Basic Concepts

CS 438: Spring 2014

Matthew Caesar

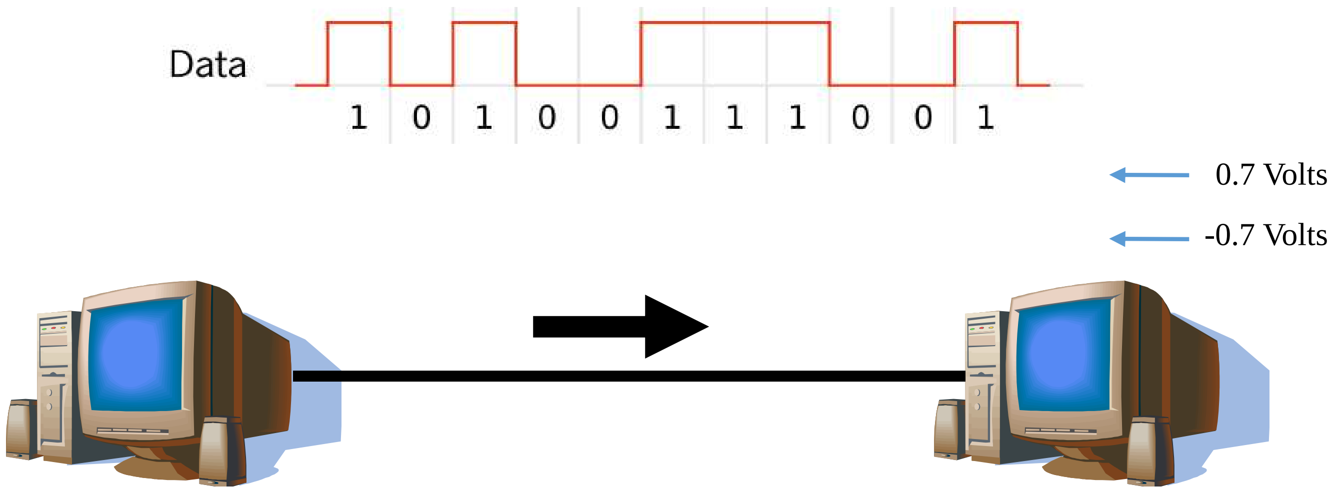
http://www.cs.illinois.edu/~caesar/cs438

Today

- 10,000 foot view of the Internet
- Basic concepts
  - links
  - packet delays
  - circuit switching
  - packet (datagram) switching
- Layering

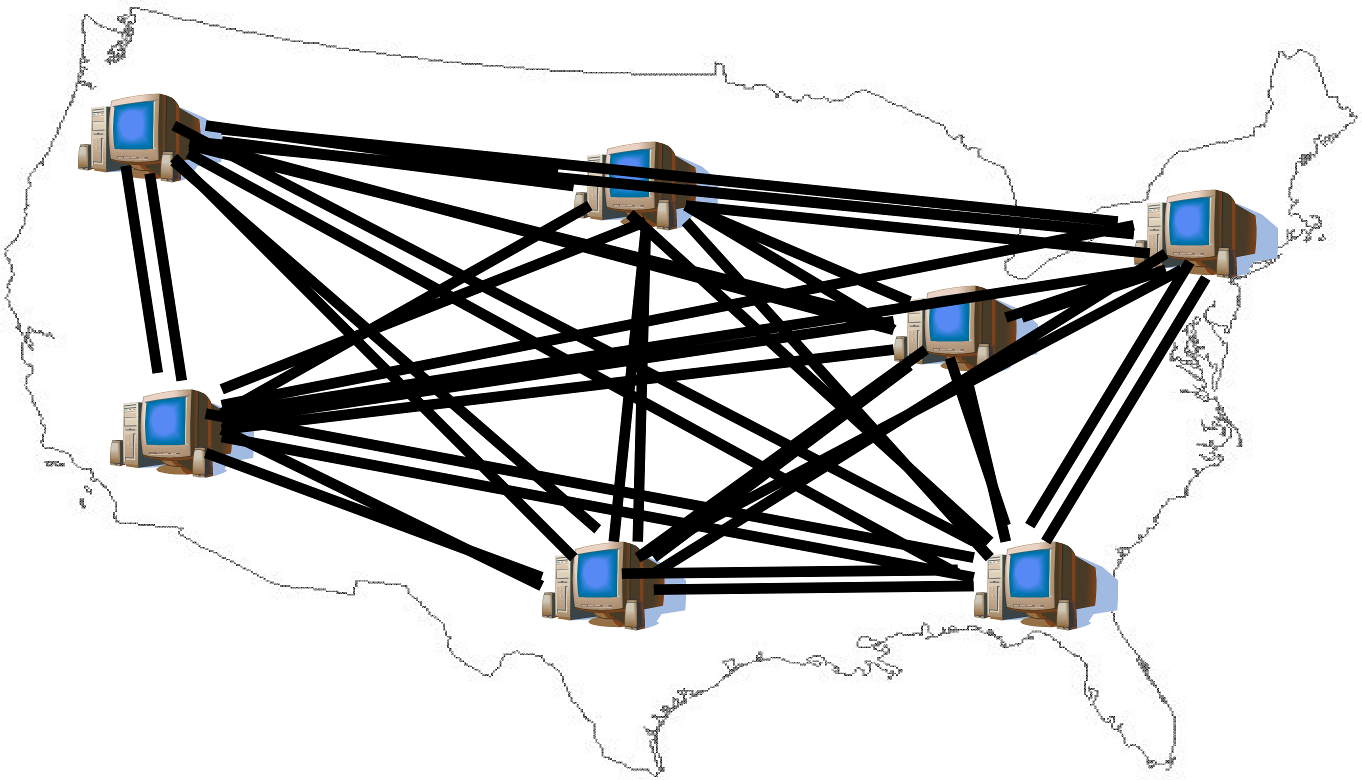
A 10,000 foot view of the Internet

How can Two Hosts Communicate?



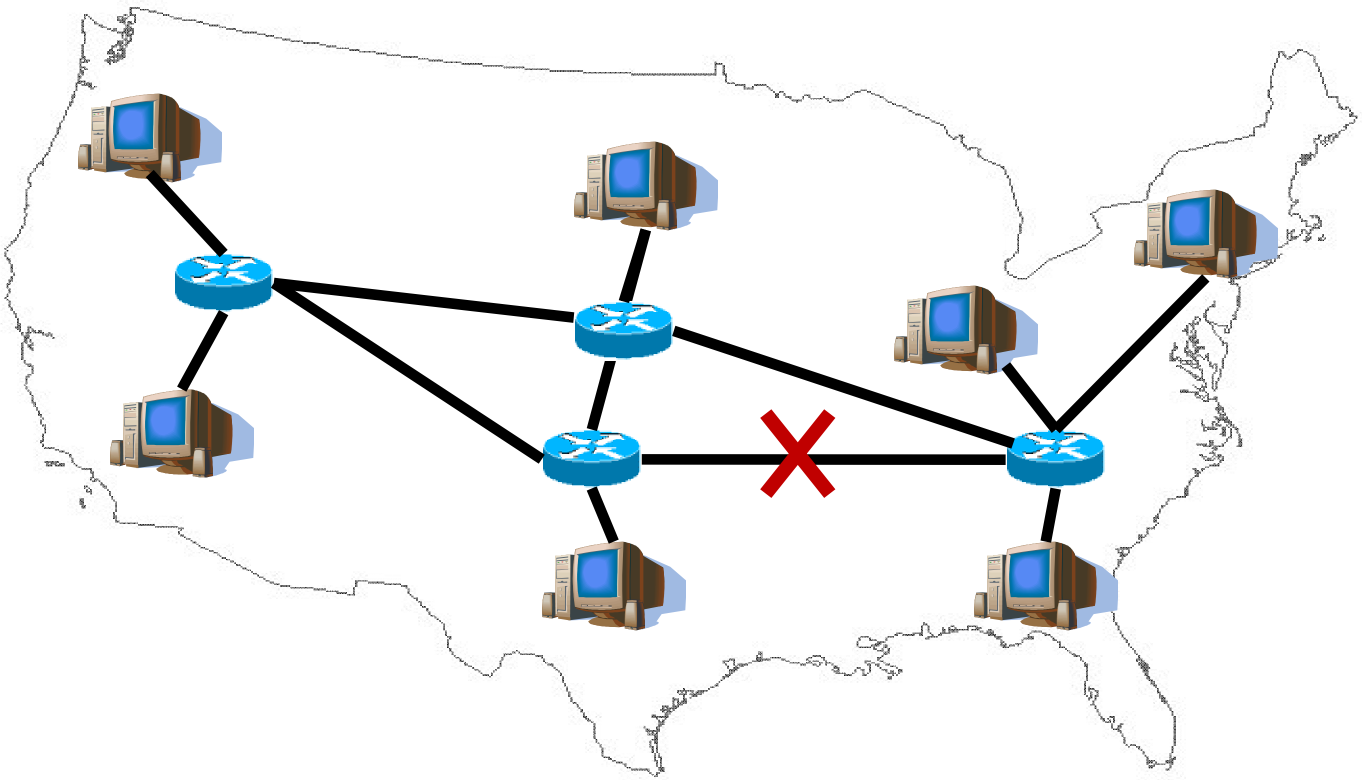
- Encode information on modulated “Carrier signal”
    - Phase, frequency, and amplitude modulation, and combinations thereof
    - Ethernet: self-clocking Manchester coding ensures one transition per clock
    - Technologies: copper, optical, wireless

How can many hosts communicate?



- Naïve approach: full mesh
- Problem:
  - Obviously doesn't scale to the 570,937,778+ hosts in the Internet

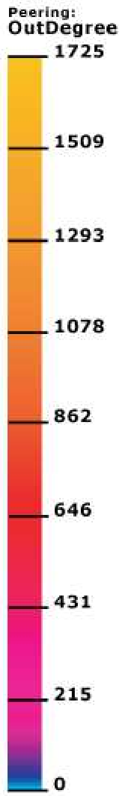
How can many hosts communicate?

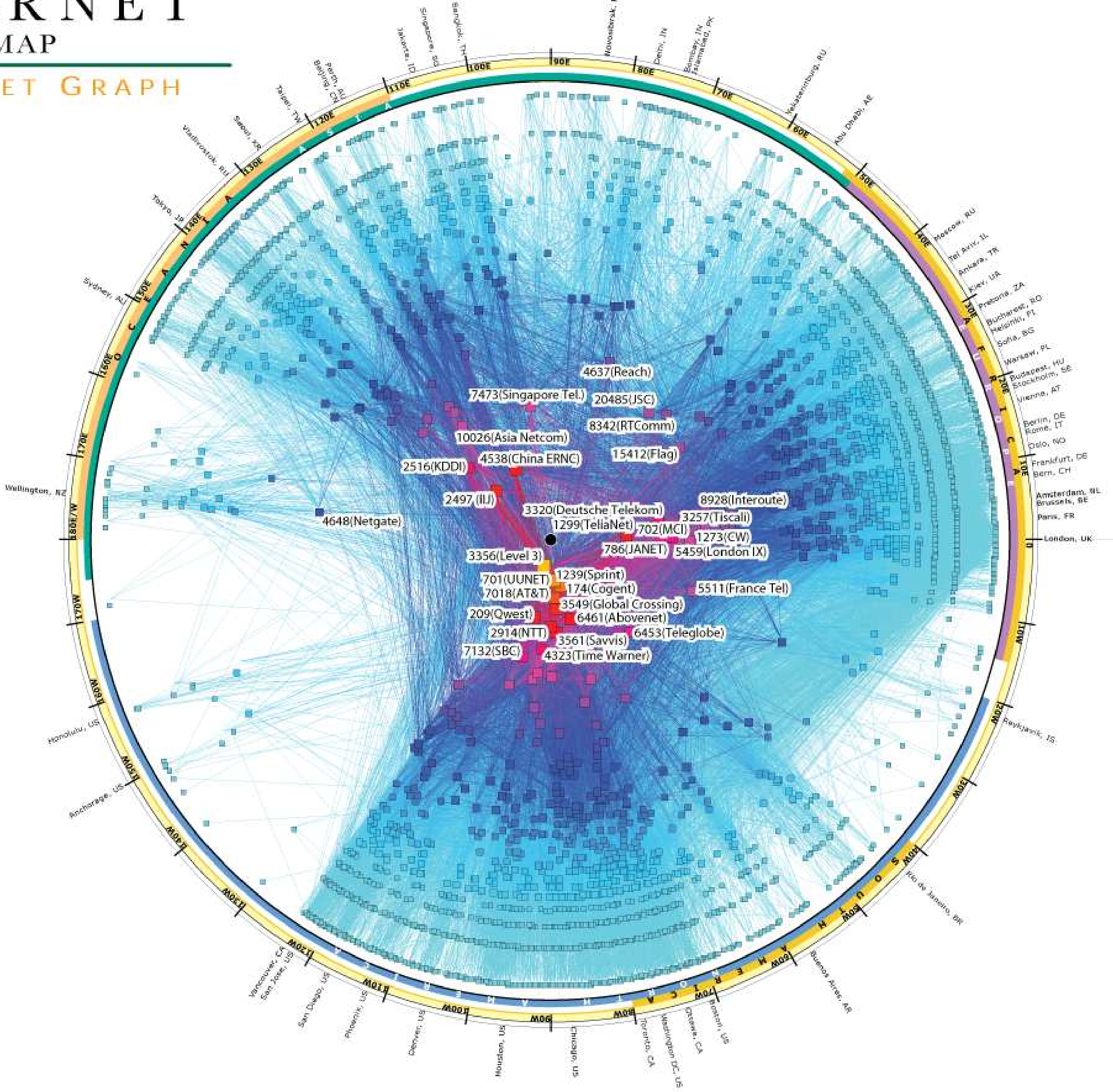


- Better approach: Multiplex traffic with routers
- Goals: make network robust to failures and attack, maintain spare capacity, reduce operational costs
    - Introduces new challenges: What topology to use? How to find and look up paths? How to identify destinations?

Complete Network Assets : XO Communications

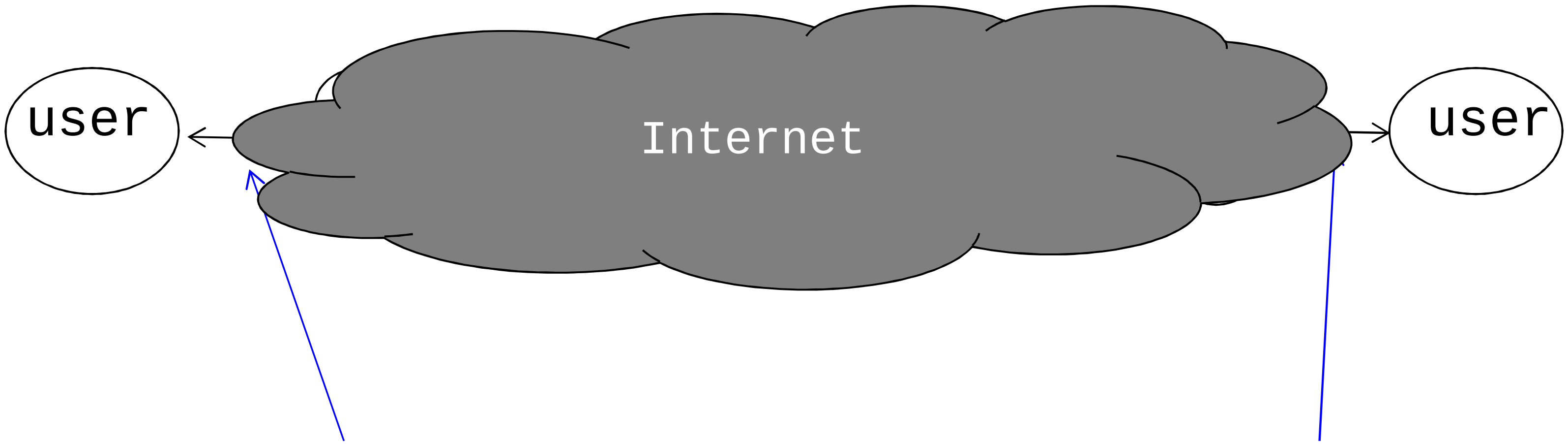
IPv4 INTERNET
TOPOLOGY MAP
AS-level INTERNET GRAPH





The Internet is a federated system

- The Internet ties together different networks
  - >20,000 ISP networks



Tied together by IP -- the "Internet Protocol" : a single common interface between users and the network and between networks

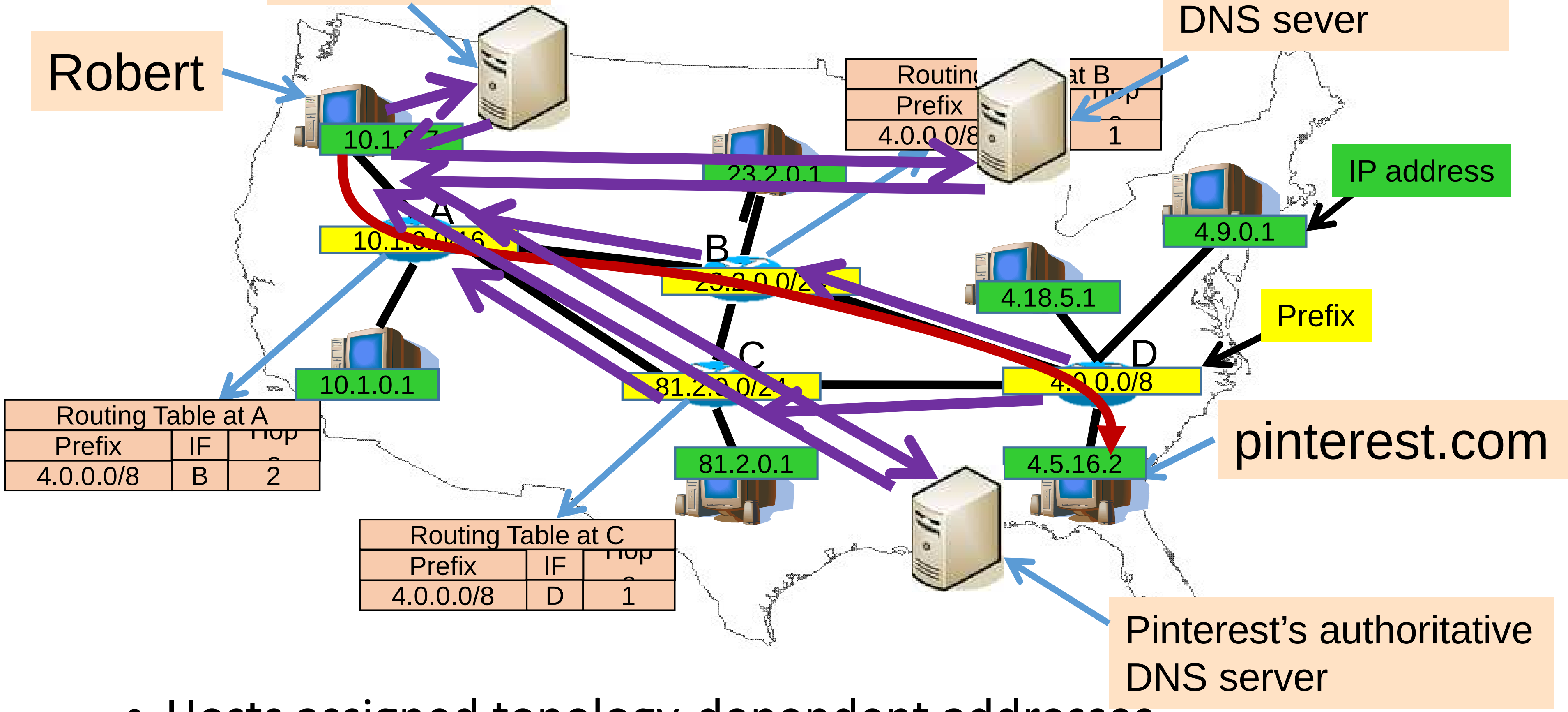
The Internet is a federated system

- The Internet ties together different networks
  - >20,000 ISP networks

- A single, common interface is great for interoperability…
- …but tricky for business

- Why does this matter?
  - ease of interoperability is the Internet's most important goal
  - practical realities of incentives, economics and real-world trust drive topology, route selection and service evolution

How uters find paths?

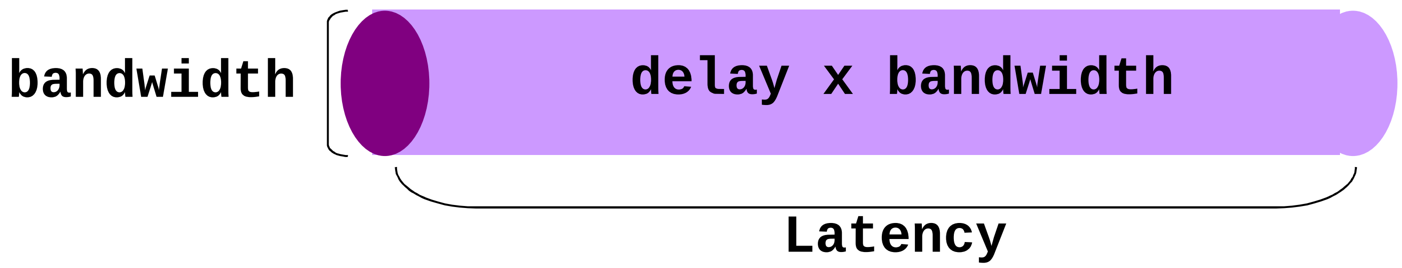


- Hosts assigned topology-dependent addresses
- Routers advertise address blocks ("prefixes")
- Routers compute "shortest" paths to prefixes
- Map IP addresses to names with DNS

Basic Concepts

Nodes and Links

Properties of Links



- Bandwidth (capacity): "width" of the link
  - number of bits sent (or received) per unit time (bits/sec or bps)
- Latency (delay): "length" of the link
  - propagation time for data to travel along the link (seconds)
- Bandwidth-Delay Product (BDP): "volume" of the link
  - amount of data that can be "in flight" at any time
  - propagation delay × bits/time = total bits in link

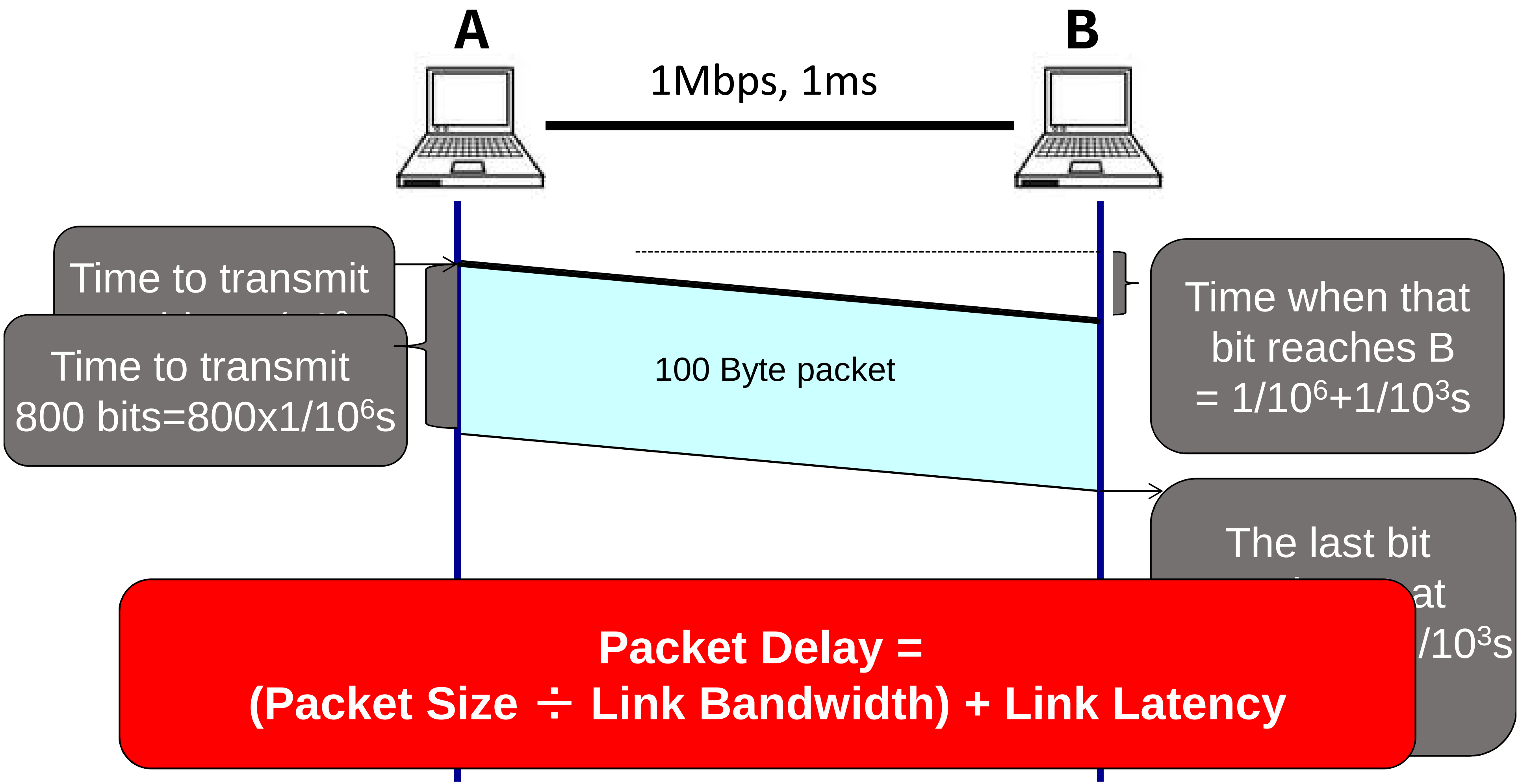
Examples of Bandwidth-Delay

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

- Cross-country over fast link:
  - BW ~10Gbps
  - Latency~10msec
  - BDP ~10^8bits (~12.5GBytes)

Packet Delay

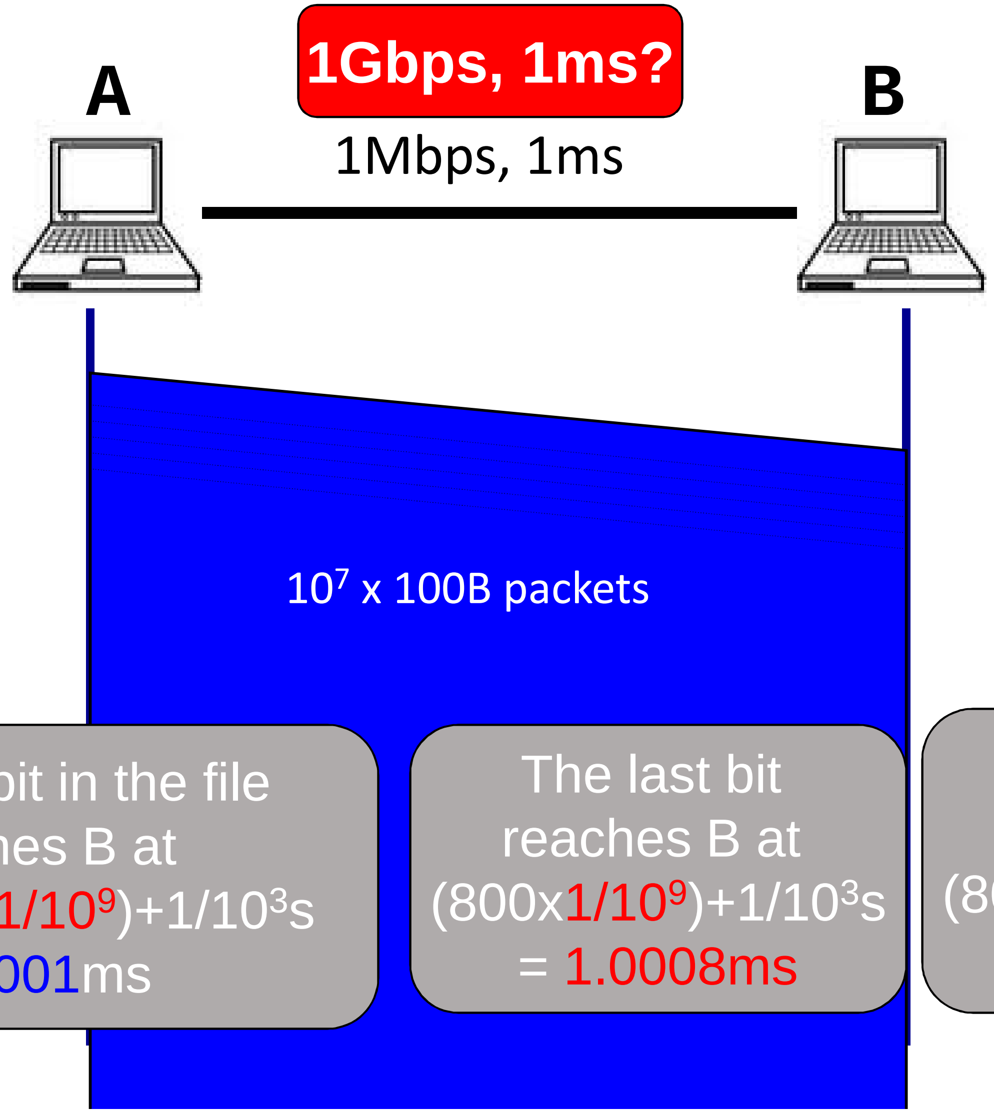
Sending a 100B packet from A to B?



Packet

1GB file in 100B packets

Sending a 100B packet from A to B?



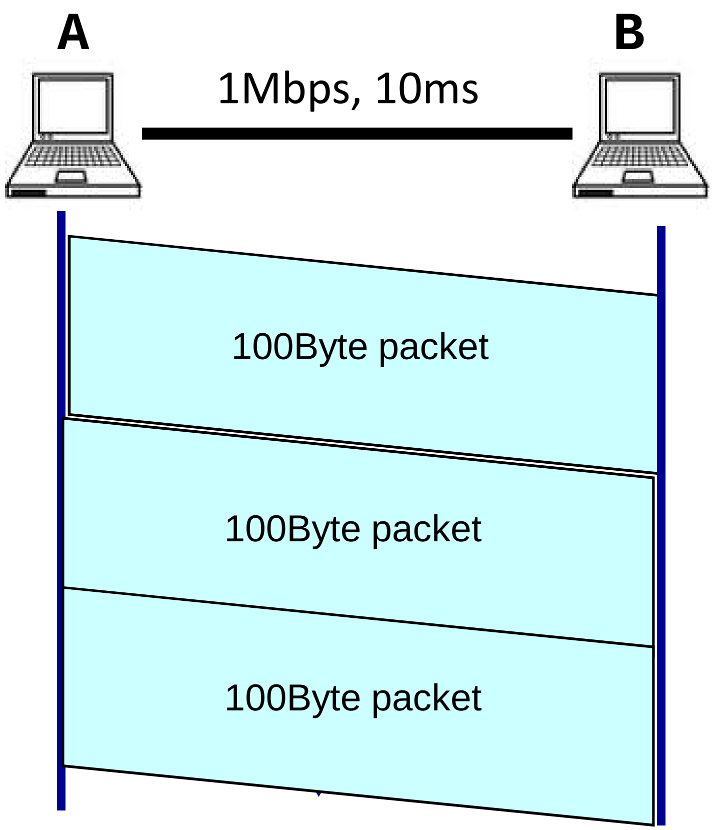
The last bit in the file reaches B at $(10^7 \times 800 \times 1/10^9) + 1/10^3$s = 8001ms

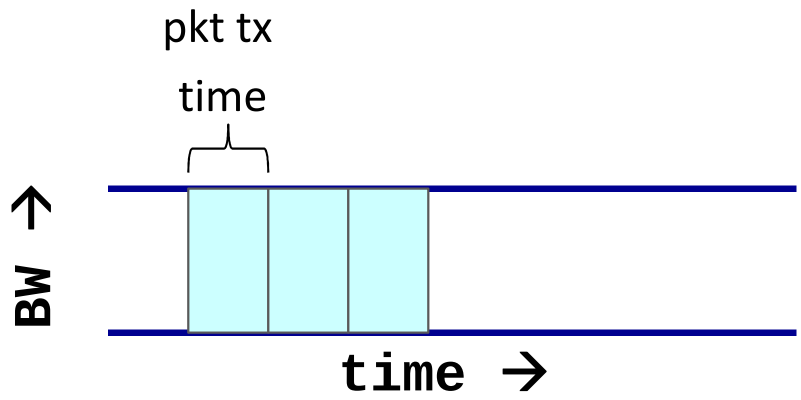
The last bit reaches B at $(800 \times 1/10^9) + 1/10^3$s = 1.0008ms

The last bit reaches B at $(800 \times 1/10^6) + 1/10^3$s = 1.8ms

Packet Delay: The "pipe" view

Sending 100B packets from A to B?

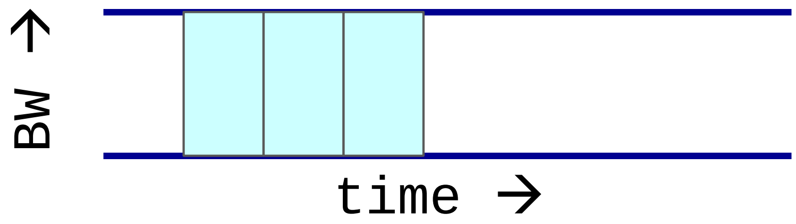




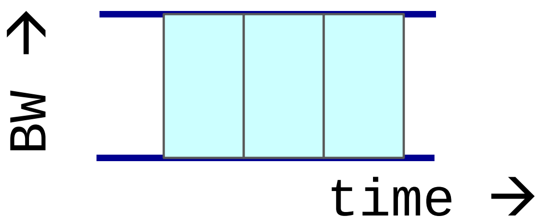
Packet Delay: The “pipe” view

Sending 100B packets from A to B?

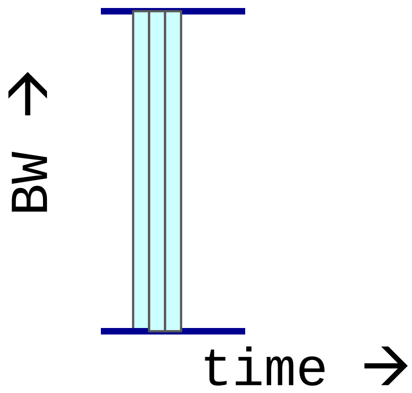
1Mbps, 10ms (BDP=10,000)



1Mbps, 5ms (BDP=5,000)



10Mbps, 1ms (BDP=10,000)

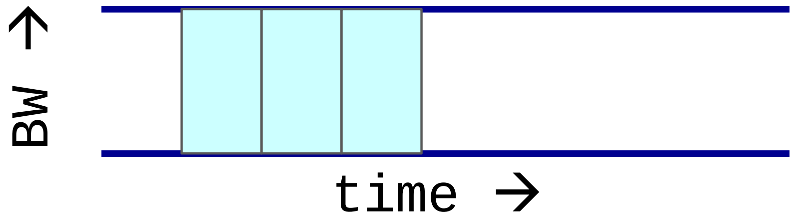


Packet Del … e “pipe” view

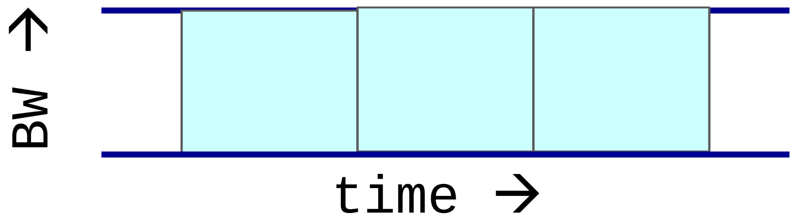
200B?

Sending 100B packets from A to B?

1Mbps, 10ms (BDP=10,000)



1Mbps, 10ms (BDP=10,000)

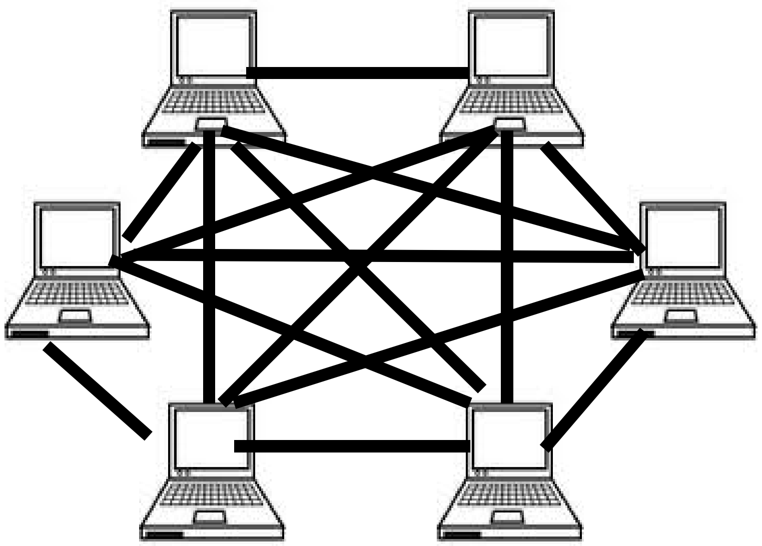


Any questions?

Nodes and Links

What if we have more nodes?

One link for every node?

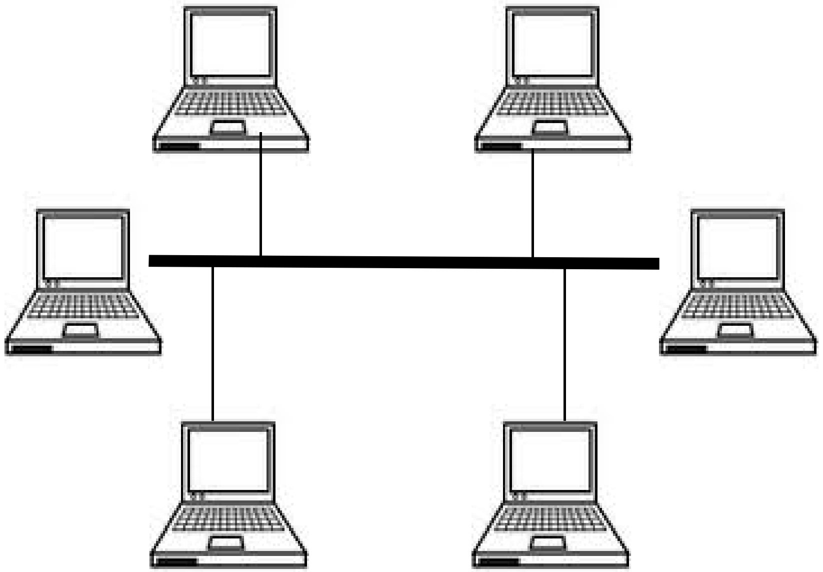


Need a <u>scalable</u> way to interconnect nodes

What if I have more nodes?

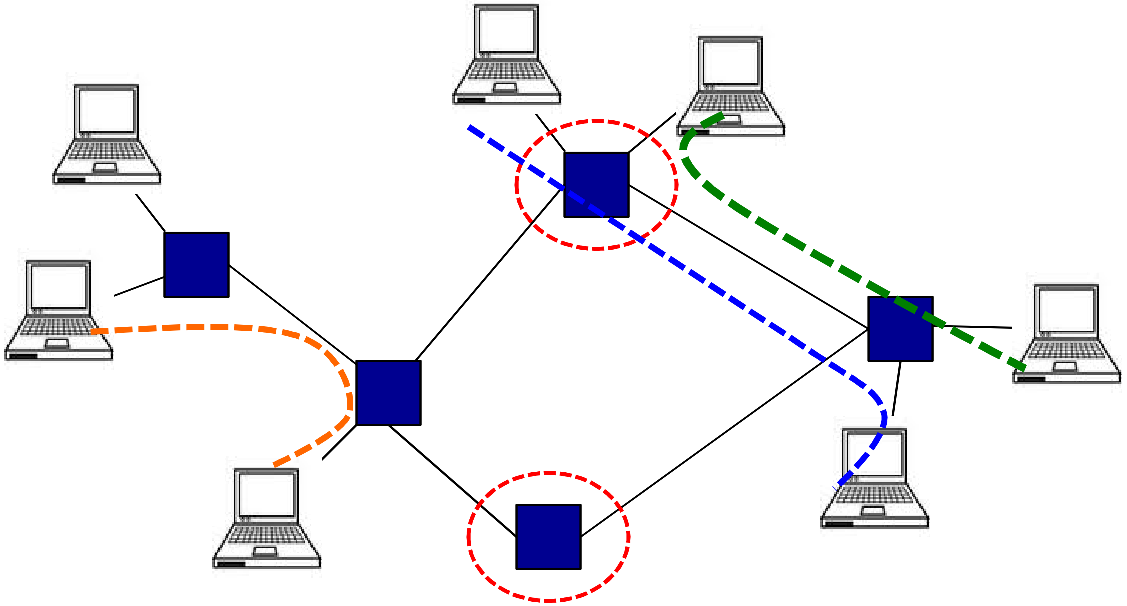
One link for all nodes?

This is a broadcast network



Solution: A switched network

Nodes share network link resources



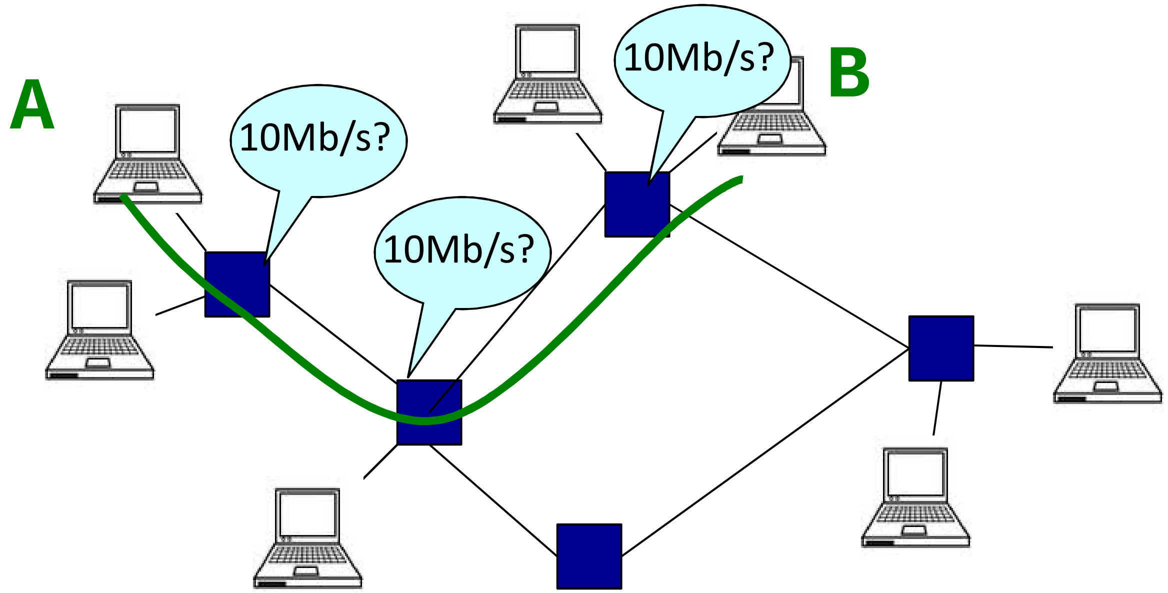
How is this sharing implemented?

Two forms of switched networks

- Circuit switching (used in the telephone network)
- Packet switching (used in the Internet)

Circuit Switching

Idea: source reserves network capacity along a path



(1) Node A sends a reservation request

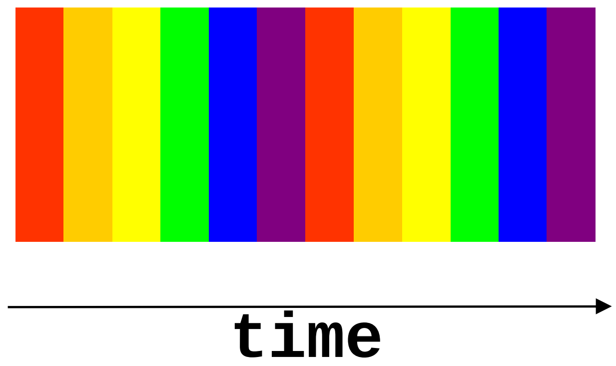
(2) Interior switches establish a connection -- i.e., “circuit”

(3) A starts sending data

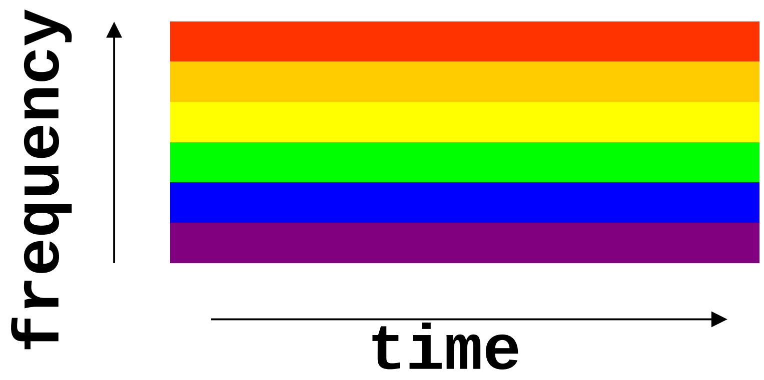
(4) A sends a “teardown circuit” message

Circuit Switching: sharing a link

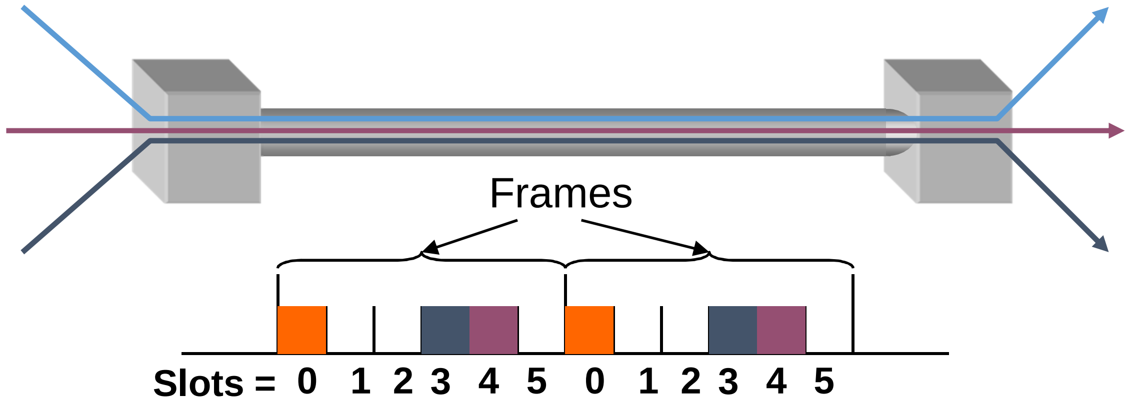
- Time-division
  - Each circuit allocated certain time slots



- Frequency-division
  - Each circuit allocated certain frequencies

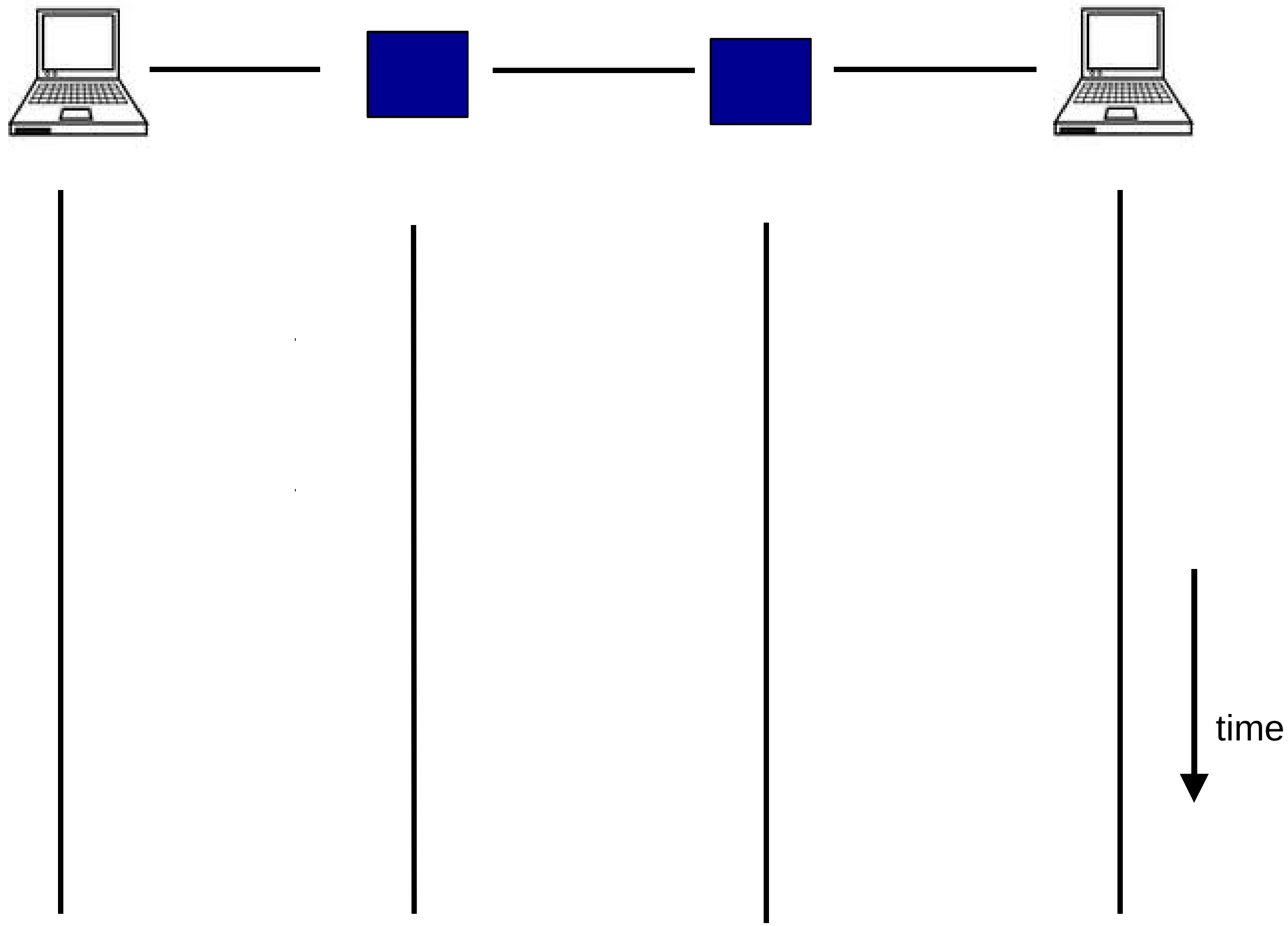


Time Division Multiplexing

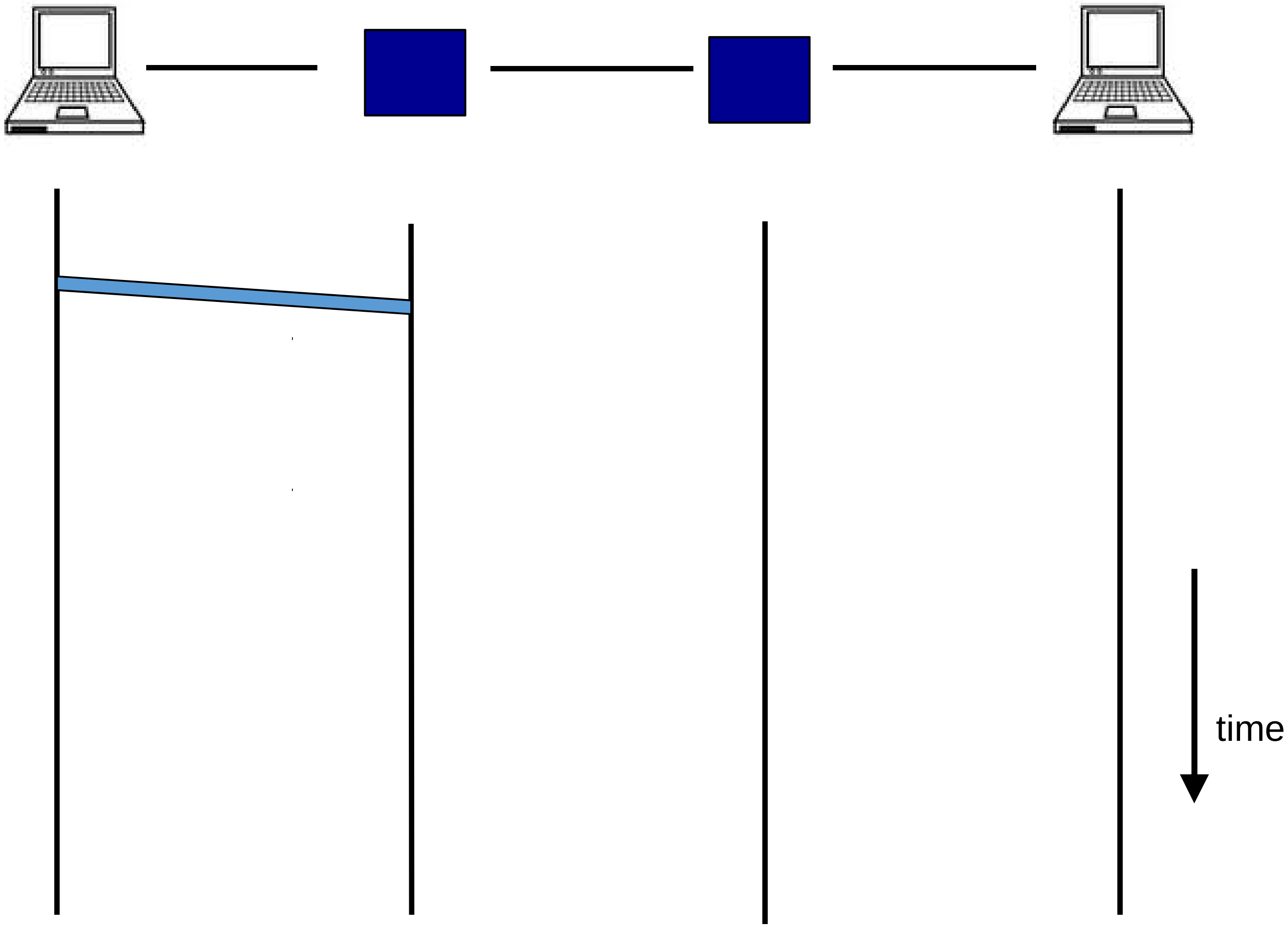


- Time divided into frames; frames into slots
- Relative slot position inside a frame determines to which conversation data belongs
  - e.g., slot 0 belongs to orange conversation
- Slots are reserved (released) during circuit setup (teardown)
- If a conversation does not use its circuit **capacity is lost!**

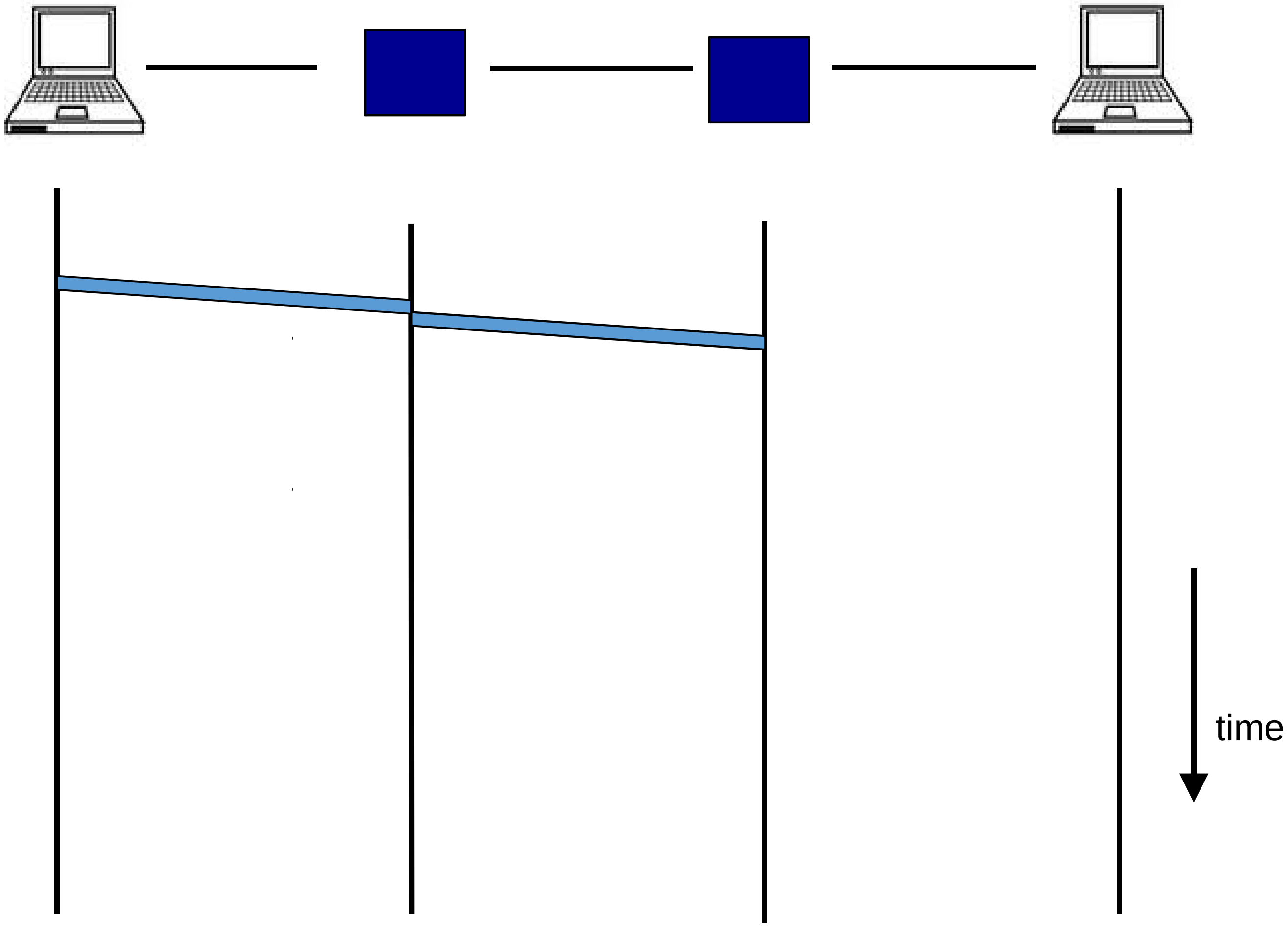
Timing in Circuit Switching



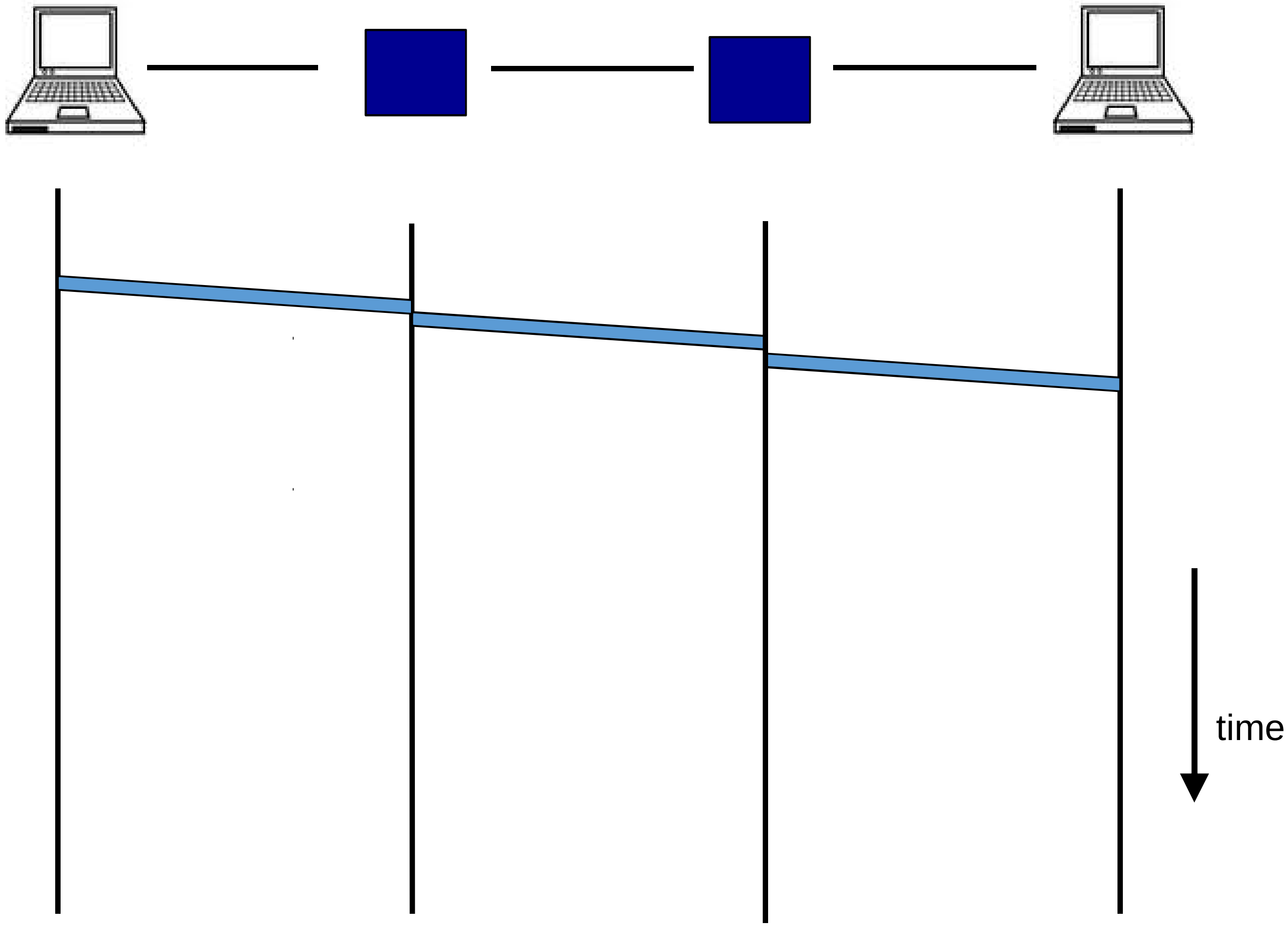
Timing in Circuit Switching



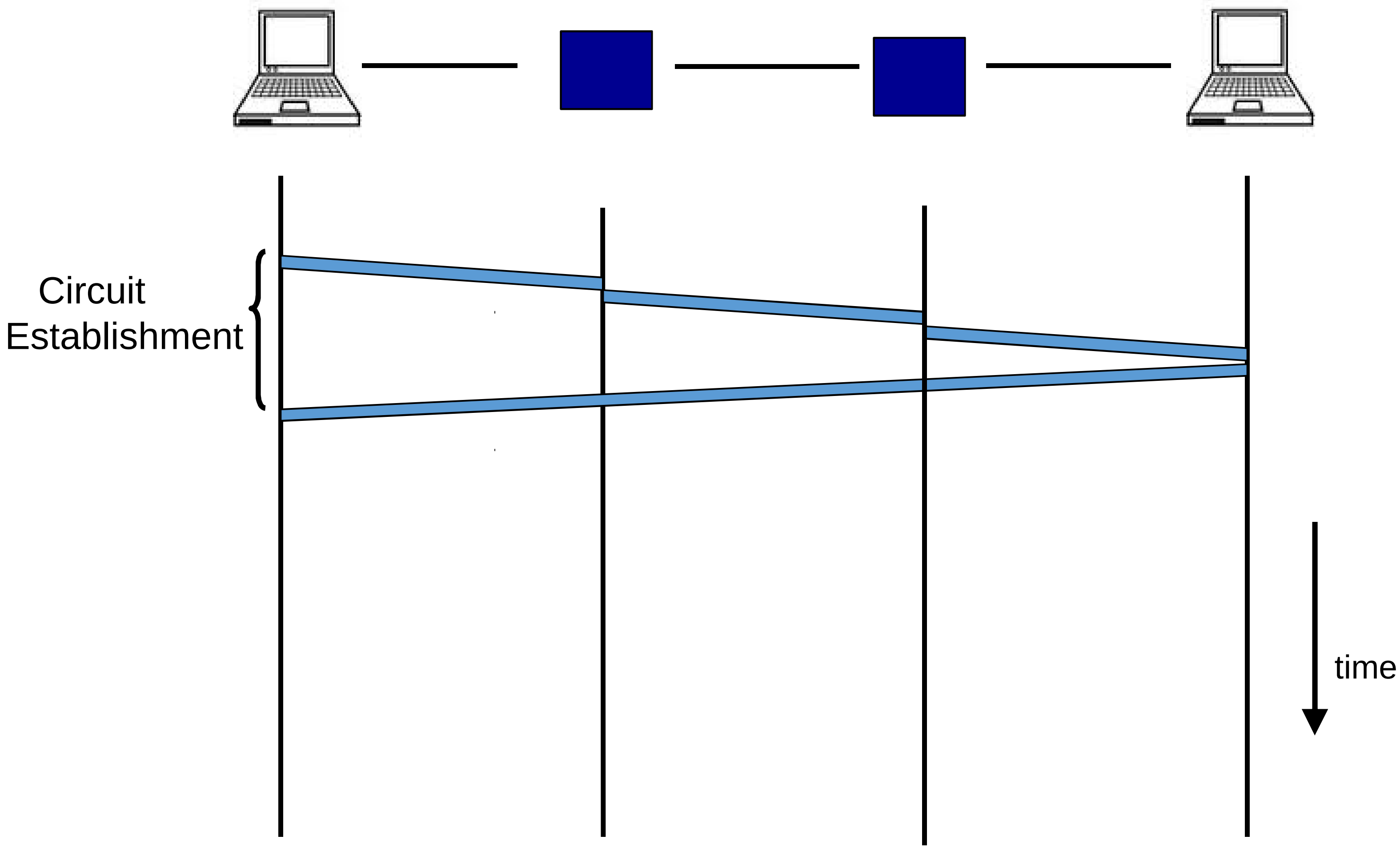
Timing in Circuit Switching



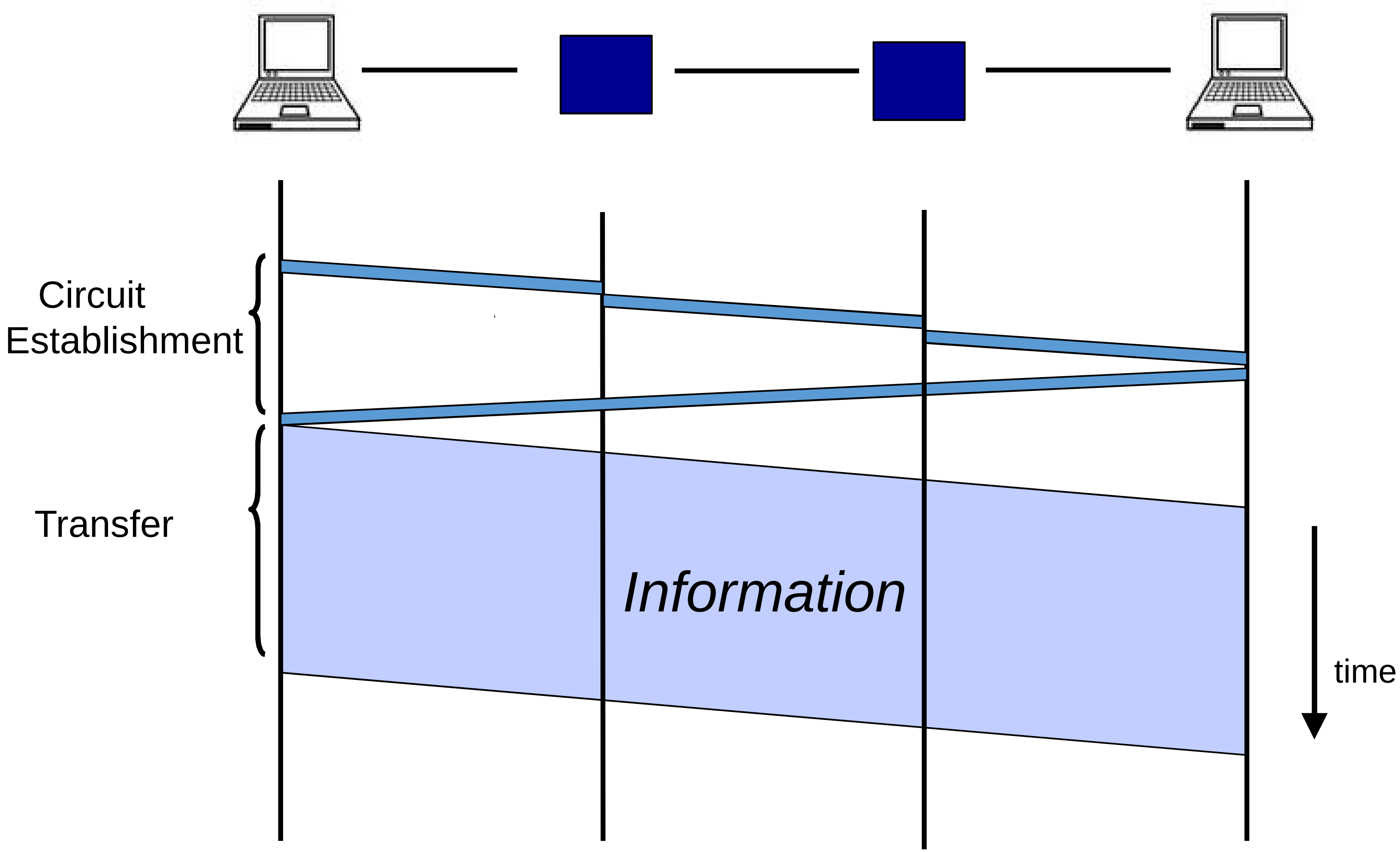
Timing in Circuit Switching



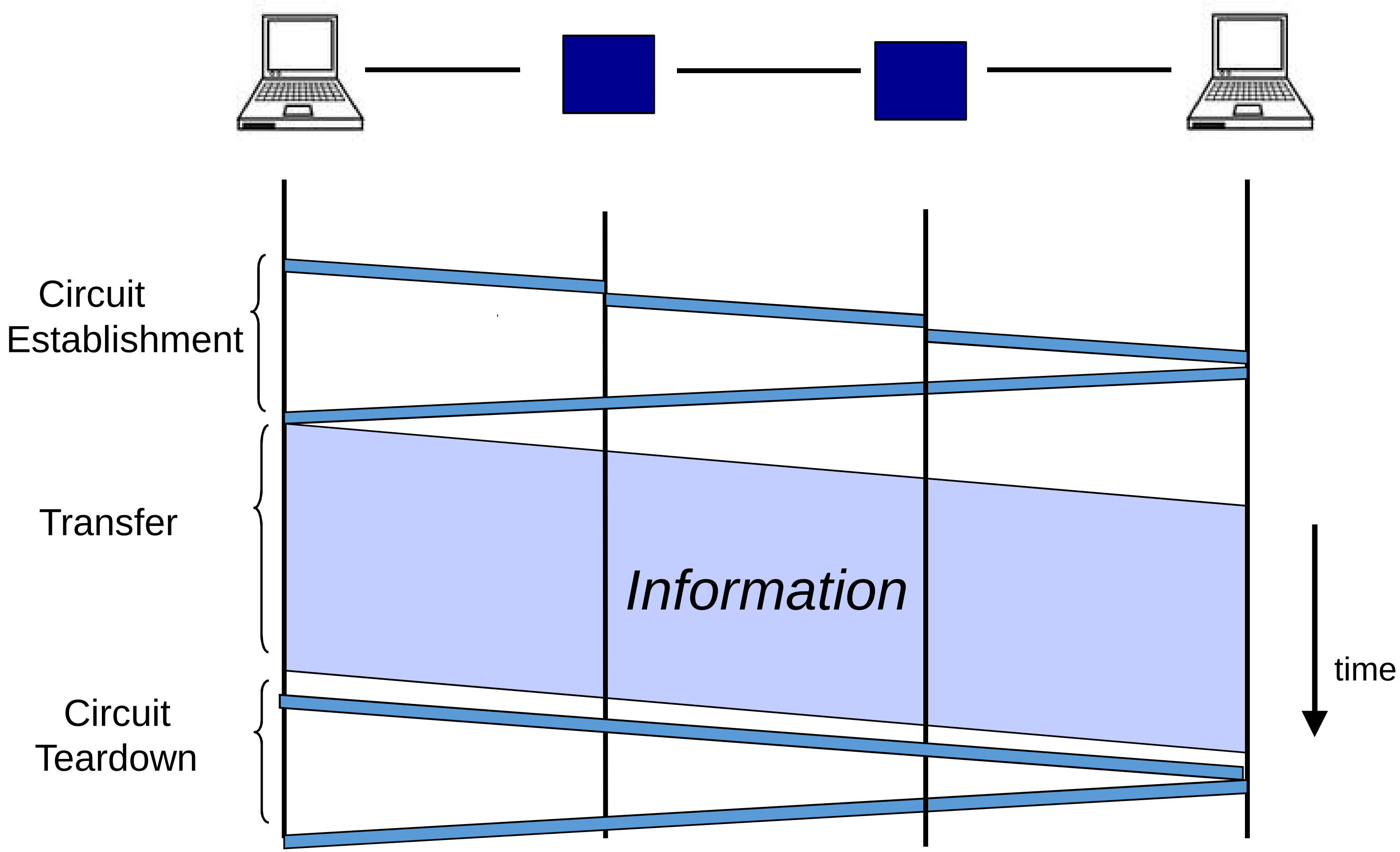
Timing in Circuit Switching



Timing in Circuit Switching



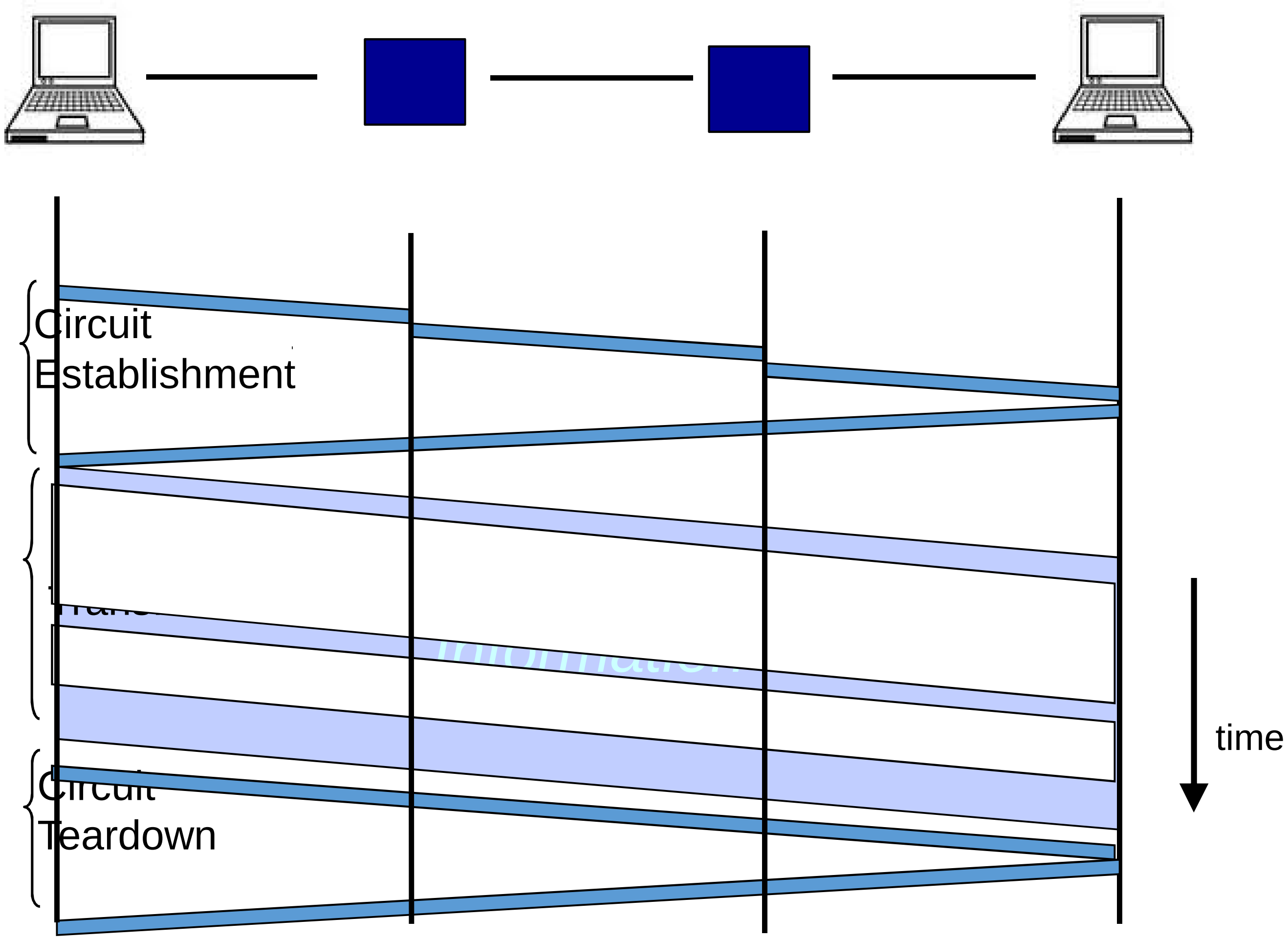
Timing in Circuit Switching



Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfer (once circuit is established)
- Cons

Timing in Circuit Switching

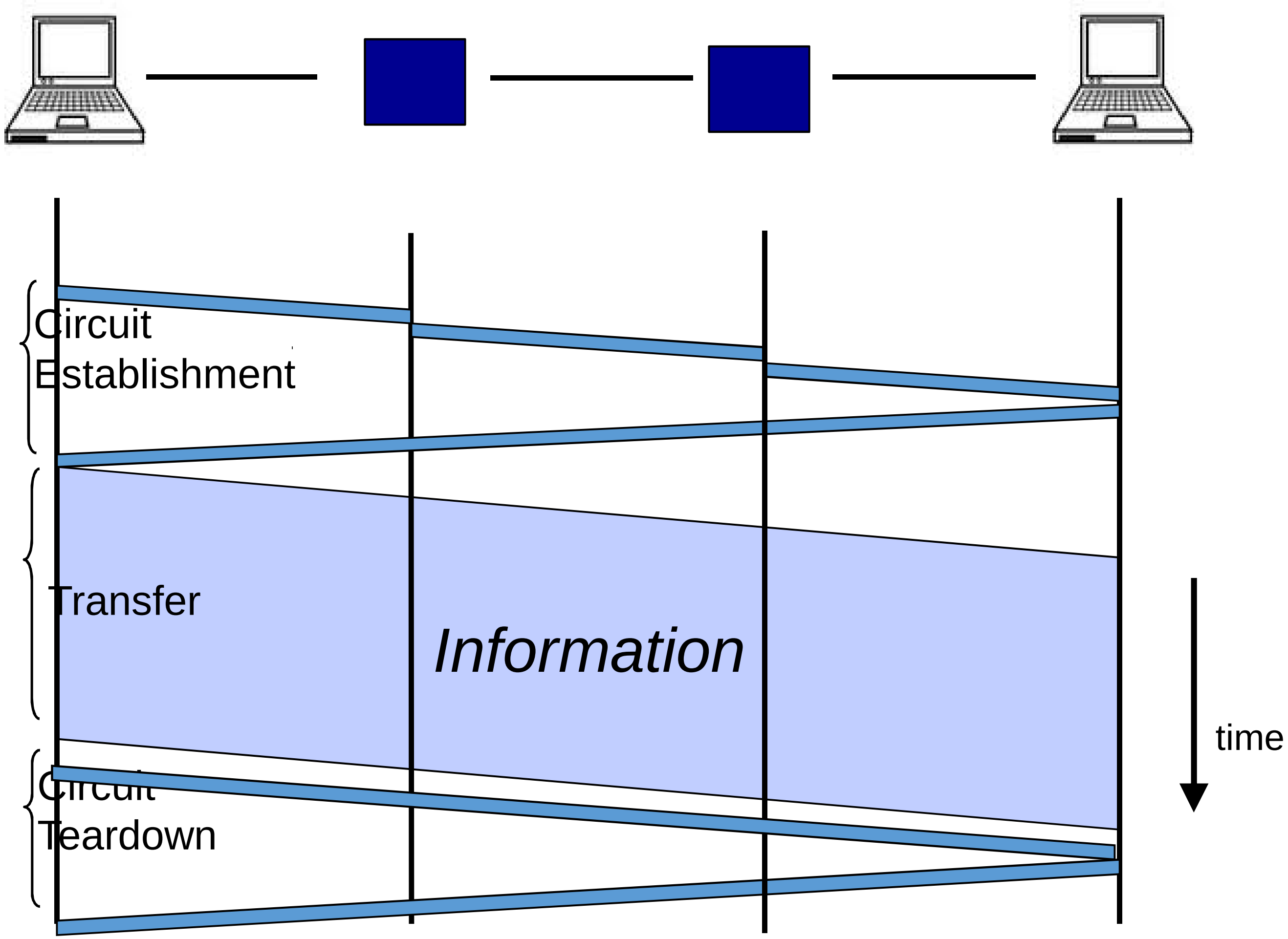


Circuit switching: pros and cons

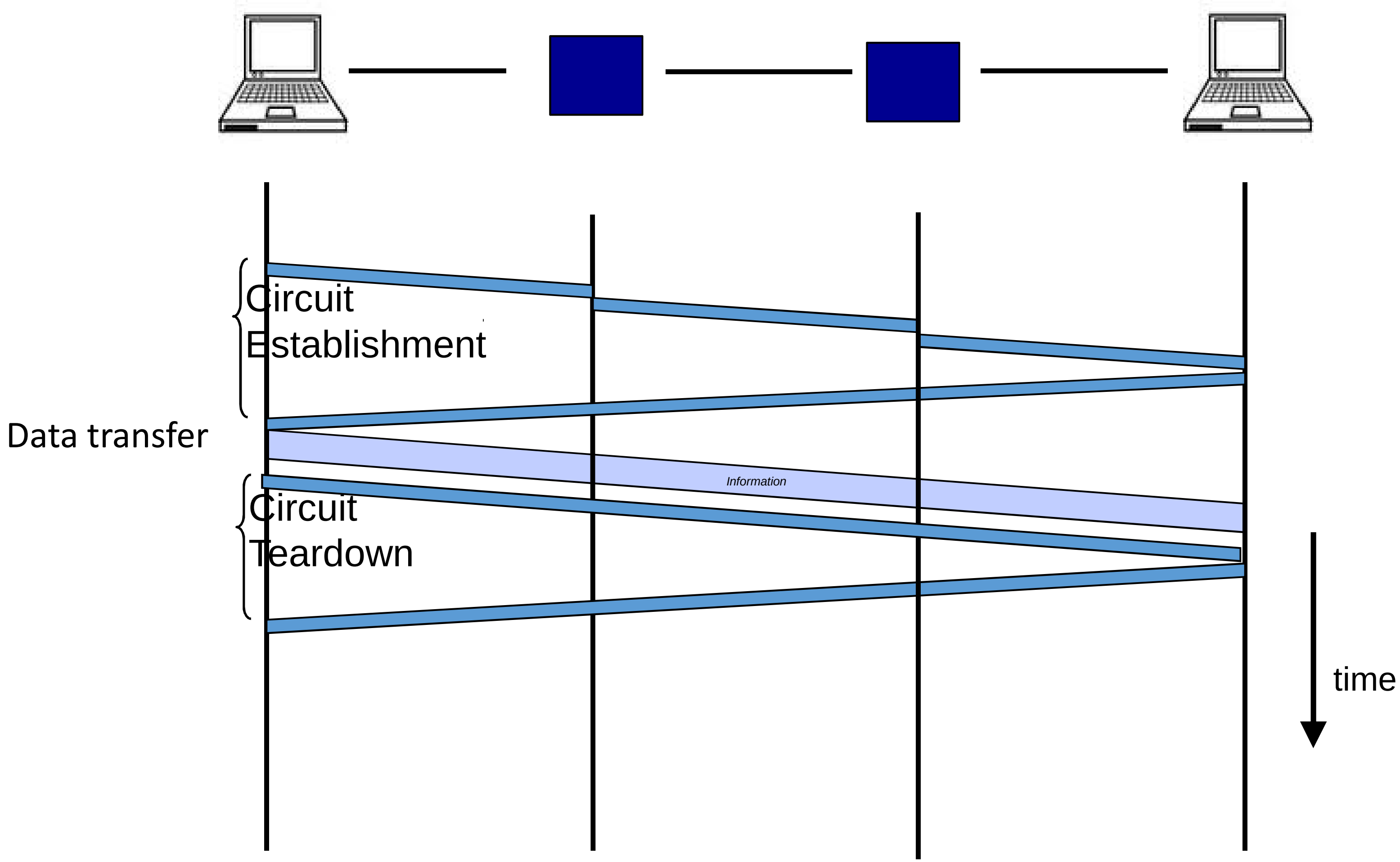
- Pros
  - guaranteed performance
  - fast transfer (once circuit is established)

- Cons
  - **wastes bandwidth if traffic is "bursty"**

Timing in Circuit Switching



Timing in Circuit Switching

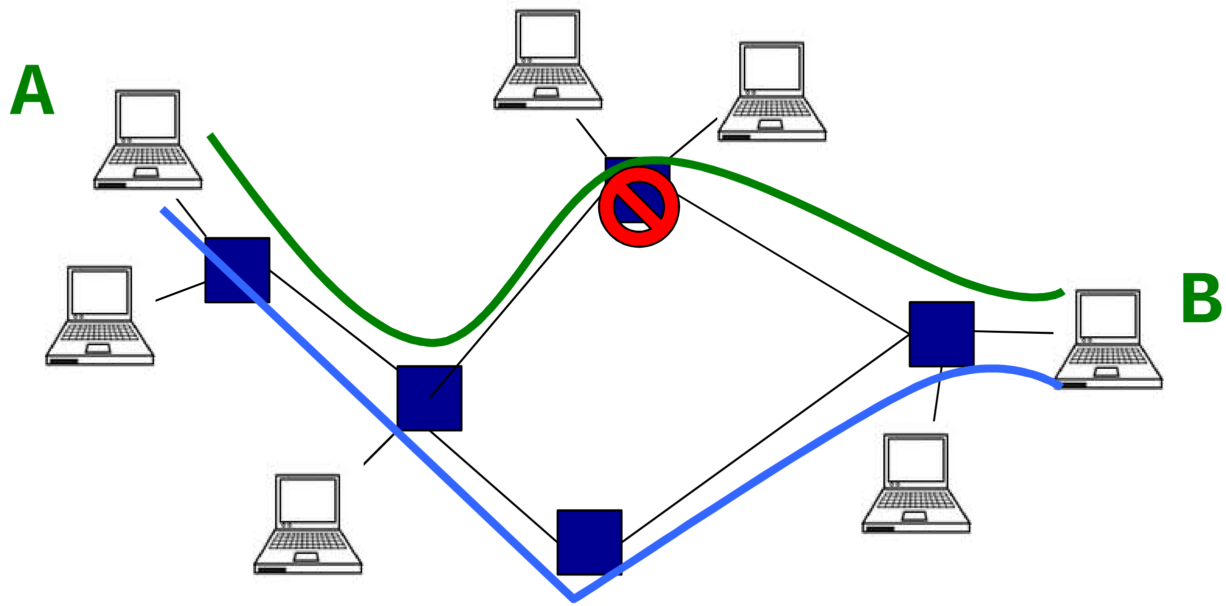


Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)

- Cons
  - wastes bandwidth if traffic is “bursty”
  - **connection setup time is overhead**

Circuit switching



Circuit switching doesn't "route around trouble"

Circuit switching: pros and cons

- Pros
  - guaranteed performance
  - fast transfers (once circuit is established)
- Cons
  - wastes bandwidth if traffic is “bursty”
  - connection setup time is overhead
  - **recovery from failure is slow**

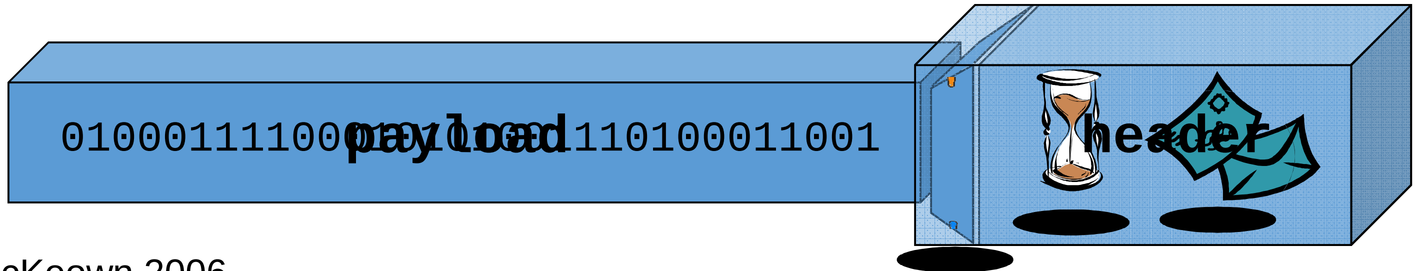
Two forms of switched networks

- Circuit switching (e.g., telephone network)
- Packet switching (e.g., Internet)

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”*

1. Internet Address
2. Age
3. Checksum to protect header



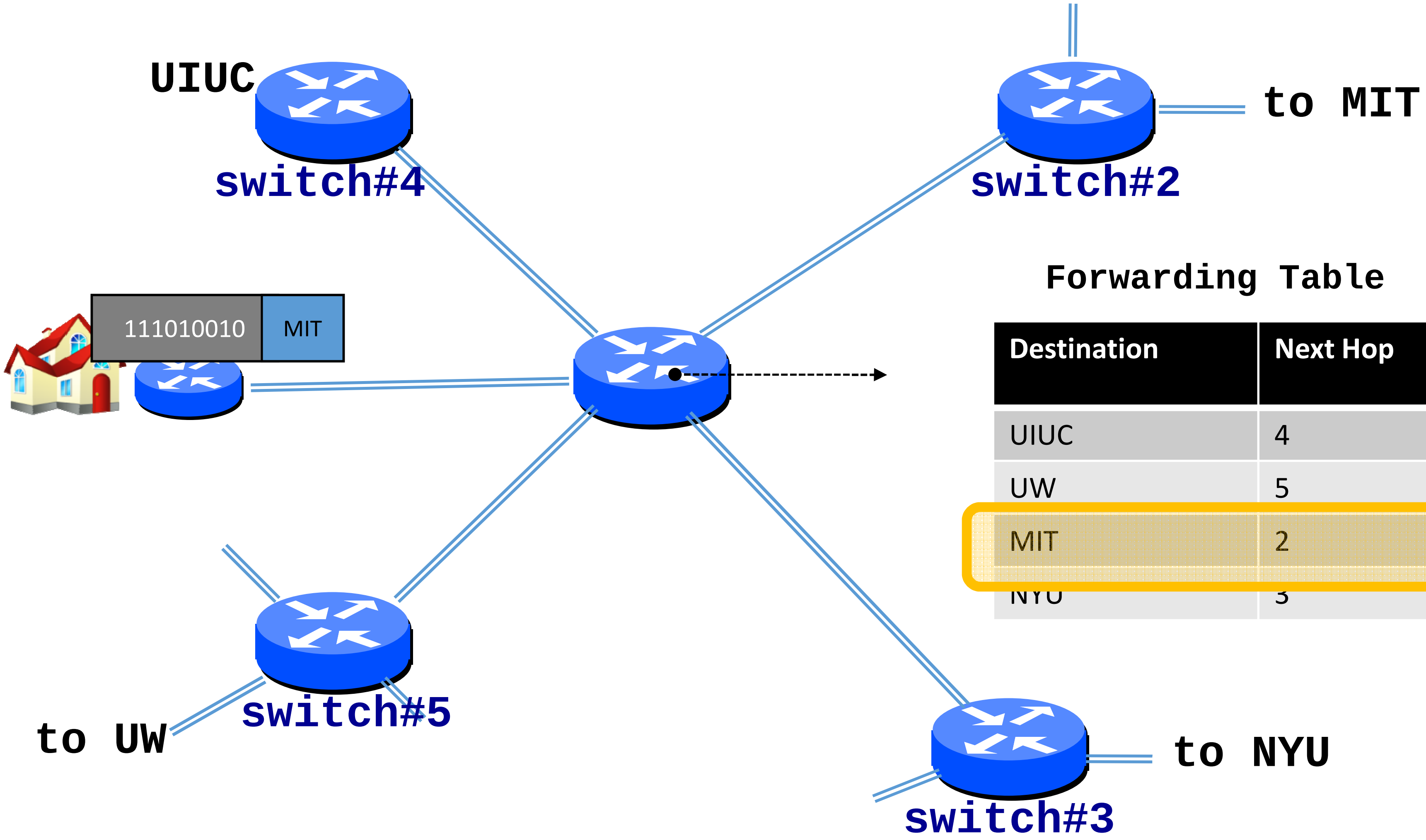
Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a "header" and "payload"*
  - payload is the data being carried
  - header holds instructions to the network for how to handle packet (think of the header as an interface!)

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers

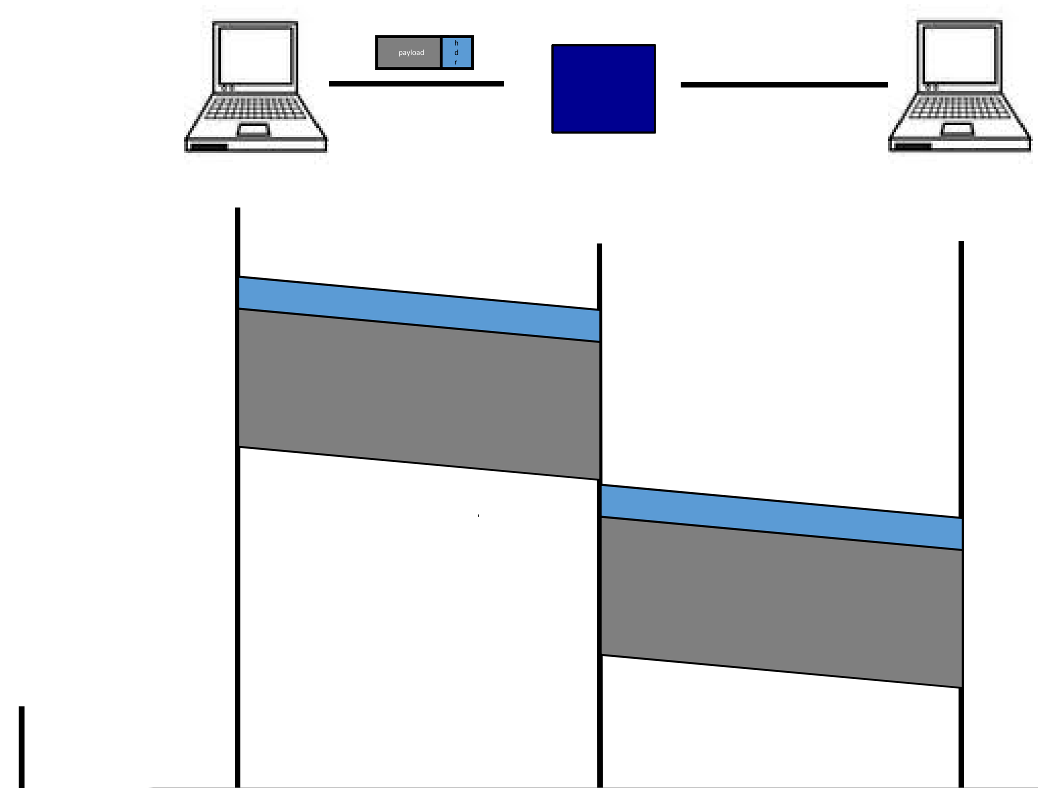
Switches forward packets



Forwarding Table

Destination	Next Hop
UIUC	4
UW	5
MIT	2
NYU	3

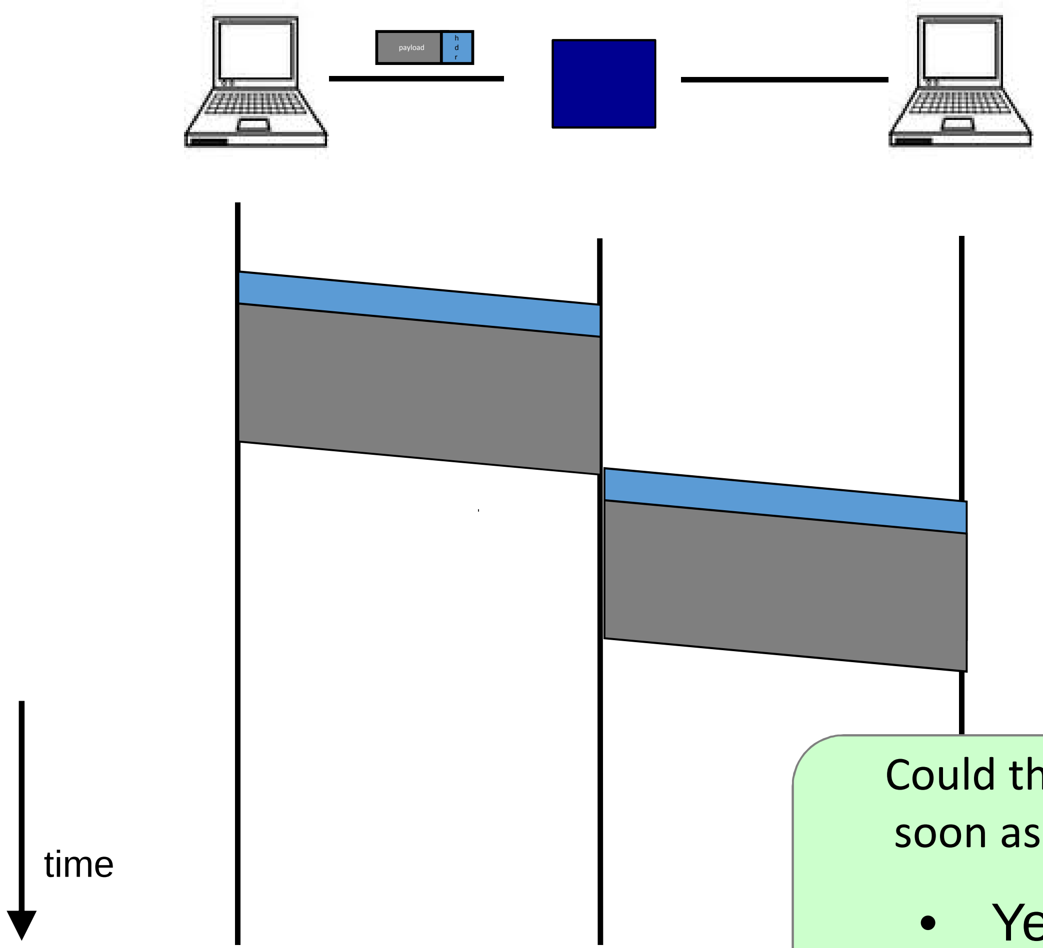
Timing in Packet Switching



What about the time to process the packet at the switch?

- We’ll assume it’s relatively negligible (mostly true)

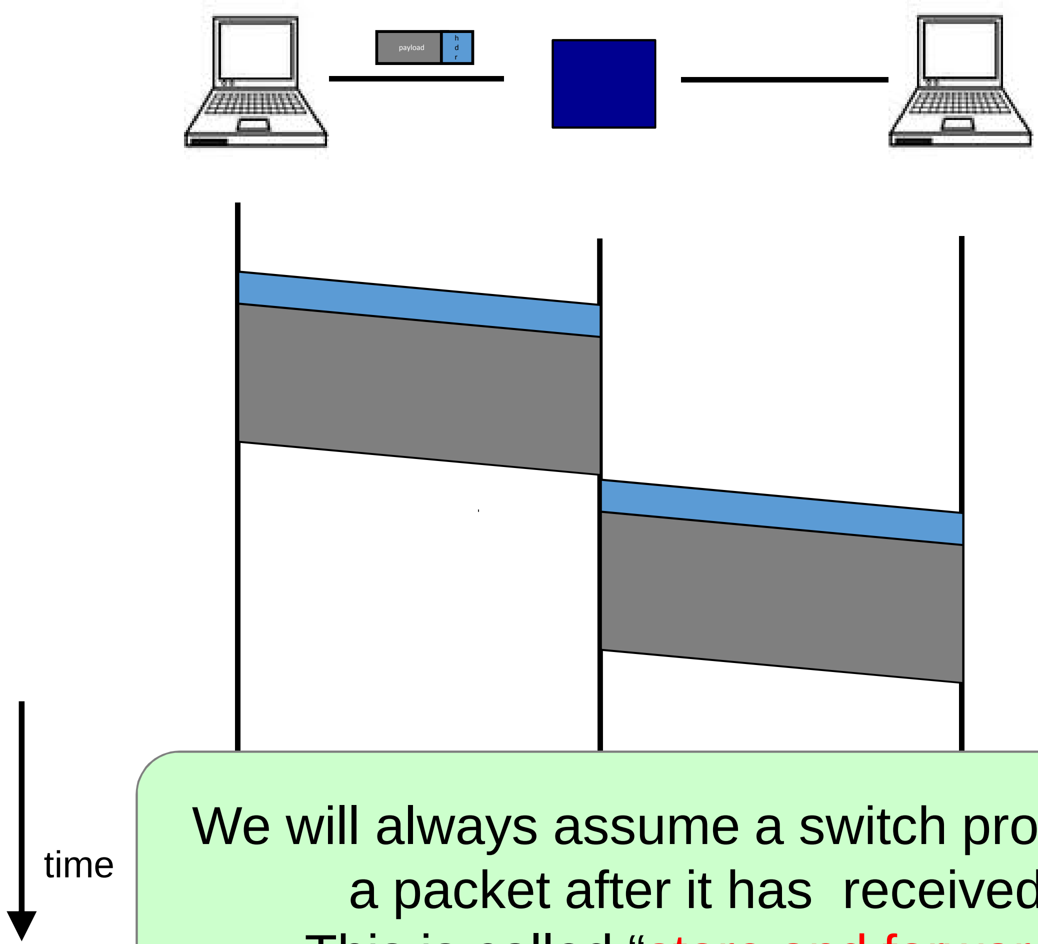
Timing in Packet Switching



Could the switch start transmitting as soon as it has processed the header?

- Yes! This would be called a “cut through” switch

Timing in Packet Switching



We will always assume a switch processes/forwards a packet after it has received it entirely. This is called "store and forward" switching

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers

Packet Switching

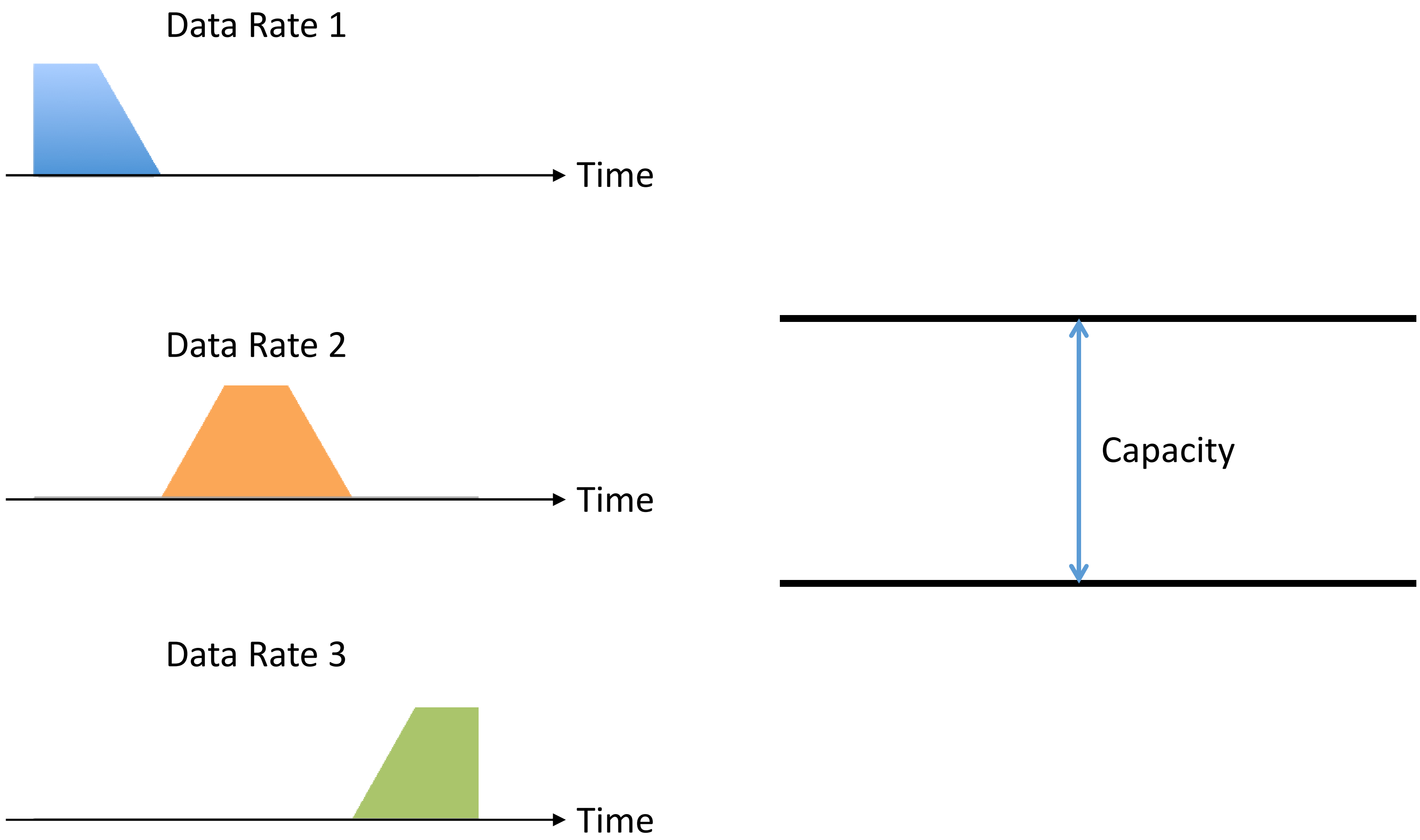
- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
  - no notion of packets belonging to a “circuit”

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead packet switching leverages statistical multiplexing

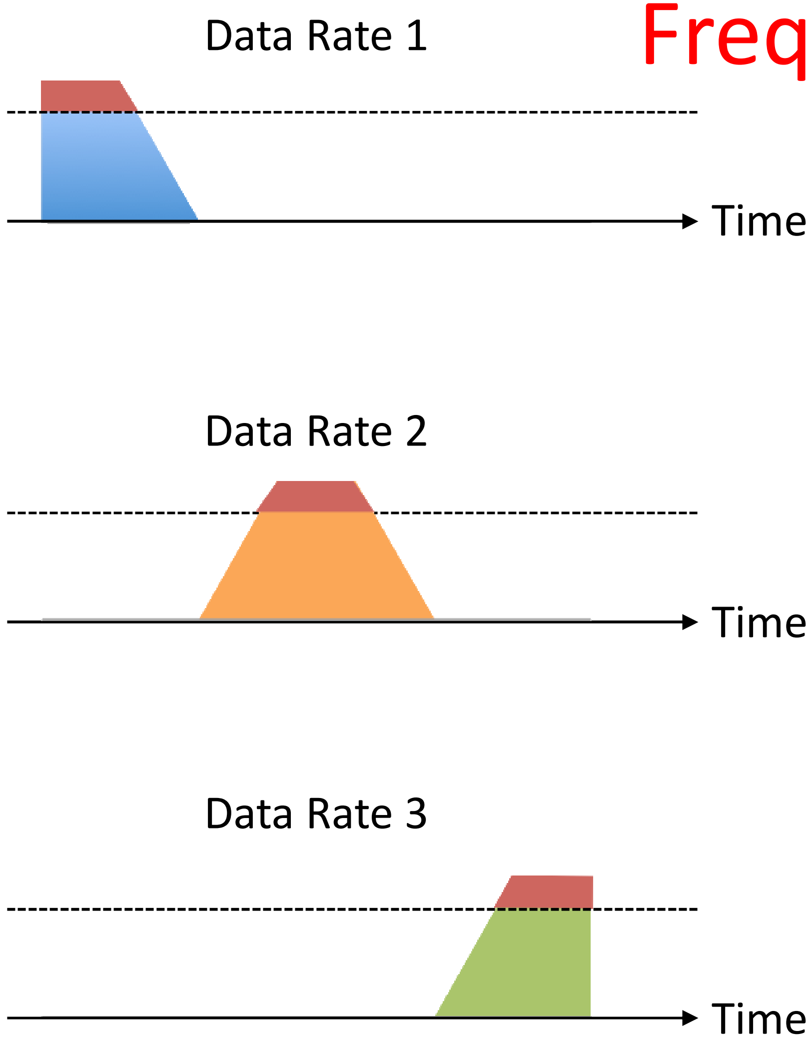
Statistical Multiplexing

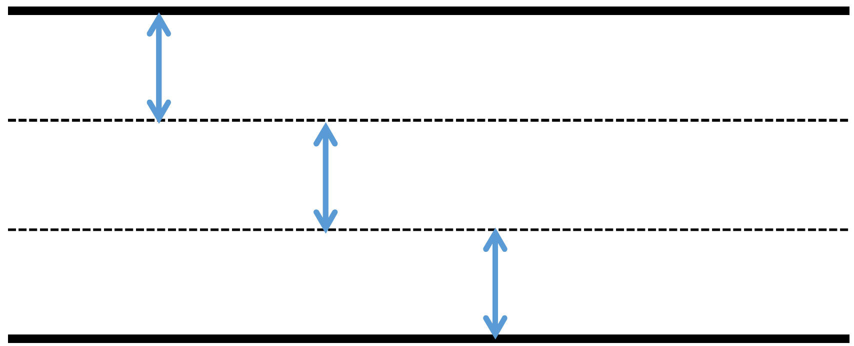
Three Flows with Bursty Traffic



When Each Flow Gets 1/3rd of Capacity

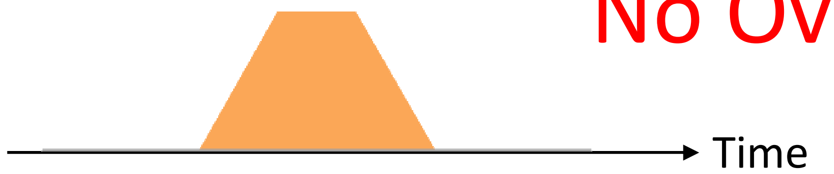
Frequent Overloading





When flows share total capacity

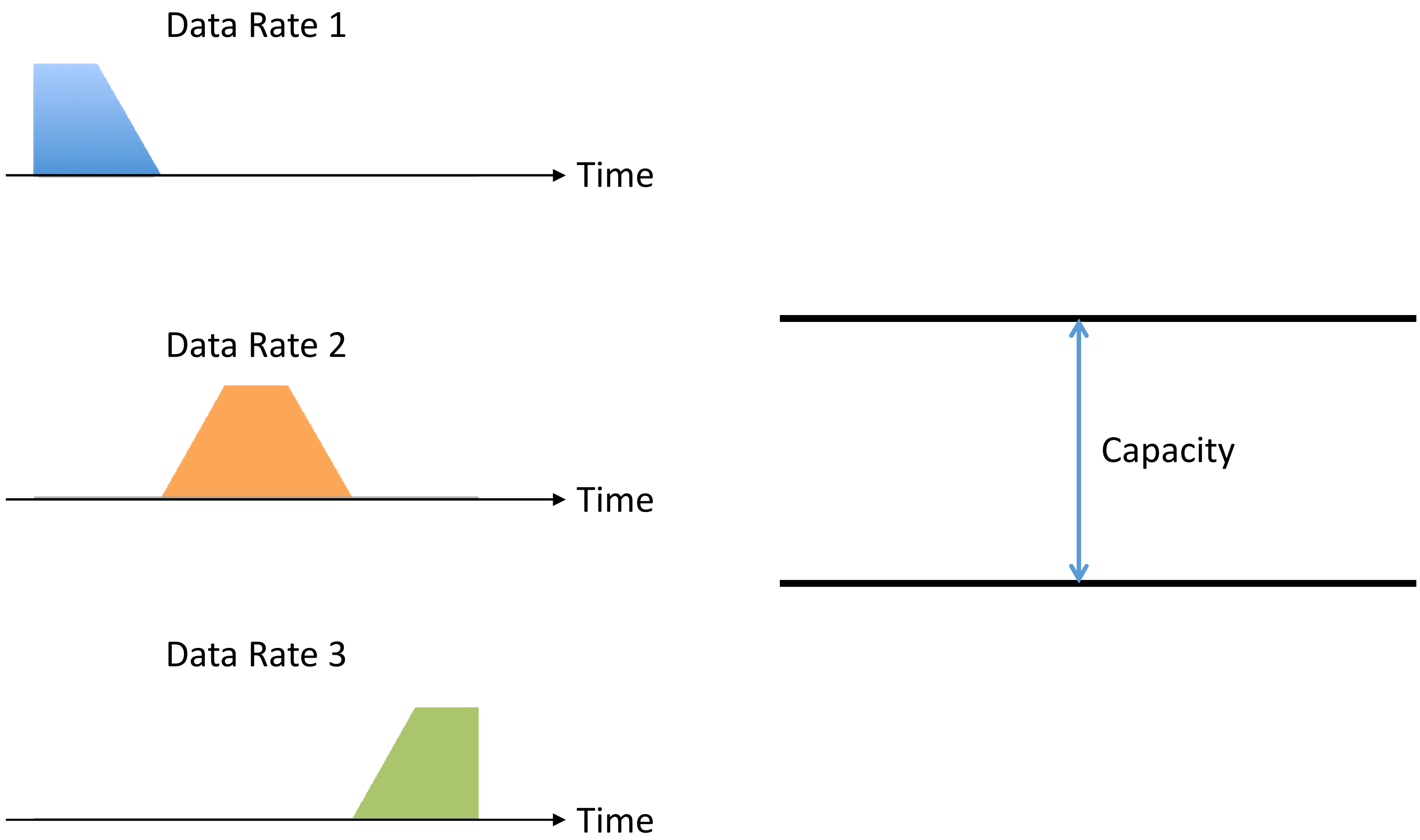
No Overloading



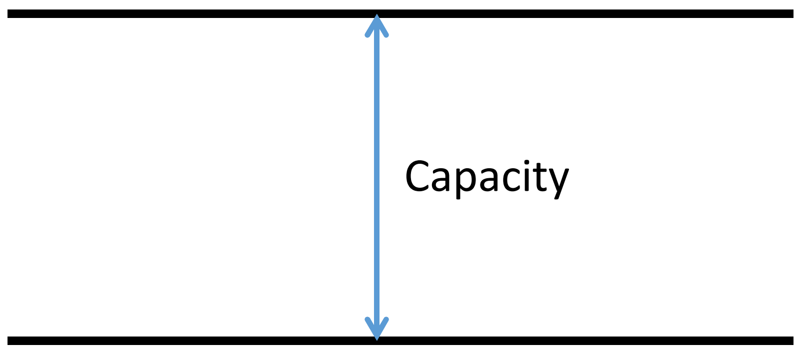
Statistical multiplexing relies on the assumption that not all flows burst at the same time.

Very similar to insurance, and has same failure case

Three Flows with Bursty Traffic

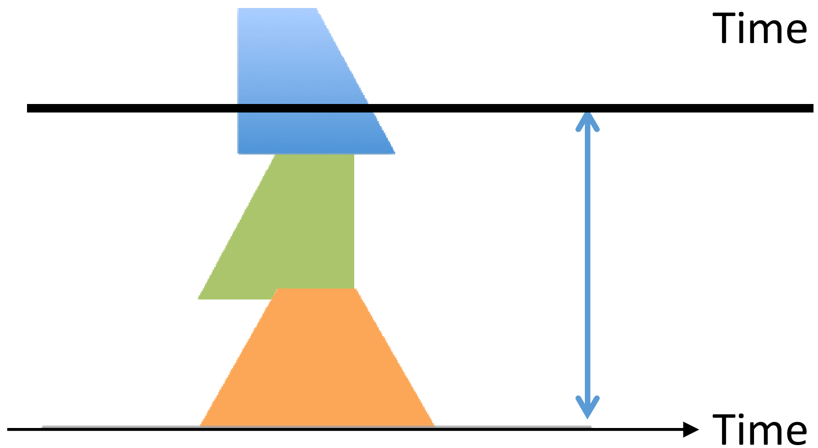


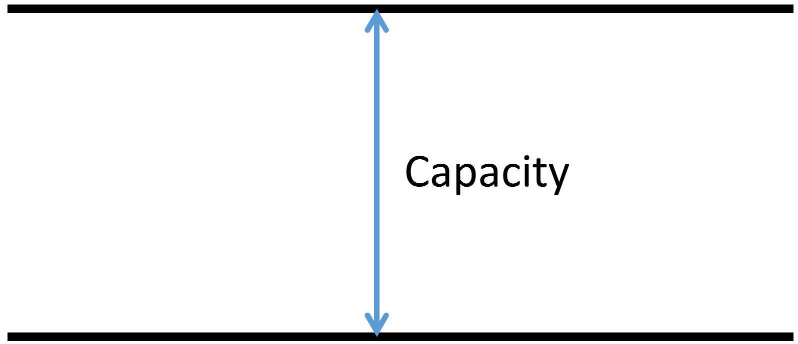
Three Flows with Bursty Traffic



Three Flows with Bursty Traffic

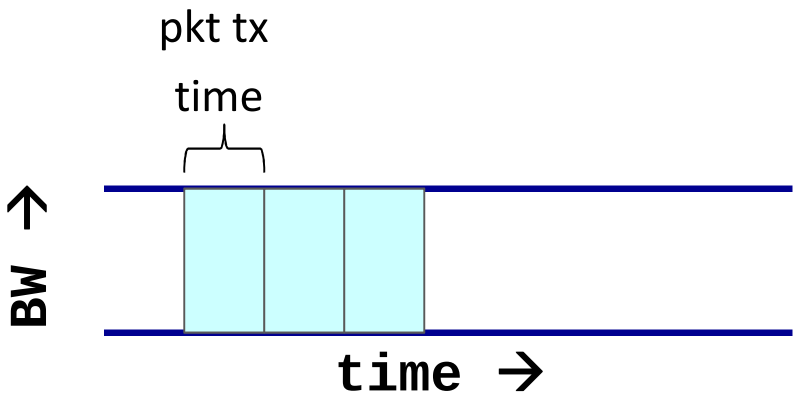
Data Rate 1+2+3 >> Capacity



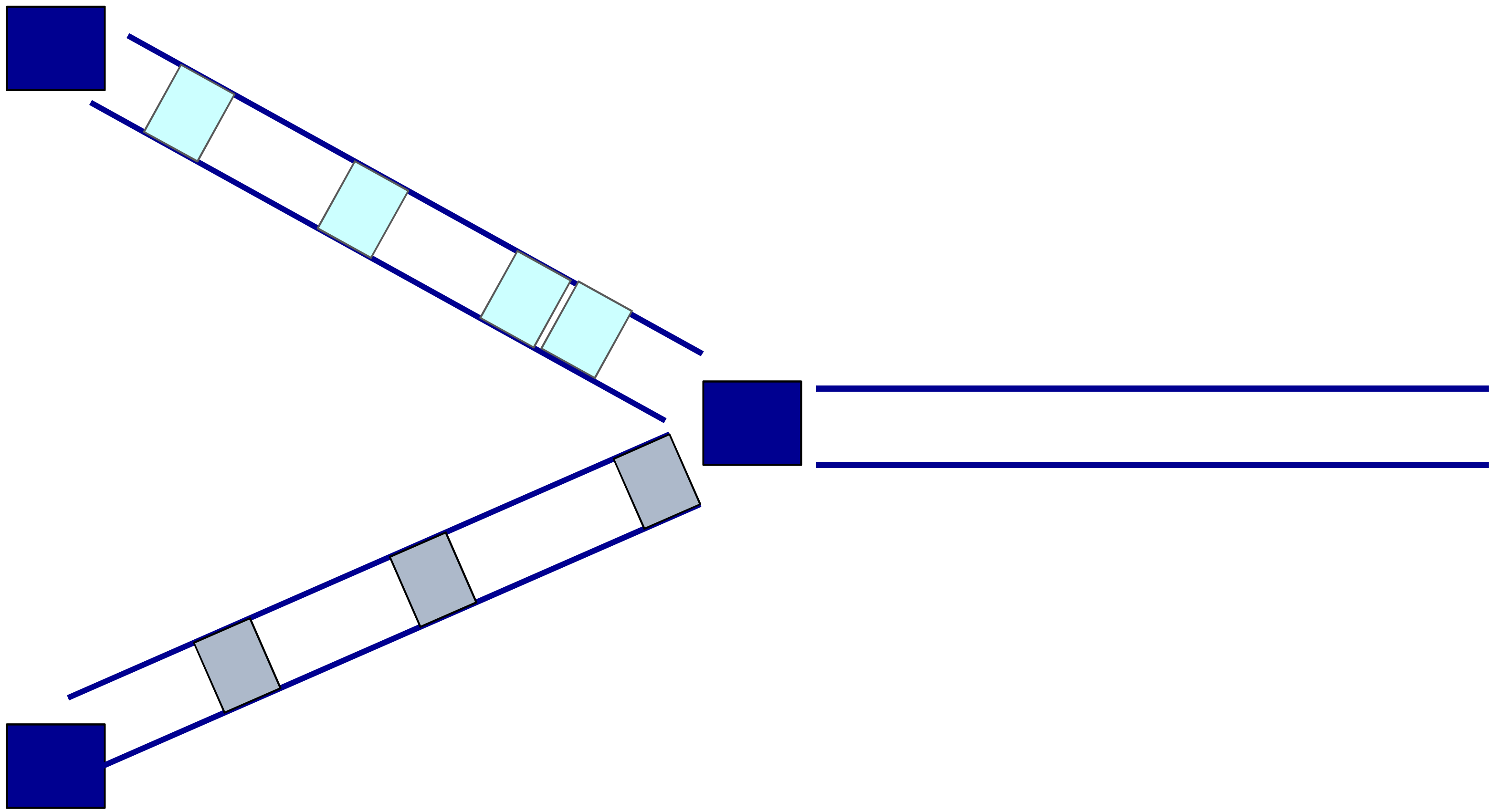


What do we do under overload?

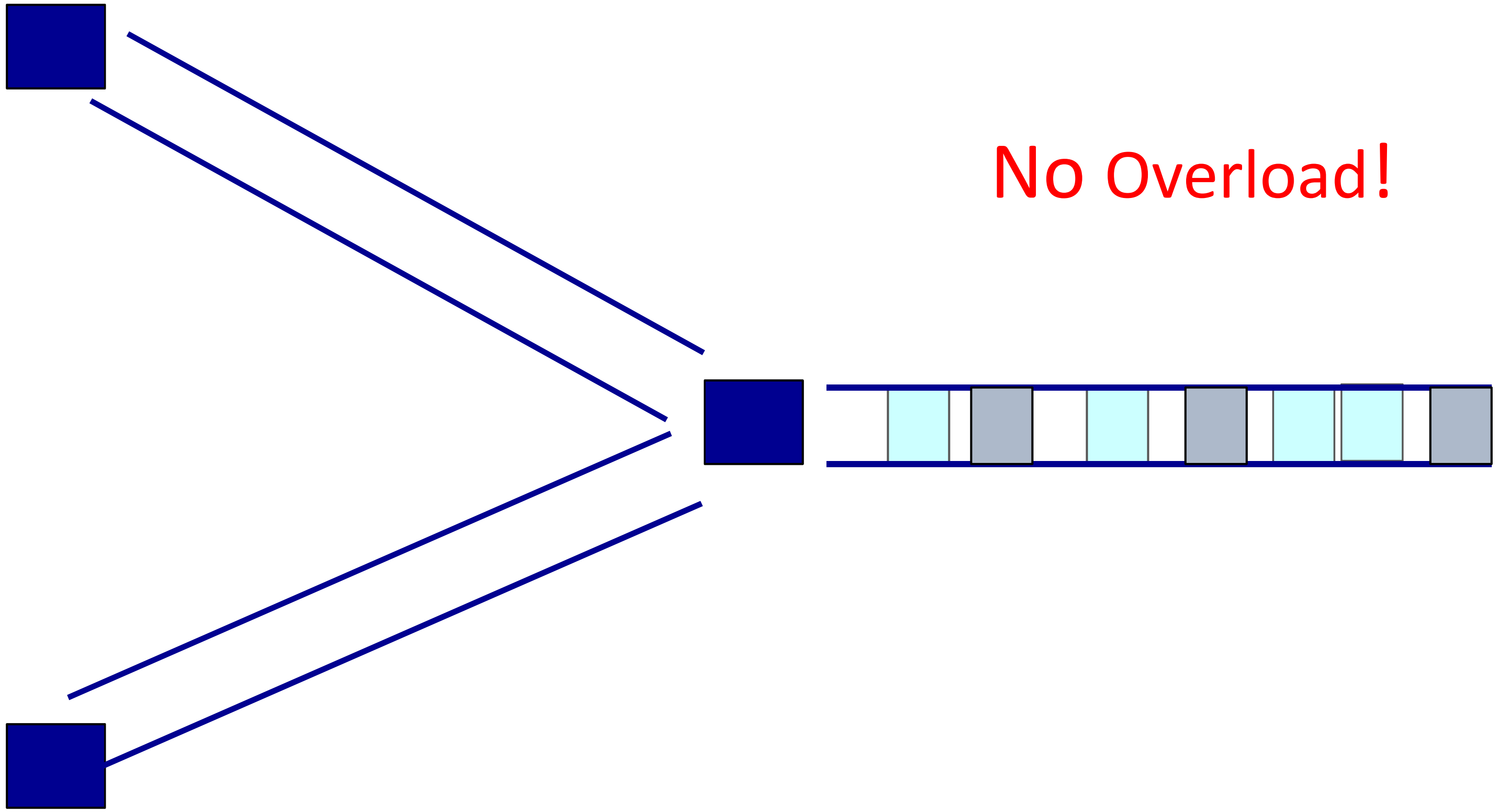
Statistical multiplexing: pipe view



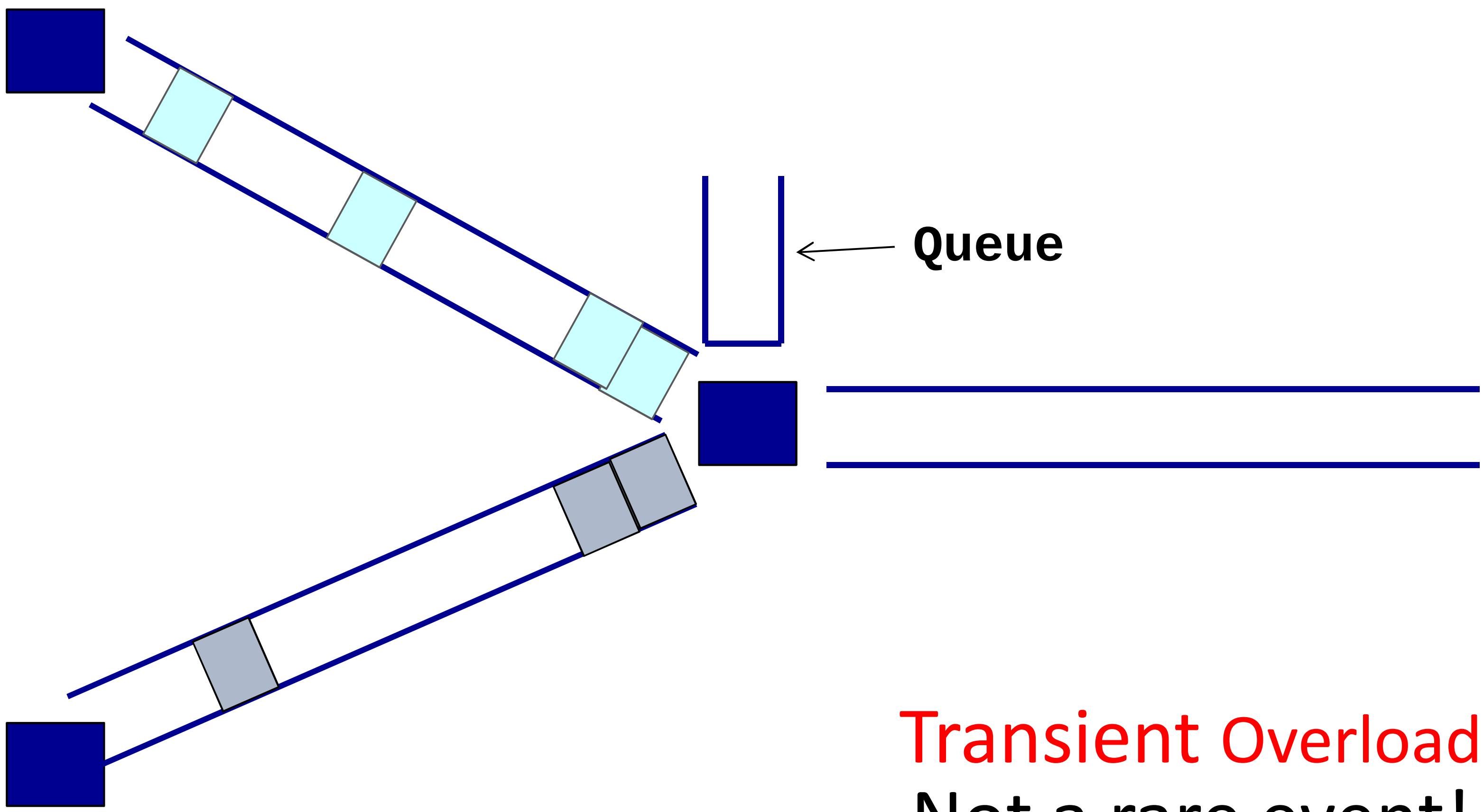
Statistical multiplexing: pipe view



Statistical multiplexing: pipe view

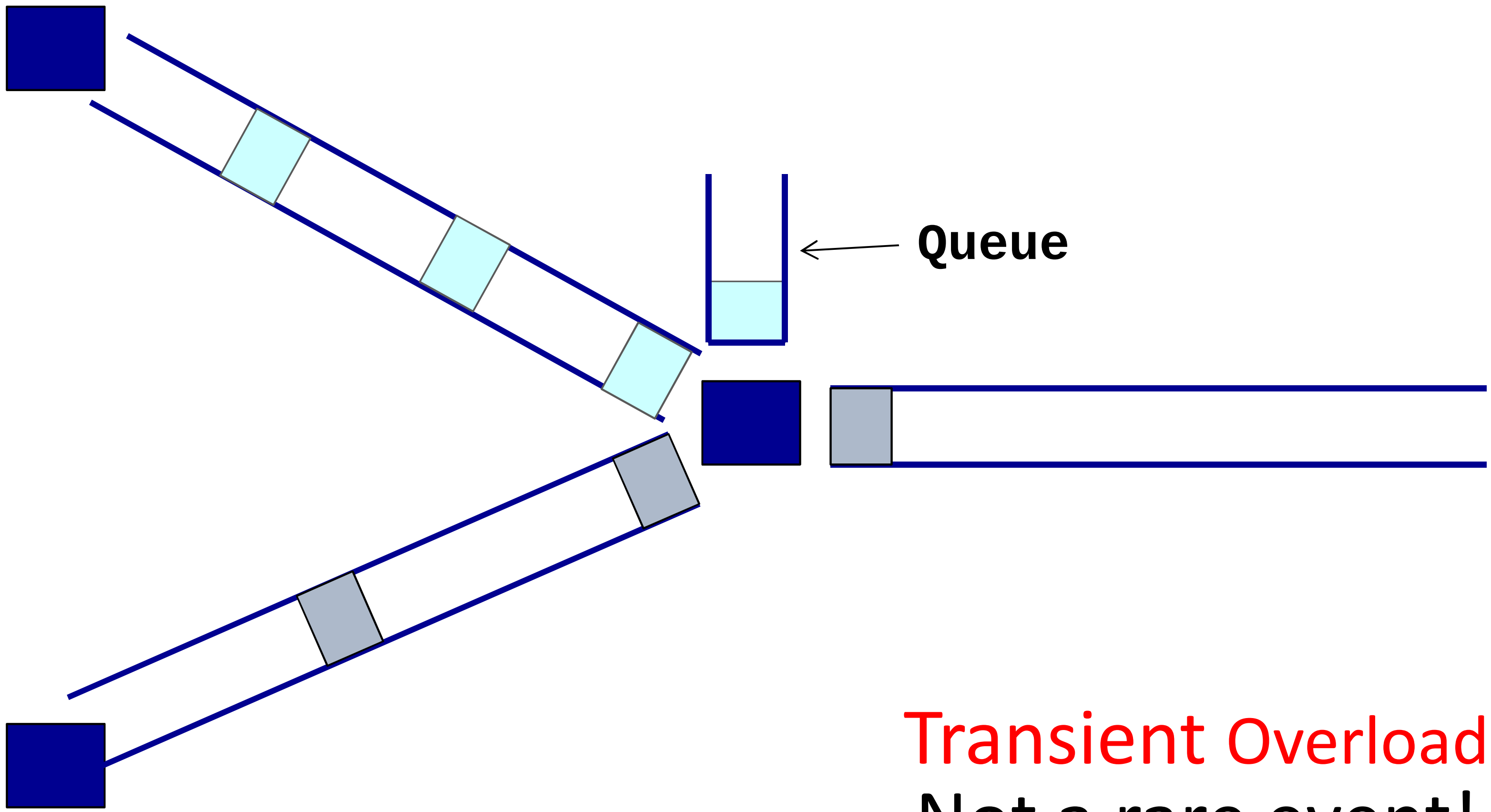


Statistical multiplexing: pipe view



Transient Overload
Not a rare event!

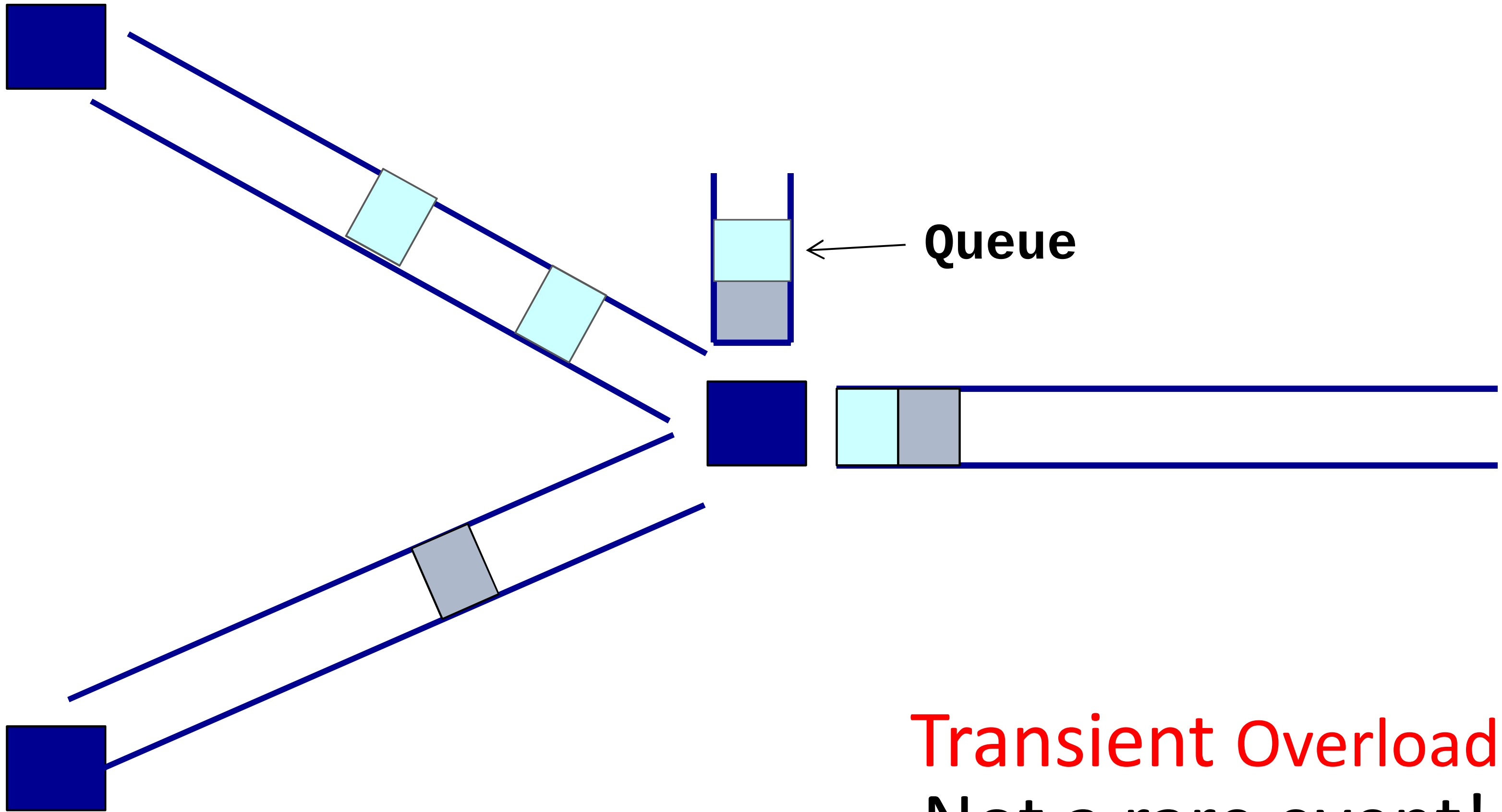
Statistical multiplexing: pipe view



Transient Overload

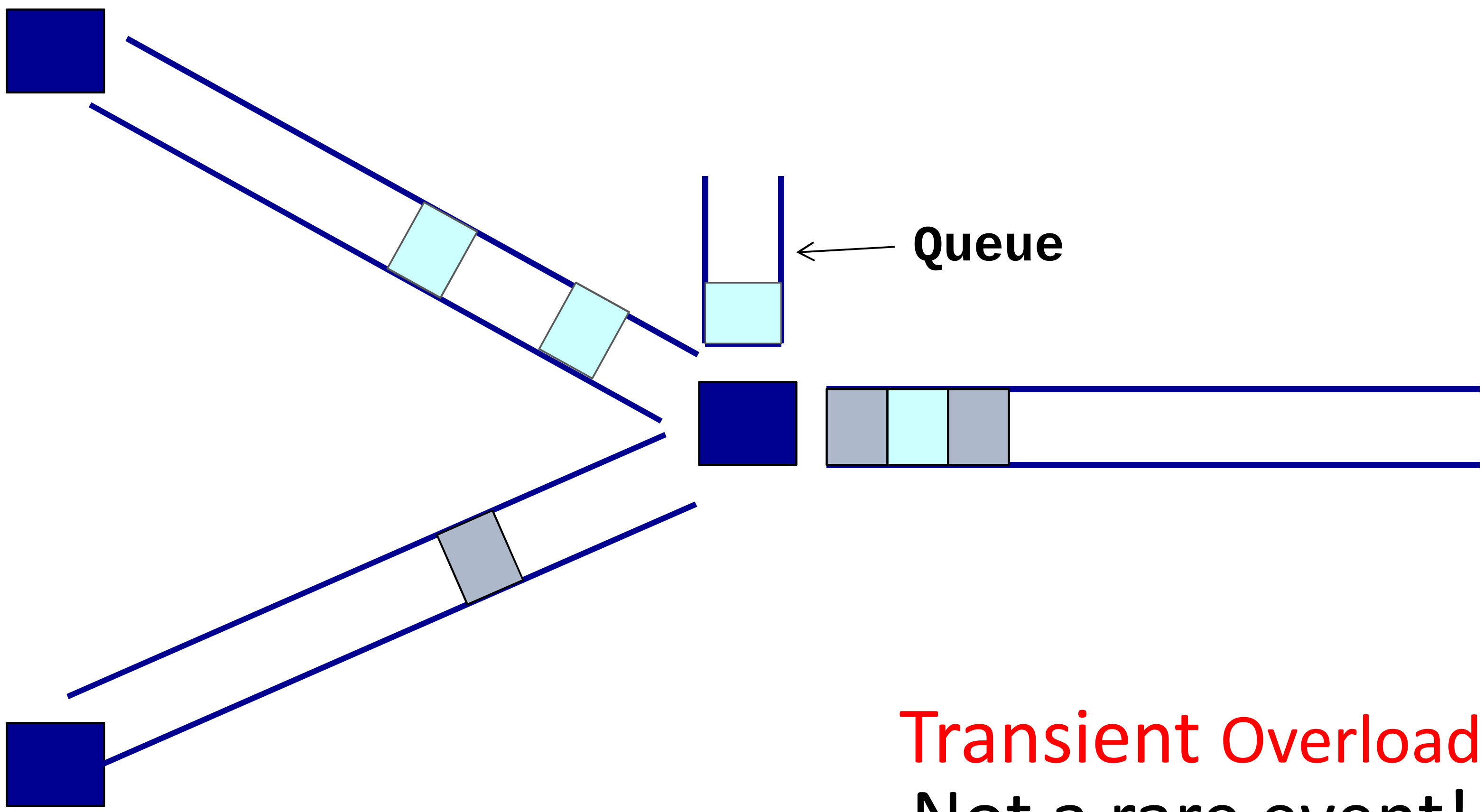
Not a rare event!

Statistical multiplexing: pipe view



Transient Overload
Not a rare event!

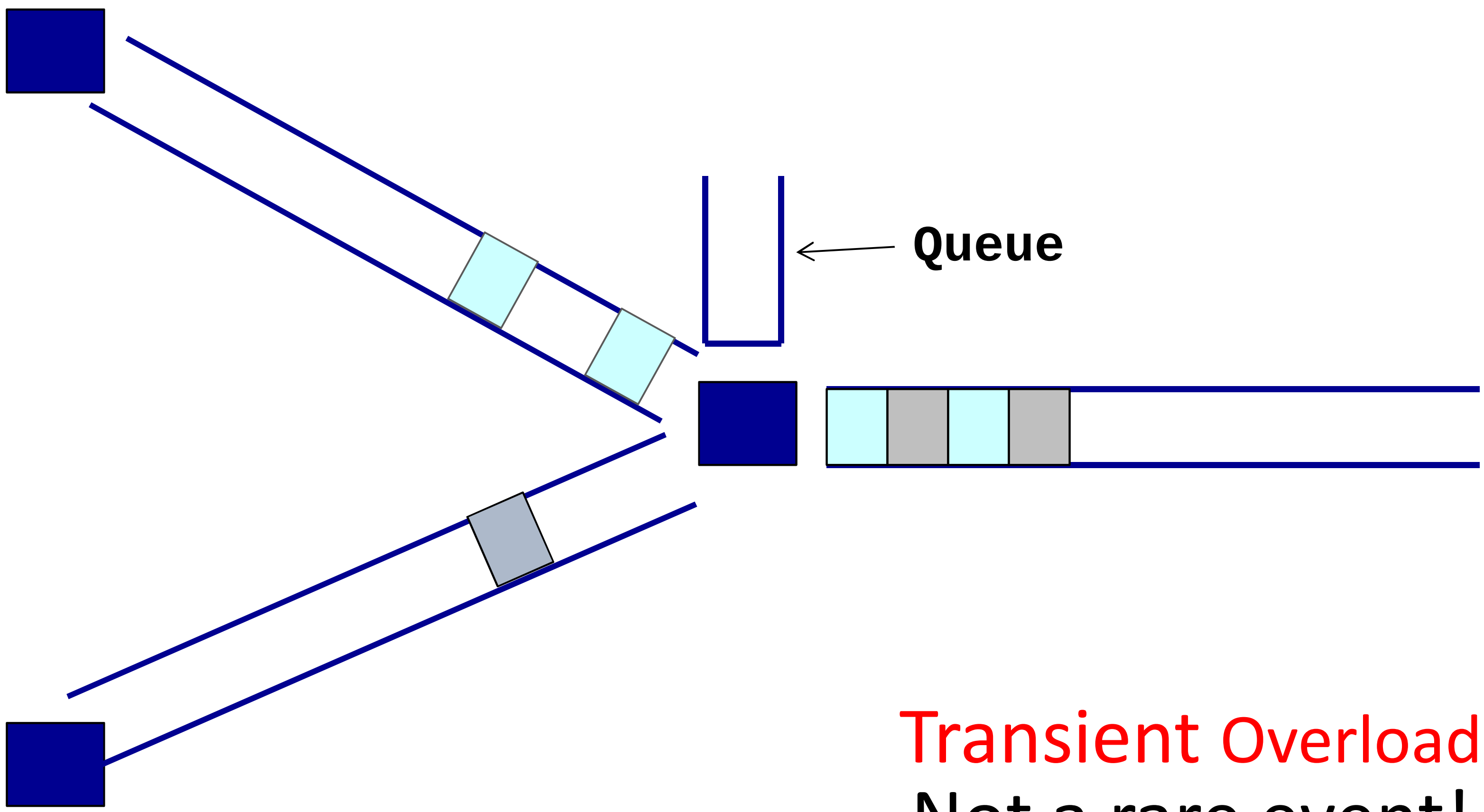
Statistical multiplexing: pipe view



Transient Overload

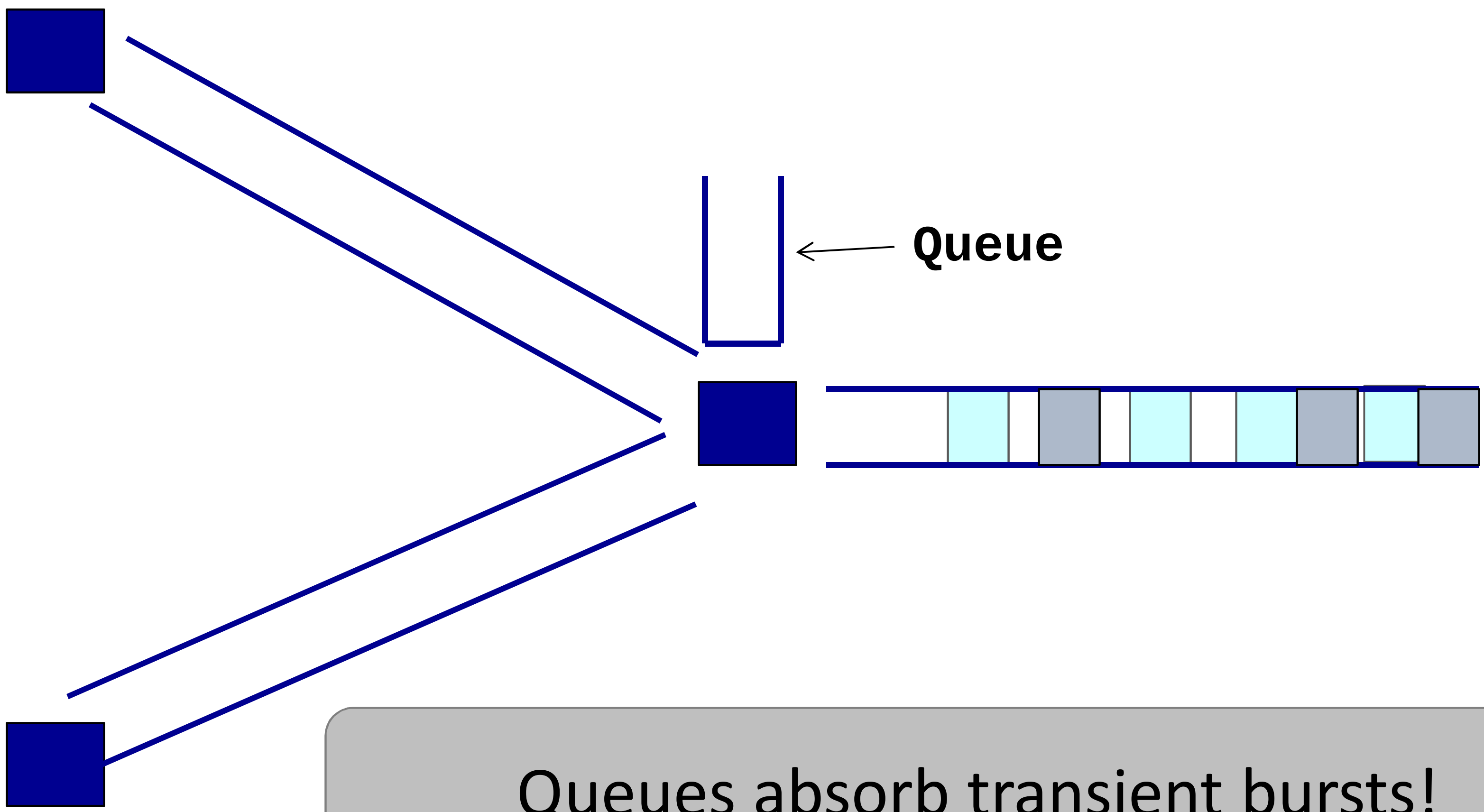
Not a rare event!

Statistical multiplexing: pipe view



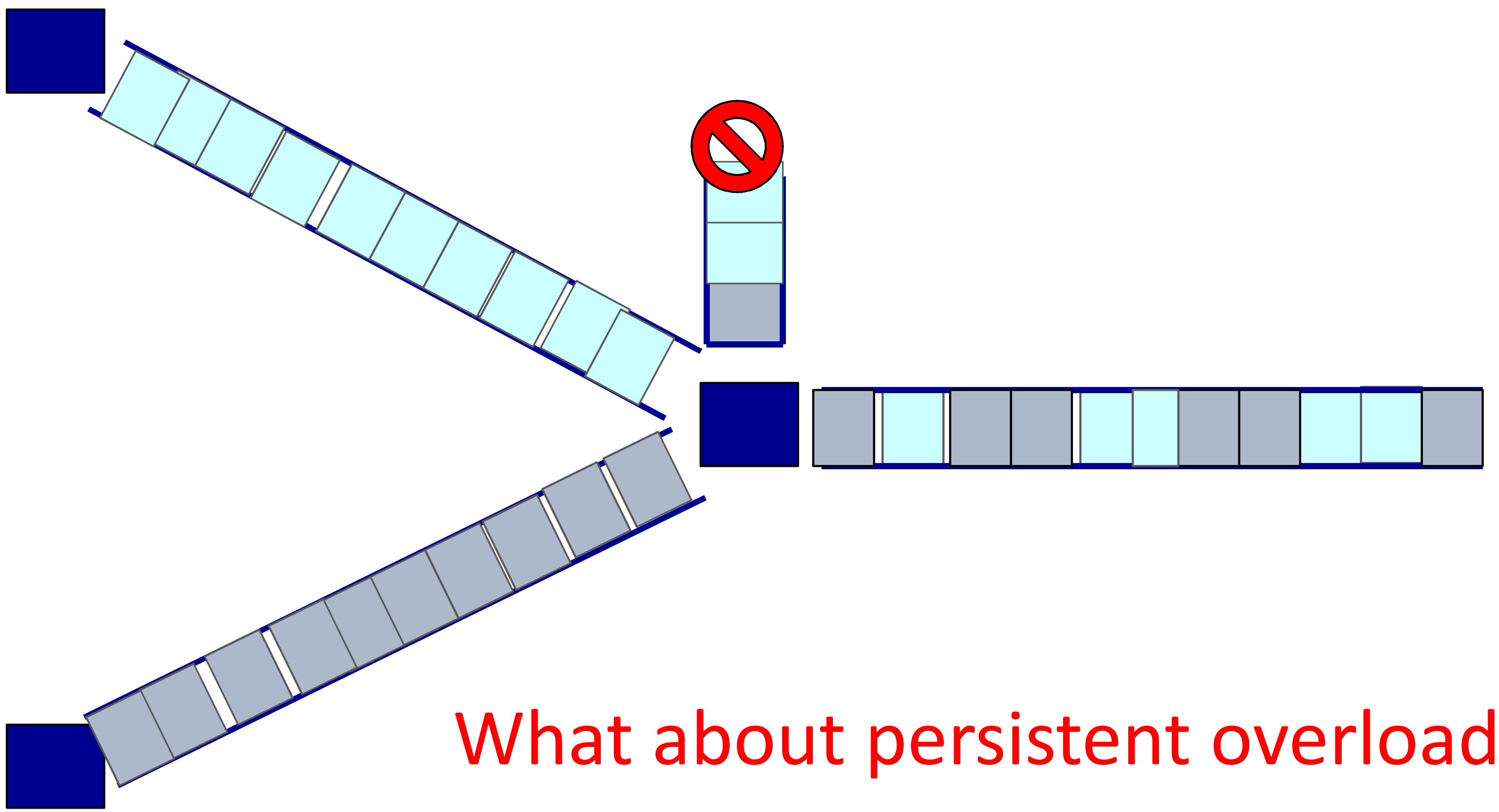
Transient Overload
Not a rare event!

Statistical multiplexing: pipe view



Queues absorb transient bursts!

Statistical multiplexing: pipe view



What about persistent overload?

Will eventually drop packets

Queues introduce queuing delays

- Recall, packet delay = tx delay + prop delay

- With queues (stat. muxing)
  - packet delay = tx delay + prop delay + queuing delay

- Queuing delay caused by “packet interference”

- Made worse at high load
  - less “idle time” to absorb bursts
  - think about traffic jams at rush hour

Basic Queuing Theory Terminology

- Arrival process: how packets arrive
  - Average rate A
  - Peak rate P
- Service process: transmission times
  - Average transmission time (function of packet size)
- W: average time packets wait in the queue
  - W for “waiting time”
- L: average number of packets waiting in the queue
  - L for “length of queue”
- Two different quantities

Little's Law (1961)

$$L = A \times W$$

- Average queue size (L) is average packet arrival rate (A) times average wait time (W)
  - Surprisingly, not influenced by arrival process distribution, service distribution, service order, etc.
- Used for router queue, circuit dimensioning
- More complex computation: X^{th} percentile queue size
  - Realm of queuing theory
  - Depends on arrival/service processes

Statistical Multiplexing is a recurrent theme in computer science

- Phone network rather than dedicated lines
  - Ancient history
- Packet switching rather than circuits
  - Today's lecture
- Cloud computing
  - Shared datacenters, rather than single PCs

Packet Switching

- Data is sent as chunks of formatted bits (Packets)
- Packets consist of a “header” and “payload”
- Switches “forward” packets based on their headers
- Each packet travels independently
- No link resources are reserved in advance. Instead packet switching leverages statistical multiplexing
  - allows efficient use of resources
  - but introduces queues and queuing delays

Circuit switching: pros and cons

- Pros
    - guaranteed performance
    - fast transfers (once circuit is established)
- Cons
    - wastes bandwidth if traffic is “bursty”
    - connection setup adds delay
    - recovery from failure is slow

Packet switching: pros and cons

- Cons
  - no guaranteed performance
  - header overhead per packet
  - queues and queuing delays
- Pros
  - efficient use of bandwidth (stat. muxing)
  - no overhead due to connection setup
  - resilient -- can `route around trouble'

Recap: you should leave this lecture with…

- A sense of how the basic `plumbing' works
  - links and switches
  - packet delays (tx, propagation, queuing)
  - statistical multiplexing and queues
  - circuit vs. packet switching

- Next: back to lofty principles
  - protocols and layering