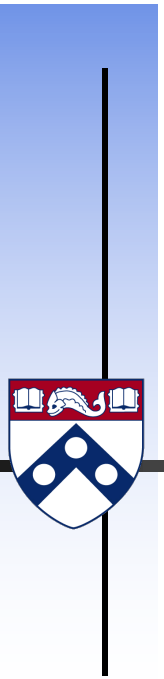


CIS 553: Networked Systems

Link Layer

February 3, 2021

Agenda

- Design goals of the Internet ✓
  - Layering and the end-to-end principle ✓
- Physical Layer ✓
  - Media types ✓
  - Performance Metrics NEXT
- Link Layer
  - Frame format and framing
  - Error detection
  - Medium access control
  - Switching

Performance Metric: Latency

- Consists of four components
  - Transmission delay
  - Propagation delay
  - Queuing delay
  - Processing delay

Transmission delay, Propagation delay: due to link properties

Queuing delay, Processing delay: due to traffic mix and switch internals

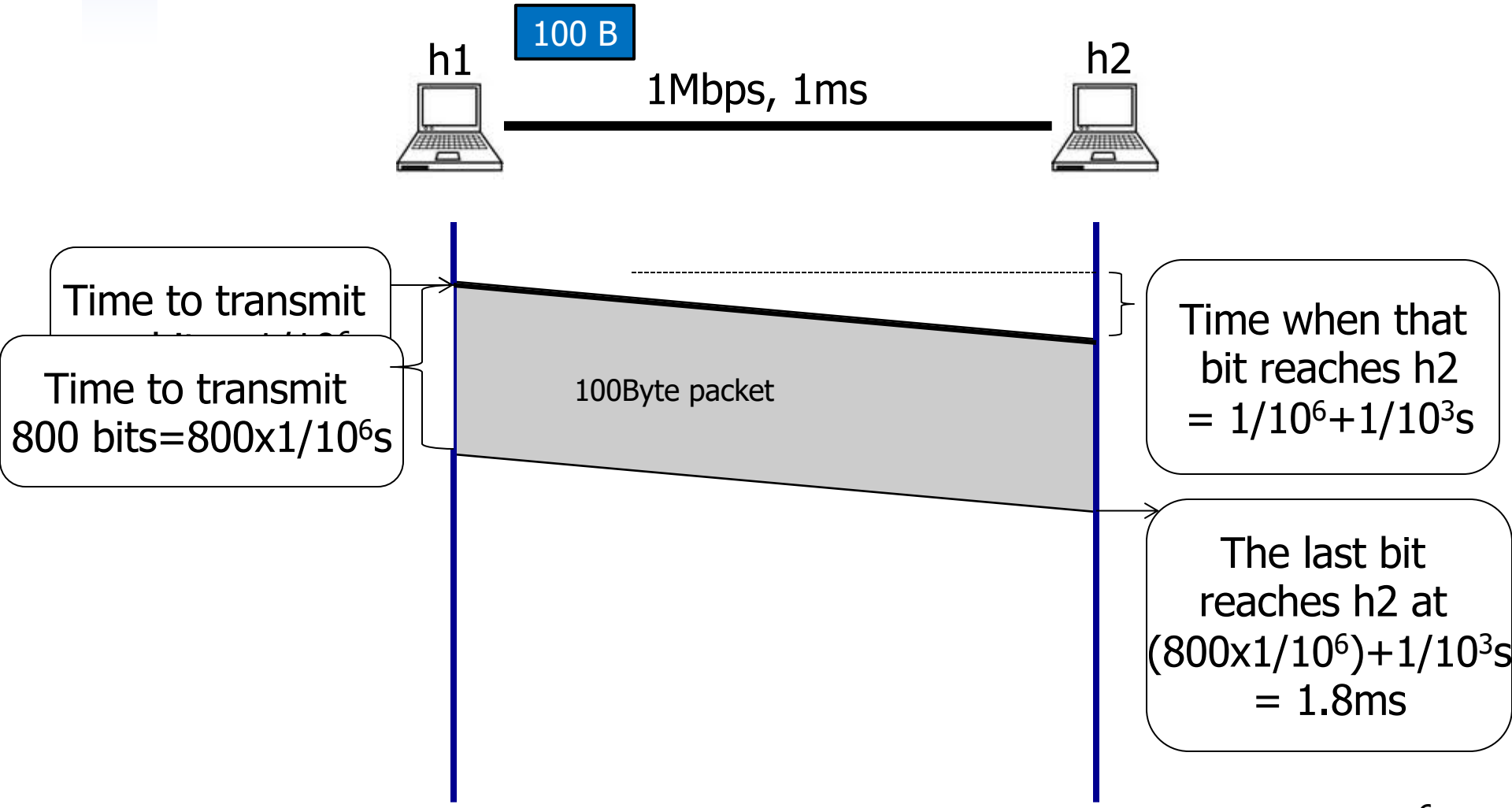
1. Transmission delay

- How long does it take to push all the bits of a message into a link?
- Message size / bandwidth of the link
  - e.g., 1000 bits / 100 Mbits per sec = 10 us

2. Propagation delay

- How long does it take to move one bit from one end of a link to the other?
- Link length / Propagation speed of link
  - E.g., 30 kilometers / $3*10^8$ meters per sec = 100 us

Computing link delay



Computing link delay

Time to transmit

Time to transmit 800 bits=800x1/10^6s

100Byte packet

Time when that bit reaches h2 = 1/10^9+1/10^3s

The last bit reaches h2 at (800x1/10^9)+1/10^3s = 1.0008ms

Performance Metric: Latency

- Consists of four components
  - Transmission delay
  - Propagation delay
  - Queuing delay
  - Processing delay

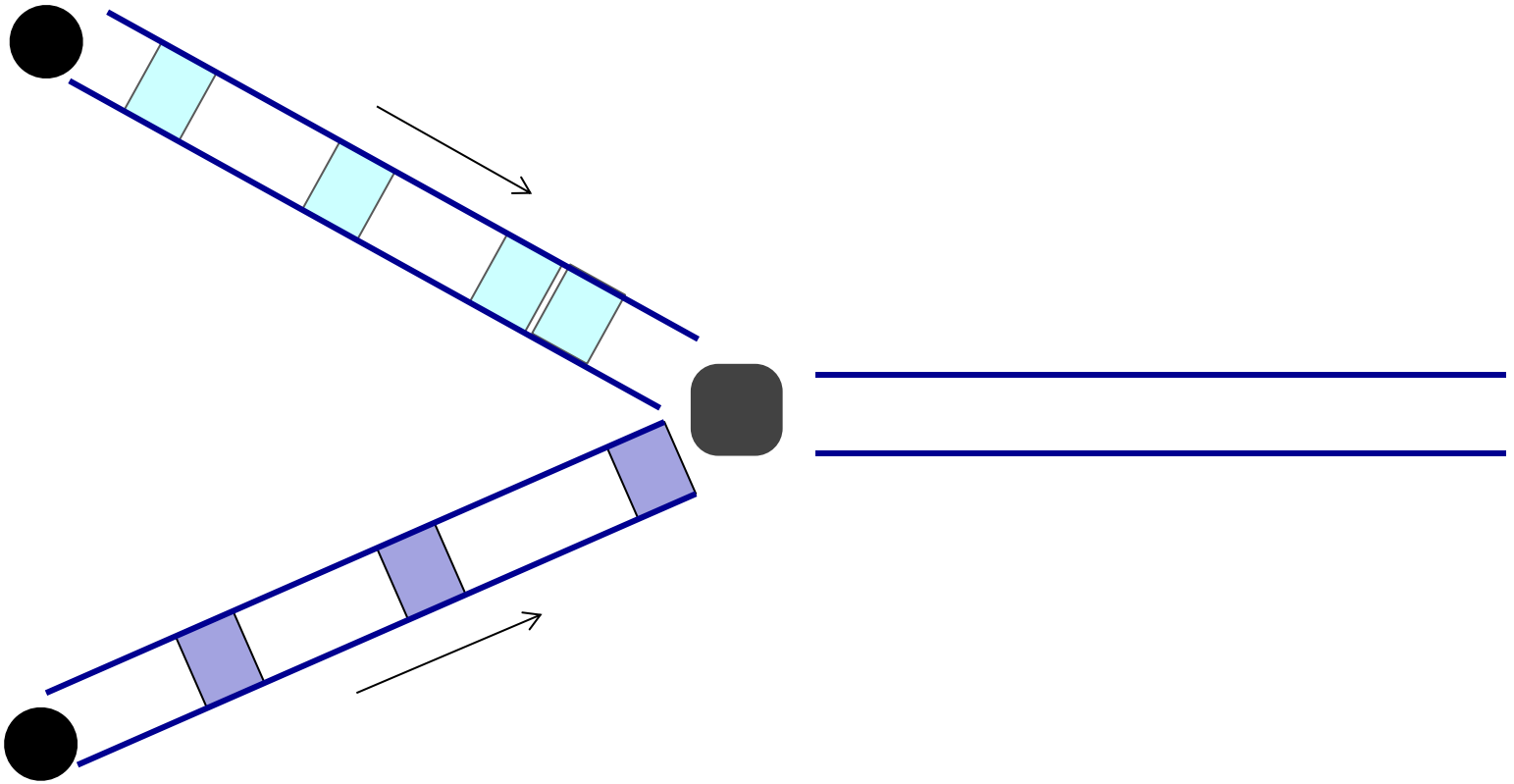
due to link properties

due to traffic mix and switch internals

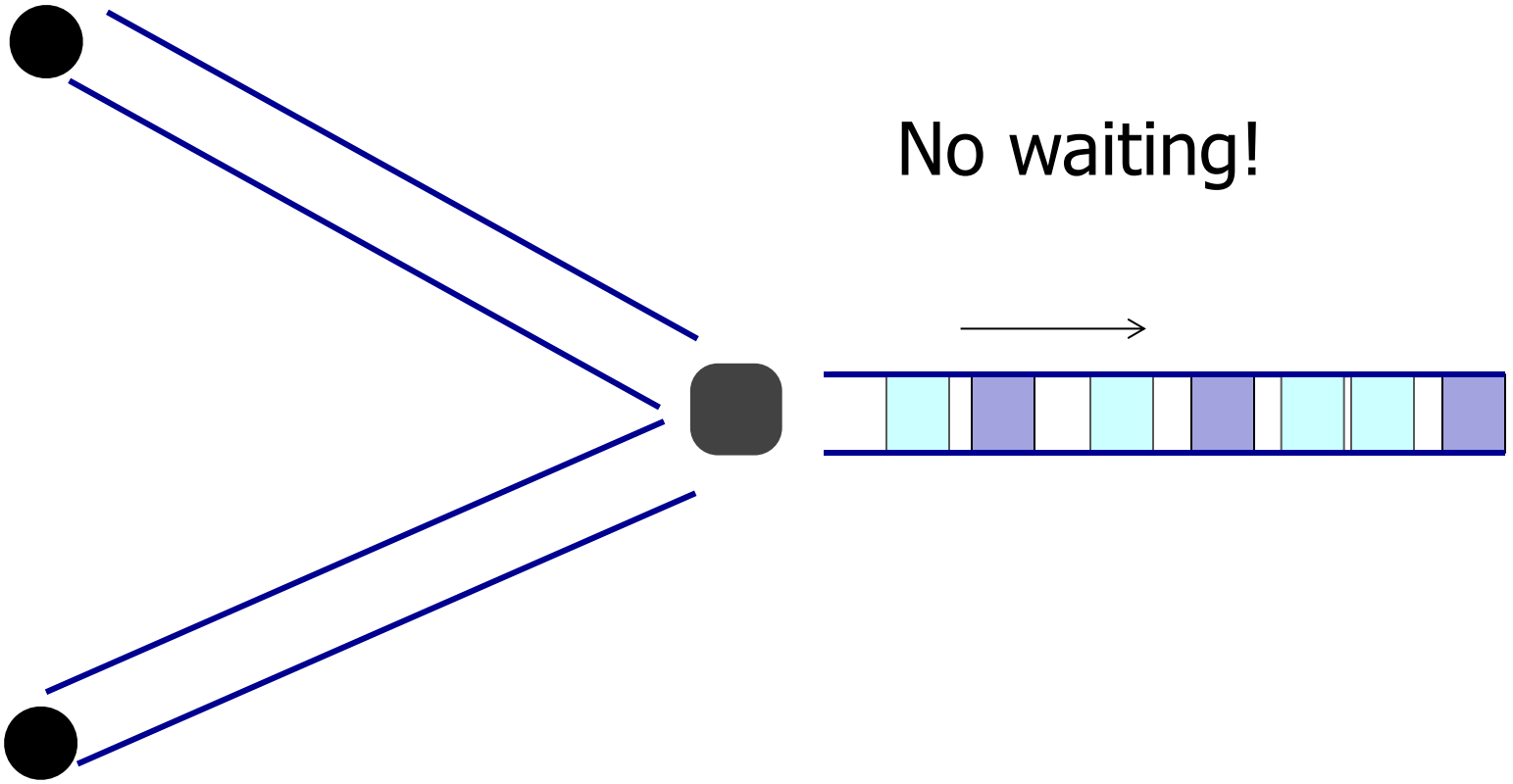
3. Queuing delay

- How long a message needs to wait before being processed

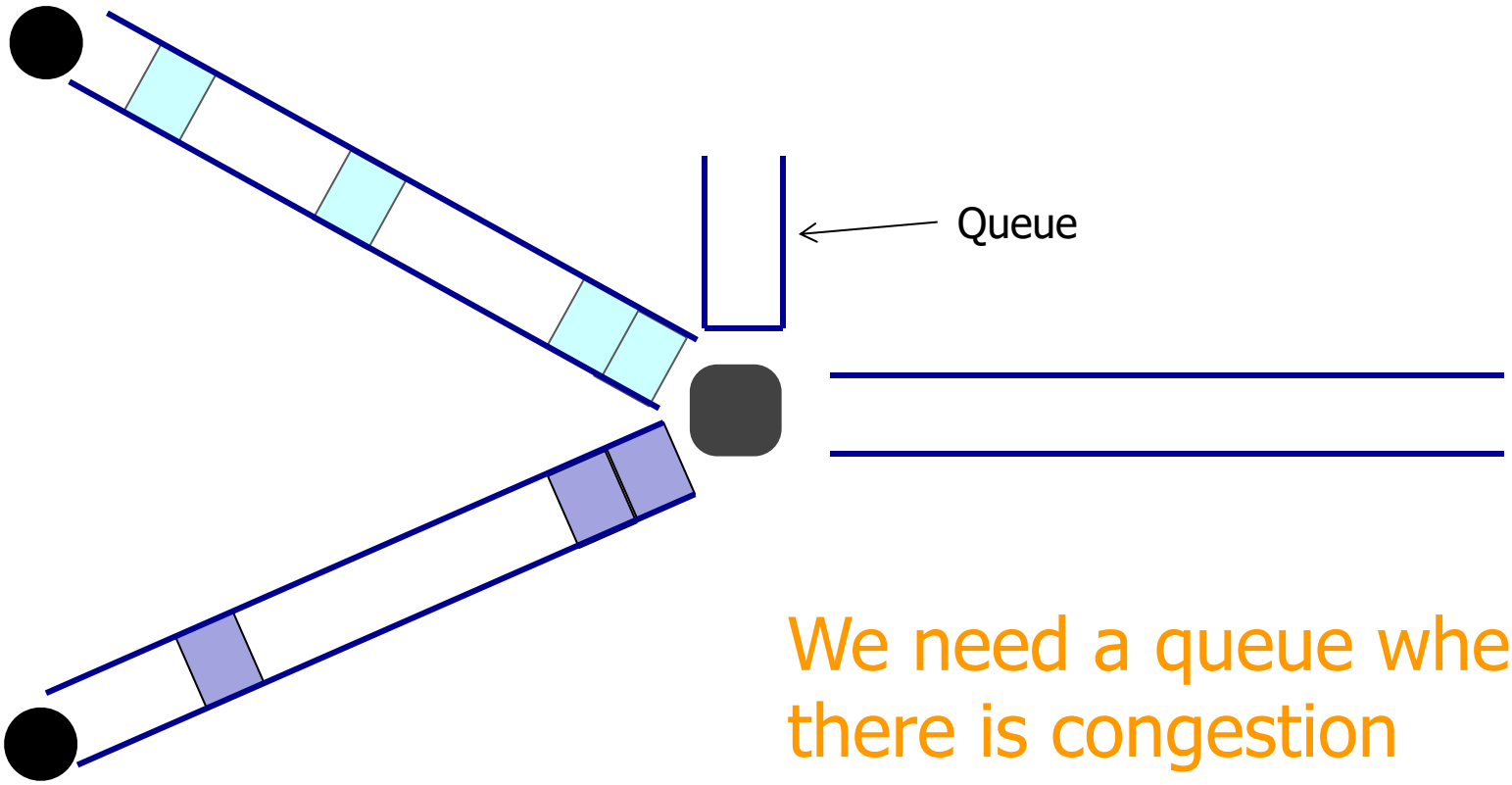
Why do messages need to wait?



Why do messages need to wait?

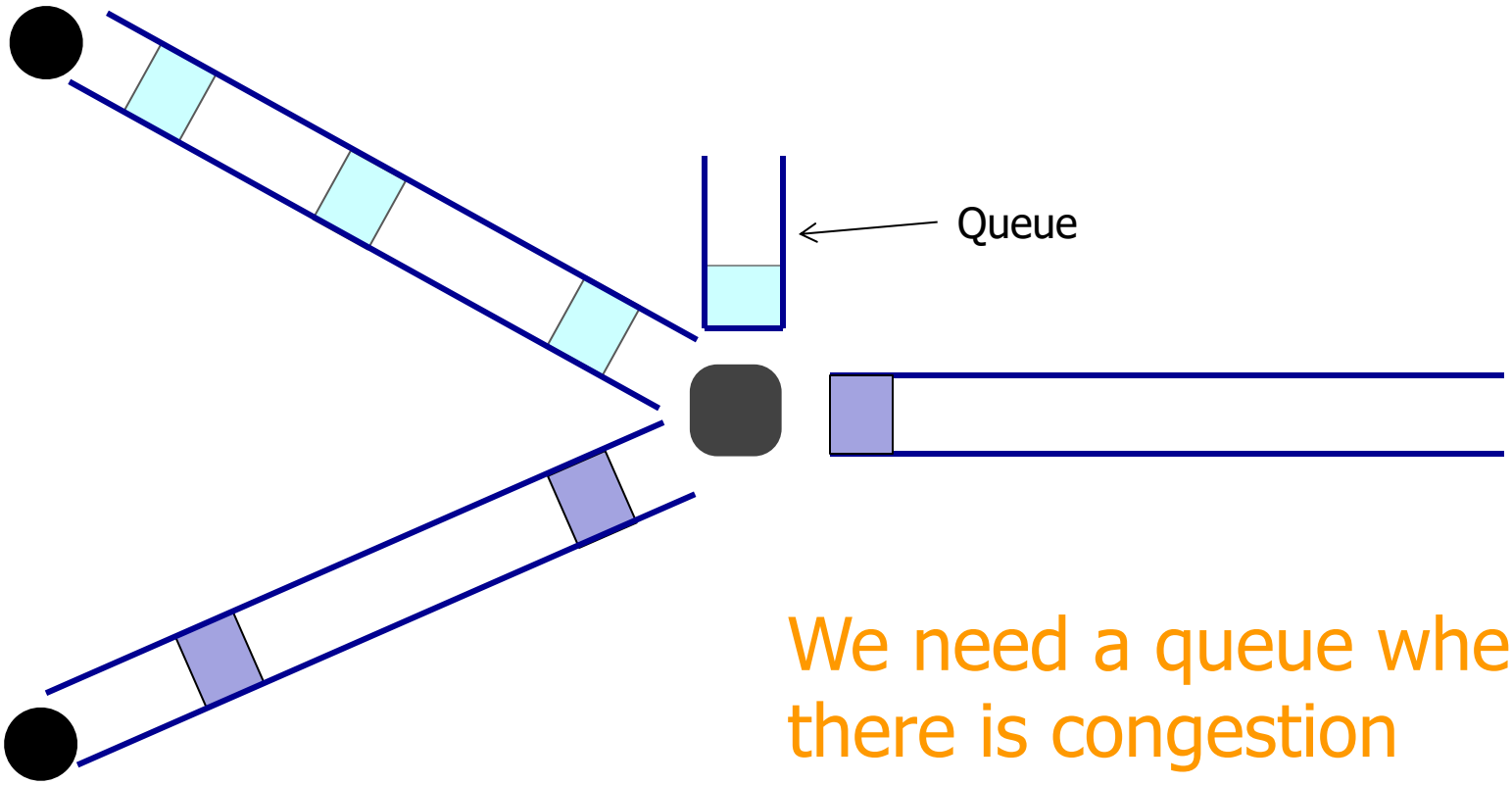


Why do messages need to wait?



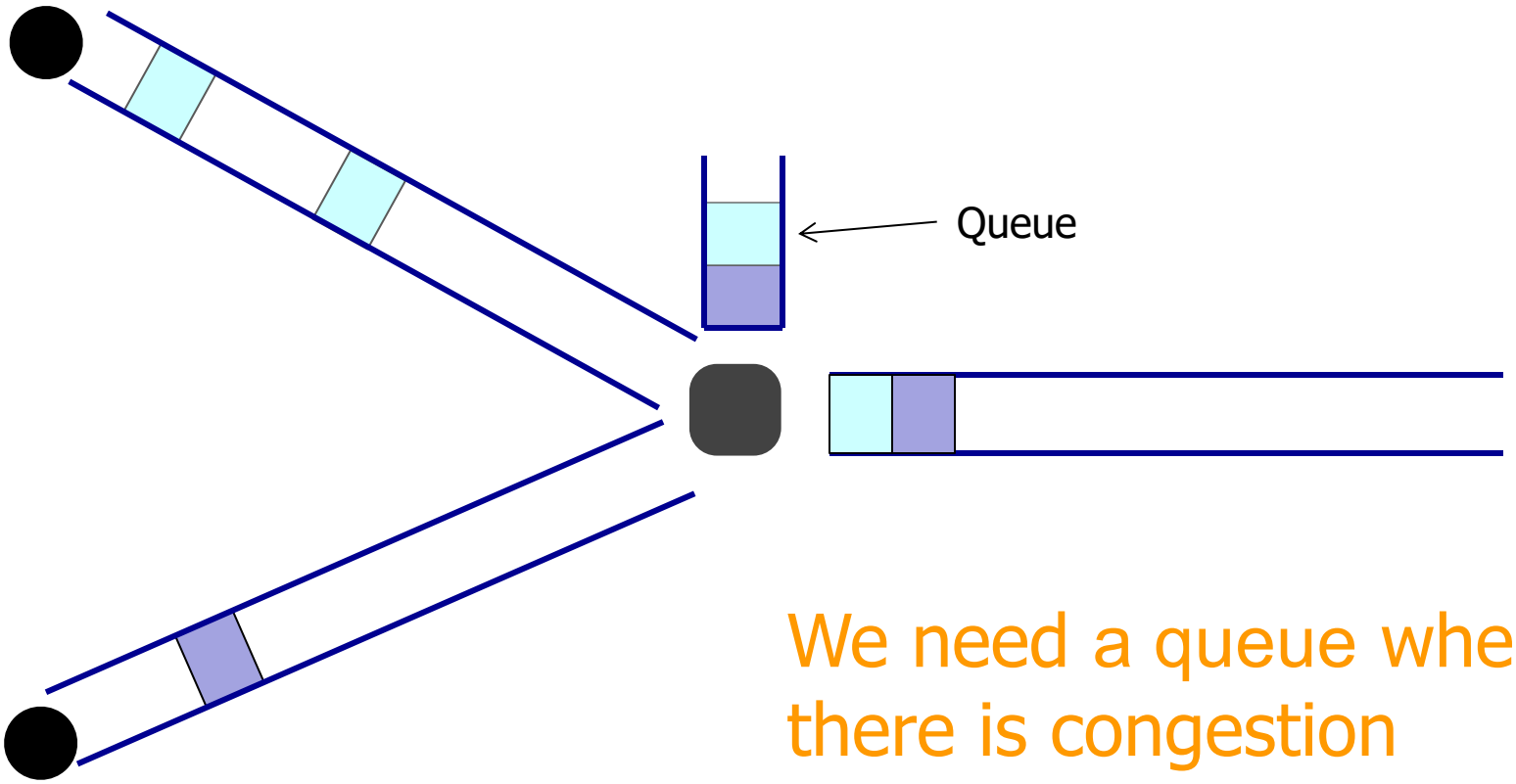
We need a queue when there is congestion

Why do messages need to wait?



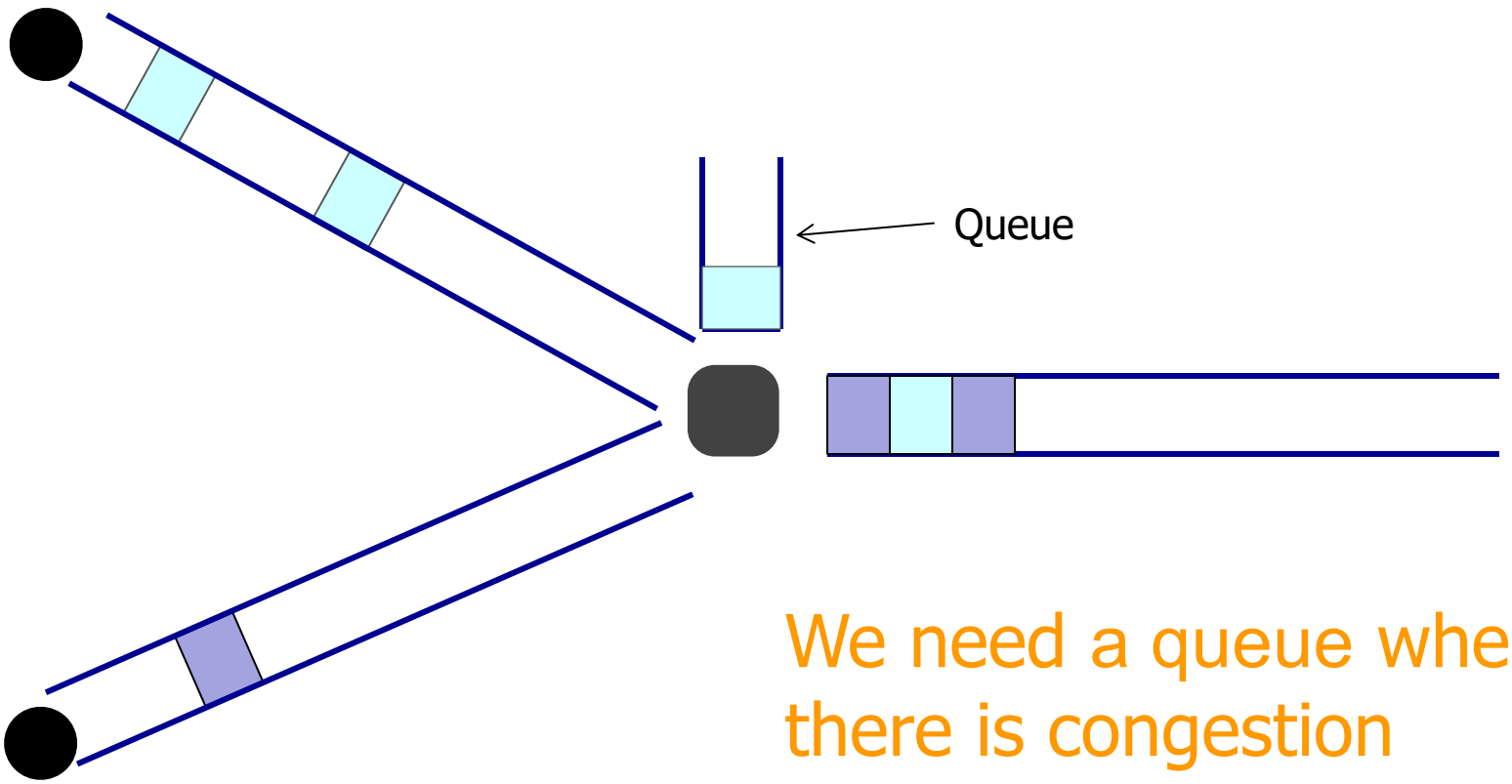
We need a queue when there is congestion

Why do messages need to wait?



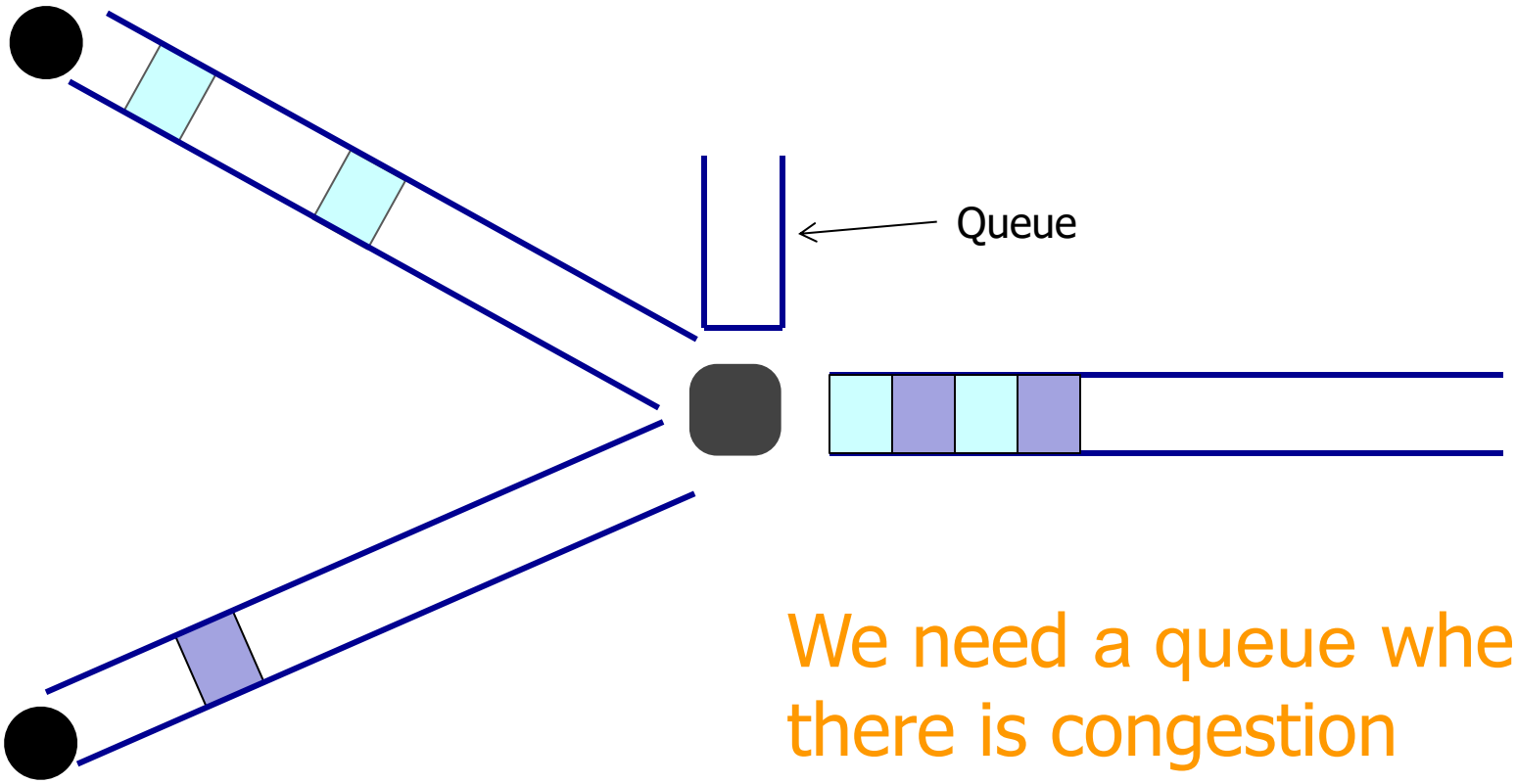
We need a queue when there is congestion

Why do messages need to wait?



We need a queue when there is congestion

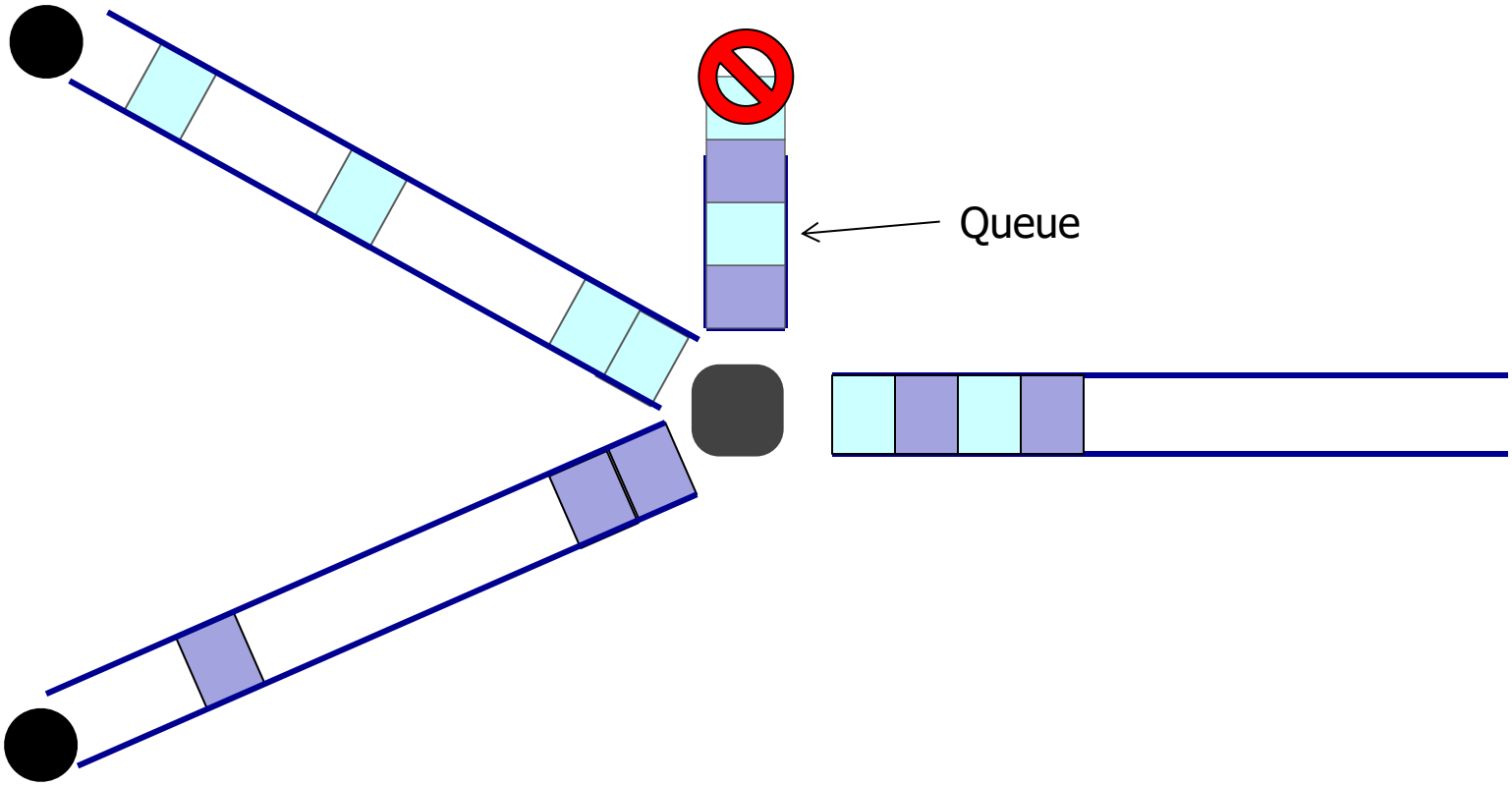
Why do messages need to wait?



We need a queue when there is congestion

Not a rare event!

Persistent congestion => packet drops



Queuing delay

- How long a message needs to wait before being processed
- Depends on traffic pattern
  - Arrival rate at the queue
  - Nature of arriving traffic (bursty or not?)
  - Bandwidth of outgoing link

Little's Law (1961)

- L := number of messages waiting in the queue
- A := rate at which messages arrive
- W := time messages wait in the queue

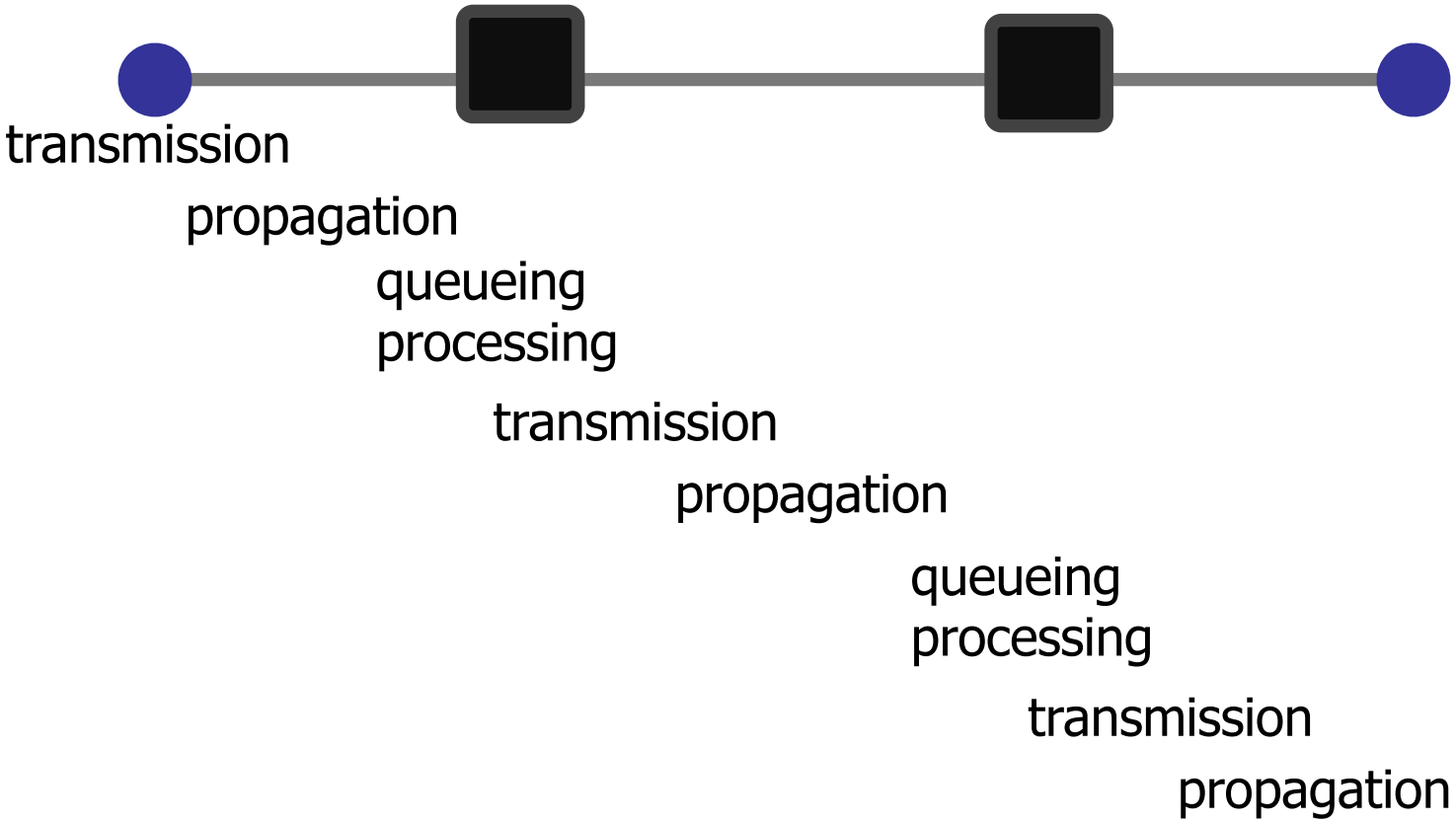
- L = A x W

- Can compute any variable given the other two
  - How big does my buffer need to be?
  - How fast do I need to handle packets to not overwhelm the buffer?

4. Processing Delay

- How long does the switch take to process a packet?
  - Negligible (100s of nanoseconds)

End-to-end delay



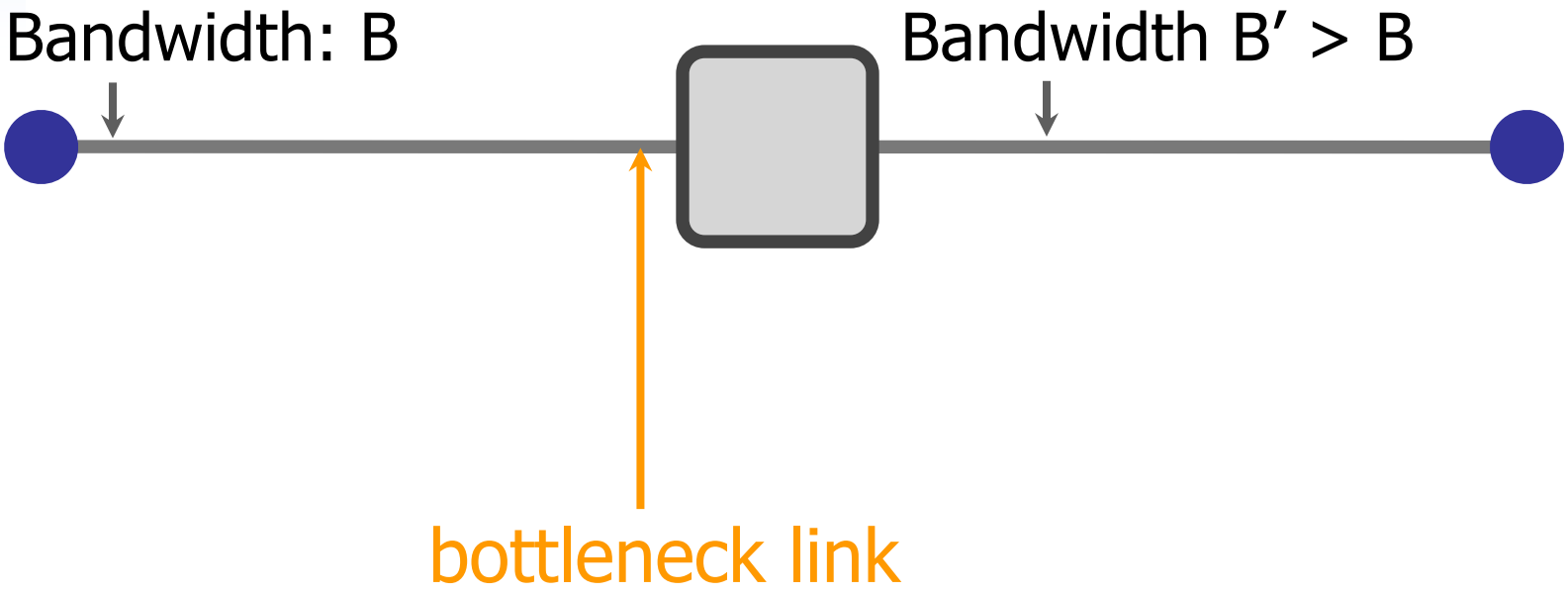
Performance Metric: Throughput

- At what rate is the destination receiving data from the source

Throughput

Bandwidth: B bits/sec

End-to-end throughput



Average throughput = min{B, B′} = B

Bandwidth-Delay Product

- Messages take space on the wire!

- The amount of data in flight is:
  - bandwidth-delay product (BDP)

    $$BDP = B \times D$$

  - Measure in bits, or in messages
  - Small for LANs, big for "long fat" pipes

BDP Examples

- Same city over a slow link:
  - Bandwidth: ~100Mbps
  - Propagation delay: ~0.1msec
  - BDP: 10,000bits (1.25KBytes)

- Cross-country over fast link:
  - Bandwidth: ~10Gbps
  - Propagation delay: ~10msec
  - BDP: 10^8bits (12.5MBytes)

Performance Metric: Loss

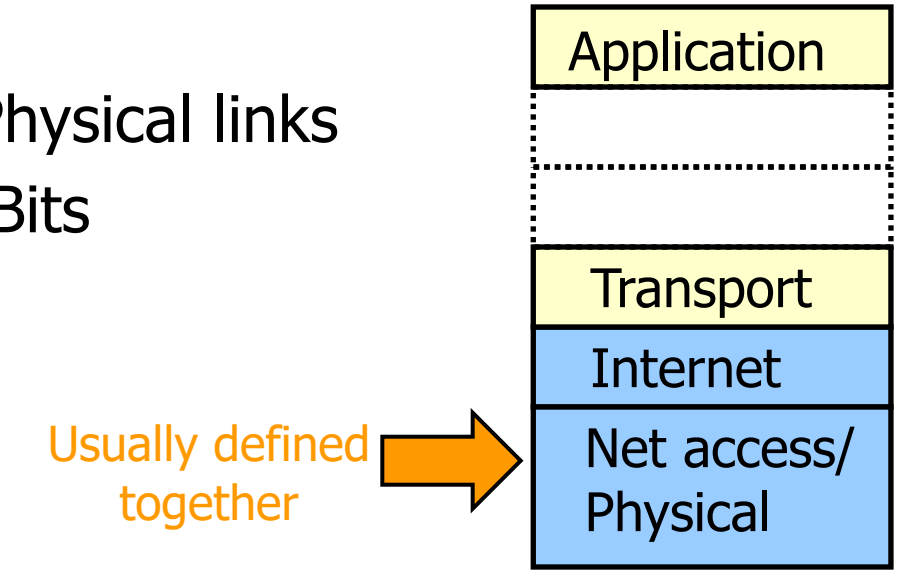
- What fraction of the packets are dropped?
- Bits are corrupted for a variety of reasons
  - Wired: Engineer link for the worst case
  - Wireless: Can't design for worst case, need to adapt data rate

Review: Physical layer

- Connects: Transmitters and receivers

- Upper interface: Stream of bits
- Lower interface: N/A (photons/electrons?)

- Name of network: Physical links
- Name of message: Bits



Agenda

- Physical Layer 
  - Media types 
  - Performance Metrics 
- Link Layer 
  - Frame format and framing
  - Error detection
  - Medium access control
  - Switching

Layer 2: Data link layer

- Connects: Physical interfaces
  - Possibly multiple
  - Usually all the same physical layer protocol
- Upper interface: A packet of data
- Lower interace: Stream of bits
- Name of network: Local Area Network (LAN)
- Name of message: Frame

Layer 2: Data link layer

- Connects: physical interfaces
  - Possibly multiple
  - Usually all the same physical layer protocol
- Upper interface: A packet of data
- Lower interace: Stream of bits
- Name of network: Local Area Network (LAN)
- Name of message: Frame

Data link layer

Provides four primary services

- Framing
  - Convert a stream of bits into messages
- Error detection and correction
  - Ensure that extracted frames haven't been corrupted
- Link access
  - Medium access control (MAC) defines when to transmit
- Reliable delivery (sometimes)

Ethernet "Frames"

- A message in Ethernet

- Preamble: 7 bytes for clock synchronization and 1 byte to indicate start of frame
- Addresses: 6 bytes
- Type: 2 bytes, higher-layer protocol (e.g., IP)
- Data payload: max 1500 bytes, min 46 bytes
- CRC: 4 bytes for error detection

Medium Access Control Address

- MAC address (e.g., 00:15:C5:49:04:A9)
  - Numerical address used within a link
  - Unique, hard-coded in the adapter when it is built
  - Flat name space of 48 bits
- Hierarchical allocation: Global uniqueness!
  - Blocks: assigned to vendors (e.g., Dell) by the IEEE
  - Adapters: assigned by the vendor from its block
- Broadcast address (i.e., FF:FF:FF:FF:FF:FF)
  - Send the frame to *all* adapters

As an aside: Promiscuous mode

- Normal adapter: receives frames sent to
  - The local MAC address
  - Broadcast address FF:FF:FF:FF:FF:FF
- Promiscuous mode
  - Receive *everything*, independent of destination MAC
- Useful for packet sniffing
  - Network monitoring
  - E.g., wireshark, tcpdump