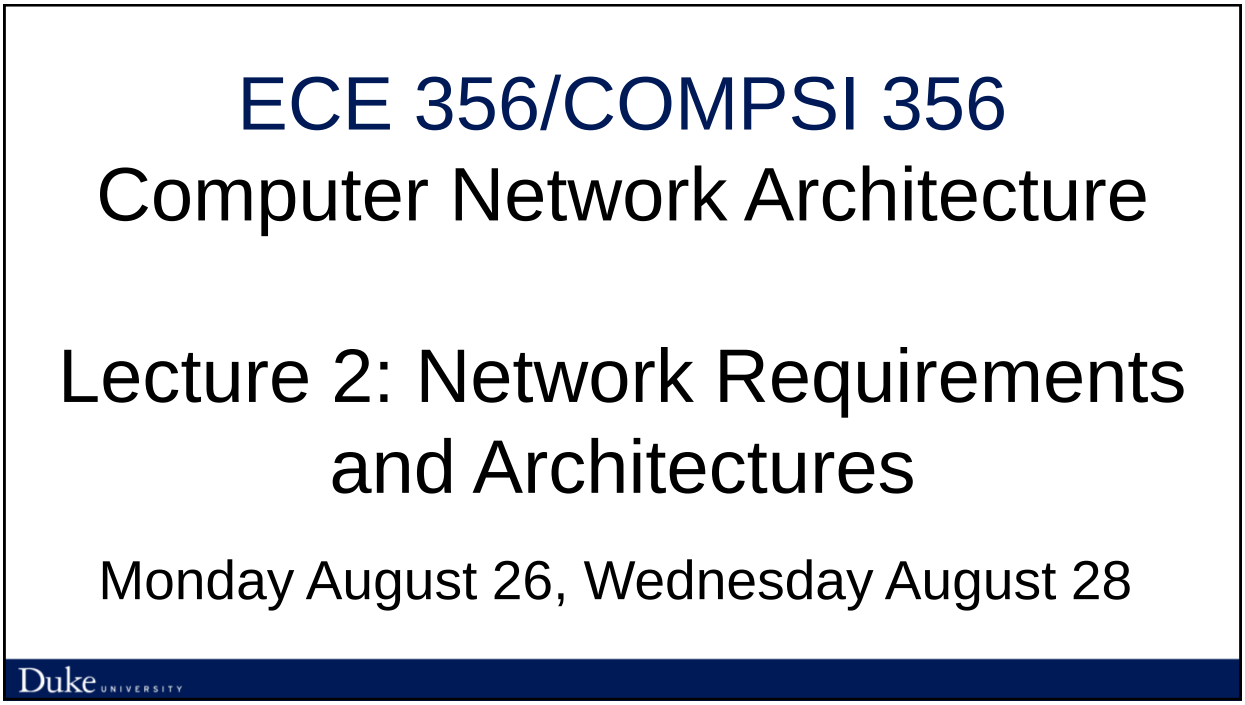
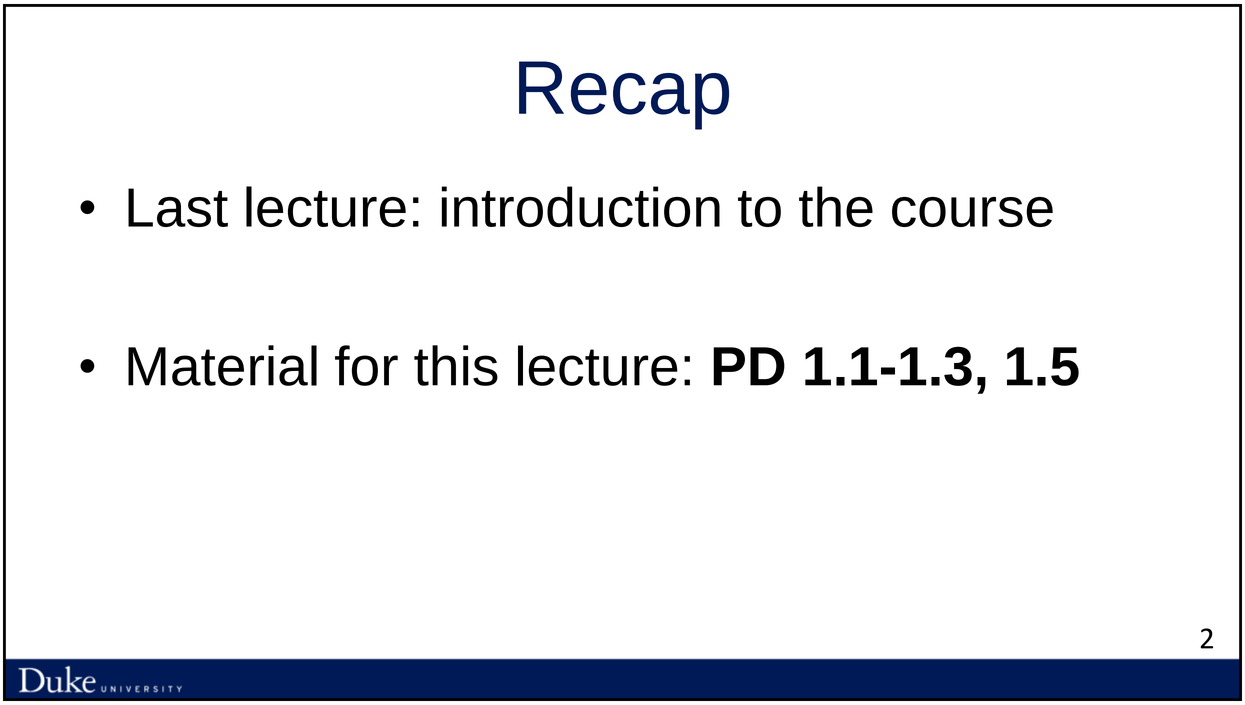


ECE 356/COMPSI 356
Computer Network Architecture

Lecture 2: Network Requirements and Architectures

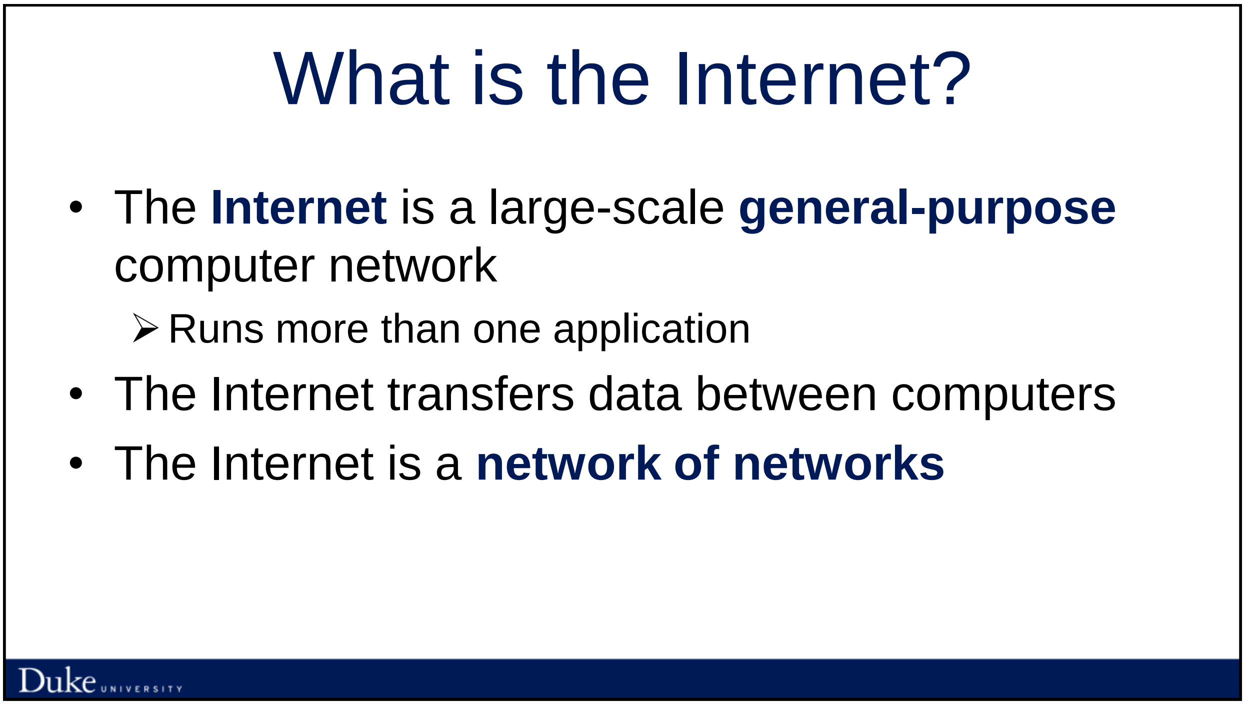
Monday August 26, Wednesday August 28



Recap

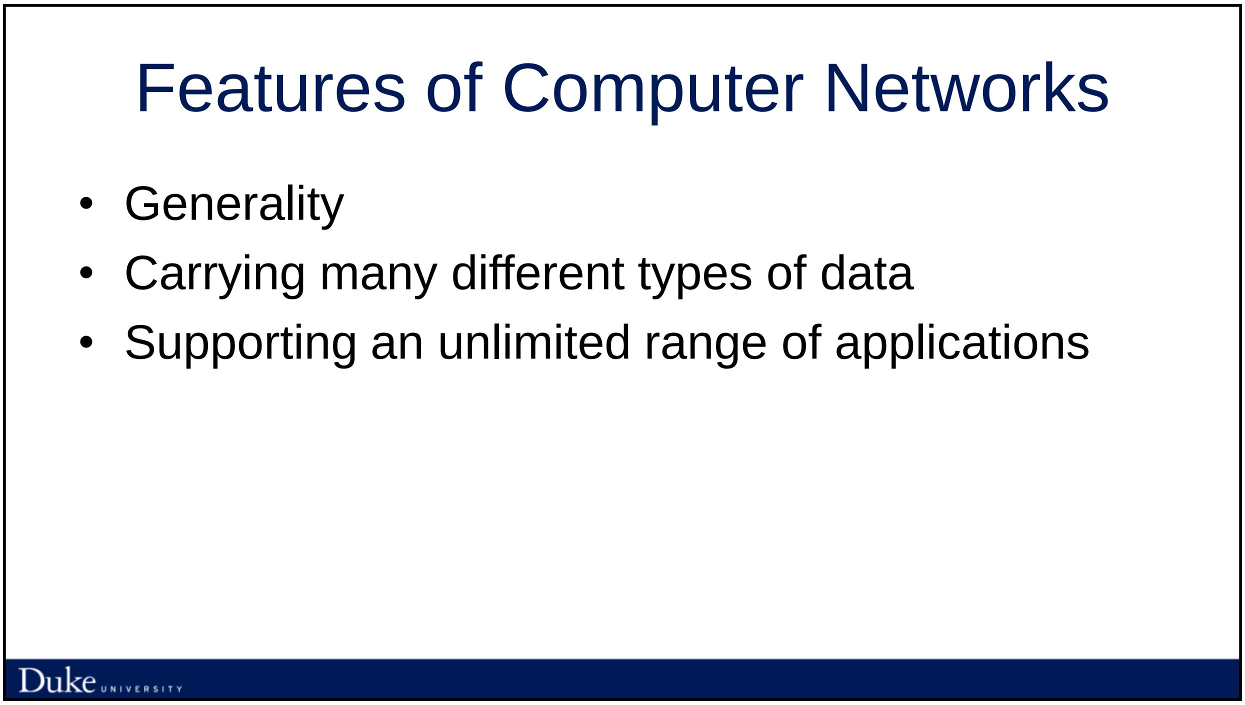
- Last lecture: introduction to the course
- Material for this lecture: **PD 1.1-1.3, 1.5**

2



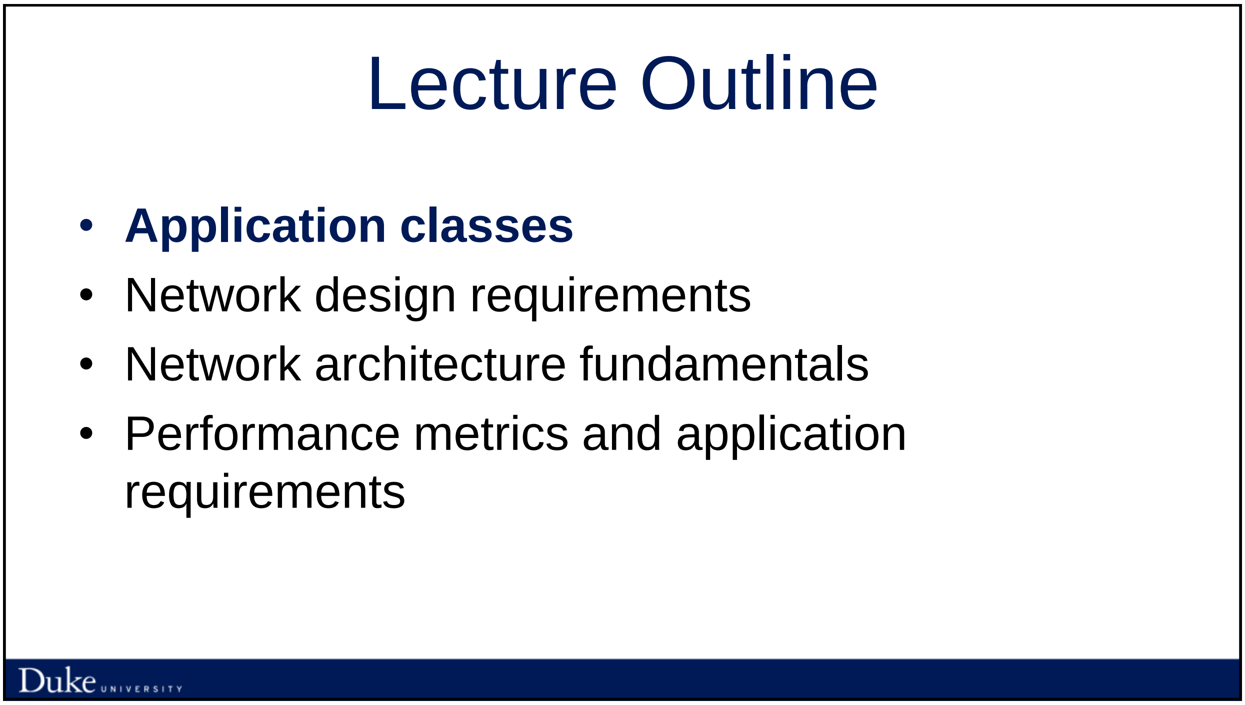
What is the Internet?

- The **Internet** is a large-scale **general-purpose** computer network
  - ➢ Runs more than one application
- The Internet transfers data between computers
- The Internet is a **network of networks**



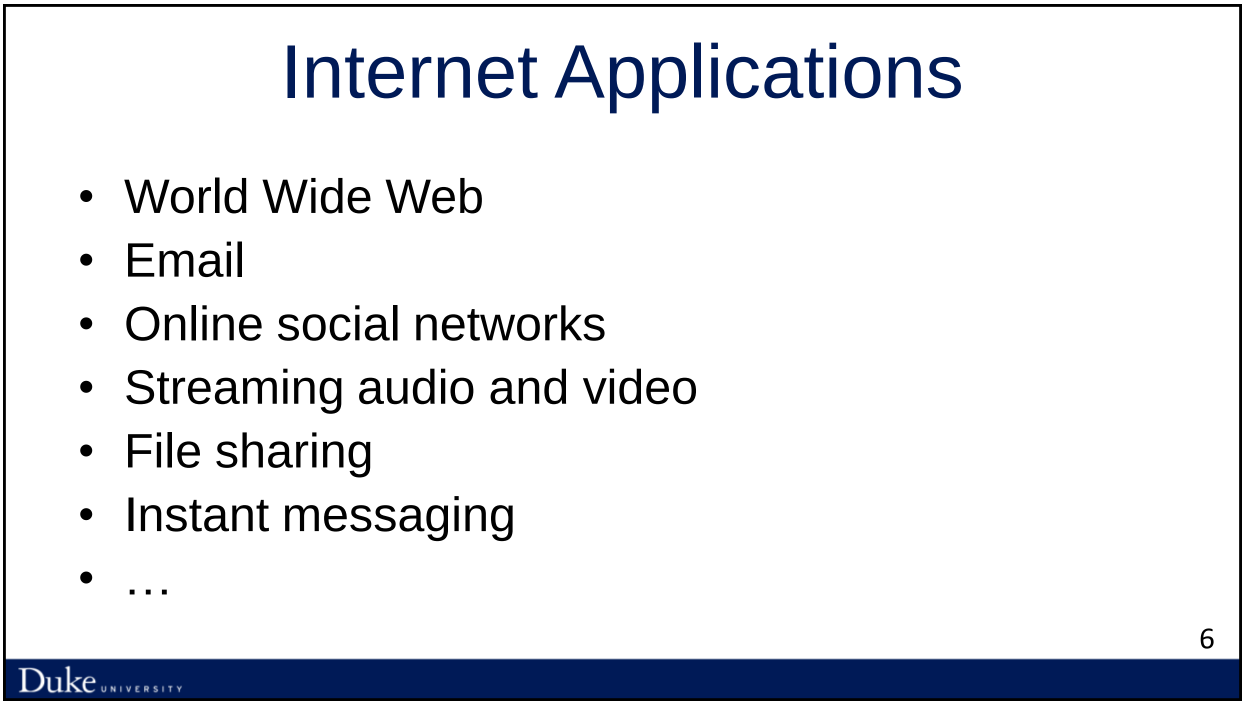
Features of Computer Networks

- Generality
- Carrying many different types of data
- Supporting an unlimited range of applications



Lecture Outline

- **Application classes**
- Network design requirements
- Network architecture fundamentals
- Performance metrics and application requirements



Internet Applications

- World Wide Web
- Email
- Online social networks
- Streaming audio and video
- File sharing
- Instant messaging
- …

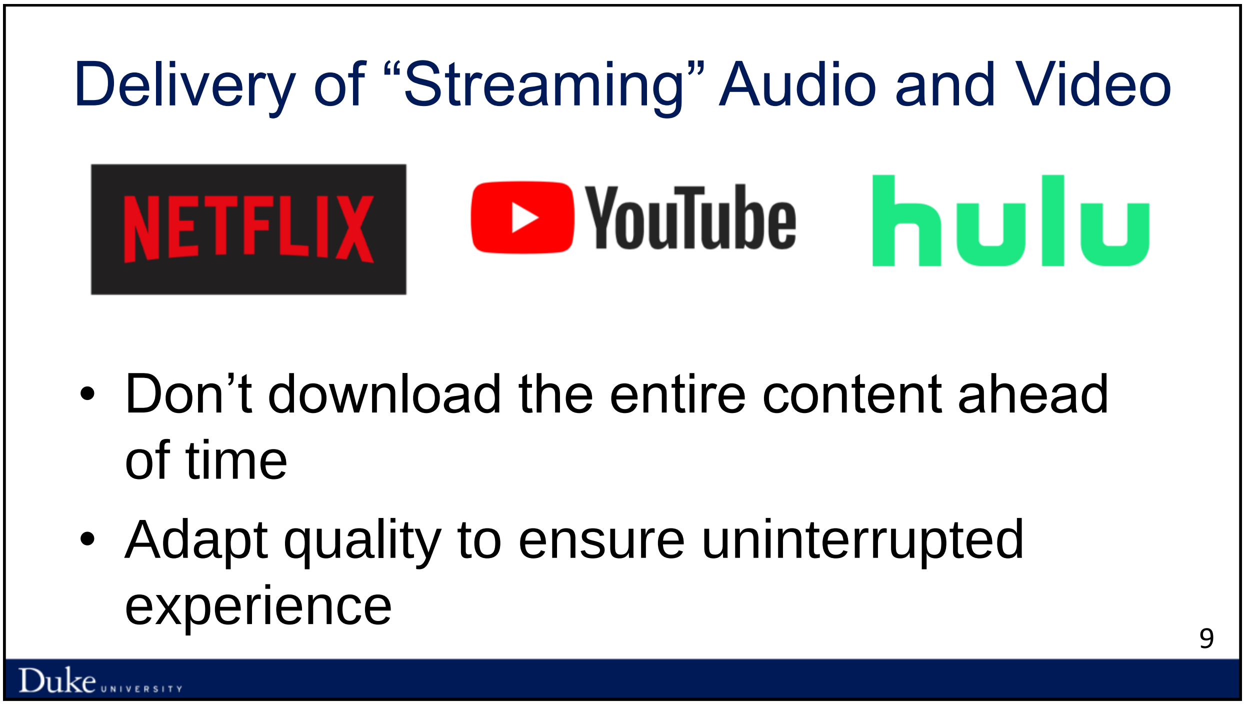
6

Web Browsing: Multiple Steps Under the Hood (1/2)

- E.g., to access http://www.duke.edu
  - ➢ Uniform Resource Locator (URL)
- At least 17 messages for one URL request
  - ➢ 6 to find the Internet Protocol (IP) address
  - ➢ 3 for the establishment of a Transmission Control Protocol (TCP) connection

Web Browsing: Multiple Steps Under the Hood (2/2)

- At least 17 messages for one URL request (cont.)
  - ➢ 4 for HTTP request and acknowledgement
    - Request: I got your request and I will send the data
    - Reply: Here is the data you requested; I got the data
  - ➢ 4 messages for tearing down TCP connection
- 100s more messages for embedded objects, ads
- Continuous traffic for maintaining the connections



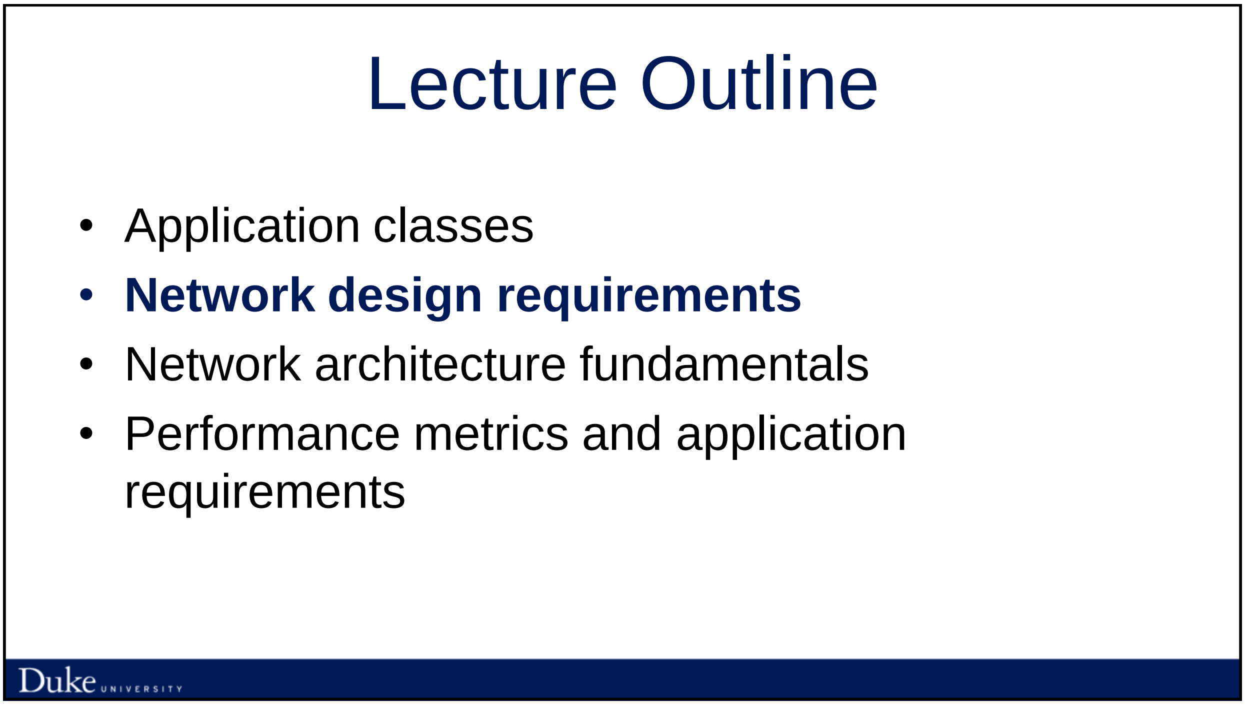
Delivery of “Streaming” Audio and Video

- Don’t download the entire content ahead of time
- Adapt quality to ensure uninterrupted experience

skype™
MICROSOFT
Cisco webex

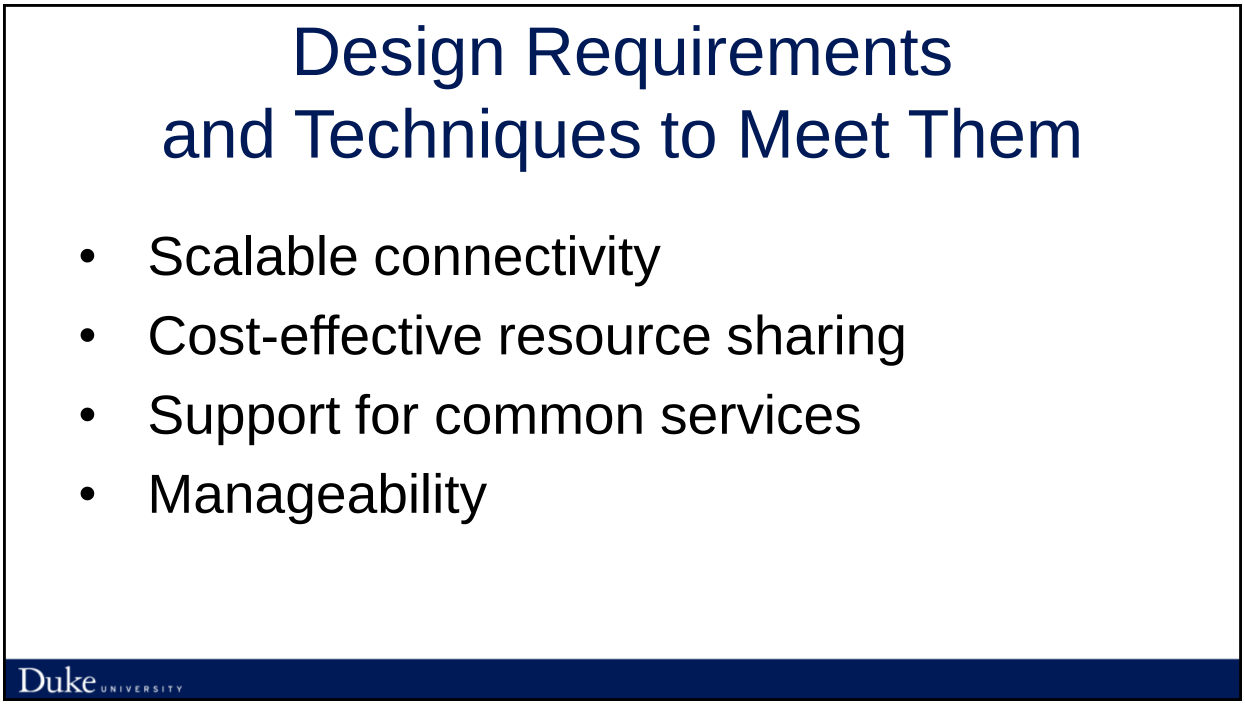
Real-time Audio and Video

- Timely interactions between participants
- Cannot pre-store, pre-code, and pre-distribute the data



Lecture Outline

- Application classes
- **Network design requirements**
- Network architecture fundamentals
- Performance metrics and application requirements



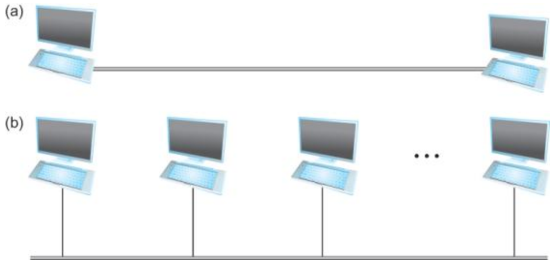
Design Requirements and Techniques to Meet Them

- Scalable connectivity
- Cost-effective resource sharing
- Support for common services
- Manageability

Scalable Connectivity

- A network must provide connectivity among a set of computers
  - Open vs close: to connect all computers or a subset of them?
  - Internet is an open network
- **Scalability**: a system is designed to grow to an arbitrary large size is said to scale
  - How to connect an arbitrary large number of computers on a network?

Connectivity Recursively Occurs at Different Levels



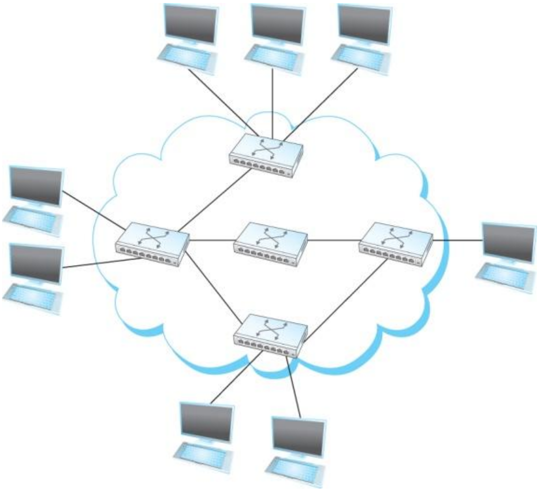
Point-to-Point

Multiple-Access

- Link-level: connect two or more computers via a physical medium or electromagnetic waves in free space
- Computers are referred to as **nodes**
- The physical medium is referred to as a **link**

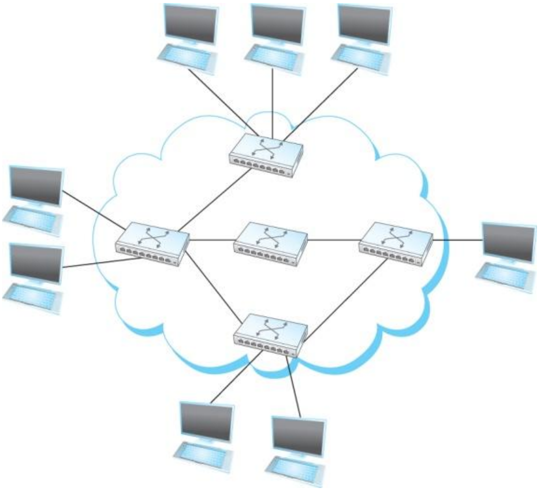
Switching

- Switching is a mechanism to achieve connectivity
- Nodes that are attached to at least two links forward data from one link to another link
- They are called **switches**
- Computers outside the cloud are called **hosts**



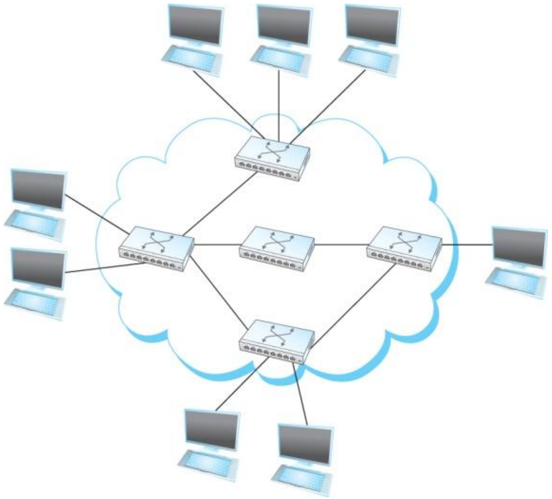
Switching: Circuit Switching

- Sets up a circuit before nodes can communicate
- Switches connect circuits on different links



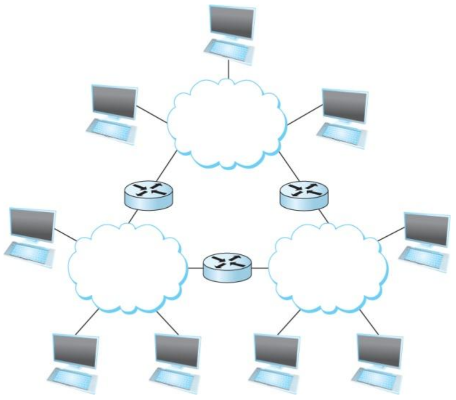
Switching: Packet Switching

- Data are split into discrete blocks of data called **packets**
- Store and forward
- Nodes send packets and switches forward them



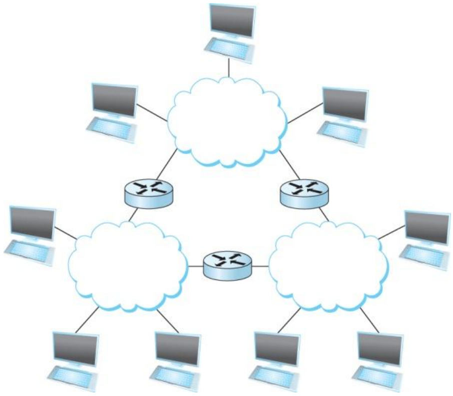
Inter-networking: Another Way to Achieve Connectivity

- An internetwork of networks
  - ➢ Each cloud is a network/a multiple-access link
  - ➢ A node that is connected to two or more networks is commonly called a **router**
    - Speaks different protocols than switches
  - ➢ An **internet** can be viewed as a “cloud”
  - ➢ Can recursively build larger clouds by connecting smaller ones



Addressing and Routing

- Physical connectivity != connectivity
- Addressing and routing are mechanisms to achieve connectivity
- Nodes are assigned addresses
- Routers compute how to reach them by running routing protocols



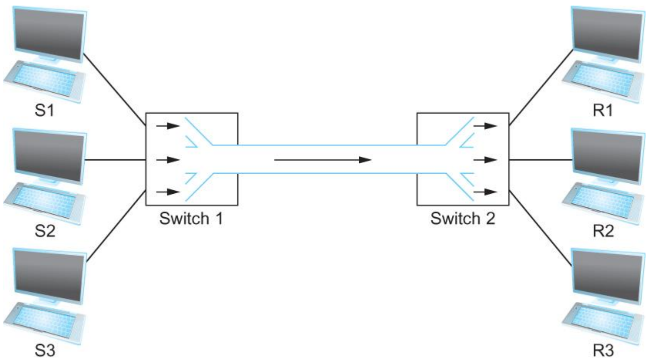
Design Requirements and Techniques to Meet them

- Scalable connectivity
- **Cost-effective resource sharing**
- Support for common services
- Manageability

Cost-effective Resource Sharing

- Question: how do all the hosts share the network when they want to communicate with each other?
  - Use at the same time
  - Fair
- **Multiplexing**: a system resource is shared among multiple users
  - Analogy: CPU sharing

Multiplexing Multiple Flows on a Single Link



Multiplex Demultiplex

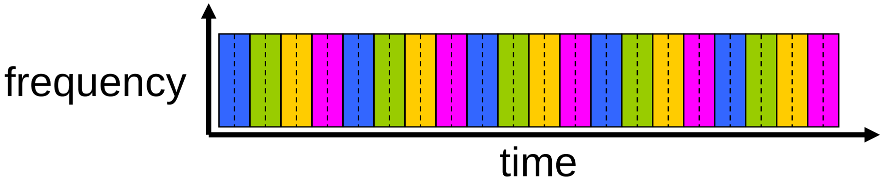
Multiplexing Mechanisms

- Time-division multiplexing (TDM)
- Frequency-division multiplexing (FDM)
- Statistical multiplexing
- …

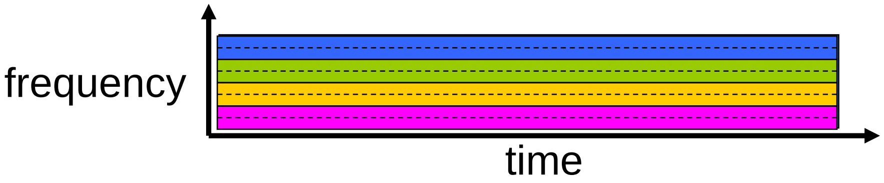
TDM and FDM

Example:
4 users 🟦 🟩 🟨 🟪

TDM



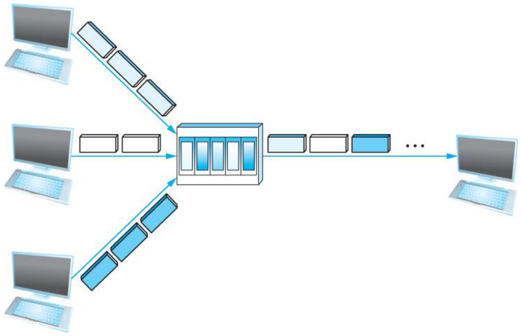
FDM



Problems with TDM and FDM

- What if a user does not have data to send all the time (over-provision)?
  - ➢ Consider web browsing
  - ➢ Inefficient use of resources
- Max # of flows is fixed and known ahead of time (under-provision)
  - ➢ Not practical to change the size of quantum or add additional quanta for TDM
  - ➢ Nor add more frequencies in FDM

Statistical Multiplexing



- The physical link is shared over time (like TDM)
- But does not have fixed pattern. This is called ***statistical multiplexing***
  - ➢ Sequence of A & B packets are sent on demand, not predetermined slots

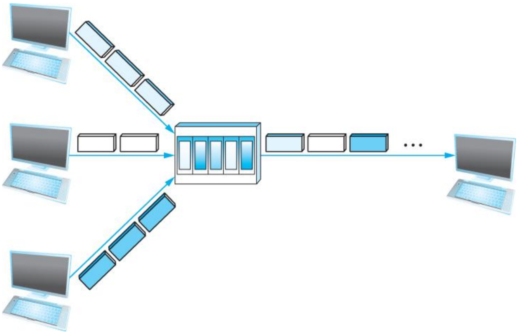
Statistical Multiplexing: Pros and Cons

- Assumption: traffic is largely bursty
- Pros: avoids idle time
- Cons: no guarantee flows would have their turns to transmit
- Some possible fixes:
  - ➢Limit maximum packet size
  - ➢Scheduling which packets got transmitted, e.g., fair queuing

Maximum Packet Size

- Divide an application message into blocks of data → **packets**
  - ➢Names at different layer: segments, frames
- Maximum packet size limit
  - ➢Flows sent on demand
  - ➢Must give each flow its turn to send
  - ➢Defines an upper bound on the size of the block of data

Packet Scheduling

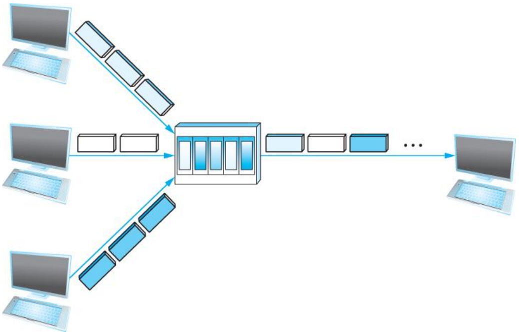


- Scheduling: which packet to send
  - ➢ First come first serve (FIFQ)
  - ➢ Weighted fair queuing

Switching vs. Multiplexing

- TDM and FDM are used in circuit switching networks
  - ➢ Require a setup as max # of flows is fixed
- Statistical multiplexing is used in packet switching networks

Congestion



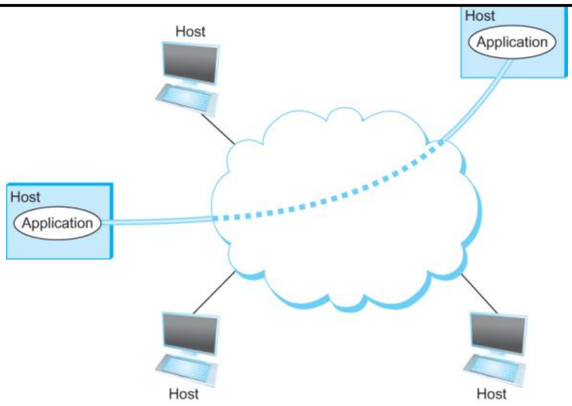
- Aggregate incoming rate > outgoing rate
- An open question
- A large buffer can help alleviate temporary congestion

Design Requirements and Techniques to Meet Them

- Scalable connectivity
- Cost-effective resource sharing
- **Support for common services**
- Manageability

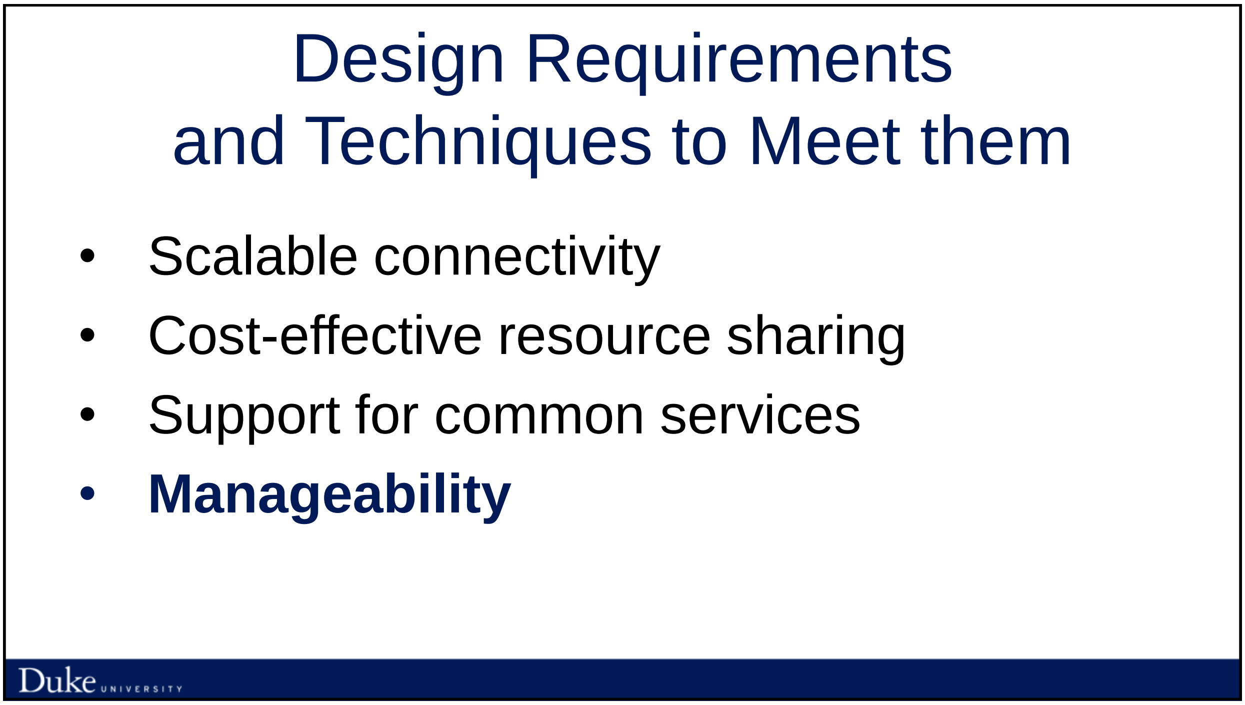
Support for Common Services

- Application developers want a network to provide services that make application programs communicate with each other, not just sending packets
  - E.g. reliably delivering an email message from a sender to a receiver
- Many complicated things need to happen
  - Can you name a few?
- Design choices
  - Application developers build all functions they need
  - Network provides common services → a layered network architecture
    - Build it once, and shared many times



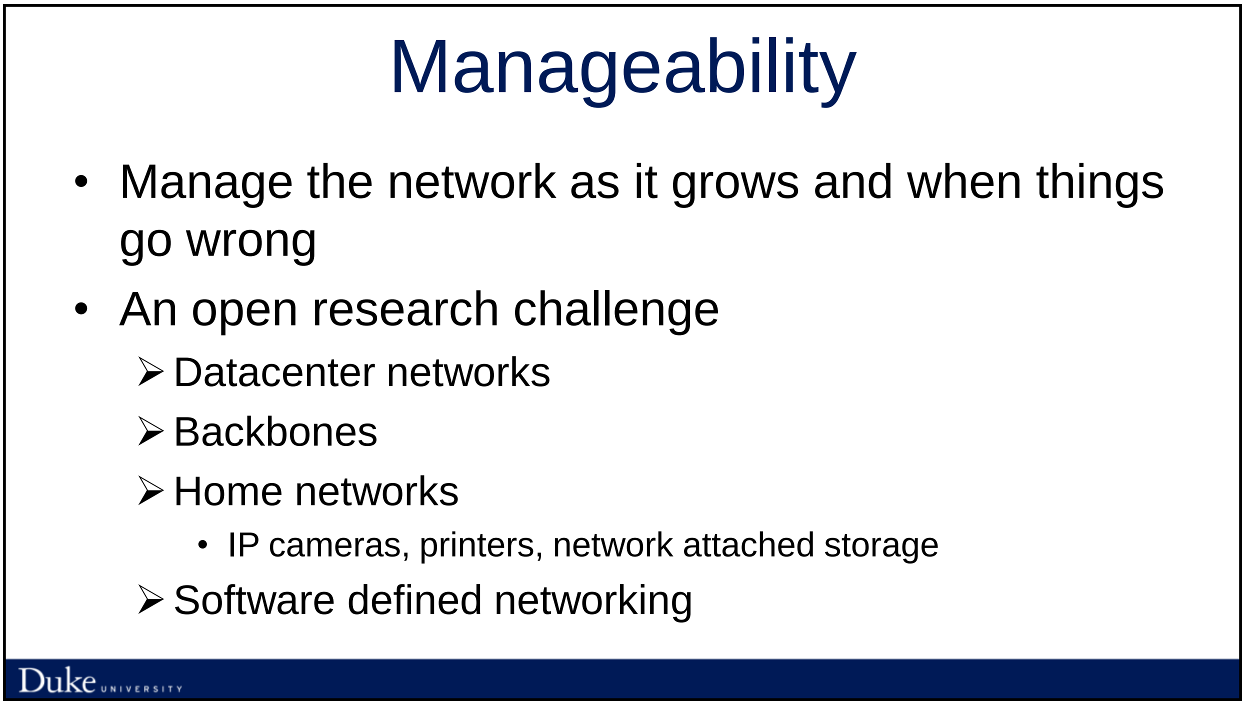
- Interactive request/reply
- Streaming of data
- Bulk data transfer
- …

- Key challenges: what services/channels to provide that can satisfy most applications at lowest costs?
- Approach: identify common patterns, then decide
  - **What** functions to implement, and **where**



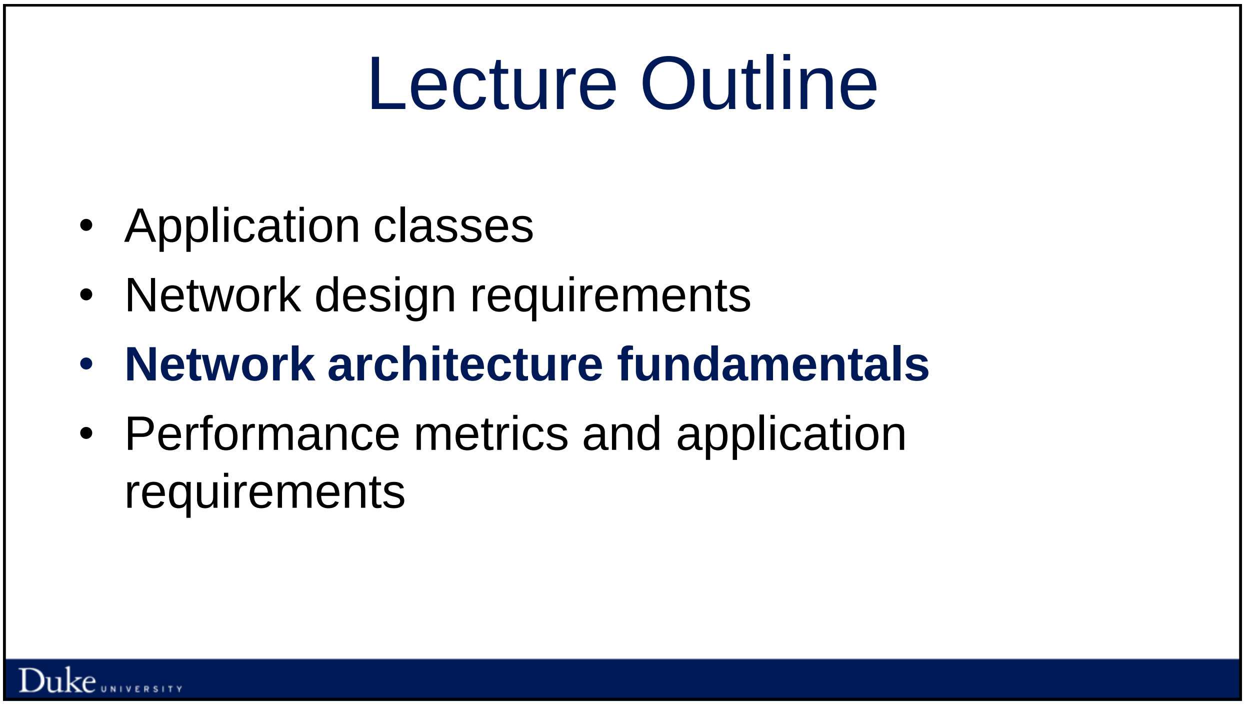
Design Requirements and Techniques to Meet them

- Scalable connectivity
- Cost-effective resource sharing
- Support for common services
- **Manageability**



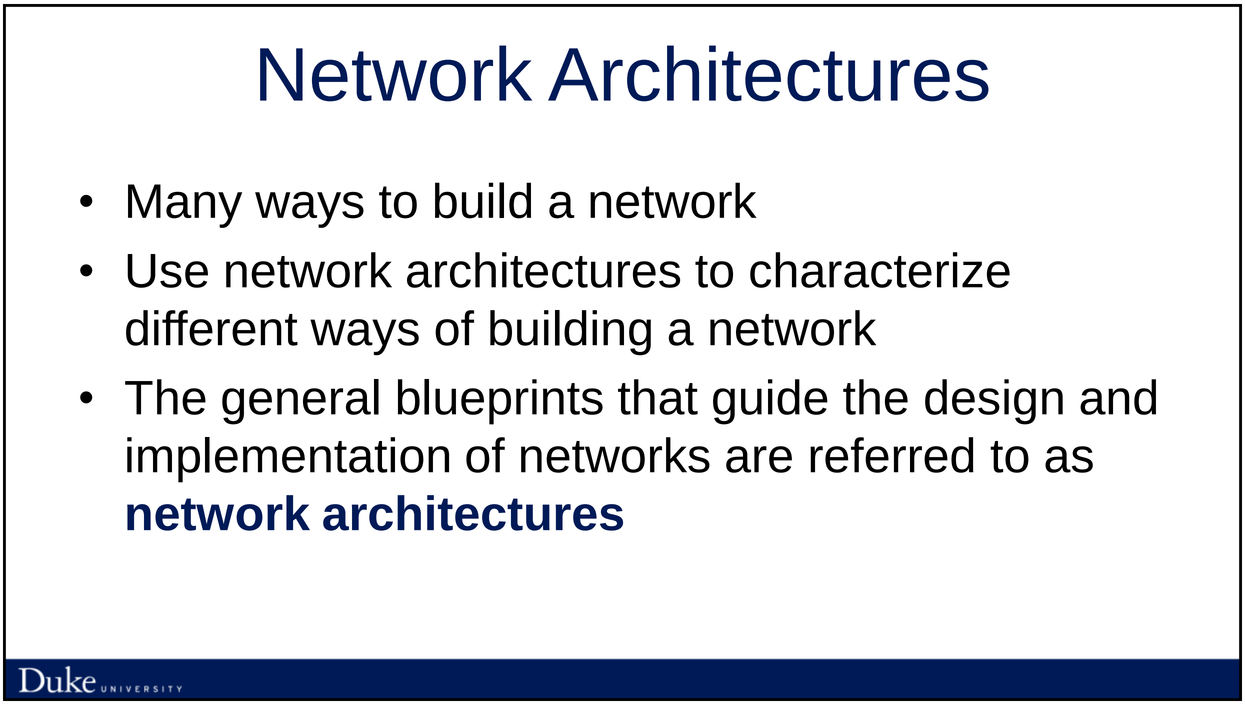
Manageability

- Manage the network as it grows and when things go wrong
- An open research challenge
  - ➢ Datacenter networks
  - ➢ Backbones
  - ➢ Home networks
    - IP cameras, printers, network attached storage
  - ➢ Software defined networking



Lecture Outline

- Application classes
- Network design requirements
- **Network architecture fundamentals**
- Performance metrics and application requirements



Network Architectures

- Many ways to build a network
- Use network architectures to characterize different ways of building a network
- The general blueprints that guide the design and implementation of networks are referred to as **network architectures**

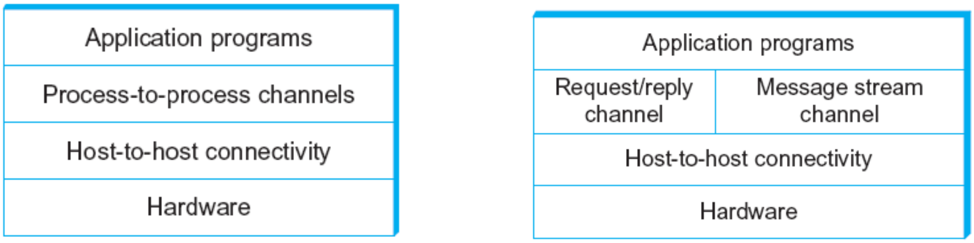
Central Concepts

- Layering
- Protocols

A Layered Architecture

- Many sub-tasks need to be accomplished
  - ➢ Find a path to the destination, reliably transfer information
- The complexity of the communication task is reduced by using **multiple protocol layers**:
  - ➢ Each protocol is implemented independently
  - ➢ Each protocol is responsible for a specific subtask
  - ➢ Protocols are grouped in a hierarchy
- The old **divide-and-conquer** principle!

Layering



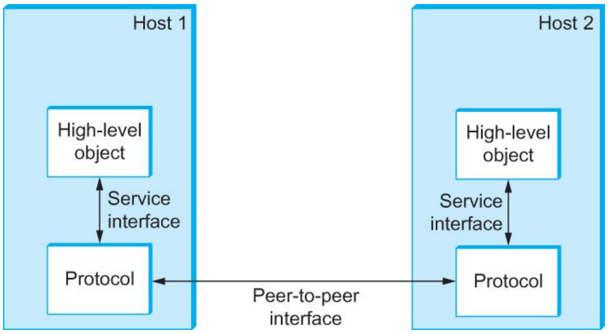
- An abstraction to handle complexity
  - ➢ A unifying model that capture important aspect of a system
  - ➢ Encapsulate the model in an object that has an interface for others to interact with
  - ➢ Hide the details from the users of the object

Advantages of Layering

- Simplify the design tasks
  - ➢ Each layer implements simpler functions
- Modular design
  - ➢ Can provide new services by modifying one layer

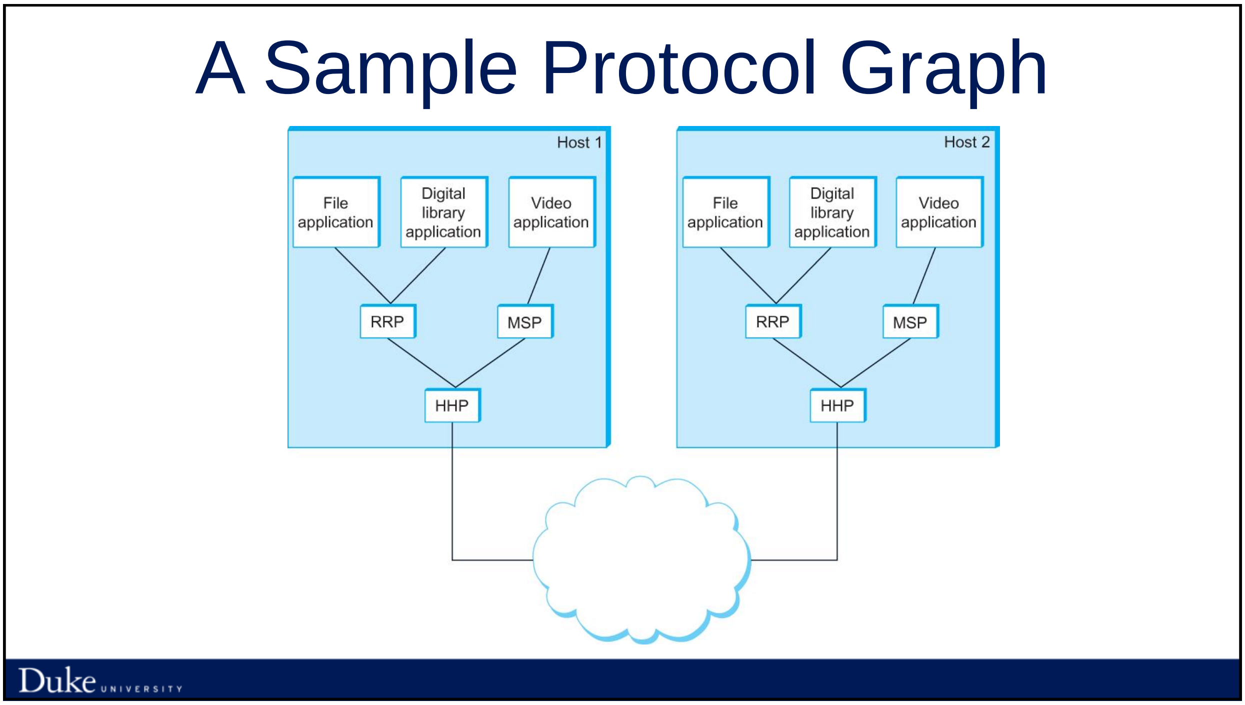
Protocols

- The abstract objects that make up the layers of a network system are called **protocols**
- Each protocol defines two different interfaces
  - Service interface
  - Peer interface

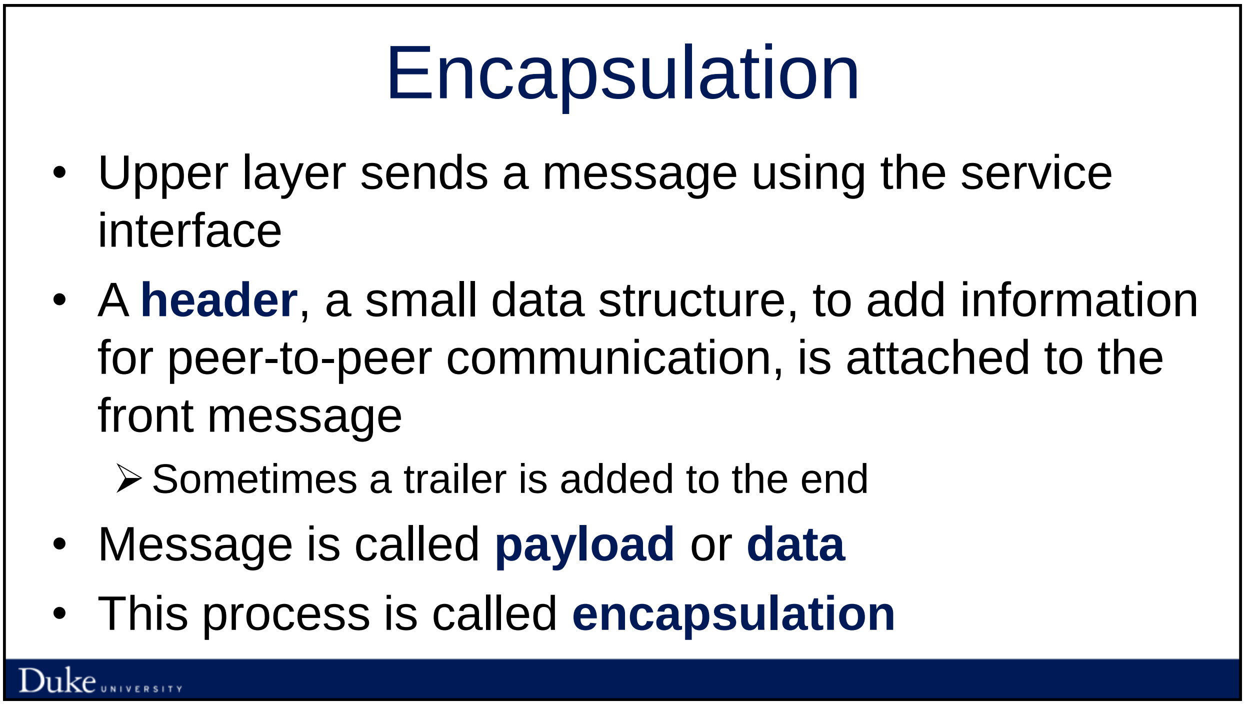


A Protocol Graph

- Peer-to-peer communication is indirect
  - Except at the hardware level
- Potentially multiple protocols at each level
- Show the suite of protocols that make up a network system with a **protocol graph**



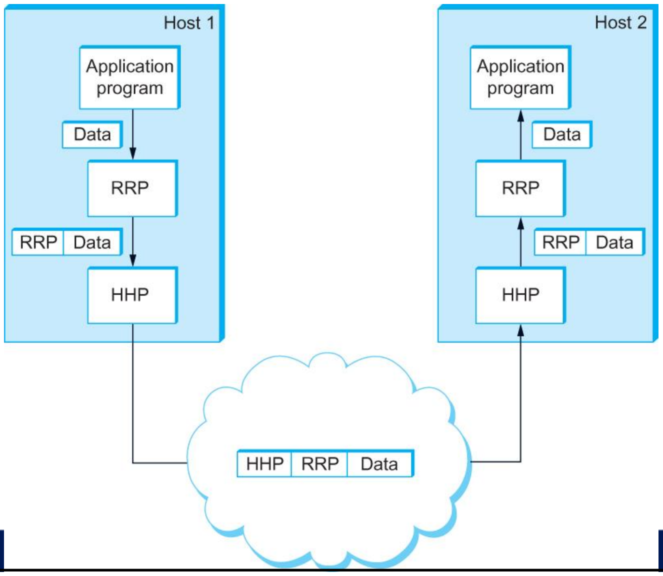
A Sample Protocol Graph



Encapsulation

- Upper layer sends a message using the service interface
- A **header**, a small data structure, to add information for peer-to-peer communication, is attached to the front message
  - ➢ Sometimes a trailer is added to the end
- Message is called **payload** or **data**
- This process is called **encapsulation**

Encapsulation: An Example



Lecture Outline

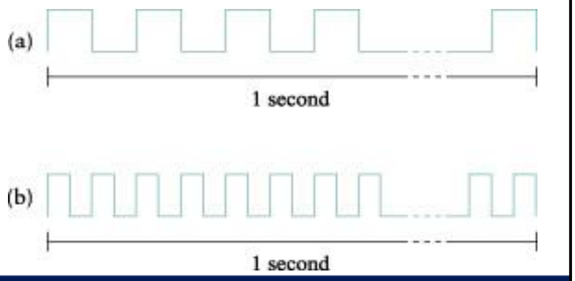
- Application classes
- Network design requirements
  - Scalable connectivity
  - Cost-effective resource sharing
  - Support for common services
  - Manageability
- Network architecture fundamentals
- **Performance metrics and application requirements**

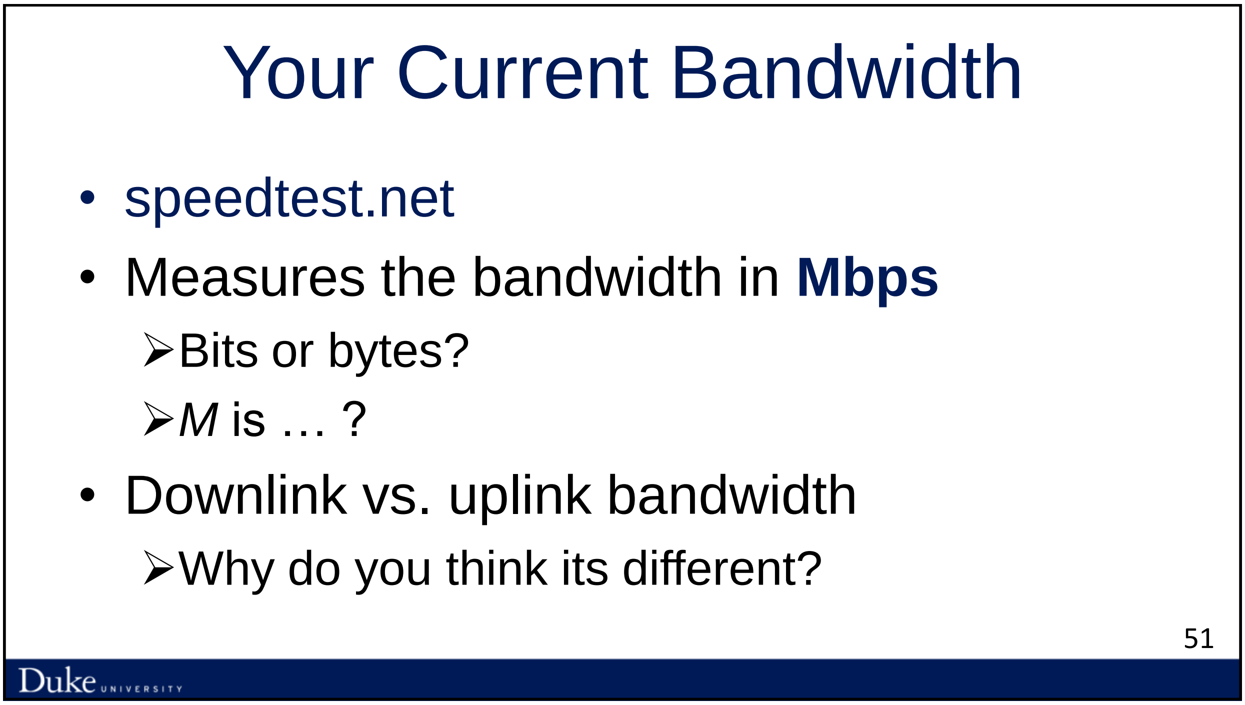
Performance

- So far: how it works
- Now: how **well** it works
- Core metrics:
  - ➢Bandwidth a.k.a. throughput
  - ➢Latency a.k.a. delay
- Application needs

Bandwidth

- Bandwidth is a measure of the width of a frequency band. E.g., a telephone line supports a frequency band 300-3300 Hz has a bandwidth of 3000 Hz
- Bandwidth of a link normally refers to the **number of bits it can transmit in a unit time**
  - ➢ A second of time as distance
  - ➢ Each bit as a pulse of width

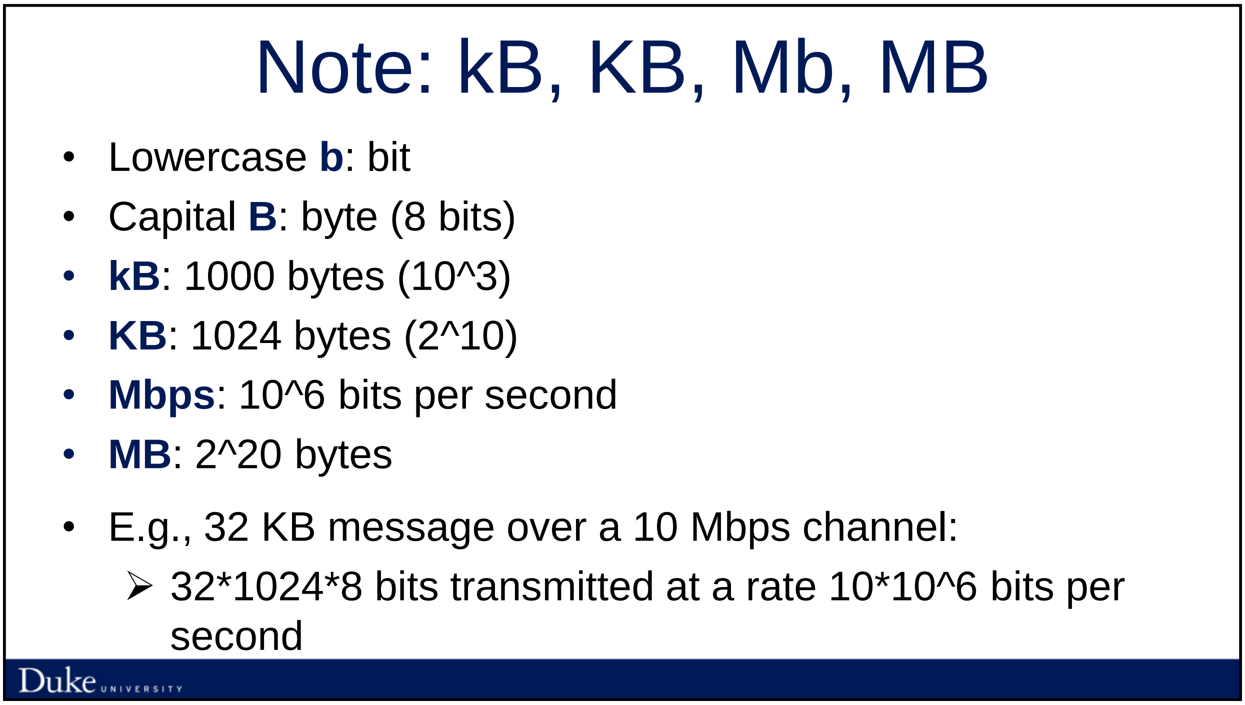




Your Current Bandwidth

- speedtest.net
- Measures the bandwidth in **Mbps**
  - Bits or bytes?
  - *M* is … ?
- Downlink vs. uplink bandwidth
  - Why do you think its different?

51



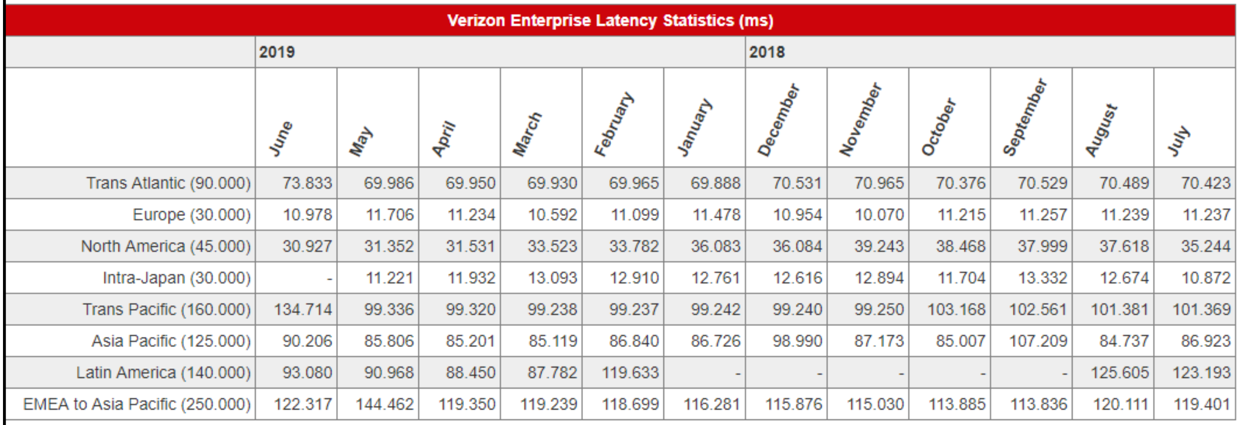
Note: kB, KB, Mb, MB

- Lowercase **b**: bit
- Capital **B**: byte (8 bits)
- **kB**: 1000 bytes (10^3)
- **KB**: 1024 bytes (2^10)
- **Mbps**: 10^6 bits per second
- **MB**: 2^20 bytes
- E.g., 32 KB message over a 10 Mbps channel:
  - 32*1024*8 bits transmitted at a rate 10*10^6 bits per second

Latency

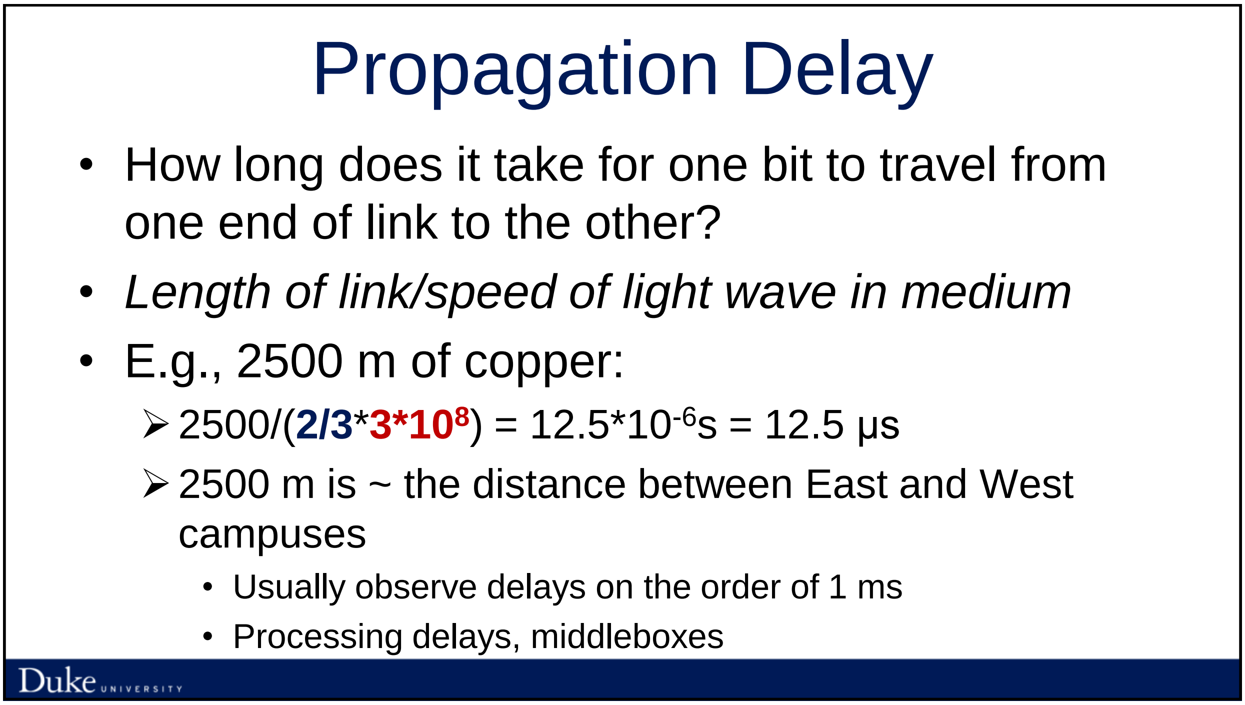
- How long it takes for a message to travel from one end of the network to the other
- Often use Round-Trip Time (RTT) as a latency measure

RTT Latency: Examples



Verizon Enterprise Latency Statistics (ms)												
	2019						2018					
	June	May	April	March	February	January	December	November	October	September	August	July
Trans Atlantic (90.000)	73.833	69.986	69.950	69.930	69.965	69.888	70.531	70.965	70.376	70.529	70.489	70.423
Europe (30.000)	10.978	11.706	11.234	10.592	11.099	11.478	10.954	10.070	11.215	11.257	11.239	11.237
North America (45.000)	30.927	31.352	31.531	33.523	33.782	36.083	36.084	39.243	38.468	37.999	37.618	35.244
Intra-Japan (30.000)	-	11.221	11.932	13.093	12.910	12.761	12.616	12.894	11.704	13.332	12.674	10.872
Trans Pacific (160.000)	134.714	99.336	99.320	99.238	99.237	99.242	99.240	99.250	103.168	102.561	101.381	101.369
Asia Pacific (125.000)	90.206	85.806	85.201	85.119	86.840	86.726	98.990	87.173	85.007	107.209	84.737	86.923
Latin America (140.000)	93.080	90.968	88.450	87.782	119.633	-	-	-	-	-	125.605	123.193
EMEA to Asia Pacific (250.000)	122.317	144.462	119.350	119.239	118.699	116.281	115.876	115.030	113.885	113.836	120.111	119.401

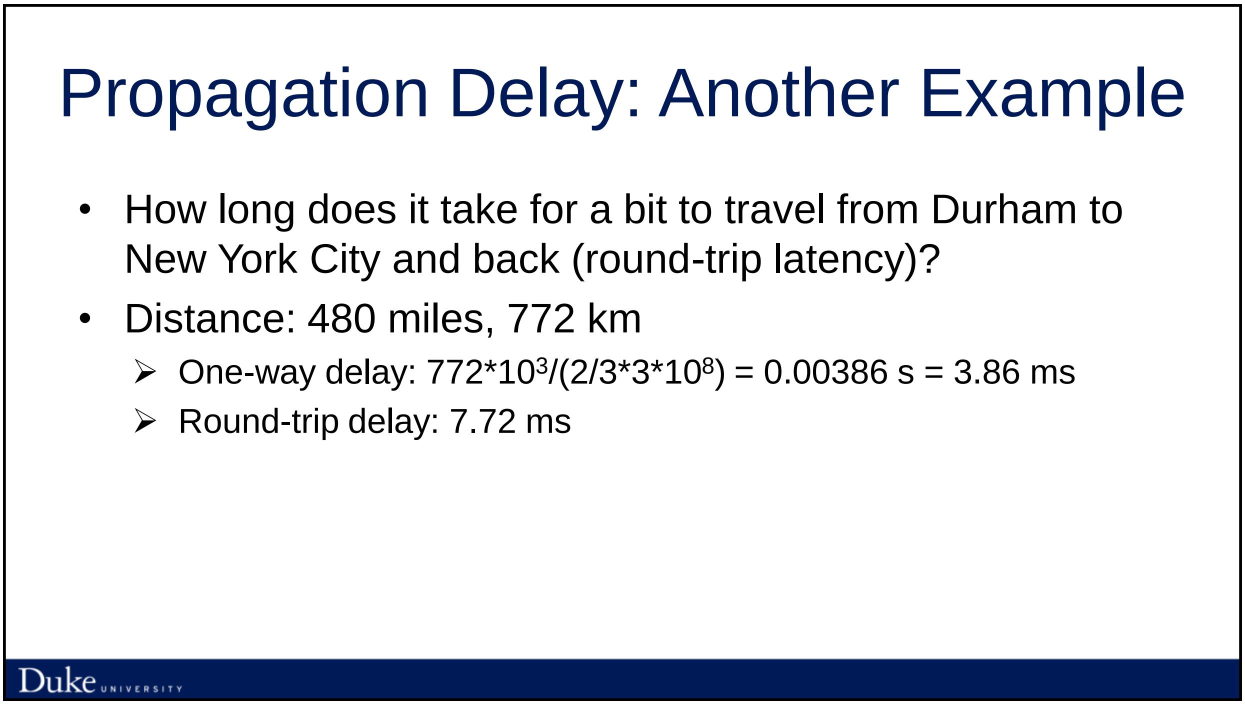
- 100 ms feels instantaneous to web users



Propagation Delay

- How long does it take for one bit to travel from one end of link to the other?
- *Length of link/speed of light wave in medium*
- E.g., 2500 m of copper:
  - $2500/(\mathbf{2/3}*\mathbf{3*10^{8}}) = 12.5*10^{-6}$s = 12.5 μs
  - 2500 m is ~ the distance between East and West campuses
    - Usually observe delays on the order of 1 ms
    - Processing delays, middleboxes

Duke UNIVERSITY



Propagation Delay: Another Example

- How long does it take for a bit to travel from Durham to New York City and back (round-trip latency)?
- Distance: 480 miles, 772 km
  - One-way delay: $772*10^{3}/(2/3*3*10^{8})$ = 0.00386 s = 3.86 ms
  - Round-trip delay: 7.72 ms

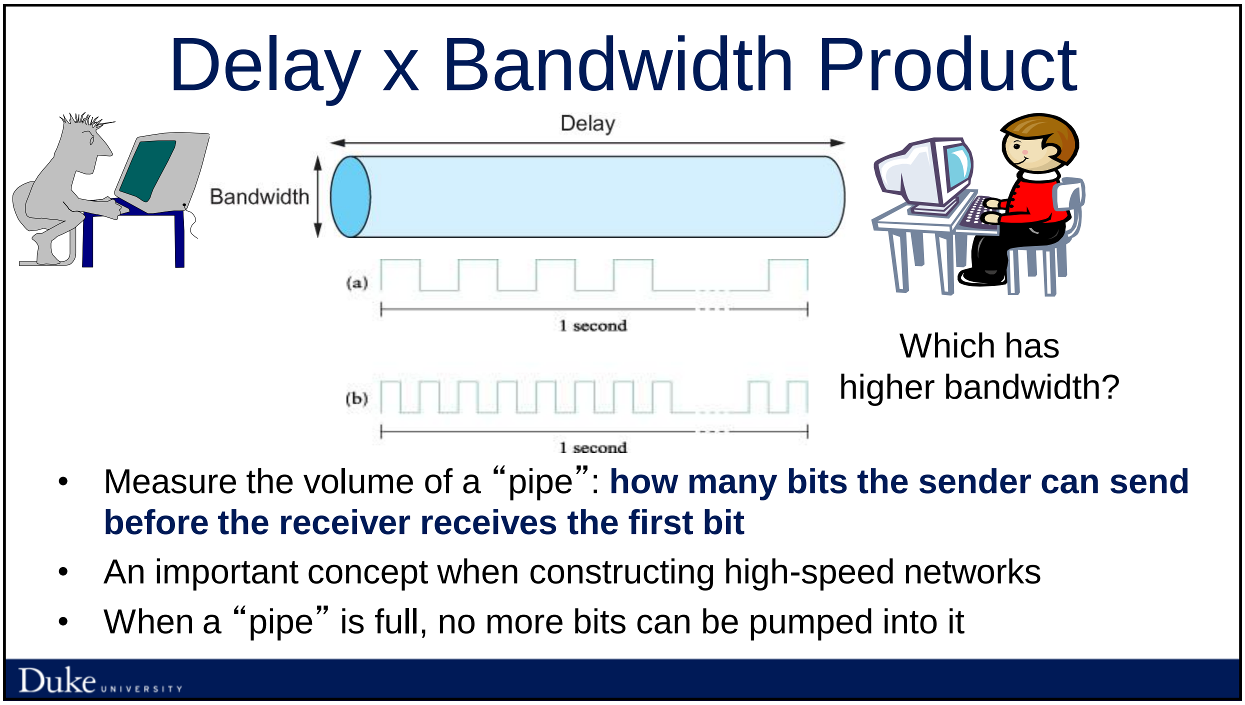
Duke UNIVERSITY

Latency

- Latency = Propagation + transmit + queue
- Propagation = distance/speed of light
- Transmit = size/bandwidth

Bandwidth & Latency as Performance Metrics

- Relative importance depends on the application
  - ➢ One bit transmission => propagation is important
  - ➢ Large bytes transmission => bandwidth is important
- Bandwidth is ever-increasing, while latency is bounded by the speed of light

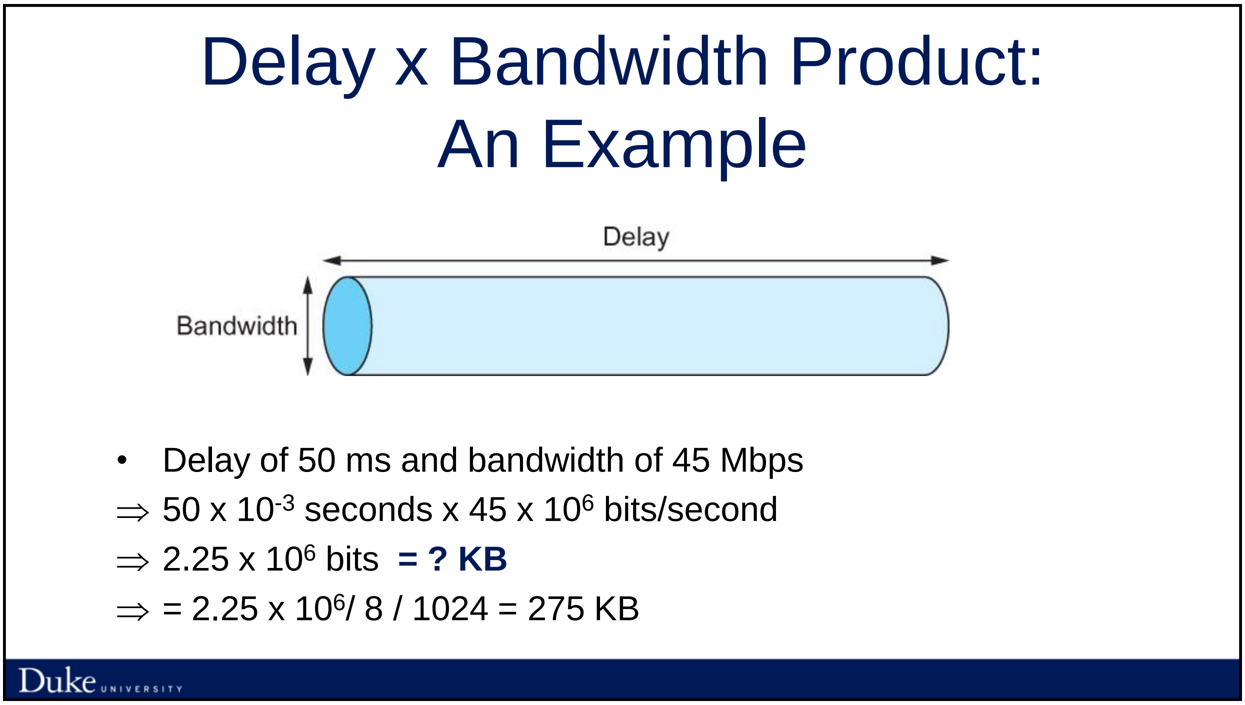


Delay x Bandwidth Product

Delay
Bandwidth
(a)
1 second
(b)
1 second

Which has higher bandwidth?

- Measure the volume of a “pipe”: **how many bits the sender can send before the receiver receives the first bit**
- An important concept when constructing high-speed networks
- When a “pipe” is full, no more bits can be pumped into it



Delay x Bandwidth Product: An Example

Delay
Bandwidth

- Delay of 50 ms and bandwidth of 45 Mbps

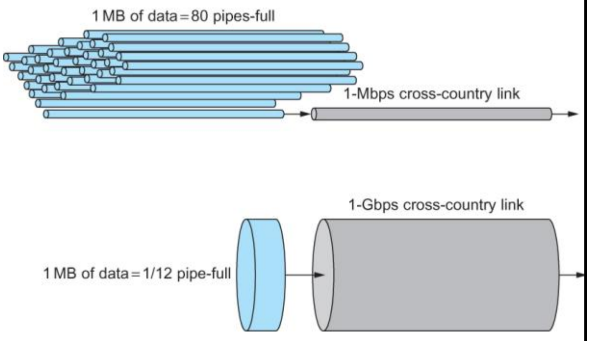
⇒ 50 x 10^{-3} seconds x 45 x 10^{6} bits/second

⇒ 2.25 x 10^{6} bits **= ? KB**

⇒ = 2.25 x 10^{6}/ 8 / 1024 = 275 KB

High Speed vs. Low Speed Links

- A high speed link can send more bits in a unit time than a low speed link
- Link throughput calculations:
  - Effective throughput = Transfer size / transfer time
  - Transfer time = RTT + Transfer size x (1/Bandwidth)


1 MB of data = 80 pipes-full
1-Mbps cross-country link
1-Gbps cross-country link
1 MB of data = 1/12 pipe-full

RTT = 100 ms

Duke UNIVERSITY

Link Throughput Calculations

- BW = 1 Gbps, RTT = 100 ms, 1MB data
- Delay x bandwidth product = $1*10^9$ bps * 0.1s = 100 Mb
- Transfer time = RTT + Transfer size x (1/Bandwidth) = 100ms + 1MB x 1/1Gbps = 108 ms
- Effective throughput = Transfer size/ transfer time = 1MB/108ms = 74.1Mbps
  - Why is it less than 1Gbps?

Duke UNIVERSITY

Application Performance Needs: Bandwidth and Latency

- Sometimes well-known and/or bounded
- E.g., capture and transmit images for object recognition with *AlexNet*:
  - Frame size 256x256 pixels as per AlexNet specifications
  - Each pixel represented in RBG: 3 values of 8 bits each
  - (3x8x256x256)/8 = 196,608 bytes = 192 KB
- E.g, Internet webpage response latency: 100 ms round-trip
  - Humans perceive it as immediate

Application Performance Needs: Jitter

- Latency variation: deviation from the mean
- Jitter is problematic for voice, gaming, video conferencing, control, augmented reality, …

Lecture Summary

- Application classes
- Network design requirements
- Network architecture fundamentals
- Performance metrics and application requirements

Next Lecture

- Internet architecture