EECS 489
Computer Networks

Winter 2025

Mosharaf Chowdhury

Material with thanks to Aditya Akella, Sugih Jamin, Philip Levis, Sylvia Ratnasamy, Peter Steenkiste, and many other colleagues.

Agenda

- Introductions
- Class policies, logistics, and roadmap
- Overview of the basics
  - How is the network shared?
  - How do we evaluate a network?
  - What is a network made of?

Staff

Aditya Singhvi

Efe Akinci

Alex de la Iglesia

- Office hours: See course webpage
- No office hours this week or discussions this week

Mosharaf Chowdhury

- @Michigan since 2016
- Research: SymbioticLab.org

- Office hours: Thursdays 2:45PM – 3:45PM (in-person @ 4820 BBB)

- No office hours this week
- Lectures will be recorded

489 in EECS curriculum

- **EECS 281**
  - High-level logic ⇒ Programs
  - Coding skills learned in 281 are critical for 489 assignments
- *EECS 482*
  - How do machines work?
  - Execute programs, interact with users, etc.
  - Prior 482 experience is not needed

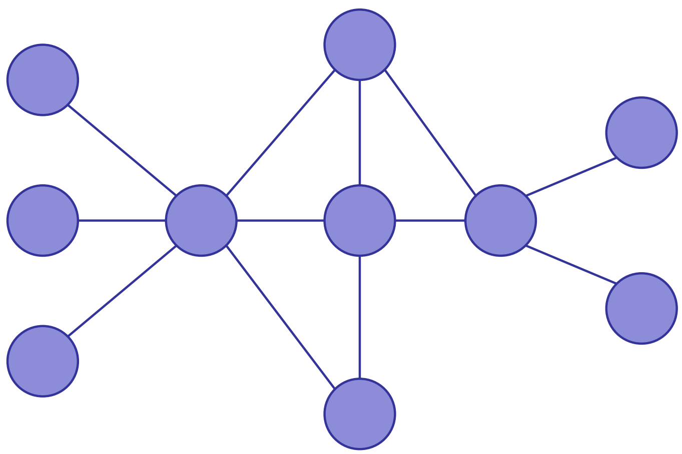
What is missing?

- How do we access *most* services?
  - Examples include search engines, social networks, video streaming, etc.
- How do two machines communicate?
  - When they are directly connected
  - When they are not directly connected

- Using a network

What is a network?

- A system of “links” that interconnect “nodes” in order to move “information” between nodes



- We will focus primarily on the Internet

What is EECS 489 about?

- To learn about (at a high level)
  - How the Internet works
  - Why it works the way it does
  - How to reason about complicated design problems

- What it's not about
  - How to write web services
  - How to design web pages
  - …

Class workload

- Four assignments/projects
  - First one is an individual assignment
  - The rest are done as groups
- Exams:
  - Midterm: Week of February 24 (TBD)
  - Final: April 29, 1:30 PM – 3:30 PM
    - » Final covers only the materials after midterm

Grading

	Allocation
Assignment 1	5%
Assignment 2	15%
Assignment 3	15%
Assignment 4	15%
Midterm	25%
Final	25%

The assignments

- Assignment 1: measure end-to-end throughput and delay of networks (i.e., simple speed test)
- Assignment 2: video streaming from CDNs (i.e., simple Netflix)
- Assignment 3: reliable transport (i.e., how to transfer data over an unreliable network)
- Assignment 4: router design (i.e., how do internal elements of the network work)

All on (emulated) realistic networks using *mininet*

Bonus Quizzes

- ~10 MCQ and solution key for each of the 20 lectures
- Made online sometime after the lecture; live for at least 48 hours
- Completing all counts for a maximum of 2% on top of your final grade
  - How well you do doesn't matter

Enrollment and wait list

- Wait-listed students will be admitted in the order of wait list
- If you’re planning to drop, please do so soon!

Communication protocol

- Website: https://github.com/mosharaf/eecs489/
  - Assignments, lecture slides
- Confidential content on Canvas & Gradescope
- Ed for all communication
  - Sign up at https://edstem.org/us/join/jwtp2b

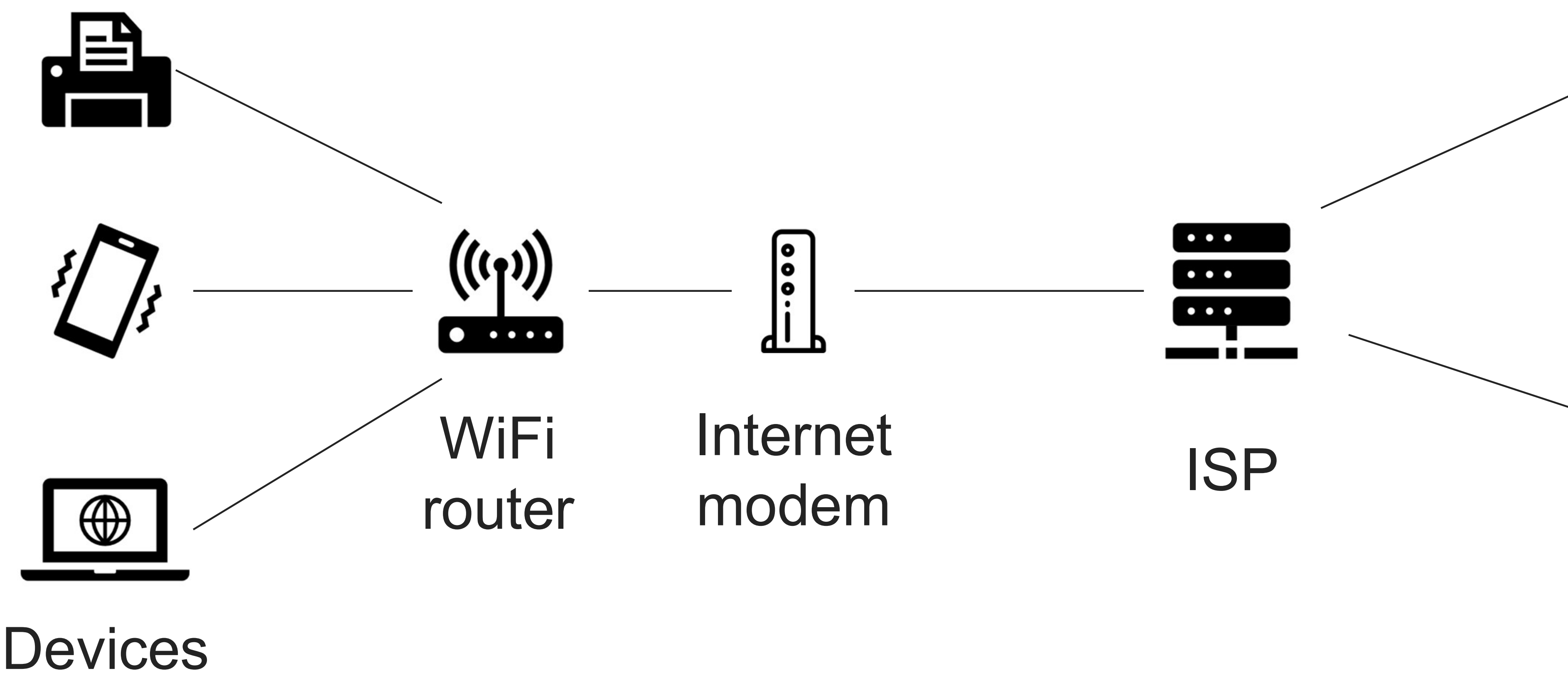
- Assignment submission via g489.eecs.umich.edu

Policies on late submission, re-grade request, cheating …

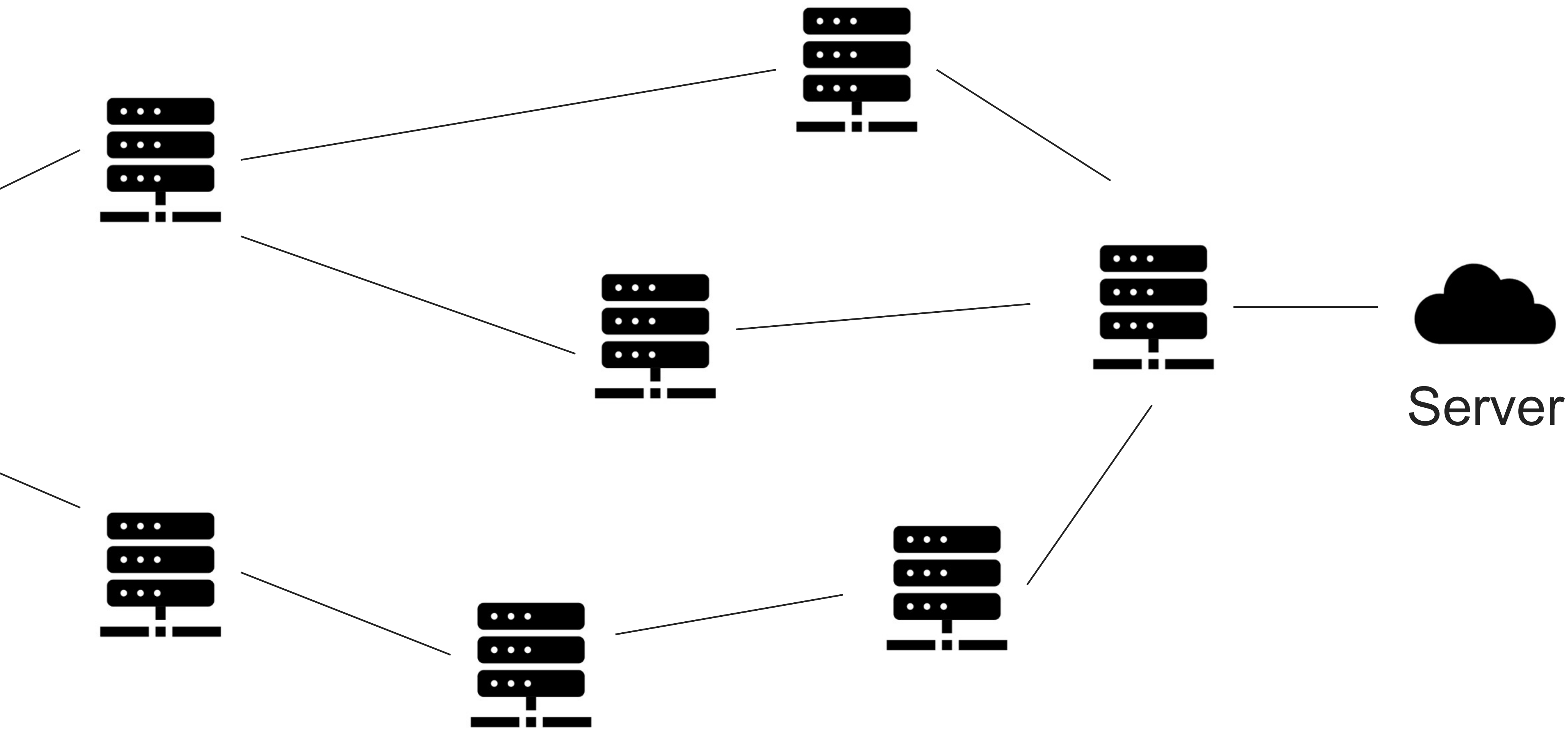
- Detailed description in the course webpage
- Don’t cheat!

LET'S TALK INTERNET

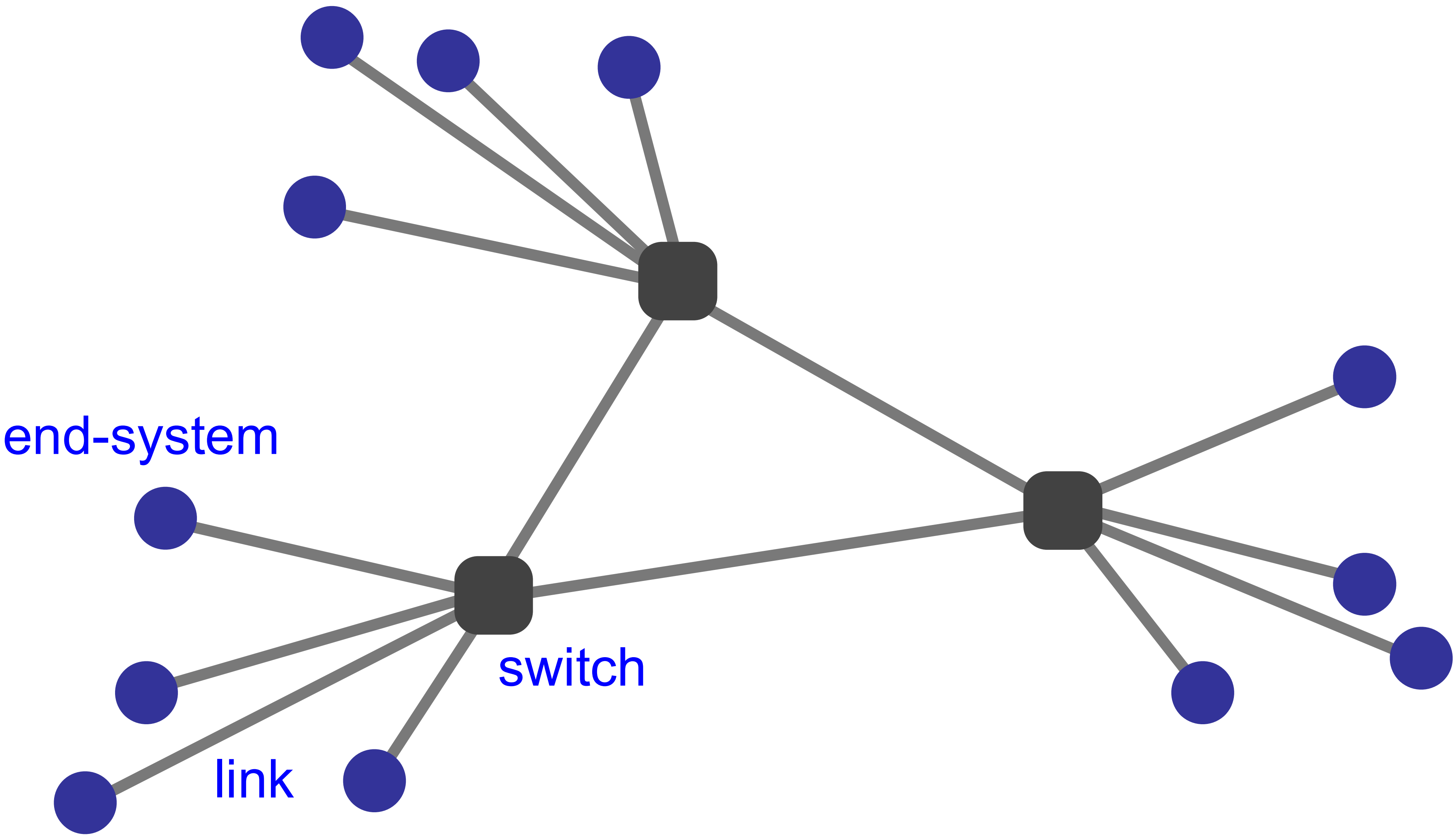
From the home...



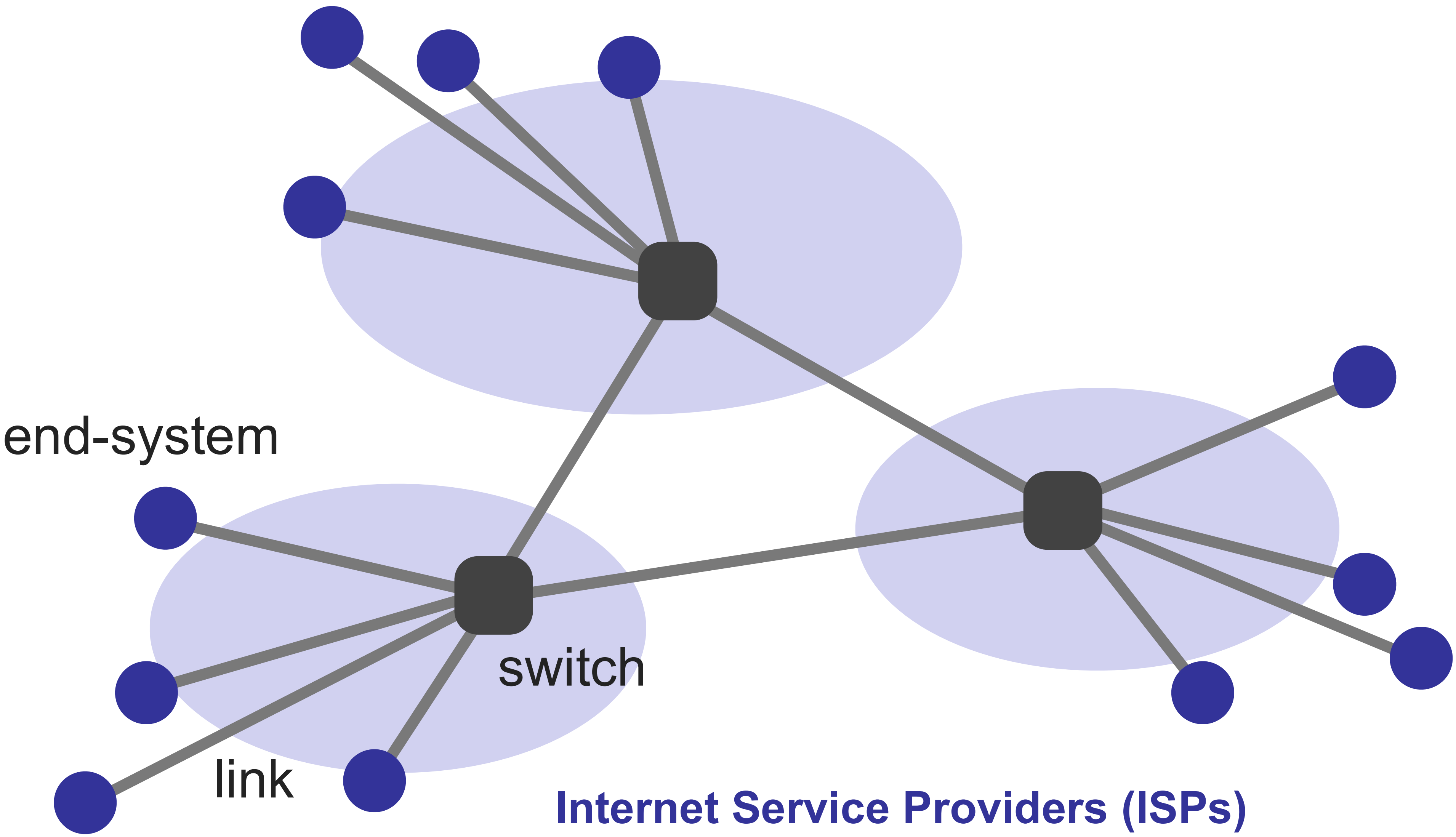
Across the world...



The Internet consists of many end-systems

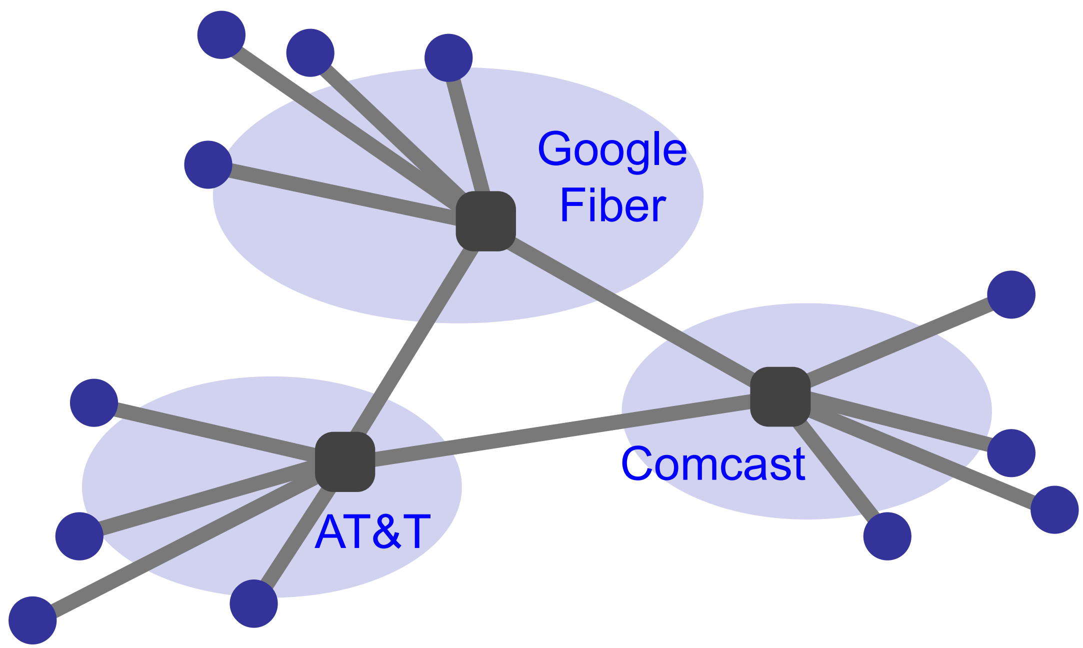


Managed by many parties



A federated system: Logical view

- The Internet ties together different networks **by the IP protocol**
  - *A common interface binds them all together*



A federated system: Physical view

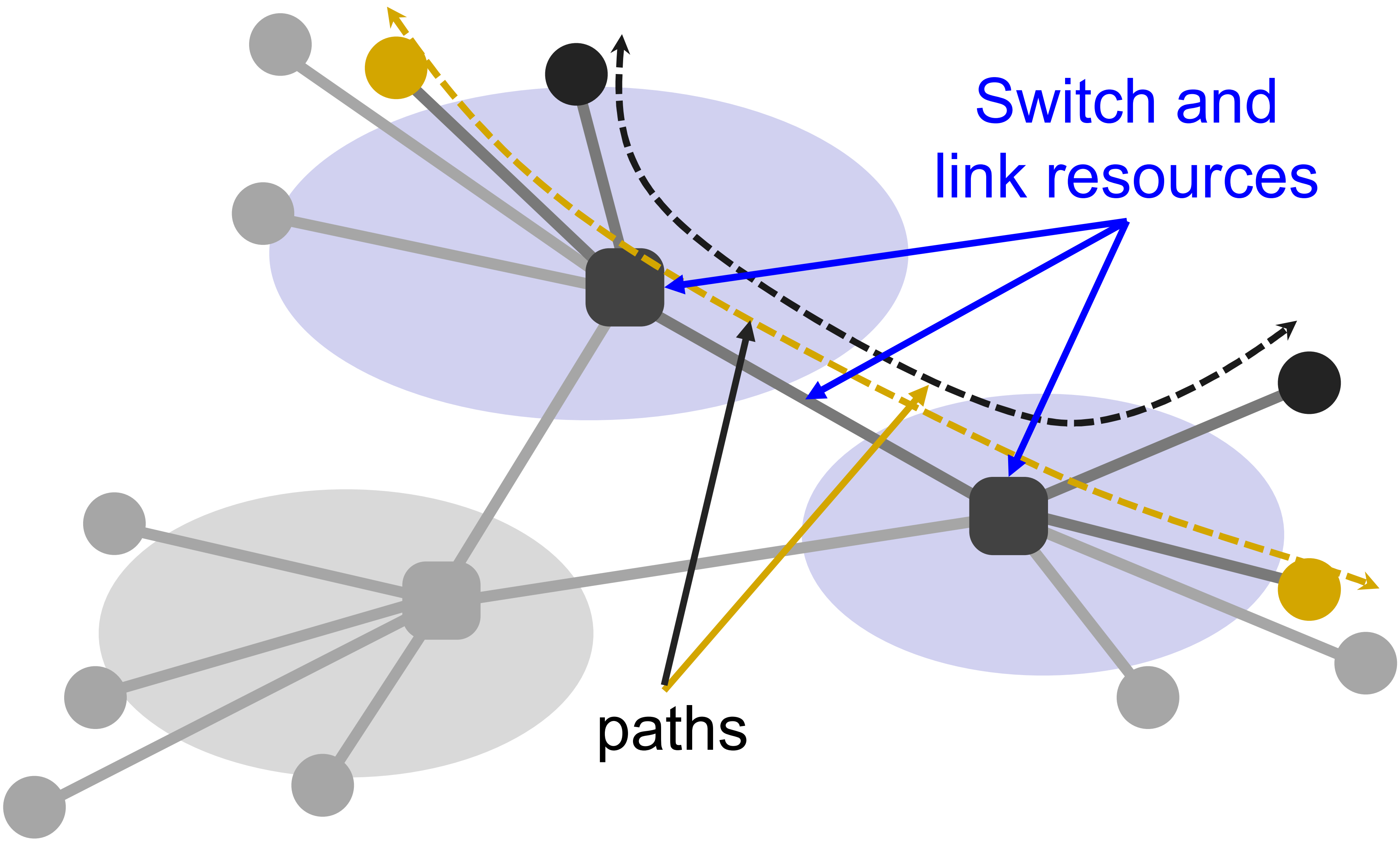
https://kmcd.dev/posts/internet-map-2023/

INTERNET IS A SWITCHED NETWORK (OF NETWORKS)

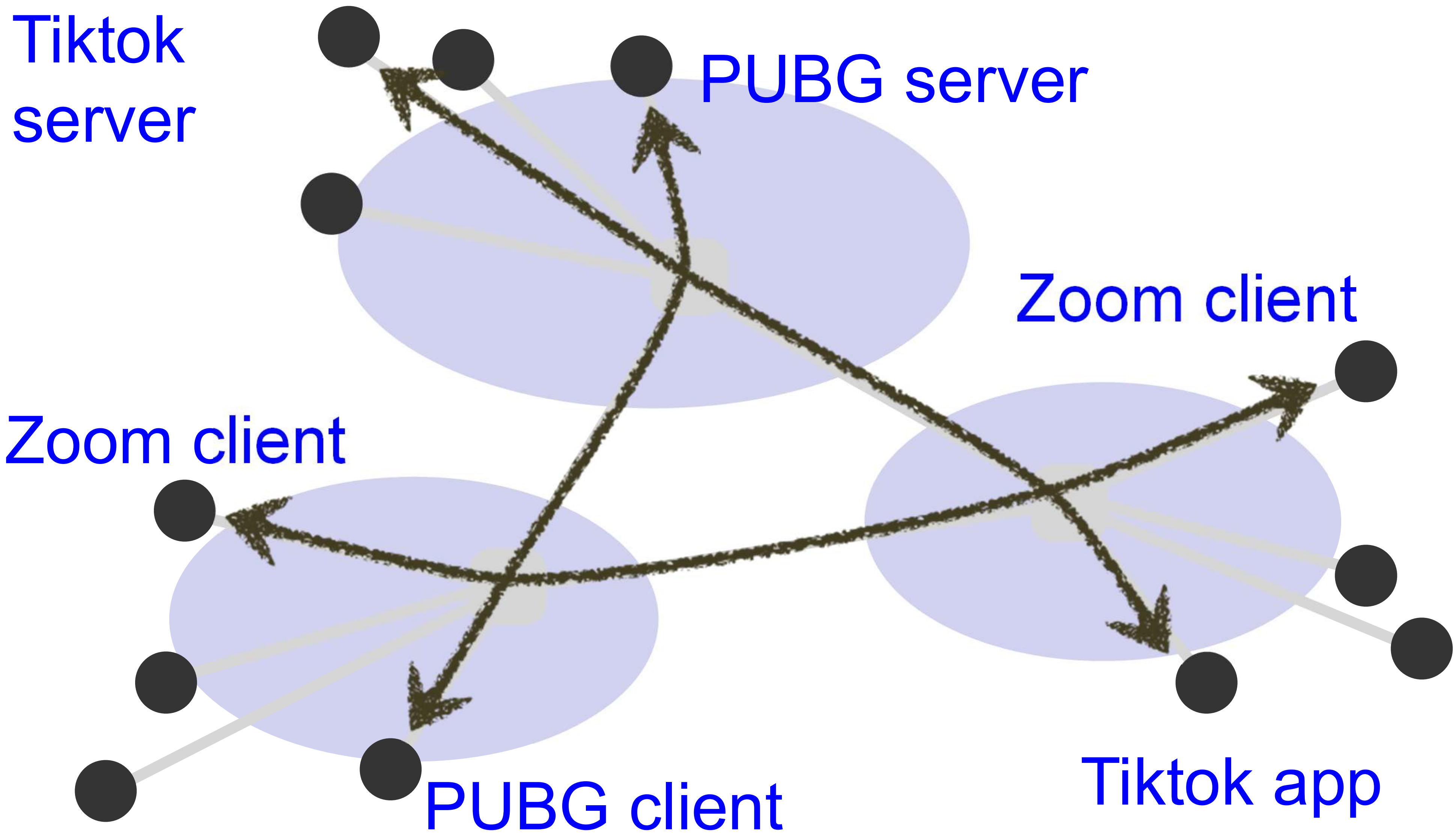
Switched networks

- End-systems and networks connected by switches instead of directly connecting them
  - Why?
- Allows us to scale
  - For example, directly connecting N nodes to each other would require N^2 links!

When do we need to share the network?



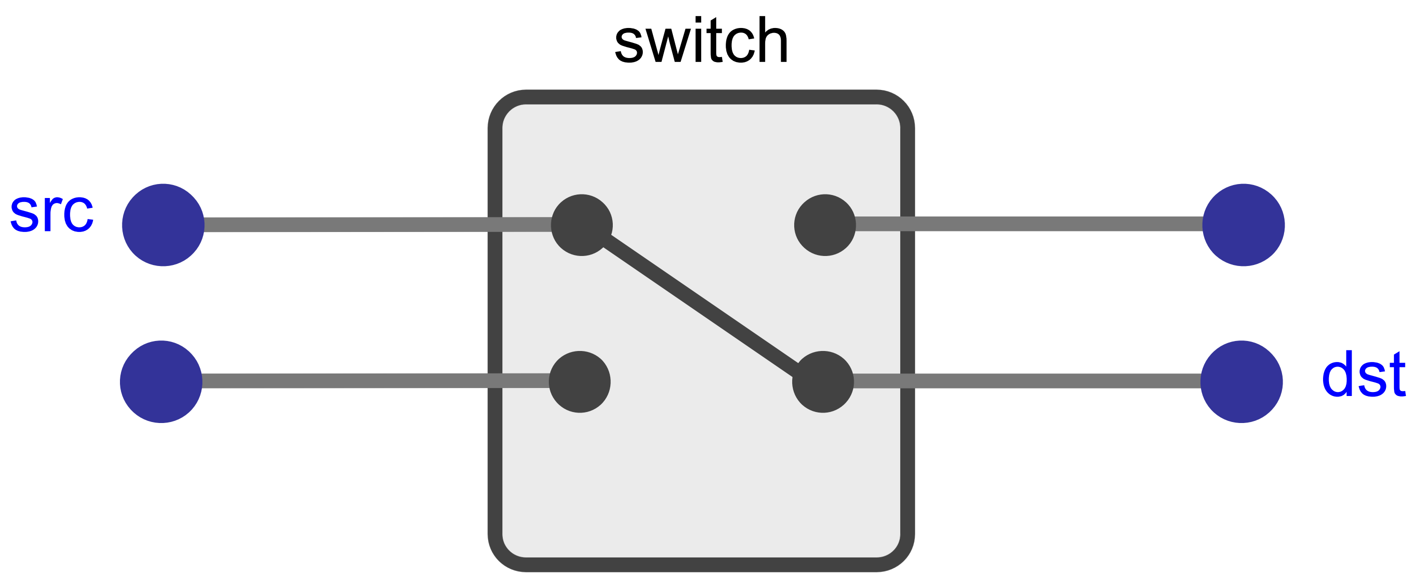
Shared among many services



Two ways to share switched networks

- Circuit switching
  - Resource reserved per connection
  - Admission control: per connection
- Packet switching via statistical multiplexing
  - Packets treated independently, on-demand
  - Admission control: per packet

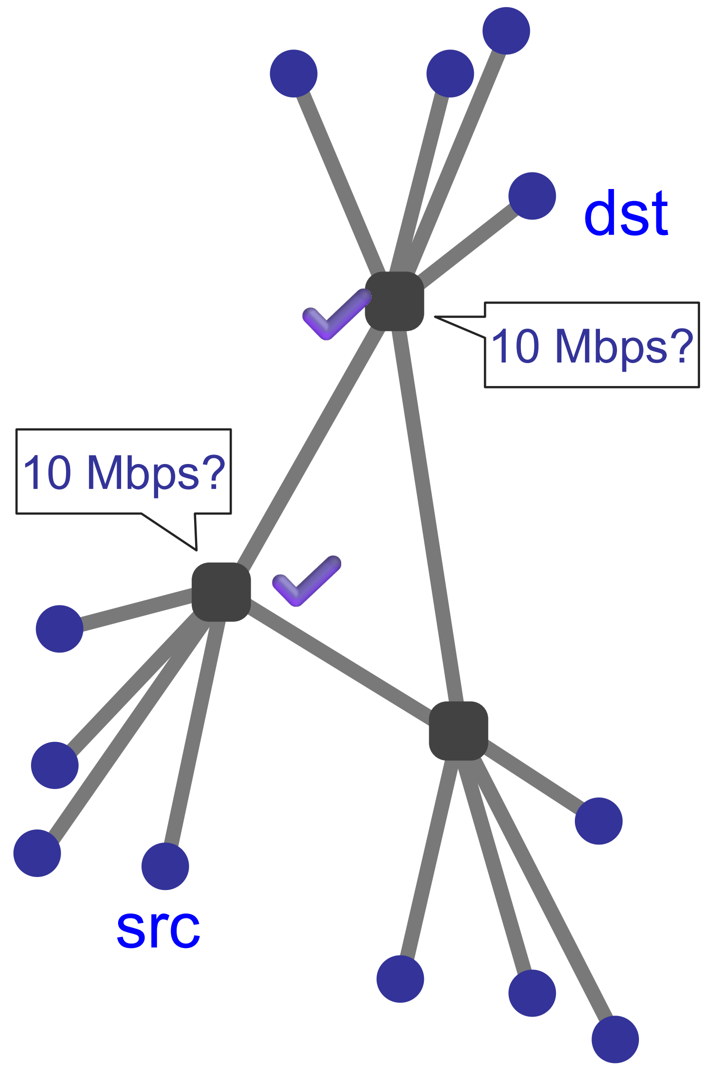
Circuit switching



- Reservation establishes a "circuit" within a switch

Circuit switching

1. src sends reservation request to dst
2. Switches create circuit *after* admission control
3. src sends data
4. src sends teardown request

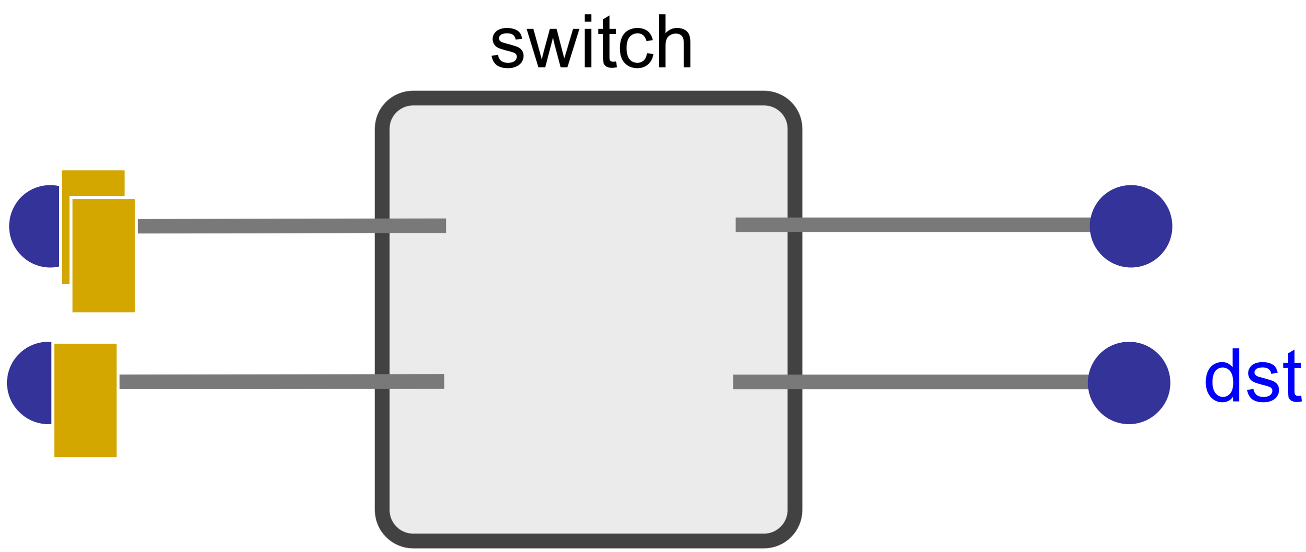


Circuit switching

- Pros
  - Predictable performance
  - Simple/fast switching (once circuit established)

- Cons
  - Complexity and delay from circuit setup/teardown
  - Inefficient when traffic is bursty
  - Switch fails → its circuit(s) fails

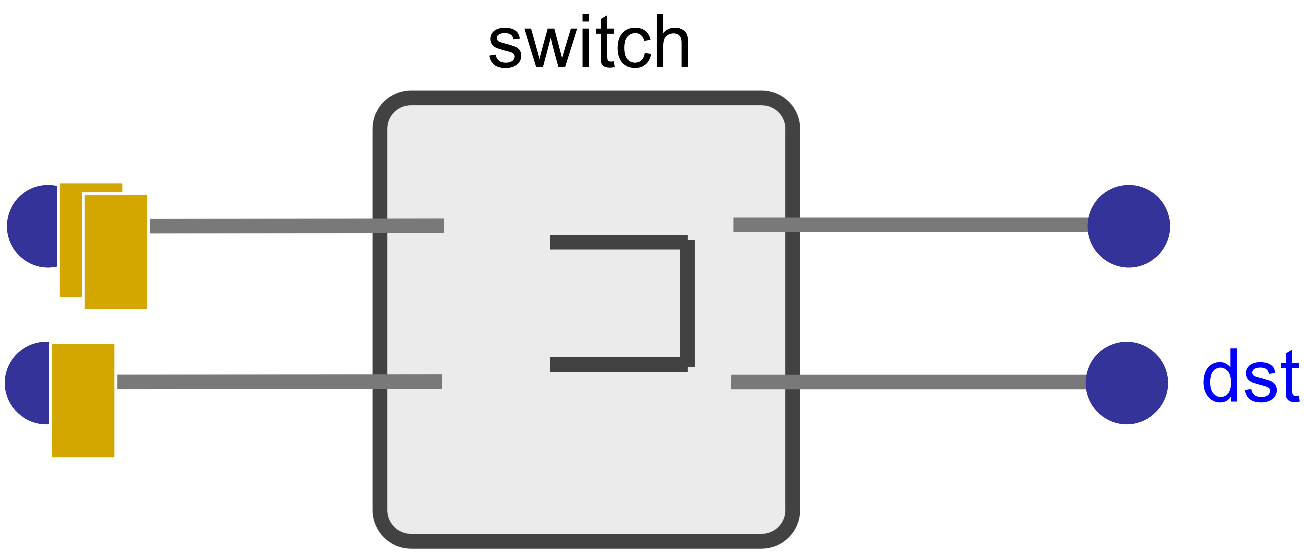
Packet switching



Each packet contains destination (dst)

Each packet treated independently

Packet switching



- Each packet contains destination (dst)
- Each packet treated independently
- With buffers to absolve transient overloads

Packet switching

- Pros
  - Efficient use of network resources
  - Simpler to implement
  - Robust: can “route around trouble”

- Cons
  - Unpredictable performance
  - Requires buffer management and congestion control

Statistical multiplexing

- Allowing more demands than the network can handle
  - Hoping that not all demands are required at the same time
  - Results in unpredictability
  - Works well except for the extreme cases

MASSIVE Scale

5+ Billion users

>1.1 Billion websites

~350 Billion emails sent per day

>6.9 Billion smartphones

>3 Billion monthly active Facebook users

>1 Billion hours of YouTube watched per day

…

5-MINUTE BREAK!

HOW DO WE EVALUATE A NETWORK?

Performance metrics

Delay

Loss

Throughput

Delay

- How long does it take to send a packet from its source to destination?

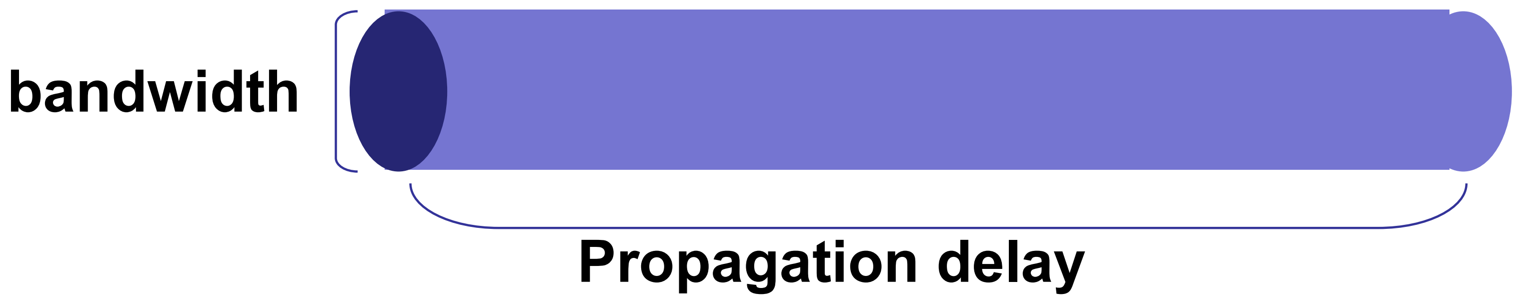
Delay

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

due to link properties

due to traffic mix and switch internals

A network link



Link bandwidth

- Number of bits sent/received per unit time (bits/sec or bps)

Propagation delay

- Time for one bit to move through the link (seconds)

1. Transmission delay

How long does it take to push all the bits of a packet into a link?

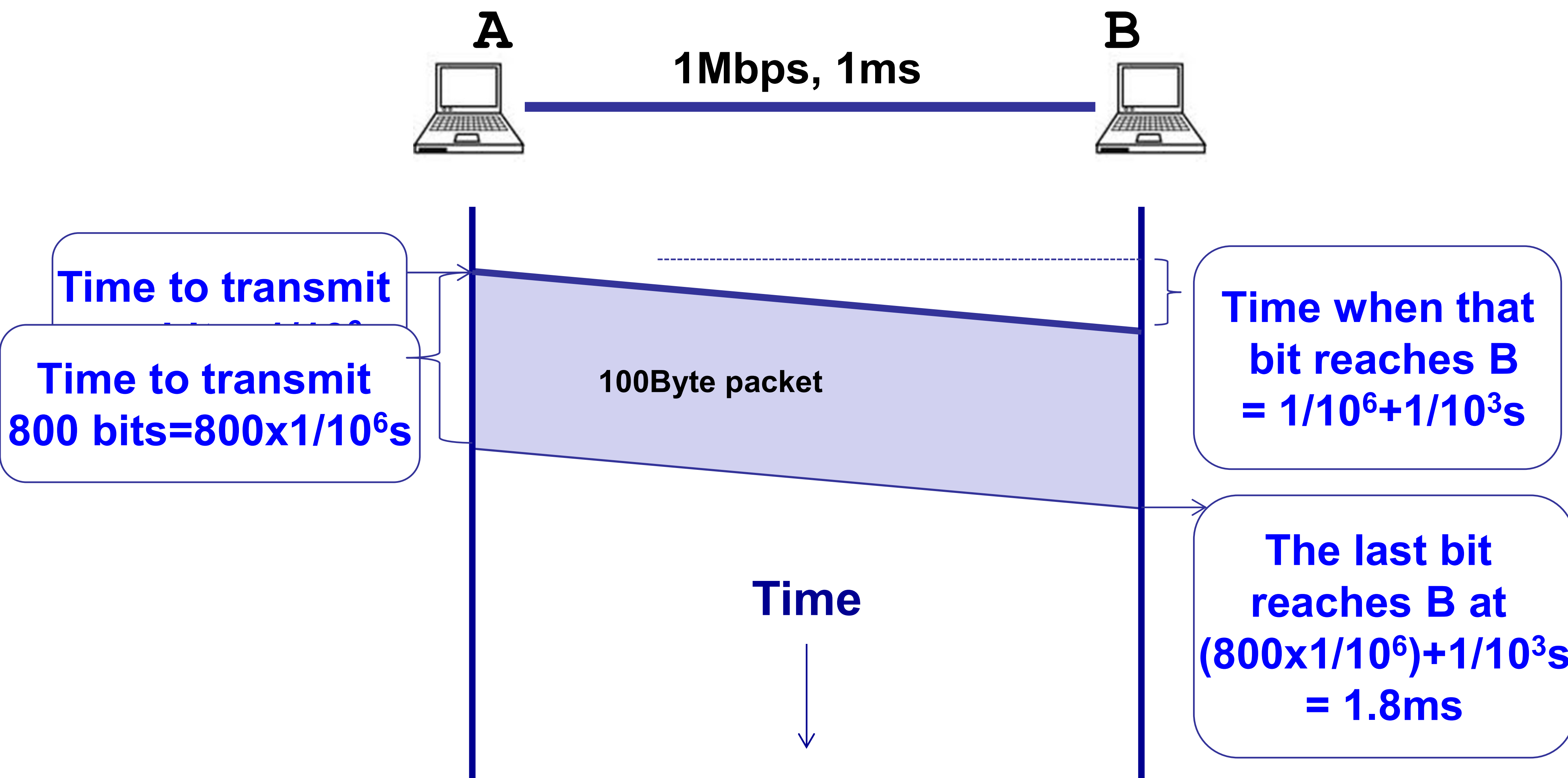
Packet size / Bandwidth of the link

- 1000 bits / 100 Mbits per sec = 10^{-5} sec
- 1000 bits / 1 Gbits per sec = 10^{-6} sec

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 = 10^{-4} sec

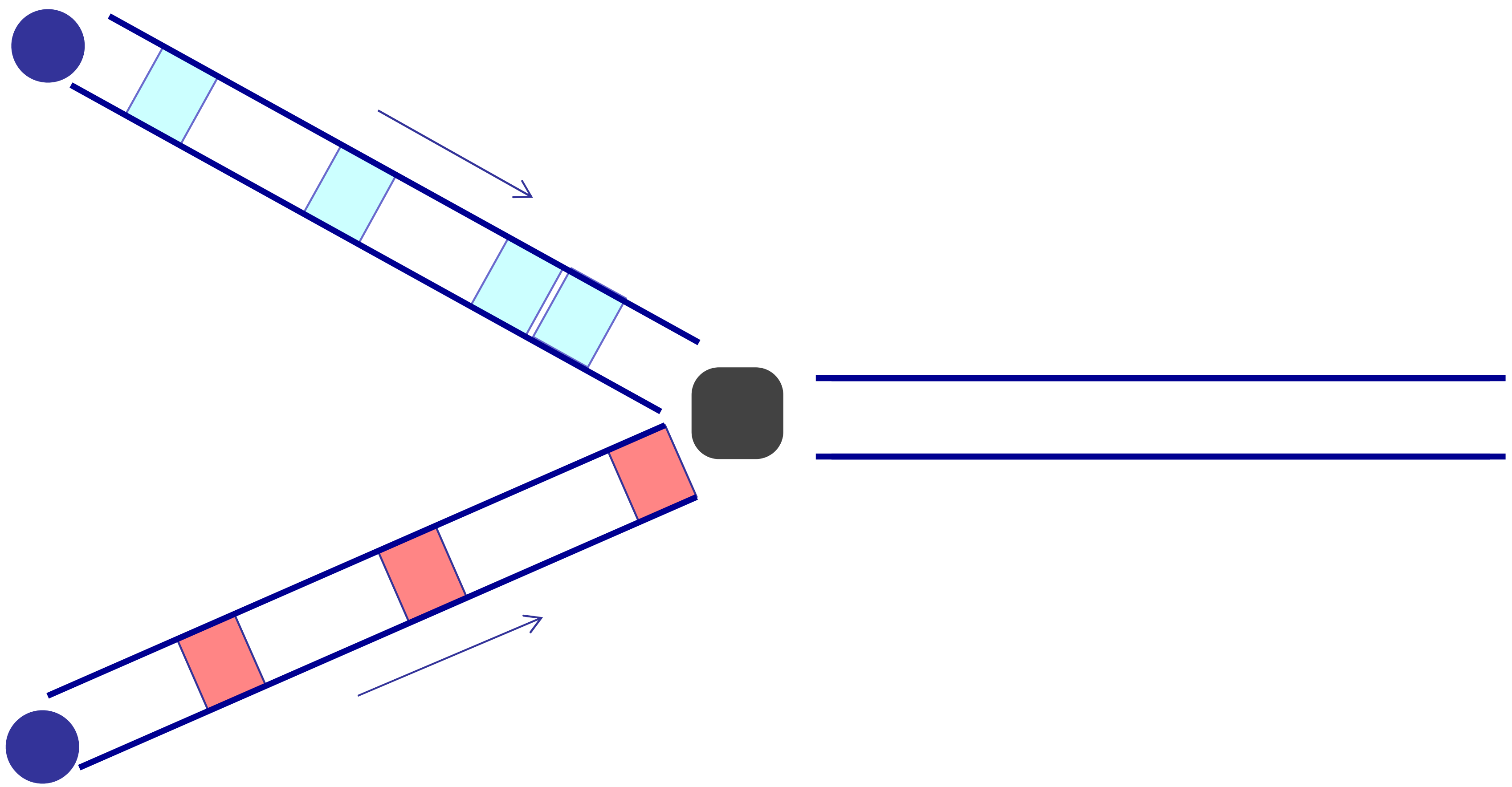
Packet delay
Sending a 100-byte packet



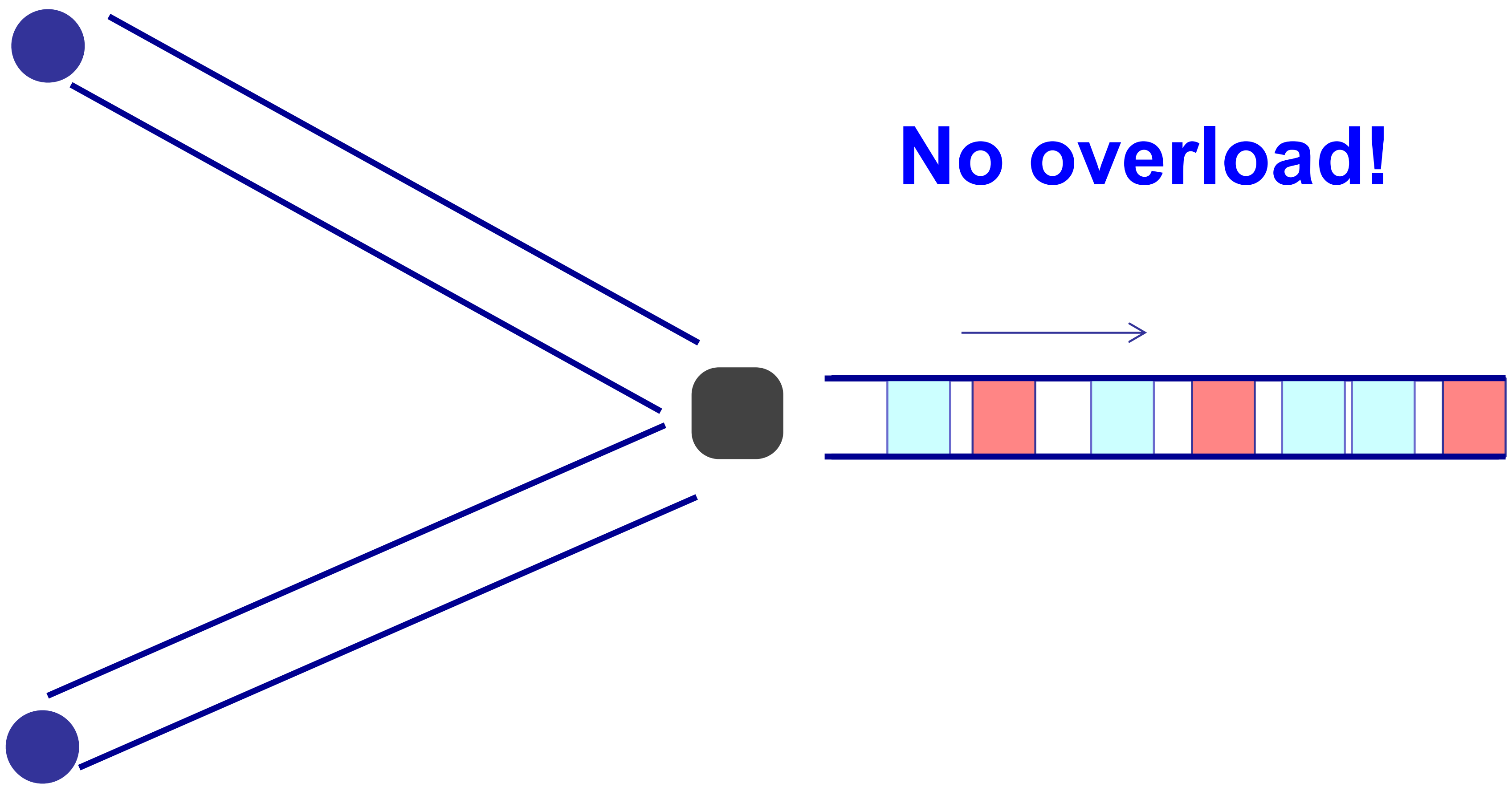
3. Queuing delay

How long does a packet have to sit in a buffer before it is processed?

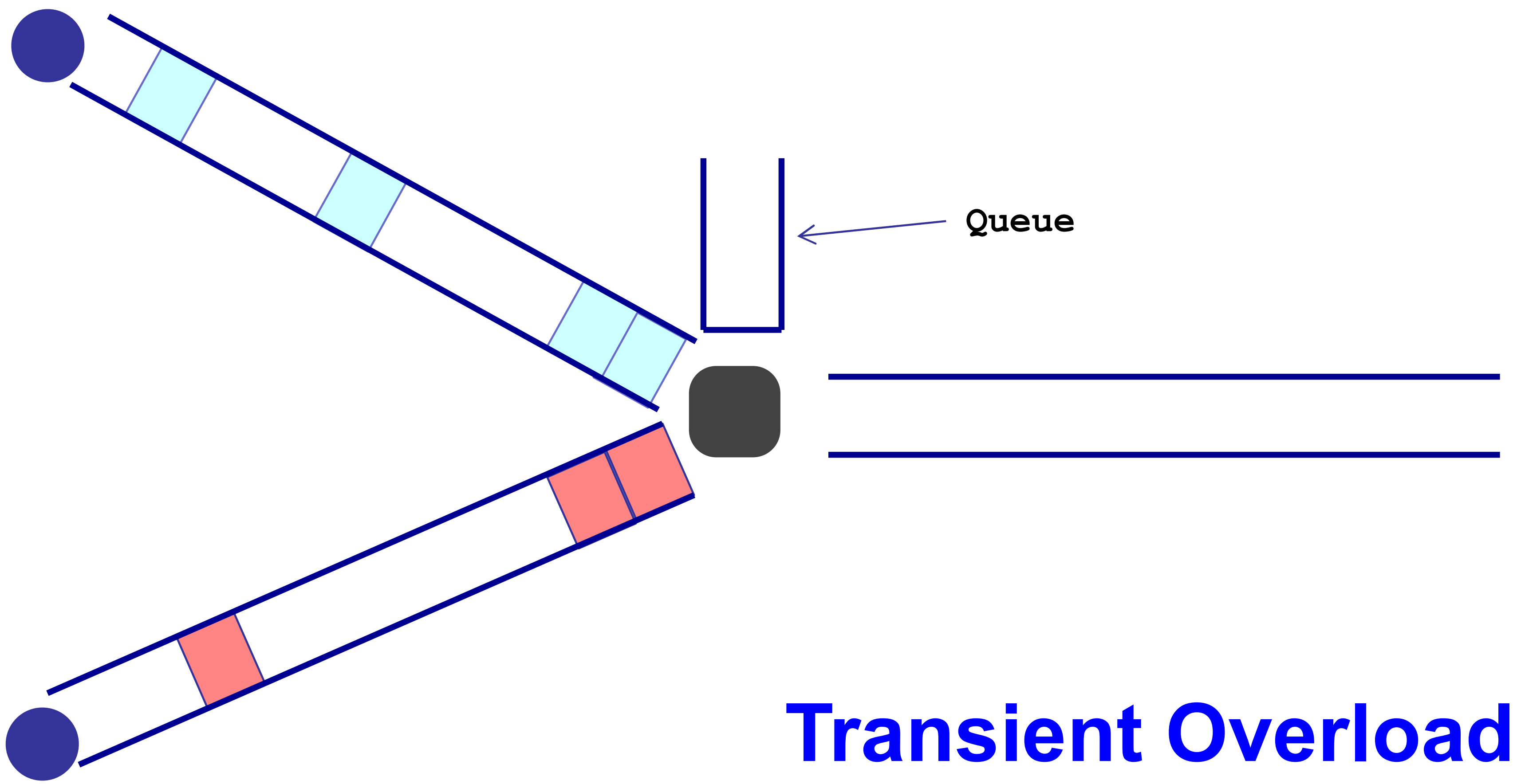
Queueing delay: “pipe” view



Queueing delay: "pipe" view



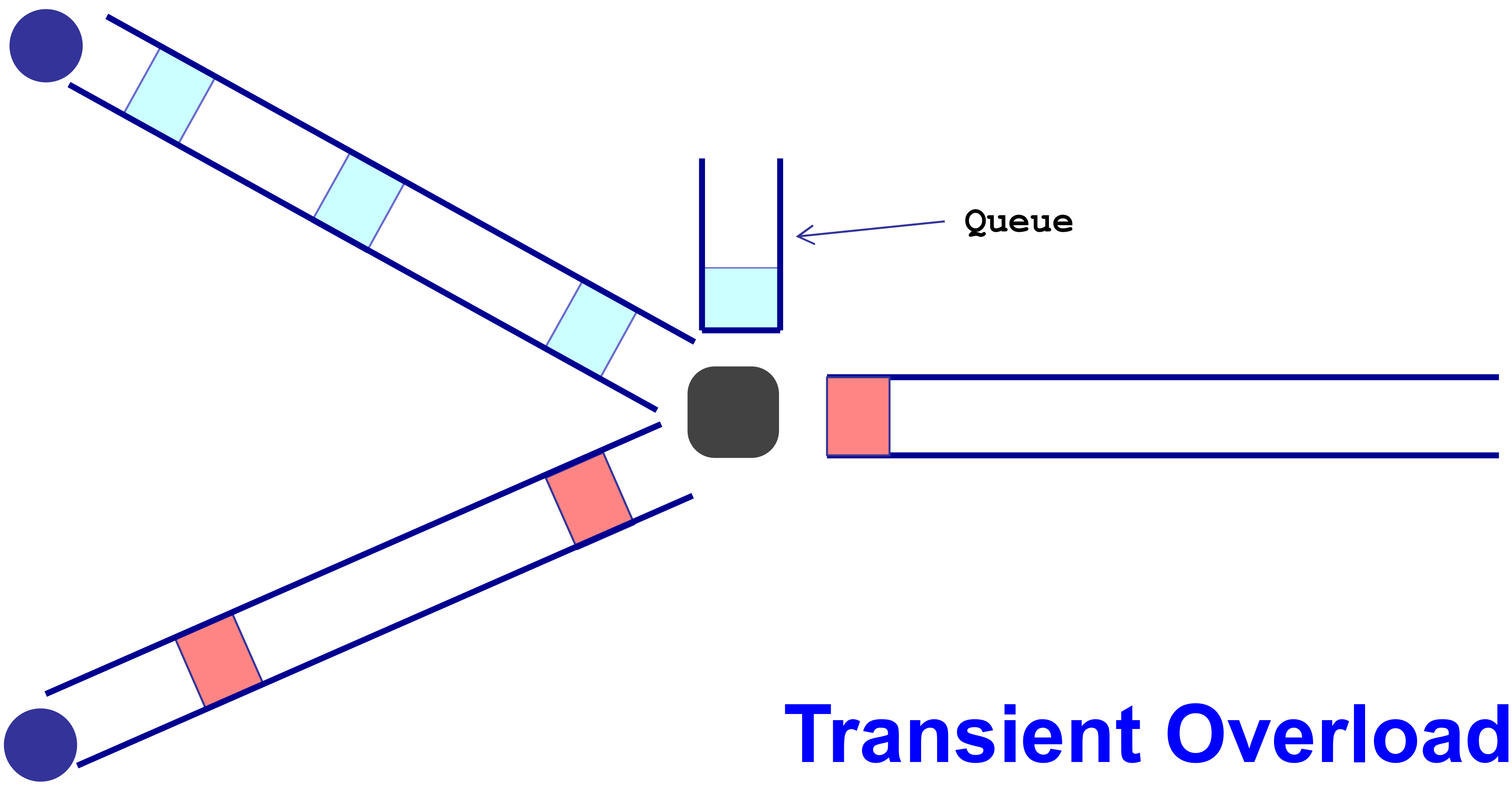
Queueing delay: “pipe” view



Transient Overload

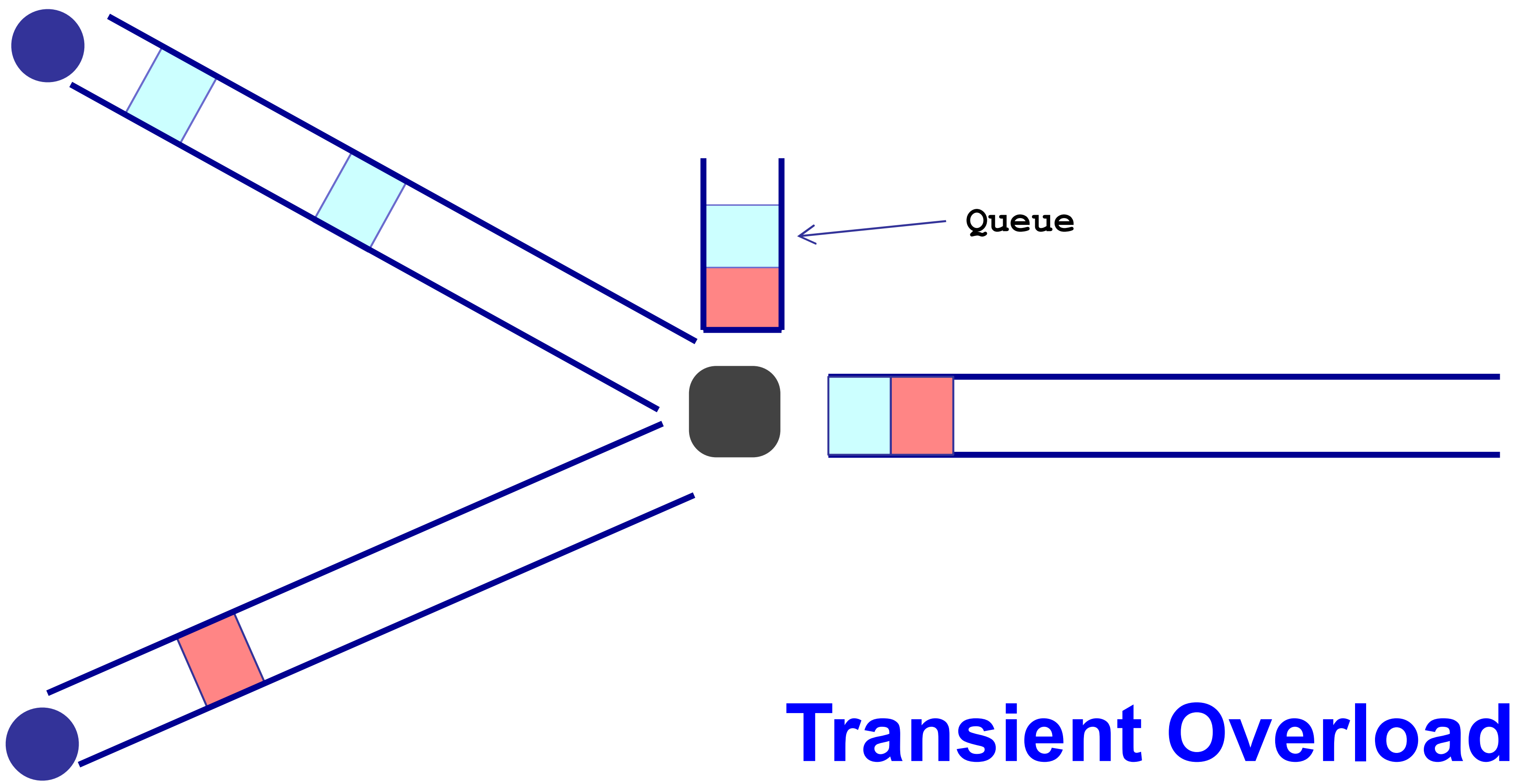
Not a rare event!

Queueing delay: "pipe" view



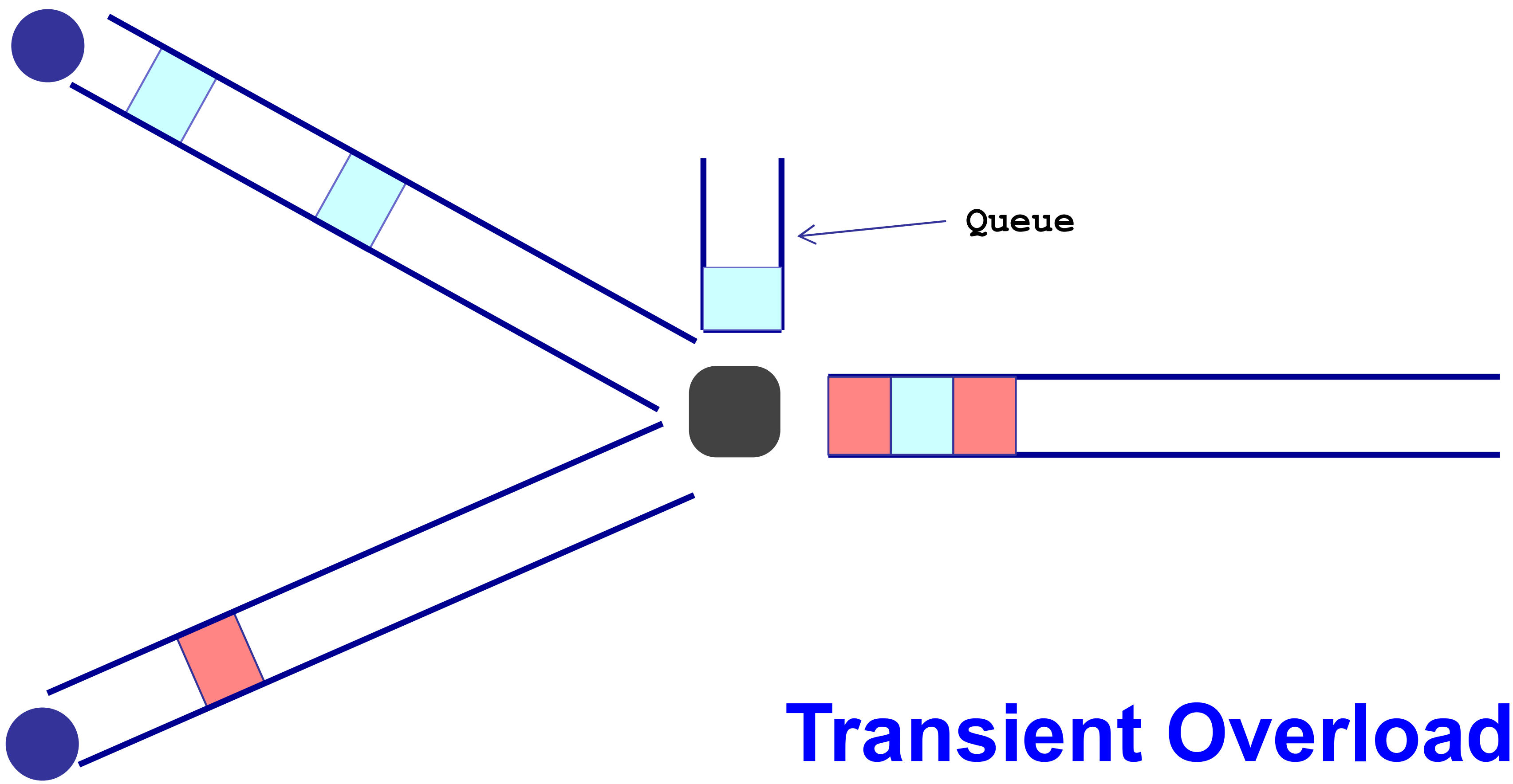
Transient Overload

Queueing delay: “pipe” view



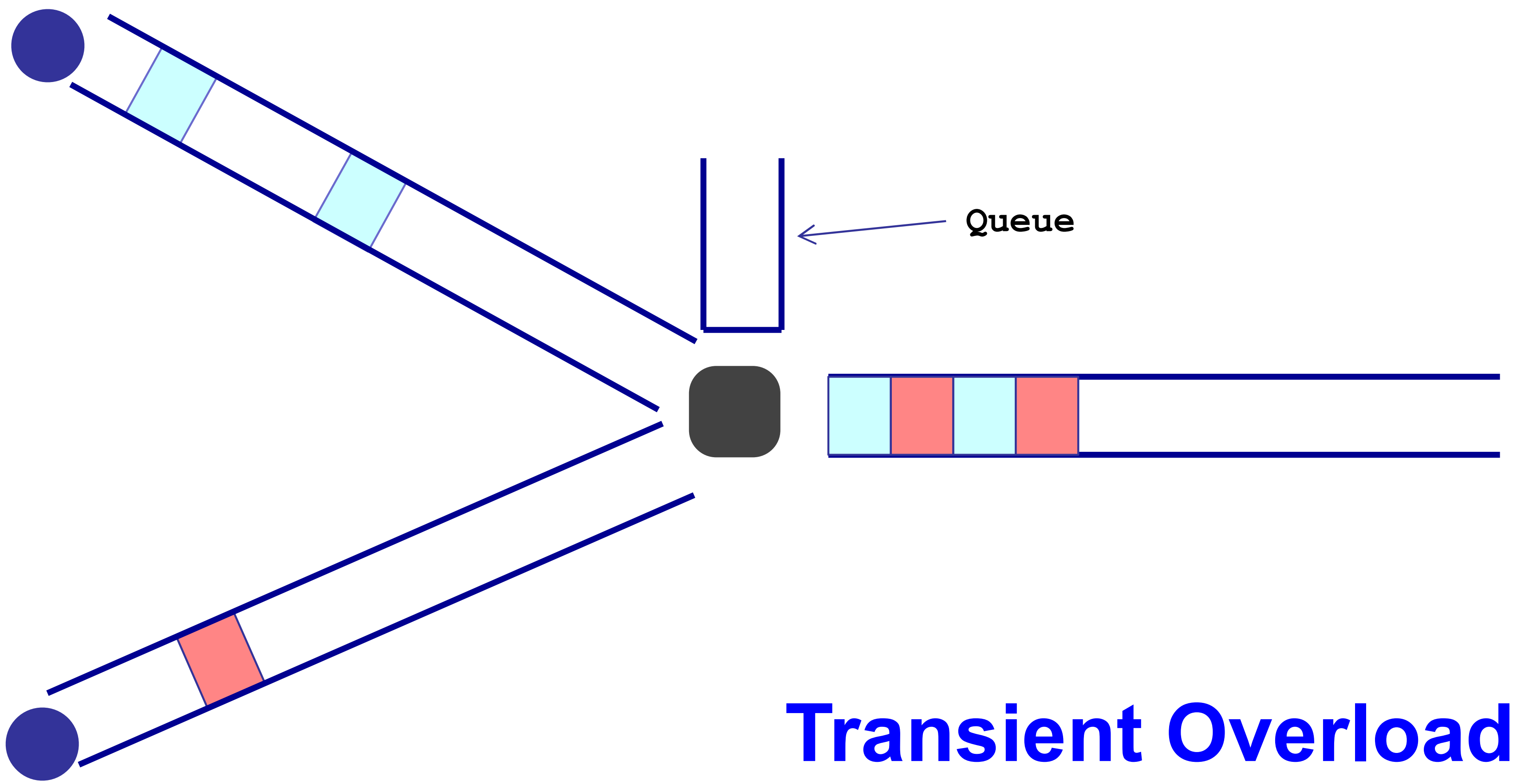
Transient Overload

Queueing delay: "pipe" view



Transient Overload

Queueing delay: “pipe” view



Transient Overload

Queueing delay

- How long does a packet have to sit in the buffer before it is processed?

- Depends on traffic pattern
  - Arrival rate at the queue
  - Nature of arriving traffic (bursty or not?)
  - Bandwidth of outgoing link

- Not easy to compute

Basic queueing theory terminology

- Arrival process: how packets arrive
  - Average rate A
- W: average time packets wait in the queue
  - W for “waiting time” (queuing delay)
- L: average number of packets waiting in the queue
  - L for “length of queue”

Little's Law (1961)

- L = A x W

- Compute L: count packets in queue every second

- Why do you care?
  - Easy to compute L, harder to compute W

4. Processing Delay

- How long does the switch take to process a packet?
  - Negligible

End-to-end delay

transmission

+propagation

+queueing
+processing

+transmission

+propagation

+queueing
+processing

+transmission

+propagation

Round Trip Time (RTT)

- Time for a packet to go from a source to a destination and to come back

- Why do we care?
  - Measuring delay is hard from one end

- RTT/2 equals *average* end-to-end delay
  - Why not exact?

Loss

- What fraction of the packets sent to a destination are dropped?

Throughput

At what rate is the destination receiving data from the source

Link Throughput

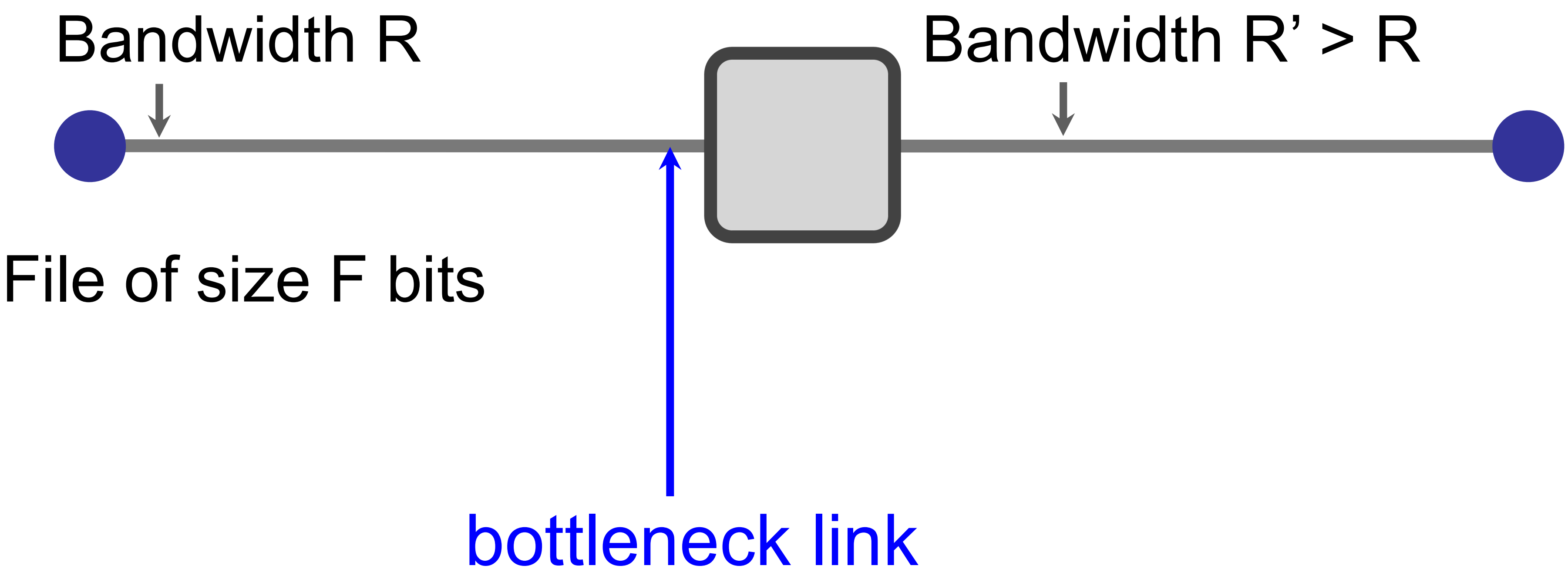
Bandwidth R bits/sec

File of size F bits

Transfer time (T) = F/R + propagation delay

Average throughput = F/T ≈ R

End-to-end throughput



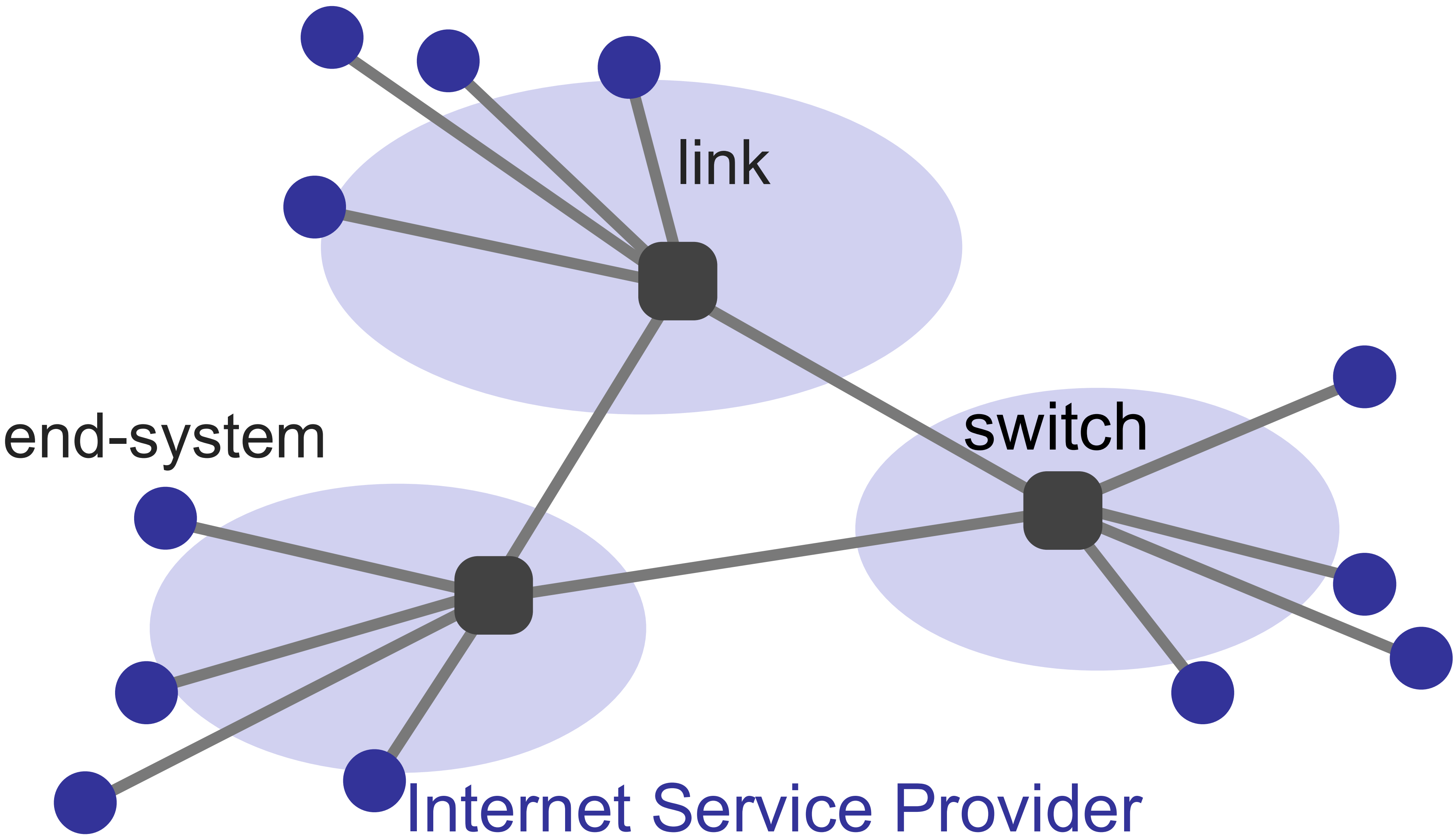
Average throughput = min{R, R’} = R

Summary

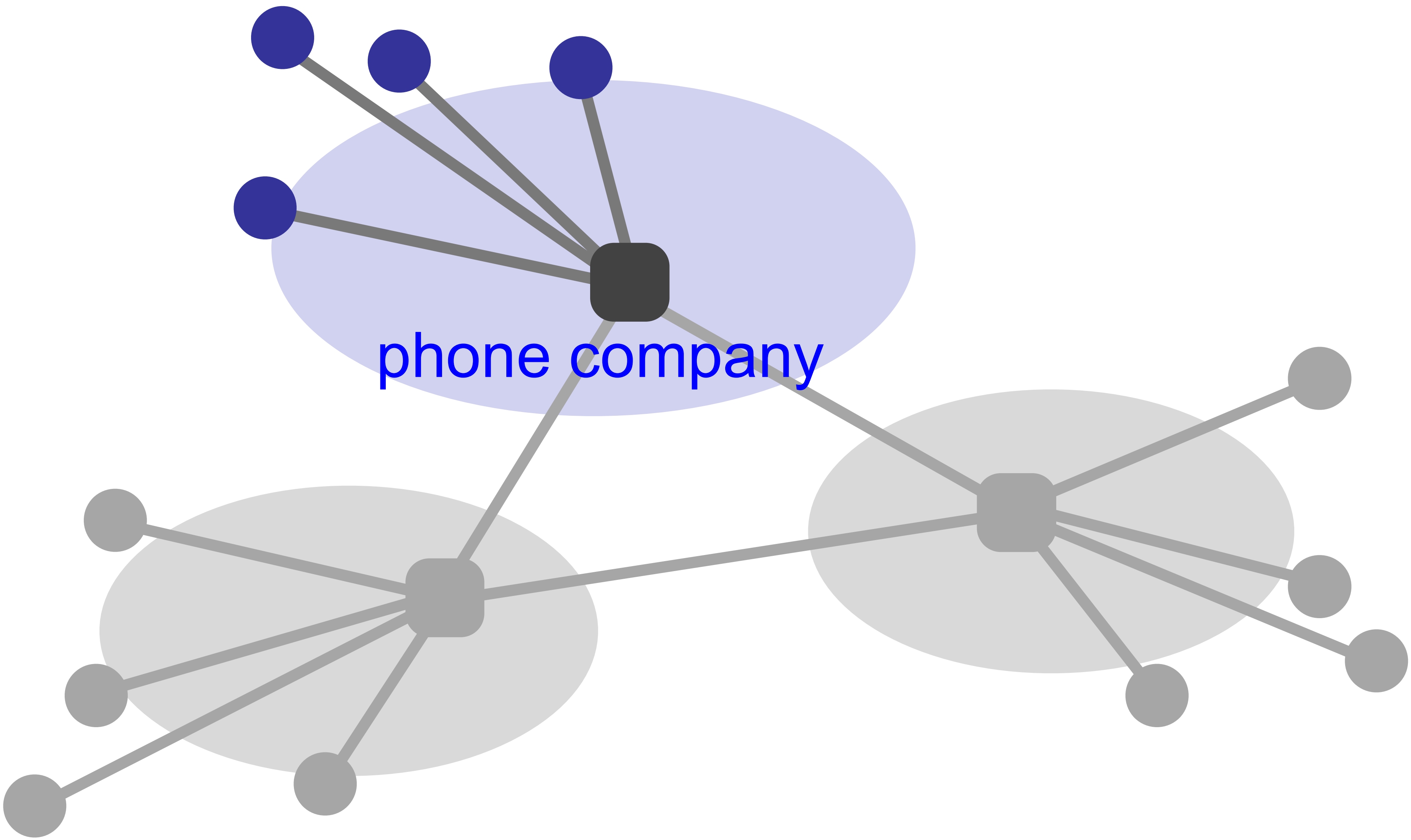
- How is the network shared?
  - On-demand or via reservation
- How do we evaluate a network?
  - Bandwidth, delay, loss, …

WHAT IS THE NETWORK MADE OF?

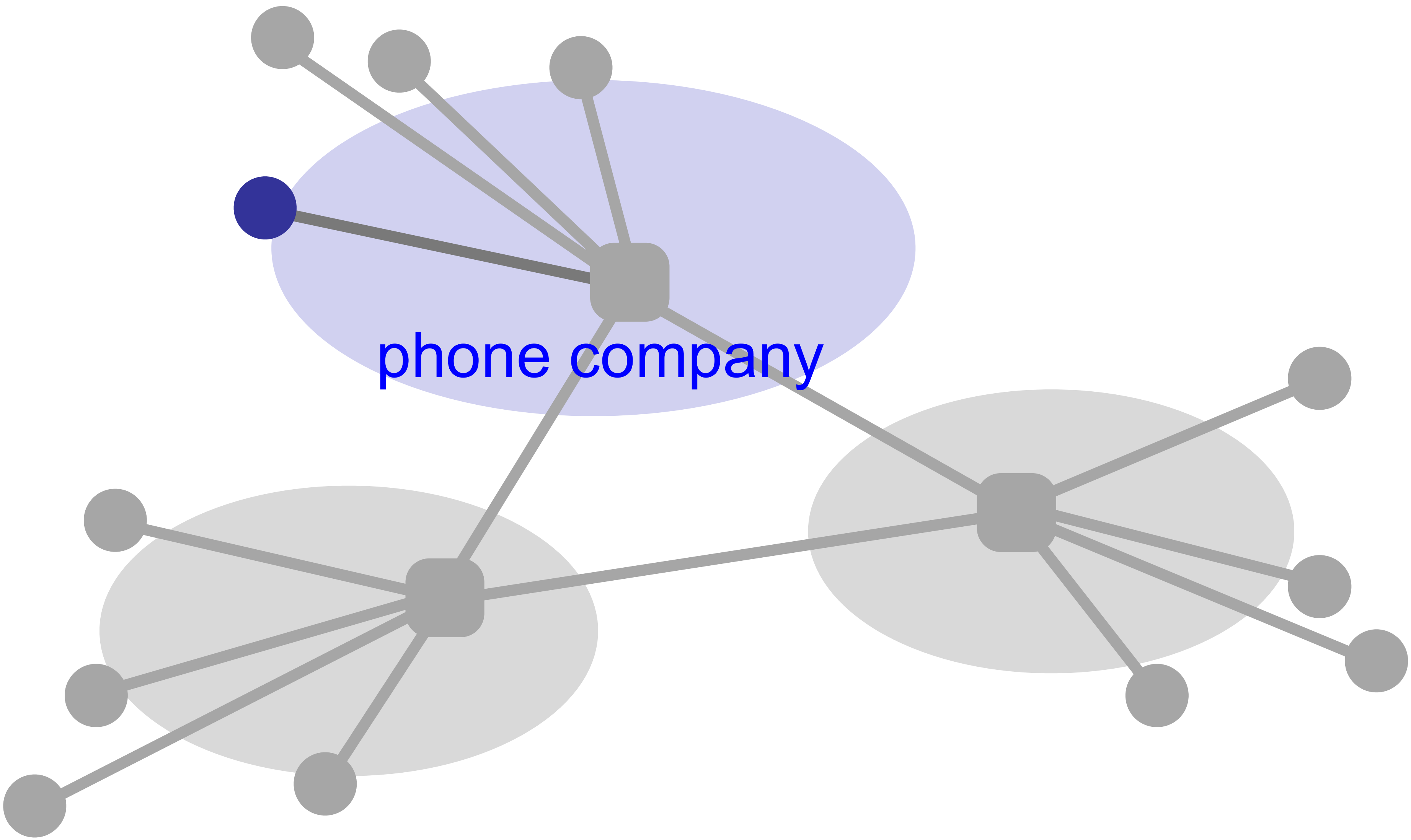
What is a network made of?



What is a network made of?

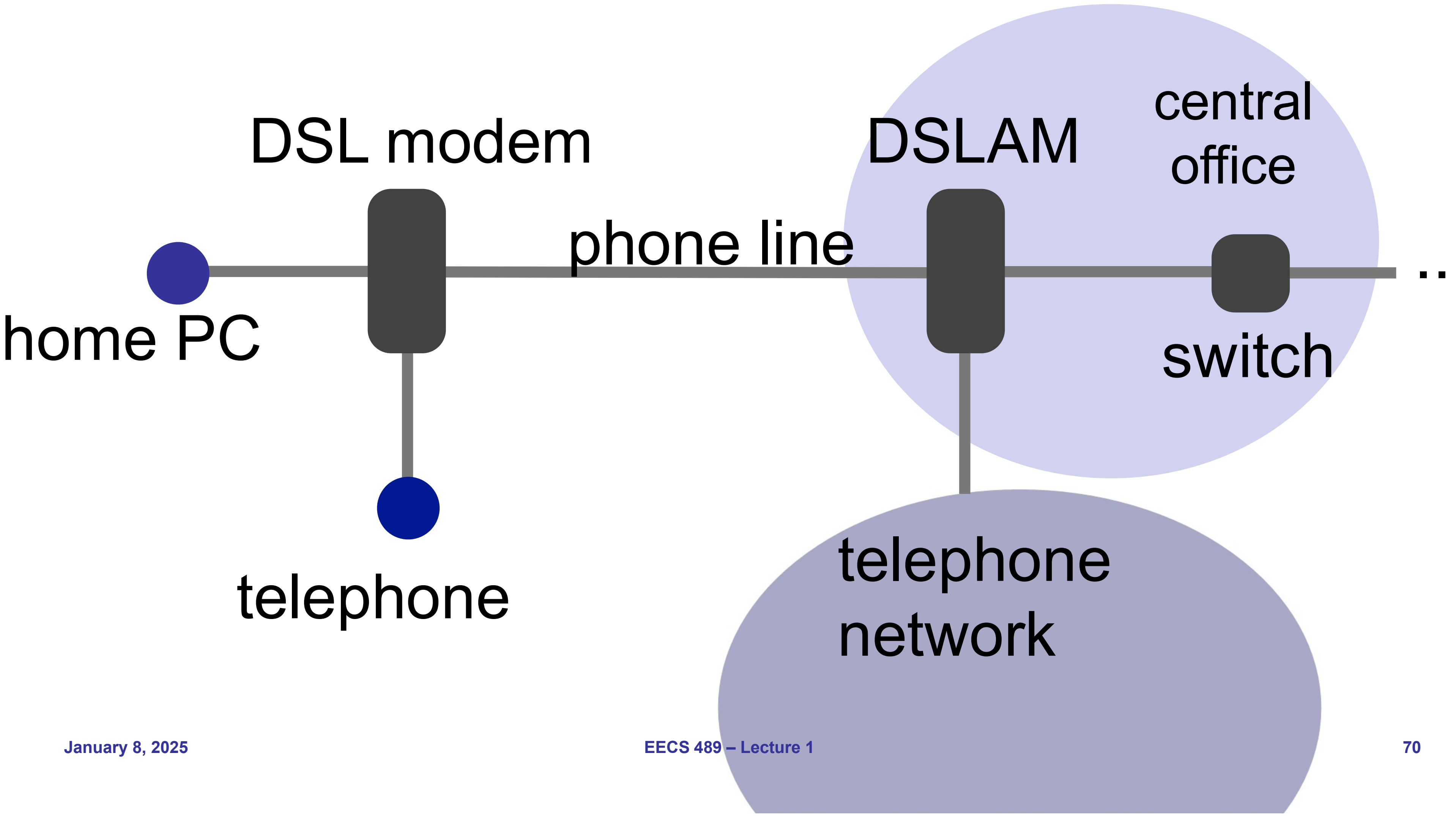


What is a network made of?



The last hop

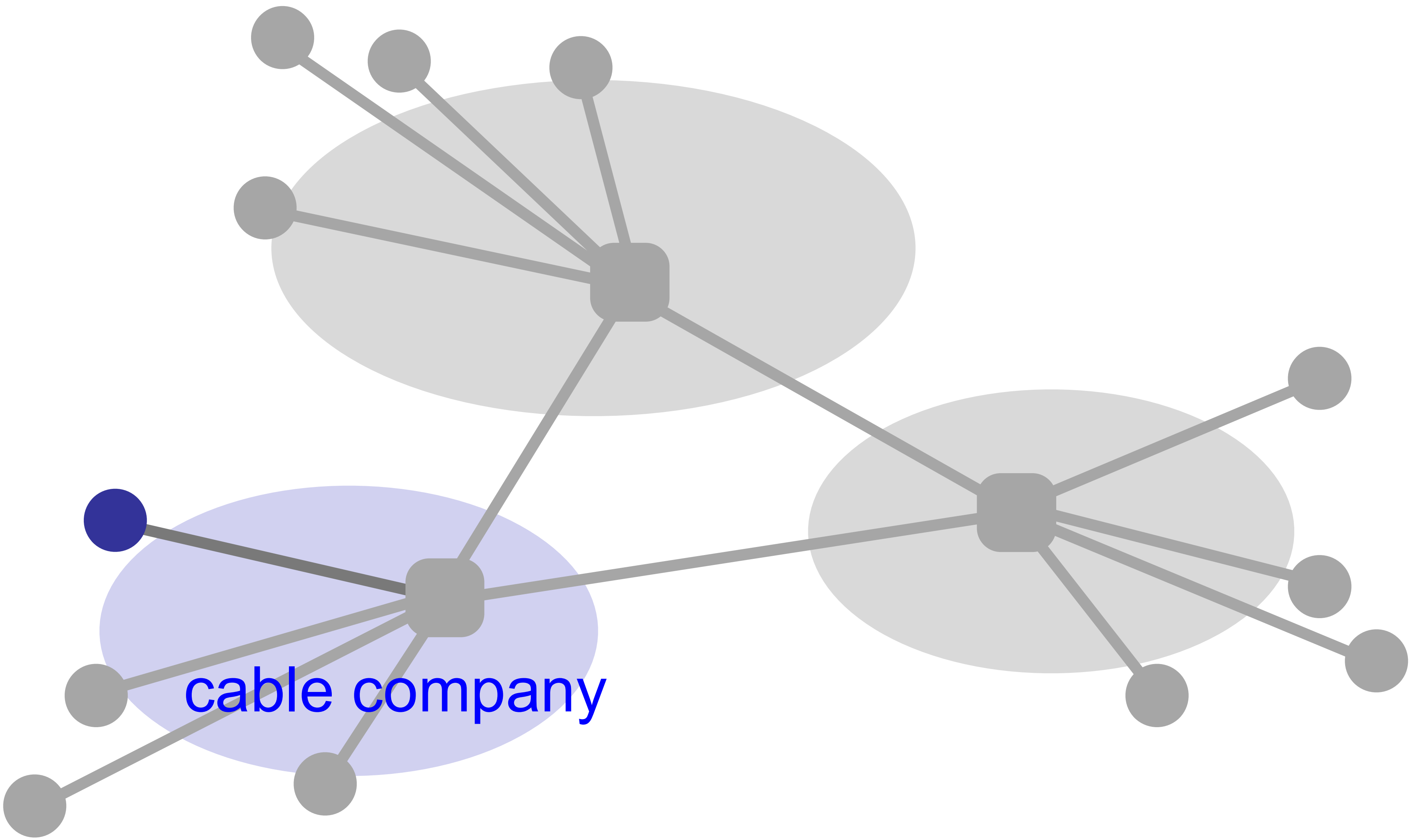
How do we connect?



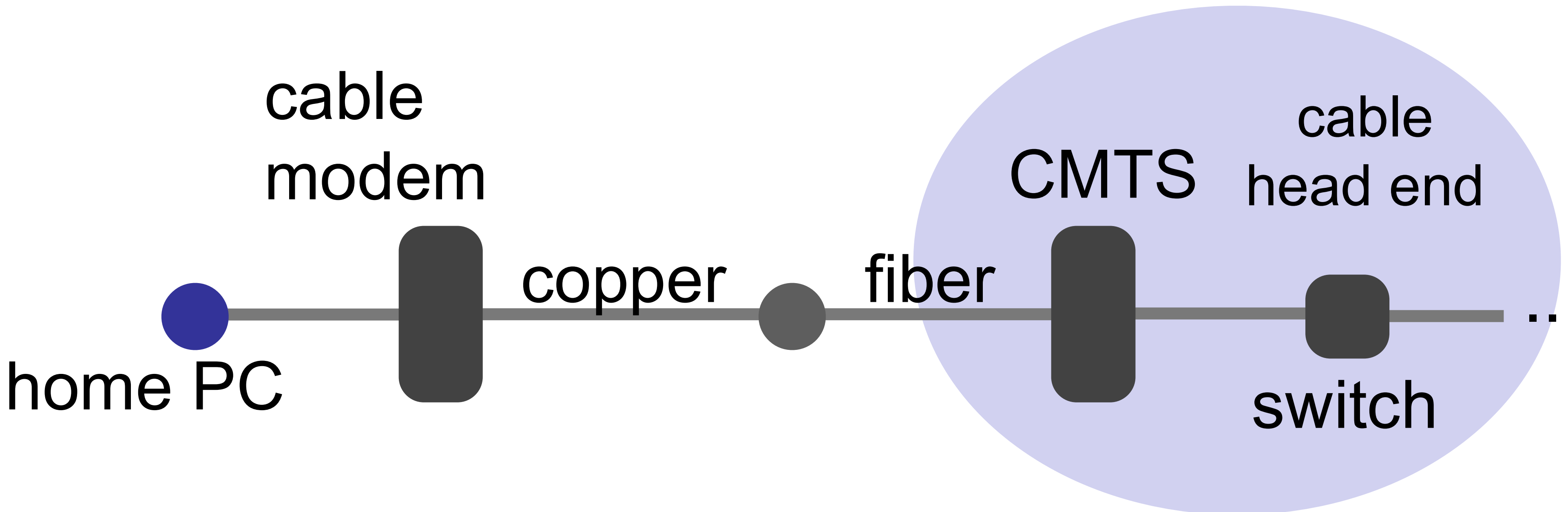
Digital Subscriber Line (DSL)

- Twisted pair copper
- 3 separate channels
  - downstream data channel
  - upstream data channel
  - 2-way phone channel
- up to 25 Mbps downstream
- up to 2.5 Mbps upstream

How about an cable provider as an ISP?



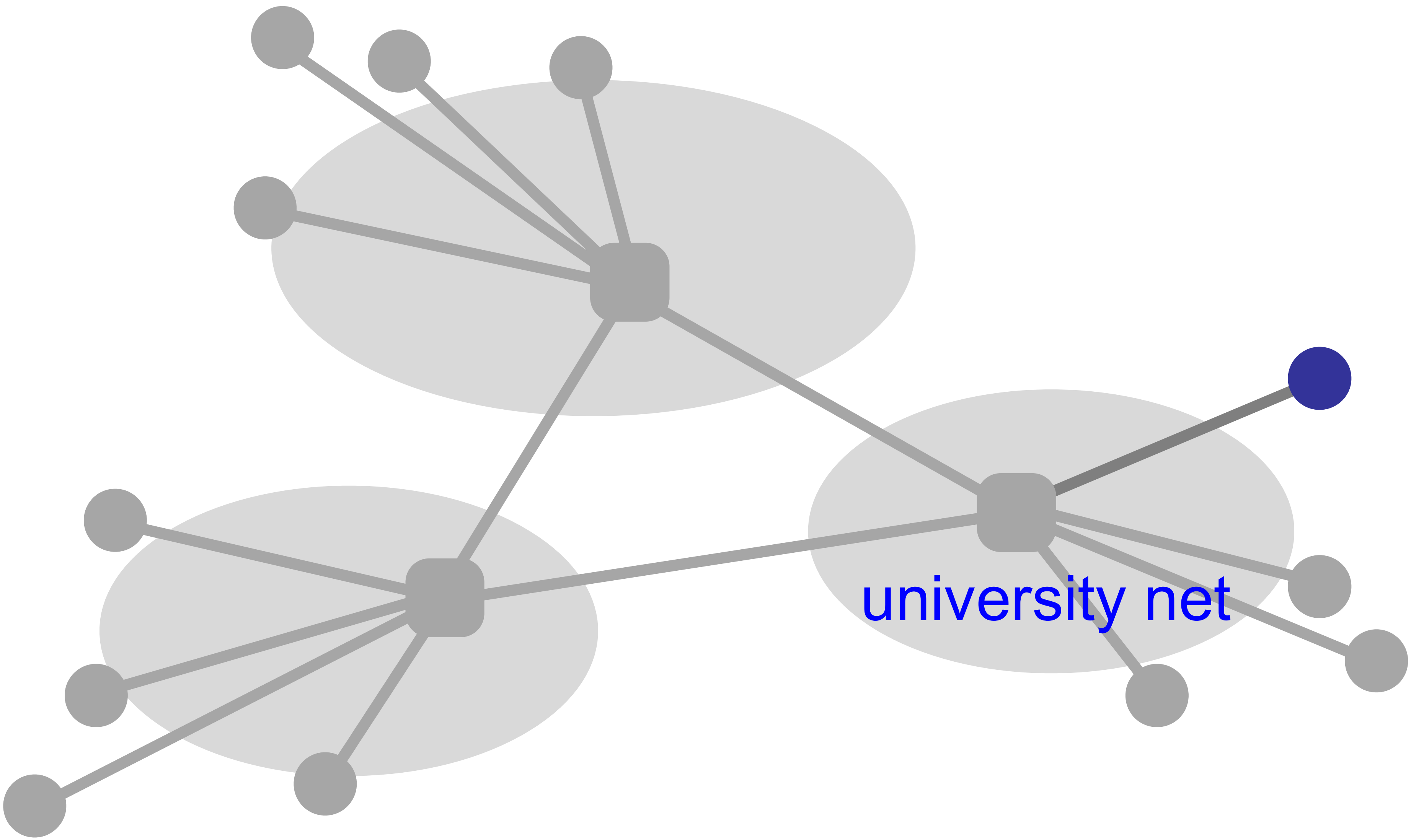
Connecting via cable



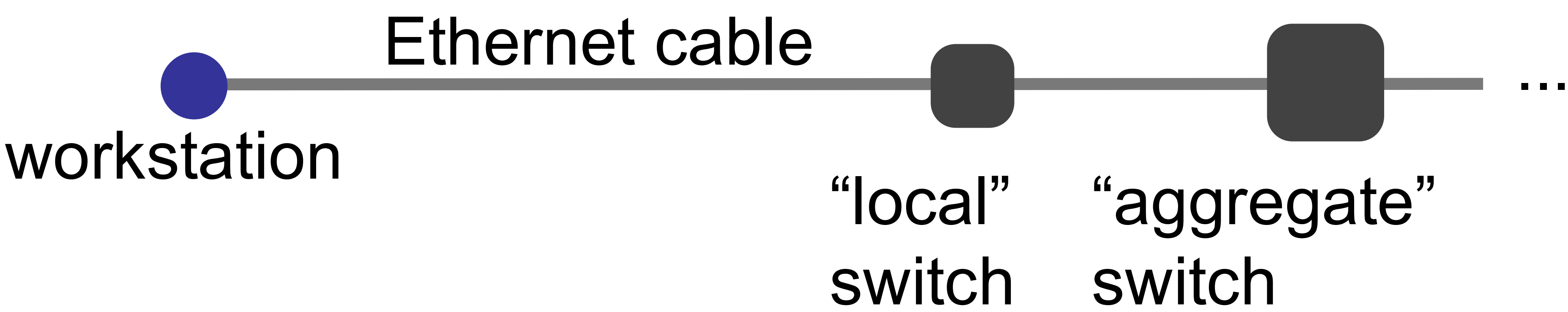
Cable

- Coaxial copper & fiber
- Up to 42.8 Mbps downstream
- Up to 30.7 Mbps upstream
- Shared broadcast medium

Any other means?



Ethernet


Ethernet cable
workstation
"local" switch
"aggregate" switch
...

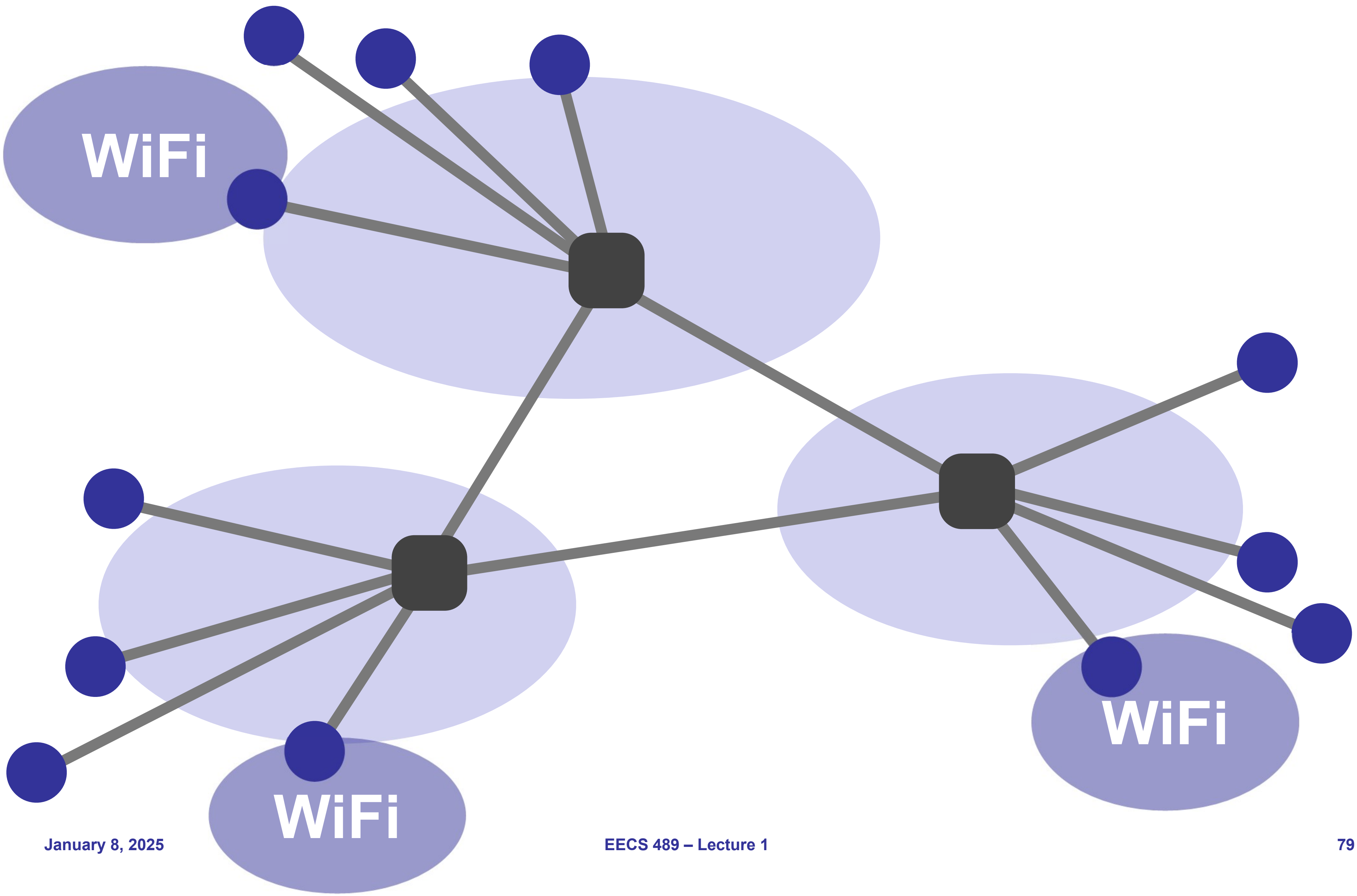
Ethernet

- Twisted pair copper
- 100 Mbps, 1 Gbps, 10 Gbps (each direction)

Many other ways

- Cellular (smart phones)
- Satellite (remote areas)
- Fiber to the Home (home)
- Optical carrier (Internet backbone)

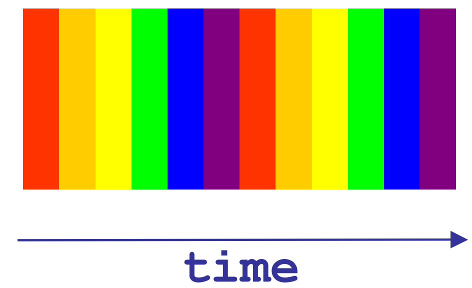
Where is WiFi?



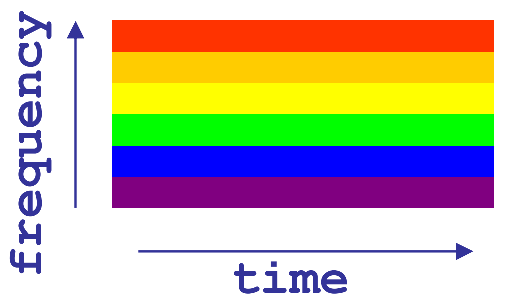
DETAILS ON CIRCUIT SWITCHING

Many kinds of circuits

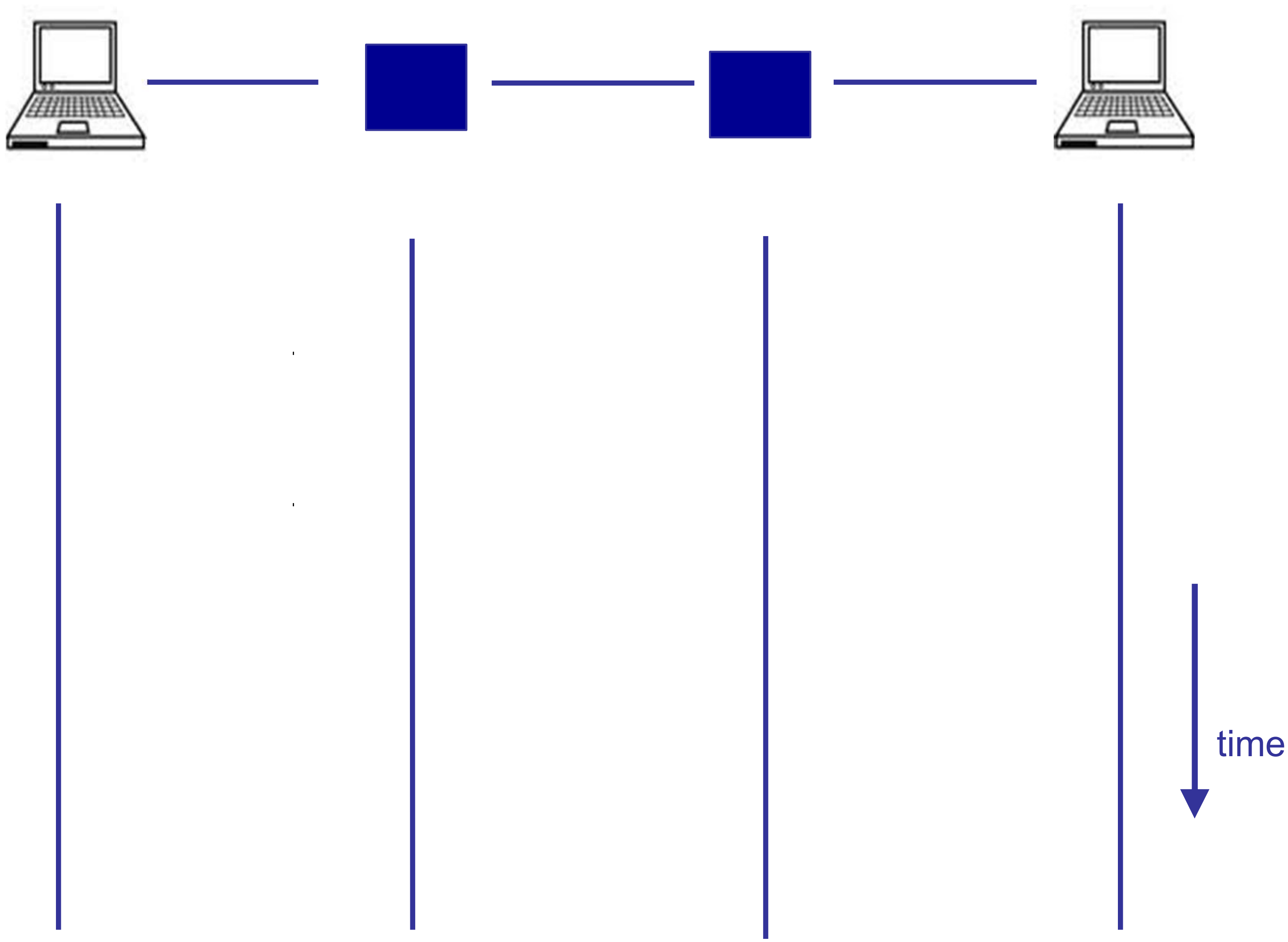
- Time division multiplexing
  - divide time in time slots
  - separate time slot per circuit



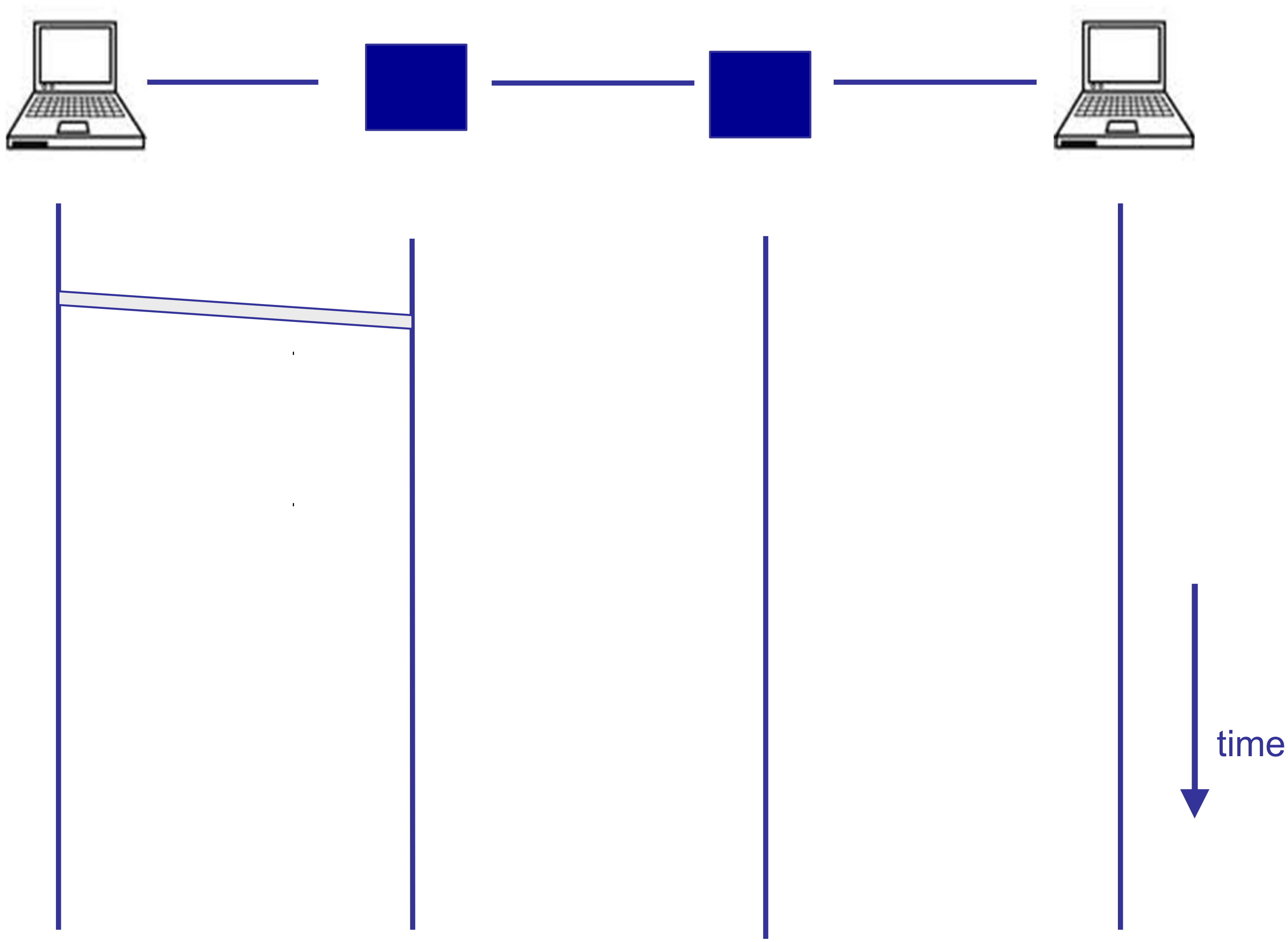
- Frequency division multiplexing
  - divide frequency spectrum in frequency bands
  - separate frequency band per circuit



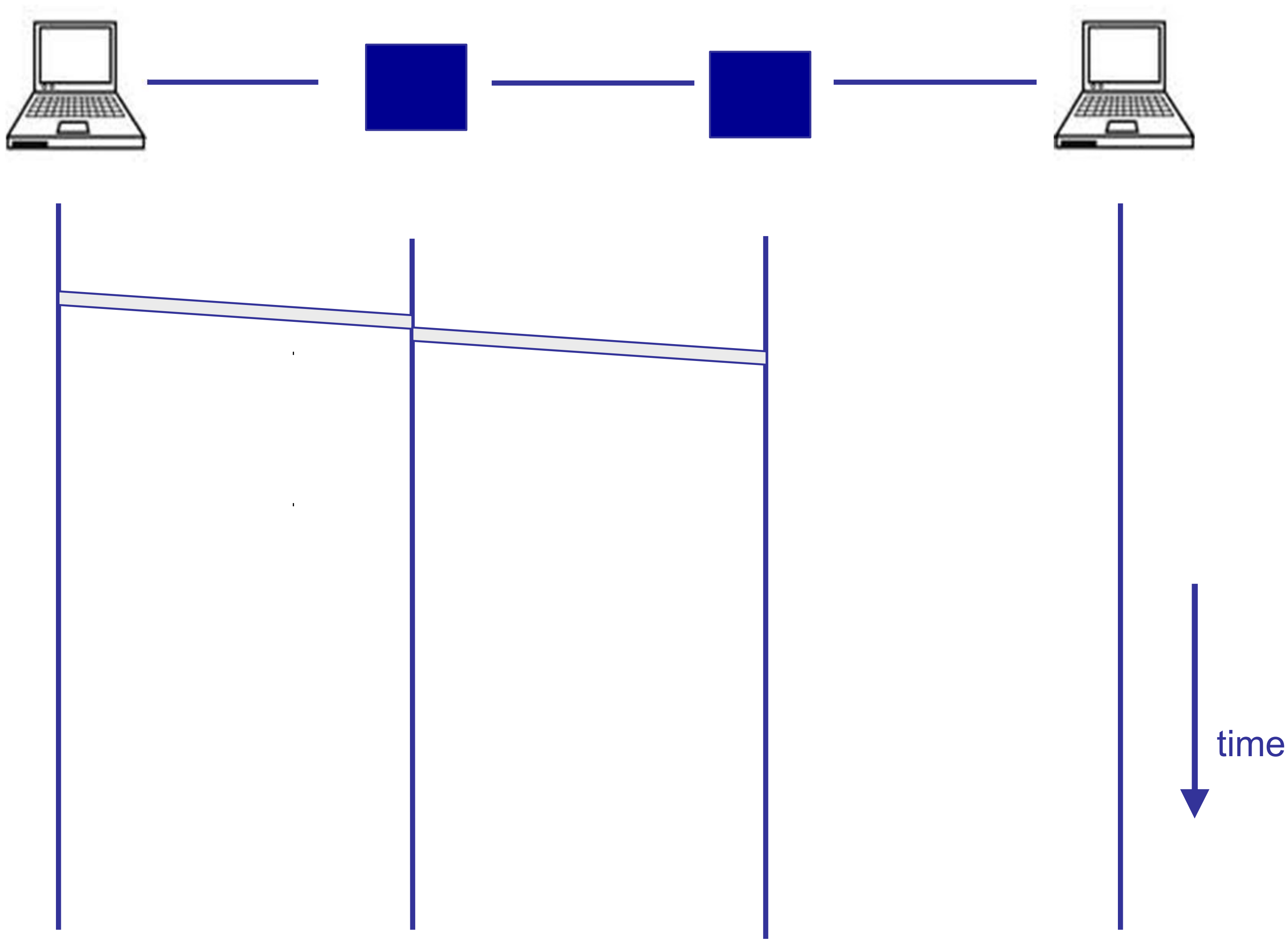
Timing in circuit switching



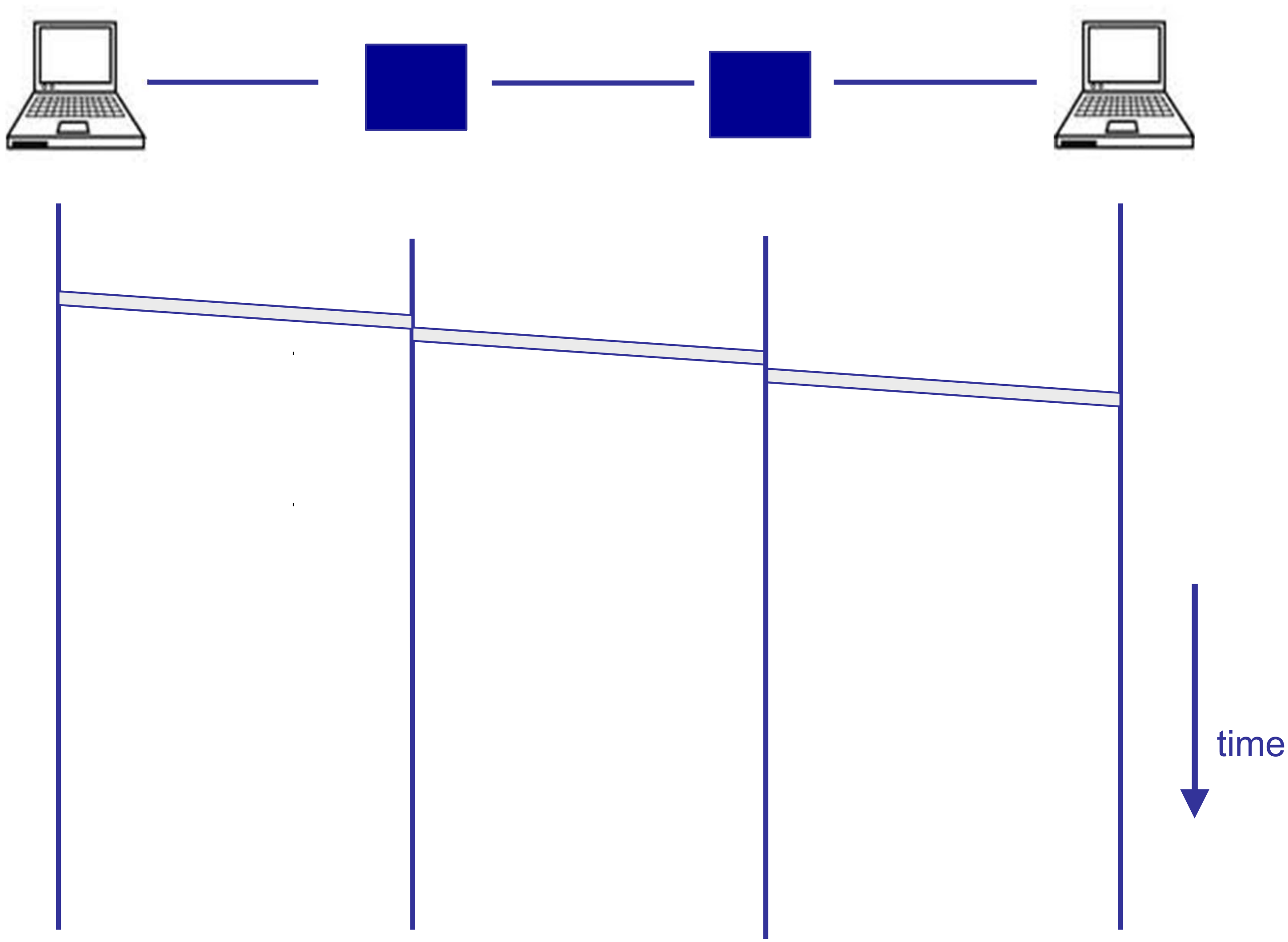
Timing in circuit switching



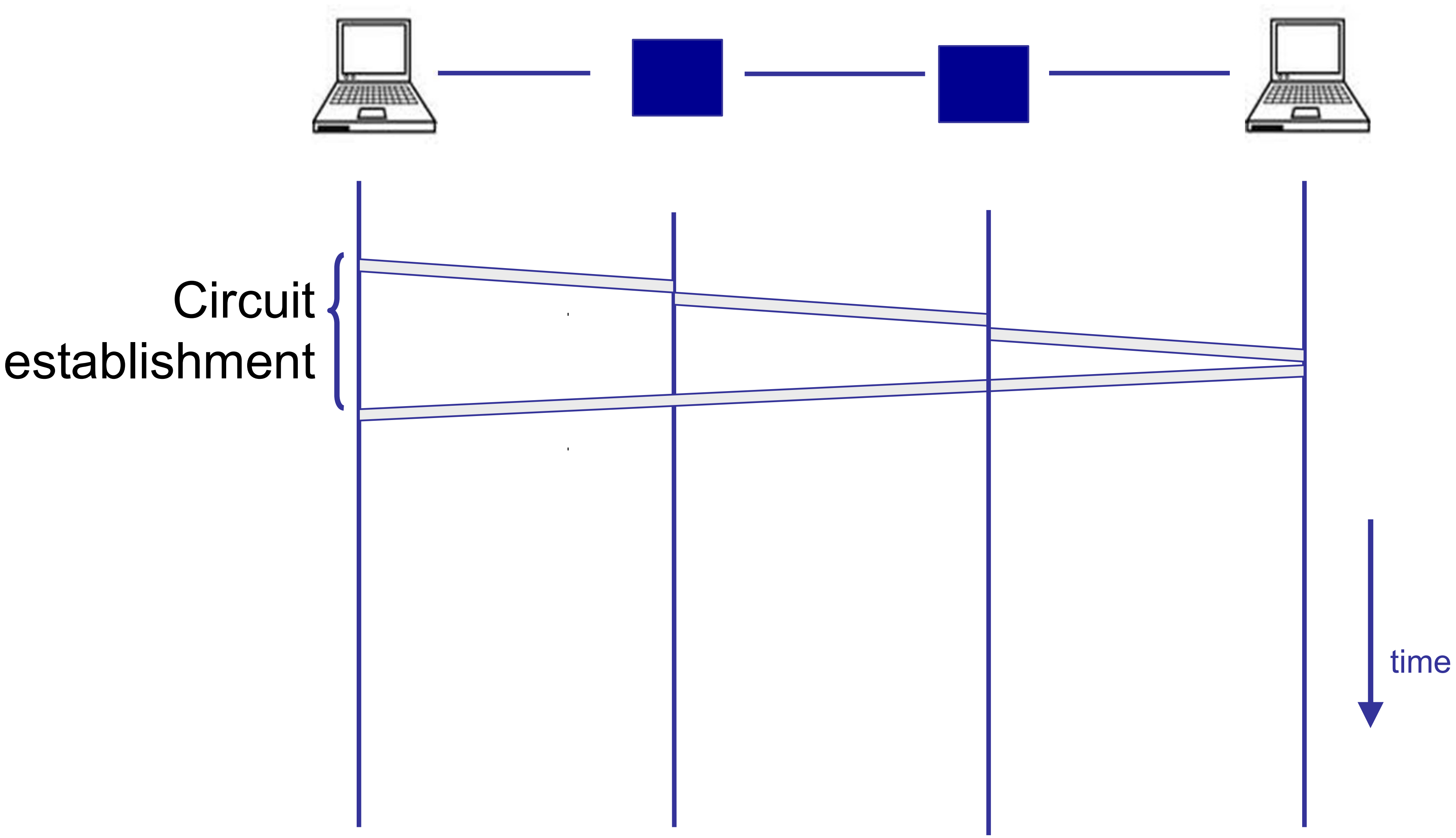
Timing in circuit switching



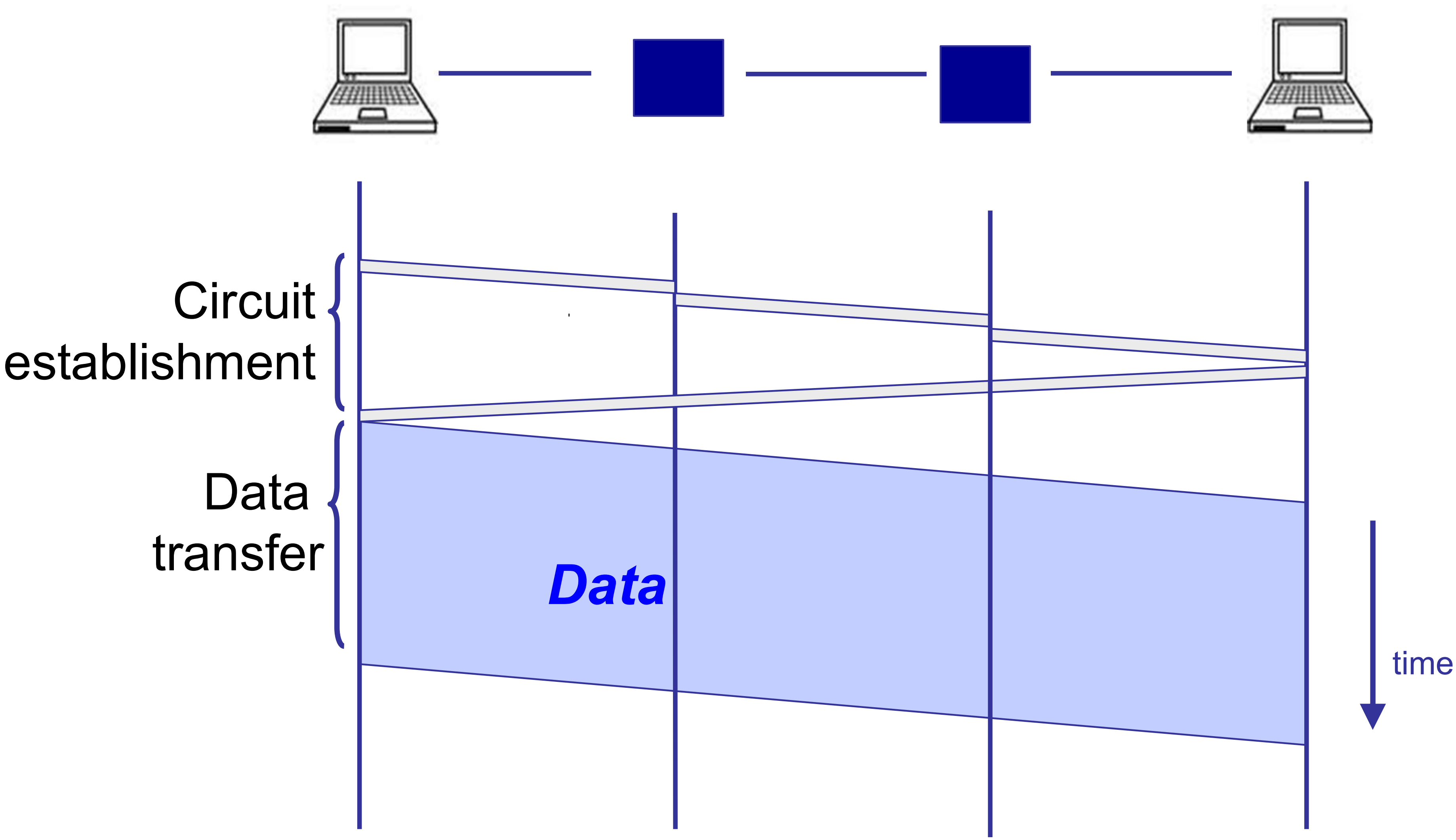
Timing in circuit switching



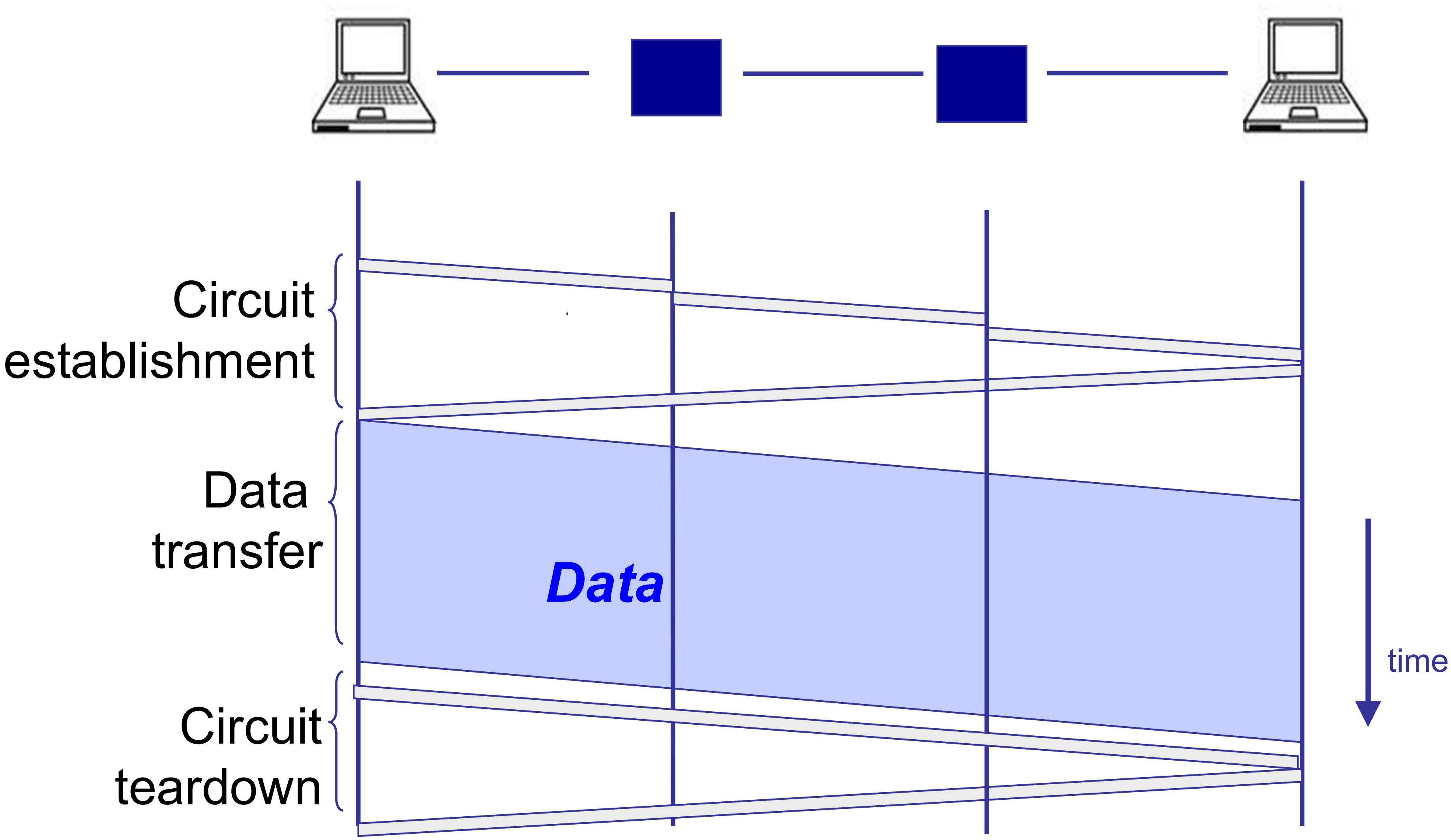
Timing in circuit switching



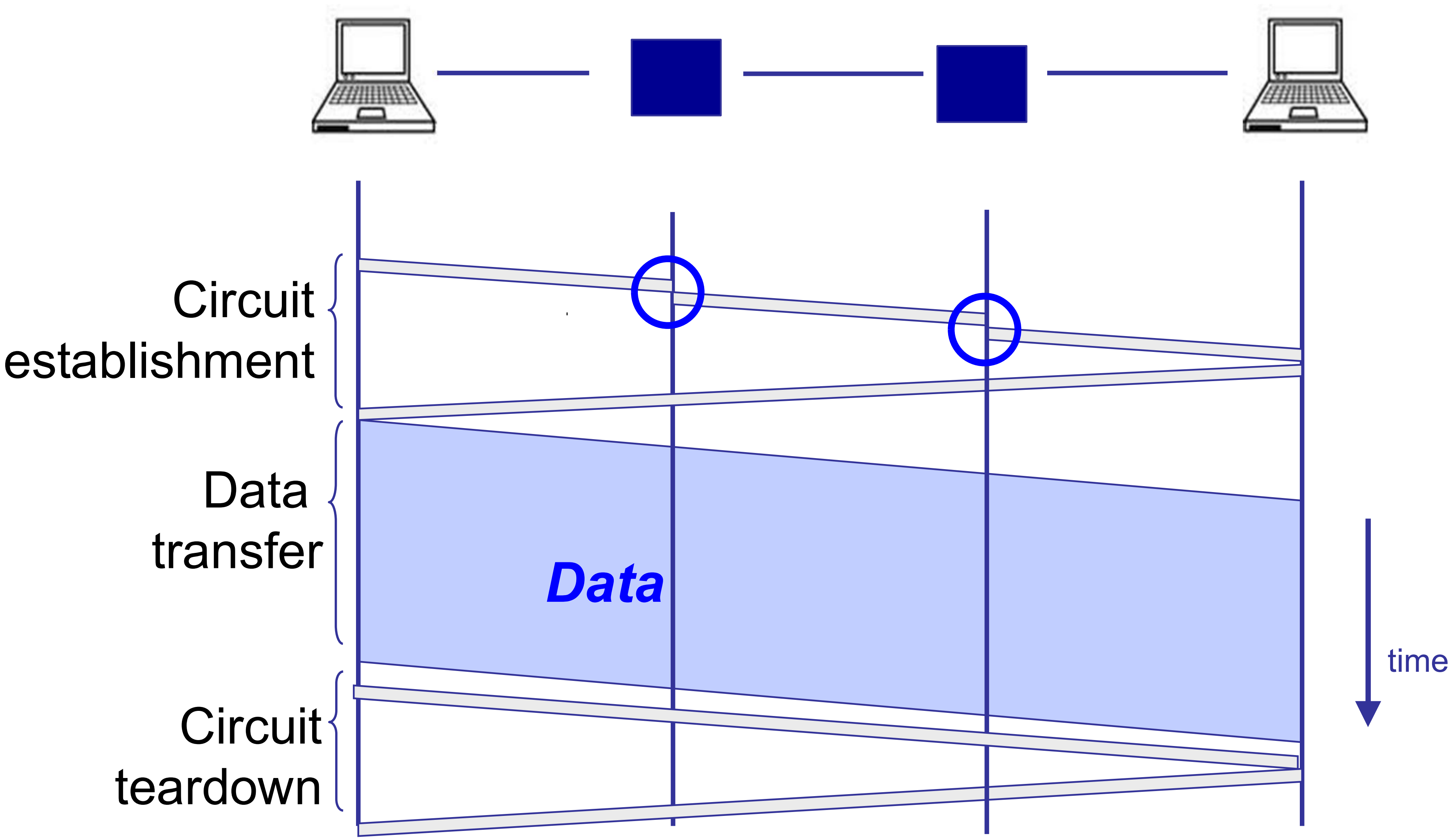
Timing in circuit switching



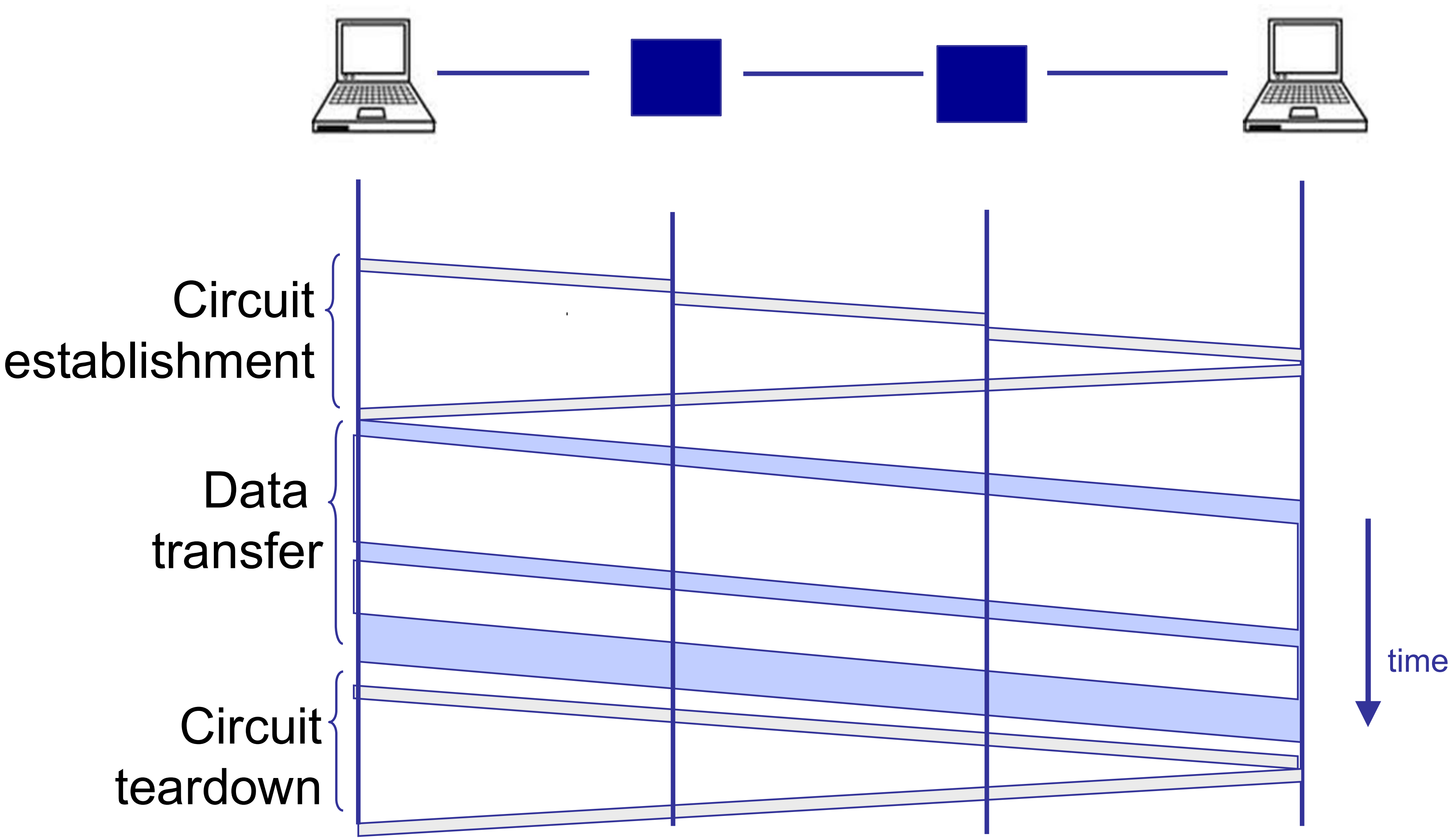
Timing in circuit switching



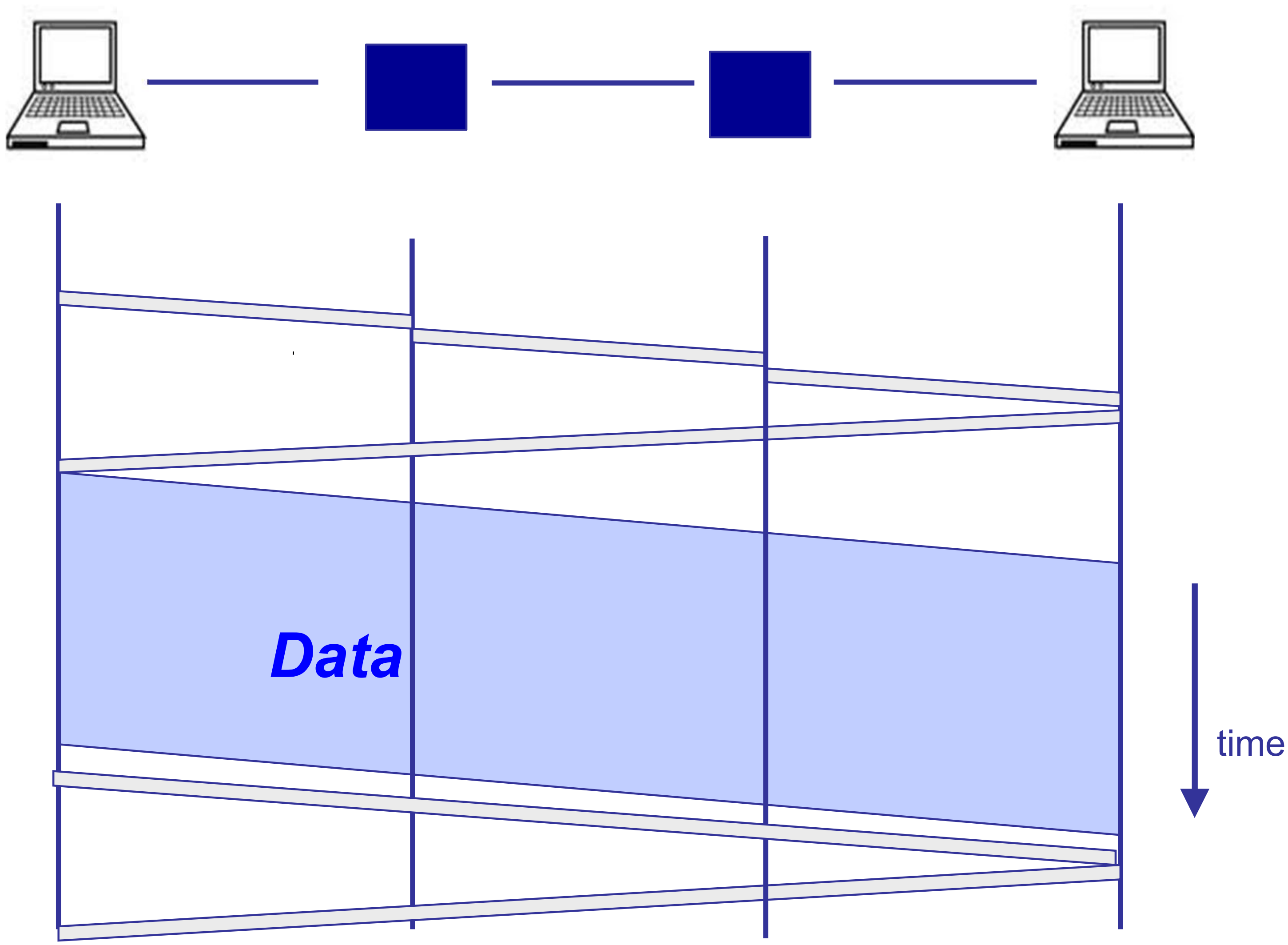
Why the delays?



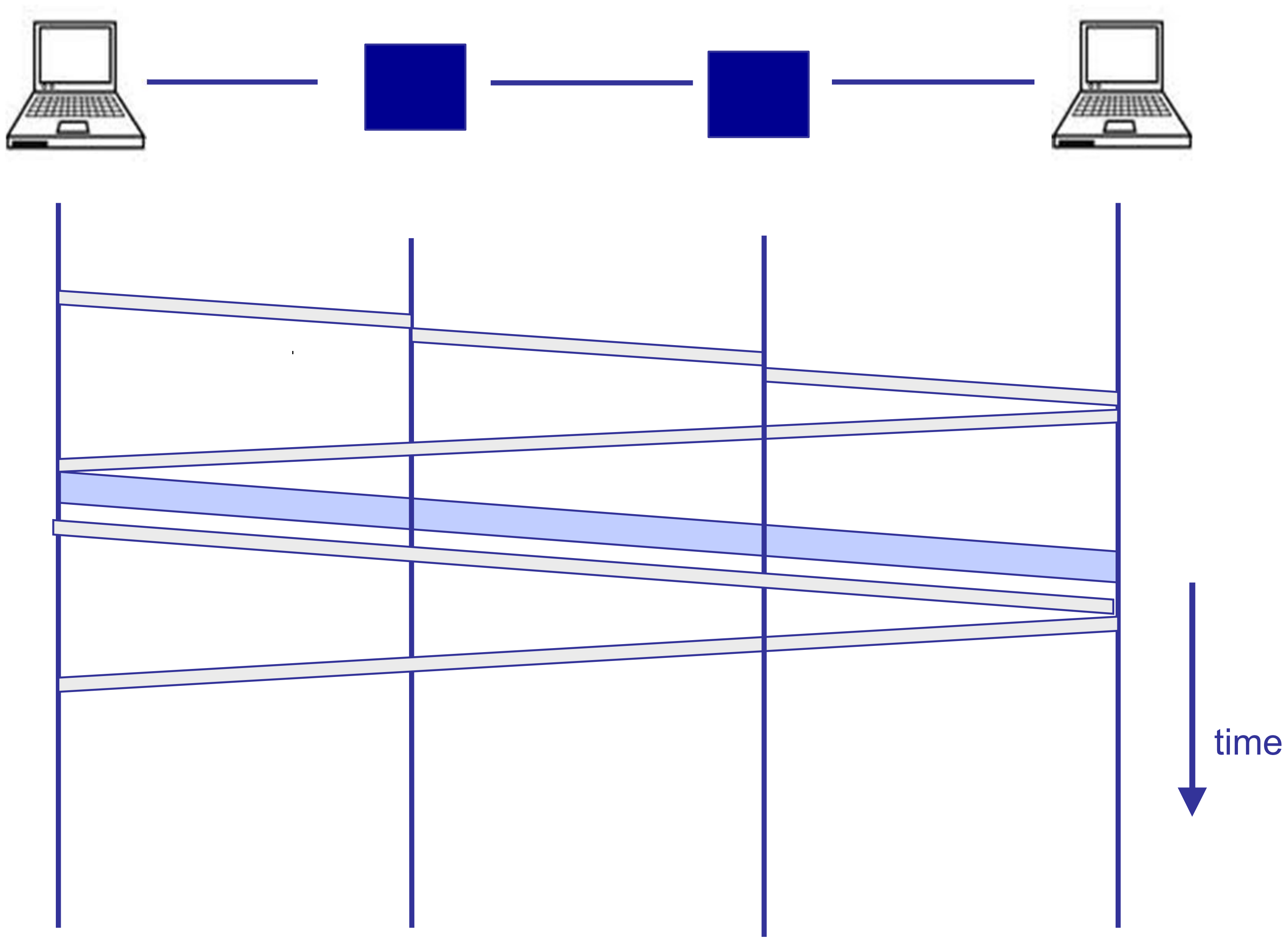
Timing in circuit switching



Timing in circuit switching



Timing in circuit switching



Have we found the right solution?

We don’t really know

What we do know

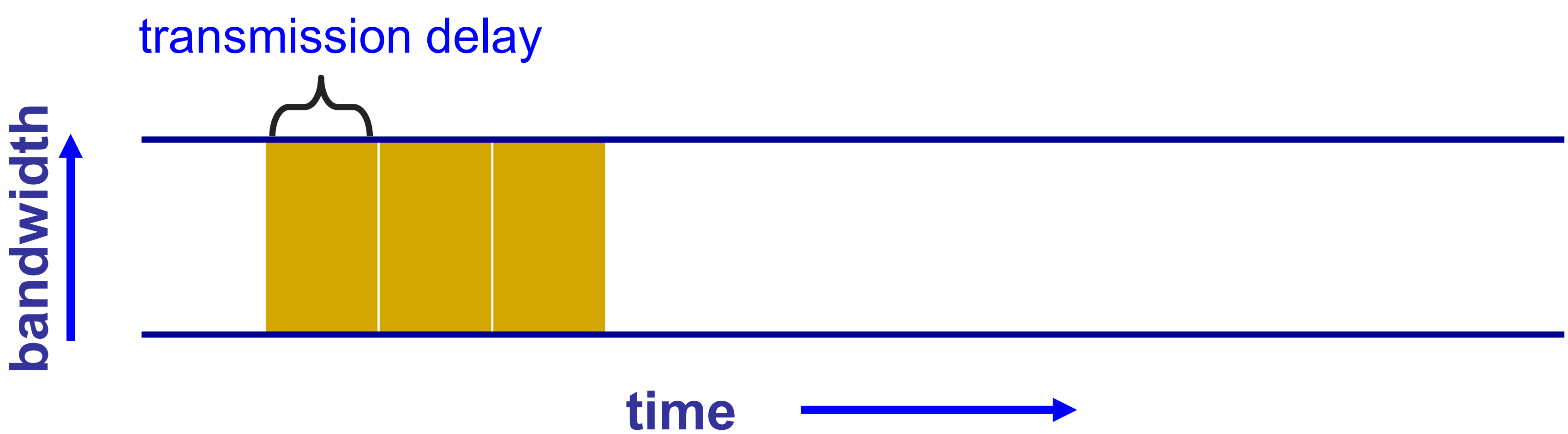
- The early Internet pioneers produced a solution that was successful beyond all imagining
- Several enduring architectural principles and practices emerged from their work

Still, it is just one design with many questions

The Internet is a lesson

- In how to reason through the design of a <u>very</u> complex system
  - What are our goals and constraints?
  - What's the right prioritization of goals?
  - How do we decompose a problem?
  - Who does what? How?
  - What are the interfaces between components?
  - What are the tradeoffs between design options?

Pipe view of a link



- Transmission delay decreases as bandwidth increases

A network link: BDP

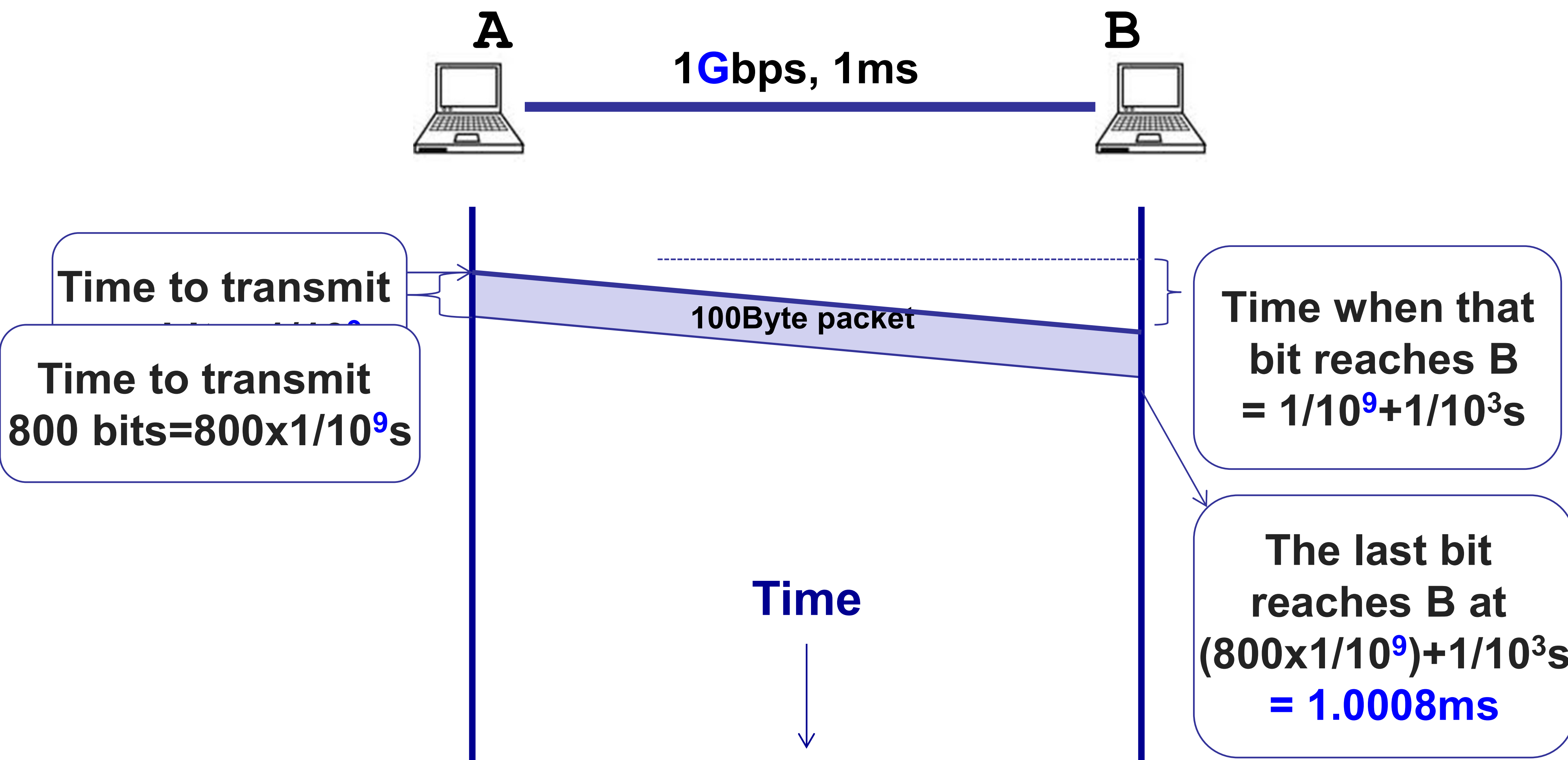
- Link bandwidth
  - Number of bits sent/received per unit time (bits/sec or bps)
- Propagation delay
  - Time for one bit to move through the link (seconds)
- Bandwidth-Delay Product (BDP)
  - Number of bits “in flight” at any time
- BDP = bandwidth × propagation delay

BDP Examples

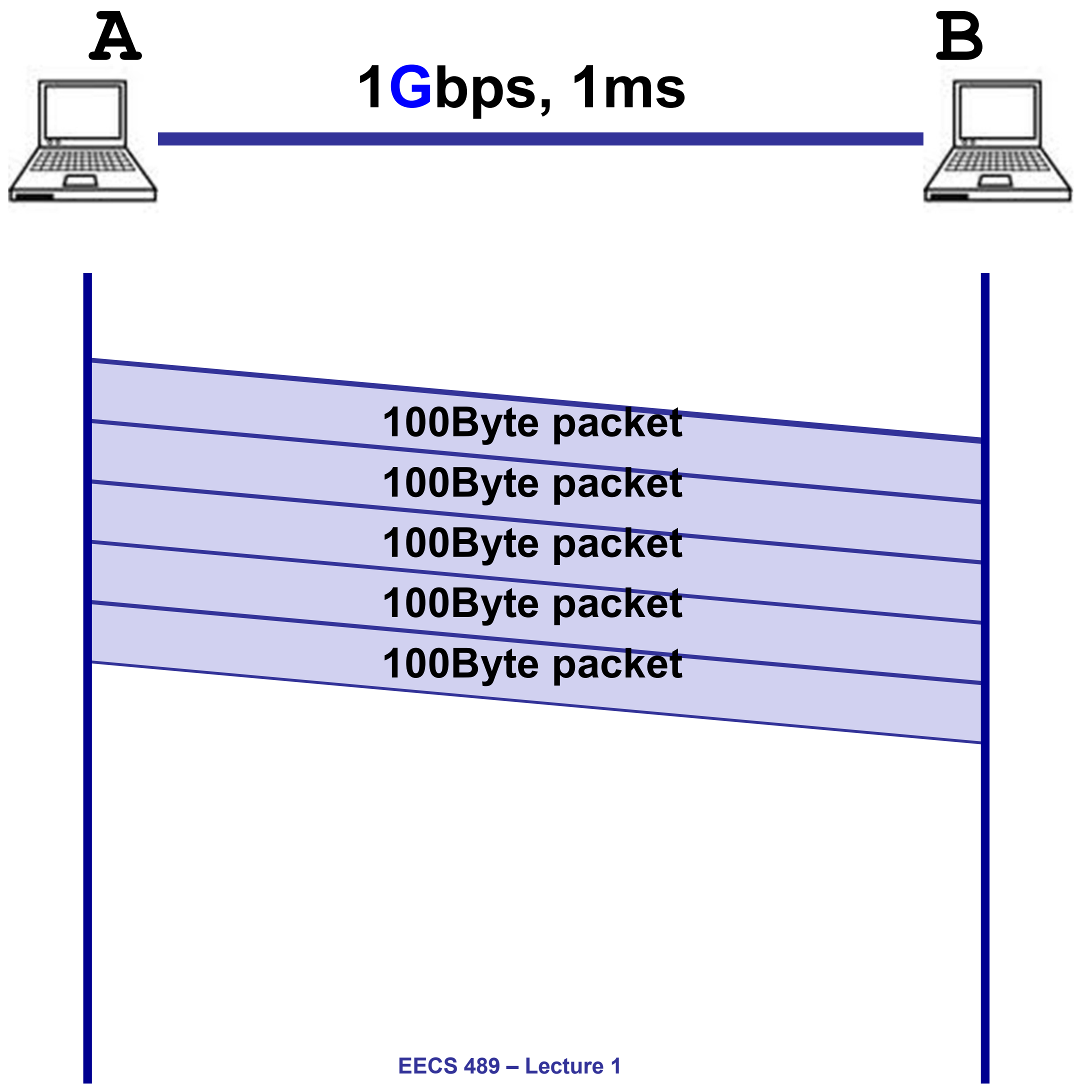
- Same city over a slow link:
  - Bandwidth: ~100Mbps
  - Propagation delay: ~0.1msec
  - BDP: 10,000bits (1.25KBytes)

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

Packet delay
Sending a 100-byte packet



Sending a large file using 100-byte packets



Persistent overload leads to packet drop/loss