CMPE 344 Computer Networks
Spring 2022

Foundations

Reading: Peterson and Davie, §1.1-1.5

Sources of slides:

Peterson and Davie, Computer Networks: A Systems Approach, 6th ed., Morgan Kaufmann, 2021.
Tanenbaum, Feamster, and Wetherall, Computer Networks, 6th ed., Pearson, 2021.
Kurose and Ross, Computer Networking, 8th ed., Pearson, 2021.

Goals of CMPE 344

- Emphasize on design choices made when building computer networks
- Understand why networks are designed the way they are
- Discuss "systems" approach to understand the big picture
- Gain insight on architectural implications and interaction of components rather than focusing on rigidly defined layers

Network Applications

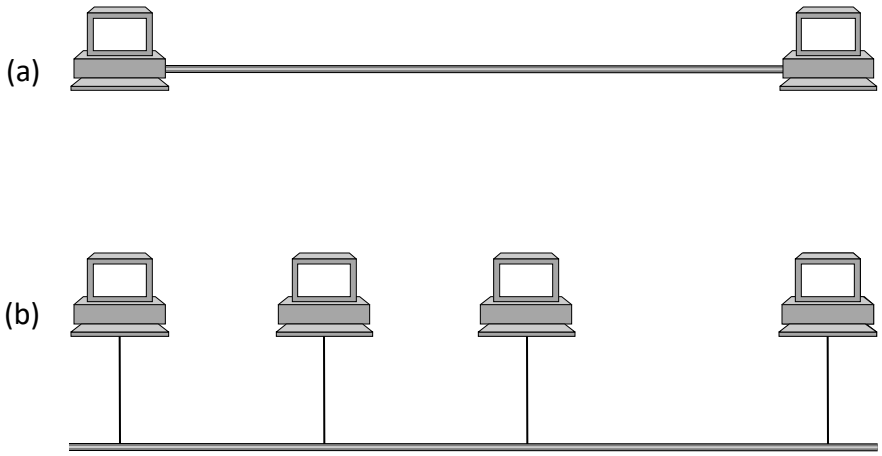
- Most people are familiar with network through applications:
  - World Wide Web
  - Email
  - Online social network
  - Streaming audio/video
  - Videoconferencing (e.g. Skype™)
  - File sharing
  - Instant messaging
  - … and many more

Network Applications

Links and nodes

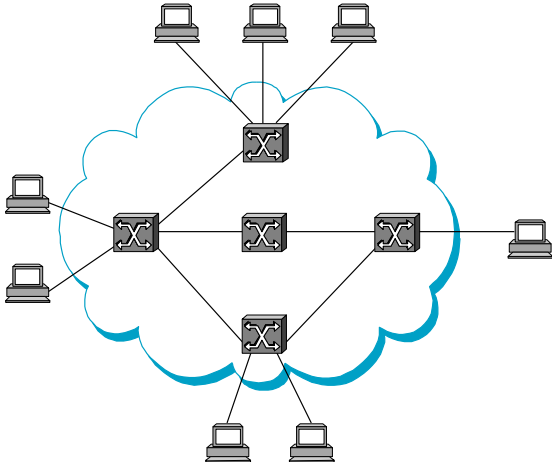
- A link is a physical medium that connects computers or nodes.
  - A node is a piece of hardware such as a PC, a switch, a router, etc.…
- Key challenge in network design: Scalability
  - A system that is designed to support growth to an arbitrarily large size is said to scale
  - How should we connect all the nodes in the world?

Connectivity: Direct links



(a) Point-to-point
(b) Multiple access (Sharing a single physical link)

Connectivity: Switched network



- Connectivity between two nodes does not necessarily imply a direct physical connection between them
- Nodes can be indirectly connected: A switched network contains forwarding nodes or switches

Types of switched networks

- Circuit-switched networks
  - Telephone networks
  - A dedicated circuit across a sequence of links is established. Source node sends a stream of bits across this circuit to a destination node
- Packet-switched networks
  - Computer networks
  - Discrete blocks of data are sent in a (usually) store-and-forward manner

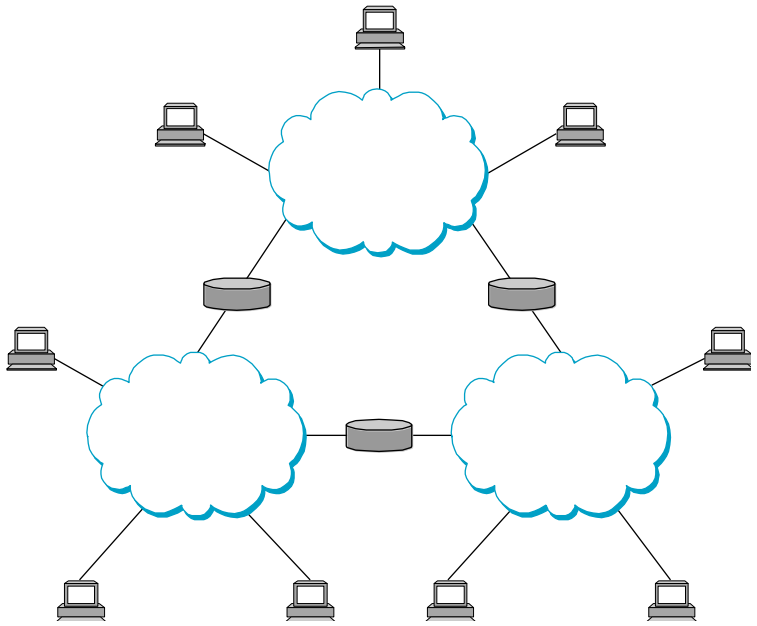
Packets

- Nodes in packet switched networks send discrete blocks of data to each other
  - These blocks correspond to some piece of application data such as a file, a piece of email or an image
  - The blocks are called packets or messages (more on this later)

Store-and-forward

- In packet switched networks, each node
  - first receives complete packet over some link
  - stores the packet in its internal memory
  - and then forwards the complete packet to the next node
- Primary function of packet switches is to store and forward packets

Interconnection of networks



An internetwork or an internet

- A node that connects two or more networks is called a router or gateway
- The Internet is an example of an internet

Addressing and routing

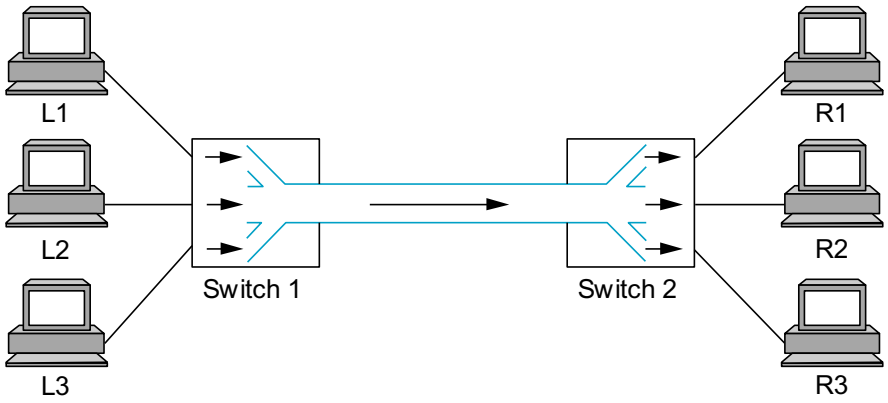
- Each node in a network must have an identifier called its address (a byte string)
- When a source node wants the network to deliver a message to a certain destination, it must specify the address of the destination node
- If the sending and receiving nodes are not directly connected, switches and routers use the address to decide how to forward the message toward the destination
- The process of determining systematically how to forward messages toward the destination node based on its address is called routing

Unicasting, broadcasting, and multicasting

- Unicast: Source sends a message to a single destination node
- Broadcast: Source sends a message to all the nodes on the network
- Multicast: Source sends a message to some subset of the other nodes (but not all of them)
- Thus, in addition to node-specific addresses, a network may have to support broadcast and multicast addresses as well
- Other “cast”s:
  - Anycast: One-to-one-of-many (Applications?)
  - Convergecast: Many-to-one (Applications?)

Cost-effective resource sharing

- How do hosts share the network and the same link when they all want to use it at the same time?
- Multiplexing: Sharing a system resource among multiple users



Multiplexing/demultiplexing

TDM, FDM, CDMA

- Synchronous time-division multiplexing (STDM) or just time-division multiplexing (TDM)
  - Time is divided into equal-sized quanta and each user is given a chance to send data in a round-robin manner
- Frequency-division multiplexing (FDM)
  - Each source transmits at a different frequency
- Code-division multiple access (CDMA)
  - Each source has its own “code”
  - Suitable for bursty data (we’ll define “bursty” later)
  - In fact, it is a “multiple access” method (see more later)

Q: Find out what kind of multiplexing is used in cellular networks and radio/TV broadcasting.

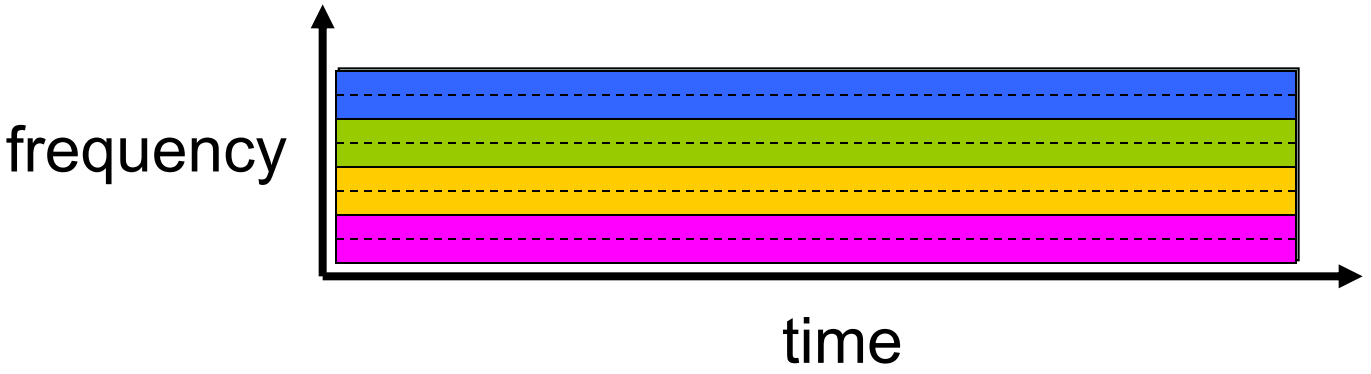
Q: What kind of multiplexing is used in fiber optic communications? Hint: Optical carriers can be described by their wavelengths.

TDM vs. FDM

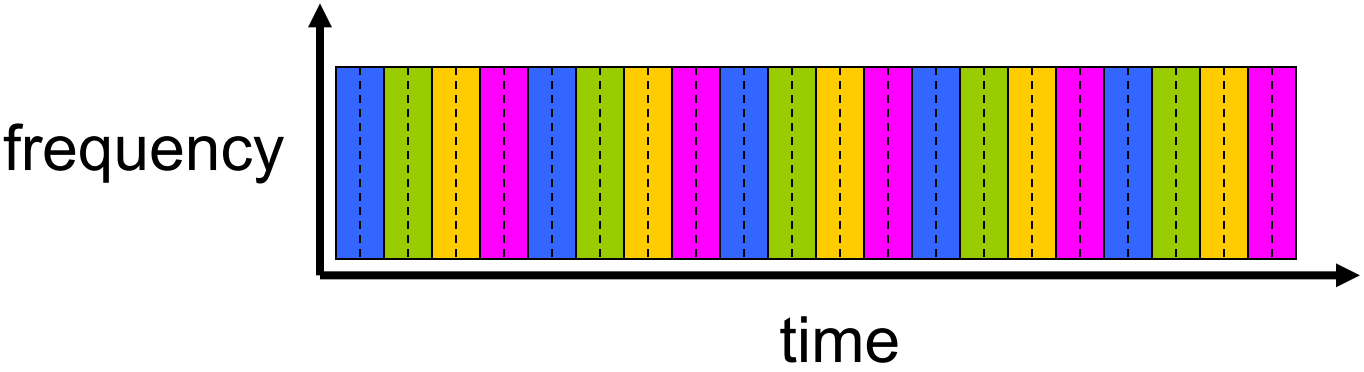
Example:

4 users

FDM



TDM



Limitations of TDM and FDM

- If one of the pairs does not have any data to send, its share of physical link (time quantum or frequency) remains idle even if one of the other pairs has data to transmit
- In computer communications the idle time can be large! (Q: Why?)
  - We say that computer communications traffic is bursty.
- Also, the maximum number of flows in these schemes is fixed ahead of time

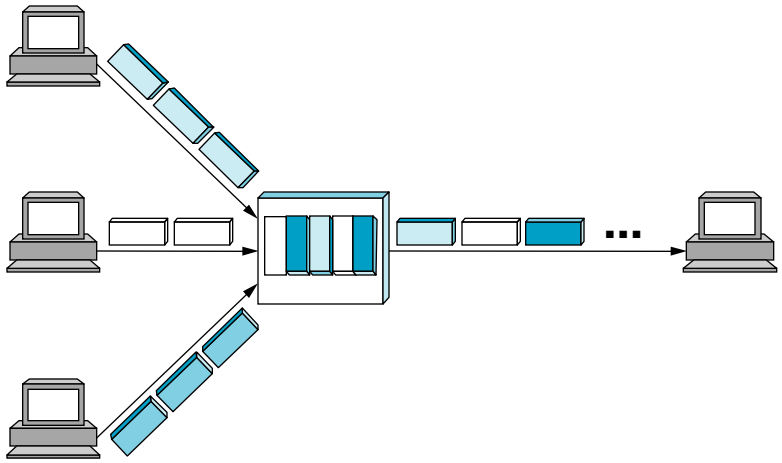
Statistical Multiplexing

- *Not all of the pairs are active at the same time!*
- Like STDM: Link is shared over time
- Unlike STDM: Data is transmitted from each user on demand rather than in a predetermined slot
- Statistical multiplexing is efficient!

Q: How does one ensure that transmission decisions are made fairly?

More on statistical multiplexing

- A packet switched network fragments a large message into packets (that have a maximum size) so that other flows can have a turn to transmit their own
- Receiver will have to reassemble the packets back into original message



Multiplexing packets onto a shared link

More on statistical multiplexing

- The switch may have to buffer packets in its memory if it receives packets faster than the shared link can handle
- If switch receives packets faster than it can send them for an extended period of time, resource will be congested
  - Packets will be dropped
  - Packets will be delayed excessively

Statistical multiplexing vs. circuit switching

- Suppose users share a 1 Mbps link and each user alternates between periods of activity (generates data at constant rate 100 kbps) and periods of inactivity
- Further, each user is active 10% of time
- With circuit switching (TDM, FDM), 100 kbps must be reserved for each user at all times
  - E.g., TDM: 1-sec. frame is divided into 10 time slots of 100 ms each
  - The link can support only 10 simultaneous users

Statistical multiplexing vs. circuit switching

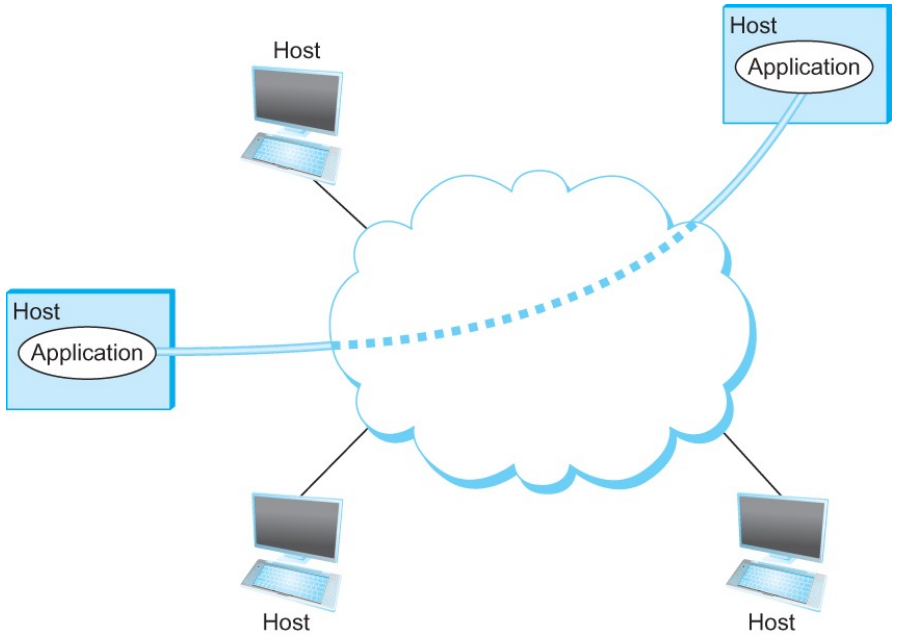
- With statistical multiplexing, the probability that each user is active is 0.1 (10%)
- If there are 35 users, the probability that there are 11 or more simultaneously active users is approx. 0.0004 (see next slide)
- When there are 10 or fewer simultaneously active users (which happens with probability 0.9996), the aggregate arrival rate of data is less than or equal to 1 Mbps (the output rate of the link)
- When there are more than 10 active users, queue will begin to grow (until aggregate rate falls below 1 Mbps)
- The probability of having more than 10 simultaneously active users is miniscule, the performance is the same as circuit switching, and 3 times the number of users are allowed

Statistical multiplexing vs. circuit switching

- Let p=0.1
- Probability of having n simultaneously active users out of 35 at any given time is
  - $\binom{35}{n} p^n(1-p)^{35-n}$
- Probability that there are 11 or more users transmitting simultaneously is
  - $1 - \sum_{n=0}^{10} \binom{35}{n} p^n(1-p)^{35-n} = 0.0004$

Support for common services

- Goal: Hide the complexity of the network from the application
- Logical channels: Application-to-application communication path or a pipe



- Processes communicating over an abstract channel
- Channel must meet the functionality requirements of applications

Common communication patterns

- After understanding the communication needs of a representative collection of applications, extract their common communication requirements, and finally incorporate the functionality that meets these requirements in the network
- Examples: Request/reply channels and message stream channels
  - Can you give example applications that make use of these channels?

Characterization of networks according to their sizes

PAN	Personal area network	Around an individual
SAN	Storage area network	In a room
LAN	Local area network	~ 1 km
MAN	Metropolitan area network	~ 10 km
WAN	Wide area network	Worldwide

Three types of failures

- Bit errors
  - 1 is turned into a 0 or vice versa
  - Causes: Lightning, power surges, microwave ovens
  - These are rare: 1 out of 10^6-10^7 on copper cable, 1 out of 10^{12}-10^{14} on fiber
  - Single bit errors and burst errors
- Packet losses
  - Uncorrectable bit errors or packet drops at intermediate nodes due to lack of buffer space
- Node- or link-level failures
  - Crashed computers, broken links, misconfiguration
  - Q: How do you distinguishing between a failed computer and one that is merely slow? What about the difference between between one that has been cut and one that is very flaky and therefore introduces a high number of bit errors? (See more in CMPE 449 Distributed Systems)

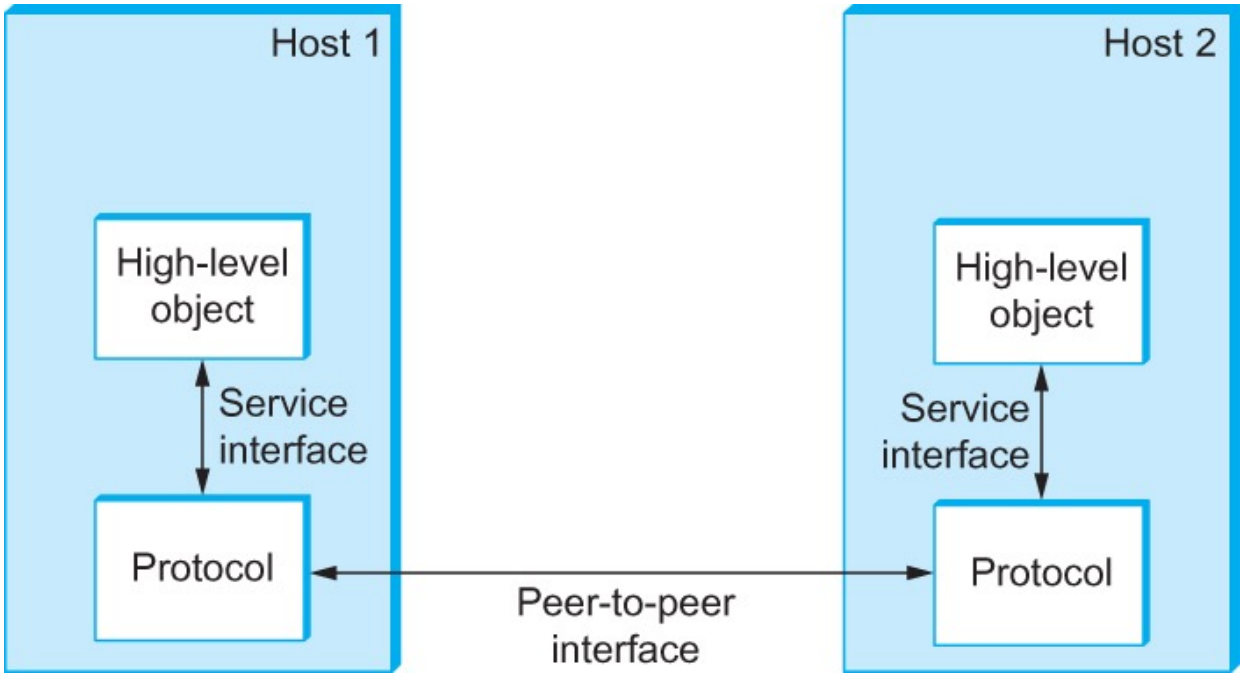
Layering

- A fundamental design and implementation concept
- When designing and analyzing complex systems, we usually abstract away the details of components and provide an interface for other components of the system
- Services provided at higher layers are implemented in terms of services provided by lower layers

Protocols

- Protocols
  - are abstract objects that make up the layers of a network system
  - provide communication service that higher level objects (e.g. application processes or higher level protocols) use to exchange messages
  - In fact, the term “protocol” is “overloaded”: Specification of interface or module that implements it
- Two different interfaces are generally provided
  - Service interface: To other objects on the same computer
  - Peer interface: To a protocol’s counterpart (peer) on another machine

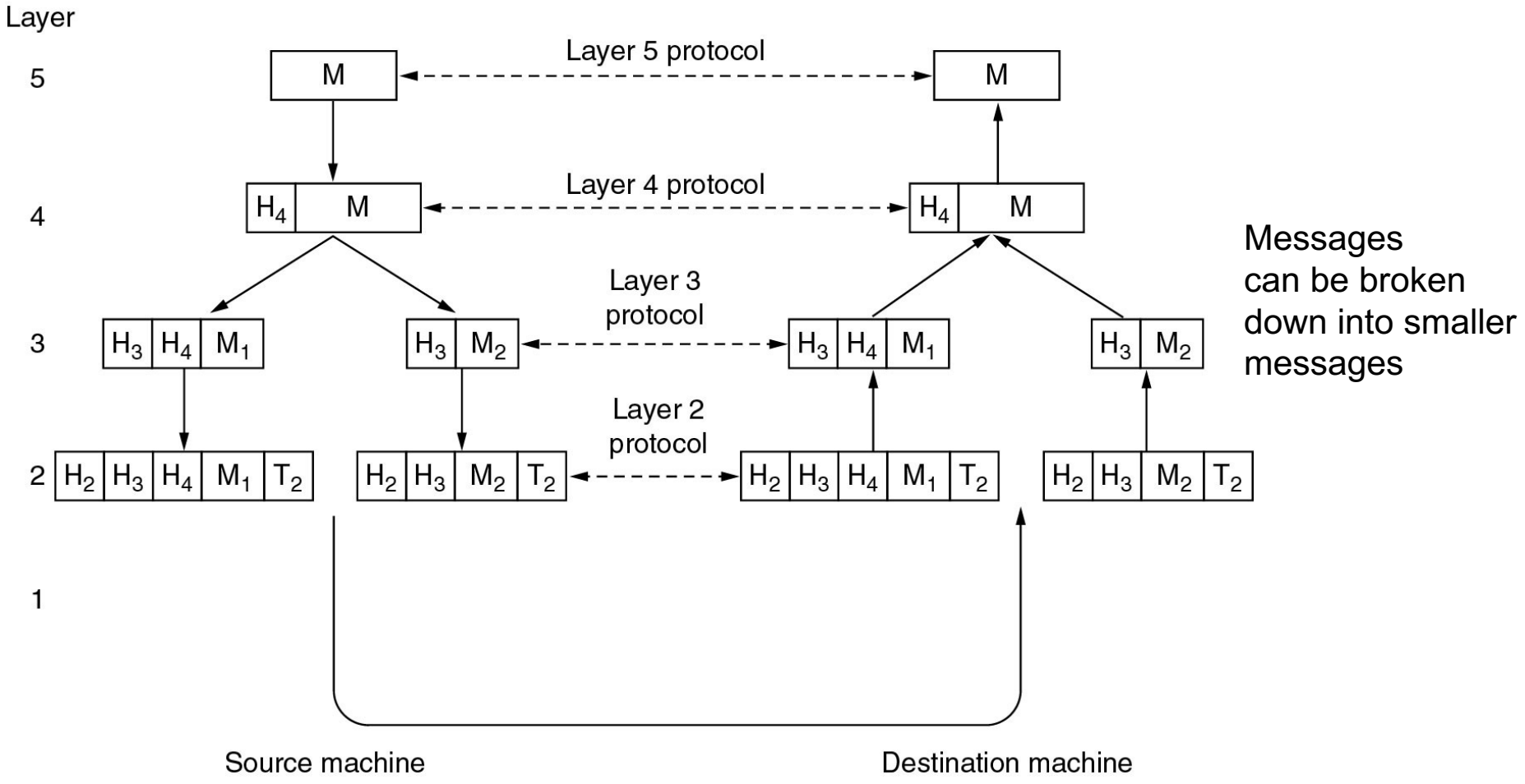
Service and peer interfaces of protocols



Encapsulation

- A header (and/or trailer) is attached to a message body or payload
- Referred to as multiplexing and demultiplexing up and down the protocol graph
- Headers attached to messages contain an identifier that records the higher level object to which the message belongs
- Trailers may contain information for error detection/correction

Encapsulation (cont.)

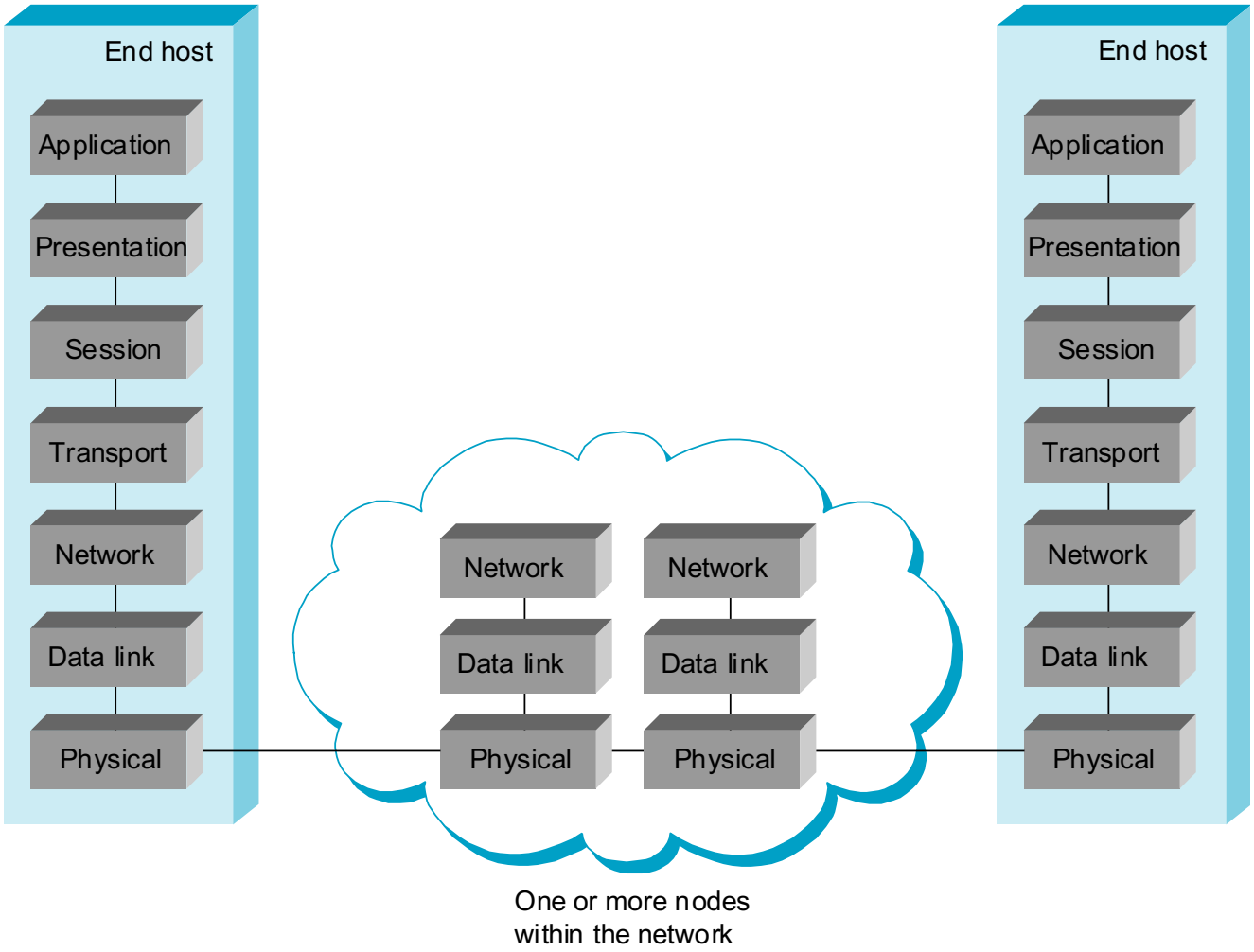


Encapsulation/decapsulation in an example 5-layer architecture

Network architecture

- The set of rules governing the form and context of a protocol graph is called a network architecture
  - The suite of protocols that make up a network system is represented by a protocol graph
- International Standards Organization (ISO) and Internet Engineering Task Force (IETF) are examples of standardization bodies for network architectures

Open Systems Interconnection (OSI) reference model



Description of layers

- Physical Layer
  - Handles the transmission of raw bits over a communication link
- Data Link Layer
  - Collects a stream of bits into a larger aggregate called a frame
  - Network adaptor along with device driver in OS implement the protocol in this layer
- Network Layer
  - Handles routing among nodes within a packet-switched network
  - Unit of data exchanged between nodes in this layer is called a packet

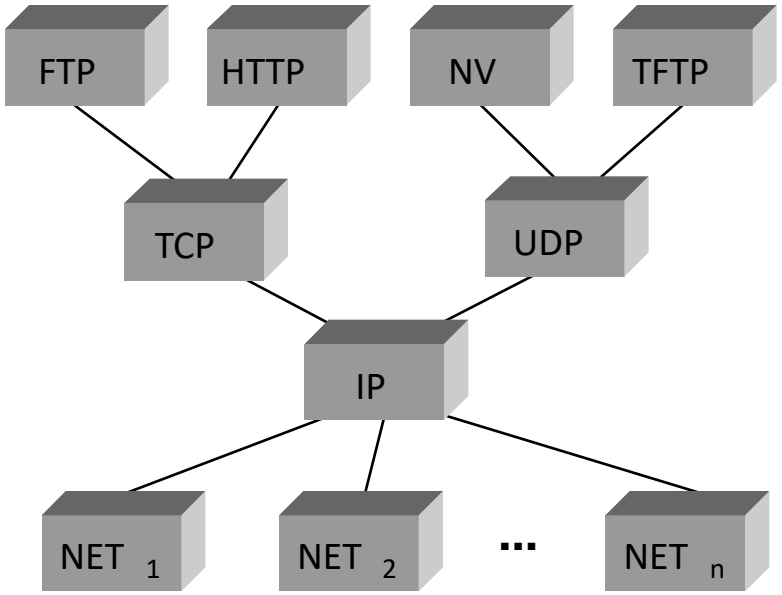
The lower three layers are implemented on all network nodes

Description of layers

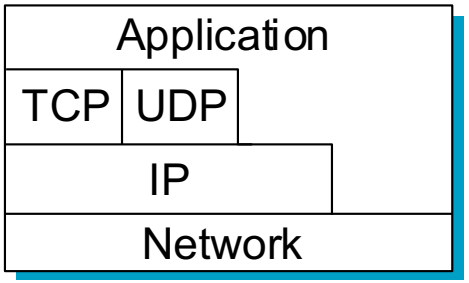
- Transport Layer
  - Implements a process-to-process channel
  - Unit of data exchanges in this layer is called a message
- Session Layer
  - Provides a name space that is used to tie together the potentially different transport streams that are part of a single application
  - Example: Management of an audio stream and a video stream that are being combined in a teleconferencing application
- Presentation Layer
  - Concerned about the format of data exchanged between peers
  - Is an integer is 16, 32, or 64 bits long? Is the most significant byte transmitted first or last? How is a video stream formatted?
- Application Layer
  - Standardize common type of exchanges

The transport layer and the higher layers typically run only on end-hosts and not on the intermediate switches and routers

Internet Architecture



Protocol graph: The hourglass shape



Reference model:
No strict layering

- Evolved out of ARPANET (DARPA/US DoD)
- Significance of the hourglass: Minimal and carefully chosen set of global capabilities that allow many higher layer applications and lower layer communication technologies to coexist, share capabilities, and evolve rapidly

IETF Internet architecture

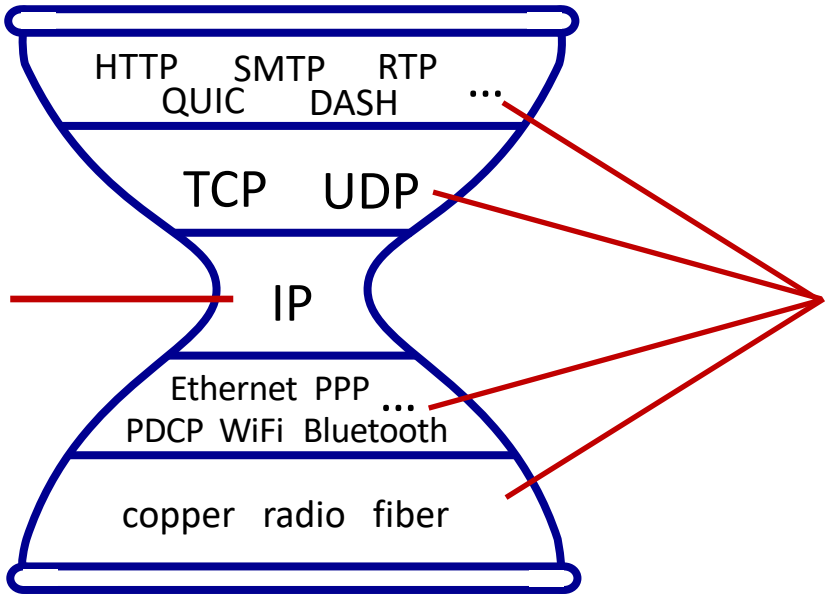
- Main features
  - Does not imply strict layering. The application is free to bypass the defined transport layers and to directly use IP or other underlying networks
  - An hour-glass shape – wide at the top, narrow in the middle and wide at the bottom. IP serves as the focal point for the architecture (Thin waist: Minimalist approach): Common method for exchanging packets among a wide collection of networks and arbitrarily many transport protocols, each offering a different channel abstraction to application programs
  - In order for a new protocol to be officially included in the architecture, there needs to be both a protocol specification and at least one (and preferably two) representative implementations of the specification

The IP hourglass

Young Internet!

Internet's "thin waist":

- *one* network layer protocol: IP
- *must* be implemented by every (billions) of Internet-connected devices



many protocols in physical, link, transport, and application layers

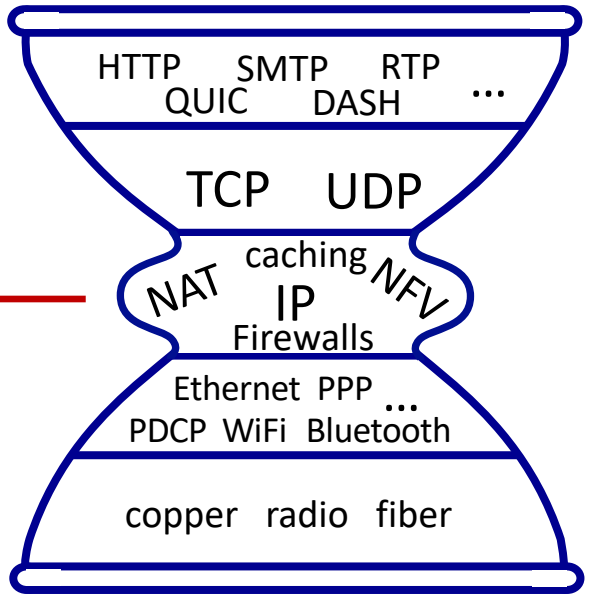
Slide from Kurose and Ross, 8th ed.

The IP hourglass, at middle age

The Internet is about 40 years old!

Internet's middle age "love handles"?

- middleboxes, operating inside the network



Slide from Kurose and Ross, 8th ed.

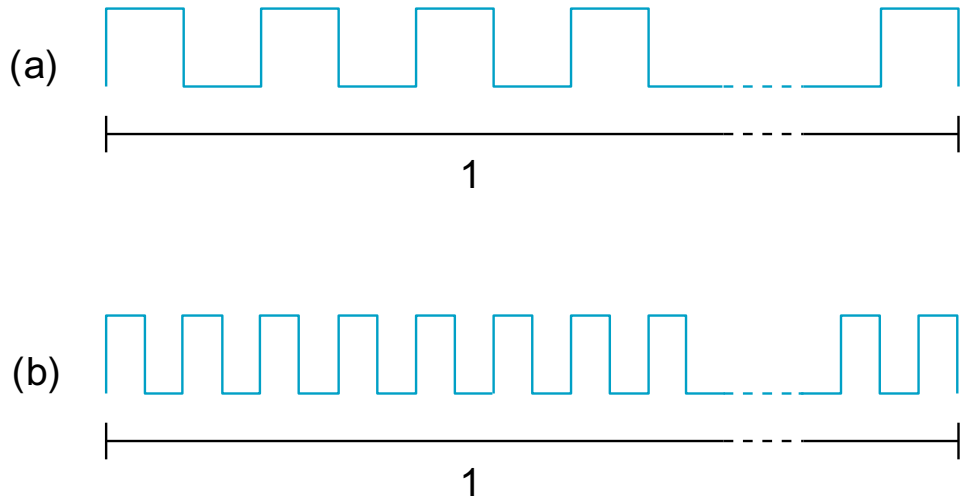
Implementing network software

- Network protocols are implemented as part of the operating system (OS)
- OS provides application programming interface (API) to network services
- Socket interface
  - Socket: Point where a local application process attaches to the network
  - Socket operations: Creating a socket, attaching the socket to the network, sending/receiving messages, closing the socket

Network performance

- Network performance is measured in two fundamental ways
  - Bandwidth (or sometimes throughput): Number of bits that can be transmitted over the network in a certain period of time (e.g., 10 Mbps: i.e. it takes 0.1 μs to transmit each bit)
    - Notation: KB = 2^{10} bytes; Mbps = 10^6 bits per second
  - Latency (or delay): Time it takes a message to travel from one end of a network to the other (e.g., 24 ms in transcontinental networks)
    - Round-trip time (RTT): Time it takes to send a message from one end of a network to the other and back

Bandwidth



(a) Bits transmitted at 1 Mbps
(b) Bits transmitted at 2 Mbps

We can talk about the bandwidth of a single physical link or the bandwidth of a logical process-to-process channel

Bandwidth vs. throughput

- In networking, it is customary to refer to bits-per-second (bps) as the bandwidth
- Depending on the context (communication theory, signal processing), bandwidth also refers to the range of signals that can be accommodated
  - e.g., Voice-grade telephone line, 300-3300 Hz, has a bandwidth of 3000 Hz
  - Relationship between bps and Hz will be established in Chapter 2, Shannon's Theorem
- Throughput is the bits-per-second (bytes-per-second, packets-per-second, etc.) that we can actually transmit in practice (refers to measured performance)

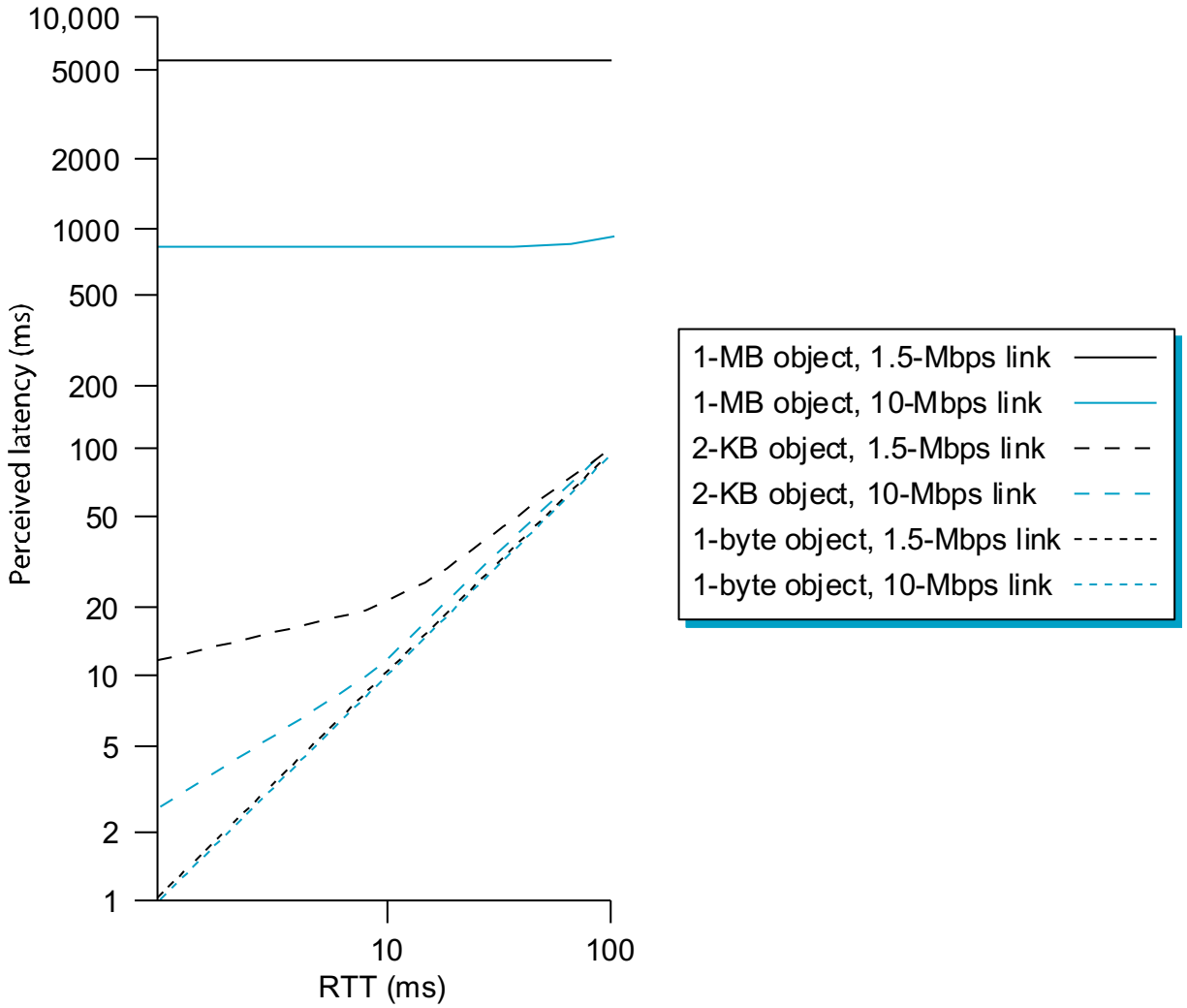
Delay and throughput

- In general, latency has 3 main components
  - Latency = Propagation + Transmit + Queue
  - Propagation = Distance / SpeedOfLight
  - Transmit = Size (bits) / Bandwidth (bps)
- Be careful since delay/latency/RTT are context dependent
  - We will make it explicit whenever necessary
- Effective end-to-end throughput
  - Throughput = TransferSize (bits) / TransferTime (sec)

Bandwidth vs. Latency

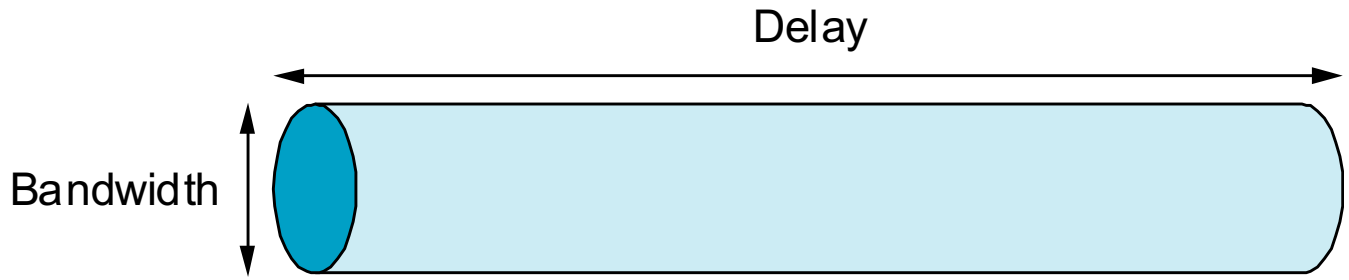
- Relative importance of bandwidth and latency depends on application
  - For large file transfer, bandwidth is critical
  - For small messages (HTTP, NFS, etc.), latency is critical
  - Variance in latency (jitter) can also affect some applications (e.g., audio/video conferencing) (see more later)

Latency vs. bandwidth domination



Delay x Bandwidth product

- Volume of the pipe: The maximum number of bits that could be in transit through the pipe at any given interval
- A transcontinental channel with a one-way latency of 50 ms and a bandwidth of 45 Mbps is able to hold 50×10^{-3} sec x 45×10^{6} bits/sec = 2.25×10^{6} bits = 280 KB of data



Sample delay x bandwidth products

Link type	Bandwidth	Distance	RTT	Delay x BW
Dial-up	56 Kbps	10 km	87 μs	5 bits
Wireless LAN	54 Mbps	50 m	0.33 μs	18 bits
Satellite	1 Gbps	35,000 km	230 ms	230 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

Here, RTT = 2 * Propagation&processing delay

Delay x bandwidth

- A network with a large delay-bandwidth product is commonly known as a long fat network (LFN)
- Performance problems might arise when this product is large
  - "Large": the product significantly exceeds 10^5 bits = 12,500 bytes
- Which links in the previous slide can be considered as an LFN?

Keeping the pipe full

- Delay x bandwidth is important to know when constructing high-performance networks because it corresponds to how many bits the sender must transmit before the first bit arrives at the receiver
- If the sender is expecting the receiver to somehow signal that bits are starting to arrive the sender can send up to 2 delay x bandwidth (= RTT x bandwidth) worth of data before hearing from the receiver

Keeping the pipe full

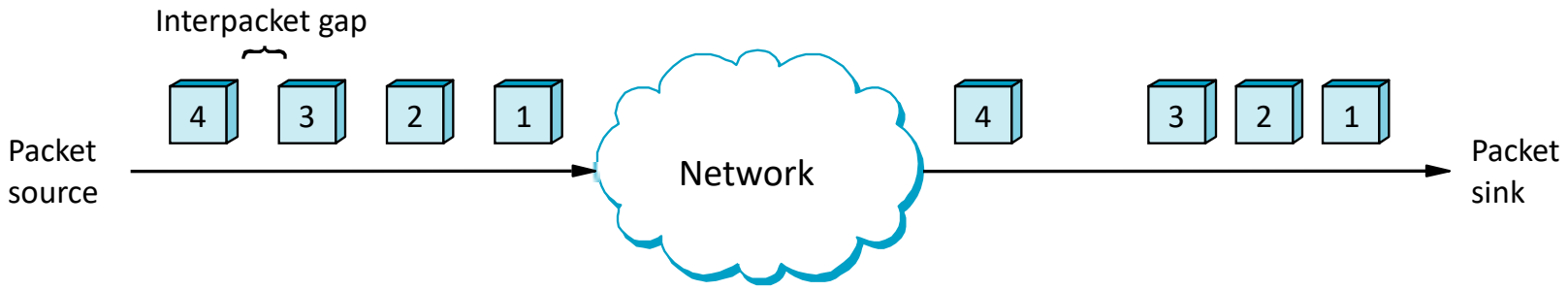
- TCP will have performance problems when there are paths with high bandwidth and long round-trip delays (LFNs)
- The relevant parameter here is the product of bandwidth (bits per second) and round-trip delay (RTT in seconds)
  - Number of bits it takes to "fill the pipe", i.e., the amount of unacknowledged data that TCP must handle to keep the pipe full

Time scales and performance

- Average vs. “instantaneous” bandwidth:
  - Time interval over which average is computed is important
  - Suppose a video application needs 2 Mbps on average:
    - If it transmits 1 Mb in first second and 3 Mb in the following second, over the 2-second interval average rate is 2 Mbps
    - However, just knowing the average may not be enough. What if the network can support no more than 2 Mb in any one second
    - Generally, one puts an upper bound on how large a “burst” can be (Burst: Peak rate maintained for some period of time)

Delay and jitter

- Delay requirement: As little delay as possible
- Jitter requirement: As little variation in delay as possible
- Smoothing out jitter is important for better performance in video applications
  - Jitter can be smoothed out by delaying the time at which playback starts



Network induces jitter!

Cloudification or softwarization of the network

- Network operators are trying to simultaneously accelerate the pace of innovation (known as feature velocity) and yet continue to offer a reliable service (preserve stability)
- Adopt the best practices of cloud providers:
  - Take advantage of commodity hardware and move all intelligence into software
  - Adopt agile engineering processes that break down barriers between development and operations
- Software Defined Networks (SDNs) is a game changer in terms of how rapidly the network evolves to support new features (See more in CMPE 448 Modern Networking Concepts)