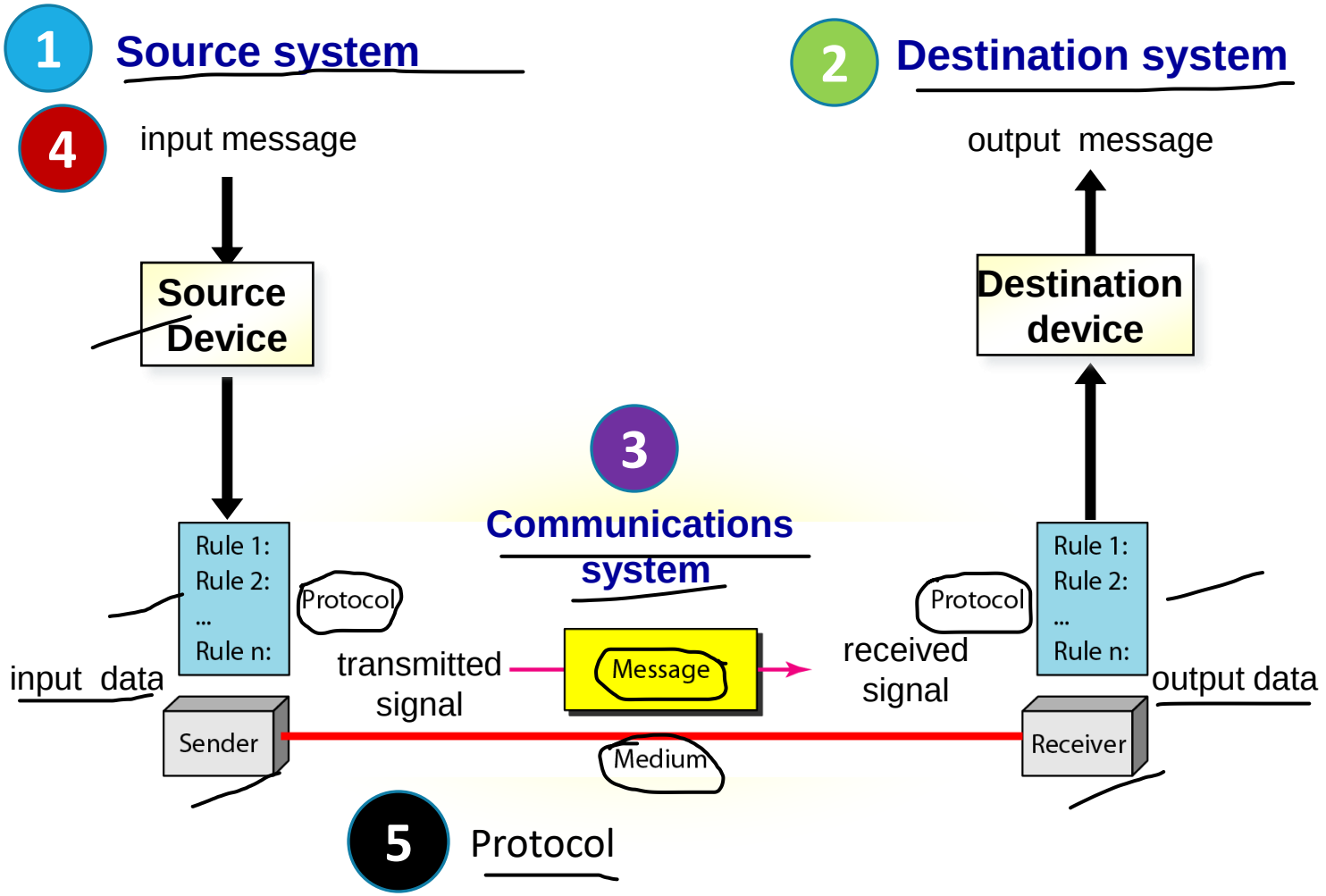
- Communications
  - Transfer of data from one device to another via some form of transmission medium
  - Communicate = sharing information
    - local (usually occurs face to face) or remote (takes place over distance)
- Telecommunications (*tele* is Greek for “far”)
  - Communication over a distance by definition
  - All type of communications using signaling such as voice (telephone), images (television), data (computers)
- Data communications
  - subset of telecommunications that involves the transmission and reception of data to and from computers and components of computer systems

Data communications

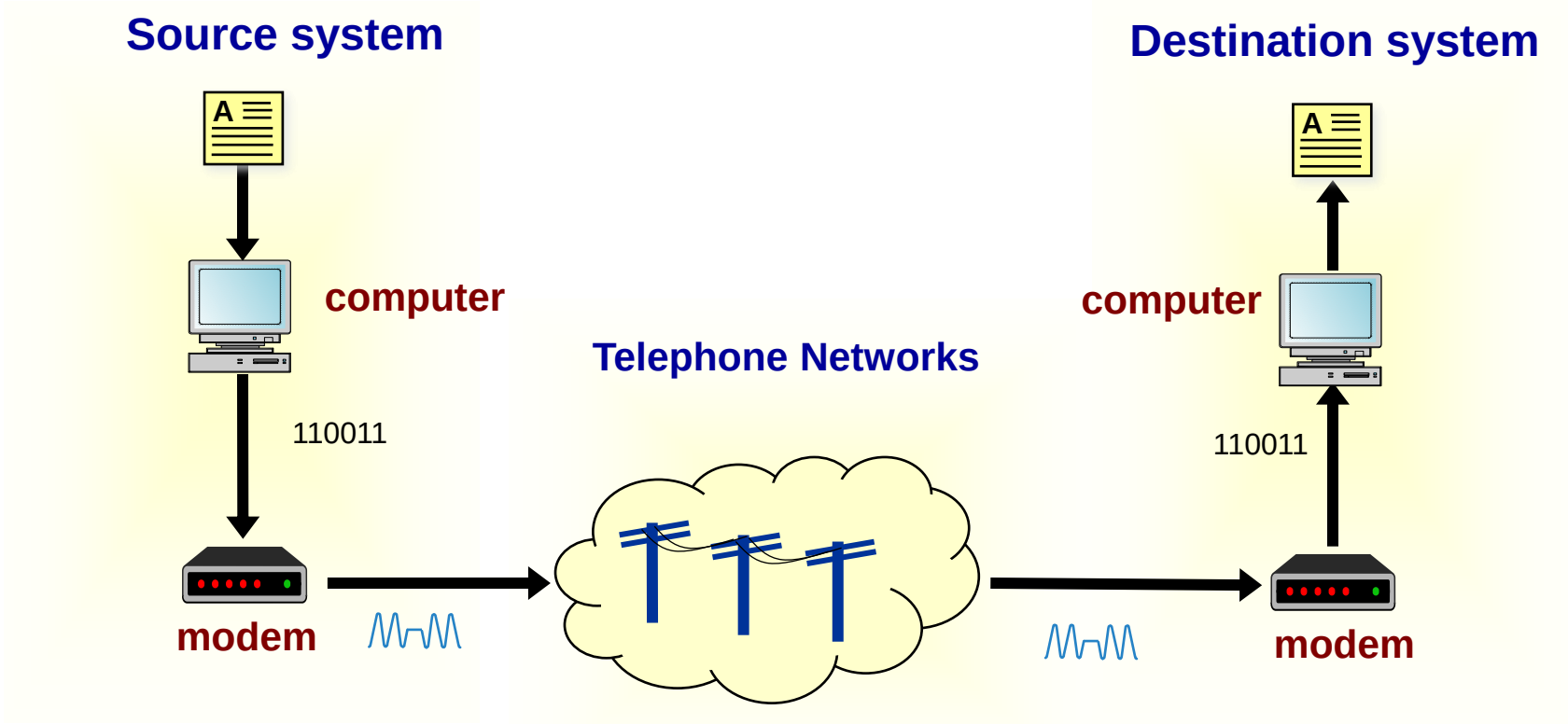
- depends on four fundamental characteristics:
  - Delivery: deliver data to the correct destination
  - Accuracy: deliver the data accurately
  - Timeliness: deliver data in a timely manner
  - Jitter: the variation in the packet arrival time
- A data communications system has five components

Component	Description	Example
Message	The message is the information (data) to be communicated	Popular forms of information include text, numbers, pictures, audio, and video
Sender	the device that sends the data message	a computer, workstation, telephone handset, video camera, and so on
Receiver	the device that receives the message	Same as sender
Transmission medium	the physical path by which a message travels from sender to receiver	twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves
Protocol	a set of rules that govern data communications	

Communication Model



Example of Communication System



Protocols and Standards

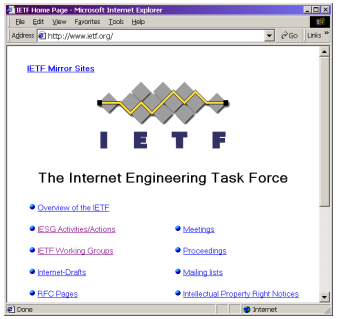
- Protocol
  - A set of rules governing data communications
    - Syntax: format of data block
    - Semantics: meaning of each section
    - Timing: speed and sequencing
- Standard
  - De facto (in practice) standards
    - → not approved but widely adopted
  - De jure (in law) standards
    - → approved by an organization

Standards Organizations

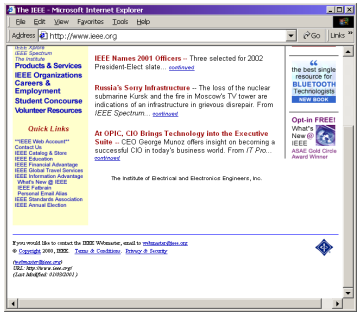
- List of major standard bodies

ITU-T

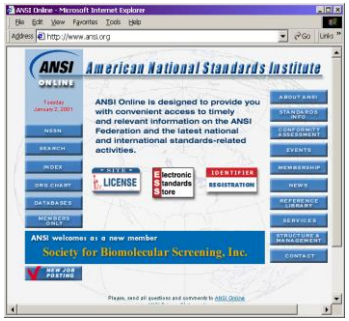
ISO



IETF



IEEE



ANSI

EIA

ITU-T

International Telecommunications Union-Telecommunication Standard Sector

- International standard organization that develops standard for telecommunications
- Some well known standards
  - V series
  - X series
  - H series
  - G Series

ISO

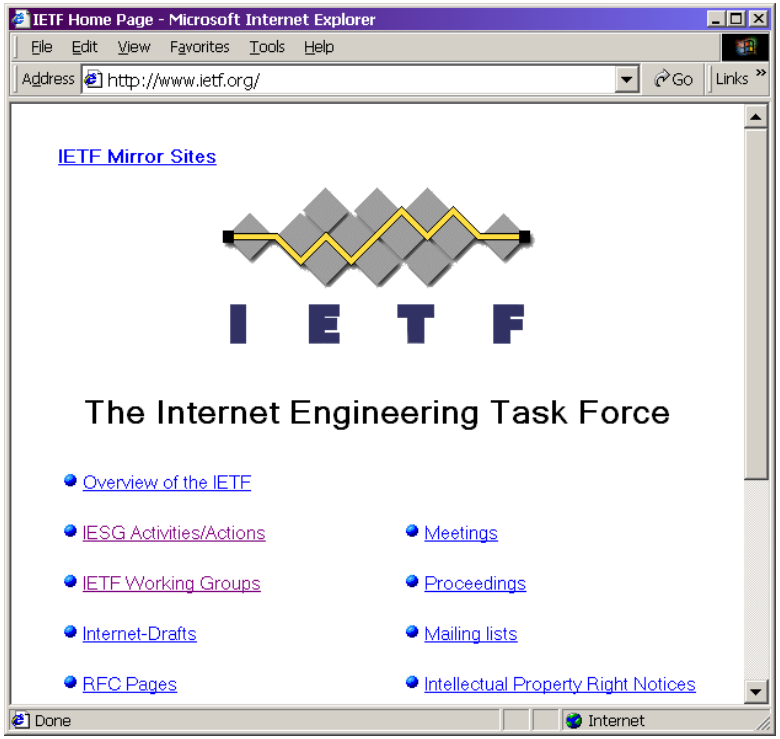
International Standards Organization

- An International standard organization that develops standard for variety of fields

IETF

Internet Engineering Task Force

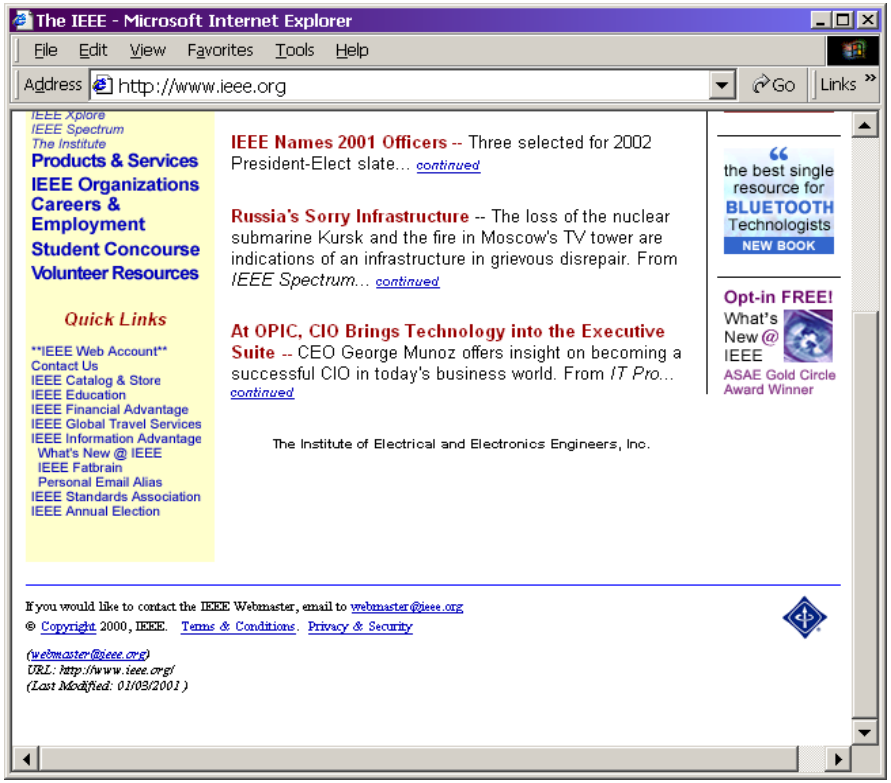
- A large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture



IEEE

Institute of Electrical and Electronics Engineers

- The largest professional group involved in standards for computing, communications, electrical and electronics

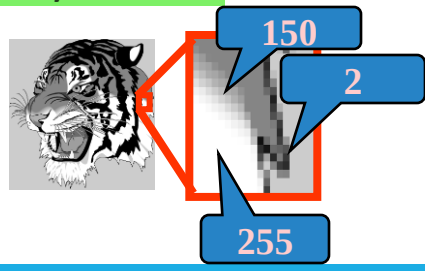


Data Representation

- text is represented as a bit pattern, a sequence of bits (0s or 1s)
  - Different sets of bit patterns have been designed to represent text symbols – code
  - the process of representing symbols is called coding.
  - **Unicode** - prevalent coding system - uses 32 bits to represent a symbol or character used in any language in the world.
    - **American Standard Code for Information Interchange (ASCII) developed some decades ago in the United States**
      - constitutes the first 127 characters in Unicode and is also referred to as **Basic Latin**
- Numbers – also represented by bit patterns
  - 8/16/32 bit integers, signed/unsigned
  - floating-point
  - ASCII is not used to represent numbers
  - It is directly converted to a binary number to simplify mathematical operations.

Data Representation …

- Images
  - composed of a matrix of pixels (picture elements) => each pixel is a small dot
  - size of the pixel depends on the *resolution*
  - *More pixels in a image* - a better representation of the image (better resolution), but more memory is needed to store the image
  - each pixel is assigned a bit pattern
    - The size and the value of the pattern depend on the image
      - For an image made of only black and white dots (e.g., a chessboard), a 1-bit pattern is enough to represent a pixel.
      - represent color images: RGB (combination of three primary colors: red, green, and blue) and YCM (combination of three other primary colors: yellow, cyan, and magenta)
  - Graphics formats JPG, PNG, *etc.*

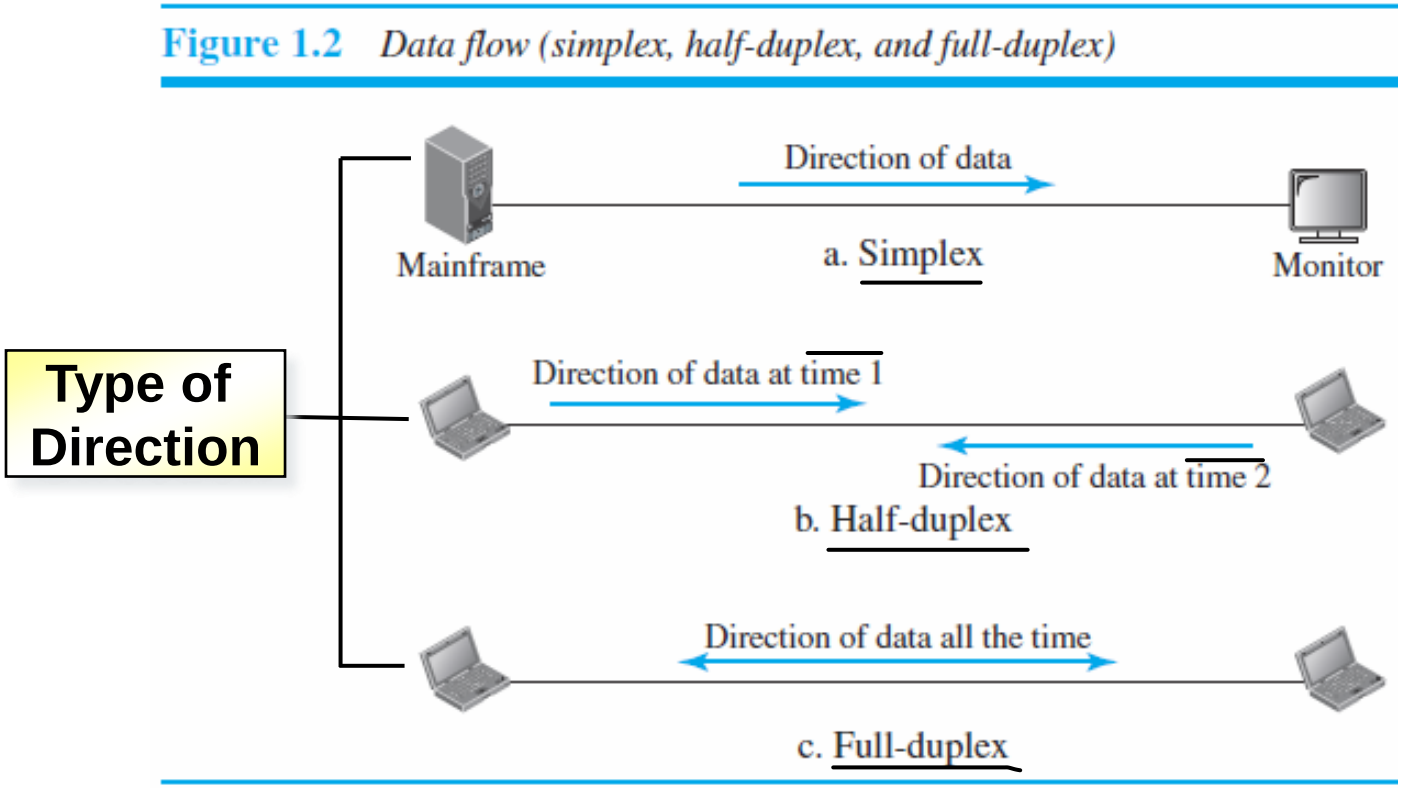


Data Representation …

- Audio →recording or broadcasting of sound or music
  - Samples of continuous, not discrete signal
  - Audio is by nature different from text, numbers, or images.
- Video → recording or broadcasting of a picture or movie
  - can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion
  - Sequence of bitmap images

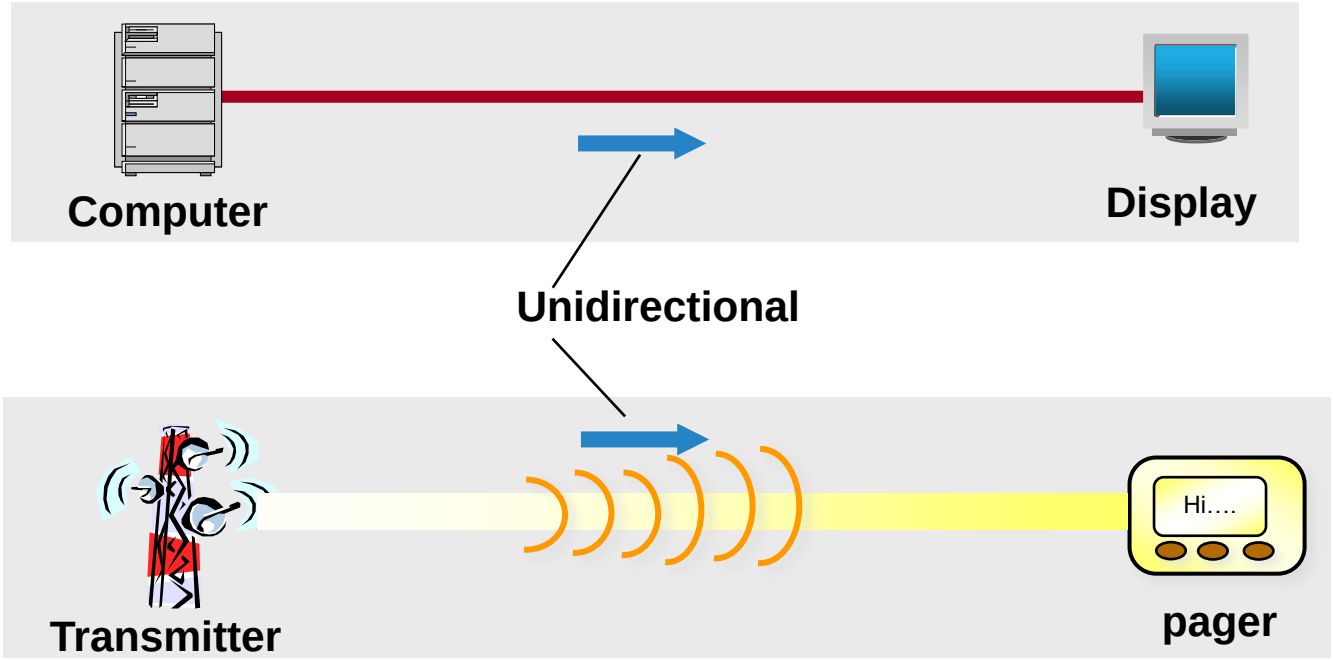
Data Flow

- Direction of data flow



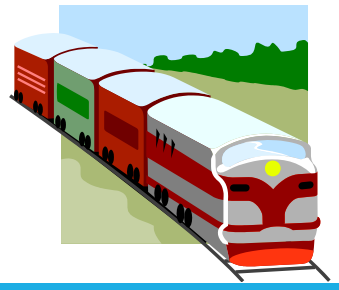
Simplex

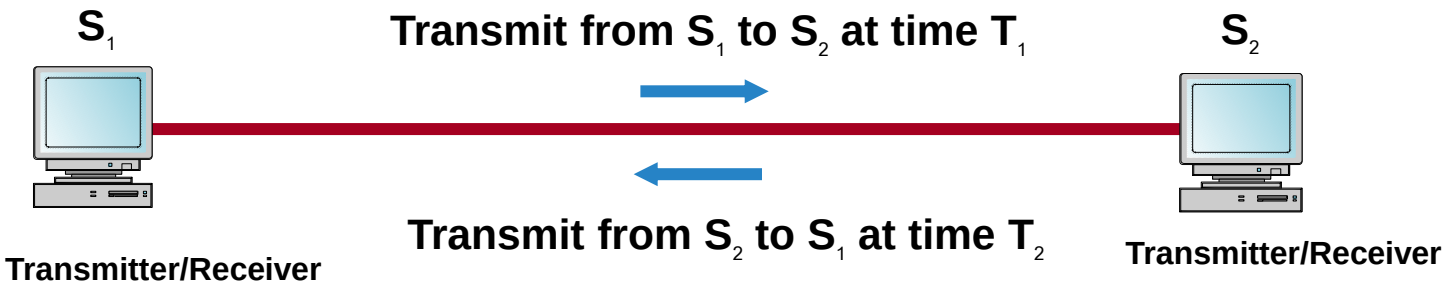
- Data are only transmitted in one direction - One device can only transmit; the other can only receive
- The simplex mode can use the entire capacity of the channel to send data in one direction.
- Eg. the keyboard can only introduce input; the monitor can only accept output.



Half-duplex

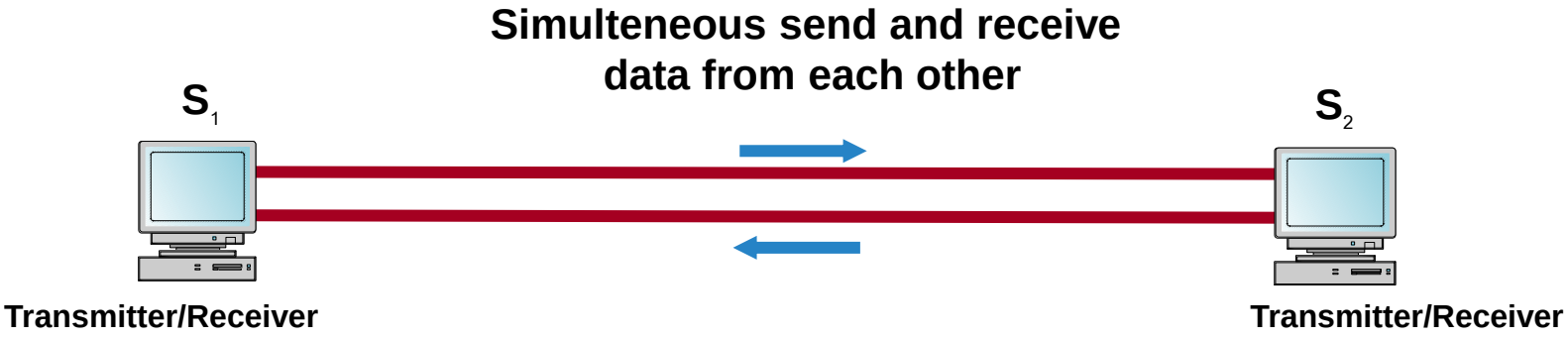
- Each stations can transmit or receive; but only one at a time (alternate in transmitting)
- When one device is sending, the other can only receive, and vice versa
- The entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time
- Example: Walkie-Talkies, CB (citizens band) radios





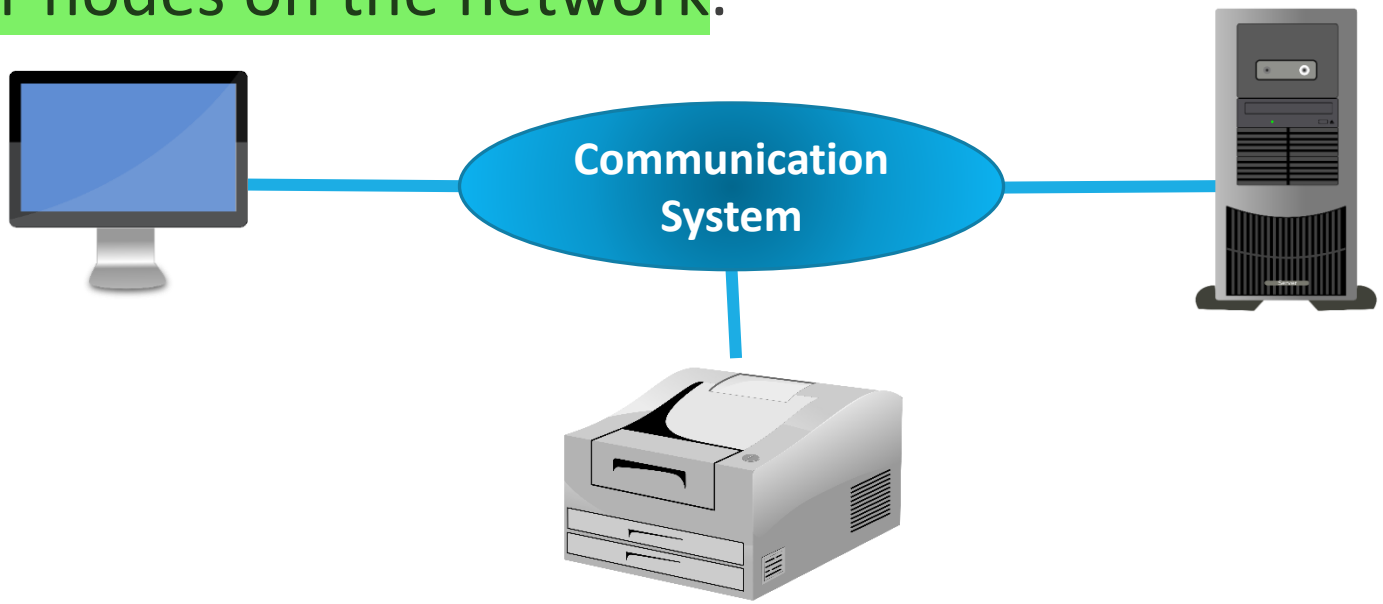
Full-duplex

- *Both stations can transmit and receive simultaneously*
- signals going in one direction share the capacity of the link with signals going in the other direction
- sharing can occur in two ways:
  - Link with two separate transmission path, one for sending and the other for receiving
  - channel is divided between signals traveling in opposite directions
- Example: telephone network -When two people are communicating by a telephone line, both can talk and listen at the same time



Networks

- A network is a set of two or more devices connected through a communication system
- A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

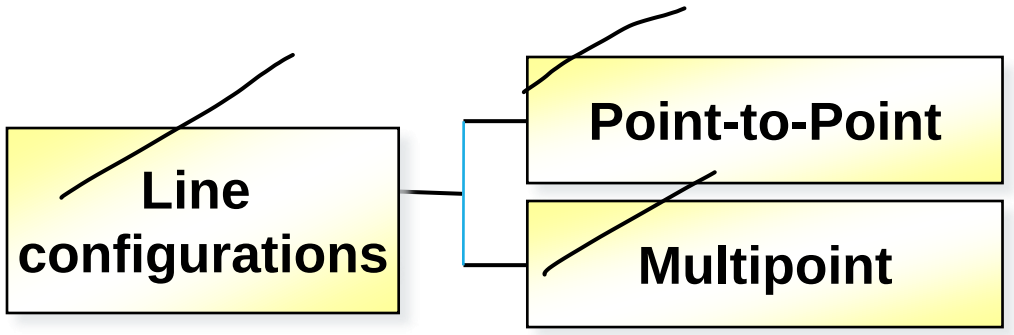


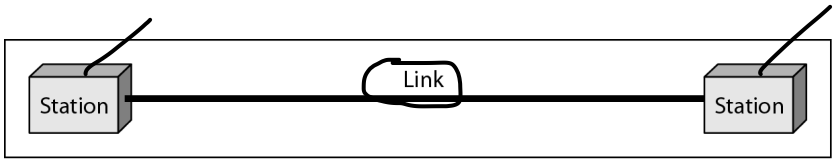
Network Criteria

- performance: measured in many ways, including transit time (amount of time required for a message to travel from one device to another) and response time (elapsed time between an inquiry and a response)
  - depends on a number of factors
    - the number of users
    - the type of transmission medium
    - the capabilities of the connected hardware
    - the efficiency of the software
  - evaluated by two networking metrics: **throughput** (expected more) and **delay** (expected less)…… Often contradictory
- Reliability: measured in terms of
  - accuracy of delivery
  - **frequency of** failure
  - the time it takes a link to recover from a failure
  - the network's robustness in a catastrophe
- Security
  - protecting data from unauthorized access
  - protecting data from damage and development
  - implementing policies and procedures for recovery from breaches and data losses

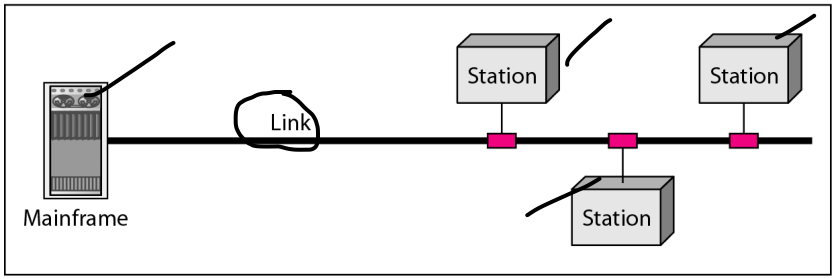
Physical Structures

- Network attributes: ***Type of Connection and Physical Topology***
- For communication to occur, two devices must be connected in some way to the same link at the same time.
  - There are two possible types of connections: point-to-point and multipoint





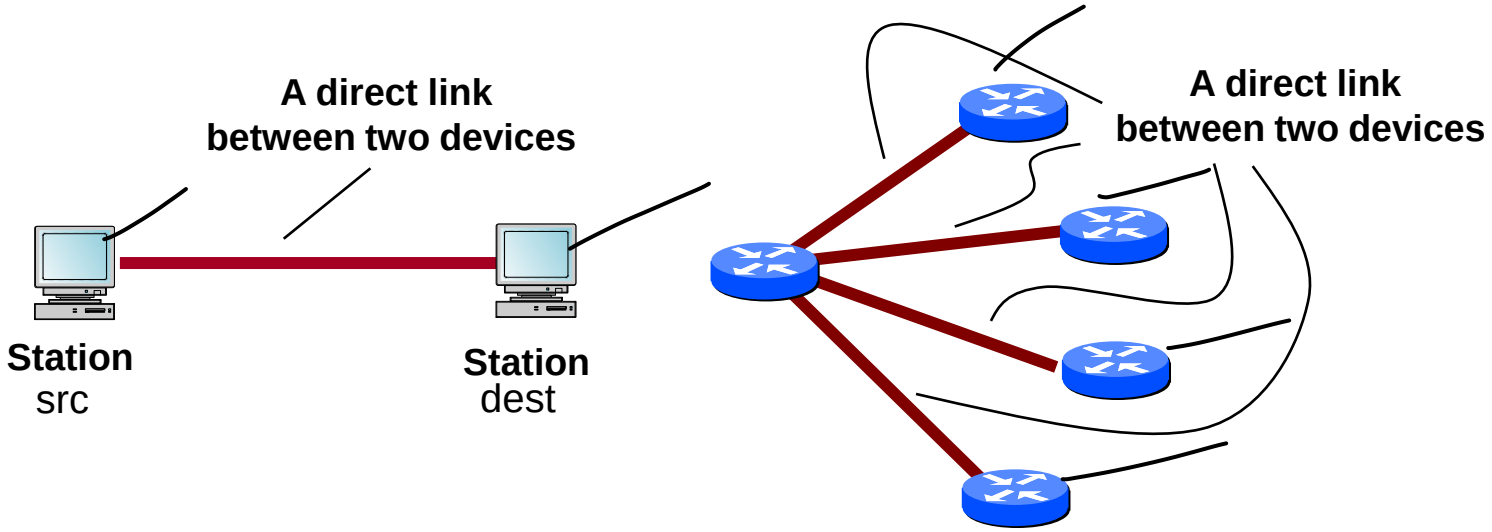
a. Point-to-point



b. Multipoint

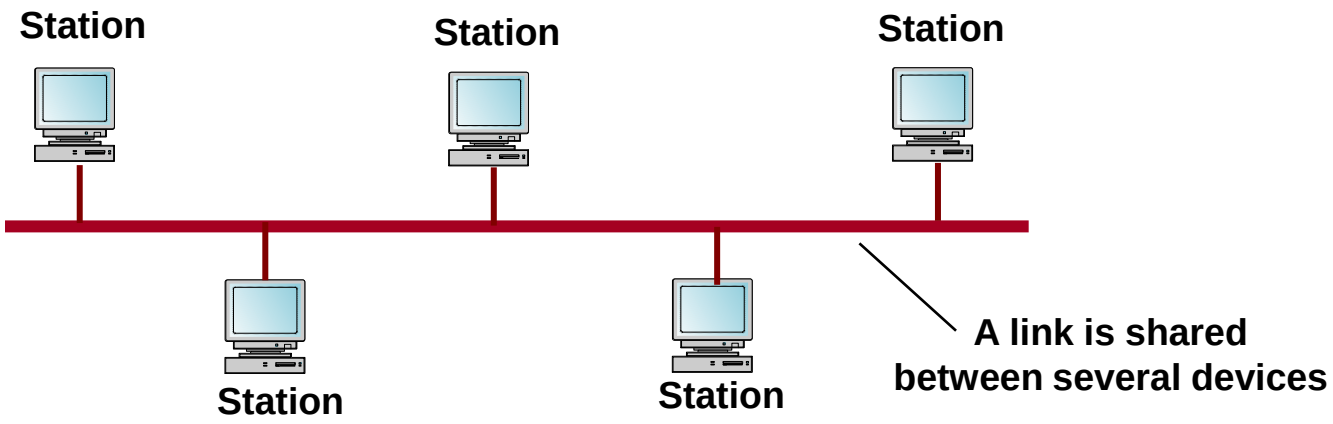
Point-to-Point

- Only two devices share the medium
- The entire capacity of the link is reserved for the transmission between those two devices



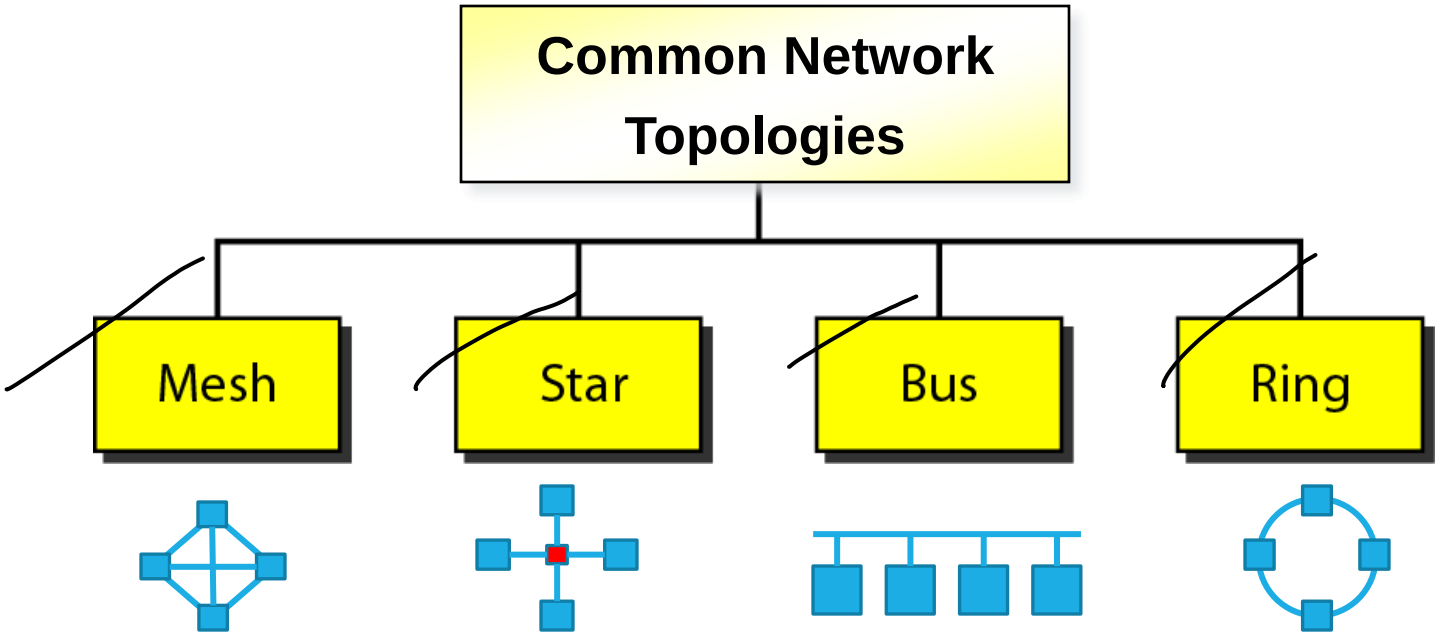
Multipoint (or Multidrop)

- More than two devices share the same medium
- The capacity of the link is shared, either spatially (use the link simultaneously)or temporally (users must take turns)



Physical Topology

- the way in which a network is laid out physically
- Network topology: geometric representation of the relationship of all the links and linking devices (usually called *nodes) to one another*

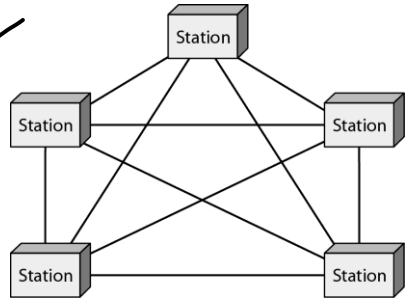


Mesh Topology - point-to-point

- every device has a dedicated point-to-point link to every other device
- number of physical links in a fully connected mesh network with *n nodes* : *n (n – 1) physical* links or ***n (n – 1) / 2 duplex-mode links***

simplex

- every device on the network must have *n – 1 input/output (I/O) ports to* be connected to the other *n – 1 stations*

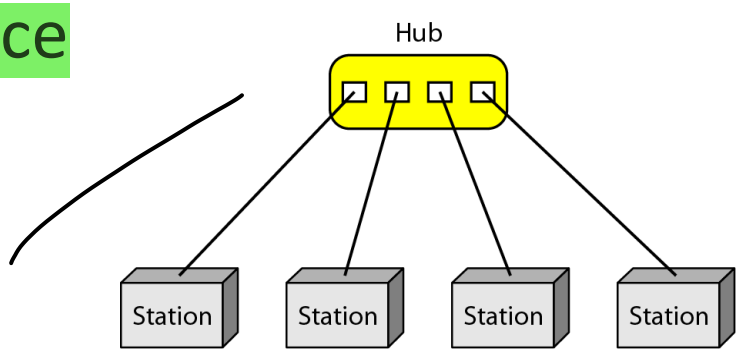


Mesh Topology

- Advantages
  - They use the dedicated links so each link can only carry its own data load – traffic problem can be avoided
  - It is robust – if one link get damaged, it cannot affect others
  - It gives privacy and security (message travels along a dedicated link)
  - Fault identification and fault isolation are easy
- Disadvantages
  - The amount of cabling and the number of I/O ports required are very large.
  - because every device must be connected to every other device, installation and reconnection are difficult
  - The sheer bulk of wiring is larger than the available space
  - Hardware required to connect each device is highly expensive.
- Applications
  - connection of telephone regional offices - each regional office needs to be connected to every other regional office
  - WAN – wide area network

Star Topology - point-to-point

- Each device has a dedicated point to point link to the central controller called “Hub” – act as a exchange
- There is no direct traffic between devices
- The transmission are occurred only through the central hub
- When device 1 wants to send data to device 2 –first sends the data to hub which then relays the data to the other connected device



Star Topology

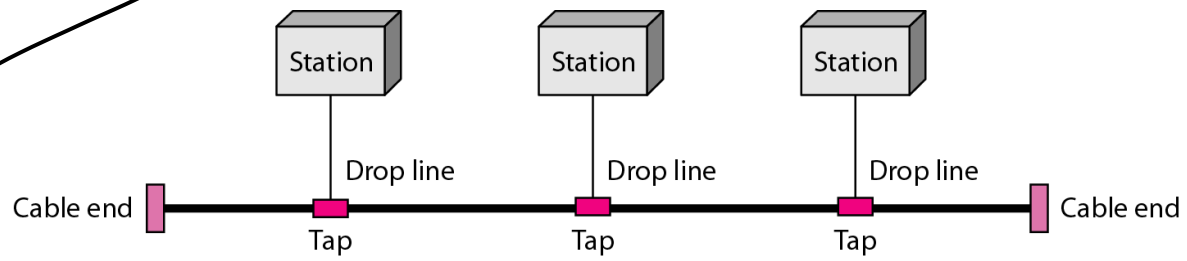
- Advantages
  - Less expensive than mesh since each device is connected only to the hub
    - each device needs only one link and one I/O port to connect it to any number of others
      - Installation and configuration are easy
      - Less cabling is needed than mesh
      - additions, moves, and deletions involve only one connection: between that device and the hub
  - Robustness - if one link fails only that link is affected – all other links remain active
    - Easy to fault identification and to remove parts
  - No disruptions to the network when connecting or removing the devices

Star Topology

- Disadvantage
  - dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead
  - often more cabling is required in a star than in some other topologies (such as ring or bus).
- Applications
  - local-area networks (LANs)
  - High-speed LANs

Bus Topology - multipoint

- One long cable acts as a **backbone to link all the devices in a** network
- Nodes are connected to the bus cable by drop lines and taps
  - drop line - a connection running between the device and the main cable
  - tap – splitter that cut the main link to create a contact with the metallic core
- Allows only one device to transmit at a time



Bus Topology

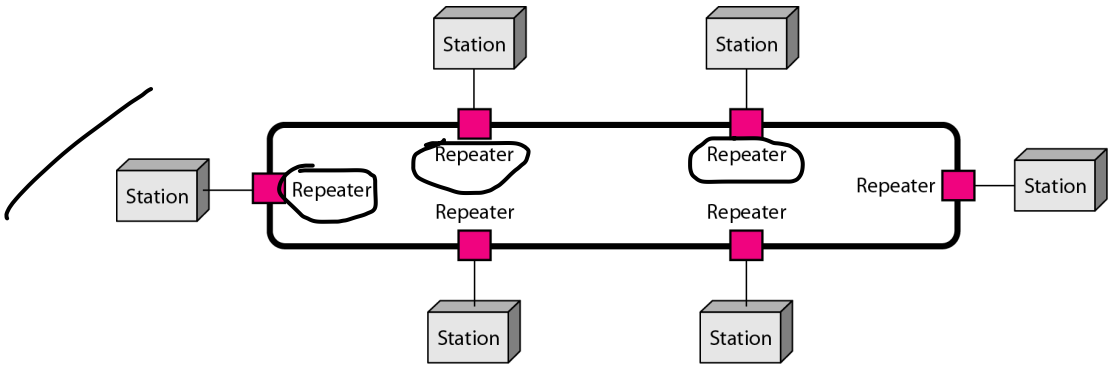
- A device want to communicate with other device on the network sends a broadcast message onto the wire and all other devices see
- But only the intended device accepts and process the message
- As a signal travels along the backbone, some of its energy is transformed into heat - it becomes weaker and weaker as it travels farther and farther
  - For this reason there is a limit on the number of taps a bus can support and on the distance between those taps

Bus Topology

- Advantages
  - ease of installation
  - less cabling than mesh or star topologies
    - Only the backbone cable stretches through the entire facility
    - Each drop line has to reach only as far as the nearest point on the backbone.
- Disadvantages
  - difficult reconnection and fault isolation
  - A bus is usually designed to be optimally efficient at installation - difficult to add new devices
    - Adding new devices may therefore require modification or replacement of the backbone
  - Signal reflection at the taps can cause degradation in quality
    - controlled by limiting the number and spacing of devices connected to a given length of cable
  - a fault or break in the bus cable stops all transmission -The damaged area reflects signals back in the direction of origin, creating noise in both directions
- Applications
  - design of early local area networks
  - Most of computer motherboards

Ring Topology – point-to-point

- each device has a dedicated point-to-point connection with only the two devices on either side of it
- A signal is passed along the ring in one direction, from device to device, until it reaches its destination
- Each device in the ring incorporates a repeater
- When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along



Ring Topology

- Advantages
  - relatively easy to install and reconfigure
  - To add or delete a device requires changing only two connections - only constraints are media and traffic considerations (maximum ring length and number of devices
  - fault isolation is simplified
    - As a signal is circulating at all times, non reception of a signal at a node within a specified period issue an alarm from that node - alerts the network operator to the problem and its location
- Disadvantage
  - unidirectional traffic
  - a break in the ring (such as a disabled station) can disable the entire network - solved by using a dual ring or a switch capable of closing off the break
- Applications
  - prevalent when IBM introduced its local-area network, Token Ring
  - Some office buildings and School premises
  - higher-speed LANs has made this topology less popular

Consideration for choosing the topology

- Money: bus network is the least expensive way to install network
- Length: of cable needed the linear bus network uses shorter length of cable
- Future growth: with start topology expending a network is easily done by adding another devices
- Cable type: most common used cable in commercial organization is twisted pair which often used with star topologies

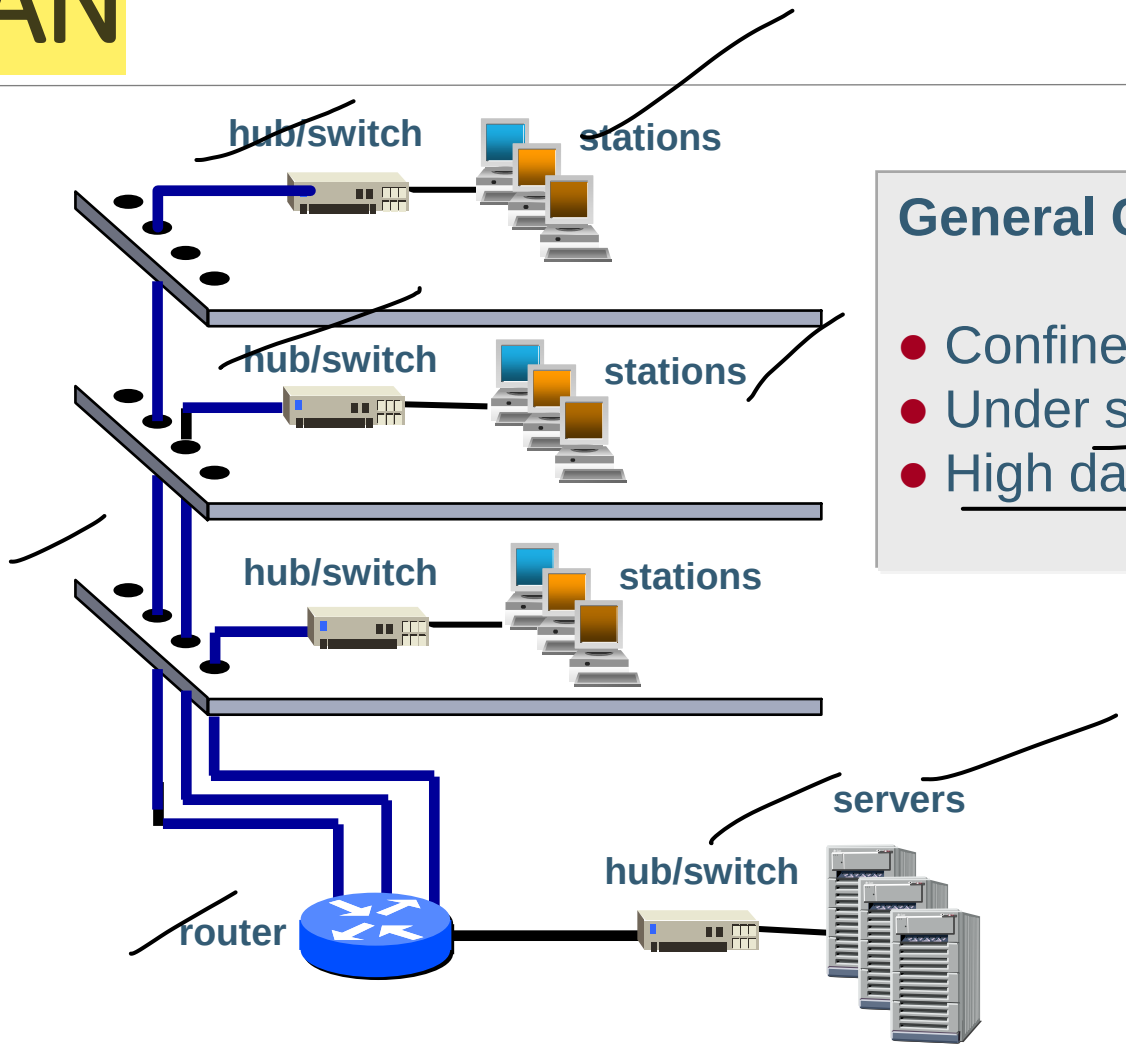
Network Types

- The criteria of distinguishing one type of network from another in the world today (difficult and sometimes confusing)
  - Size
  - geographical coverage
  - ownership.
- two types of networks, LANs and WANs, uses switching to connect networks to form an internetwork (a network of networks)

Local Area Network

- usually privately owned
- connects some hosts in a single office, building, or campus
- Simple LAN - two PCs and a printer in someone's home office
- Complex LAN include audio and video devices
- An host identifier (an address) uniquely defines the host in the LAN
- A packet carries both the source host's and the destination host's addresses

LAN

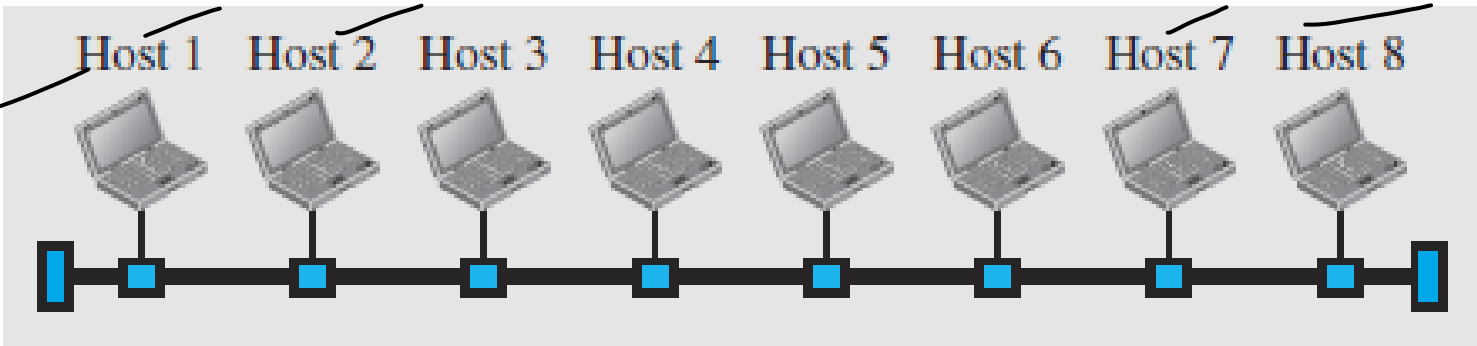


General Characteristics :

- Confined geographical area
- Under single management
- High data rate

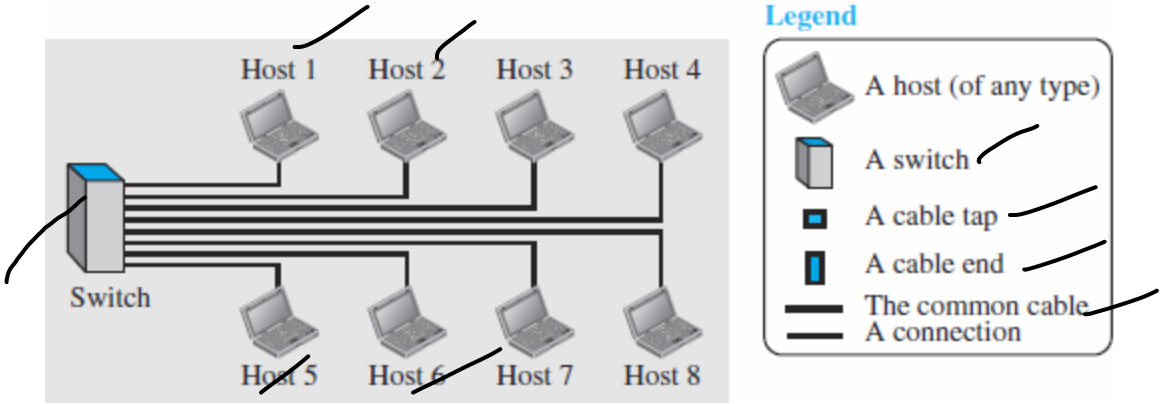
LAN with Common Cable (past)

- All hosts in a network were connected through a common cable
- A packet sent from one host to another was received by all hosts
- The intended recipient kept the packet; the others dropped the packet
- When LANs were used in isolation, they were designed to allow resources to be shared between the hosts



LAN with Switch

- Today, most LANs use a smart connecting switch
- Job of switch:
  - recognize the destination address of the packet
  - guide the packet to its destination without sending it to all other hosts.
    - The switch alleviates the traffic in the LAN
  - allows more than one pair to communicate with each other at the same time if there is no common source and destination among them
- LAN does not define the minimum or maximum number of hosts in it



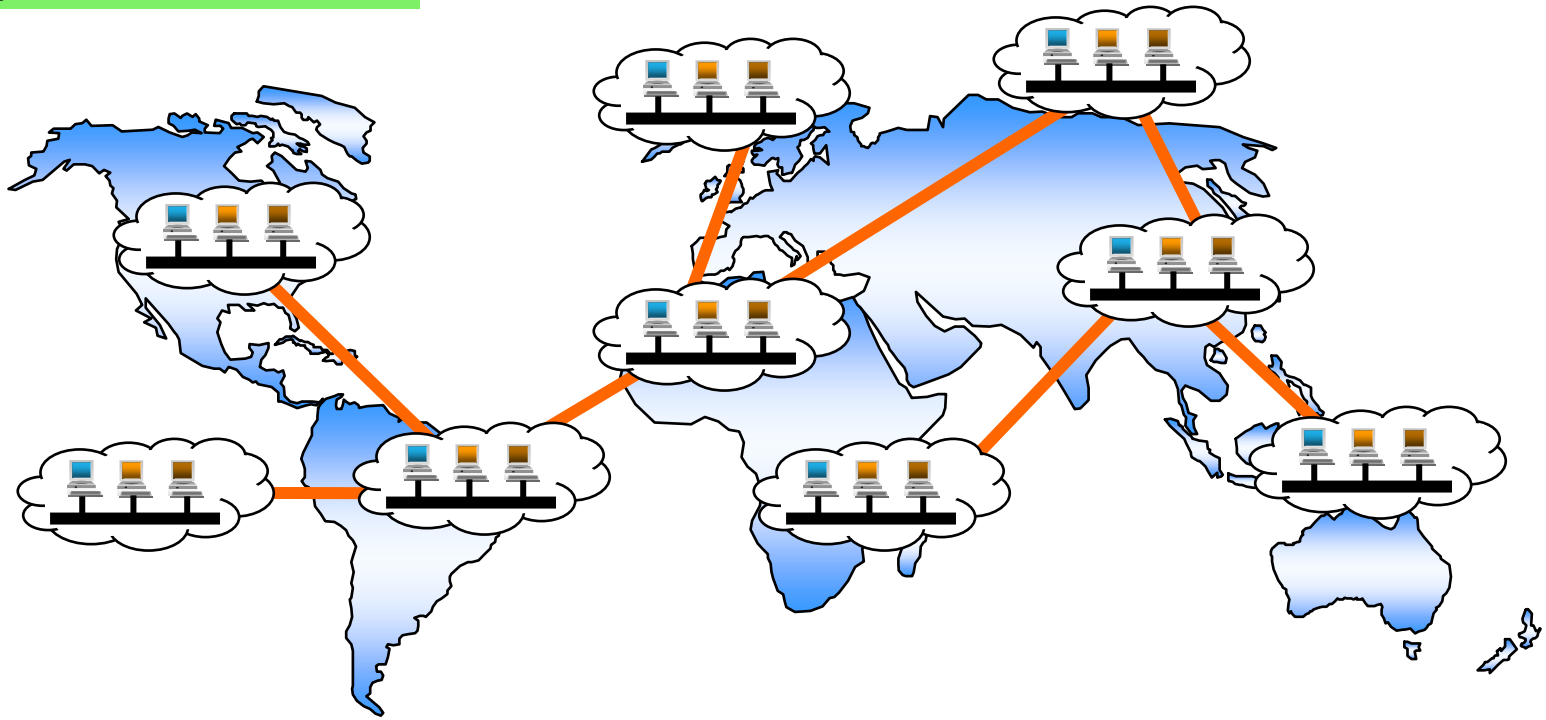
Wide Area Network

- an interconnection of devices capable of communication

LAN	WAN
Normally limited in size, spanning an office, a building, or a campus	a wider geographical span, spanning a town, a state, a country, or even the world
interconnects hosts	interconnects connecting devices such as switches, routers, or modems
normally privately owned by the organization that uses it	normally created and run by communication companies and leased by an organization that uses it.
Two types: Multipoint LAN and switched LAN	Two types: point-to-point WANs, switched WANs and internetwork

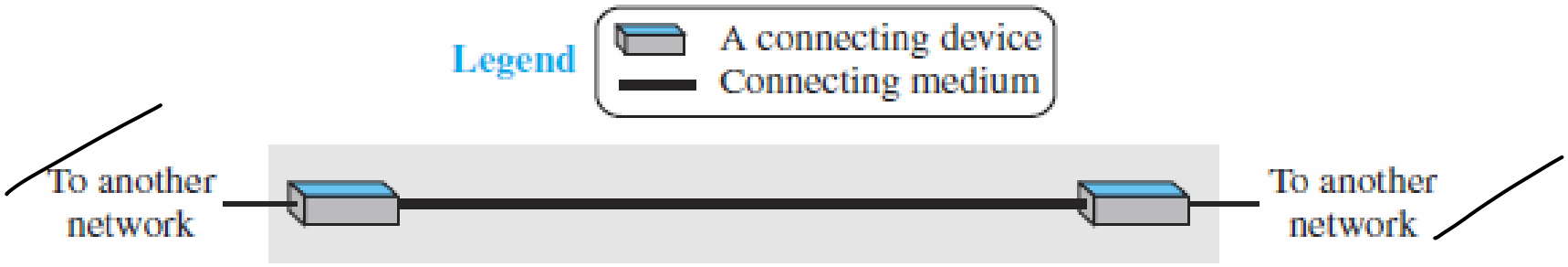
WAN

- A network that covers a relatively broad geographic area
- It often uses long-distance transmission facilities provided by public carriers



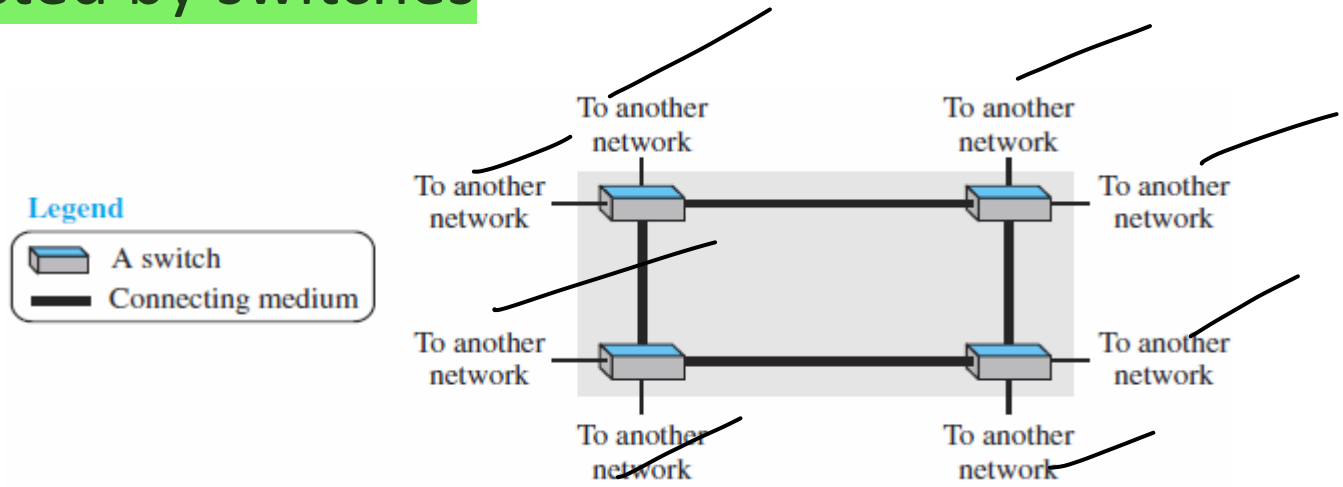
Point-to-Point WAN

- connects two communicating devices through a transmission media (cable or air)



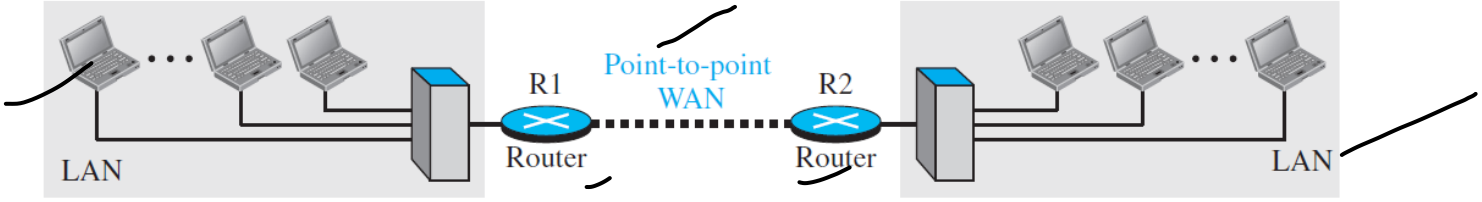
Switched WAN

- a network with more than two ends
- used in the backbone of global communication today
- a combination of several point-to-point WANs that are connected by switches



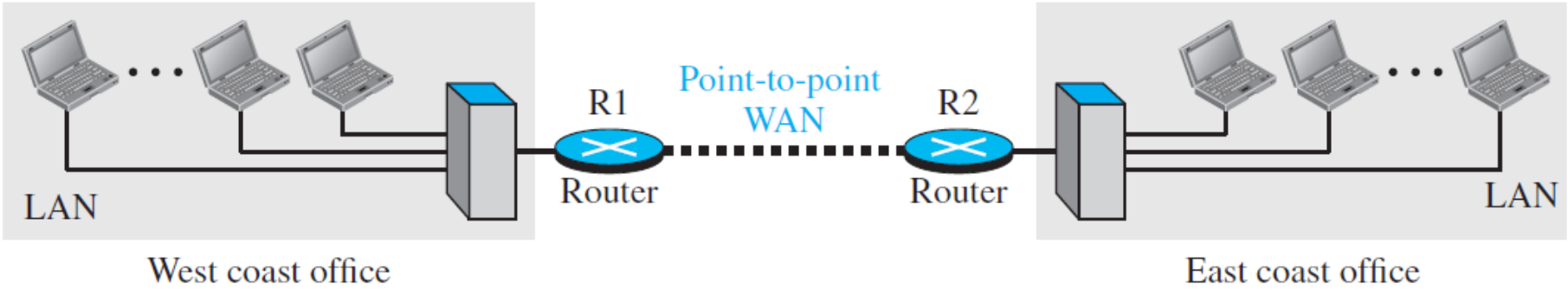
Internetwork

- Isolated LAN or WAN – rare
- LAN and WAN are connected to one another
- internetwork, or internet: connection between two or more networks
- Ex: organization with two offices, one on the east coast and the other on the west coast.
  - Each office has a LAN that allows all employees in the office to communicate with each other
  - Communication between employees at different offices possible: the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs



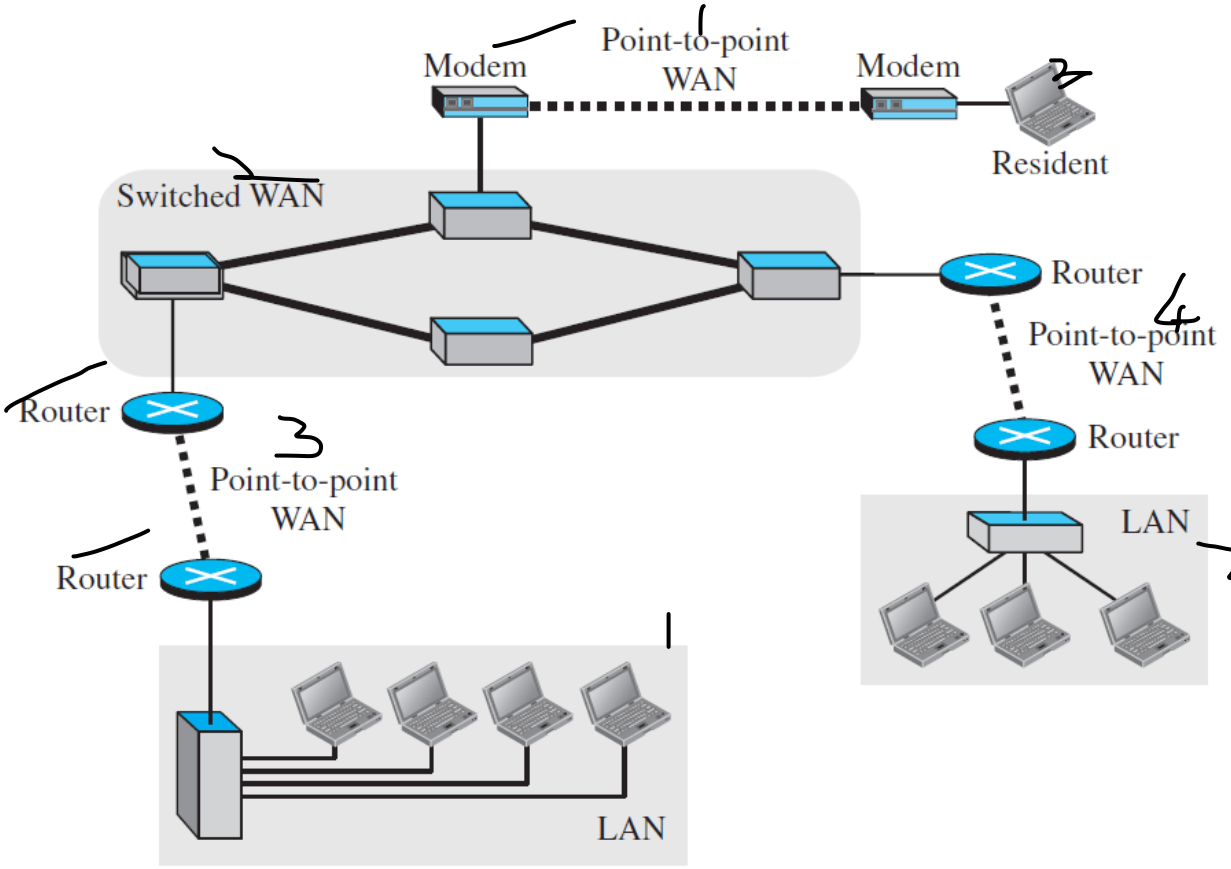
Internetwork …

- When a host in the west coast office sends a message to another host in the same office, the router blocks the message, but the switch directs the message to the destination.
- When a host on the west coast sends a message to a host on the east coast, router R1 routes the packet to router R2, and the packet reaches the destination



Internetwork …

- A heterogeneous network made of four WANs and three LANs
- One of the WANs is a switched WAN with four switches

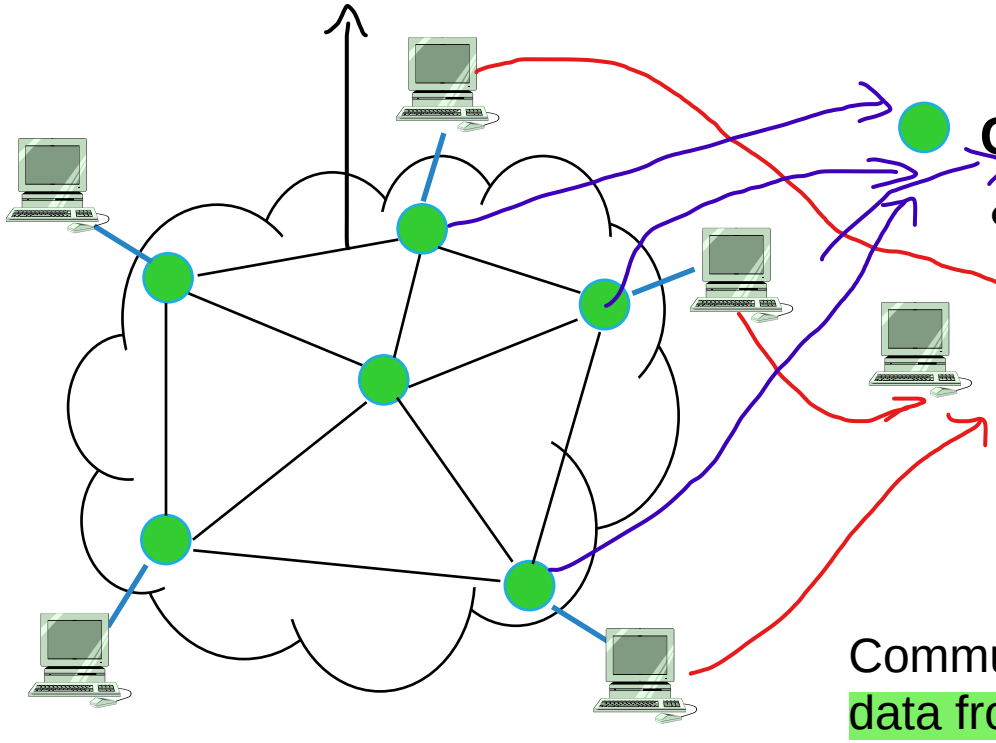


Switched Network

- switched network - a switch connects at least two links together
- A switch forward data from a network to another network when required
- The two most common types of switched networks
  - circuit-switched
  - packet-switched networks.

Switching

Switching/Communication Node Network



Communication Network Node

- provide switching facility (routing)

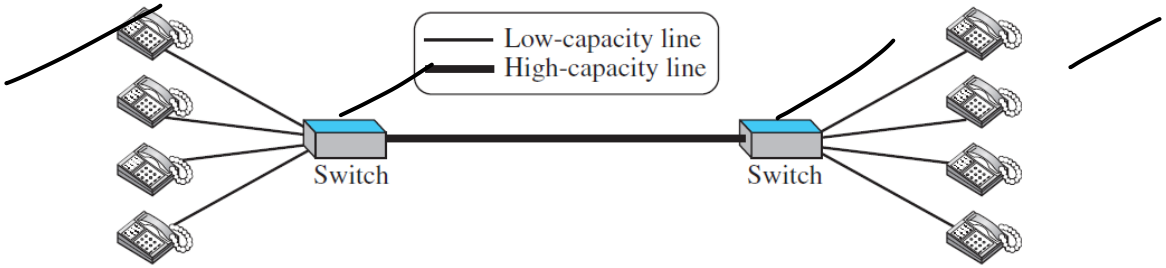
Network Station

- end node (source &destination)

Communications is achieved by transmitting data from source to destination through a network of switching nodes

Circuit-Switched Network

- a dedicated connection, called a circuit,
  - always available between the two end systems
  - the switch can only make it active or inactive
- simple switched network that connects four telephones to each end
- Telephone sets are shown instead of computers as an end system because circuit switching was very common in telephone networks in the past
  - part of the telephone network today is a packet-switched network.

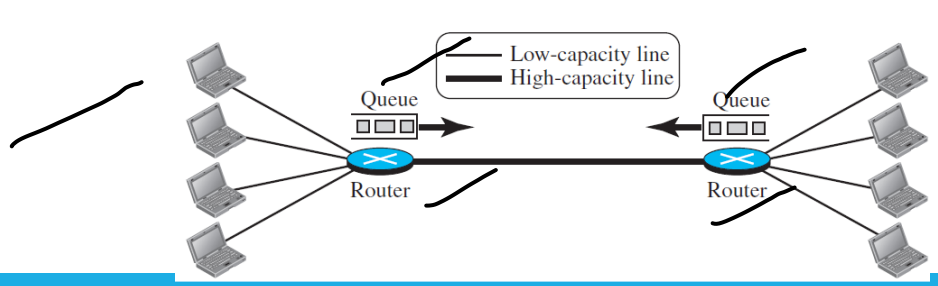


Circuit-Switched Network

- Case1: all telephone sets are busy
  - 4-people at one site are talking with 4-people at the other site
  - the capacity of the thick line is fully used
- Case 2: only one telephone set at one side is connected to a telephone set at the other side
  - only one-fourth of the capacity of the thick line is used
- Circuit-switched network is efficient only when it is working at its full capacity
  - most of the time, it is inefficient because it is working at partial capacity
- Capacity of the thick line four times the capacity of each voice line so the communication not fail when all telephone sets at one side want to be connected with all telephone sets at the other side

Packet-Switched Network

- computer network communication between the two ends - blocks of data called **packets**
- switches function for both storing and forwarding because a packet is an independent entity that can be stored and sent later
- small packet-switched network that connects four computers at one site to four computers at the other site.

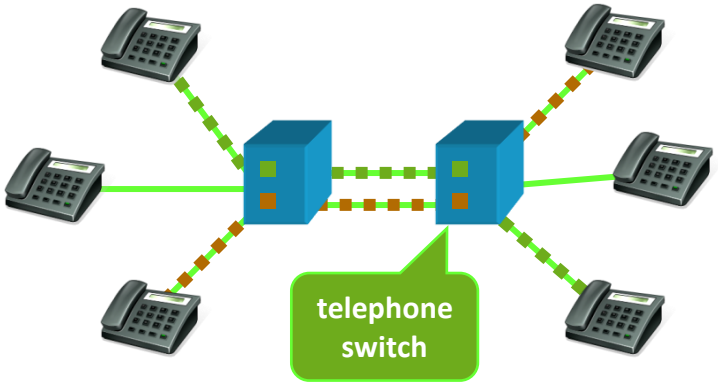


Packet-Switched Network

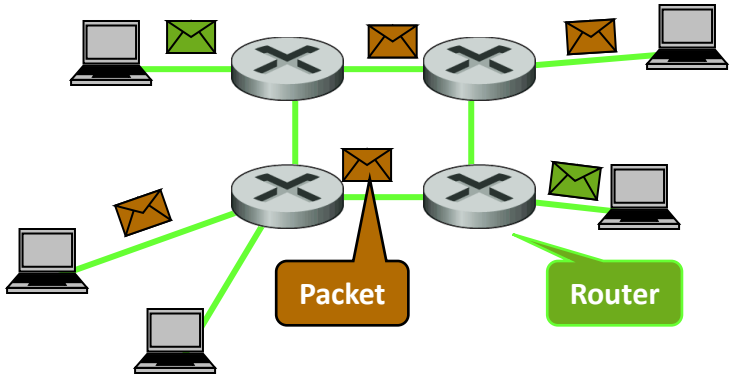
- A router in a packet-switched network has a queue that can store and forward the packet
- Capacity of the thick line is only twice the capacity of the data line connecting the computers to the routers
- If only two computers (one at each site) need to communicate with each other, there is no waiting for the packets.
- However, if packets arrive at one router when the thick line is already working at its full capacity, the packets should be stored and forwarded in the order they arrived
- packet-switched network is more efficient than a circuit switched network, but the packets may encounter some delays

Switching Technologies

- Circuit switching
  - A circuit is established for the two parties



- Packet switching
  - Data are put into packets, each stamped with source and destination addresses
  - Routers know where to forward packets



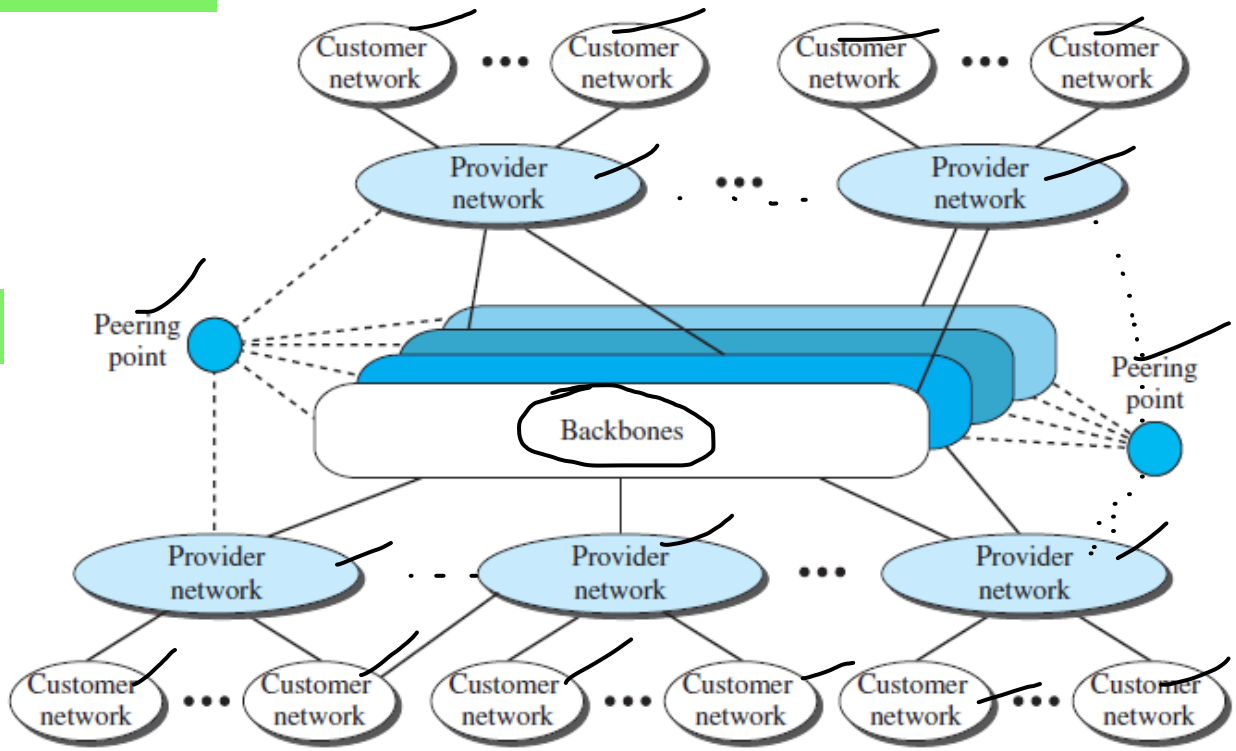
Key Differences

Aspect	Circuit Switching	Packet Switching
Data path	Dedicated single route	Different shared routes
Message	Bypass direct	Temporarily stored at each node
End point's status	Both must be ready	Only sender is ready
Utilization	Poor	Good
Data rate	Fixed	Varied
Prioritization	Not support	Support

Internet

- The largest internetwork in the world
- **Internet** (uppercase I), is composed of thousands of interconnected networks

Conceptual (not geographical) view of the Internet



Internet …

- Internet consists of several
  - Backbones (top level)
    - large networks owned by some communication companies such as Sprint, Verizon (MCI), AT&T, and NTT
    - connected through some complex switching systems, called *peering points*
  - provider networks (second level) - smaller networks
    - use the services of the backbones for a fee
    - connected to backbones and sometimes to other provider networks
  - Customer networks (the edge of the Internet)
    - actually use the services provided by the Internet
    - pay fees to provider networks for receiving services
- Backbones and provider networks are also called **Internet Service Providers (ISPs)**
  - Backbones = *international ISPs*
  - provider networks = *national or regional ISPs*

Accessing the Internet

- Internet today allows any user to become part of it
- User needs to be physically connected to an ISP
- The physical connection is normally done through a point-to-point WAN
- how user physically connected to an ISP?
  - Using Telephone Networks
  - Using Cable Networks
  - Using Wireless Networks
  - Direct Connection to the Internet

Using Telephone Networks

- Most telephone networks connected themselves to the Internet
- residences and small businesses connect to the Internet by changing the voice line between the residence or business and the telephone center to a point-to-point WAN using one of the way
  - Dial-up service
  - DSL Service

Using Telephone Networks

- Dial-up service
  - Add the telephone line to a modem that converts data to voice
  - The software installed on the computer dials the ISP and imitates making a telephone connection
  - very slow
  - when the line is used for Internet connection, it cannot be used for telephone (voice) connection
  - only useful for small residences
- ***DSL Service***
  - higher speed Internet services provided to residences or small businesses allows the line to be used simultaneously for voice and data communication

Using Cable Networks

- residents uses cable TV services instead of antennas to receive TV broadcasting over the last two
- Cable companies upgrading their cable networks by connecting to the Internet
- A residence or a small business with cable TV connection can be connected to the Internet
- Cable network provides a higher speed connection
  - the speed varies depending on the number of neighbors that use the same cable

Using Wireless Networks

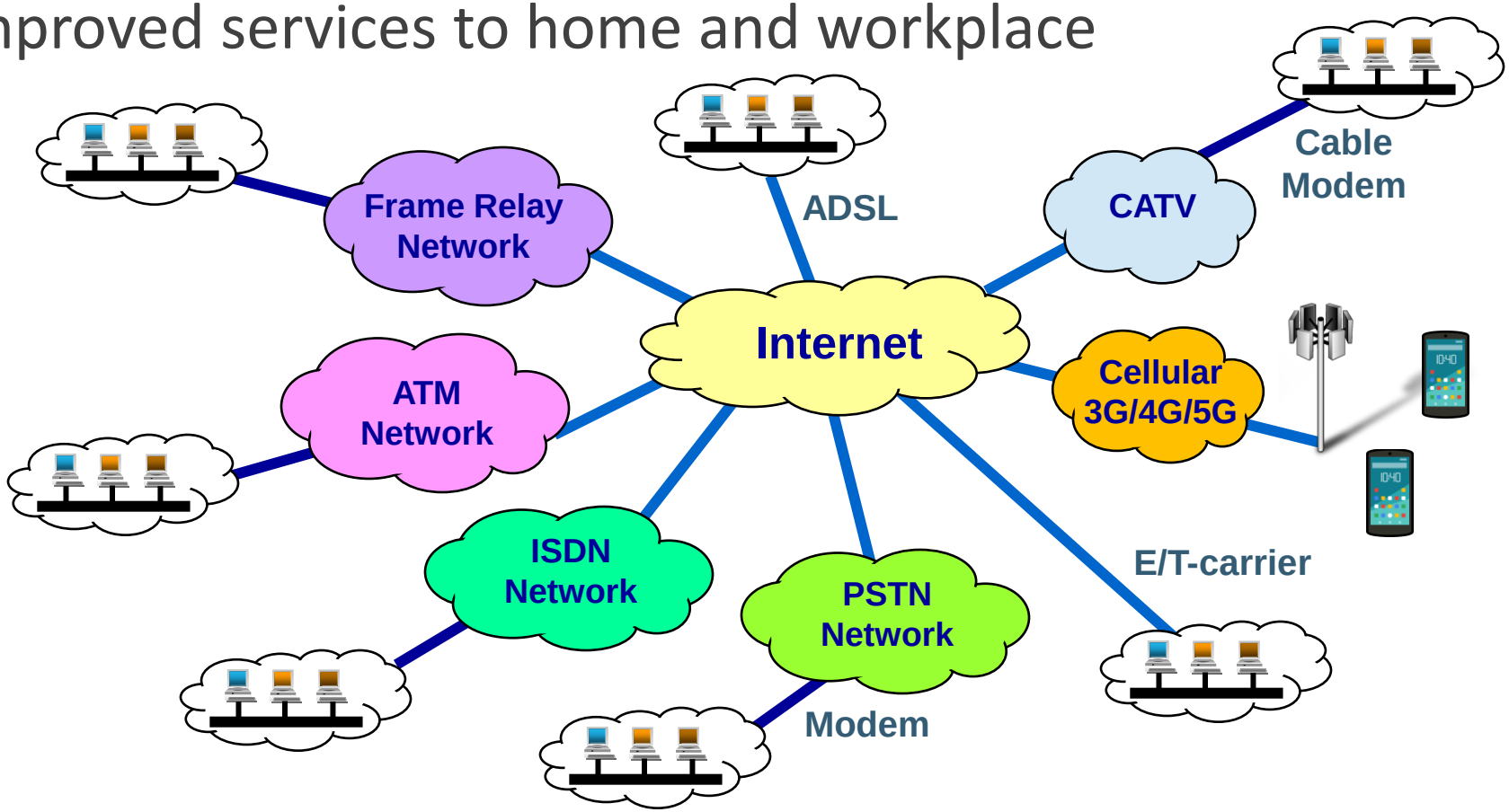
- Wireless connectivity has recently become increasingly popular
- A household or a small business can use a combination of wireless and wired connections to access the Internet
- With the growing wireless WAN access, a household or a small business can be connected to the Internet through a wireless WAN

Direct Connection to the Internet

- A large organization or a large corporation can itself become a local ISP and be connected to the Internet
  - the organization or the corporation leases a high-speed WAN from a carrier provider and connects itself to a regional ISP
    - For example, a large university with several campuses can create an internetwork and then connect the internetwork to the Internet

Technologies Around Us

- New generation of switching technologies provide improved services to home and workplace



Session-1 Summary

- Data communications are the transfer of data from one device to another via some form of transmission medium
- A data communications system must transmit data to the correct destination in an accurate and timely manner.
- The five components that make up a data communications system are the message, sender, receiver, medium, and protocol.
- Text, numbers, images, audio, and video are different forms of information.
- Data flow between two devices can occur in one of three ways: simplex, half-duplex, or full-duplex.
- A network is a set of communication devices connected by media links.
- In a point-to-point connection, two and only two devices are connected by a dedicated link.
- In a multipoint connection, three or more devices share a link.
- Topology refers to the physical or logical arrangement of a network.
- Devices may be arranged in a mesh, star, bus, or ring topology.

Summary…

- A network can be categorized as a local area network or a wide area network
- LAN is a data communication system within a building, plant, or campus, or between nearby buildings
- WAN is a data communication system spanning states, countries, or the whole world
- An internet is a network of networks

Educational tool for computer network

- many programs and utilities available for Windows and UNIX operating systems that allow us to sniff, capture, trace, and analyze packets that are exchanged between our computer and the Internet
- Some of these
  - *Wireshark* and *Ping-Plotter,* have graphical user interface (GUI)
  - others, such as *traceroute, nslookup, dig, ipconfig,* and *ifconfig,* are network administration command-line utilities

Homework

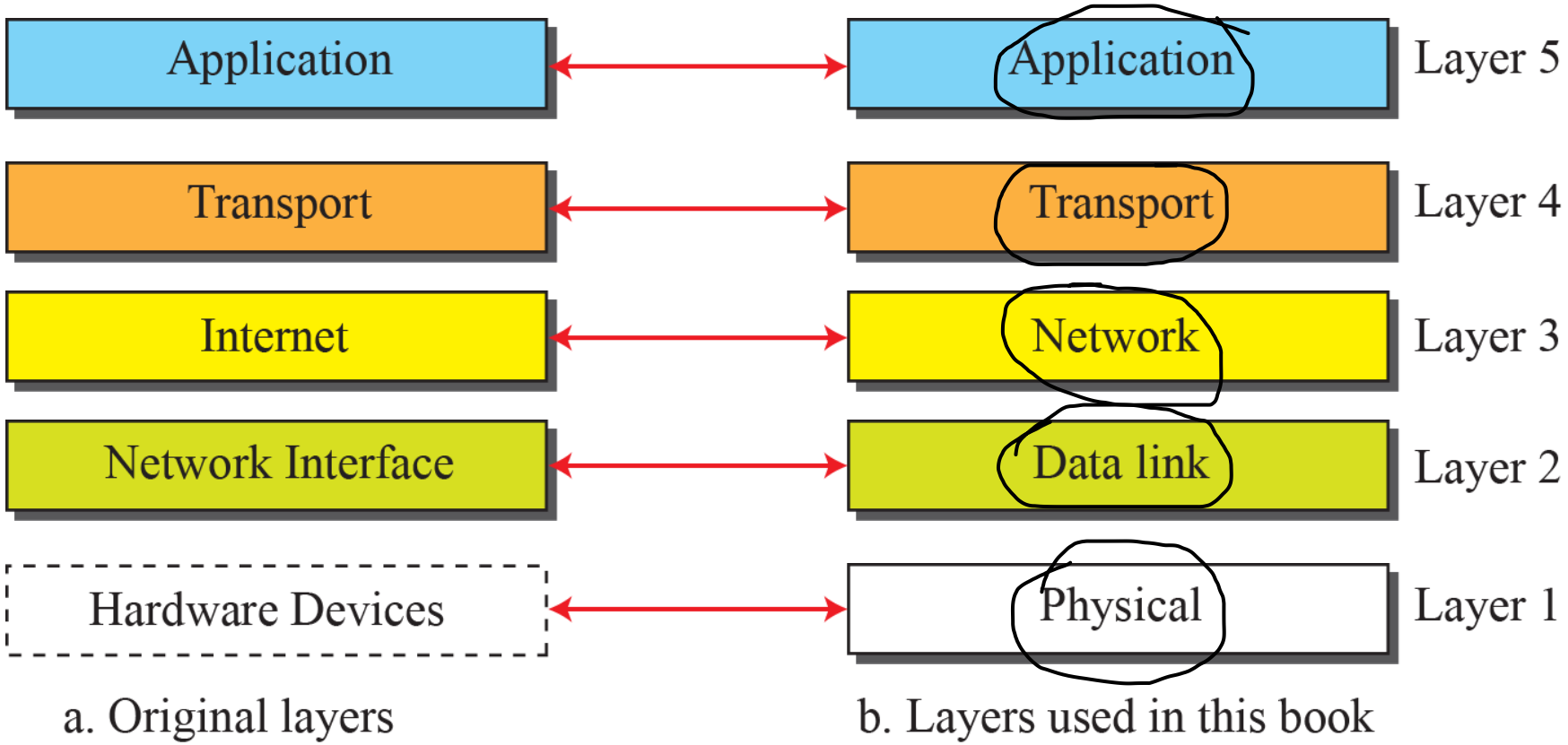
- Prepare the answers for Chapter 1 Section 1.7
  - Questions from Q1-1 to Q1-15
  - Problems rom P1-1 to P1-10

TCP/IP Protocol Suite

- Transmission Control Protocol/Internet Protocol
- a set of protocols organized in different layers
- used in the Internet today
- a hierarchical protocol: means that each upper level protocol is supported by the services provided by one or more lower level protocols
- A protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively.

ATNDP

Layers in TCP/IP Protocol Suite



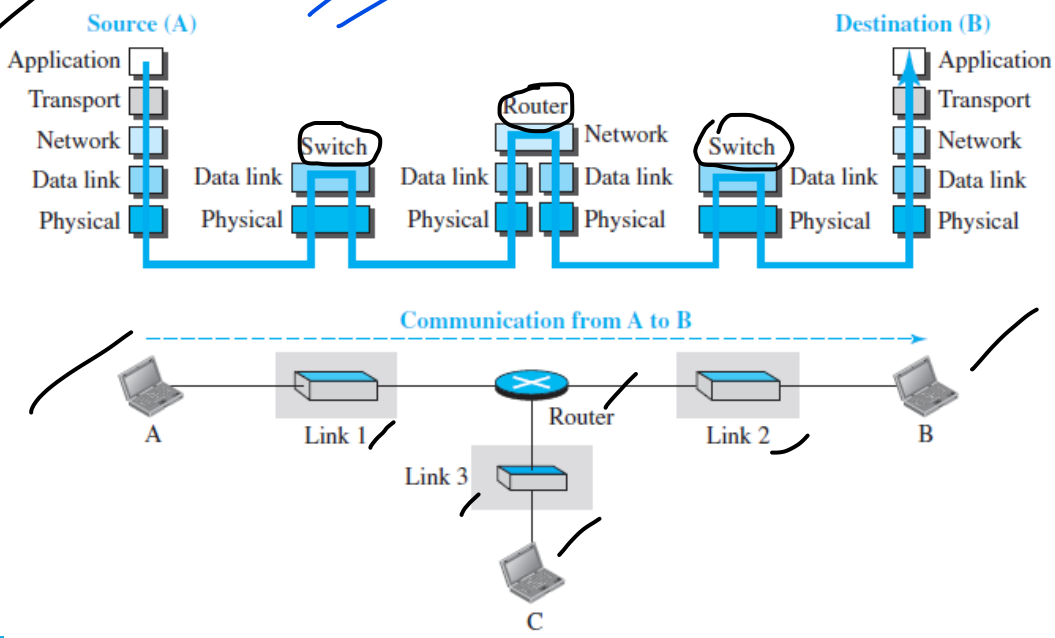
The original TCP/IP protocol suite was defined as four software layers built upon the hardware.

Today, TCP/IP is thought of as a five-layer model.

Layered Architecture

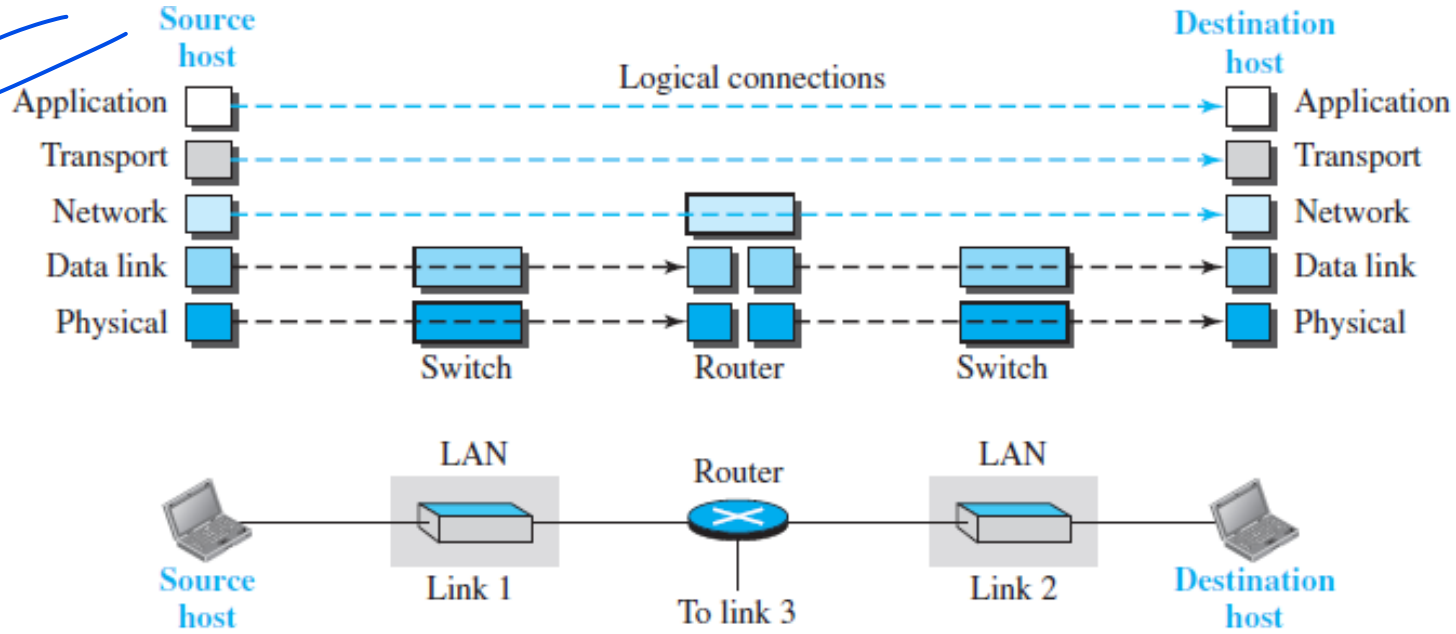
- To show how the layers in the TCP/IP protocol works, we assume that we have following internet
  - use the suite in a small internet made up of three LANs (links) - each with a link-layer switch.
  - links are connected by one router
- router -> 3-layers NDP
- link-layer switch ->2-layers DP

Nodes have all 5 layers -> ATNDP



Layers in the TCP/IP Protocol Suite

- To understand the duty of each layer, we need to think about the logical connections between layers.

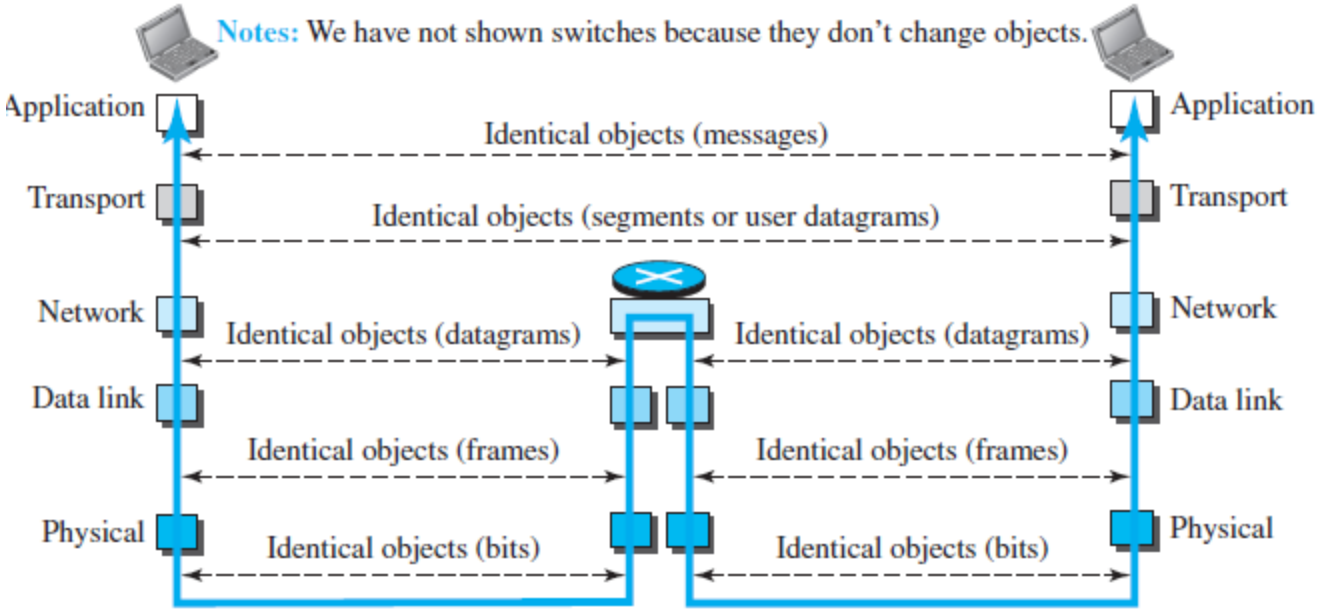


The duty of the application, transport, and network layers is end-to-end.
Their domain of duty is the internet.

The duty of the data-link and physical layers is hop-to-hop, hop is a host or router.
Their domain of duty is the link.

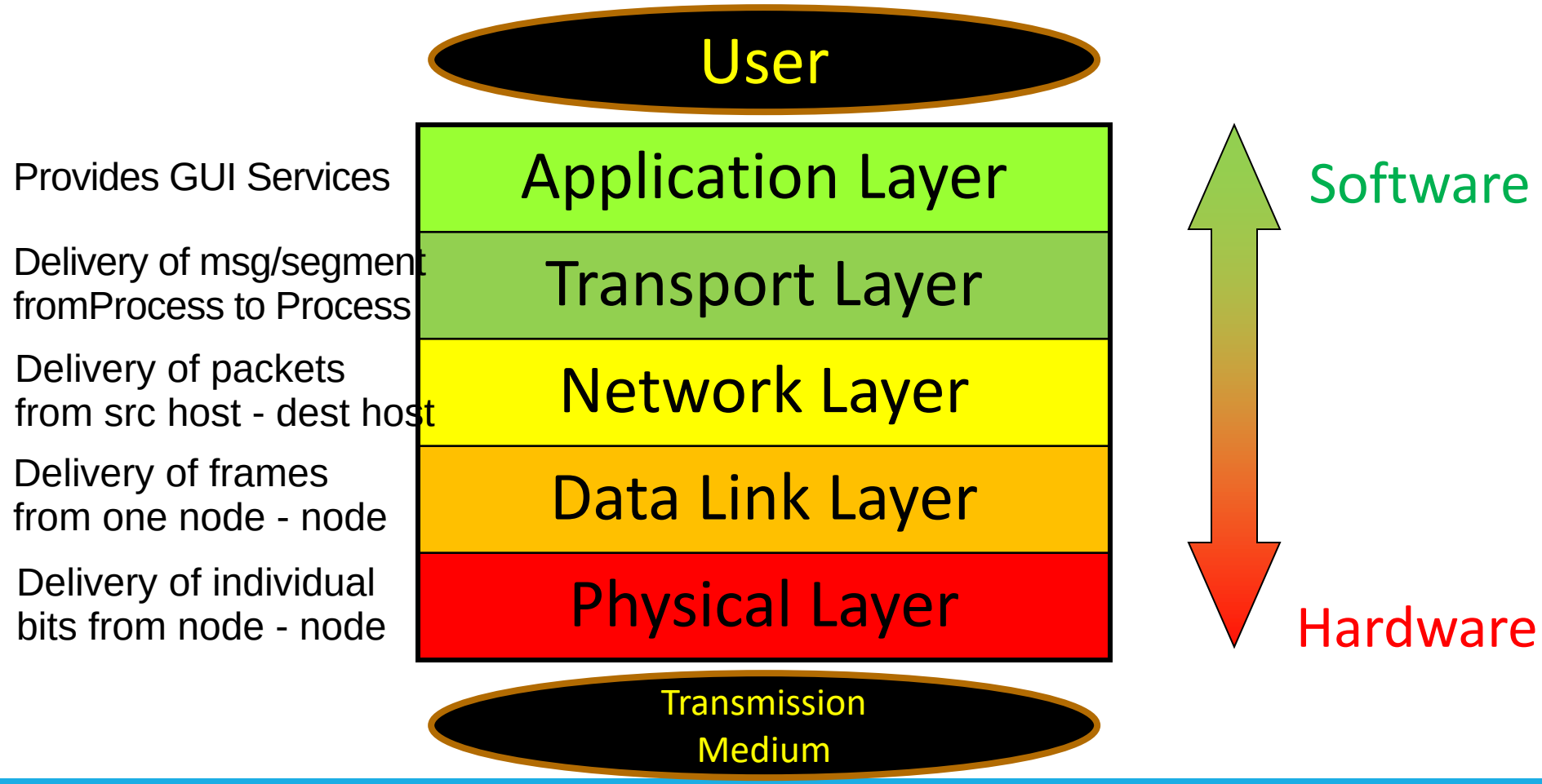
Identical objects in TCP/IP protocol suite

- the second principle: identical objects below each layer related to each device.



Internet Layer Model

- The Internet Protocol Stack

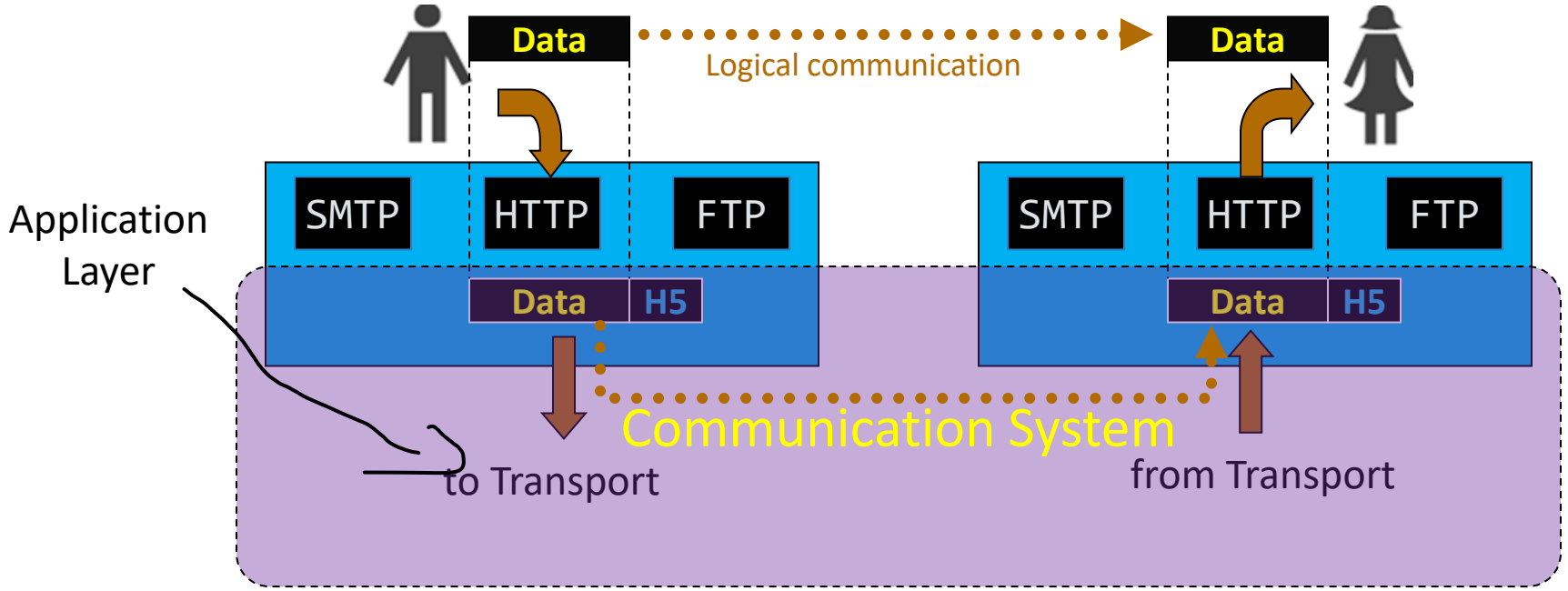


Application Layer

Addressing: Domain Name
Protocols: HTTP, SMTP, FTP

Responsible for providing services to the user

- The only layer to interact with user



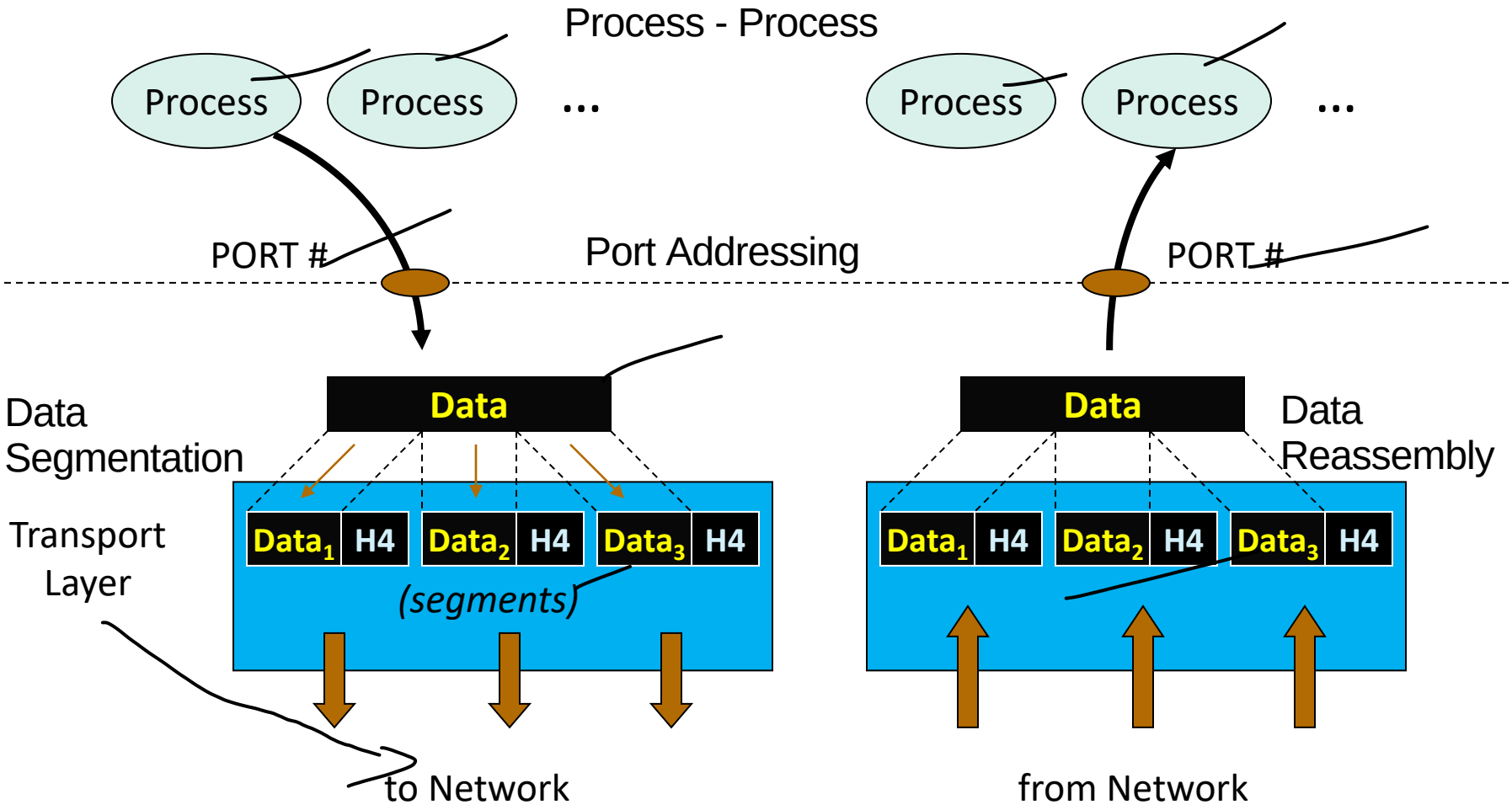
Transport Layer

Addressing: Port
Protocols: UDP TCP

Responsible for delivery of a message from one process to another

- Duties/services
  - Port addressing
  - Segmentation and reassembly
  - Connection control
  - Flow control (end-to-end)
  - Error control (end-to-end)

Transport Layer



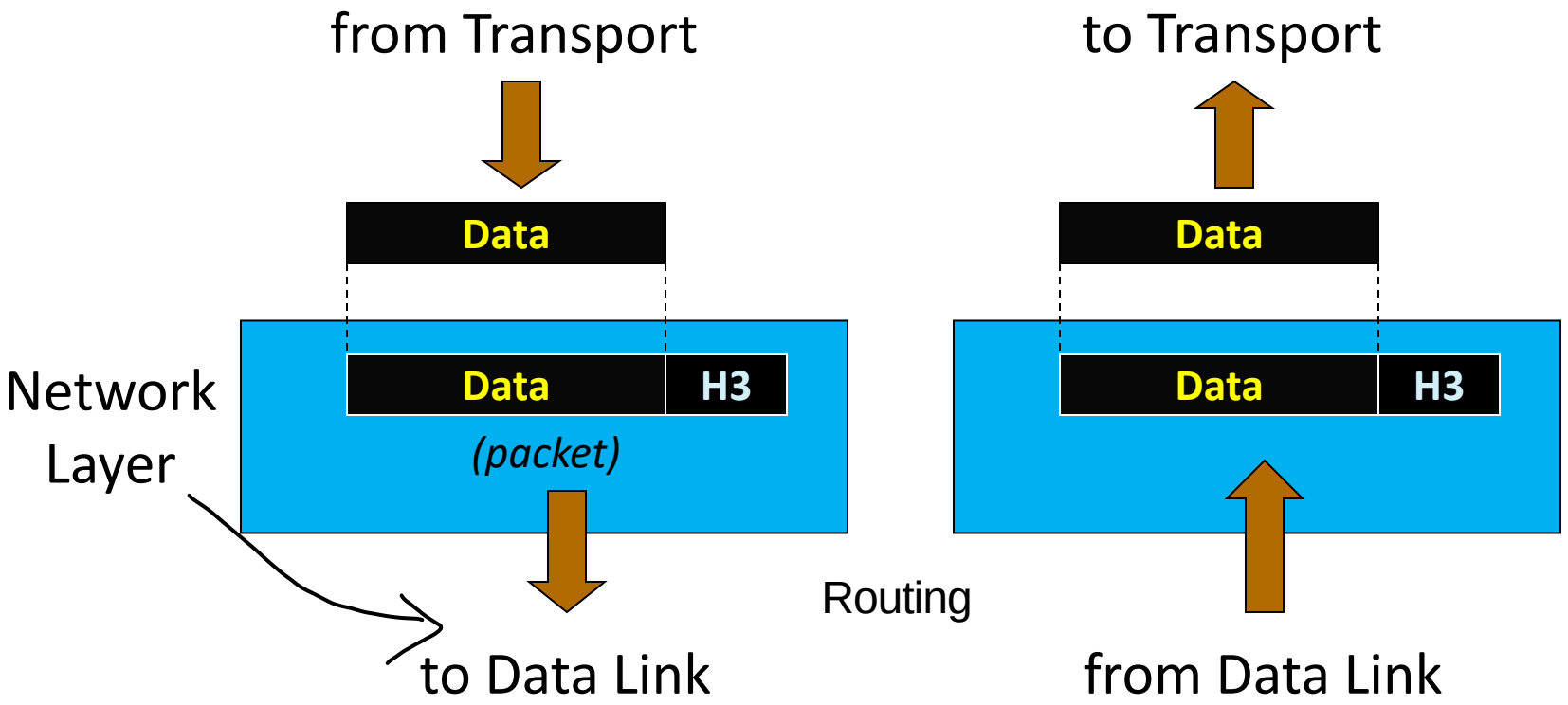
Network Layer

Addressing: IP/Logical
Protocols: IPv4, IPv6, etc

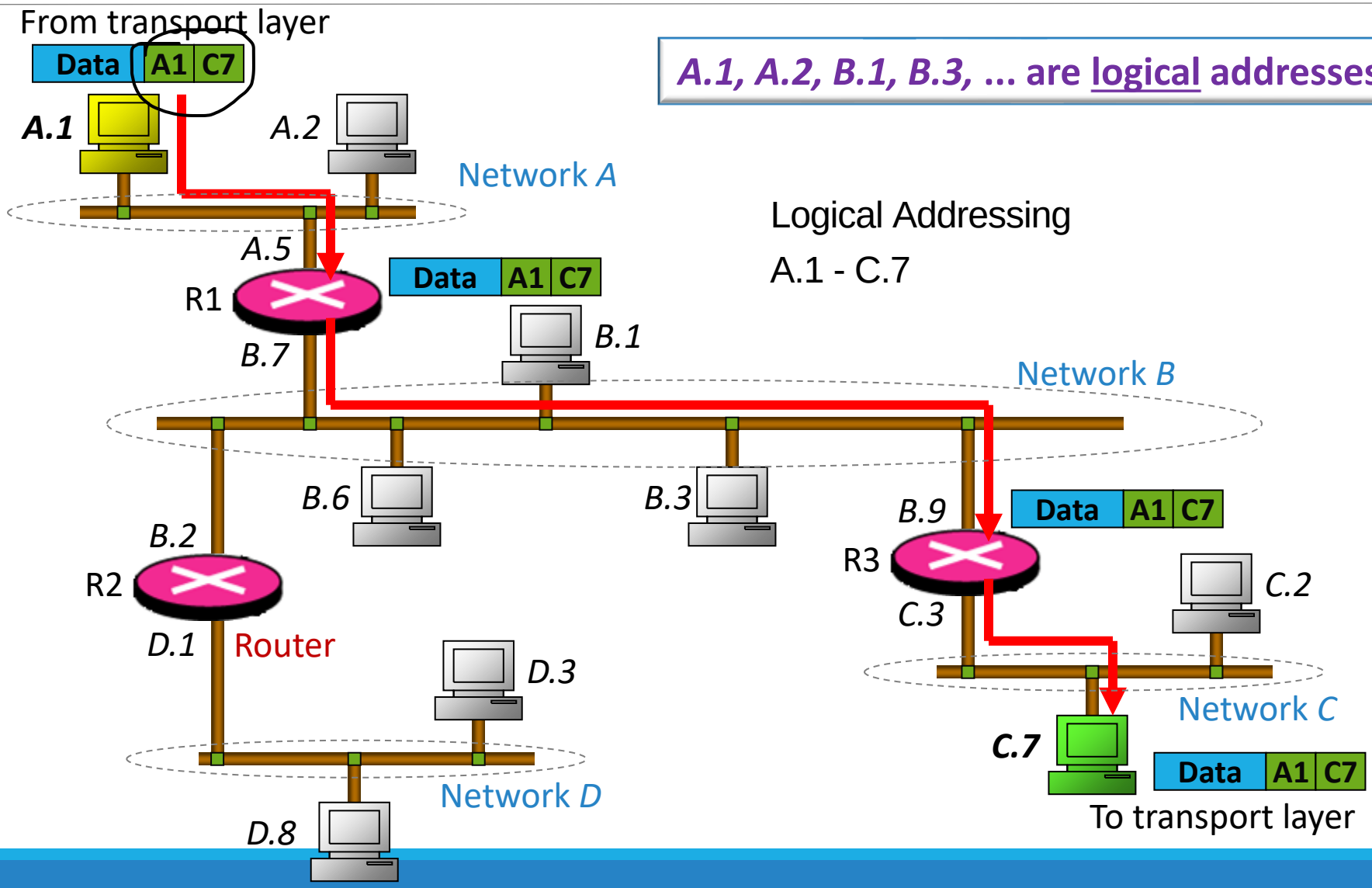
Responsible for the delivery of packets from the original source to the destination

- Duties/services
  - Logical addressing
  - Routing

Network Layer



Network Layer Operation



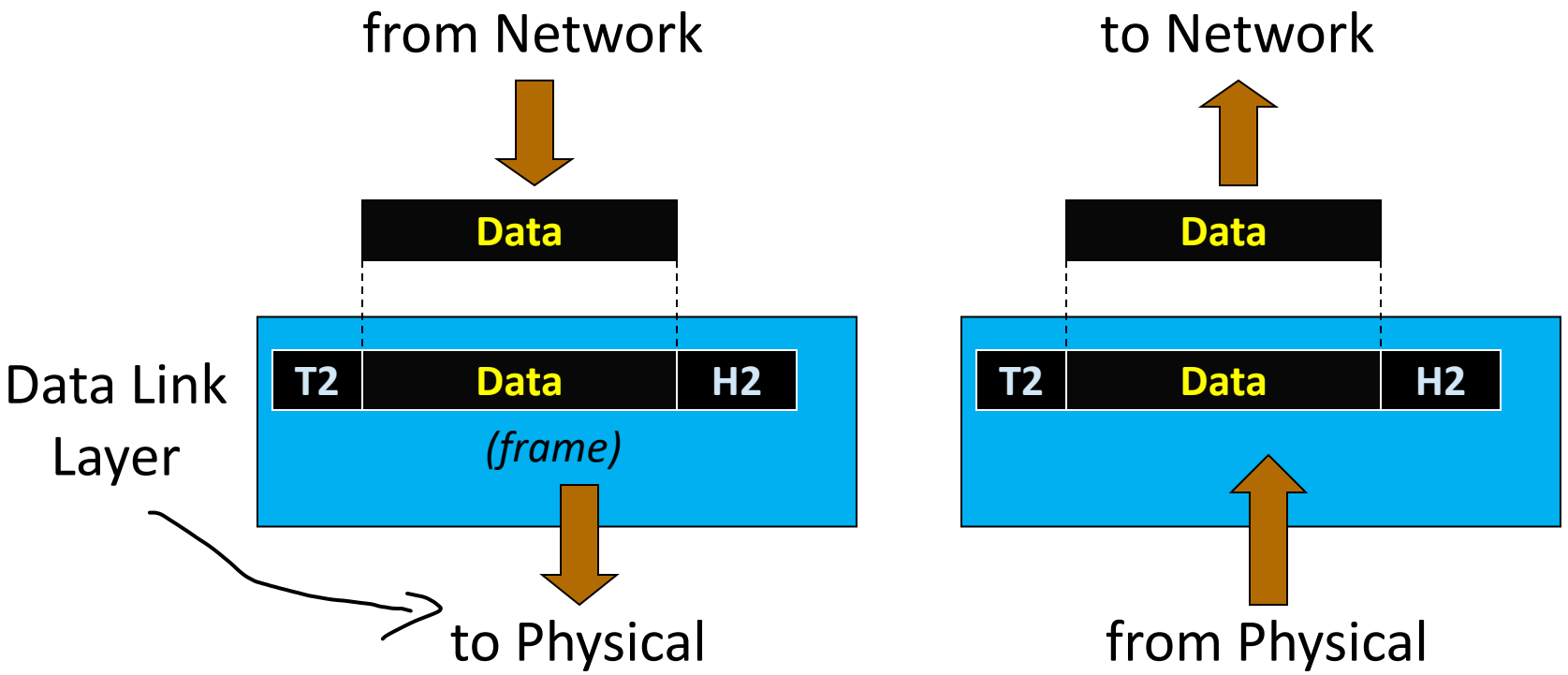
Data Link Layer

Addressing: Physical/MAC Address
Protocols: CSMA, CSMA/CD/CA, CRC, Block Coding

Responsible for transmitting frames from one node to the next

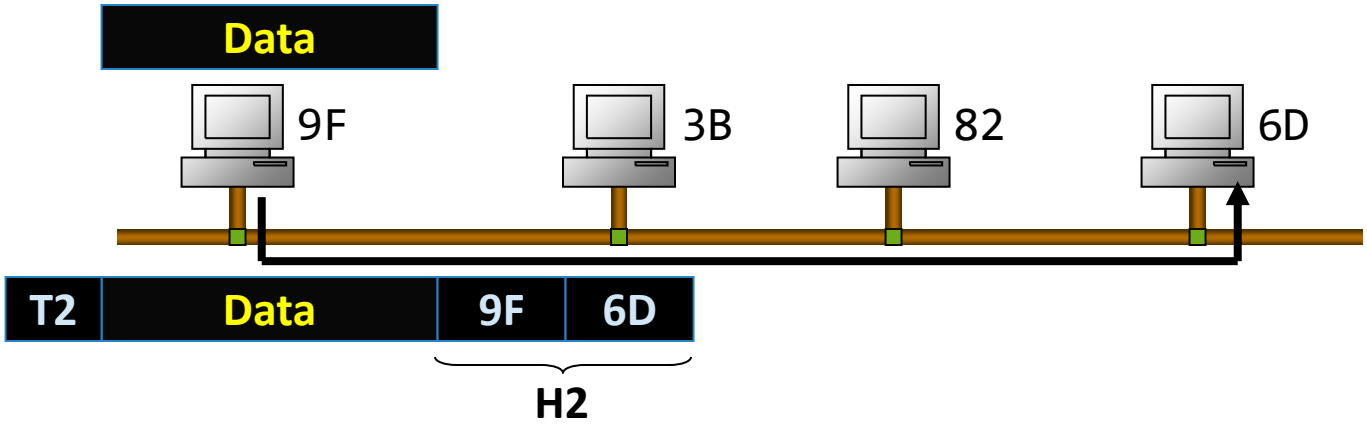
- Duties/services
  - Framing
  - Physical addressing
  - Flow control (hop-to-hop)
  - Error control (hop-to-hop)
  - Access control

Data Link Layer

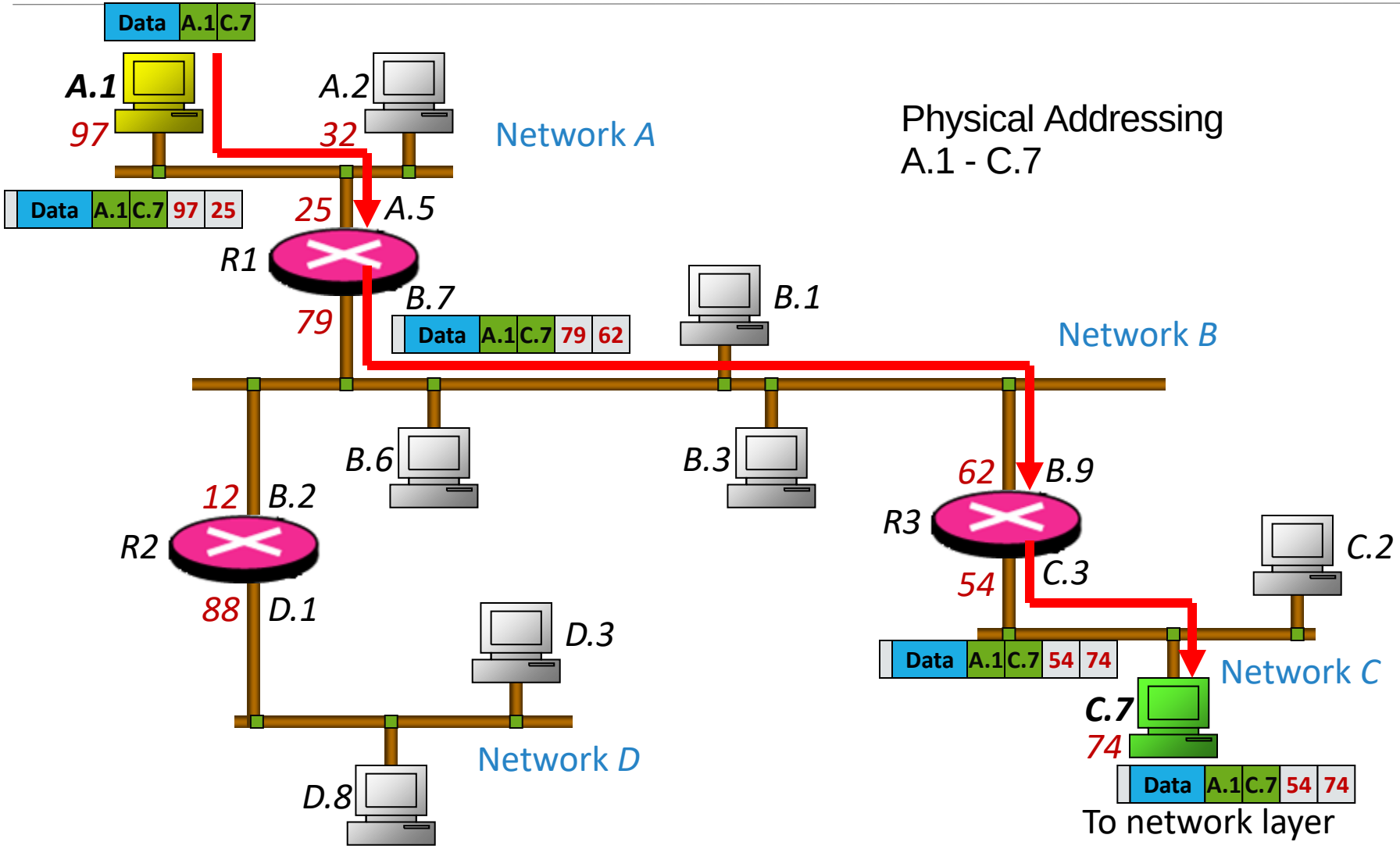


Data Link Layer

9F, 3B, 82, 6D, … are physical addresses



Data Link Layer Operation

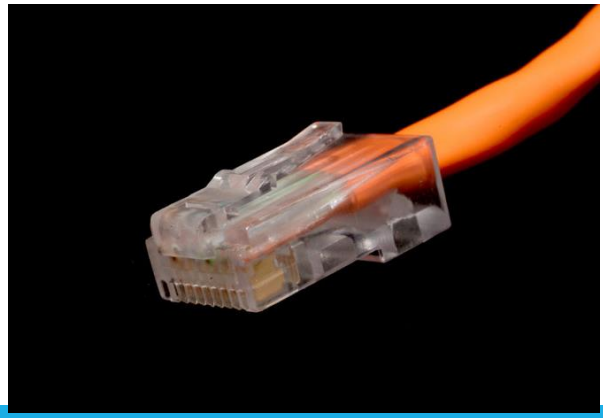


Physical Layer

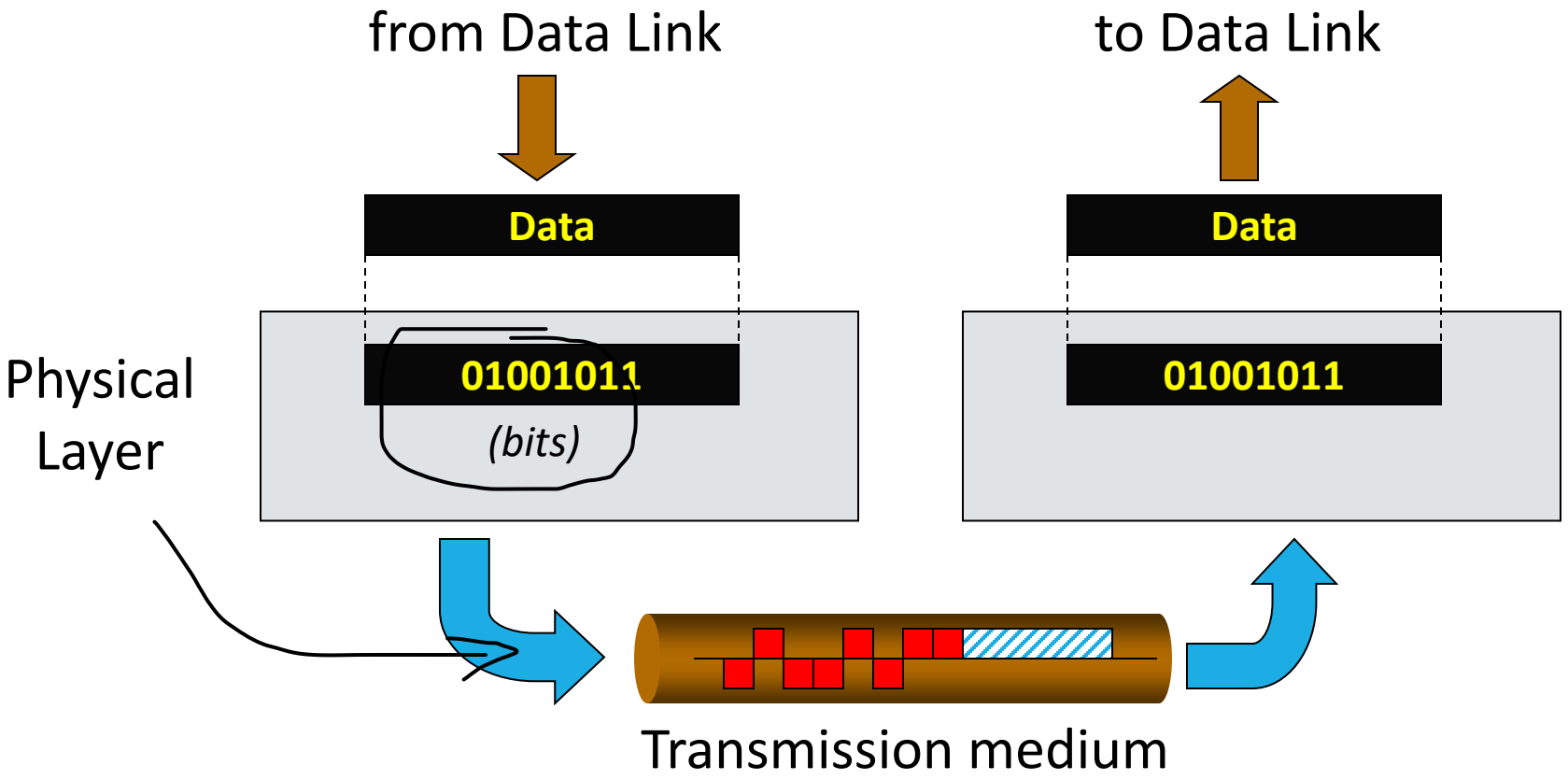
Addressing: Ethernet Address
Protocols: Ethernet, IEEE 801.22, etc

Responsible for transmitting individual bits from one node to the next

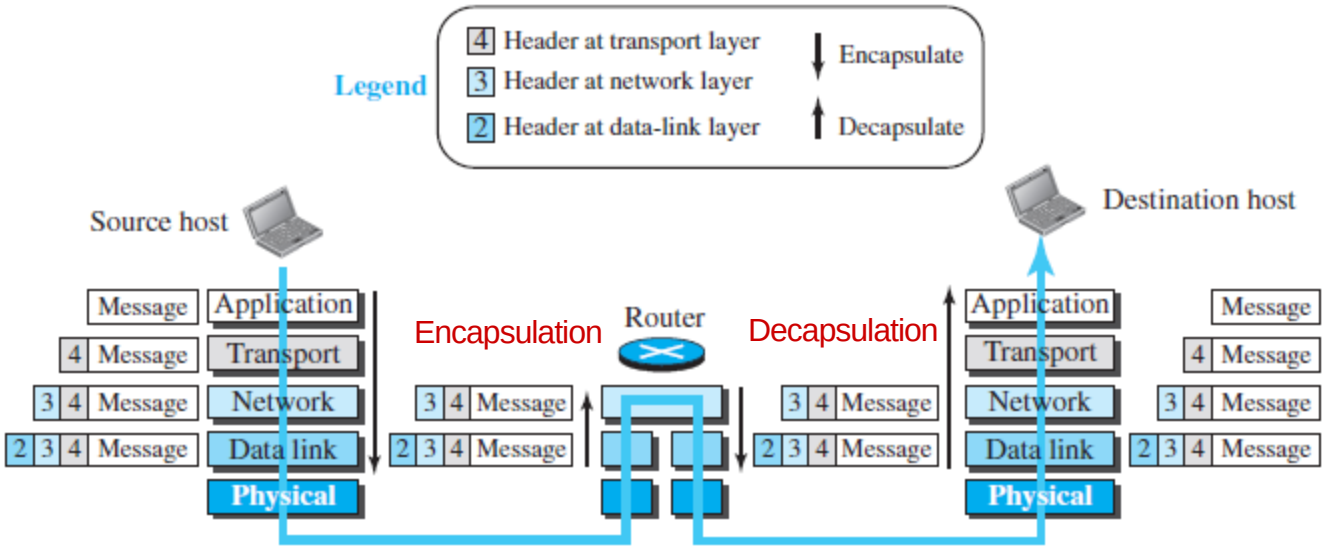
- Duties/services
  - Physical characteristics of interfaces and media
  - Representation of bits
  - Data rate (transmission rate)
  - Synchronization of bits



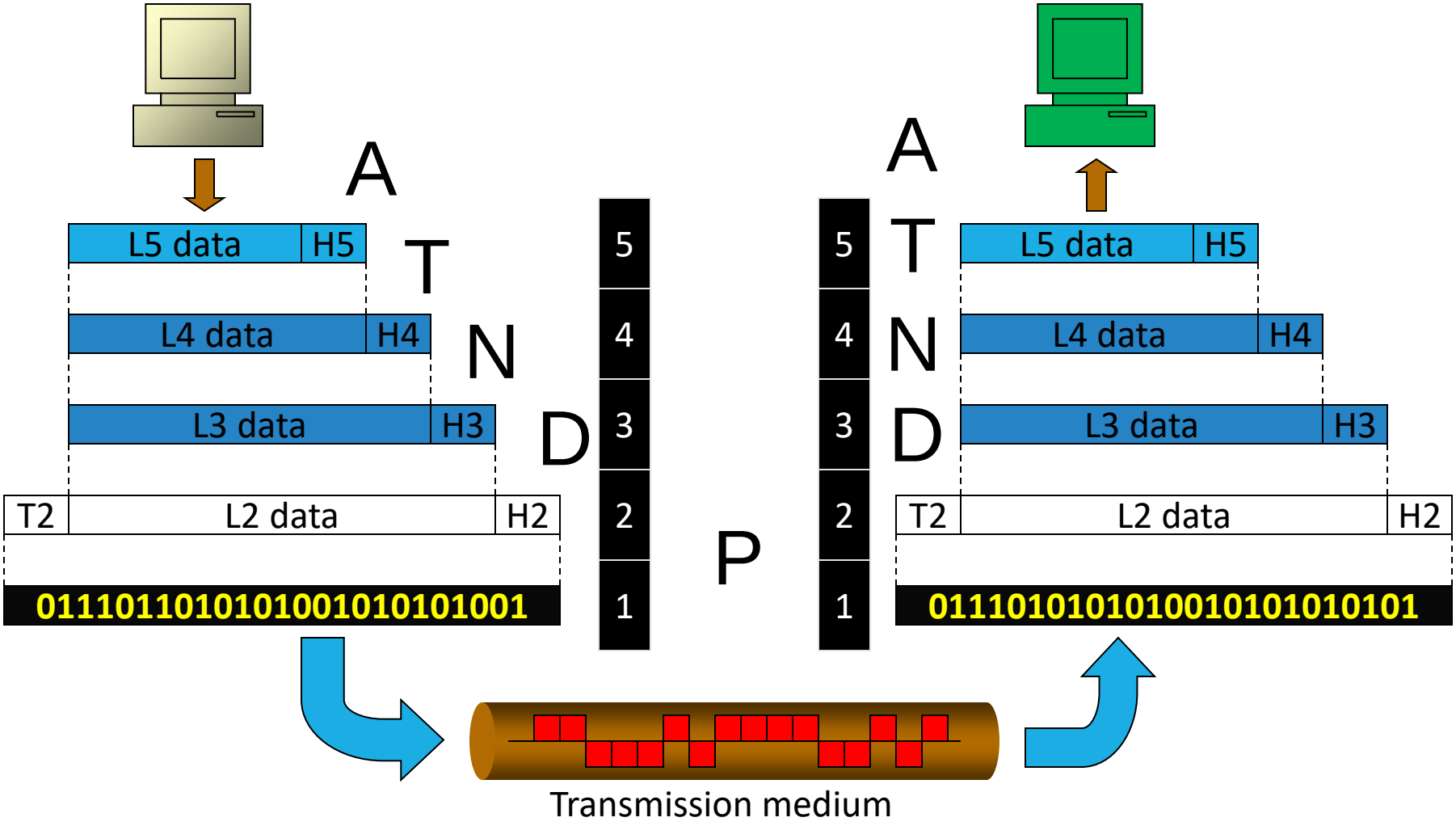
Physical Layer



Encapsulation and Decapsulation

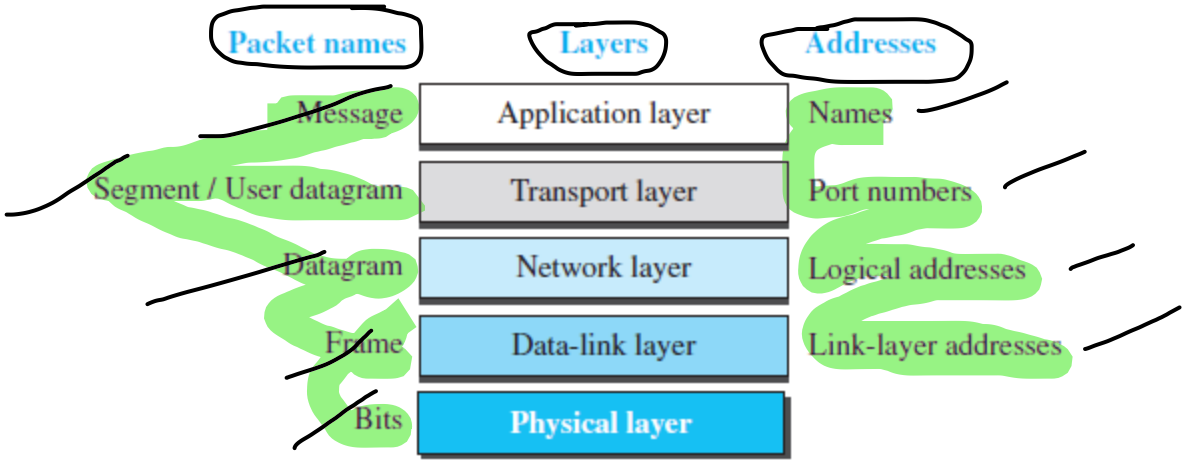


The Big Picture

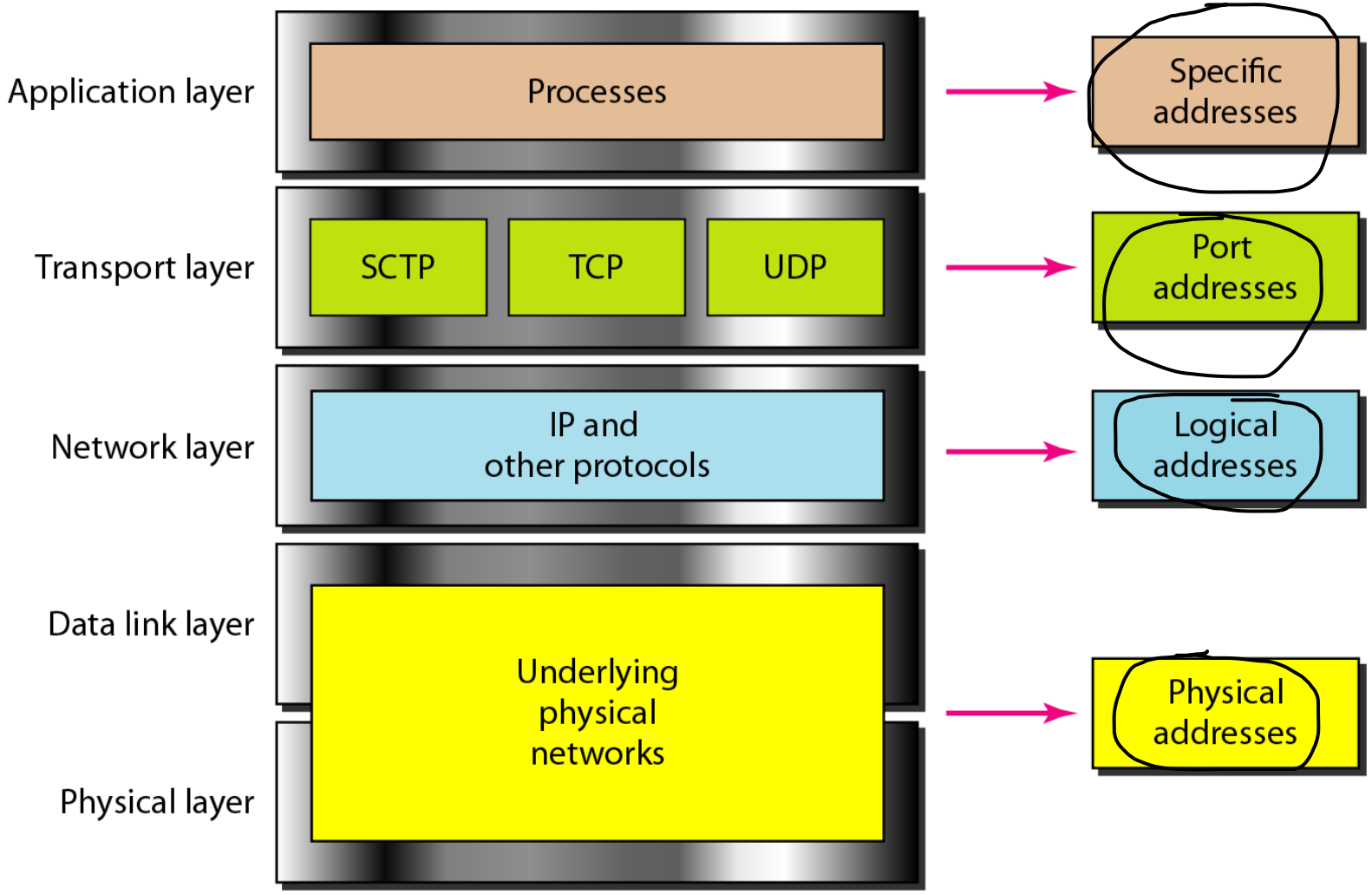


Addressing

- Any communication that involves two parties needs two addresses: source address and destination address.
- Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address.

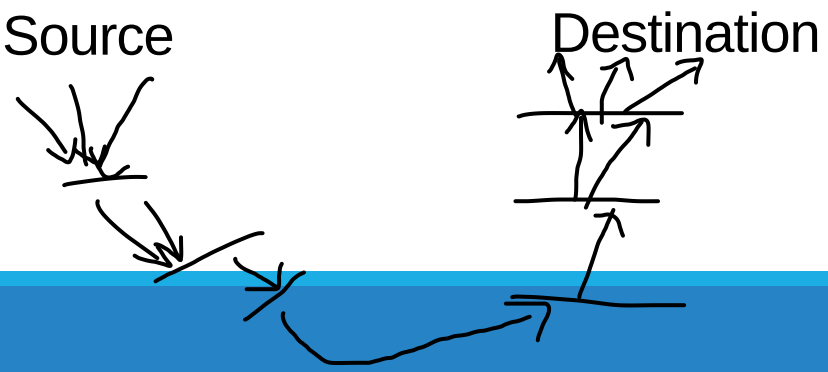


Layers and Addresses



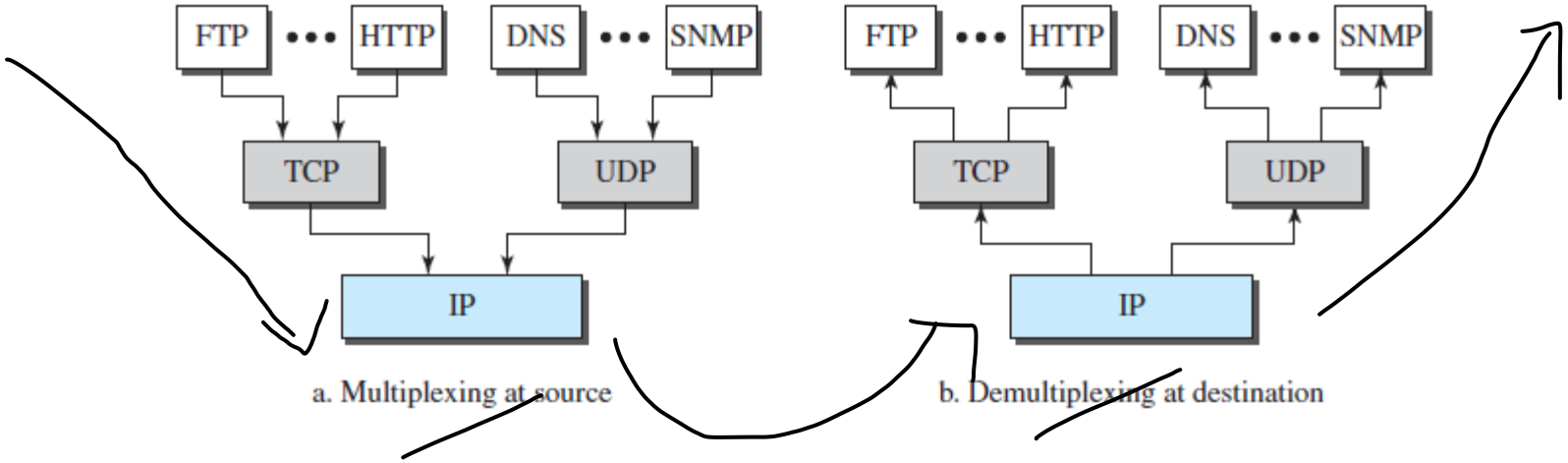
Multiplexing and Demultiplexing

- Since the TCP/IP protocol suite uses several protocols at some layers, we can say that we have multiplexing at the source and demultiplexing at the destination.
- Multiplexing in this case means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time).
- Demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time



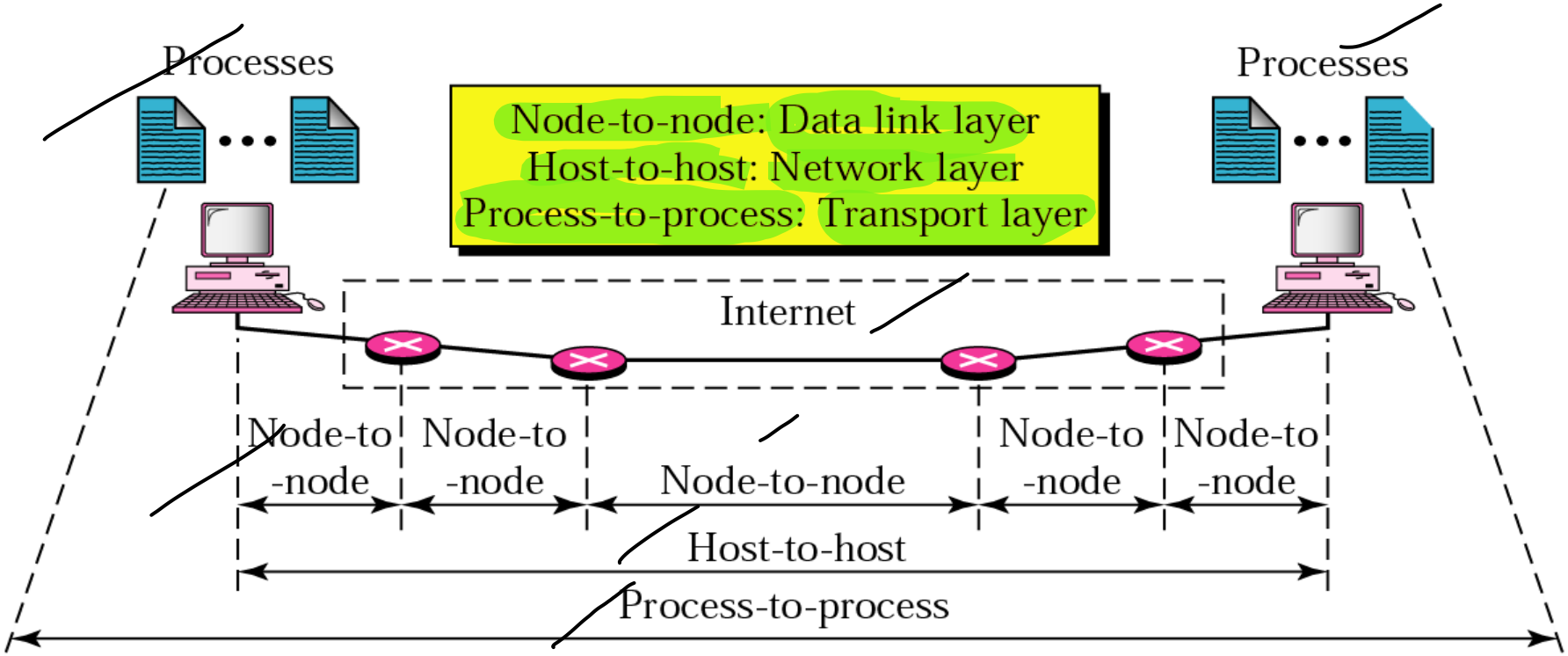
Multiplexing and Demultiplexing...

- To be able to multiplex and demultiplex, a protocol needs to have a field in its header to identify to which protocol the encapsulated packets belong.

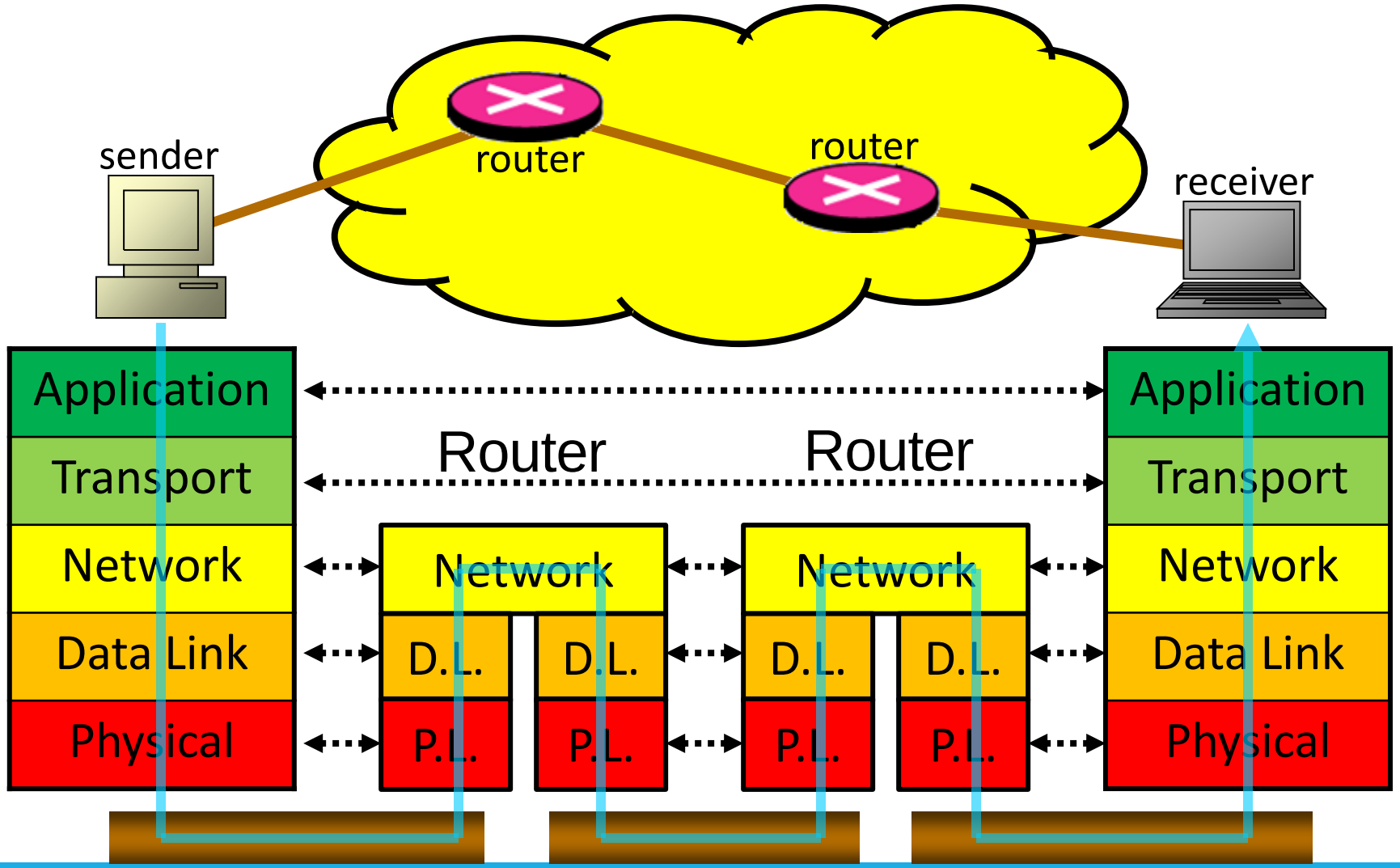


Ex: At the transport layer, either UDP or TCP can accept a message from several application-layer protocols.

Layers and Types of Data Delivery



Internet Model



Protocol Suites

- A set of protocols must be constructed
  - to ensure that the resulting communication system is complete and efficient
- Each protocol should handle a part of communication not handled by other protocols
- How can we guarantee that protocols work well together?
  - Instead of creating each protocol in isolation, protocols are designed in complete, cooperative sets called suites or families

Internet Protocol Suite

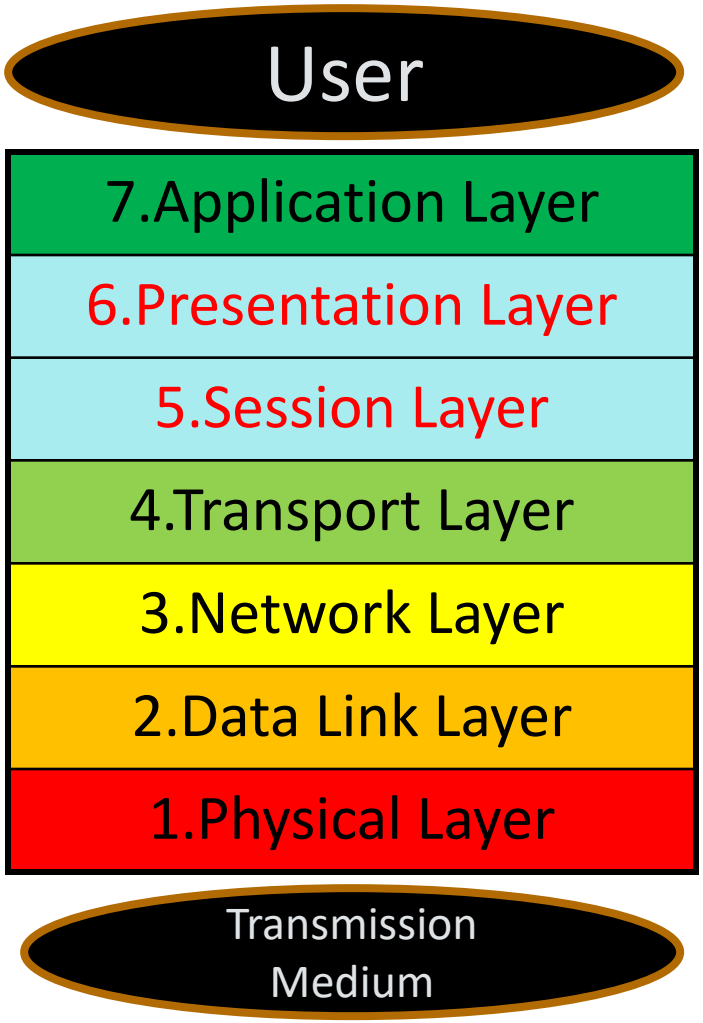
Layer	Protocols
Application	HTTP, FTP, Telnet, SMTP, ...
Transport	TCP, UDP, SCTP, ...
Network	IP (IPv4), IPv6, ICMP, IGMP, ...
Data Link	Ethernet, Wi-Fi, PPP, ...
Physical	RS-232, DSL, 10Base-T, ...

OSI MODEL

- A word we hear all the time when we talk about the Internet is protocol. A protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively. When communication is simple, we may need only one simple protocol; when the communication is complex, we need a protocol at each layer, or protocol layering.

OSI Model

Open Systems Interconnections



- OSI – ***O****pen* ***S****ystems* ***I****nterconnection*
- Developed by the ***I****nternational* ***S****tandards* ***O****rganizations* (ISO)
- Two additional layers
  - Presentation layer
  - Session layer

Session Layer

Responsible for establishing, managing and terminating connections between applications

- Duties/services
  - Interaction management
    - ⇨ Simplex, half-duplex, full-duplex
  - Session recovery

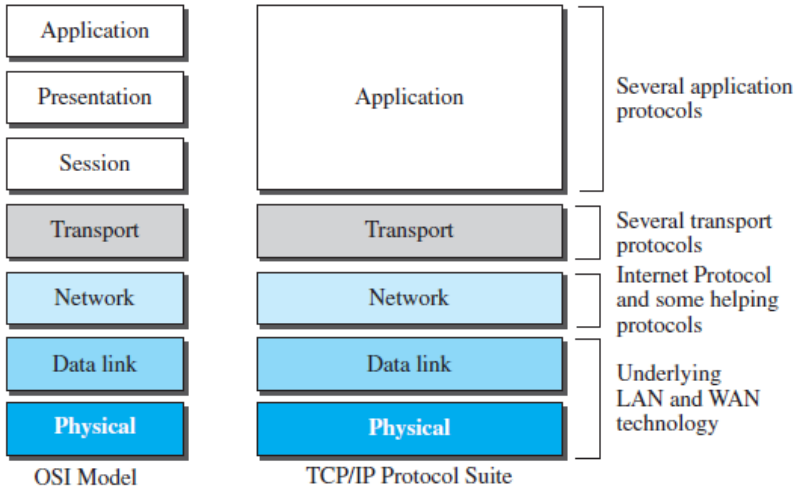
Presentation Layer

Responsible for handling differences in data representation to applications

- Duties/services
  - Data translation
  - Encryption
  - Decryption
  - Compression

OSI versus TCP/IP

- When we compare the two models, we find that two layers, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model



Lack of OSI Model's Success

- The OSI model appeared after the TCP/IP protocol suite. Most experts were at first excited and thought that the TCP/IP protocol would be fully replaced by the OSI model. This did not happen for several reasons, but we describe only three, which are agreed upon by all experts in the field:

- 1. OSI was completed when TCP/IP was fully in place. changing it would cost a lot.

- 2. some layers in the OSI model were never fully defined. For example, although the services provided by the presentation and the session layers were listed in the document, actual protocols for these two layers were not fully defined, nor were they fully described, and the corresponding software was not fully developed.

- 3. when OSI was implemented by an organization in a different application, it did not show a high enough level of performance to entice the Internet authority to switch from the TCP/IP protocol suite to the OSI model.

same performance as TCP

Summary

- Layered tasks are used for complex communication details
- Layer models were created for reference
  - Internet model
  - ISO's OSI model
- Protocol suite is a collection of protocols operating at various layers

Some network apps

- social networking
- Web
- text messaging
- e-mail
- multi-user network games
- streaming stored video (YouTube, Hulu, Netflix)
- P2P file sharing
- voice over IP (e.g., Skype)
- real-time video conferencing (e.g., Zoom)
- Internet search
- remote login
- …

Q: your favorites?

Application architectures

possible structure of applications:

- client-server
- peer-to-peer (P2P)

App-layer protocol defines

- types of messages exchanged,
  - e.g., request, response
- message syntax:
  - what fields in messages & how fields are delineated
- message semantics
  - meaning of information in fields
- rules for when and how processes send & respond to messages

open protocols:

- defined in RFCs
- allows for interoperability
- e.g., HTTP, SMTP

proprietary protocols:

- e.g., Skype

Web and HTTP

First, a review…

- *web page* consists of *objects*
- object can be HTML file, JPEG image, Java applet, audio file,…
- web page consists of *base HTML-file* which includes *several referenced objects*
- each object is addressable by a *URL,* e.g.,

`www.someschool.edu/someDept/pic.gif`

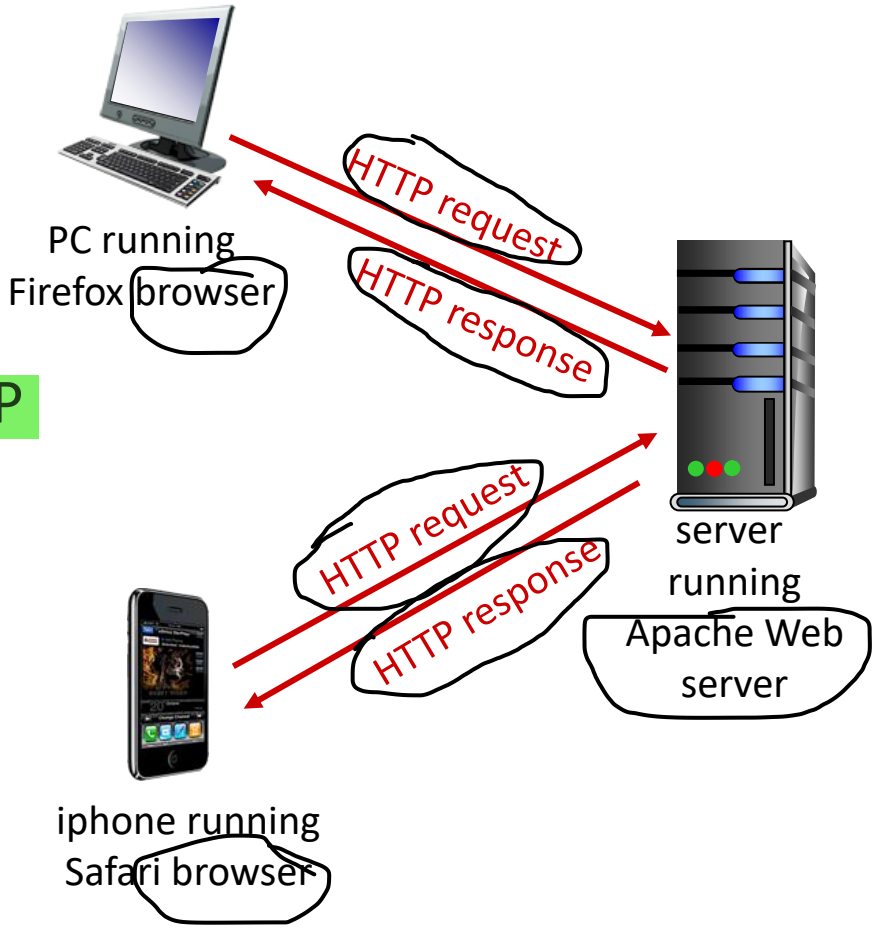
host name of server that houses the object

path name of object

HTTP overview

HTTP: hypertext transfer protocol

- Web's application layer protocol (heart of Web)
  - RFC 1945/2616
- Implemented as client/server model where client and server exchange HTTP messages
  - *client:* web browser that requests, receives, (using HTTP protocol) and "displays" Web objects
    - Internet Explore or firefox
  - *server:* Web server sends (using HTTP protocol) objects in response to requests
    - Apache and Microsoft Internet Information server



HTTP overview (continued)

uses TCP:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

HTTP is "stateless"

- server maintains no information about past client requests

aside

protocols that maintain "state" are complex!

- past history (state) must be maintained
- if server/client crashes, their views of "state" may be inconsistent, must be reconciled

HTTP connections

Non-persistent HTTP

1. TCP connection opened
2. at most one object sent over TCP connection
3. TCP connection closed

downloading multiple objects required multiple connections

Persistent HTTP

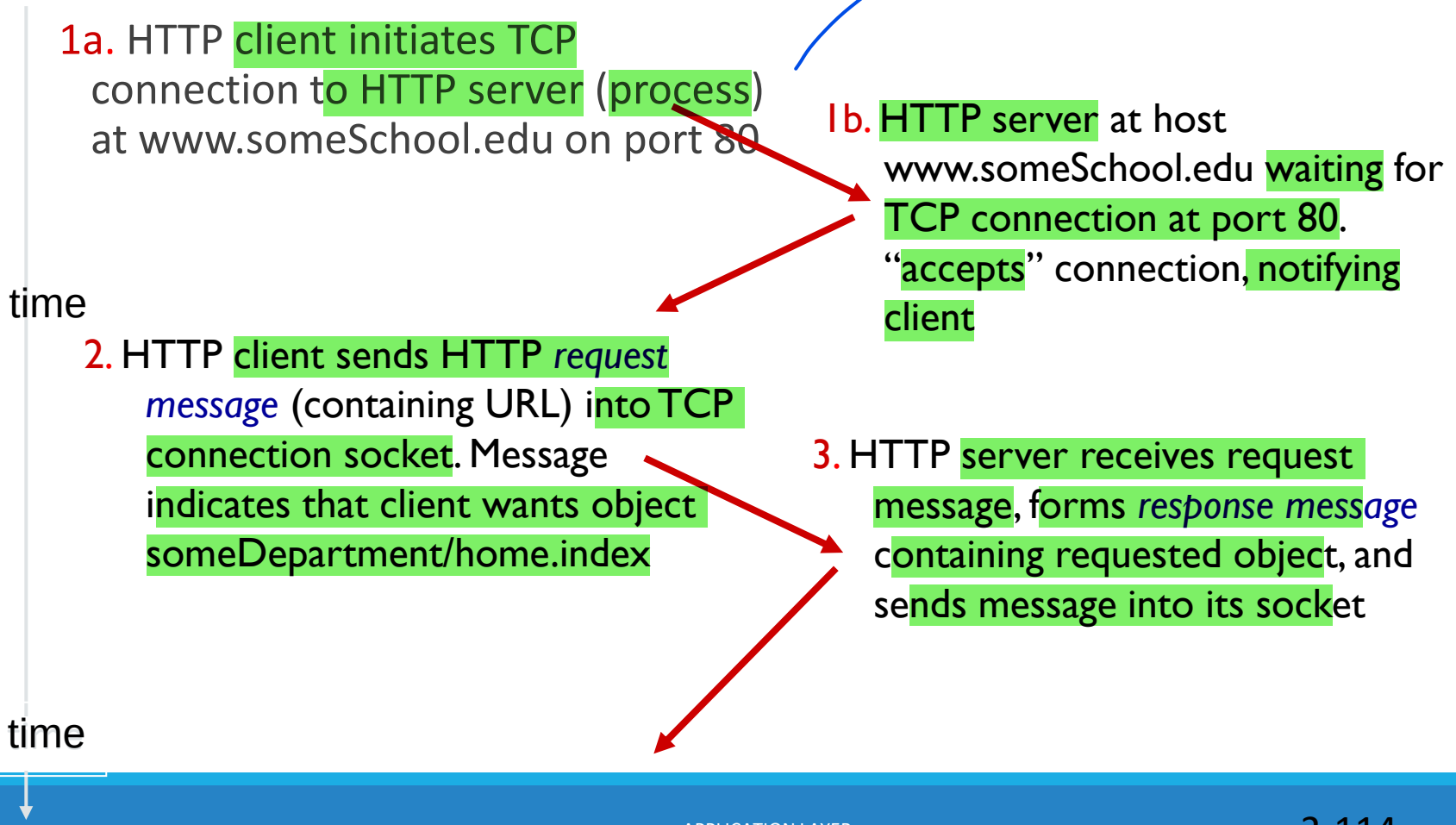
- TCP connection opened to a server
- multiple objects can be sent over *single* TCP connection between client, and that server
- TCP connection closed

Non-persistent HTTP

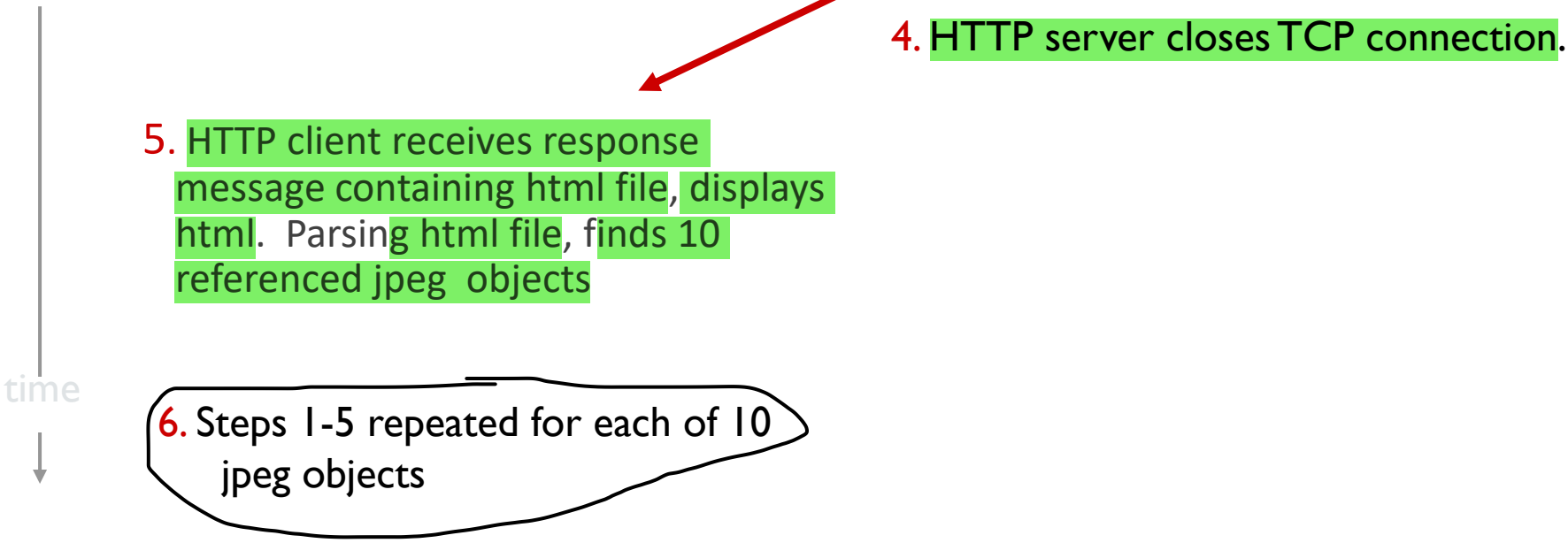
suppose user enters URL:

`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)



Non-persistent HTTP (cont.)



Non-persistent HTTP: response time

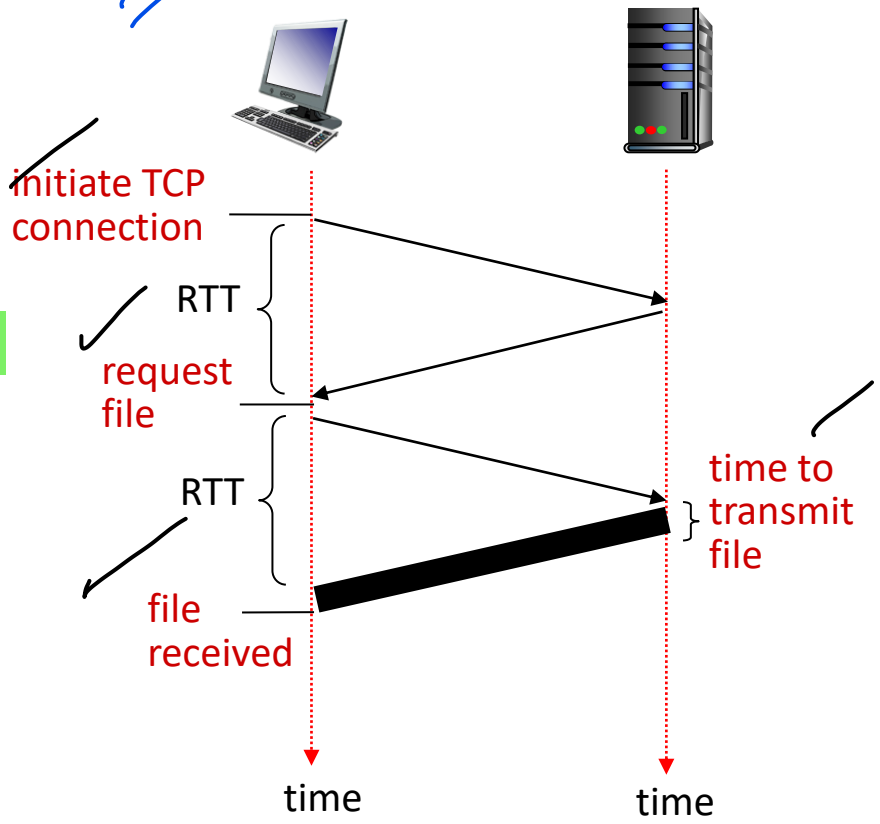
Round Trip Time

RTT (definition): time for a small packet to travel from client to server and back

HTTP response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time
- non-persistent HTTP response time =

  2RTT+ file transmission time



Persistent HTTP

non-persistent HTTP issues:

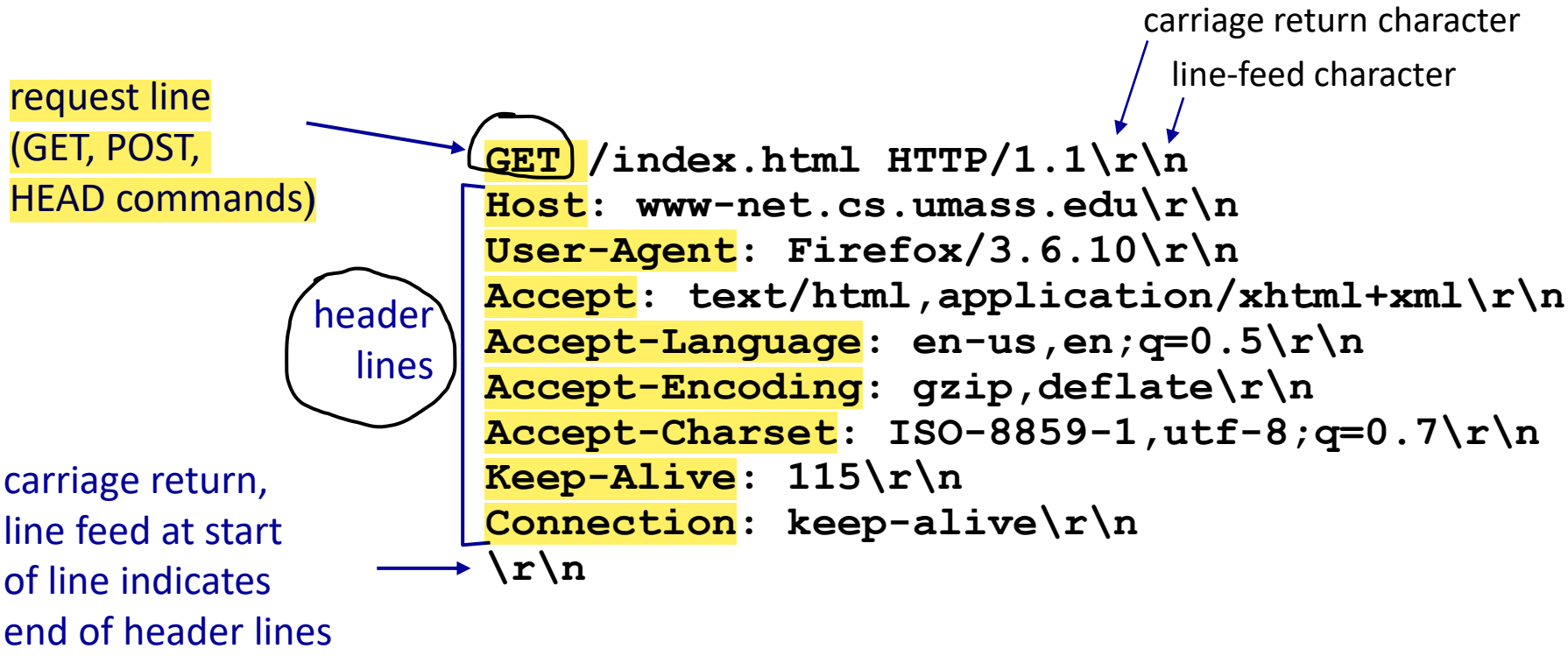
- requires 2 RTTs per object
- OS overhead for *each* TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

persistent HTTP:

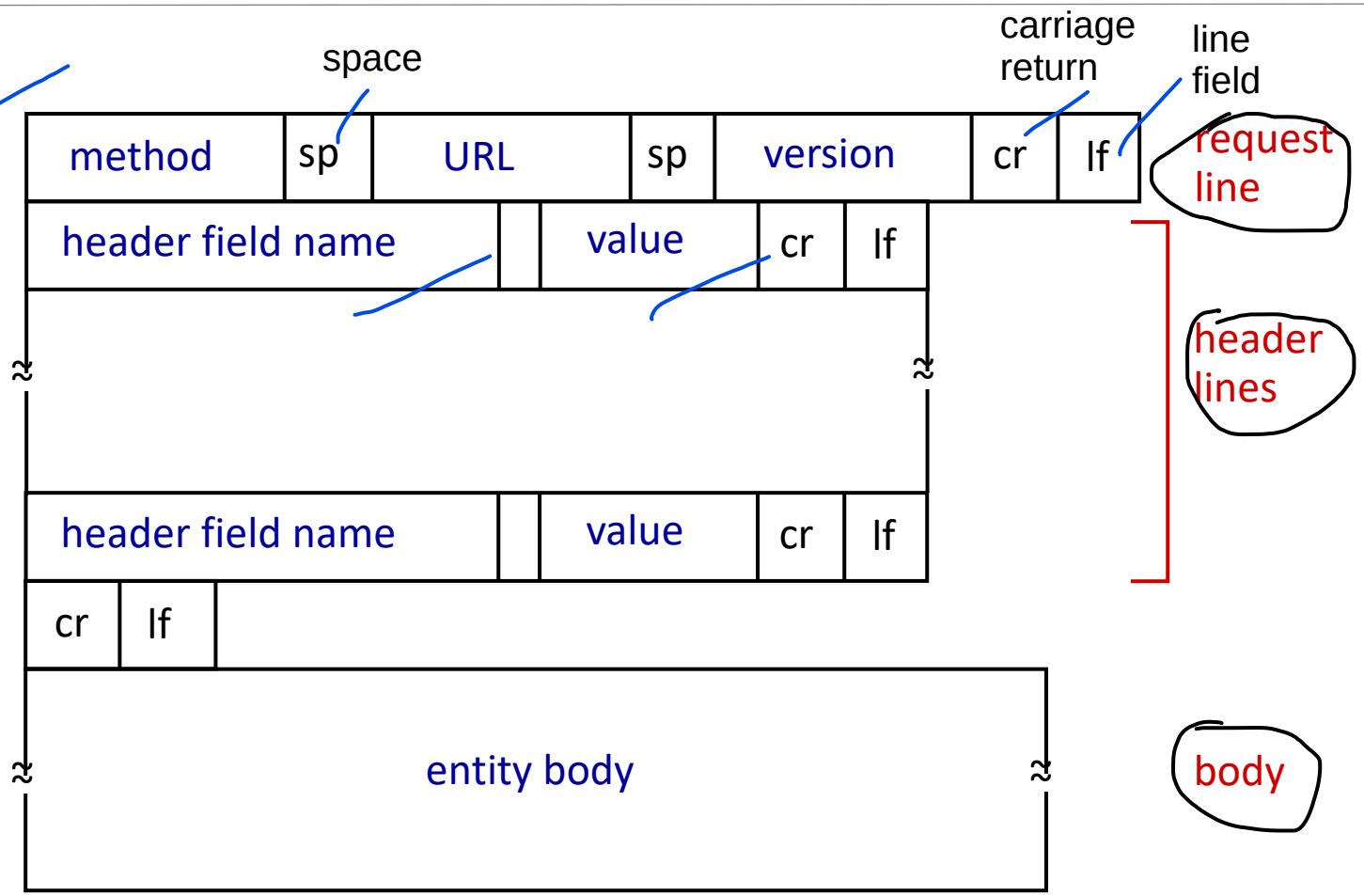
- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

HTTP request message

- two types of HTTP messages: *request, response*
- HTTP request message:
  - ASCII (human-readable format)



HTTP request message: general format



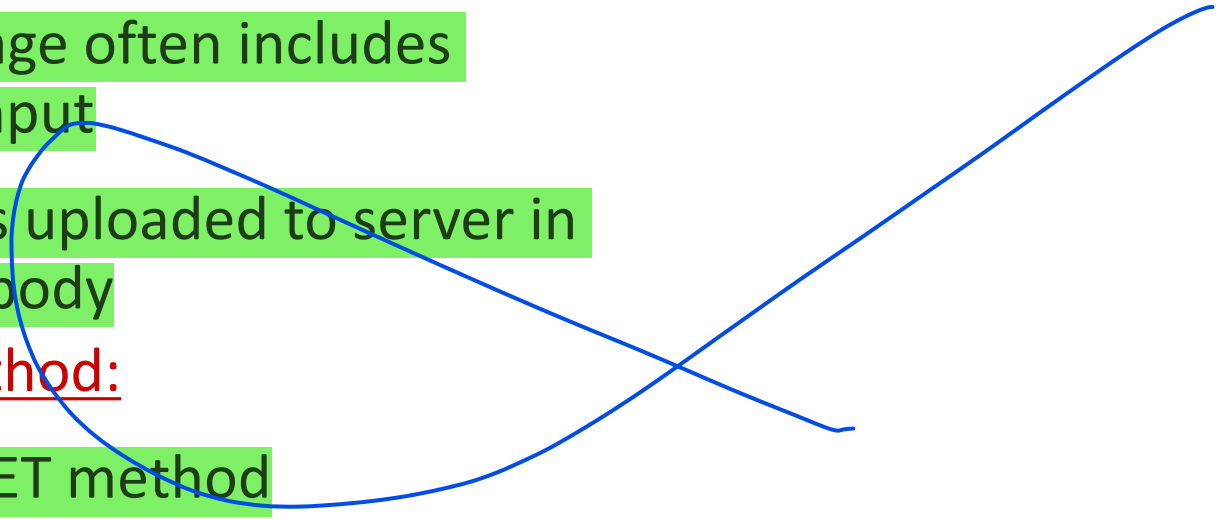
Uploading form input

POST method:

- web page often includes form input
- input is uploaded to server in entity body

URL method:

- uses GET method
- input is uploaded in URL field of request line:



```
www.somesite.com/animalsearch?monkeys&banana
```

Method types

HTTP/1.0:

- GET
- POST
- HEAD
  - asks server to leave requested object out of response

POST method:

- web page often includes form input
- user input sent from client to server in entity body of HTTP POST request message

GET method (for sending data to server):

- include user data in URL field of HTTP GET request message (following a '?'):

```
www.somesite.com/animalsearch?monkeys&banana
```

HTTP/1.1:

- GET, POST, HEAD
- PUT
  - uploads file in entity body to path specified in URL field
- DELETE
  - deletes file specified in the URL field

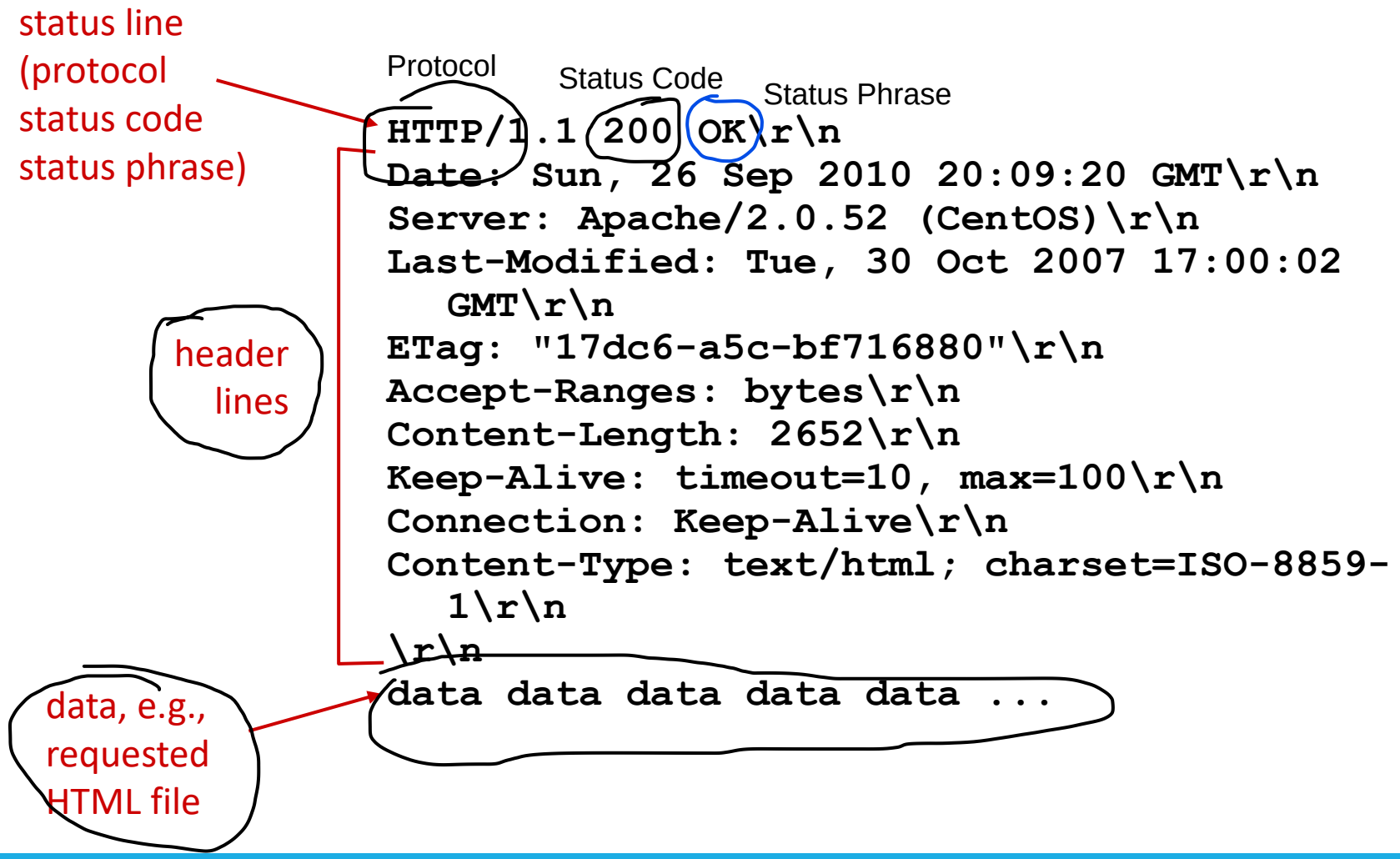
HEAD method:

- requests headers (only) that would be returned *if* specified URL were requested with an HTTP GET method.

PUT method:

- uploads new file (object) to server
- completely replaces file that exists at specified URL with content in entity body of POST HTTP request message

HTTP response message



HTTP response status codes

- status code appears in 1st line in server-to-client response message.
- some sample codes:

200 OK

- request succeeded, requested object later in this msg

301 Moved Permanently

- requested object moved, new location specified later in this msg (Location:)

400 Bad Request

- request msg not understood by server

404 Not Found

- requested document not found on this server

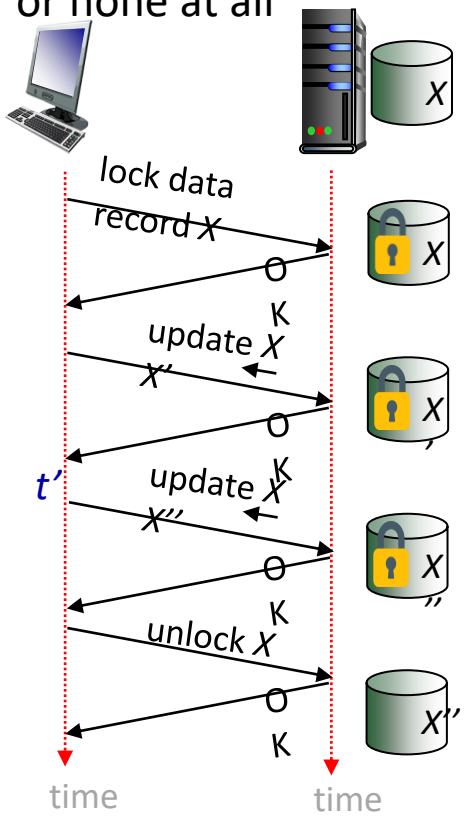
505 HTTP Version Not Supported

Maintaining user/server state: cookies

Recall: HTTP GET/response interaction is *stateless*

- no notion of multi-step exchanges of HTTP messages to complete a Web "transaction"
  - no need for client/server to track "state" of multi-step exchange
  - all HTTP requests are independent of each other
  - no need for client/server to "recover" from a partially-completed-but-never-completely-completed transaction

a stateful protocol: client makes two changes to X, or none at all



Q: what happens if network connection or client crashes at *t'* ?

Maintaining user/server state: cookies

Web sites and client browser use *cookies* to maintain some state between transactions

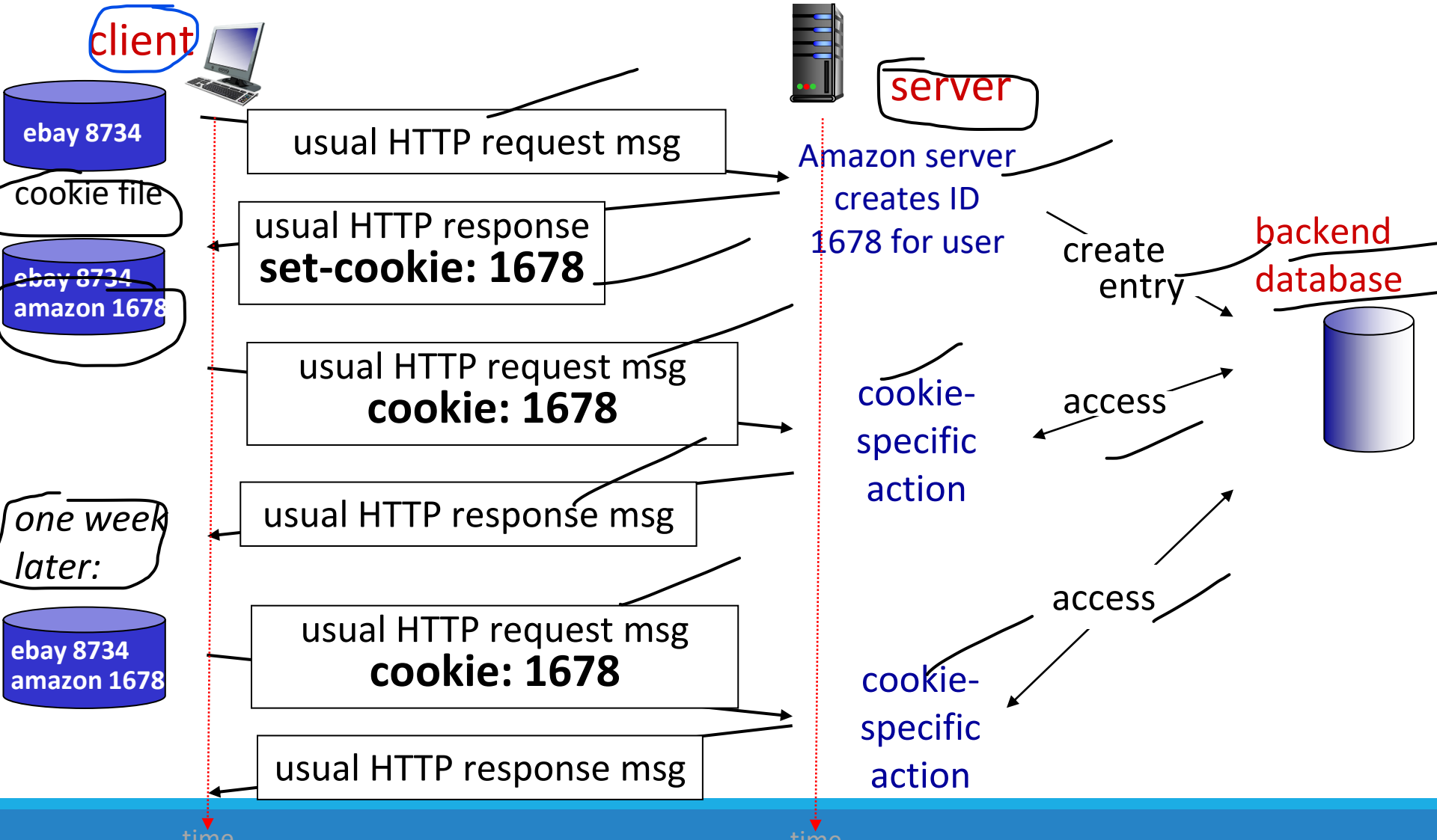
four components:

1) cookie header line of HTTP *response* message
2) cookie header line in next HTTP *request* message
3) cookie file kept on user's host, managed by user's browser
4) back-end database at Web site

Example:

- Susan uses browser on laptop, visits specific e-commerce site for first time
- when initial HTTP requests arrives at site, site creates:
  - unique ID (aka "cookie")
  - entry in backend database for ID
- subsequent HTTP requests from Susan to this site will contain cookie ID value, allowing site to "identify" Susan

Maintaining user/server state: cookies



HTTP cookies: comments

What cookies can be used for:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

Challenge: How to keep state?

- *at protocol endpoints:* maintain state at sender/receiver over multiple transactions
- *in messages:* cookies inHTTP messages carry state

aside

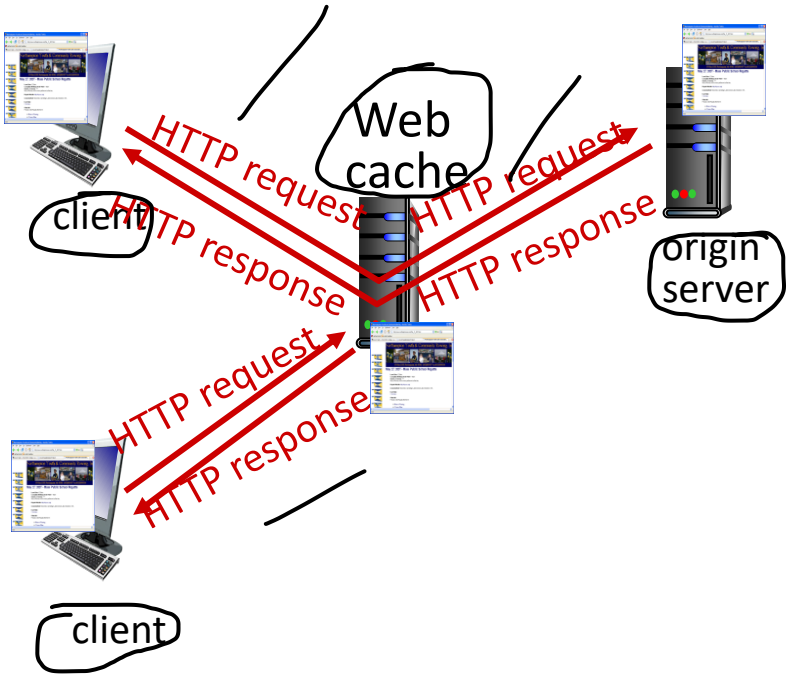
cookies and privacy:

- cookies permit sites to *learn* a lot about you on their site.
- third party persistent cookies (tracking cookies) allow common identity (cookie value) to be tracked across multiple web sites

Web caches

Goal: satisfy client requests without involving origin server

- user configures browser to point to a (local) *Web cache*
- browser sends all HTTP requests to cache
  - *if* object in cache: cache returns object to client
  - *else* cache requests object from origin server, caches received object, then returns object to client



Web caches (aka proxy servers)

- Web cache acts as both client and server
  - server for original requesting client
  - client to origin server

- server tells cache about object's allowable caching in response header:

```
Cache-Control: max-age=<seconds>
```

```
Cache-Control: no-cache
```

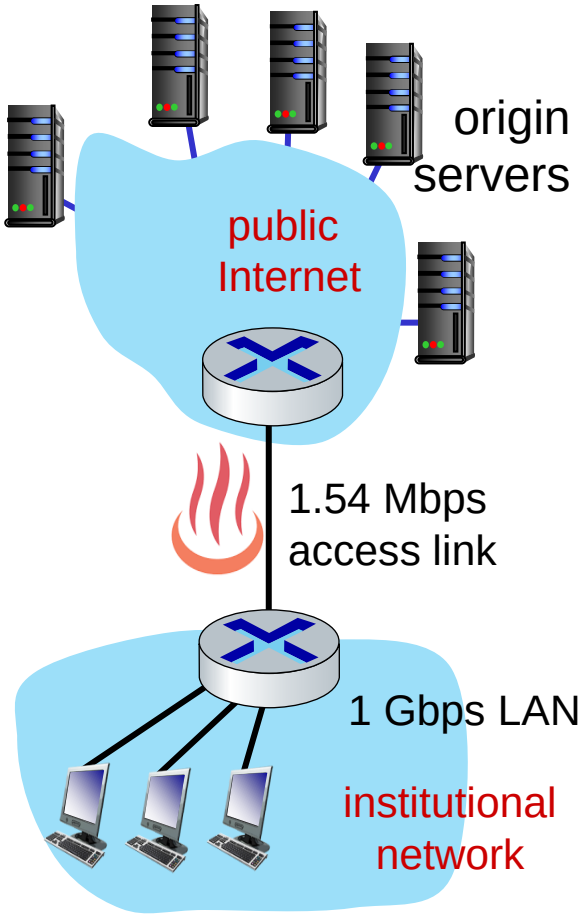
Why Web caching?

- reduce response time for client request
  - cache is closer to client
- reduce traffic on an institution's access link
- Internet is dense with caches
  - enables "poor" content providers to more effectively deliver content

Caching example

Scenario:

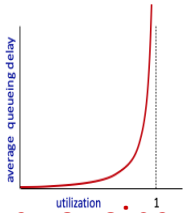
- access link rate: 1.54 Mbps
- RTT from institutional router to server: 2 sec
- web object size: 100K bits
- average request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps



Performance:

- access link utilization = .97
- LAN utilization: .0015
- end-end delay = Internet delay + access link delay + LAN delay

= 2 sec + minutes + usecs

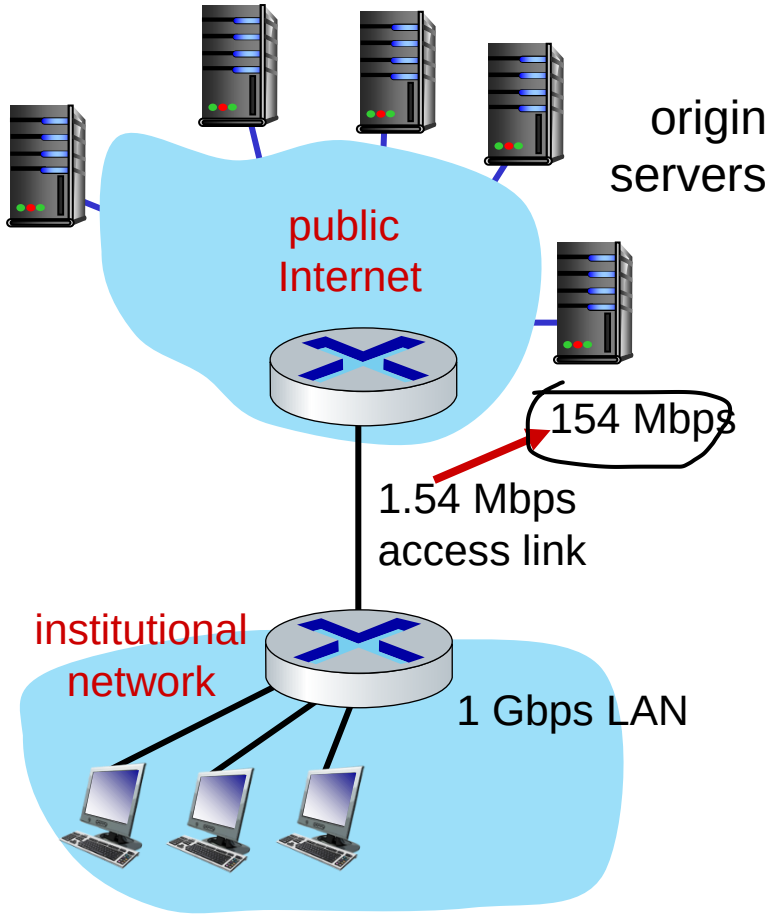


problem: large queueing delays at high utilization!

Option 1: buy a faster access link

Scenario:

- access link rate: 1.54 Mbps → 154 Mbps
- RTT from institutional router to server: 2 sec
- web object size: 100K bits
- average request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps



Performance:

- access link utilization = .97 → .0097
- LAN utilization: .0015
- end-end delay = Internet delay + access link delay + LAN delay
  = 2 sec + minutes + usecs → msecs

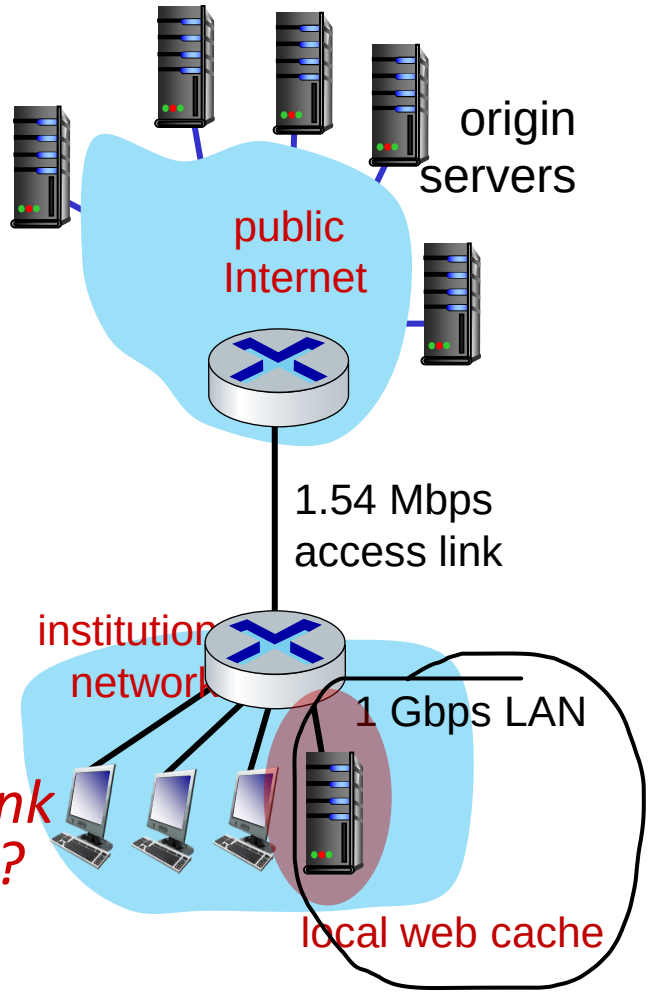
Cost: faster access link (expensive!)

Option 2: install a web cache

Scenario:

- access link rate: 1.54 Mbps
- RTT from institutional router to server: 2 sec
- web object size: 100K bits
- average request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps

Cost: web cache (cheap!)



Performance:

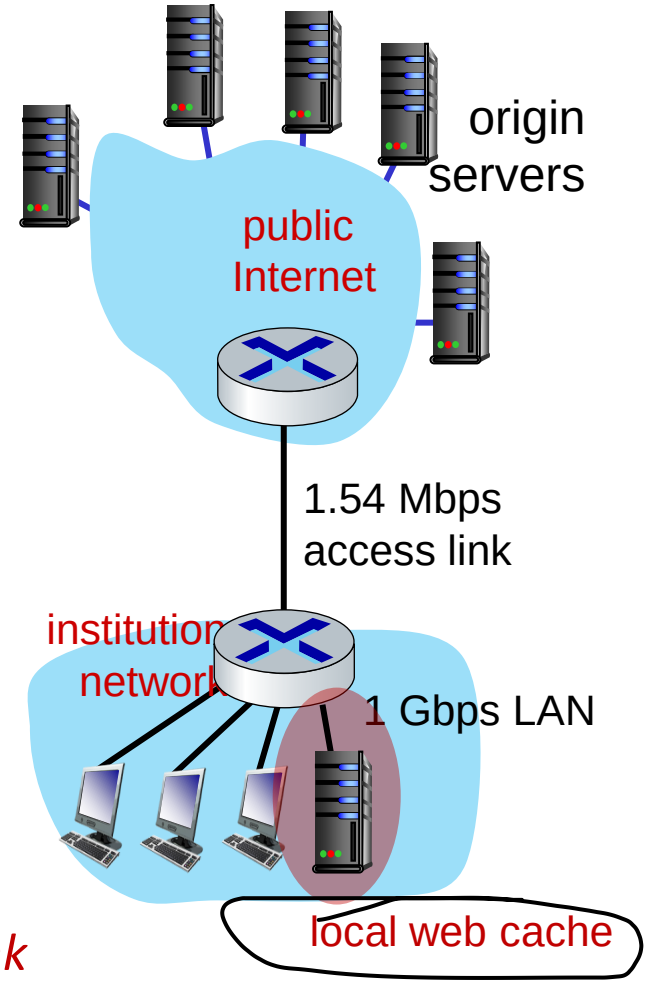
- LAN utilization: .?
- access link utilization = ?
- average end-end delay = ?

How to compute link utilization, delay?

Calculating access link utilization, end-end delay with cache:

suppose cache hit rate is 0.4:

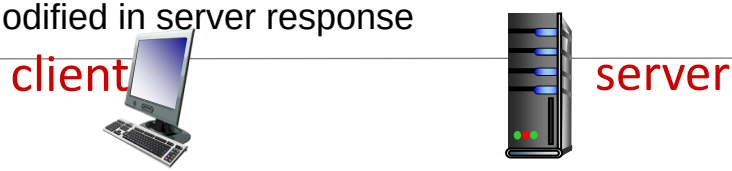
- 40% requests served by cache, with low (msec) delay
- 60% requests satisfied at origin
  - rate to browsers over access link
    = 0.6 * 1.50 Mbps = .9 Mbps
  - access link utilization = 0.9/1.54 = .58 means low (msec) queueing delay at access link
- average end-end delay:
  = 0.6 * (delay from origin servers)
  + 0.4 * (delay when satisfied at cache)
  = 0.6 (2.01) + 0.4 (~msecs) = ~ 1.2 secs



lower average end-end delay than with 154 Mbps link (and cheaper too!)

Conditional GET

Don't Get Object (only http NOT Modified Header) only if requested object is not modified in server response



Goal: don't send object if cache has up-to-date cached version

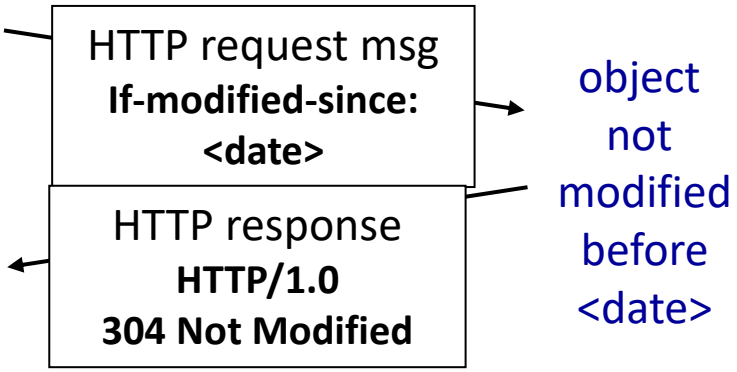
- no object transmission delay (or use of network resources)

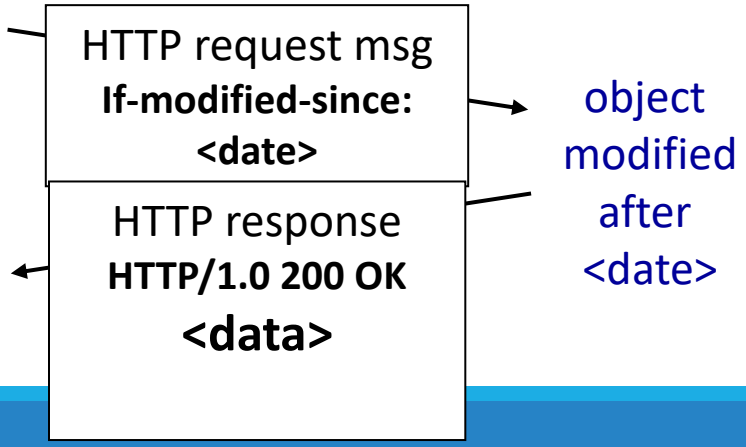
- *client:* specify date of cached copy in HTTP request

  If-modified-since: <date>

- *server:* response contains no object if cached copy is up-to-date:

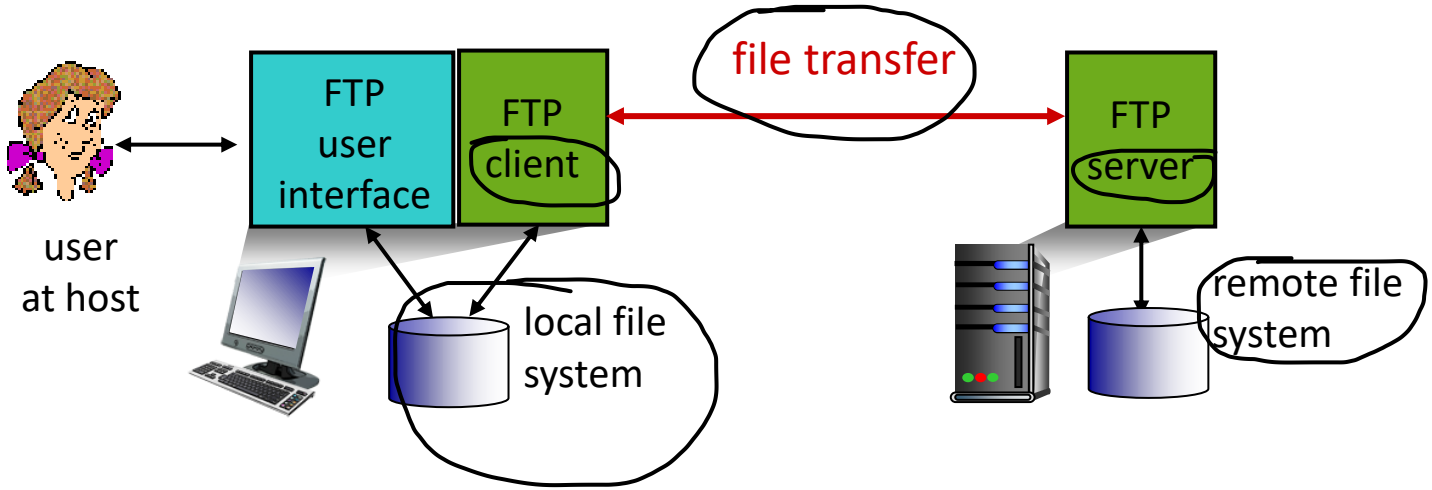
  HTTP/1.0 304 Not Modified





FTP: the file transfer protocol

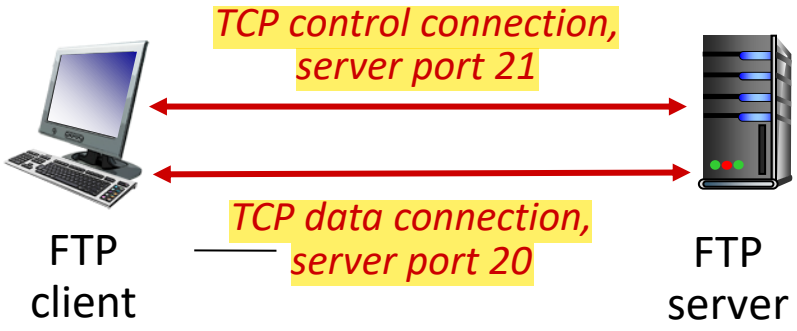
Port 21



- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21

FTP: separate control, data connections

- FTP client contacts FTP server at port 21, using TCP
- client authorized over control connection
- client browses remote directory, sends commands over control connection
- when server receives file transfer command, *server* opens 2nd TCP data connection (for file) *to* client
- after transferring one file, server closes data connection



- server opens another TCP data connection to transfer another file
- control connection: *"out of band"*
- FTP server maintains "state": current directory, earlier authentication

FTP commands, responses

sample commands:

- sent as ASCII text over control channel
- **USER *username***
- **PASS *password***
- **LIST** return list of file in current directory
- **RETR filename** retrieves (gets) file
- **STOR filename** stores (puts) file onto remote host

sample return codes

- status code and phrase (as in HTTP)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

Electronic mail

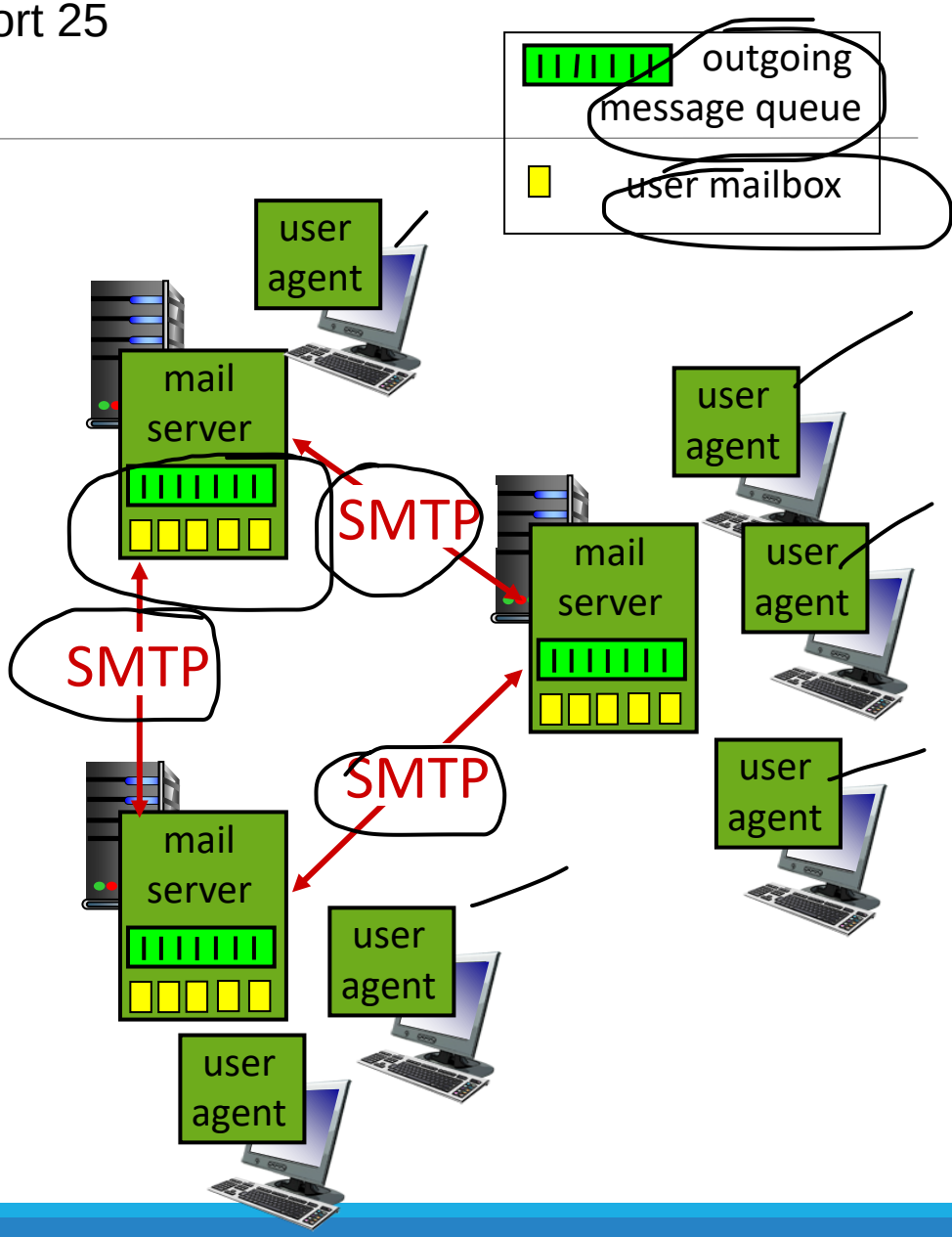
Port 25

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

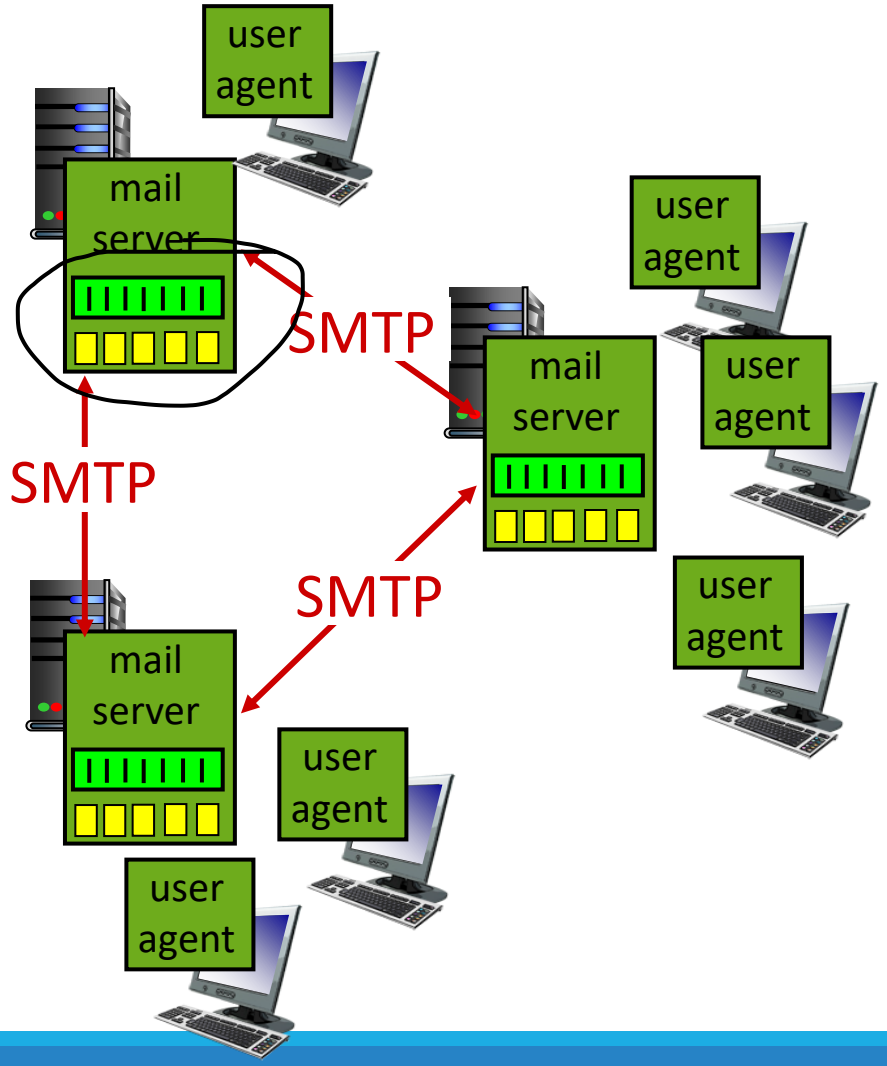
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Outlook, Thunderbird, iPhone mail client
- outgoing, incoming messages stored on server



Electronic mail: mail servers

mail servers:

- *mailbox* contains incoming messages for user
- *message queue* of outgoing (to be sent) mail messages
- *SMTP protocol* between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server



Electronic Mail: SMTP [RFC 2821]

- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction (like HTTP, FTP)
  - commands: ASCII text
  - response: status code and phrase
- messages must be in 7-bit ASCI

Scenario: Alice sends message to Bob

1) Alice uses UA to compose message "to" `bob@someschool.edu`

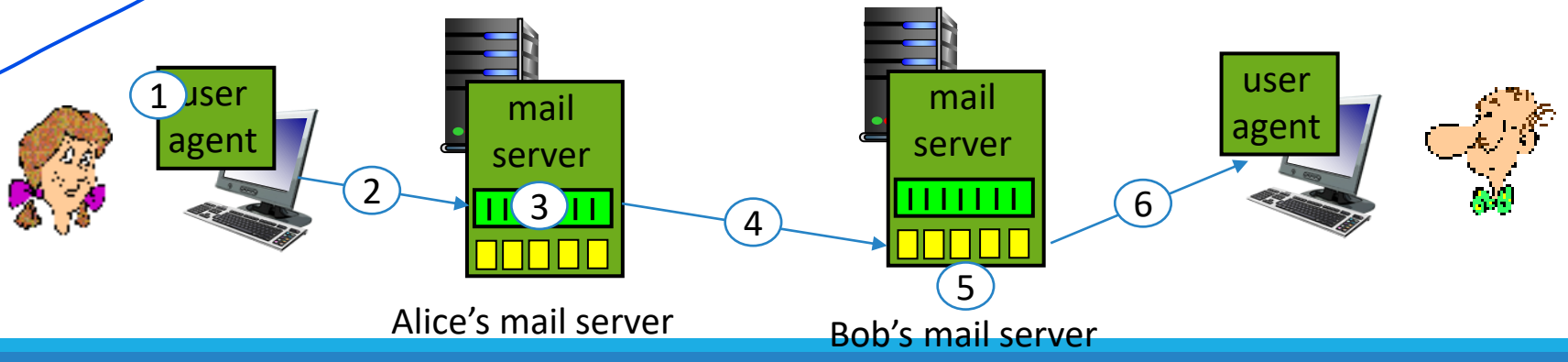
2) Alice's UA sends message to her mail server; message placed in message queue

3) client side of SMTP opens TCP connection with Bob's mail server

4) SMTP client sends Alice's message over the TCP connection

5) Bob's mail server places the message in Bob's mailbox

6) Bob invokes his user agent to read message



Sample SMTP interaction

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Try SMTP interaction for yourself:

- `telnet servername 25`
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

above lets you send email without using email client (reader)

SMTP: final words

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses `CRLF.CRLF` to determine end of message

comparison with HTTP:

- HTTP: pull
- SMTP: push
- both have ASCII command/response interaction, status codes
- HTTP: each object encapsulated in its own response msg
- SMTP: multiple objects sent in multipart msg

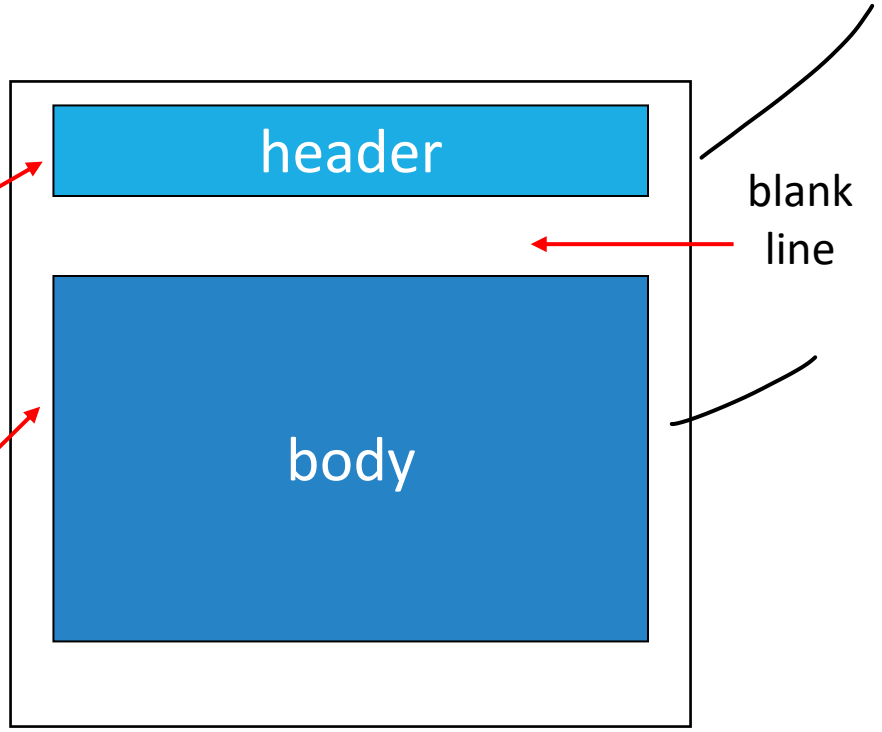
Mail message format

SMTP: protocol for exchanging email msgs

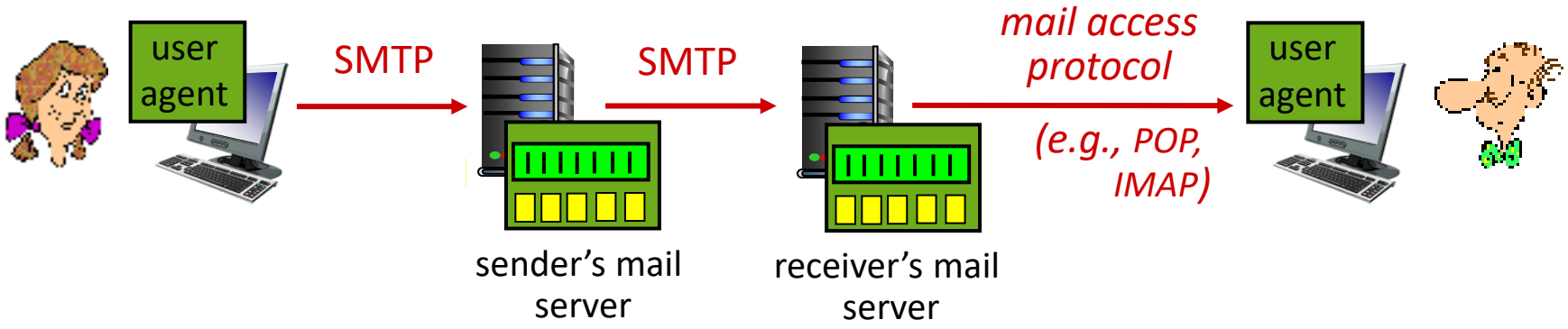
RFC 822: standard for text message format:

- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different *from* SMTP MAIL FROM, RCPT TO: commands!
- Body: the "message"
  - ASCII characters only



Mail access protocols



- SMTP: delivery/storage to receiver's server
- mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]: authorization, download
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
  - HTTP: gmail, Hotmail, Yahoo! Mail, etc.

POP3 protocol

authorization phase

- client commands:
  - **`user:`** declare username
  - **`pass:`** password
- server responses
  - **`+OK`**
  - **`-ERR`**

transaction phase, client:

- **`list:`** list message numbers
- **`retr:`** retrieve message by number
- **`dele:`** delete
- **`quit`**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

POP3 (more) and IMAP

more about POP3

- previous example uses POP3 "download and delete" mode
  - Bob cannot re-read e-mail if he changes client
- POP3 "download-and-keep": copies of messages on different clients
- POP3 is stateless across sessions

IMAP

- keeps all messages in one place: at server
- allows user to organize messages in folders
- keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

DNS: domain name system

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

DNS: services, structure

DNS services

- **hostname to IP address translation**

Whatismyip.com --→ 103.170.244.34

host aliasing
  - canonical, alias names

www.ibm.com -------→

www.relay1.west-coast.ibm.com

- **mail server aliasing**

Hotmail.com---→ relay1.west-coast.Hotmail.com

- load distribution
  - **replicated Web servers: many IP addresses correspond to one name**

why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

A: doesn't scale!

Thinking about the DNS

humongous distributed database:

- ~ billion records, each simple

handles many *trillions* of queries/day:

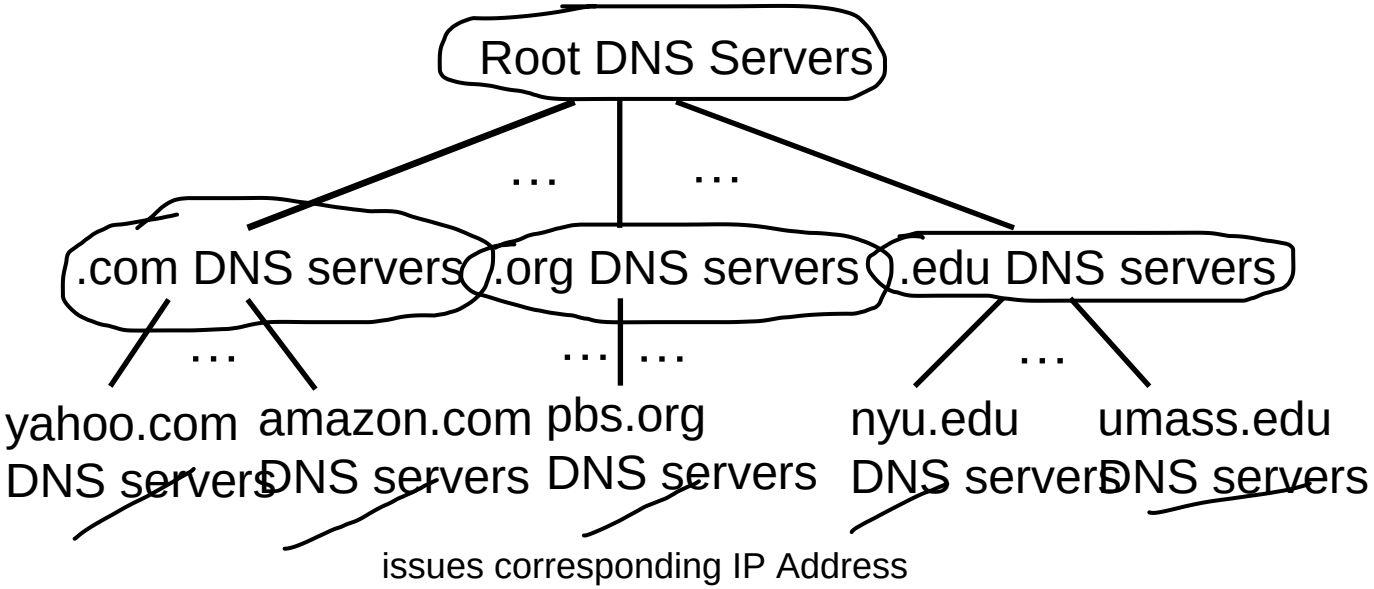
- *many* more reads than writes
- *performance matters:* almost every Internet transaction interacts with DNS - msecs count!

organizationally, physically decentralized:

- millions of different organizations responsible for their records

"bulletproof": reliability, security

DNS: a distributed, hierarchical database



Root

Top Level Domain

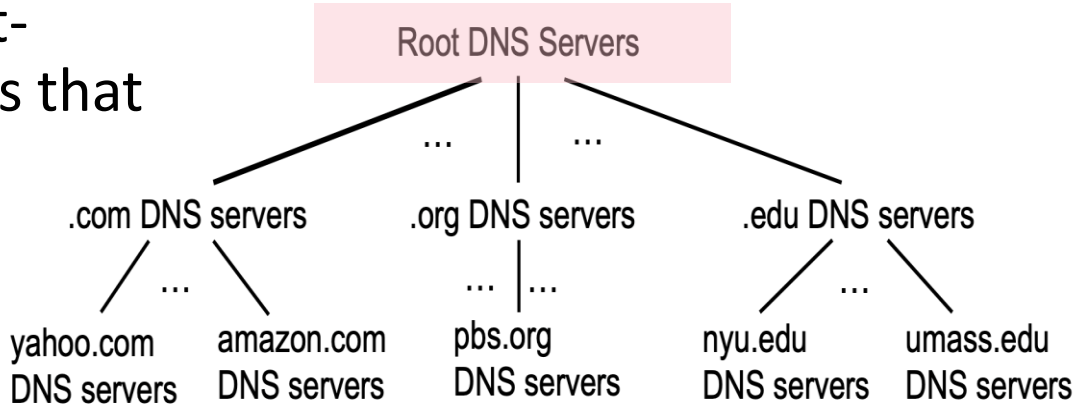
Authoritative

Client wants IP address for www.amazon.com; 1st approximation:

- client queries root server to find .com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

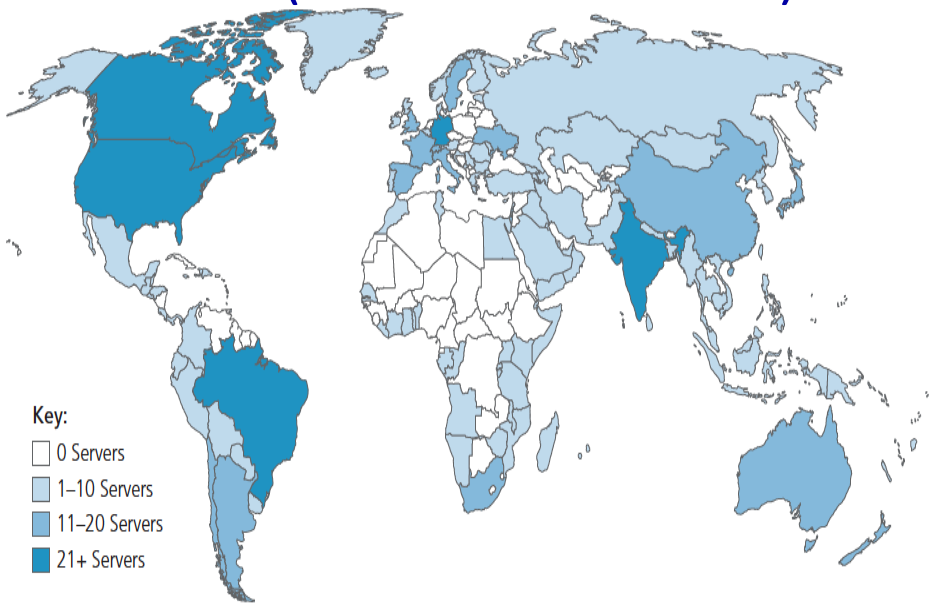
- official, contact-of-last-resort by name servers that can not resolve name



DNS: root name servers

- official, contact-of-last-resort by name servers that can not resolve name
- *incredibly important* Internet function
  - Internet couldn't function without it!
  - DNSSEC – provides security (authentication, message integrity)
- ICANN (Internet Corporation for Assigned Names and Numbers) manages root DNS domain

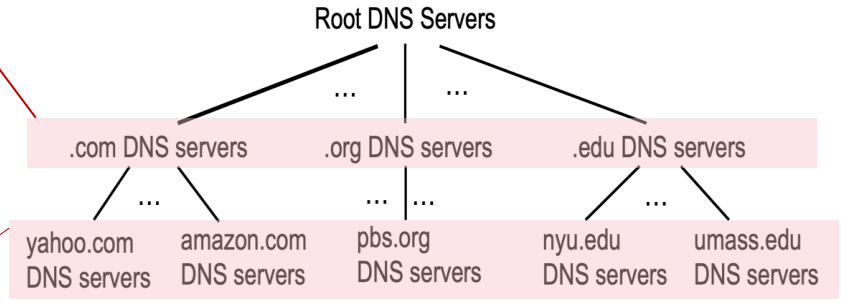
13 logical root name "servers" worldwide each "server" replicated many times (~200 servers in US)



Top-Level Domain, and authoritative servers

Top-Level Domain (TLD) servers:

- responsible for .com, .org, .net, .edu, .aero, .jobs, .museums, and all top-level country domains, e.g.: .cn, .uk, .fr, .ca, .jp
- Network Solutions: authoritative registry for .com, .net TLD
- Educause: .edu TLD



authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name servers

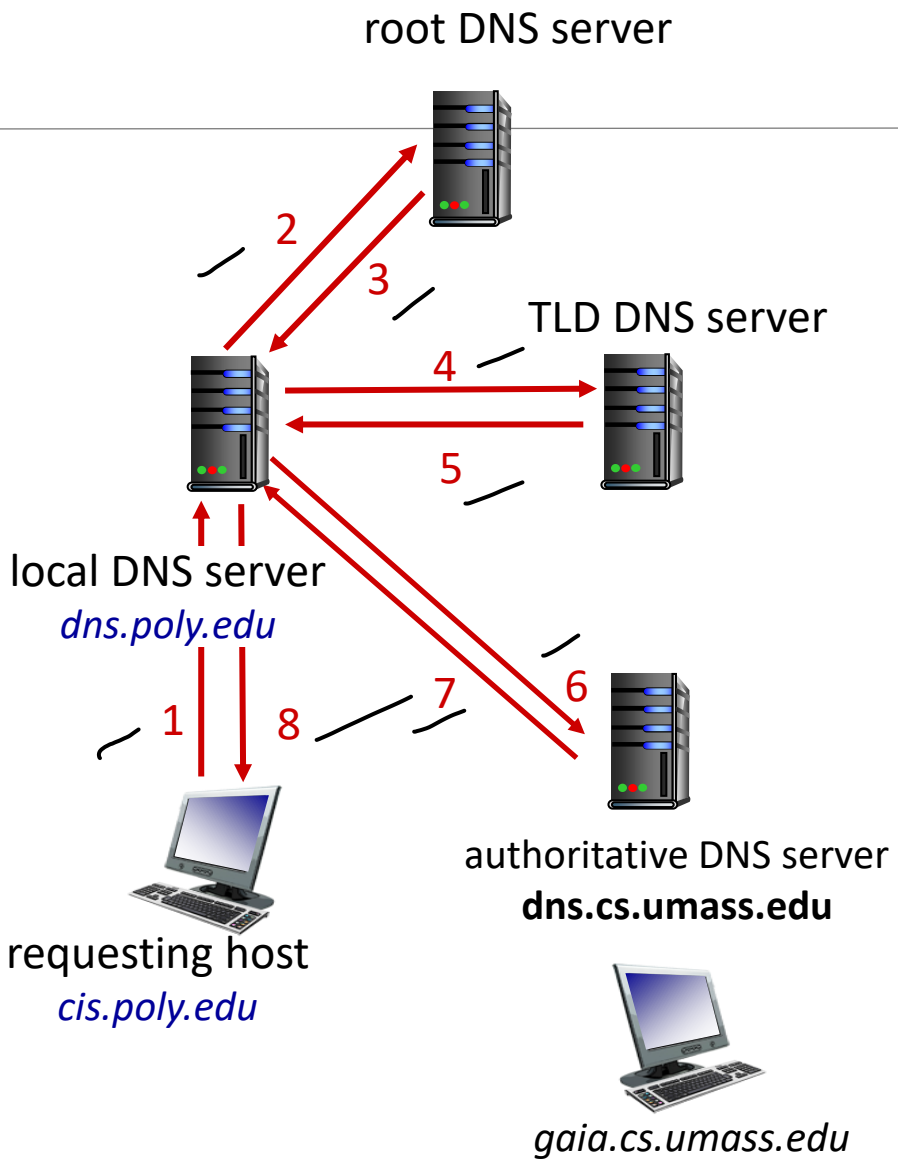
- when host makes DNS query, it is sent to its *local* DNS server
  - Local DNS server returns reply, answering:
    - from its local cache of recent name-to-address translation pairs (possibly out of date!)
    - forwarding request into DNS hierarchy for resolution
  - each ISP has local DNS name server; to find yours:
    - MacOS: `% scutil --dns`
    - Windows: `>ipconfig /all`
- local DNS server doesn't strictly belong to hierarchy

DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

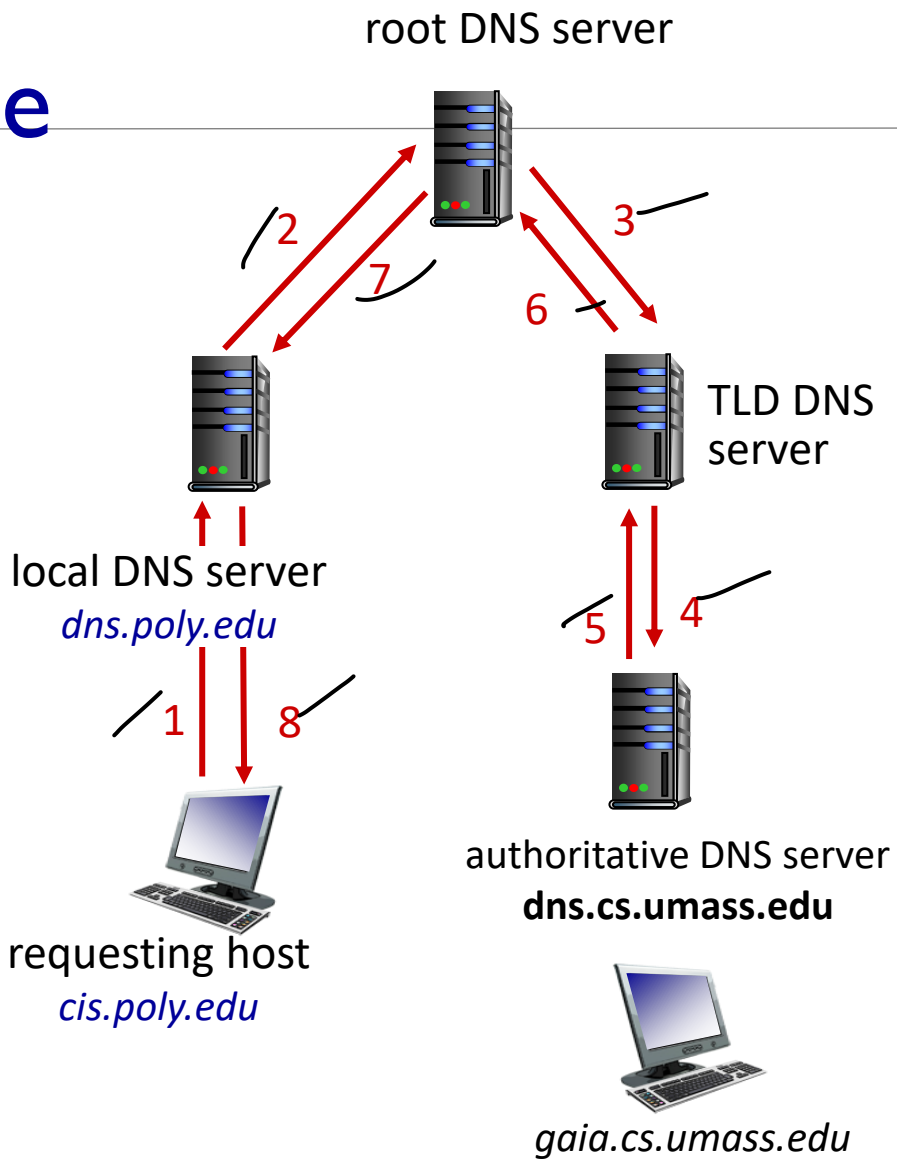
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



DNS: caching, updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
    - thus root name servers not often visited
- cached entries may be *out-of-date* (best effort name-to-address translation!)
  - if name host changes IP address, may not be known Internet-wide until all TTLs expire
- update/notify mechanisms proposed IETF standard
  - RFC 2136

DNS records

DNS: distributed db storing resource records (RR)

RR format: **(name, value, type, ttl)** time to last

type=A

- **name** is hostname
- **value** is IP address

type=NS

- **name** is domain (e.g., foo.com)
- **value** is hostname of authoritative name server for this domain

type=CNAME

- **name** is alias name for some "canonical" (the real) name
- **www.ibm.com** is really **servereast.backup2.ibm.com**
- **value** is canonical name

type=MX

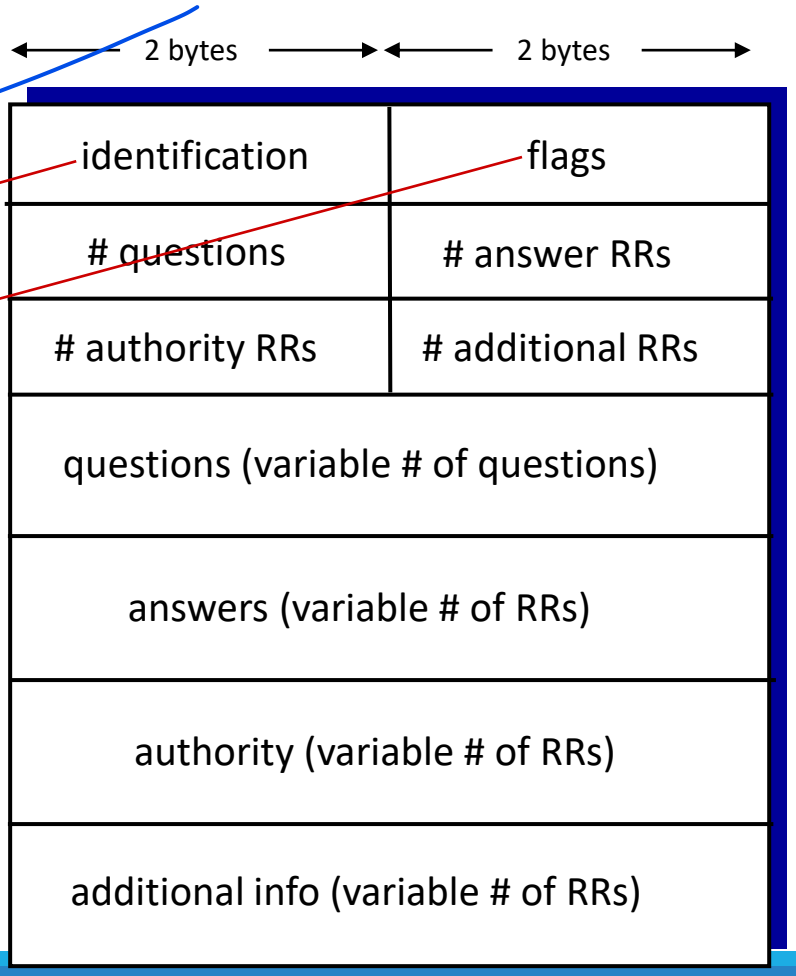
- **value** is name of mailserver associated with **name**

DNS protocol, messages

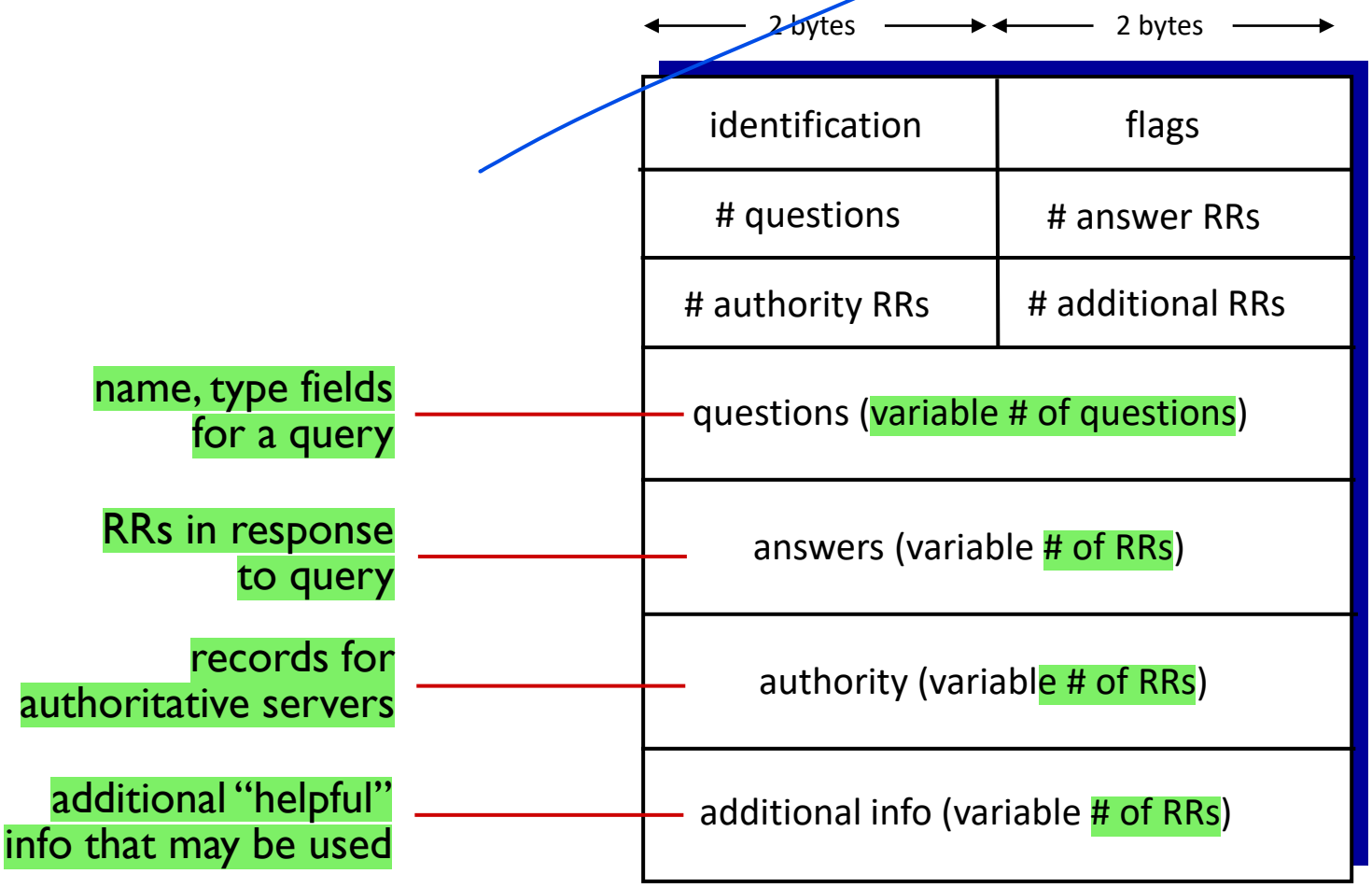
- *query* and *reply* messages, both with same *message format*

msg header

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



DNS protocol, messages



Inserting records into DNS

How RR get into the database at start?

- example: new startup "Network Utopia"
- register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions till 1999 for com, net and org domains)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:

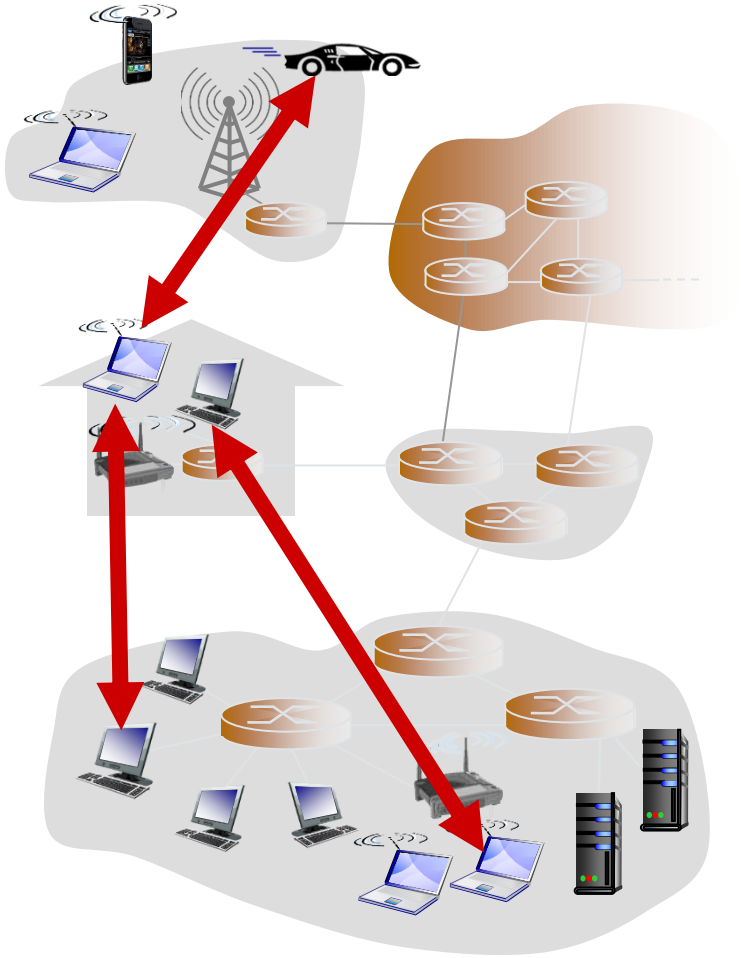
```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

Domain — HostName — NS
HostName — IP Address — A

- create authoritative server type A record for www.networkuptopia.com; type MX record for mail server networkutopia.com
- People will be able to visit your web site and send email to employees

Pure P2P architecture

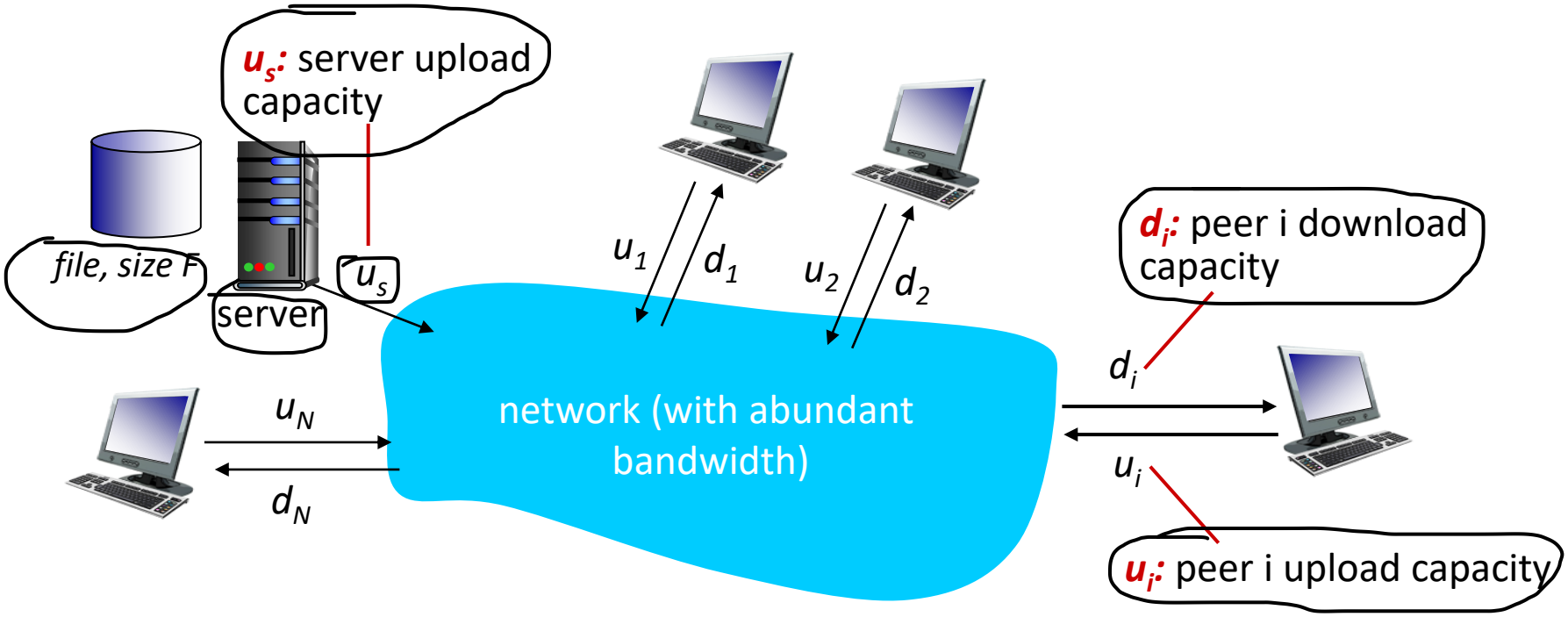
- *no* always-on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
  - *self scalability* – new peers bring new service capacity, and new service demands
- peers are intermittently connected and change IP addresses
  - complex management
- examples: P2P file sharing (BitTorrent), streaming (KanKan), VoIP (Skype)



File distribution: client-server vs P2P

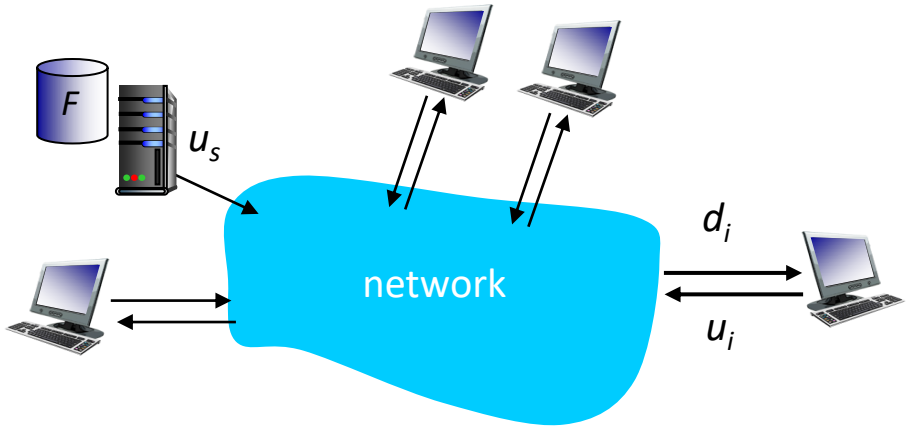
Question: how much time to distribute file (size *F*) from one server to *N peers*?

- peer upload/download capacity is limited resource



File distribution time: client-server

- *server transmission:* must sequentially send (upload)one copy of file to N peers:
  - time to send one copy: F/u_s
  - time to send to N peers: NF/u_s
- *client:* each client must download file copy
  - d_{min} = min client download rate
  - min client download time: F/d_{min}



Transmission Time= Packet size/ Bit Rate

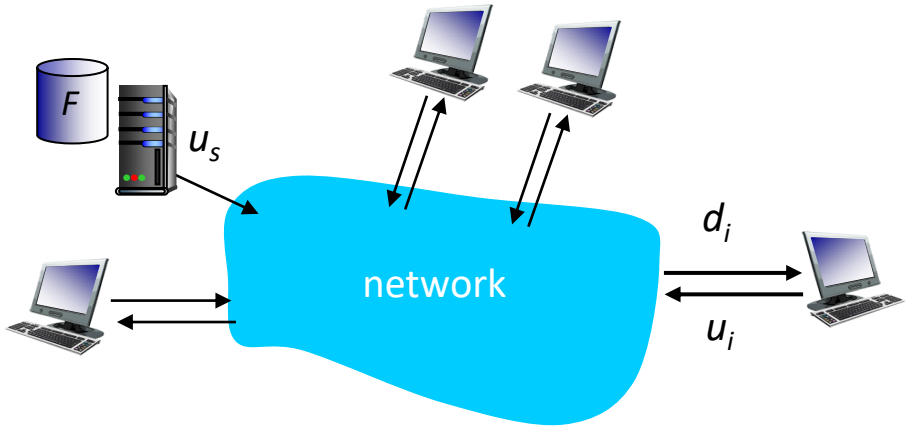
time to distribute F to N clients using client-server approach

$$D_{c\text{-}s} \geq max\{NF/u_s, F/d_{min}\}$$

increases linearly in N

File distribution time: P2P

- *server transmission:* must upload at least one copy
  - time to send one copy: F/u_s
- *client:* each client must download file copy
  - min client download time: F/d_{min}



- *clients:* as aggregate must download *NF* bits
  - max upload rate (limting max download rate) is $u_s + \Sigma u_i$

time to distribute F to N clients using P2P approach

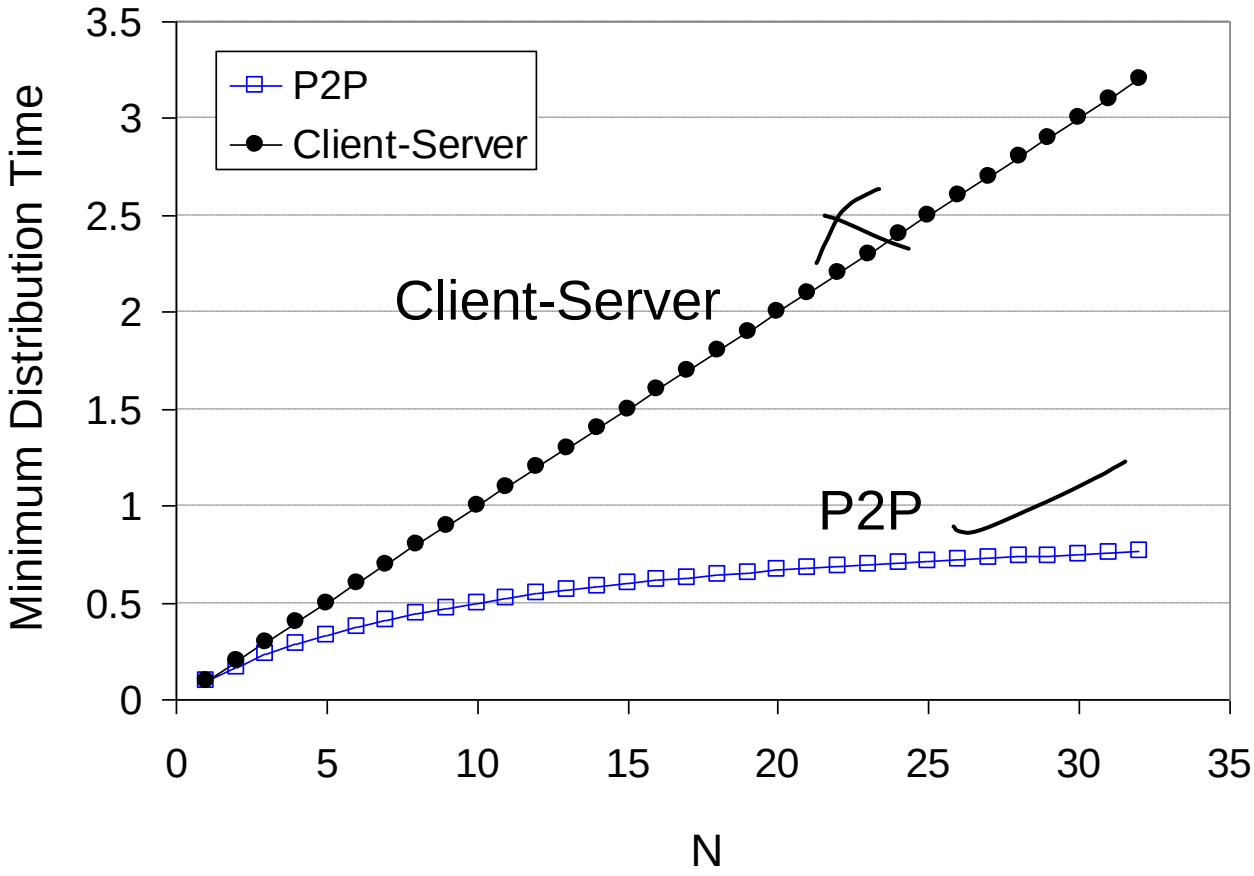
$$D_{P2P} > \underline{max}\{F/u_s, F/d_{min}, NF/(u_s + \Sigma u_i)\}$$

increases linearly in *N* …

… but so does this, as each peer brings service capacity

Client-server vs. P2P: example

client upload rate = u, F/u = 1 hour, $u_s = 10u$, $d_{min} \geq u_s$



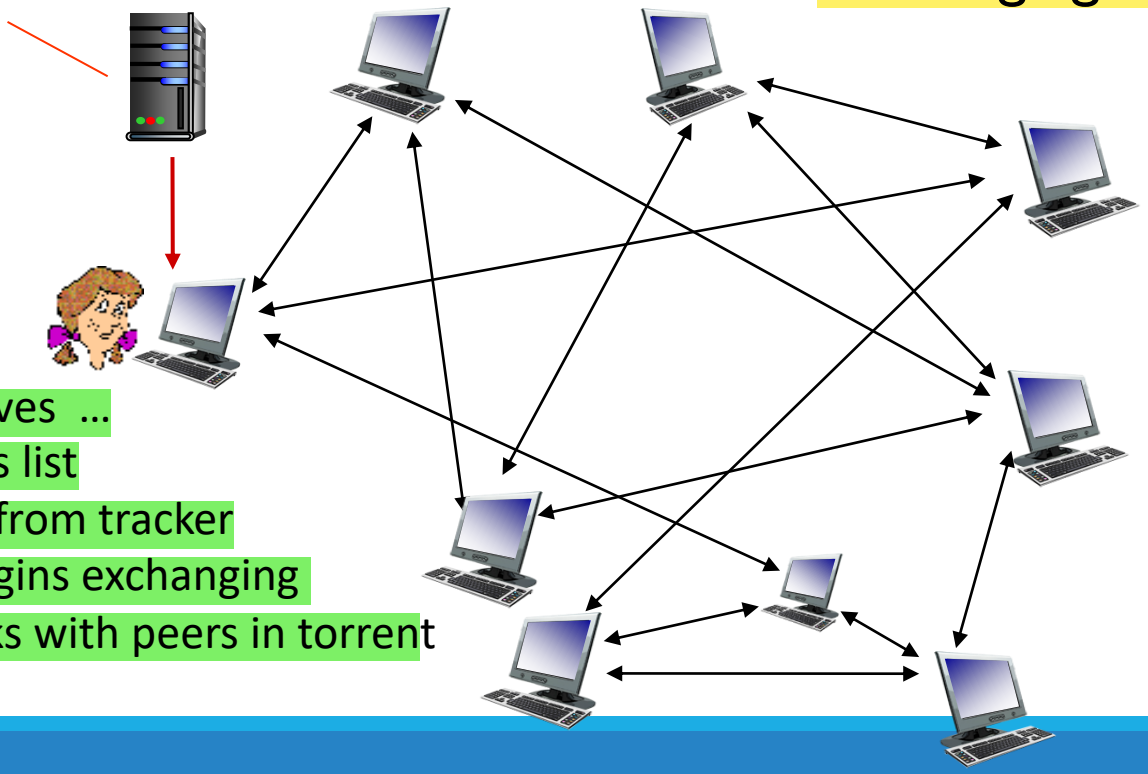
P2P file distribution: BitTorrent

- file divided into 256Kb chunks
- peers in torrent send/receive file chunks

Daymnnnnn!!

tracker: tracks peers participating in torrent

torrent: group of peers exchanging chunks of a file



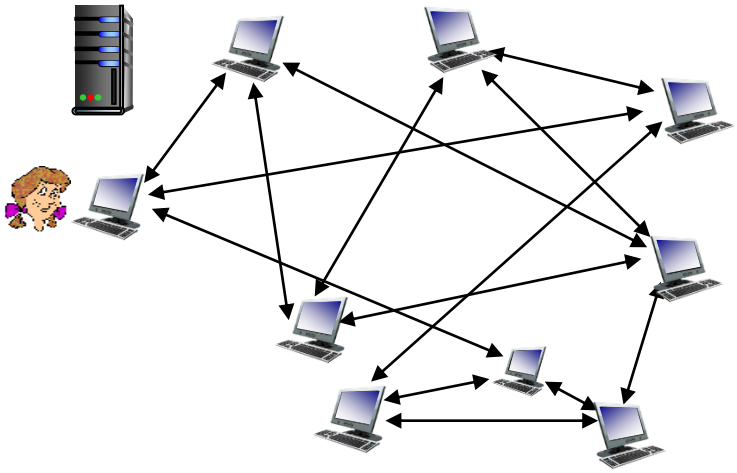
Alice arrives …

… obtains list of peers from tracker

… and begins exchanging file chunks with peers in torrent

P2P file distribution: BitTorrent

- peer joining torrent:
  - has no chunks, but will accumulate them over time from other peers
  - registers with tracker to get list of peers, connects to subset of peers ("neighbors")



- while downloading, peer uploads chunks to other peers
- peer may change peers with whom it exchanges chunks
- *churn:* peers may come and go
- once peer has entire file, it may (selfishly) leave or (altruistically) remain in torrent

BitTorrent: requesting, sending file chunks

requesting chunks:

- at any given time, different peers have different subsets of file chunks
- periodically, Alice asks each peer for list of chunks that they have
- Alice requests missing chunks from peers, rarest first

sending chunks: tit-for-tat

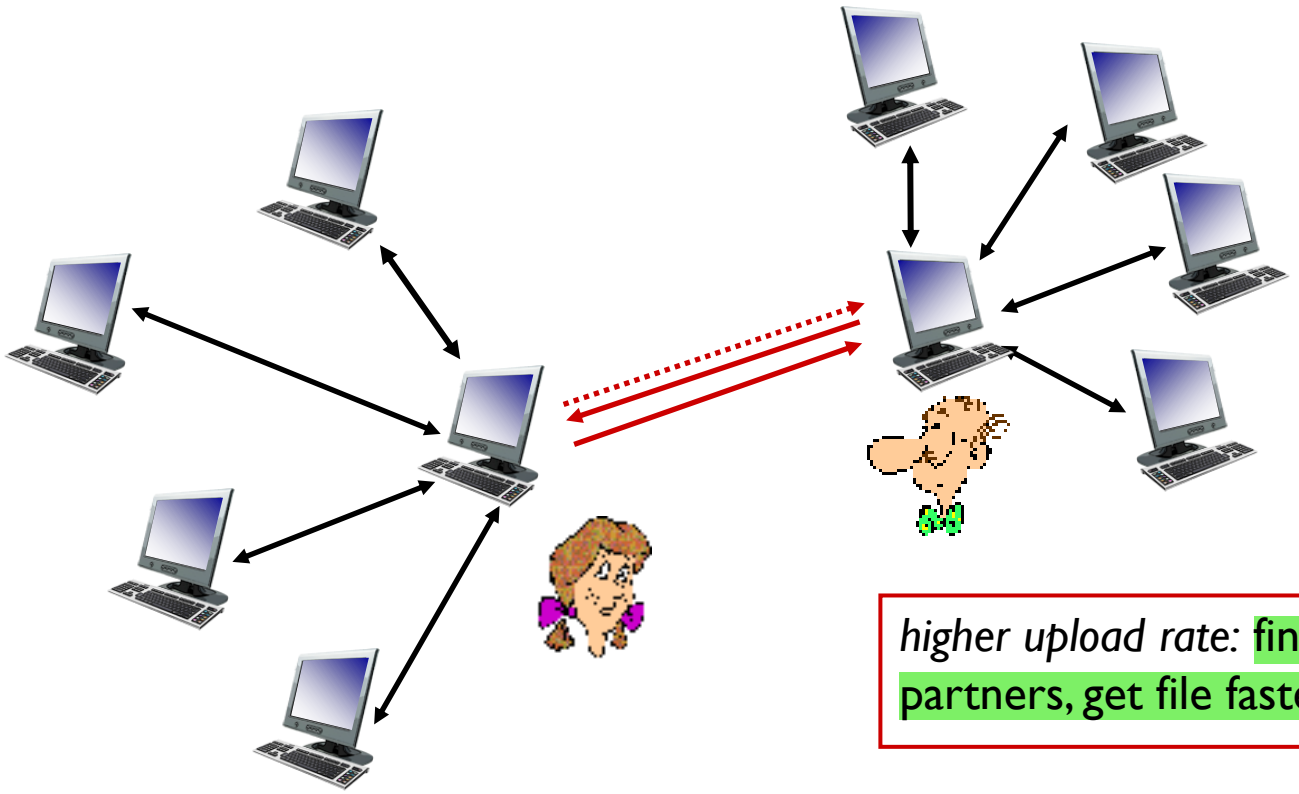
- Alice sends chunks to those four peers currently sending her chunks *at highest rate*
  - other peers are choked by Alice (do not receive chunks from her)
  - re-evaluate top 4 every10 secs
- every 30 secs: randomly select another peer, starts sending chunks
  - "optimistically unchoke" this peer
  - newly chosen peer may join top 4

BitTorrent: tit-for-tat

(1) Alice "optimistically unchokes" Bob

(2) Alice becomes one of Bob's top-four providers; Bob reciprocates

(3) Bob becomes one of Alice's top-four providers



higher upload rate: find better trading partners, get file faster !