Computer Networks

Chapter 1: Network Services, Applications Architecture, and Layers

Prof. Xudong Wang

Email: wxudong@sjtu.edu.cn

Wireless and Networking Lab
UM-SJTU Joint Institute
Shanghai Jiao Tong University
http://wanglab.sjtu.edu.cn

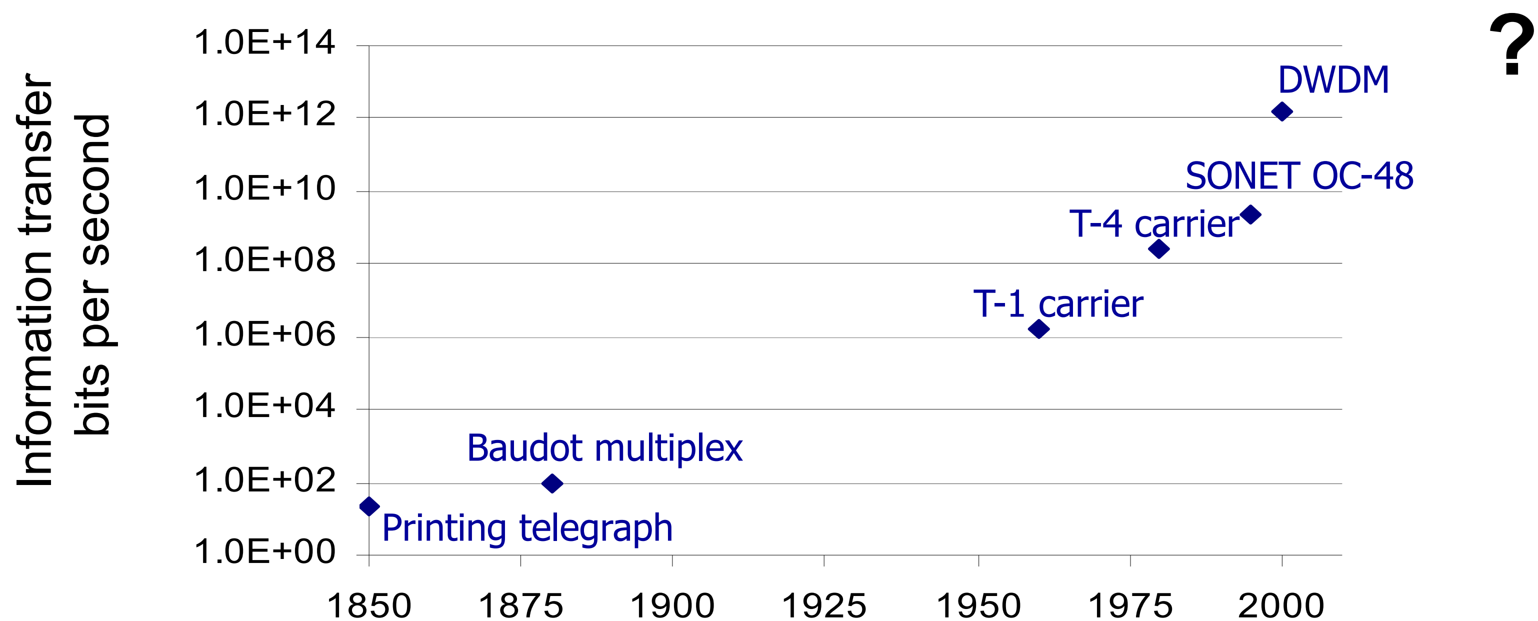
Communication Services and Applications

- Purpose
  - A communication service enables the exchange of information between users at different locations
- Examples
  - Courier: USPS, Fedex, UPS, Netflix, etc.
  - Telegraph
  - Telephone
  - Cell phone: voice, VoIP, video streaming, SMS, etc.
  - Internet based services: web-browsing, email, instant messaging, VoIP, video, IPTV, social networking, etc.

What is a Communication Network?

- The equipment (hardware & software) and facilities that provide the basic communication service
- Virtually invisible to the user; Usually represented by a cloud
- Equipment
  - Routers, servers, switches, multiplexers, hubs, modems, …
- Facilities
  - Copper wires, coaxial cables, optical fiber
  - Ducts, conduits, telephone poles …

History of Communication Networks



Telegraph networks

Telephone networks

Internet, Optical & Wireless networks

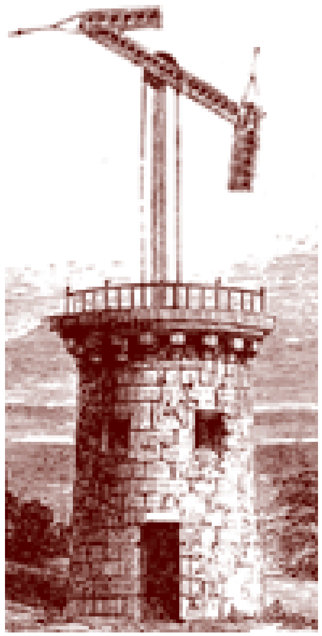
Next Generation Internet

Network Evolution

- Telegraph Networks
  - Message switching & digital transmission
- Telephone Networks
  - Circuit Switching
  - Analog transmission → digital transmission
  - Mobile communications
- Internet
  - Packet switching & computer applications
- Next-Generation Internet
  - Multiservice packet switching network

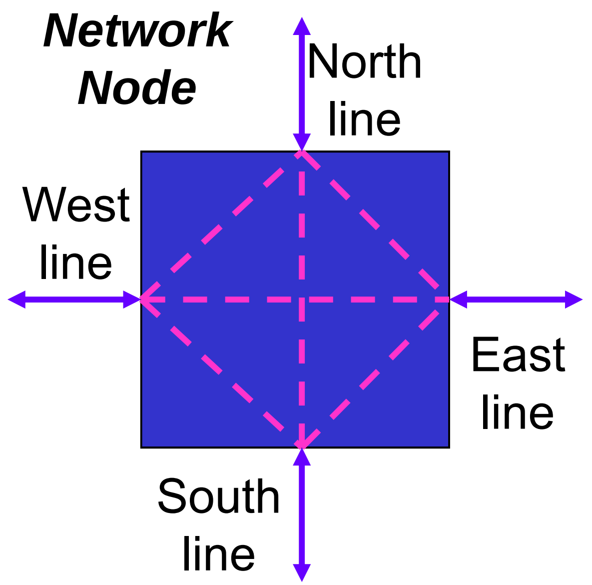
Telegraph (Visual Telegraph)

- Claude Chappe invented optical telegraph in the 1790’s
- Semaphore mimicked a person with outstretched arms with flags in each hand
- Different angle combinations of arms & hands generated hundreds of possible signals
- Code for enciphering messages kept secret
- Signal could propagate 800 km in 3 minutes!



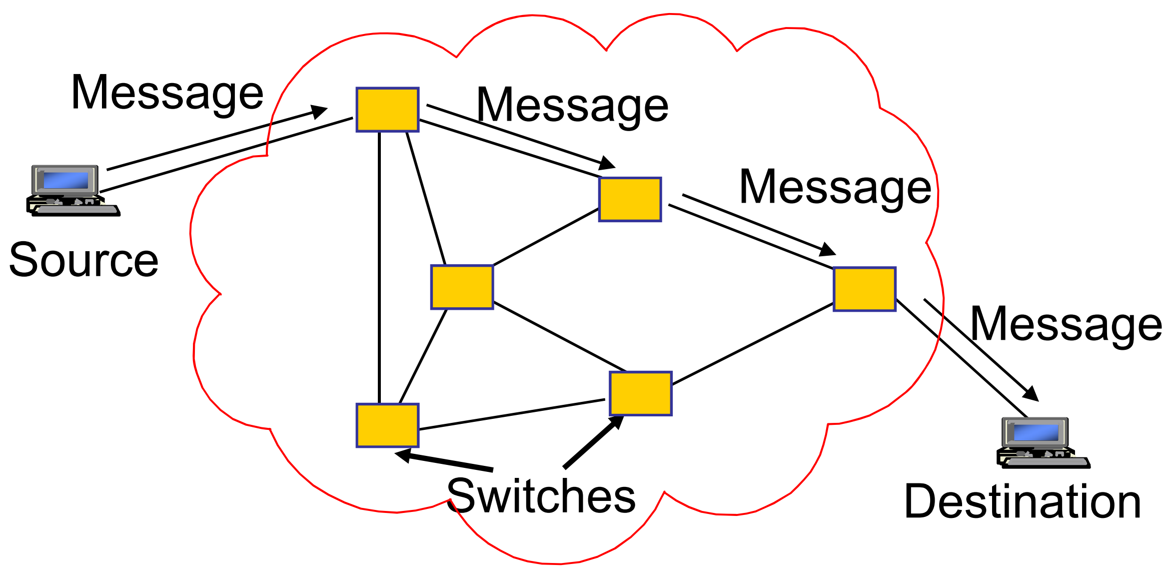
Telegraph (key concept)

- Network nodes were created where several visual telegraph lines met (Paris and other sites)
- *Store-and-Forward* Operation:
  - Messages arriving on each line were decoded
  - Next-hop in ***route*** determined by destination ***address*** of a message
  - Each message was carried by hand to next line, and stored until operator became available for next transmission



Telegraph (Electric Telegraph)

- Architecture
  - Message switching & Store-and-Forward operation
  - Key elements: Addressing, Routing, Forwarding
- Signal propagates at the speed of light
  - Approximately 2 x 10^8 meters/second in cable



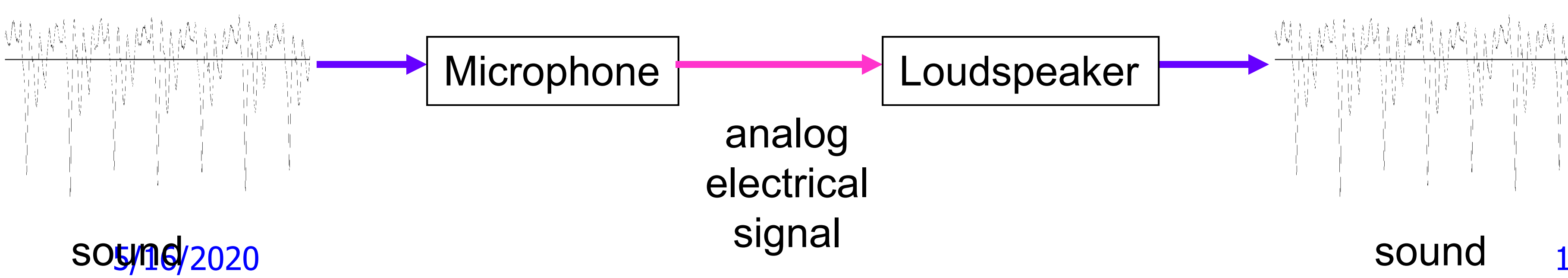
Telegraph (Key Elements)

- Digital transmission
  - Text messages converted into symbols (dots/dashes, zeros/ones) → a simple version of modulation and coding, e.g., Morse code
  - Transmission system designed to convey symbols
- Multiplexing
  - *Framing* needed to recover text characters
- Message Switching
  - Messages contain source & destination *addresses*
  - *Store-and-Forward:* Messages forwarded hop-by-hop across network
  - *Routing* according to destination address

Telephone (Bell's Telephone)

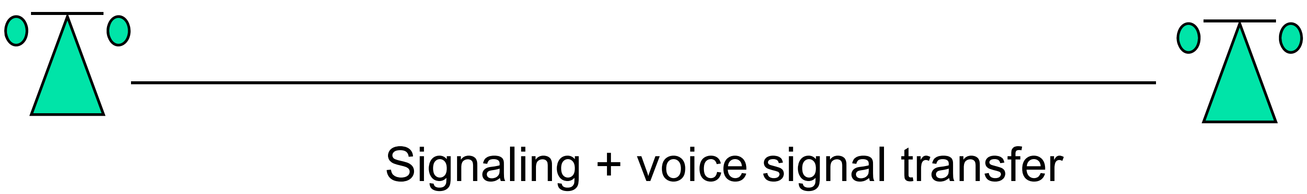
- Alexander Graham Bell (1875) working on harmonic telegraph to multiplex telegraph signals
- Discovered voice signals can be transmitted directly
  - Microphone converts voice pressure variation (sound) into *analogous* electrical signal
  - Loudspeaker converts electrical signal back into sound
- Telephone patent granted in 1876
- Bell Telephone Company founded in 1877

Signal for "ae" as in cat



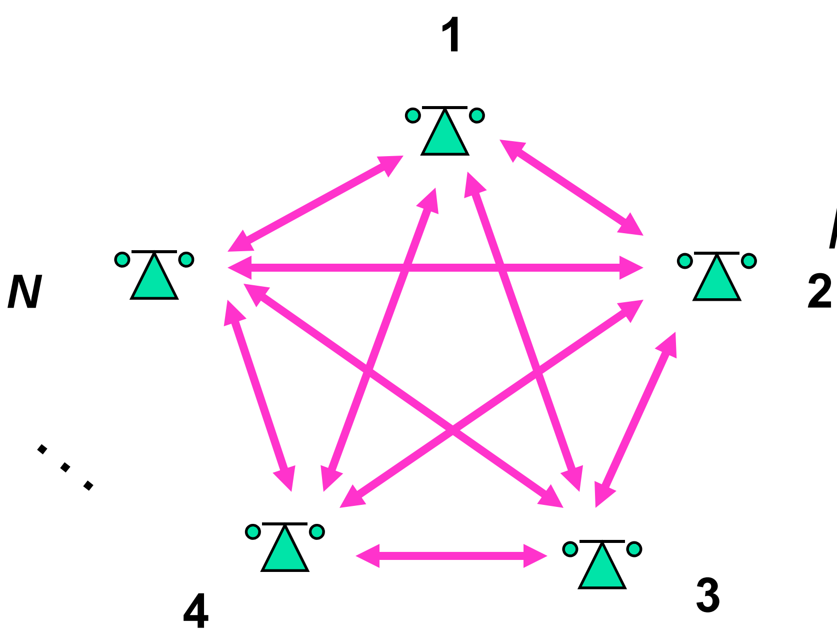
Telephone (Signaling)

- Signaling required to establish a call
  - Flashing light and ringing devices to alert the called party
  - Called party information for operator to establish calls



Telephone (The N^2 Problem)

- For *N* users to be fully connected *directly*
- Requires $N(N-1)/2$ connections
- Requires too much space for cables
- Inefficient & costly since connections not always on

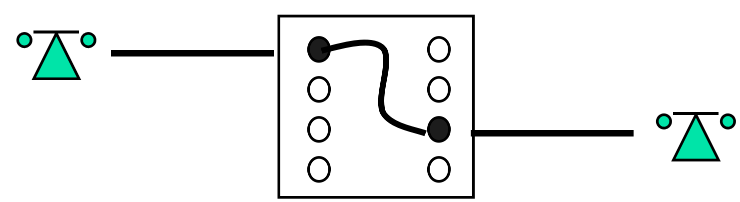


$N = 1000$

$N(N-1)/2 = 499500$

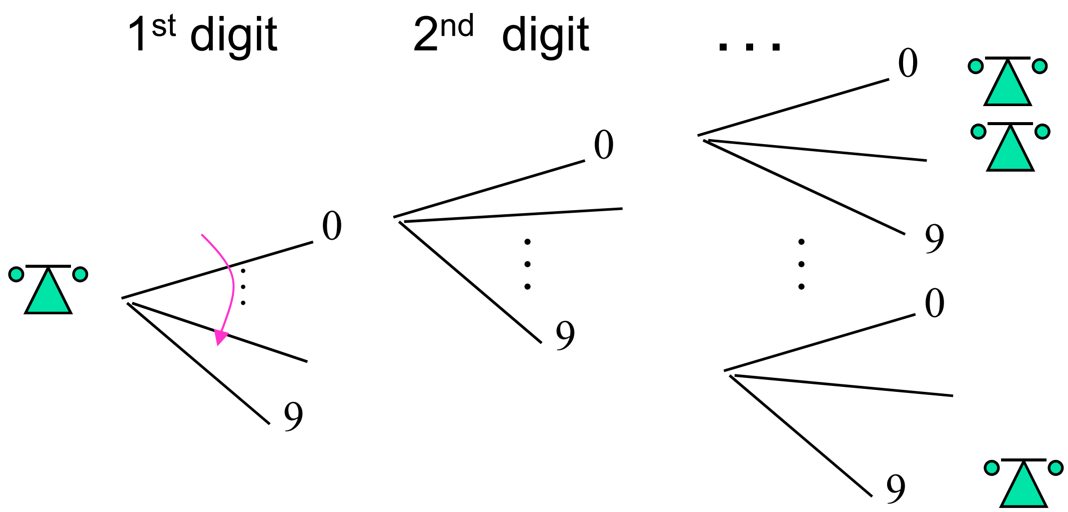
Telephone (Circuit Switching)

- Patchcord panel switch invented in 1877
- Operators connect users on demand
  - Establish *circuit* to allow electrical current to flow from inlet to outlet
- Only *N* connections required to central office

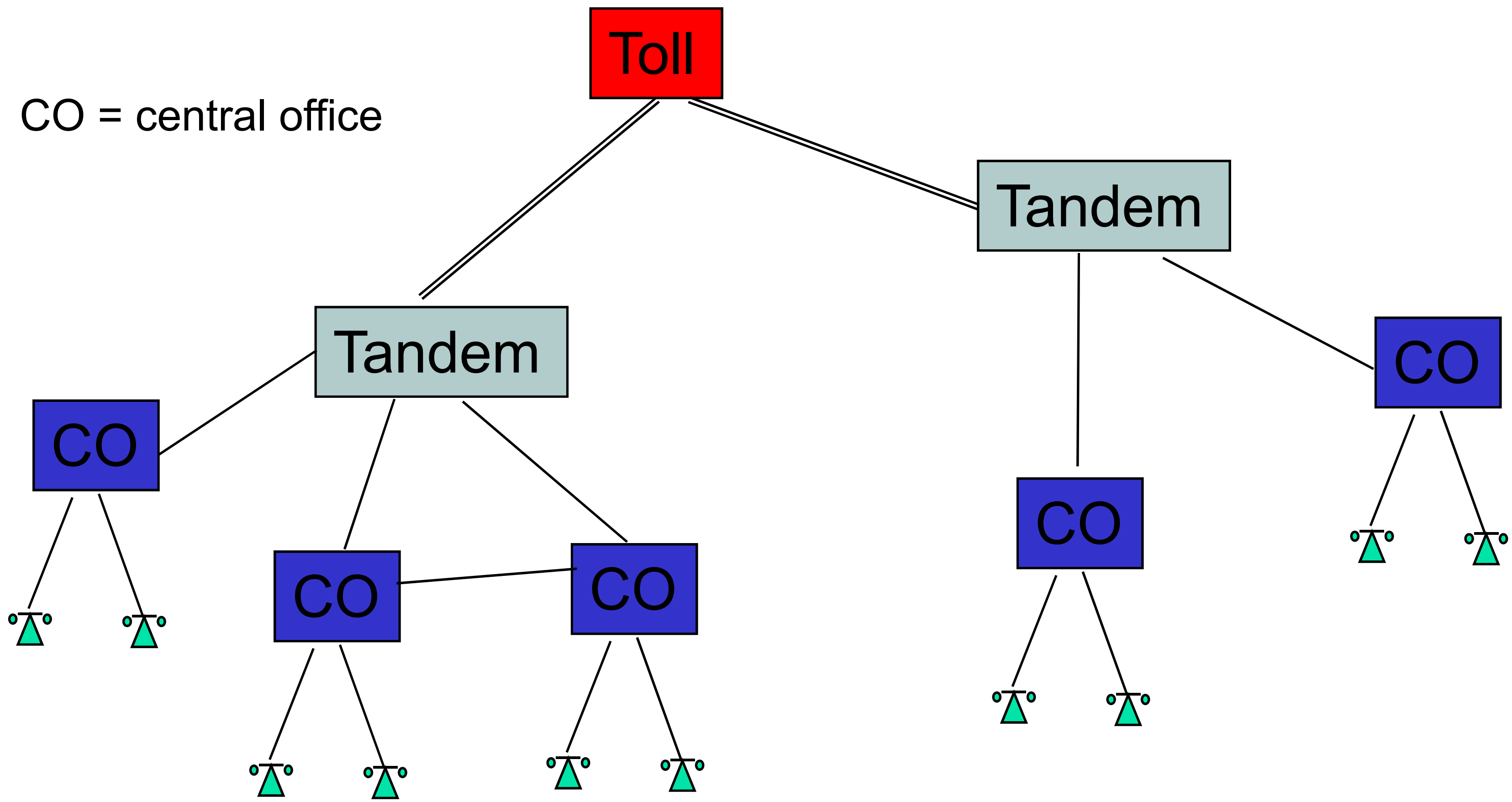


Telephone (Manual versus Automatic Circuit Switching)

- Human operators intelligent & flexible
  - But expensive and not always discreet
- Strowger invented automated switch in 1888
  - Each current pulse advances wiper by 1 position
  - User dialing controls connection setup
- Decimal telephone numbering system
- Hierarchical network structure simplifies routing
  - Area code, exchange (CO), station number



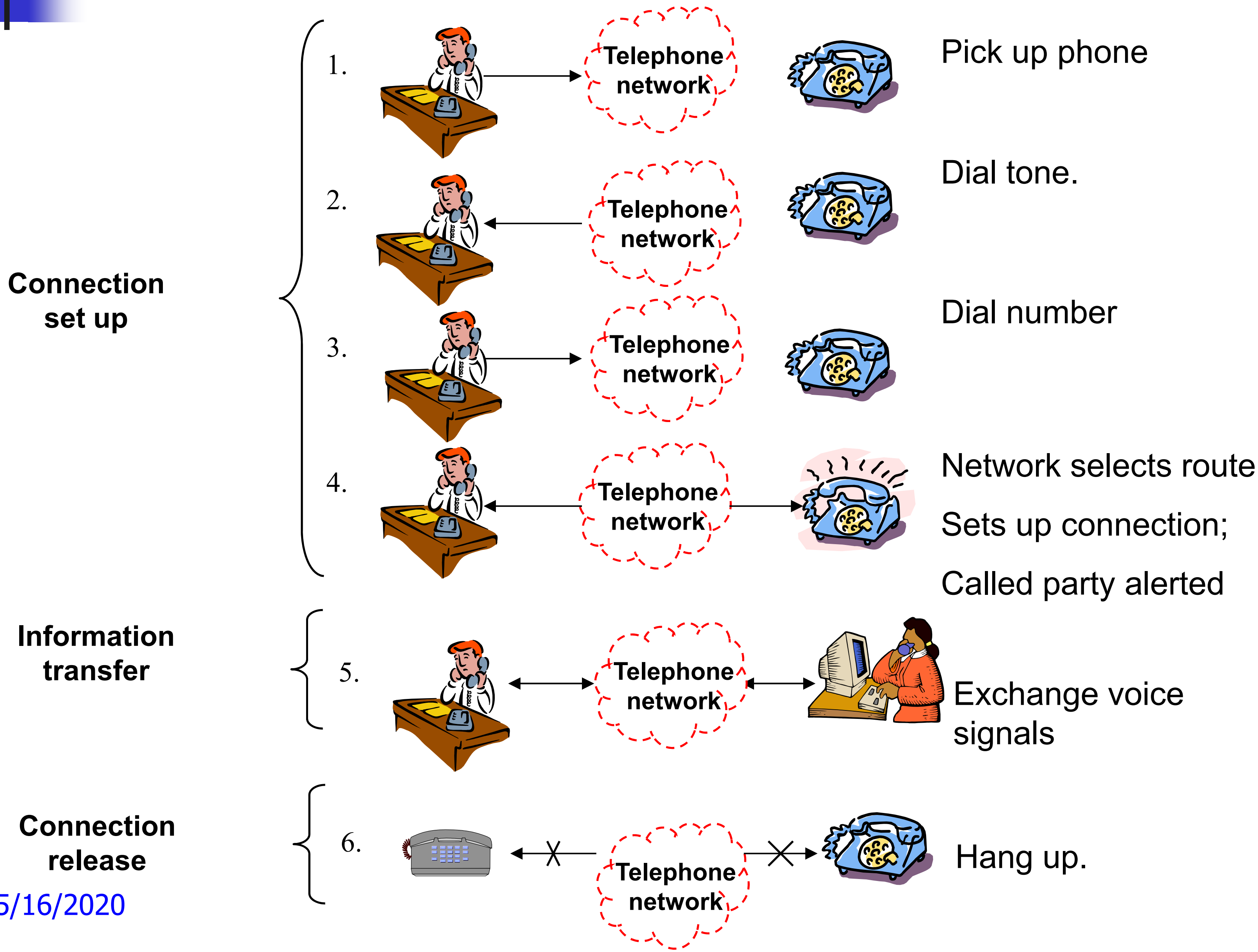
Telephone (Hierarchical Network Architecture)



Telephone subscribers connected to local CO (central office)

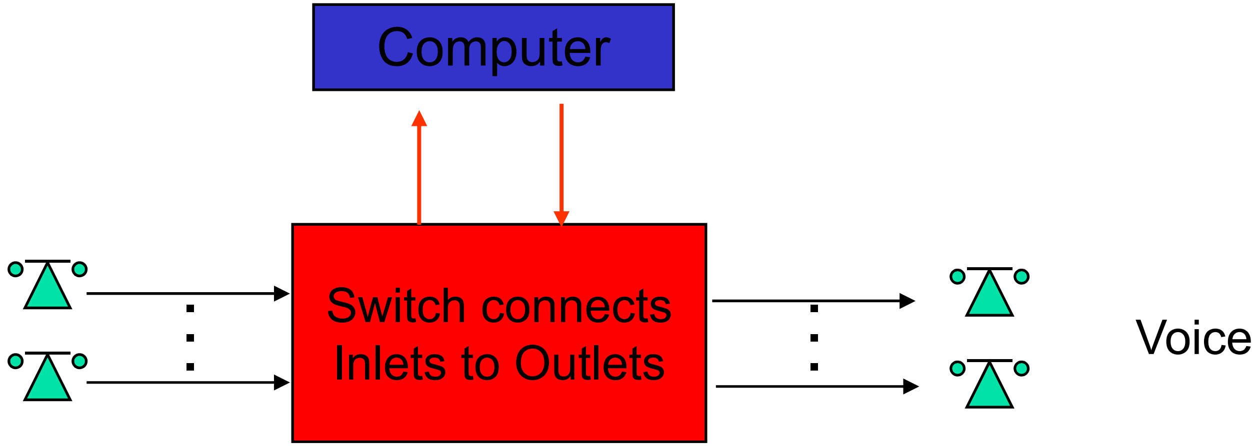
Tandem & Toll switches connect CO's

Telephone (3 Phases of a Call)



Telephone (Computer Connection Control)

- A computer controls connection in telephone switch
- Computers exchange *signaling messages* to:
  - Coordinate set up of telephone connections
  - To implement new services such as caller ID, voice mail, . . .
  - To enable *mobility and roaming in* cellular networks
- "Intelligence" inside the network
- A separate *signaling network* is required



Telephone (Digitization)

- Pulse Code Modulation digital voice signal
  - Voice gives 8 bits/sample x 8000 samples/sec = 64×10^3 bps
- Time Division Multiplexing for digital voice
  - T-1 multiplexing (1961): 24 voice signals = 1.544×10^6 bps
- Digital Switching (1980s)
  - Switch TDM signals without conversion to analog form
- Digital Cellular Telephony (1990s)
- Optical Digital Transmission (1990s)
  - One OC-192 optical signal = 10×10^9 bps
  - One optical fiber carries 160 OC-192 signals = 1.6×10^{12} bps!

All digital transmission, switching, and control

Key Elements of Today's Telephone

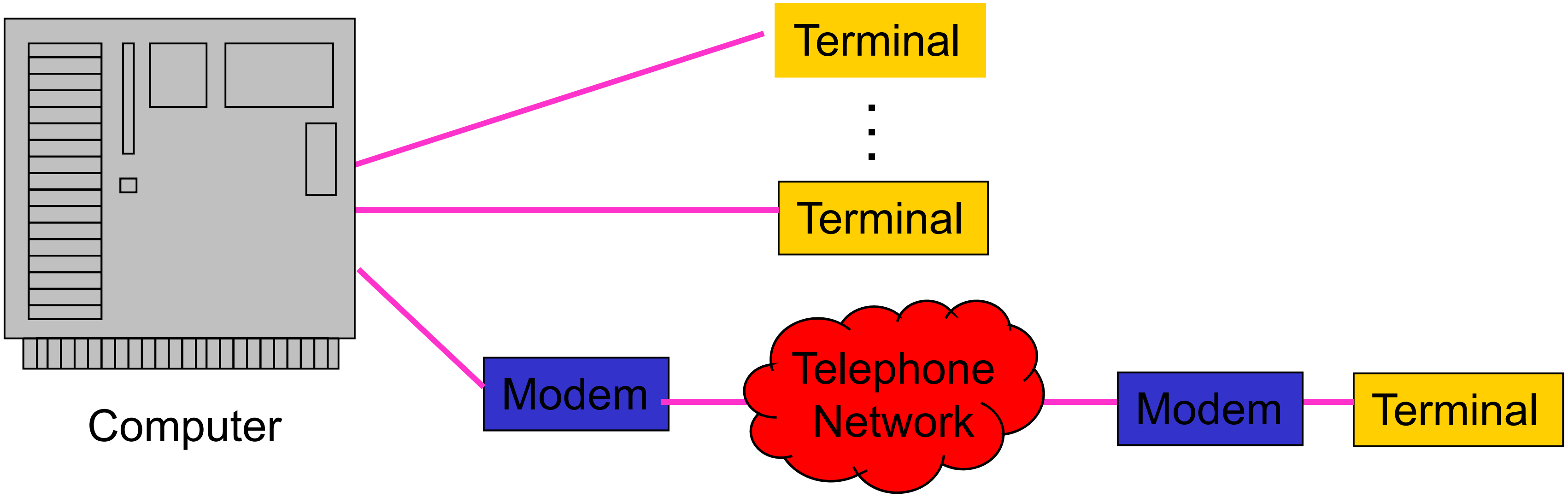
- Digital transmission & switching
  - Digital voice; Time Division Multiplexing
- Circuit switching
  - User signals for call setup and tear-down
  - Route selected during connection setup
  - End-to-end connection across network
  - Signaling coordinates connection setup
- Hierarchical Network
  - Decimal numbering system
  - Hierarchical structure; simplified routing; scalability
- Signaling Network
  - Intelligence inside the network

Computer Network Evolution

- *1950s*: Telegraph technology adapted to computers
- *1960s*: Dumb terminals access shared host computer
  - SABRE airline reservation system
- *1970s*: Computers connect directly to each other
  - ARPANET packet switching network
  - TCP/IP internet protocols
  - Ethernet local area network
- *1980s & 1990s*: New applications and Internet growth
  - Commercialization of Internet
  - E-mail, file transfer, web, P2P, . . .
  - Internet traffic surpasses voice traffic
- Today’s computer networks
  - Hybrid of different wired and wireless networks

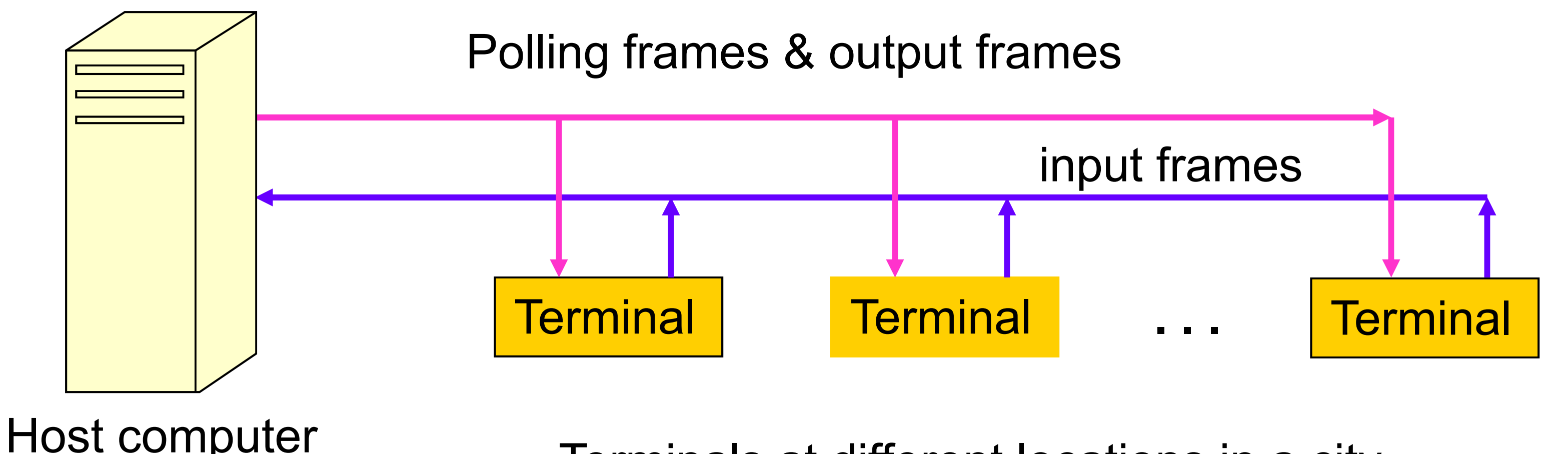
Example 1: Terminal-Oriented Networks

- Early computer systems very expensive
- Time-sharing methods allowed multiple terminals to share local computer
- Remote access via telephone modems



Example 1: Medium Access Control

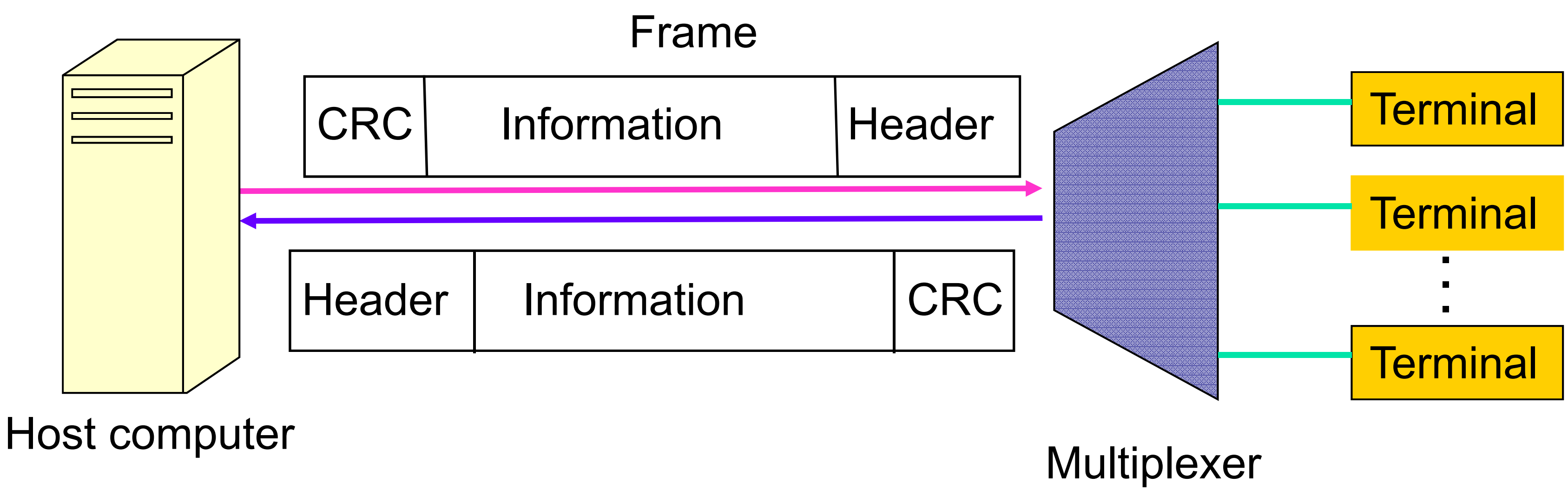
- Terminals generated messages sporadically
- Frames carried messages to/from attached terminals
- Address in frame header identified terminal
- Terminals need to be coordinated to access the same medium (the communication line to the computer)
- *Medium Access Controls* were developed to coordinate access by multiple terminals
- Example: Polling protocol on a multidrop line



Terminals at different locations in a city
Must avoid collisions on inbound line

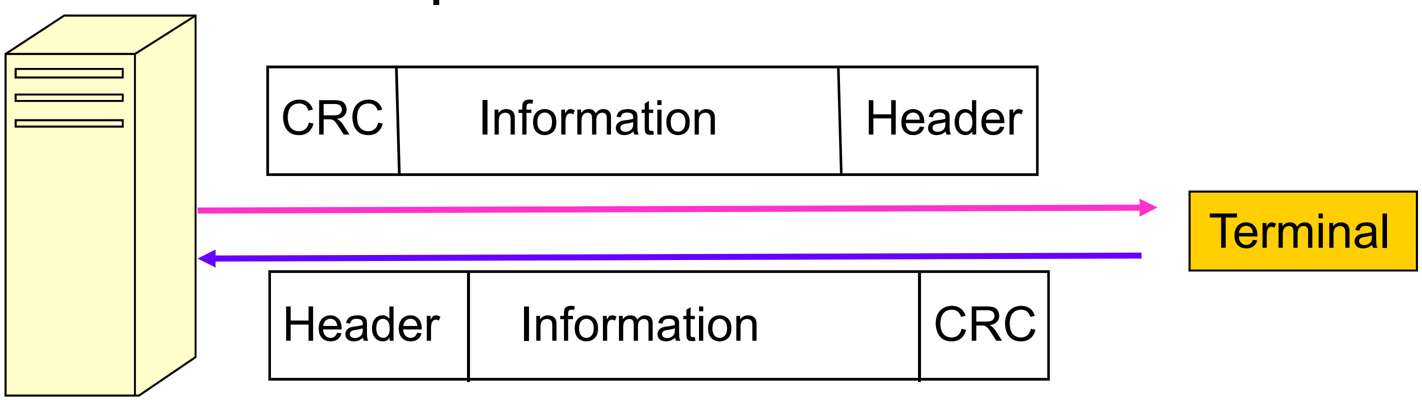
Example 1: Multiplexing

- Dedicated communication lines were expensive
- Statistical multiplexer allows a line to carry *frames* that contain messages to/from multiple terminals
- Frames are buffered at *multiplexer* until line becomes available, i.e. store-and-forward
- *Address* in frame header identifies terminal
- Header carries other *control* information



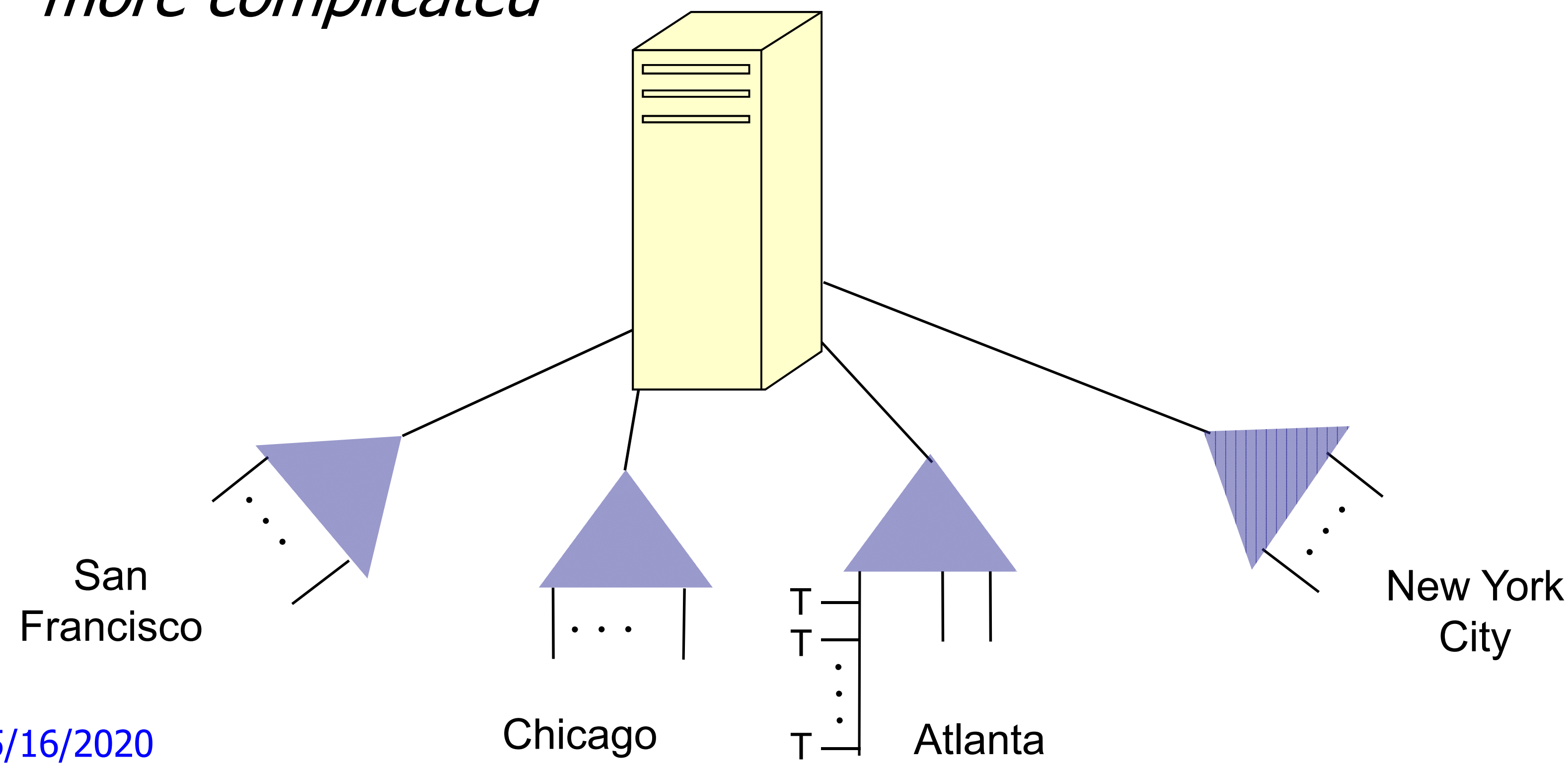
Example 1: Error Control

- Communication lines introduce errors
- Error checking codes used on frames
  - "Cyclic Redundancy Check" (CRC) calculated based on frame header and information payload
    - for *forward error control*
  - Header also carries ACK/NAK control information
    - for *ARQ (auto-repeat-request)*
- Retransmission requested when errors detected



Example 2: Tree-Topology Networks

- National & international terminal-oriented networks
- Routing is very simple (to/from host)
- Each network typically handles a single application
- *Mobility or roaming makes the problem become much more complicated*



Example 3: Computer-to-Computer Networks

- As cost of computing dropped, terminal-oriented networks viewed as too inflexible and costly
- Need to develop flexible computer networks
  - Interconnect computers as required
  - Support many applications
- Application Examples
  - File transfer between arbitrary computers
  - Execution of a program on another computer
  - Multiprocess operation over multiple computers

Example 3: The Need of Packet Switching

- Network should support multiple applications
  - Transfer arbitrary message size
  - Low delay for interactive applications
  - But in traditional store-and-forward operation, long messages induce high delay on interactive messages
- Packet switching introduced
  - Network transfers packets using store-and-forward
  - Packets have maximum length
  - Break long messages into multiple packets
- ARPANET (Advanced Research Projects Agency) testbed led to many innovations

Example 3: ARPANET Packet Switching

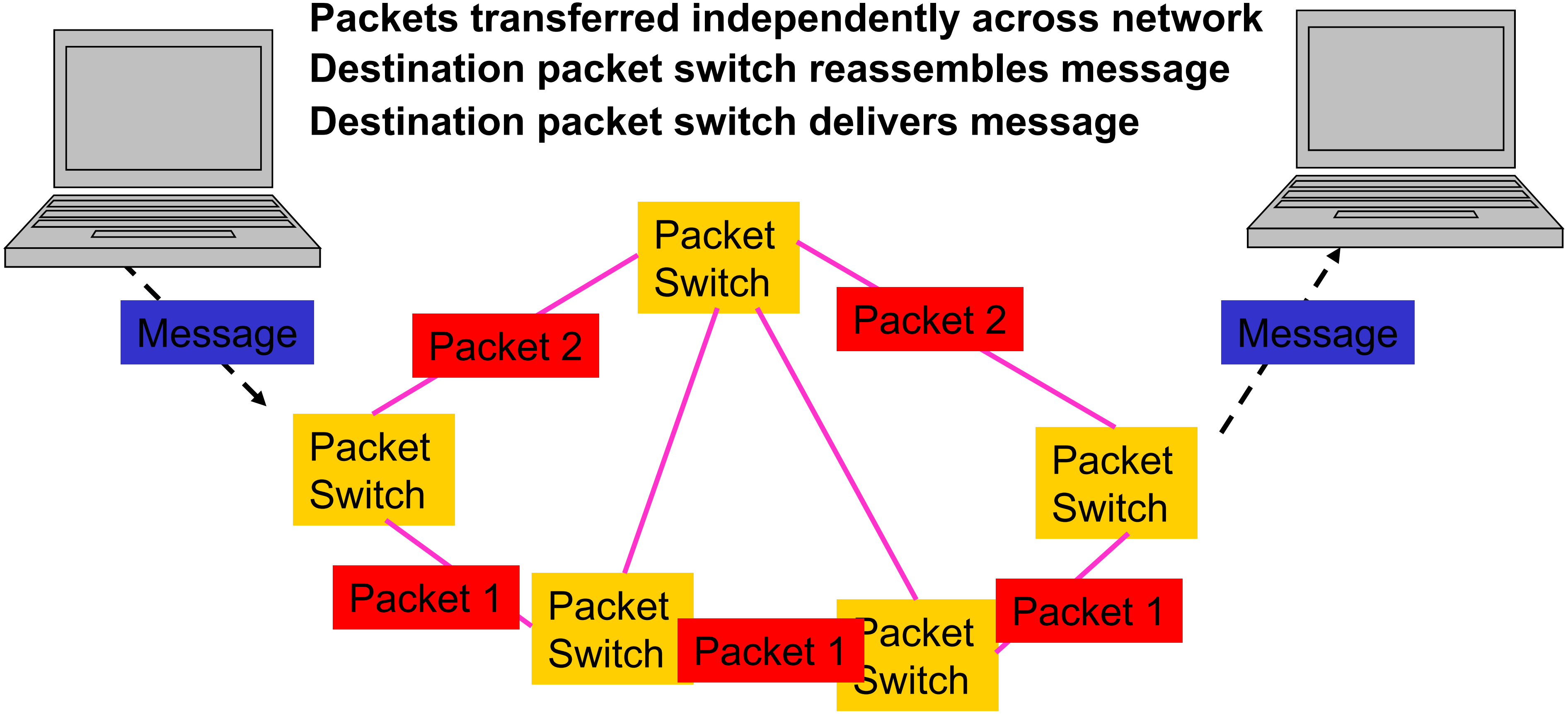
Host generates message

Source packet switch converts message to packet(s)

Packets transferred independently across network

Destination packet switch reassembles message

Destination packet switch delivers message



Example 3: ARPANET Routing

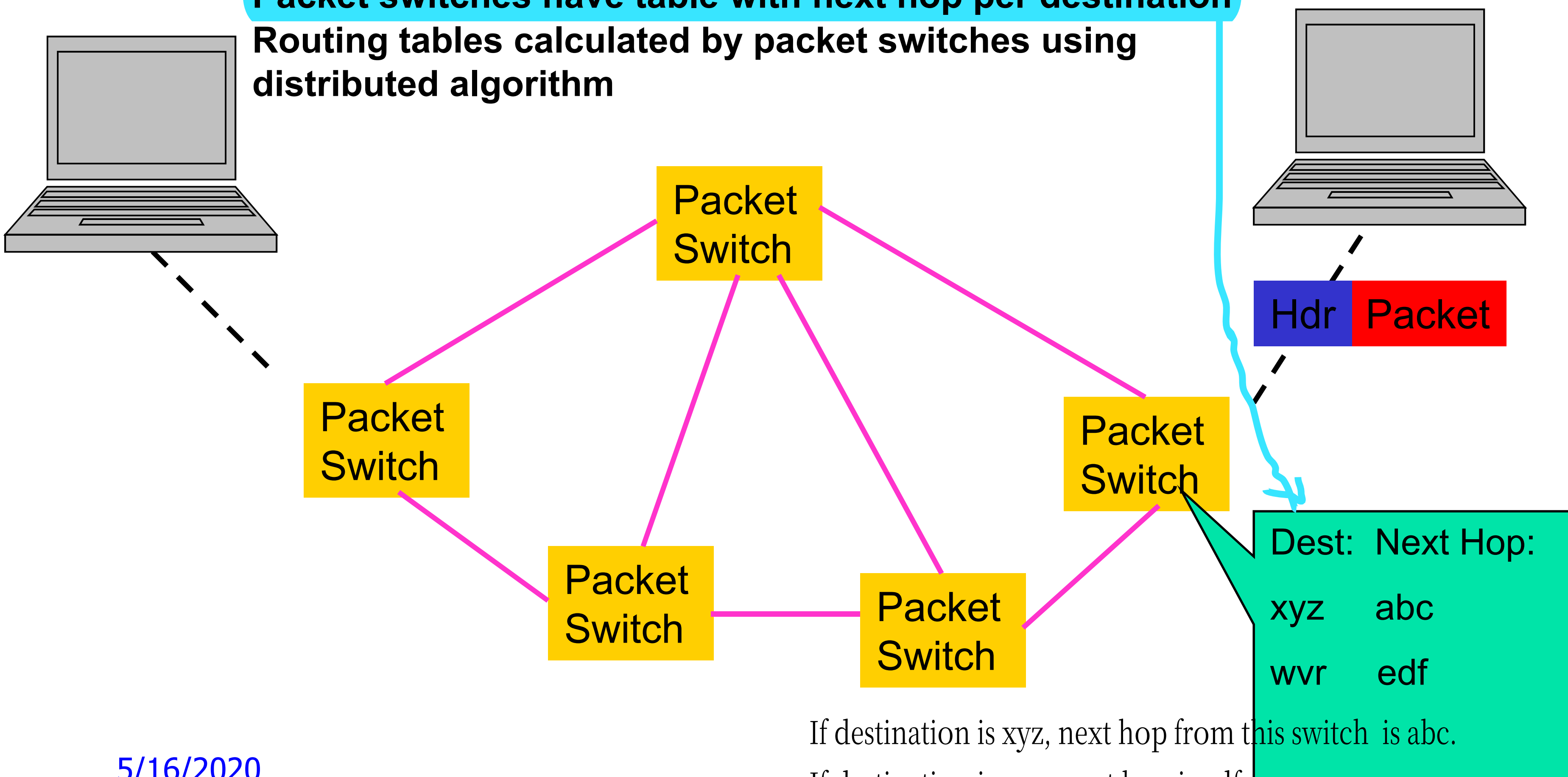
Routing is highly nontrivial in computer networks

No connection setup prior to packet transmission

Packets header includes source & destination addresses

Packet switches have table with next hop per destination

Routing tables calculated by packet switches using distributed algorithm



If destination is xyz, next hop from this switch is abc.

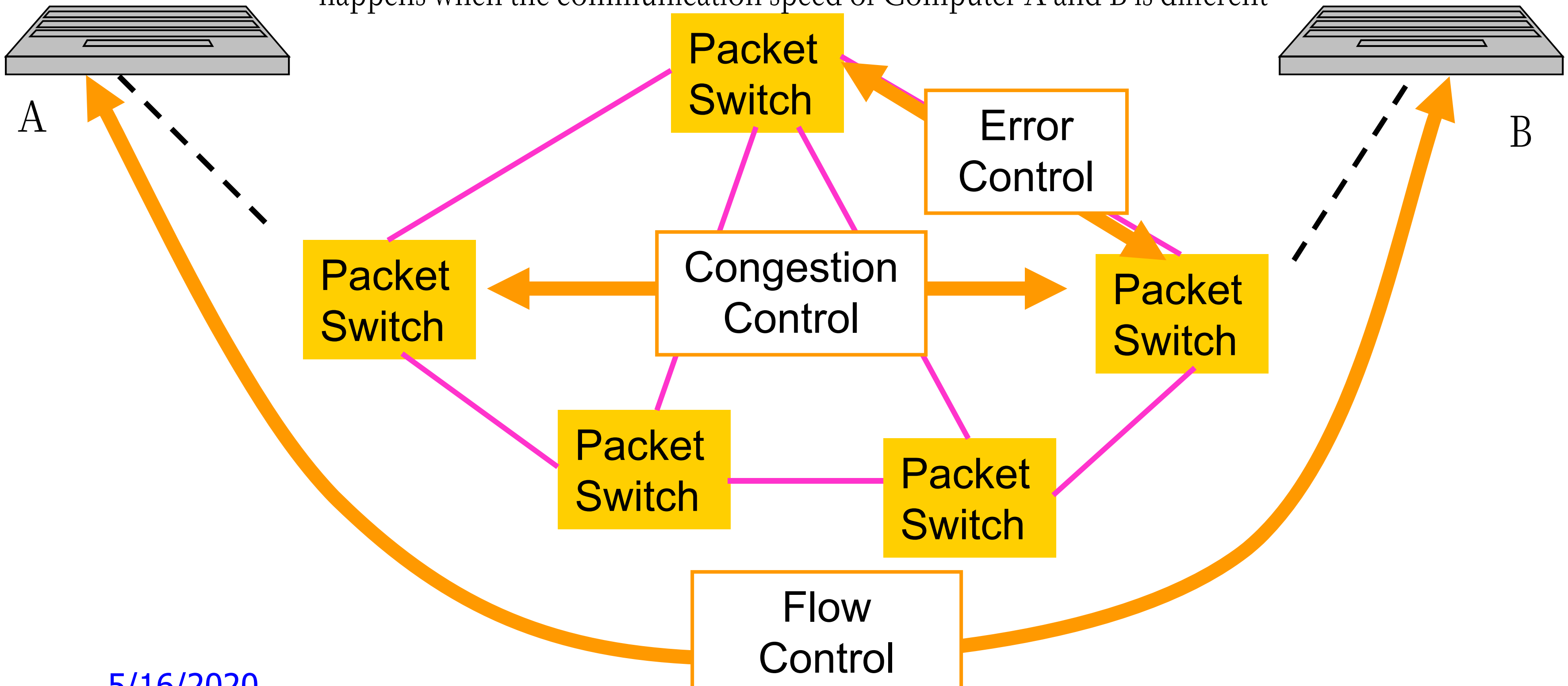
If destination is wvr, next hop is edf.

Example 3: Other ARPANET Networking Functions

Error control **between adjacent packet switches**

Congestion control **between source & destination**
packet switches limit number of packets in transit
to avoid congestions in the network and prevent losing packets due to congestions

Flow control **between host computers**
prevents buffer overflow
happens when the communication speed of Computer A and B is different

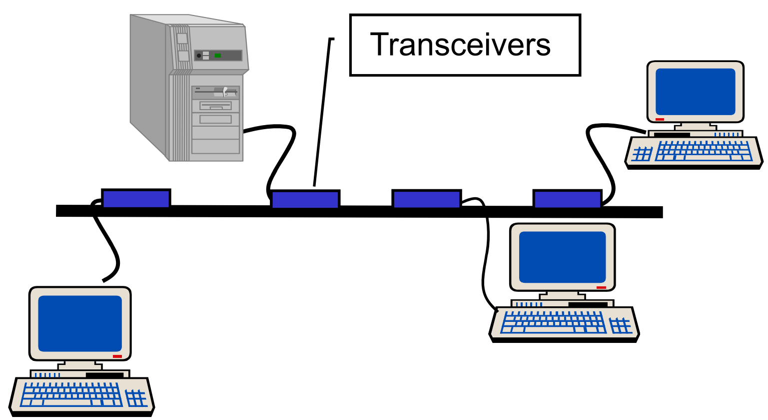


Example 4: Ethernet

- In 1980s, affordable workstations available
- Need for low-cost, high-speed networks
  - To interconnect local workstations
  - To access local shared resources (printers, storage, servers)
- Low cost, high-speed communications with low error rate possible using coaxial cable
- Ethernet is the standard for high-speed wired access to computer networks

Example 4: Ethernet MAC

- Network interface card (NIC) connects workstation to LAN
- Each NIC has globally unique address
- Frames are broadcast into coaxial cable
- NICs listen to medium for frames with their address
- Transmitting NICs listen for collisions with other stations, and abort or reschedule retransmissions

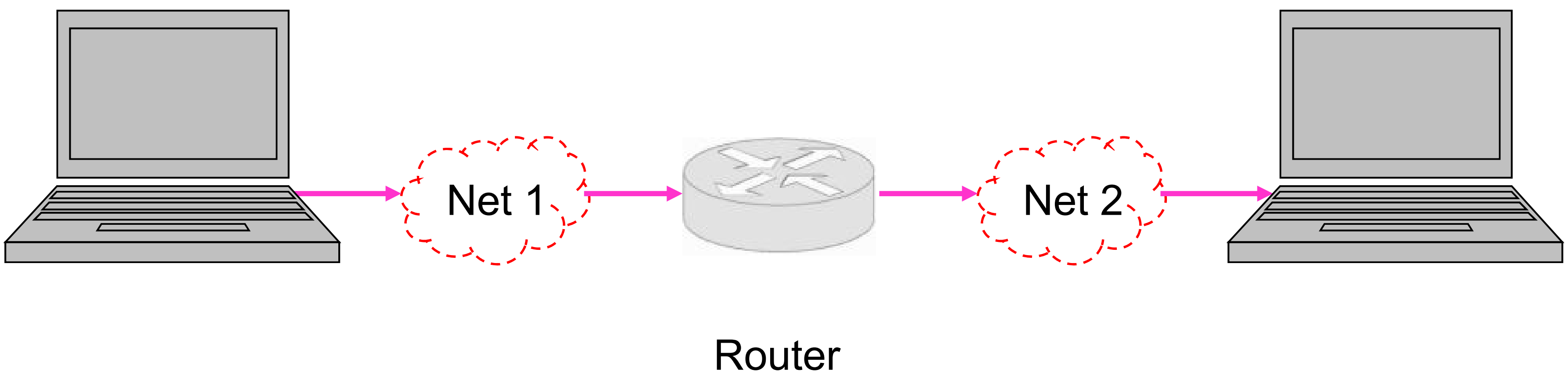


Example 5: Internet

- Different network types emerged for data transfer between computers
- ARPA also explored packet switching using satellite and packet radio networks
- Each network has its protocols and is possibly built on different technologies
- *Internetworking protocols* required to enable communications between computers attached to *different* networks
- ***Internet***: a network of networks

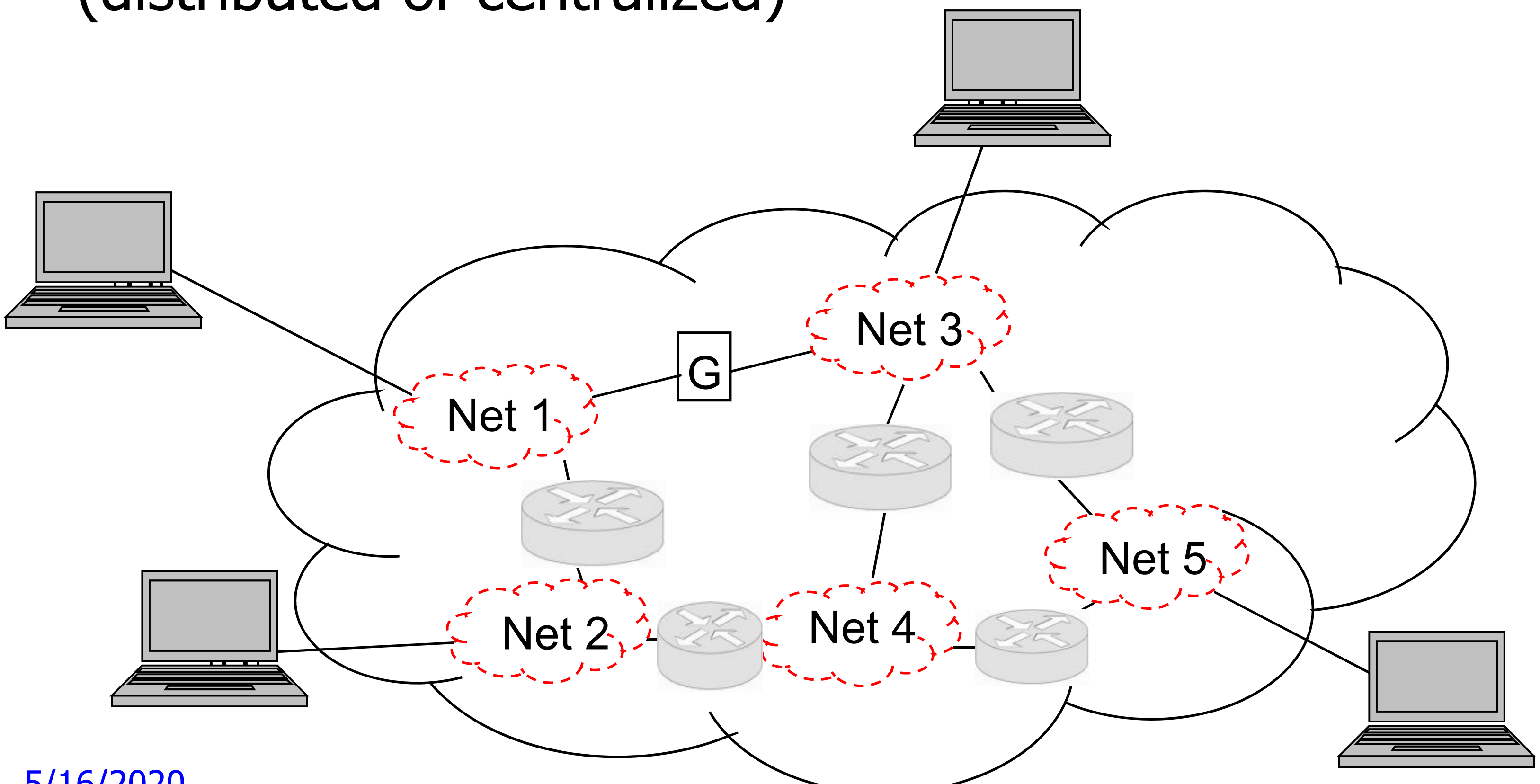
Example 5: IP

- *Routers (gateways)* interconnect different networks
- Host computers prepare IP packets and transmit them over their attached network
- Routers forward IP packets across networks
- *Best-effort* IP transfer service, no retransmission



Example 5: Internet Addressing & Routing

- Hierarchical address: Net ID + Host ID
- IP packets routed according to Net ID
- Routers compute routing tables using certain algorithms (distributed or centralized)

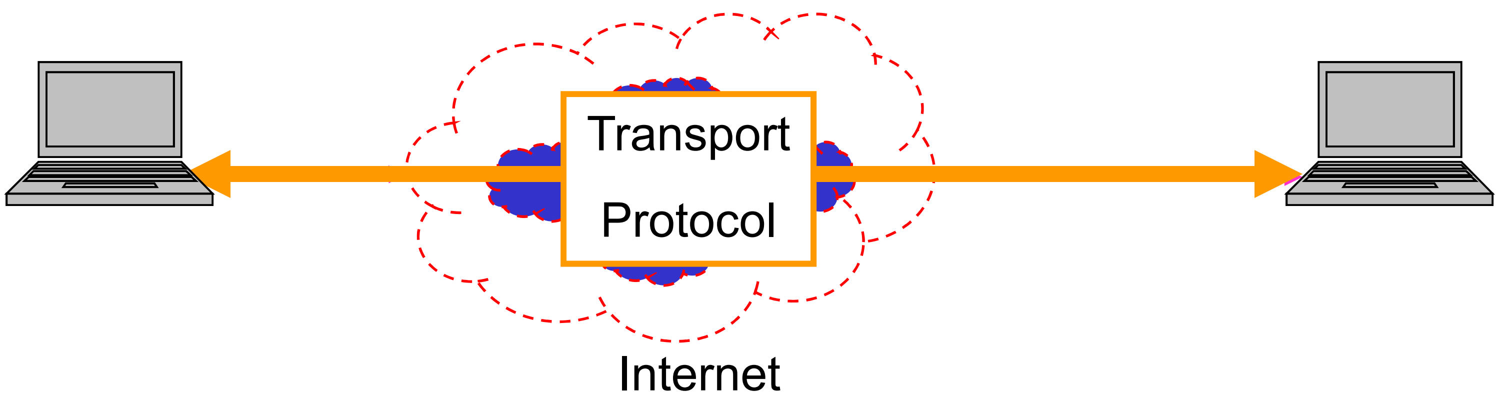


Example 5: Internet Names and IP Addresses

- Routing is done based on 32-bit IP addresses
- Dotted-decimal notation
  - 125.85.30.154
- Hosts are also identified by name
  - Easier to remember
  - Hierarchical name structure
  - E.g.: www.umji.sjtu.edu.cn
  - Domain Name System (DNS) provided conversion between names and addresses

Example 5: Transport IP Packets

- Host computers run two transport protocols on top of IP to enable process-to-process communications
- *User Datagram Protocol* (UDP) enables best-effort transfer of individual block of information
- *Transmission Control Protocol* (TCP) enables reliable transfer of a stream of bytes



Example 5: More Info on Internet

- All Internet applications run on TCP or UDP
- TCP: HTTP (web), SMTP (e-mail), FTP (file transfer, telnet (remote terminal)
- UDP: DNS, RTP (voice & multimedia)
- TCP & UDP incorporated into computer operating systems
- Any application designed to operate over TCP or UDP will run over the Internet

Elements of Computer Networks

- *Digital transmission* involves a lot of digital communication technologies
- Exchange of *frames* between adjacent equipment
  - Framing and error control
- *Medium access control* regulates sharing of broadcast medium.
- *Addresses* identify attachment to network or internet.
- Transfer of *packets* across a packet network
- *Internetworking* across multiple networks using routers
- Distributed calculation of *routing tables*

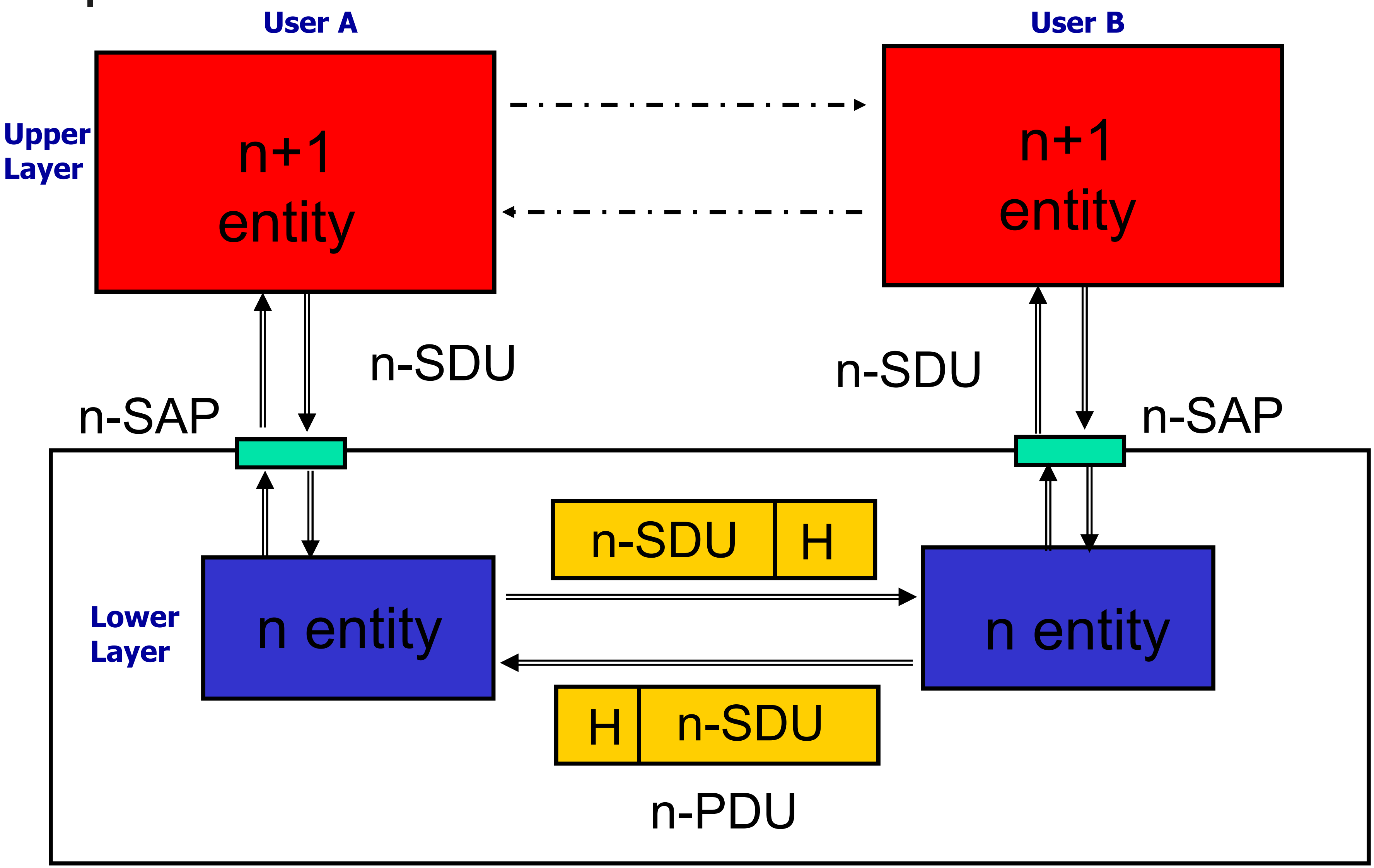
Elements of Computer Networks (continued)

- *Congestion control* inside the network
- *Segmentation and reassembly* of messages into packets at the ingress to and egress from a network or internetwork
- *End-to-end transport protocols* for process-to-process communications
- *Applications* that build on the transfer of messages between computers.
- *Intelligence is needed at the edge of the network.*

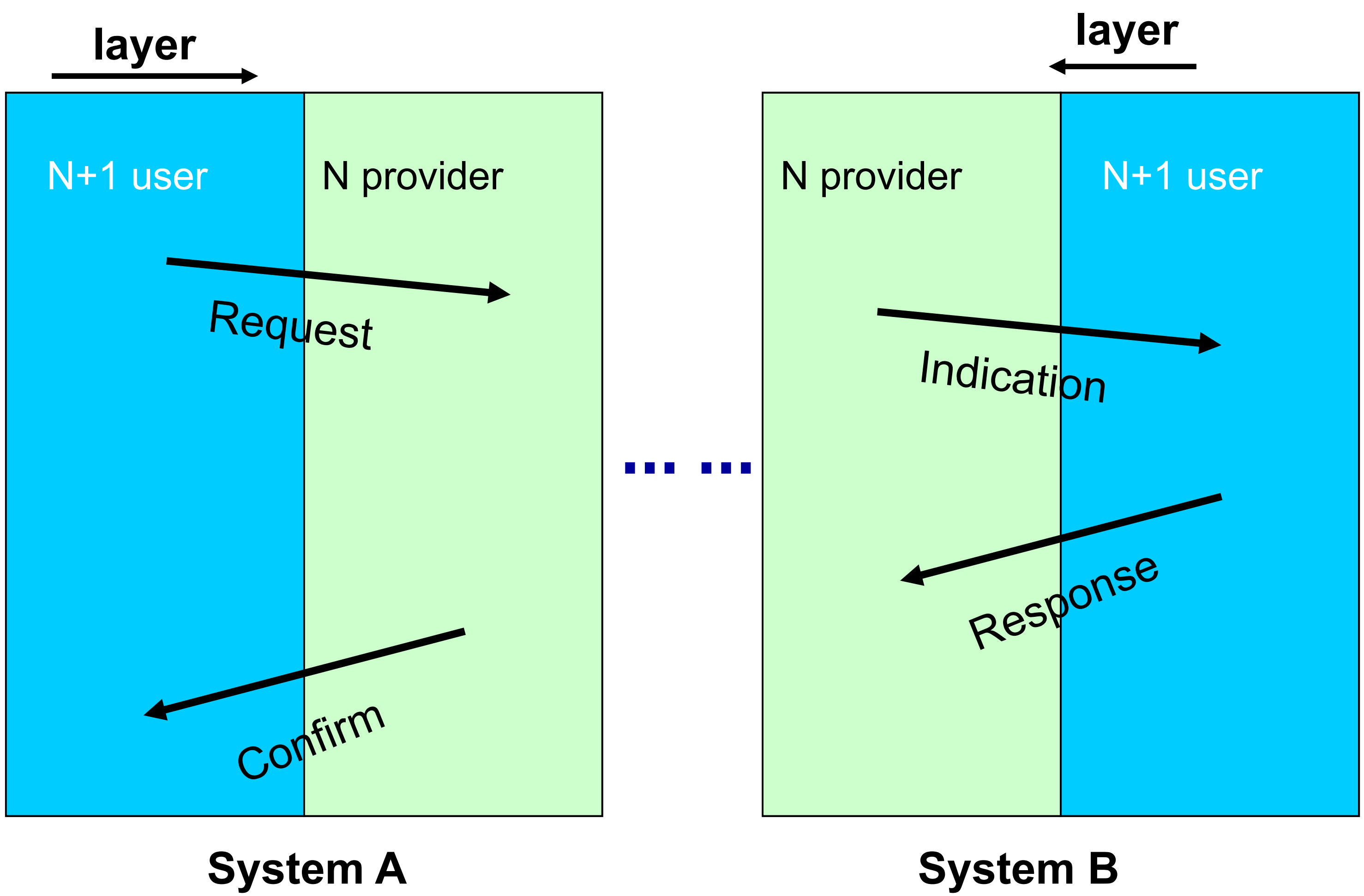
Layers, Services, and Protocols

- The overall communications process between two or more machines connected across one or more networks is very complex
- ***Layering*** partitions related communications functions into groups that are manageable
- Each layer provides a ***service*** to the layer above
- Each layer operates according to a ***protocol***

Information Flow between Two Layers



Interactions between Layers



Why Layering?

- Layering simplifies design, implementation, and testing by partitioning overall communications process into parts
- Protocol in each layer can be designed separately from those in other layers
- Protocol makes "calls" for services from layer below
- Layering provides flexibility for modifying and evolving protocols and services without having to change layers below
- Monolithic non-layered architectures are costly and inflexible

Is Layering always Good?

Protocols

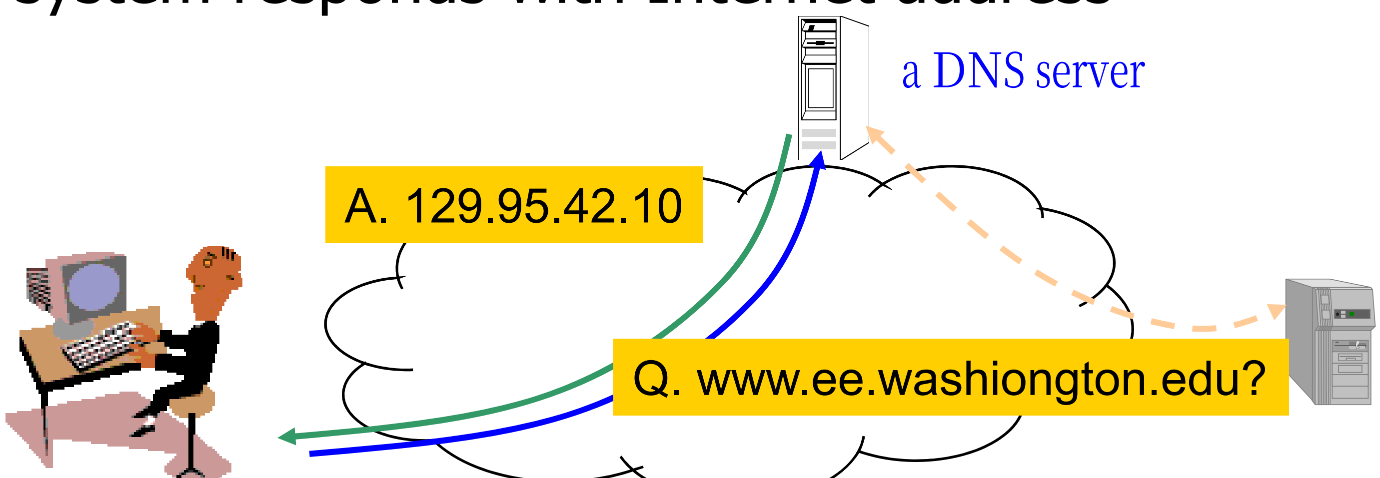
- Communications between different communication entities requires very specific unambiguous rules
- A protocol is a set of rules that governs how two or more communicating parties are to interact
- Examples
  - Internet Protocol (IP)
  - Transmission Control Protocol (TCP)
  - HyperText Transfer Protocol (HTTP)
  - Simple Mail Transfer Protocol (SMTP)

An Example on Interactions between Layers: Web Browsing

- World Wide Web allows users to access resources (i.e. documents) located in computers connected to the Internet
- Documents are prepared using HyperText Markup Language (HTML)
- A browser application program is used to access the web
- The browser displays HTML documents that include *links* to other documents
- Each link references a *Uniform Resource Locator* (URL) that gives the name of the machine and the location of the given document

Step 1: Get the IP Address of the Link

- User clicks on http://www.ee.washington.edu/
- URL contains Internet name of machine (www.ee.washiongton.edu), but not Internet address
- Internet needs Internet address to send information to a machine
- Browser software uses Domain Name System (DNS) protocol to send query for Internet address
- DNS system responds with Internet address



DNS Protocol

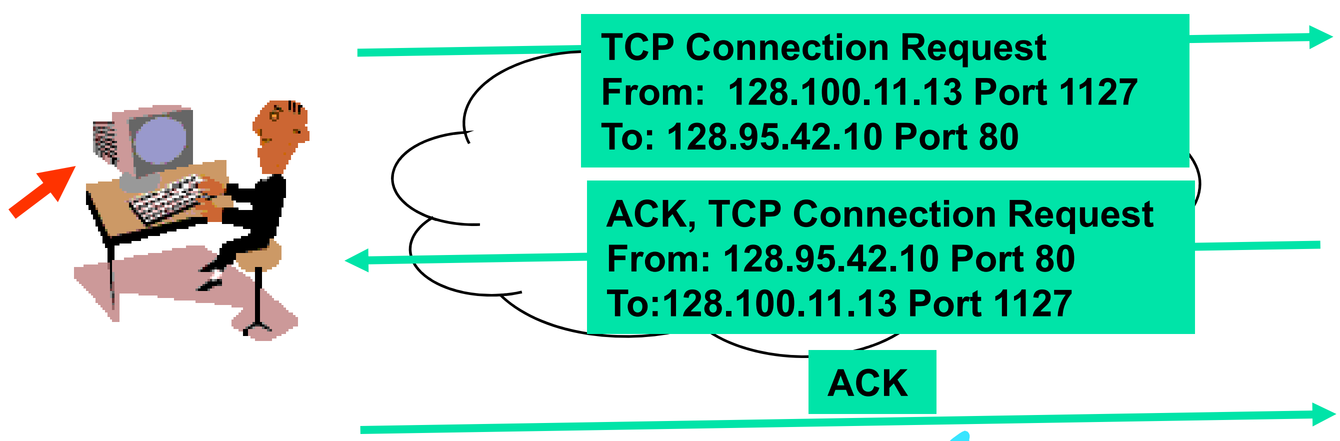
- DNS protocol is an application layer protocol
- DNS is a distributed database that resides in multiple machines in the Internet
- DNS protocol allows queries of different types
  - Name-to-address or Address-to-name
  - Mail exchange
- DNS usually involves short messages and so uses service provided by UDP
- Well-known port 53

UDP Protocol

- UDP is a transport layer protocol
- Provides *best-effort datagram service* between two processes in two computers across the Internet
- Port numbers distinguish various processes in the same machine
- UDP is *connectionless*
- Datagram is sent immediately
- Quick, simple, but not reliable

Step 2: Access the Link, but setup the link first

- Browser software uses HyperText Transfer Protocol (HTTP) to send request for document
- HTTP server waits for requests by listening to a well-known port number (80 for HTTP)
- HTTP client sends request messages through an "ephemeral port number," e.g. 1127
- HTTP needs a Transmission Control Protocol (TCP) connection between the HTTP client and the HTTP server to transfer messages reliably



Three-way Handshake

TCP Protocol

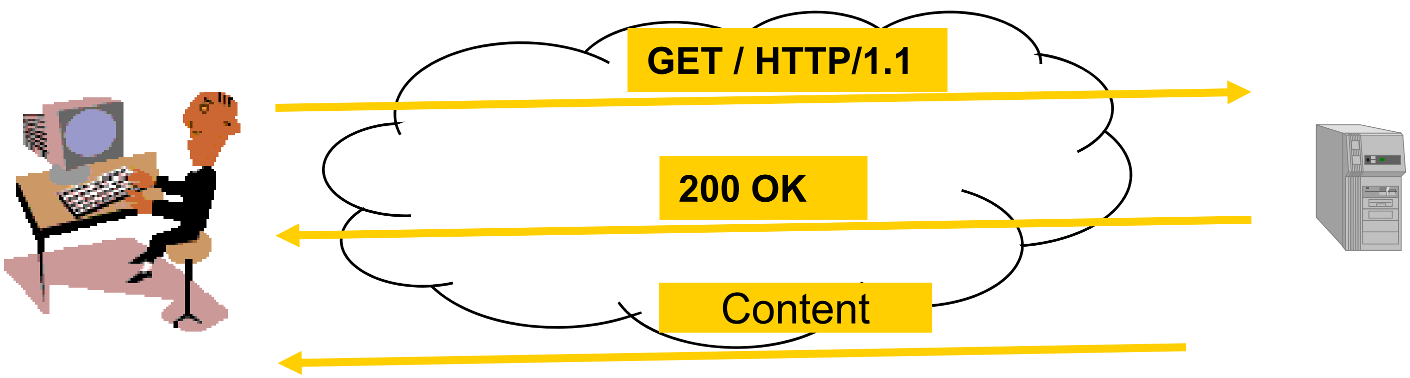
- TCP is a transport layer protocol
- Provides *reliable byte stream service* between two processes in two computers across the Internet
- Sequence numbers keep track of the bytes that have been transmitted and received
- Error detection and retransmission used to recover from transmission errors and losses
- TCP is *connection-oriented*: the sender and receiver must first establish an association and set initial sequence numbers before data is transferred
- Connection ID is specified uniquely by

(send port #, send IP address, receive port #, receiver IP address)

HTTP for Accessing the Web

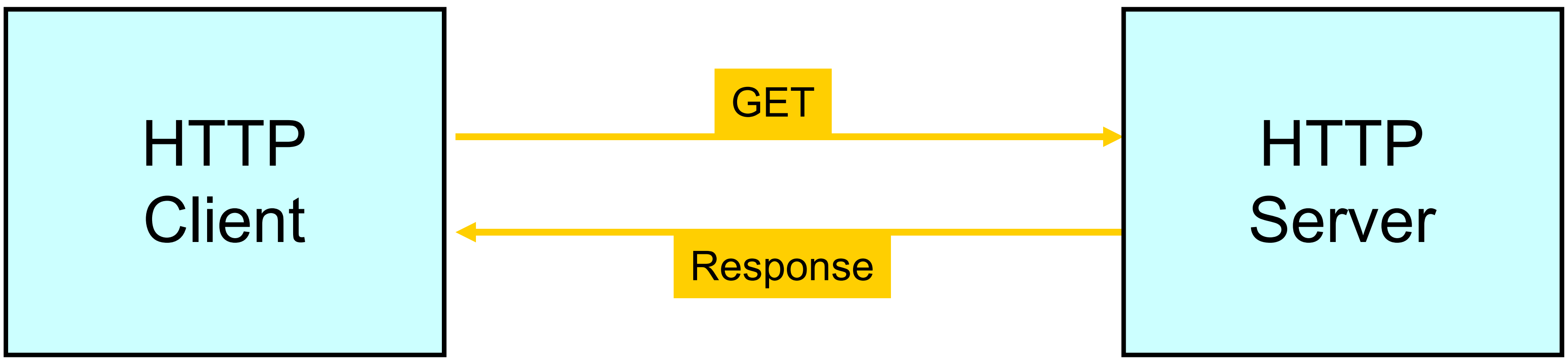
- HTTP client sends its request message: “GET …”
- HTTP server sends a standard status response: “200 OK”
- HTTP server sends requested file
- Browser displays document

- *Clicking a link sets off a chain of events across the Internet!*

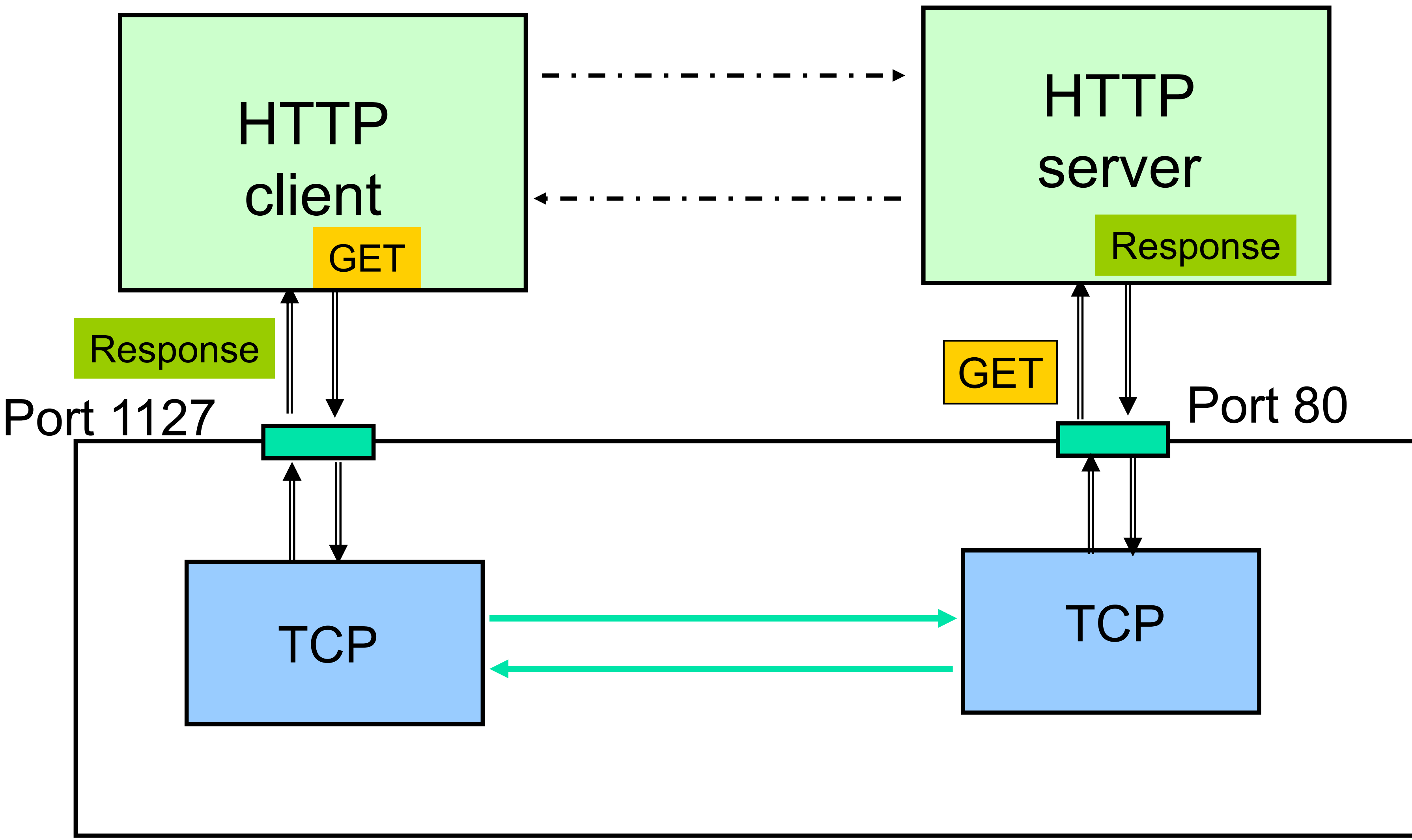


More Details on HTTP

- HTTP assumes messages can be exchanged directly between HTTP client and HTTP server
- In fact, HTTP client and server are processes running in two different machines across the Internet
- HTTP uses the reliable stream transfer service provided by TCP



HTTP/TCP Interactions



Well-Known Protocol Reference Models

- OSI 7-Layer Model
- LAN Model
- Internet Model
- Hybrid Model
- ATM Model   Asynchronous Transmission Mode
- Other Models

- Question:
  - 1. Why do we need a protocol model?
  - 2. Why are there so many different models?

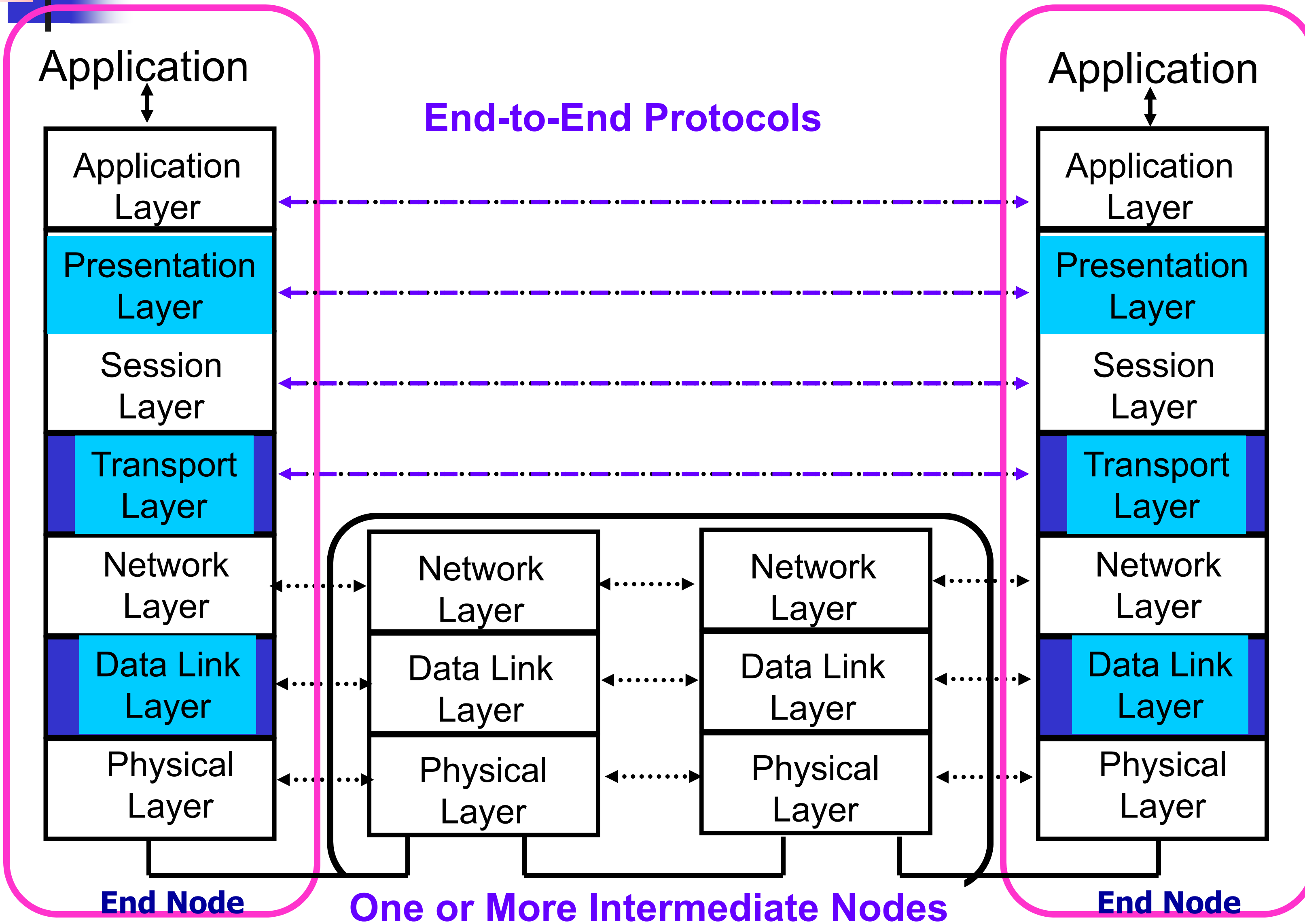
Open System Interconnection (OSI)

- By the 1970s every computer vendor had developed its own proprietary layered network architecture
- Problem: computers from different vendors could not be networked together
- Network architecture:
  - Definition of all the layers
  - Design of protocols for every layer
- Open Systems Interconnection (OSI) was an international effort by the International Organization for Standardization (ISO) to enable multivendor computer interconnection

OSI Protocol Model

- Describes a seven-layer abstract reference model for a network architecture
- Purpose of the reference model was to provide a framework for the development of protocols
- OSI also provided a unified view of layers, protocols, and services, which is still in use in the development of new protocols
- Detailed standards were developed for each layer, but most of these are not in use
- TCP/IP protocols preempted deployment of OSI protocols

The Architecture of the OSI Model



OSI Layers: Physical

- Transfers bits across link
- Definition & specification of the physical aspects of a communications link
  - Mechanical: cable, plugs, pins...
  - Electrical/optical: modulation, signal strength, voltage levels, bit times, …
  - Functional/procedural: how to activate, maintain, and deactivate physical links…
- Examples: Ethernet, DSL, cable modem, telephone modems…
- Media: Twisted-pair cable, coaxial cable optical fiber, radio, infrared,

OSI Layers: Data Link

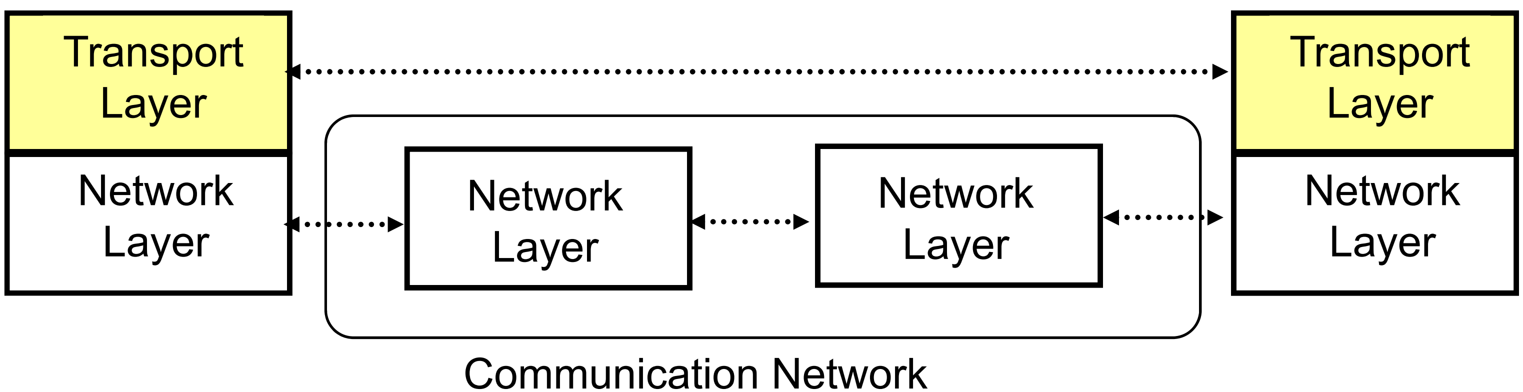
- Groups bits into frames
- Transfers *frames* across *direct* connections
- Detection of bit errors; Retransmission of frames
- Activation, maintenance, & deactivation of data link connections
- Medium access control for local area networks
- Flow control
- Buffer management

OSI Layers: Network

- Transfers *packets* across multiple links and/or multiple networks
- *Addressing* must scale to large networks
- Nodes *jointly* execute *routing* algorithm to determine paths across the network
- *Forwarding* transfers packet across a node
- *Congestion control* to deal with traffic surges
- Connection setup, maintenance, and teardown when connection-based (Optional)

OSI Layers: Transport

- Transfers data end-to-end from process in a machine to process in another machine
- Reliable stream transfer or quick-and-simple single-block transfer
- Port numbers enable multiplexing
- Message segmentation and reassembly
- Connection setup, maintenance, and release or simply connectionless

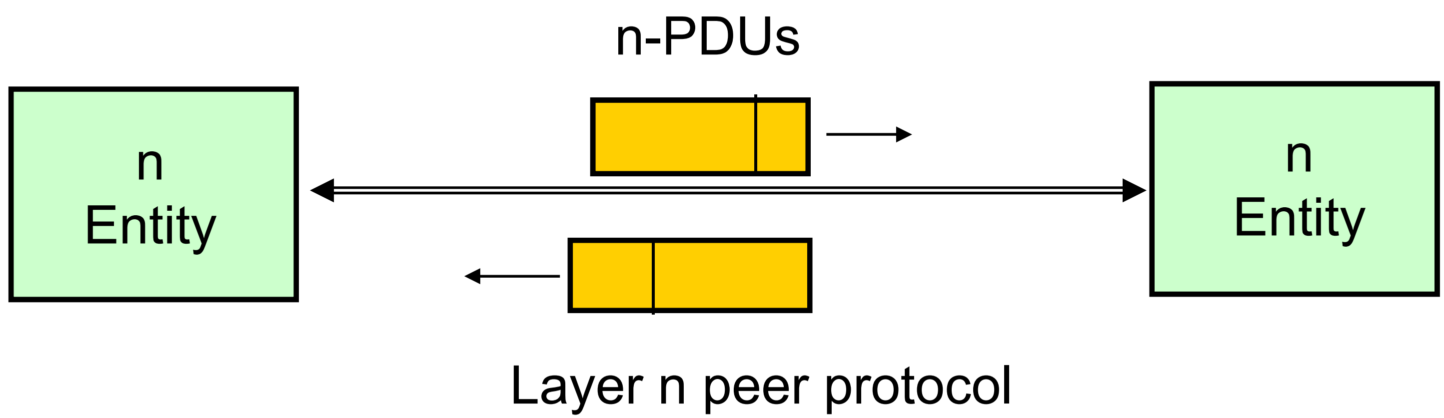


OSI Layers: Session, Presentation, Application

- Presentation Layer: machine-independent representation of data…
- Session Layer: dialog management, recovery from errors, …
- Application Layer: Provides services that are frequently required by applications: DNS, web acess, file transfer, email…

OSI Unified View on Protocols

- Layer n in one machine interacts with layer n in another machine to provide a service to layer n +1
- The entities comprising the corresponding layers on different machines are called *peer processes.*
- The machines use a set of rules and conventions called the *layer-n protocol*.
- Layer-n peer processes communicate by exchanging *Protocol Data Units* (PDUs)



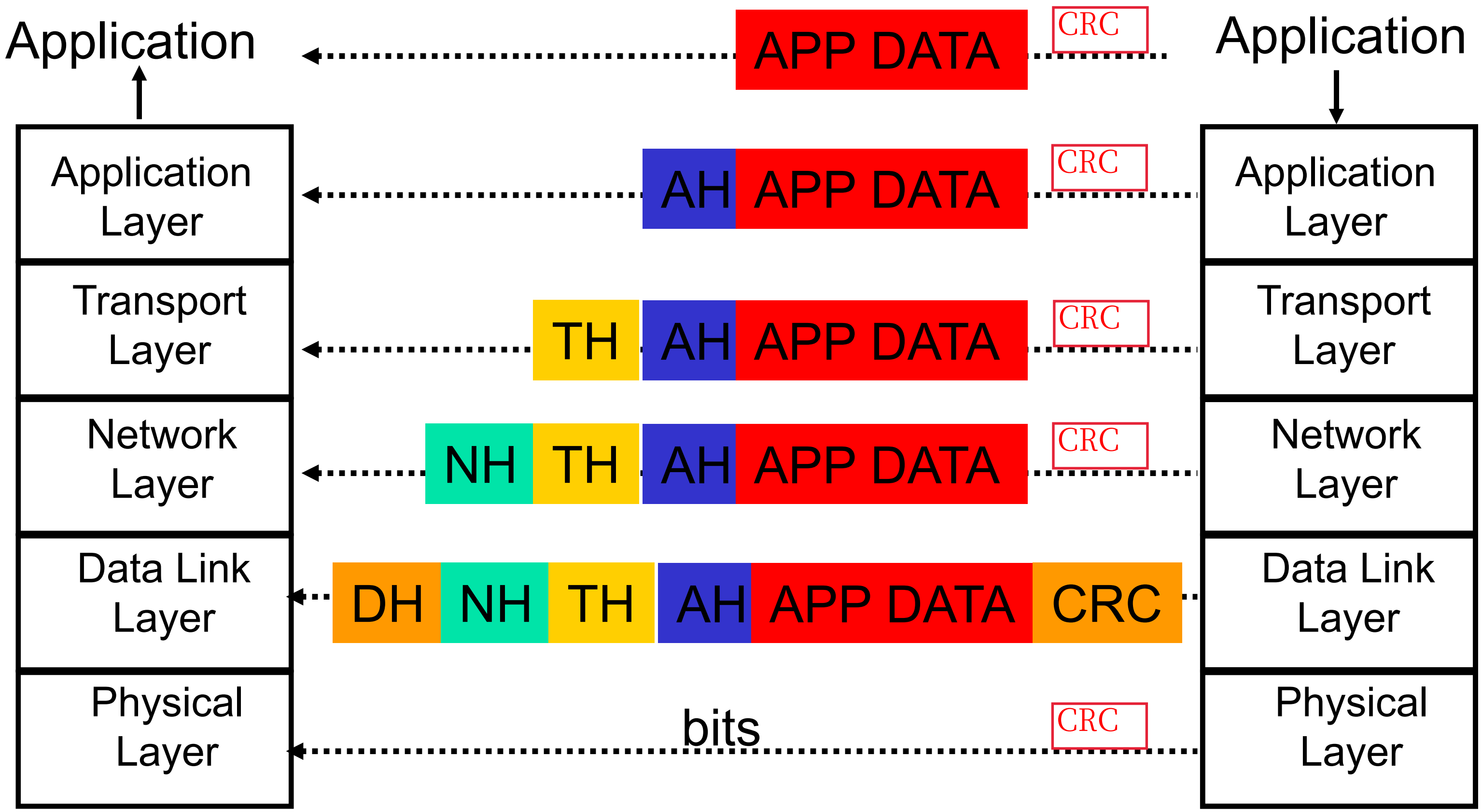
OSI Unified View on Services

- Communication between peer processes is virtual and actually indirect
- Layer n+1 transfers information by invoking the services provided by layer n
- Services are available at *Service Access Points* (SAP's)
- Each layer passes data & control information to the layer below it until the physical layer is reached and transfer occurs
- The data passed to the layer below is called a *Service Data Unit* (SDU)
- SDU's are *encapsulated* in PDU's

Headers & Trailers

- Each protocol uses a header that carries addresses, sequence numbers, flag bits, length indicators, etc…
- Error detection bits , e.g., CRC check bits, may be appended for error detection

每一个layer都可以append CRC

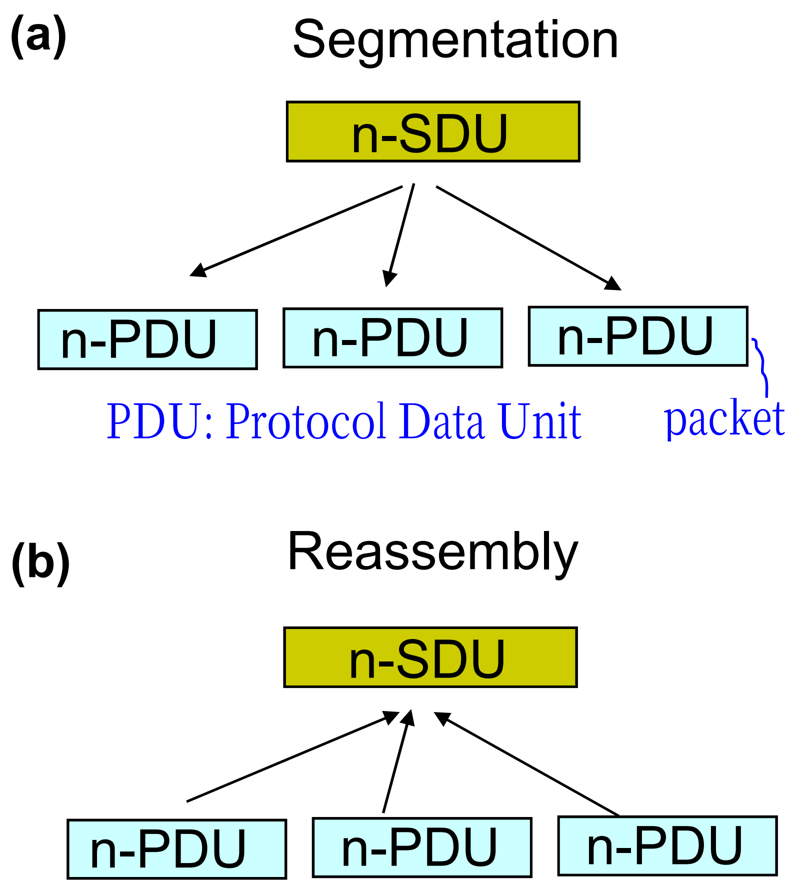


Segmentation & Reassembly

happens when data is too large

- A layer may impose a limit on the size of a data block that it can transfer for implementation or other reasons
- Thus a layer-n SDU may be too large to be handled as a single unit by layer-(n-1)
- Sender side: SDU is segmented into multiple PDUs
- Receiver side: SDU is reassembled from sequence of PDUs

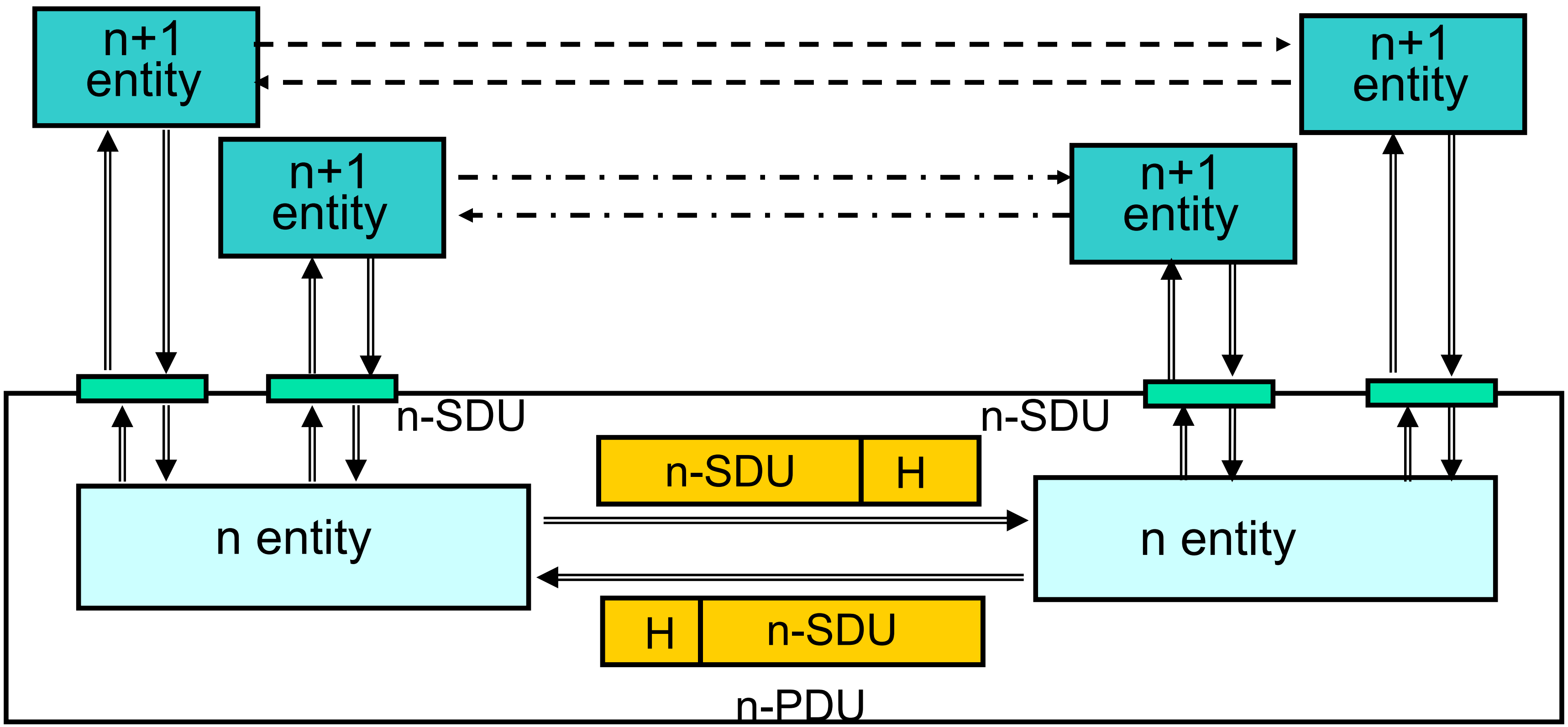
SDU: Service Data Unit (raw data)



Segmentation happens in every layer but practically, we want to avoid this because this will cause a lot of overhead.

Multiplexing

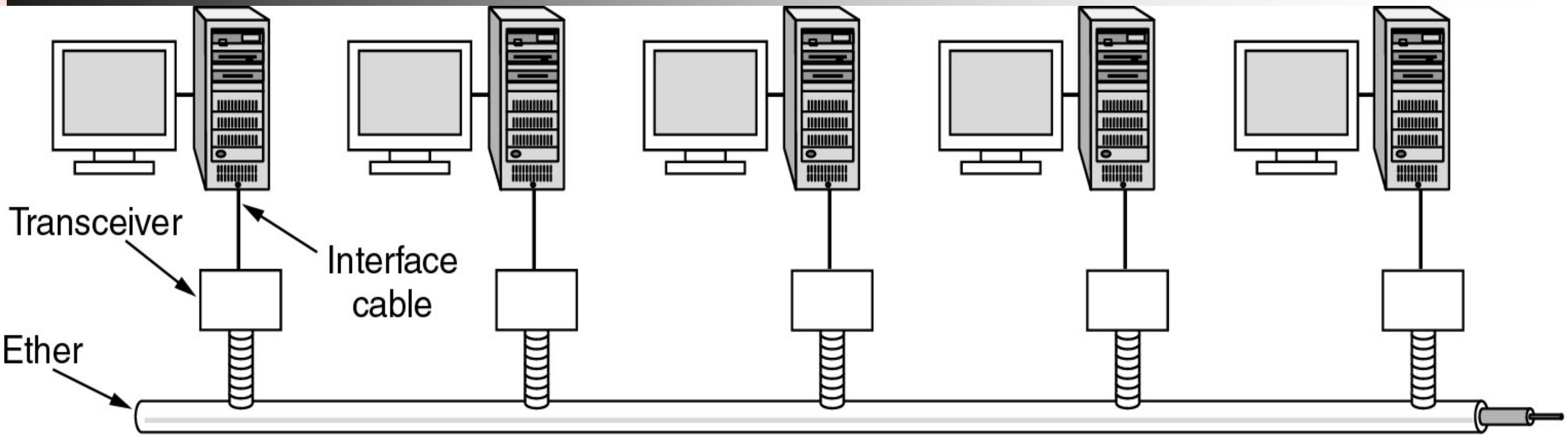
- Sharing of layer n service by *multiple* layer n+1 users
- Multiplexing tag or ID required in each PDU to determine which users an SDU belongs to



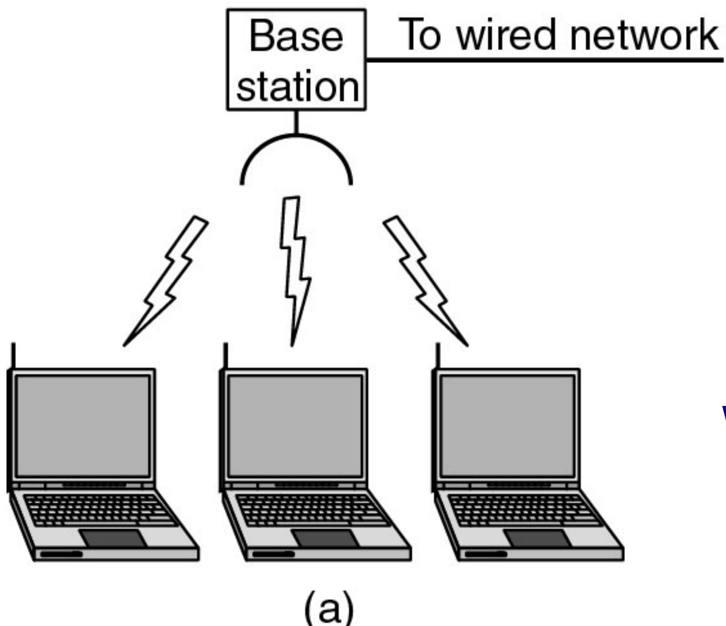
A Critique of the OSI Model and Protocols

- Why OSI did not take over the world
  - Bad timing
  - Bad politics
  - Bad technology ?
  - Bad implementations ?

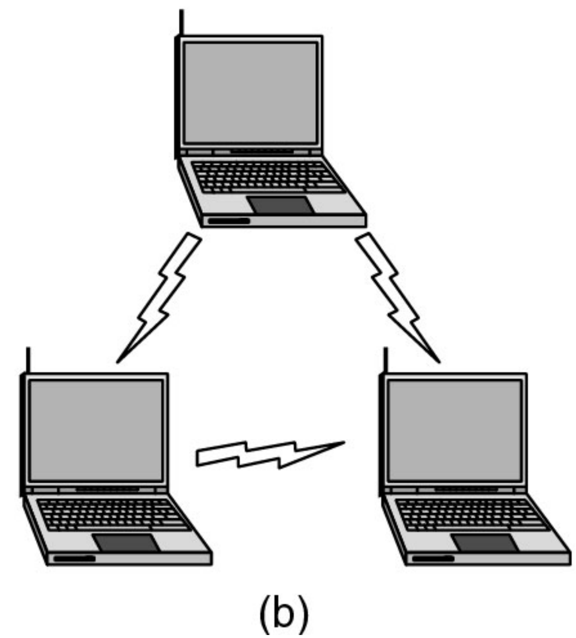
Local Area Networks



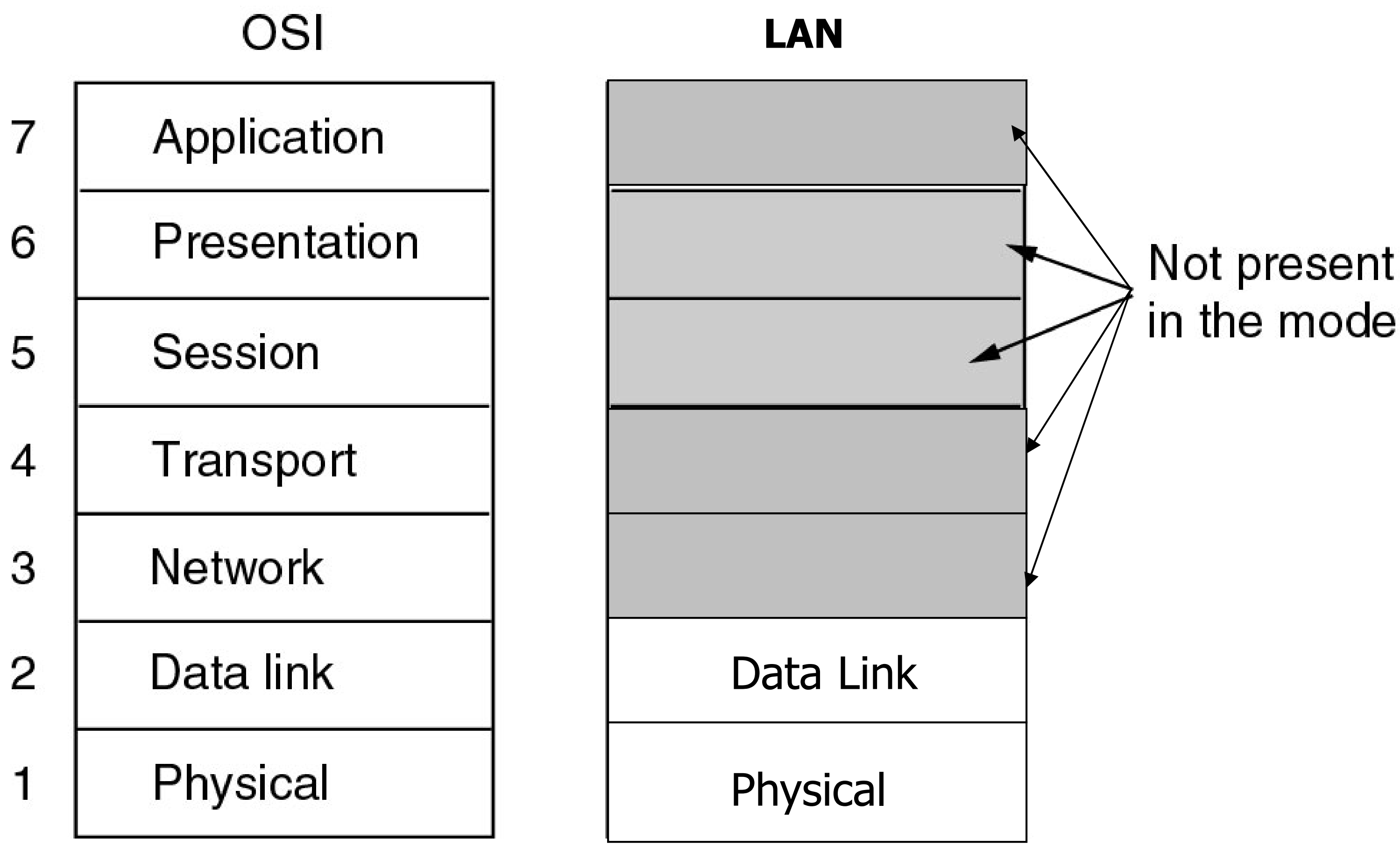
Wired LANs, e.g., Ethernet



Wireless LANs

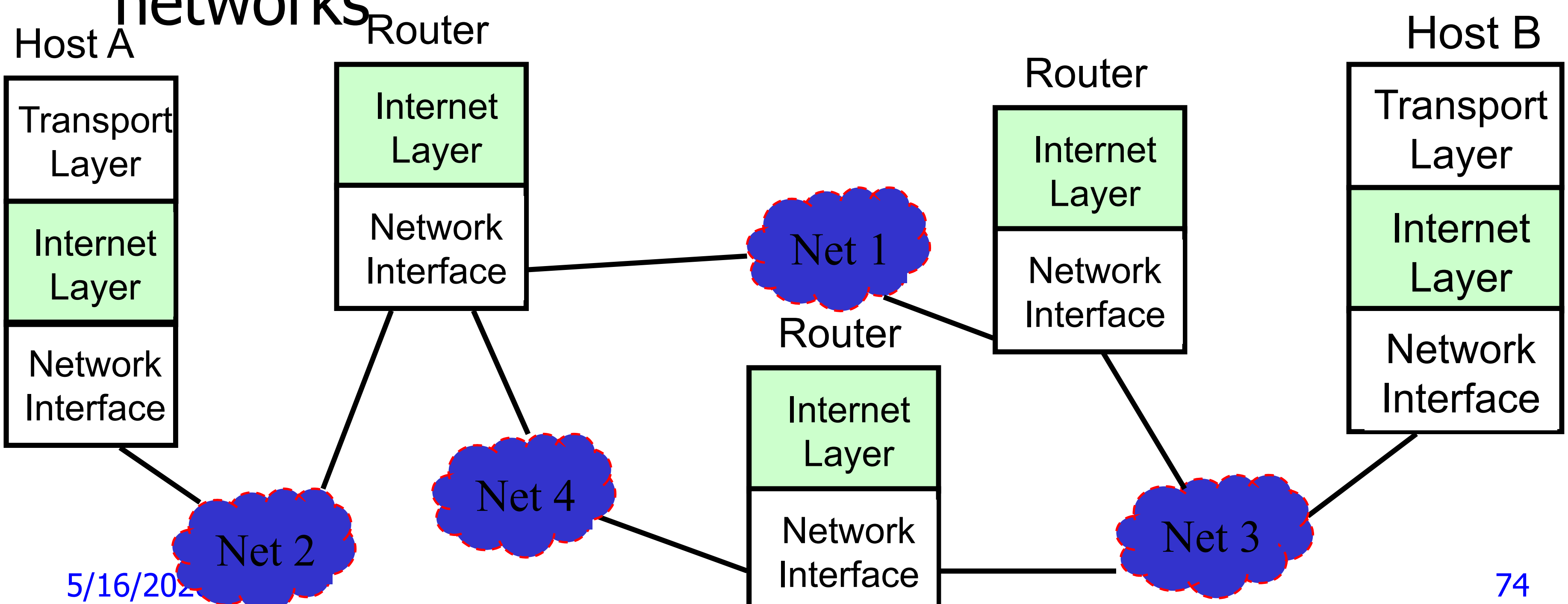


LAN Protocol Reference Model

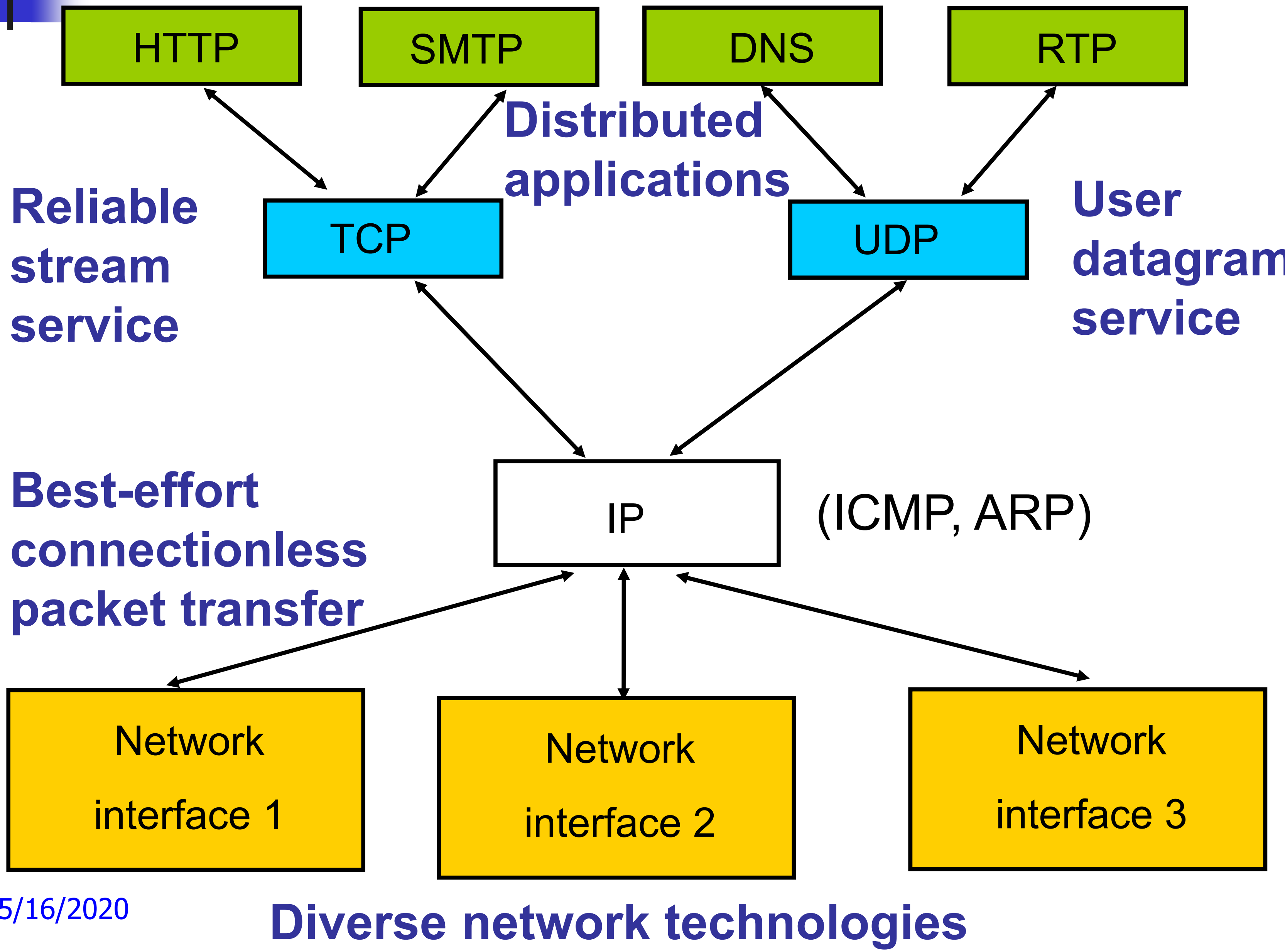


Internet Protocol Approach

- IP packets transfer information across Internet
  - *Host A IP → router→ router…→ router→ Host B IP*
- IP layer in each router determines next hop (router)
- Network interfaces transfer IP packets across networks



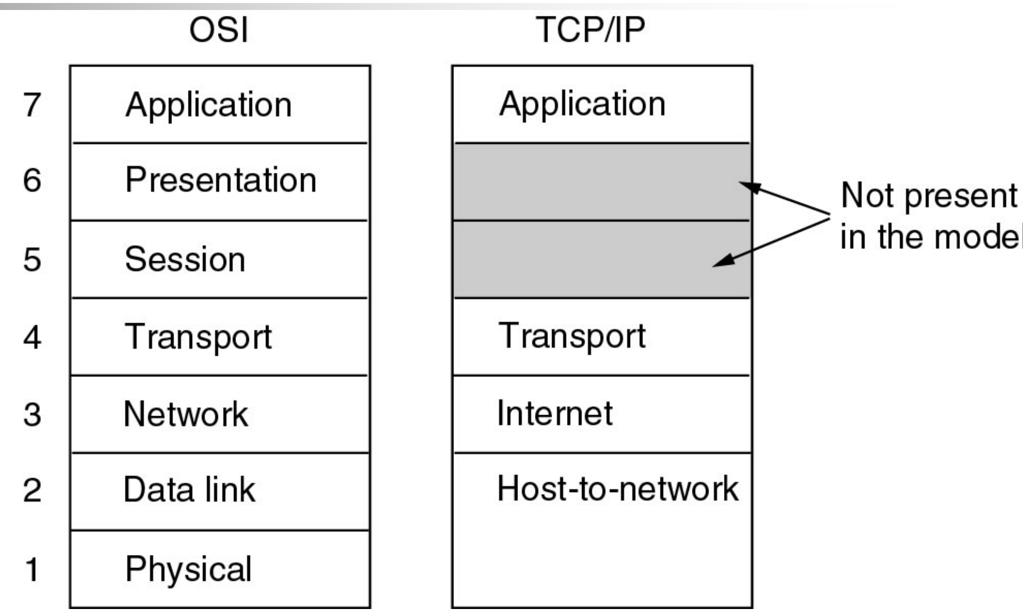
Internet Protocol Suite

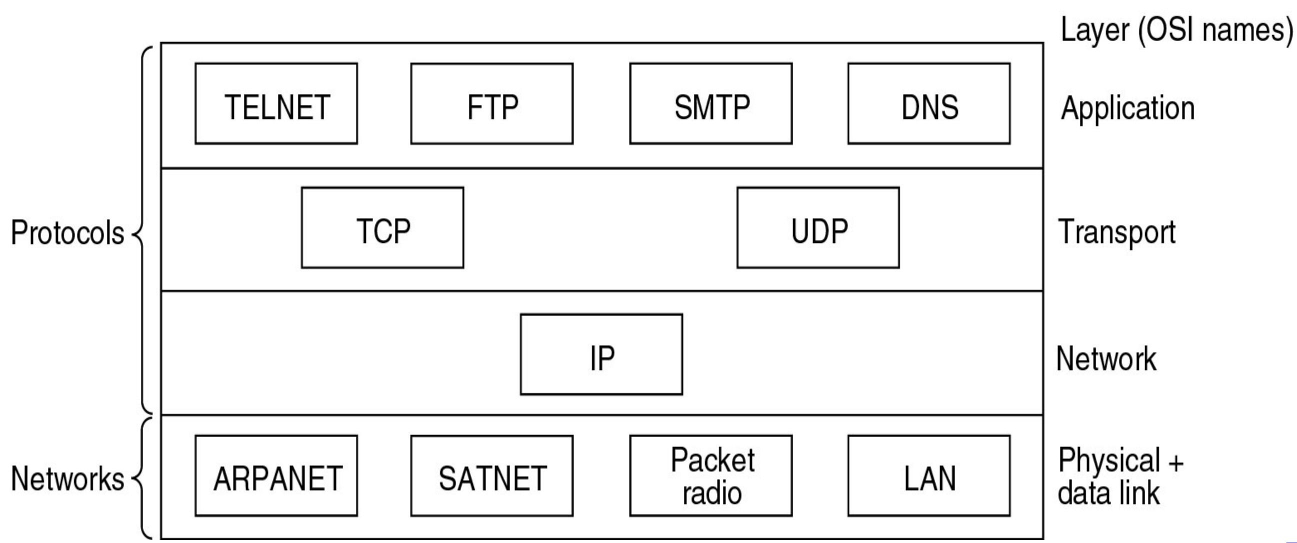


Internet Reference Model

Problems:

- **Not a general model**
- **Host-to-network "layer" not really a layer**
- **No mention of physical and data link layers**
- **Minor protocols deeply entrenched, hard to replace**



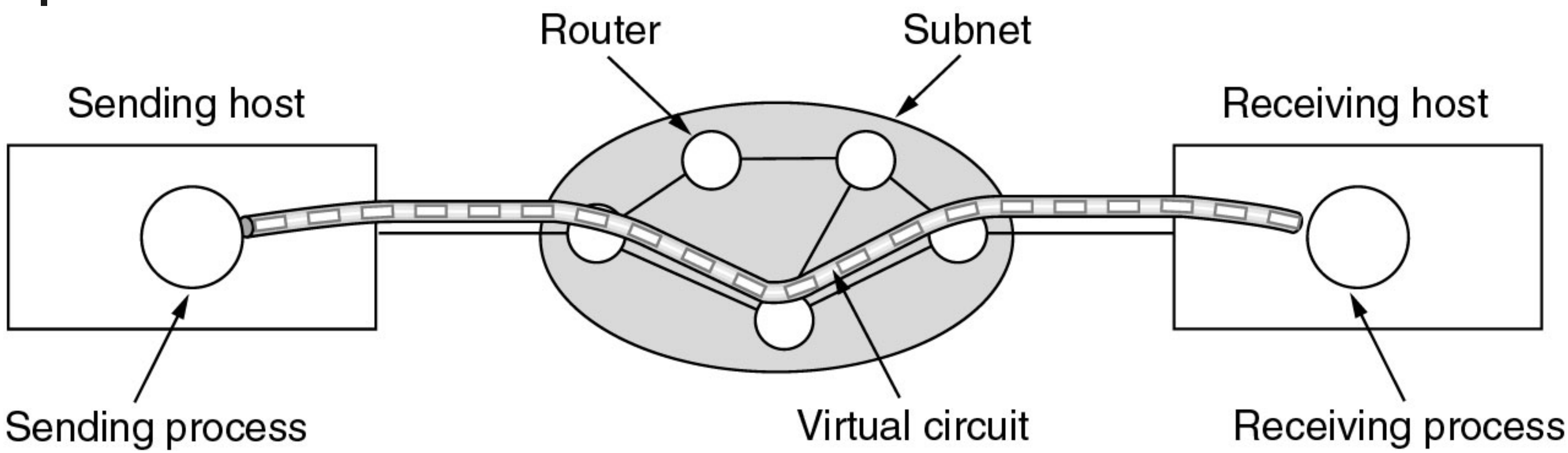


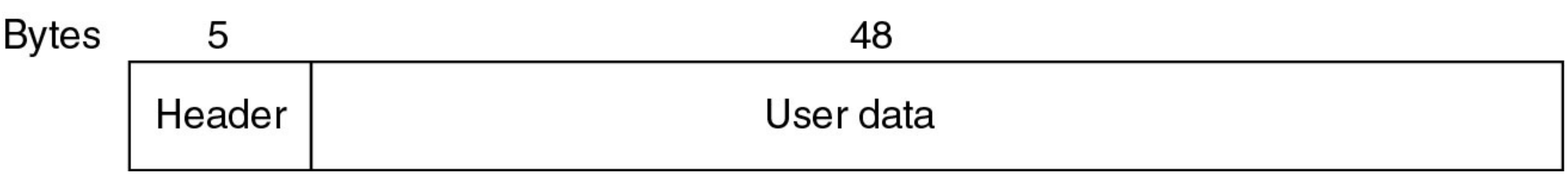
Hybrid Protocol Model

- A widely accepted model
- Reference for integrating all different networks

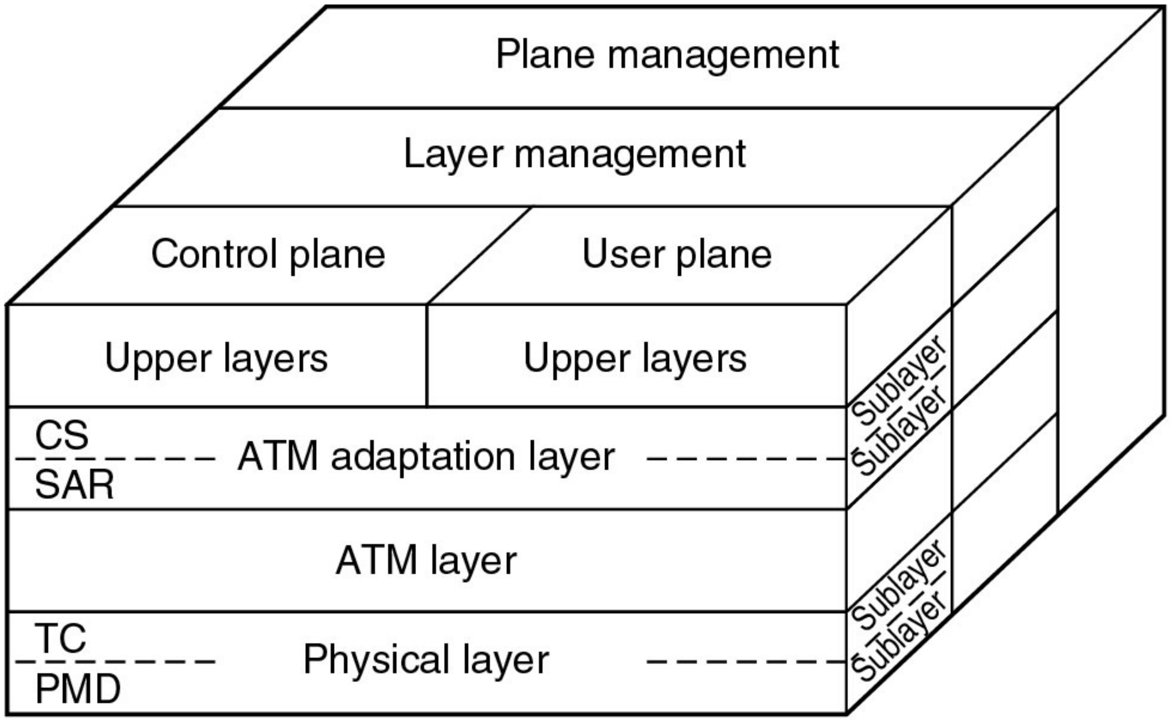
5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

ATM Virtual Circuit





ATM Protocol Model



CS: Convergence sublayer
SAR: Segmentation and reassembly sublayer
TC: Transmission convergence sublayer
PMD: Physical medium dependent sublayer

ATM versus OSI

OSI layer	ATM layer	ATM sublayer	Functionality
3/4	AAL	CS	Providing the standard interface (convergence)
		SAR	Segmentation and reassembly
2/3	ATM		Flow control Cell header generation/extraction Virtual circuit/path management Cell multiplexing/demultiplexing
2	Physical	TC	Cell rate decoupling Header checksum generation and verification Cell generation Packing/unpacking cells from the enclosing envelope Frame generation
1		PMD	Bit timing Physical network access

Discussions on Layering

Standards

- Standards Committees or Groups
  - ISO
  - ITU
  - IEC
  - IEEE
  - IETF
  - ANSI
  - 3GPP, 3GPP2
  - ATM Forum
  - Many more …

IEEE 802 Standards

Number	Topic
802.1	Overview and architecture of LANs
802.2 ↓	Logical link control
802.3 *	Ethernet
802.4 ↓	Token bus (was briefly used in manufacturing plants)
802.5	Token ring (IBM's entry into the LAN world)
802.6 ↓	Dual queue dual bus (early metropolitan area network)
802.7 ↓	Technical advisory group on broadband technologies
802.8 †	Technical advisory group on fiber optic technologies
802.9 ↓	Isochronous LANs (for real-time applications)
802.10 ↓	Virtual LANs and security
802.11 *	Wireless LANs
802.12 ↓	Demand priority (Hewlett-Packard's AnyLAN)
802.13	Unlucky number. Nobody wanted it
802.14 ↓	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth)
802.16 *	Broadband wireless
802.17	Resilient packet ring

↓ goes on

Frequently Used Terms

- Network Topology
  - Point-to-point, point-to-multipoint (PMP), multipoint-to-multipoint (mesh), ad hoc (ad-hoc)
- Transmission Types
  - Unicast, multicast, broadcast
- Connectionless versus Connection Oriented
- Best Effort versus Guaranteed
- Message, Circuit, and Packet Switching
  - Virtual circuit (ATM)

Related Technologies

- Transmission Media
  - Wired, optical, wireless
- Digital Communications
  - Propagation (wired or wireless), communication theory, coding theory (channel coding and source coding, information theory
- Communication Protocols
  - Data link (MAC, ARQ, etc.)
  - Routing
  - Transport
  - Application
- Circuit Design
- Computer Systems