Basic Concepts

CS168, Fall 2014

Sylvia Ratnasamy

http://inst.eecs.berkeley.edu/~cs168/fa14/

Administrivia

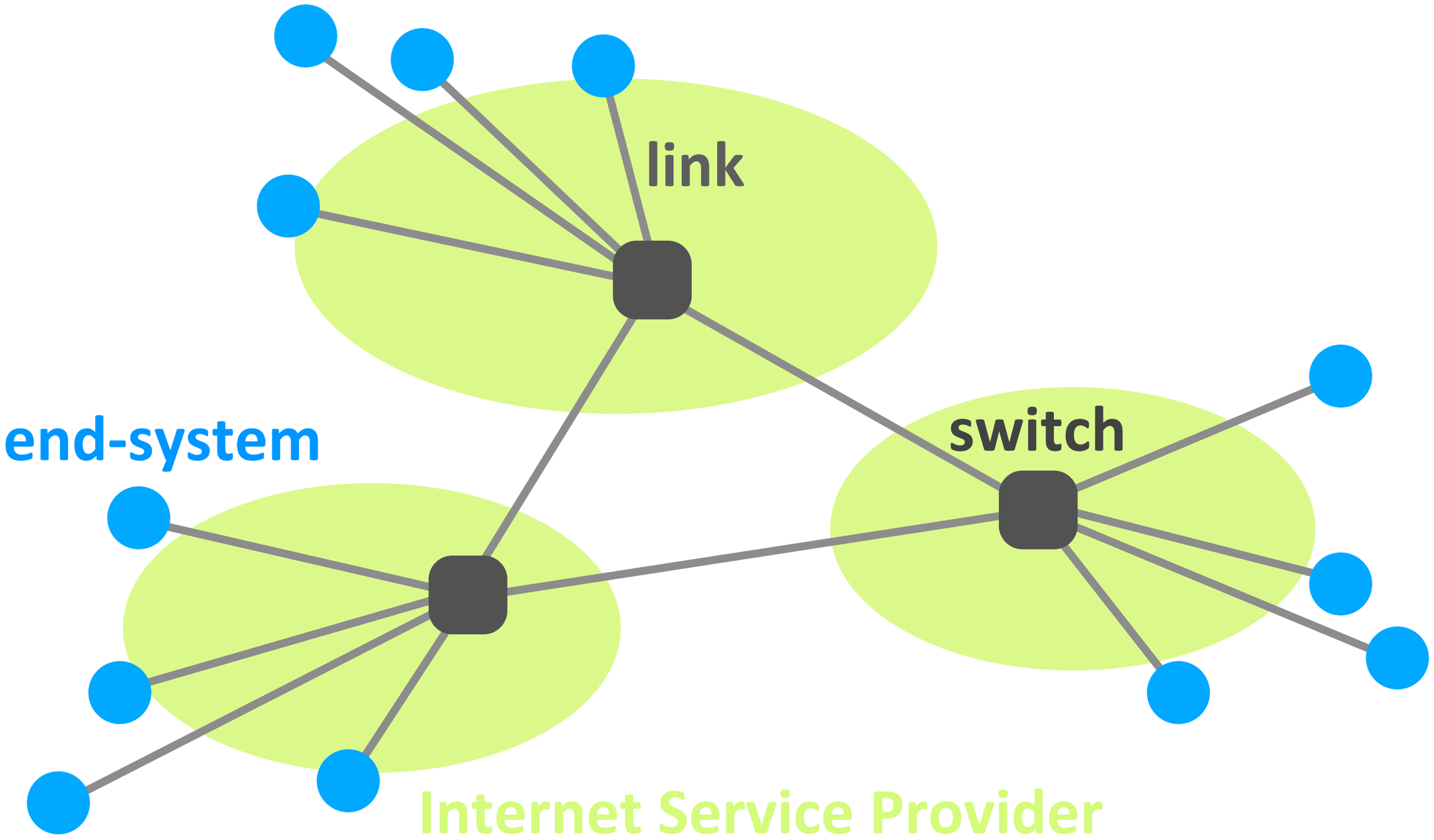
- Discussion sections start Wednesday, Sep 10
- Homework#1 out this week
  - we will be using PandaGrader for submissions; stay tuned for details

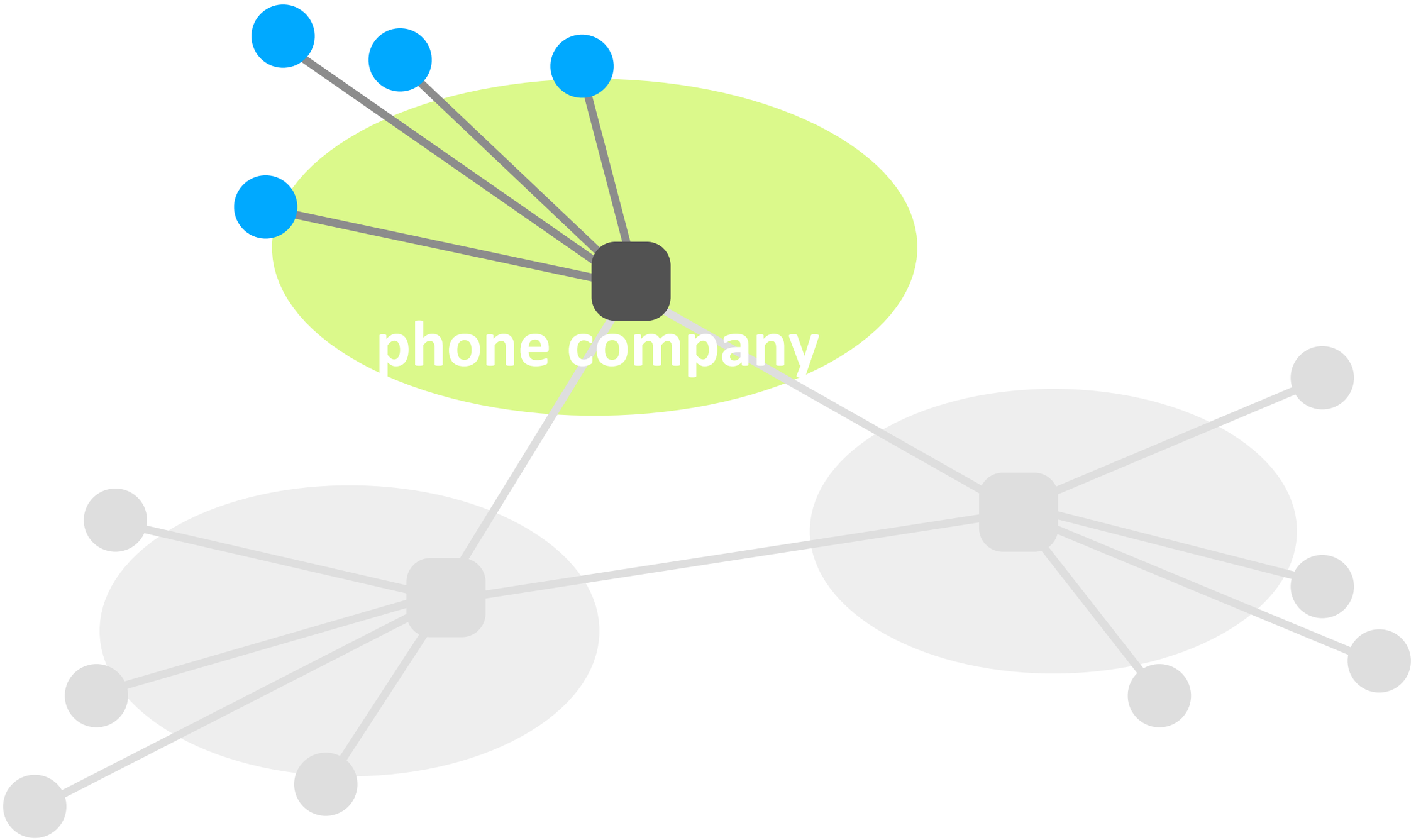
Today

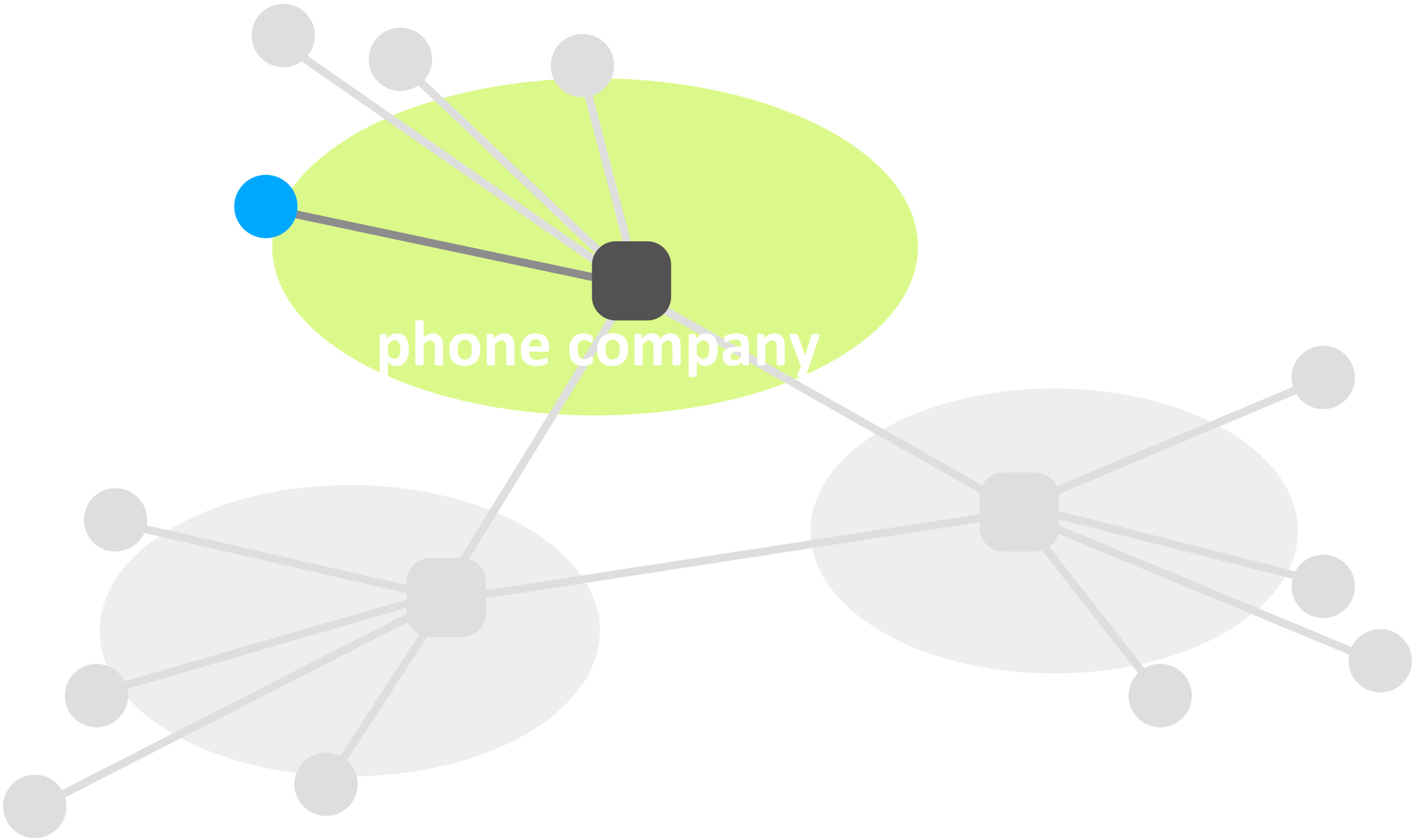
- What is a network made of?
- How is it shared?
- How do we evaluate a network?

Today

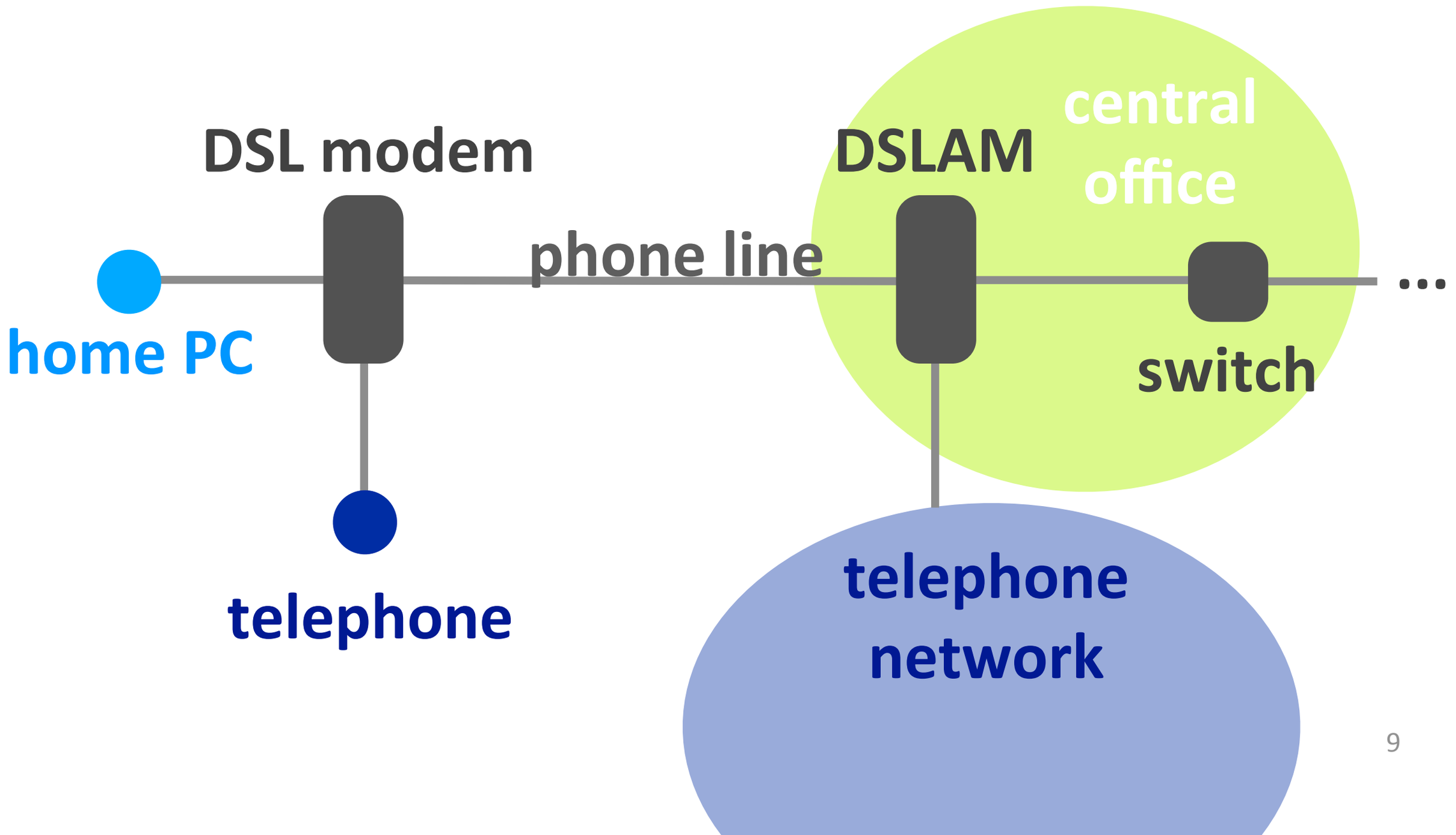
- What is a network made of?
- How is it shared?
- How do we evaluate a network?


link
end-system
switch
Internet Service Provider

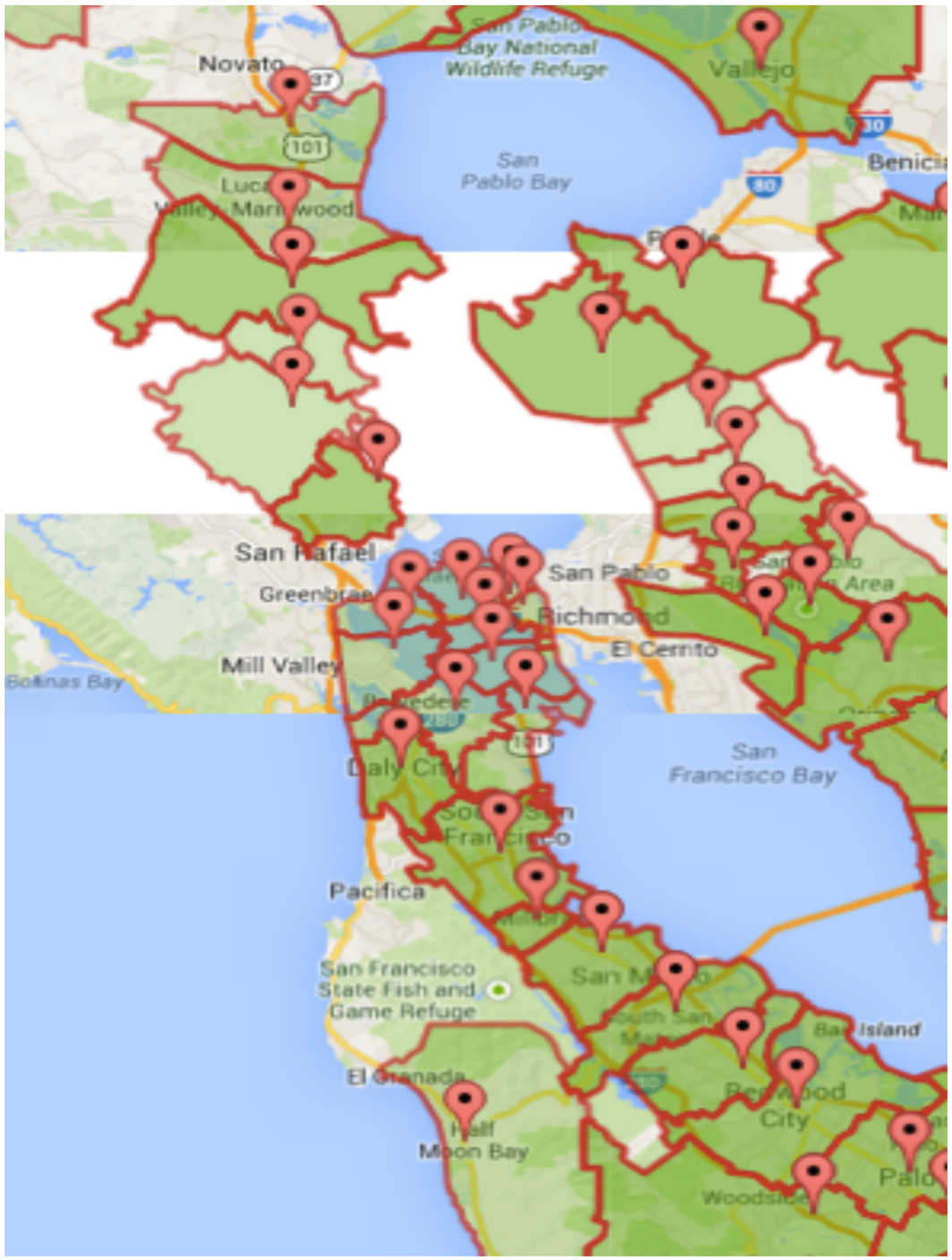

phone company


phone company


home PC
switch


DSL modem
DSLAM
central office
phone line
...
home PC
switch
telephone
telephone network

AT&T Central Offices in the Bay Area

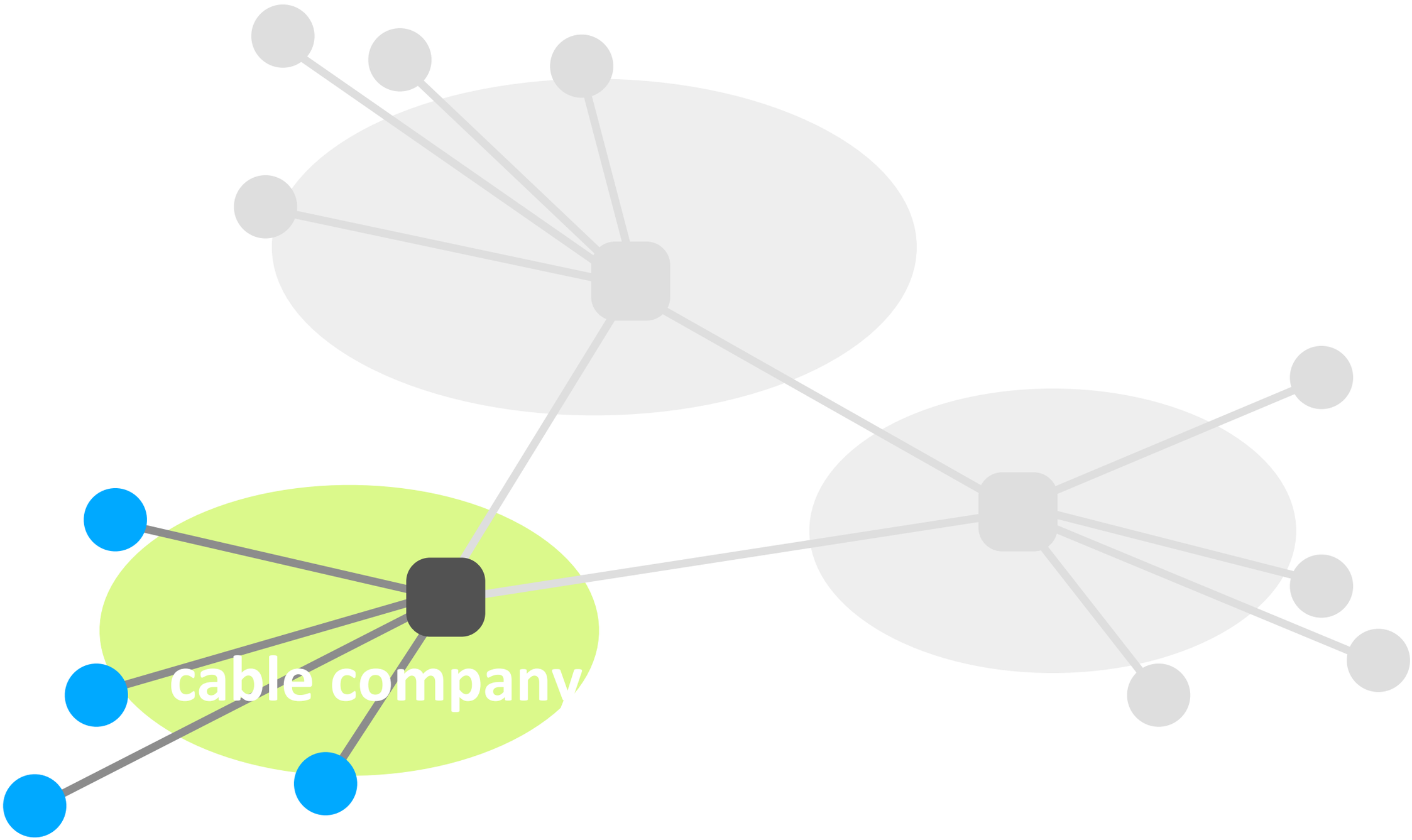


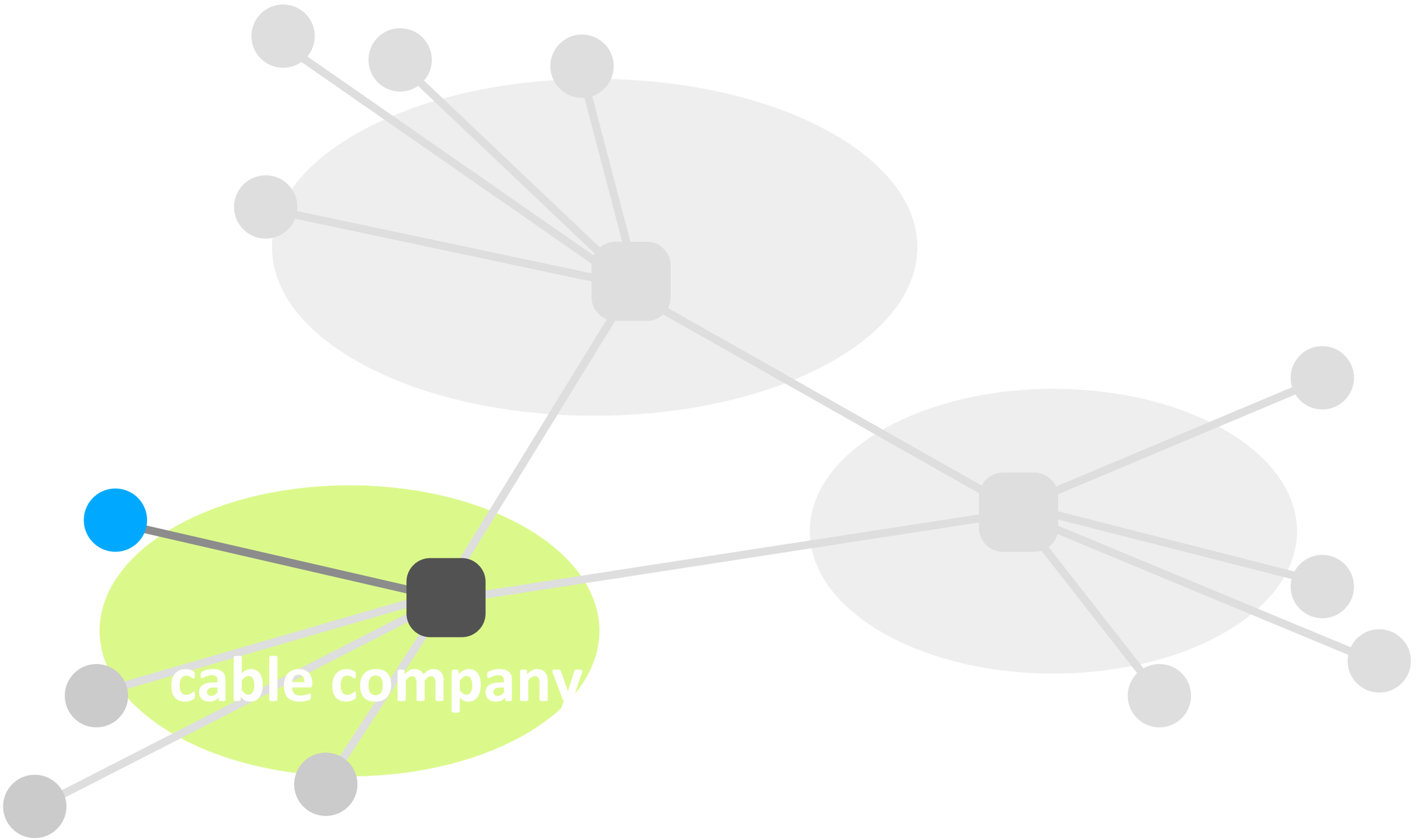
AT&T Central Office in Berkeley

Digital Subscriber Line (DSL)

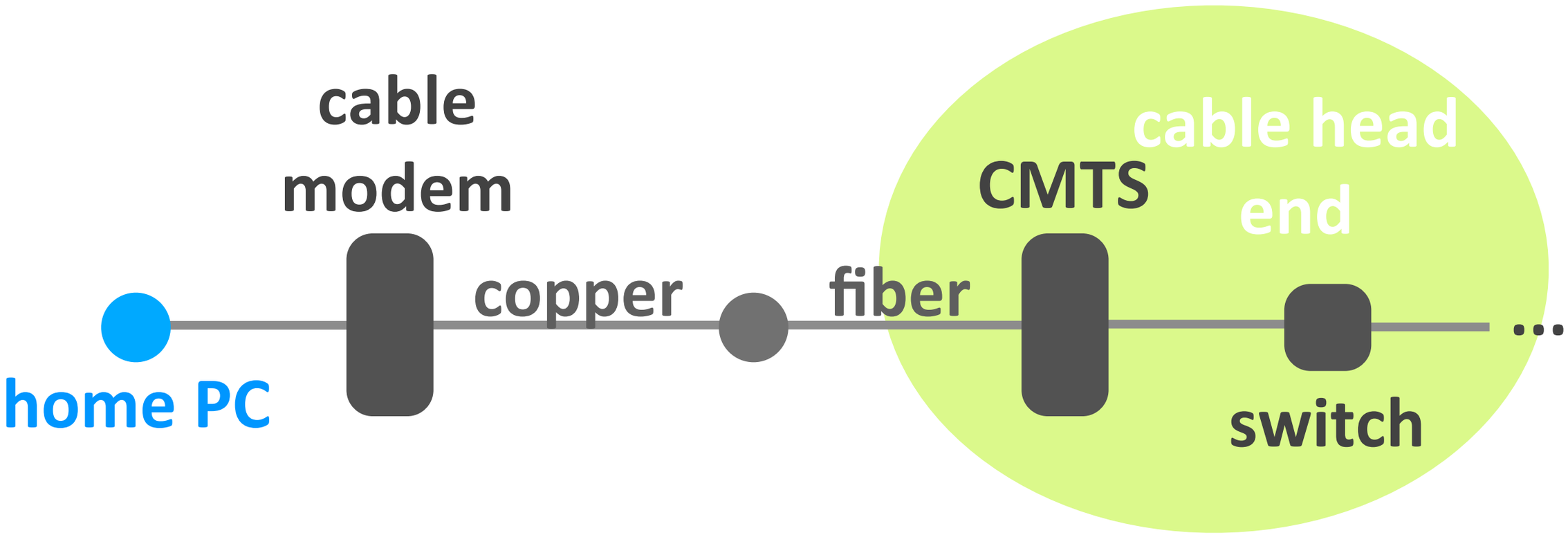
- Twisted pair copper
- 3 separate channels
  - *downstream data channel*
  - *upstream data channel*
  - *2-way phone channel*
- up to 25 Mbps downstream
- up to 2.5 Mbps upstream

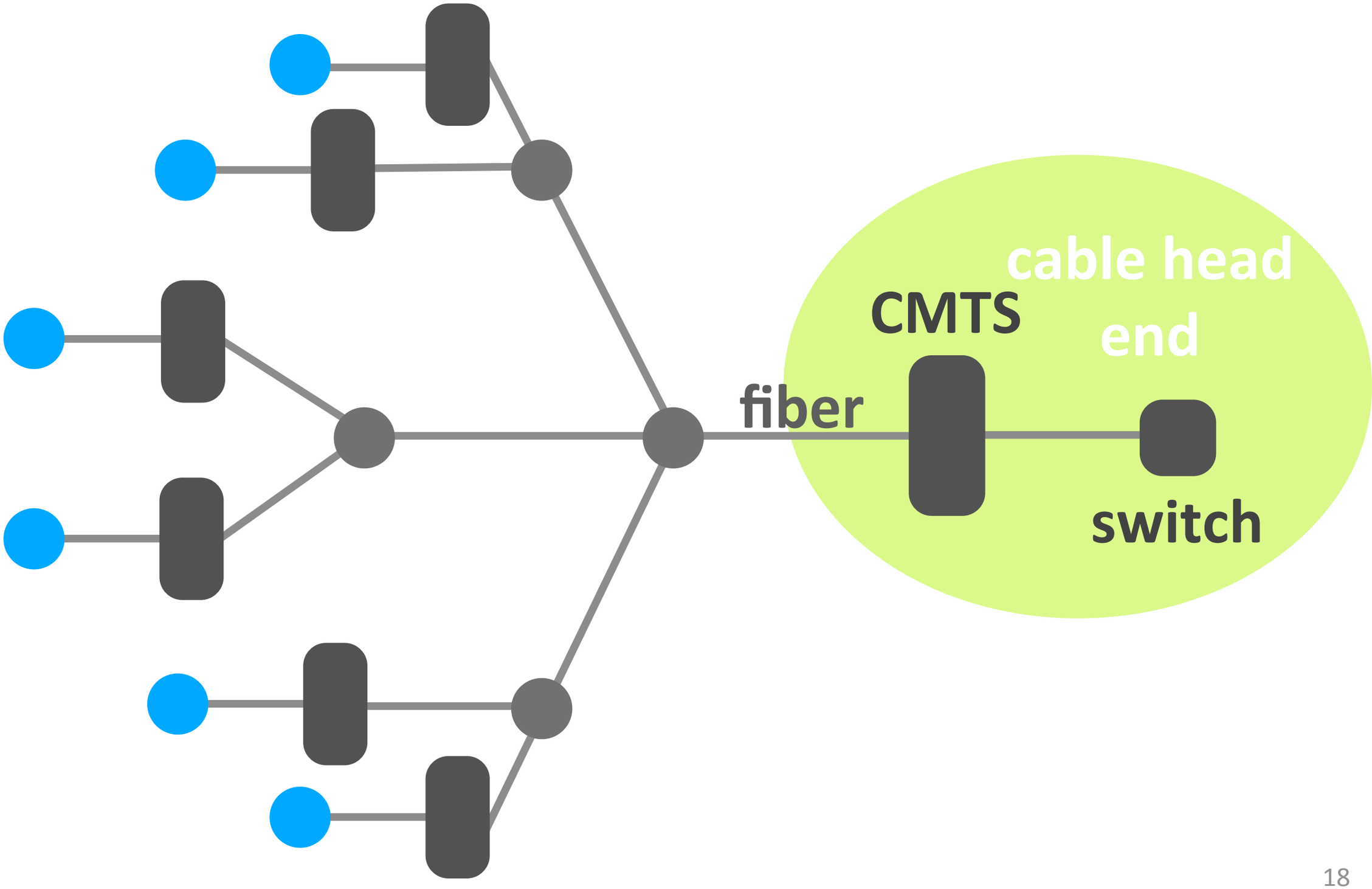
Why phone lines?


cable company


cable company

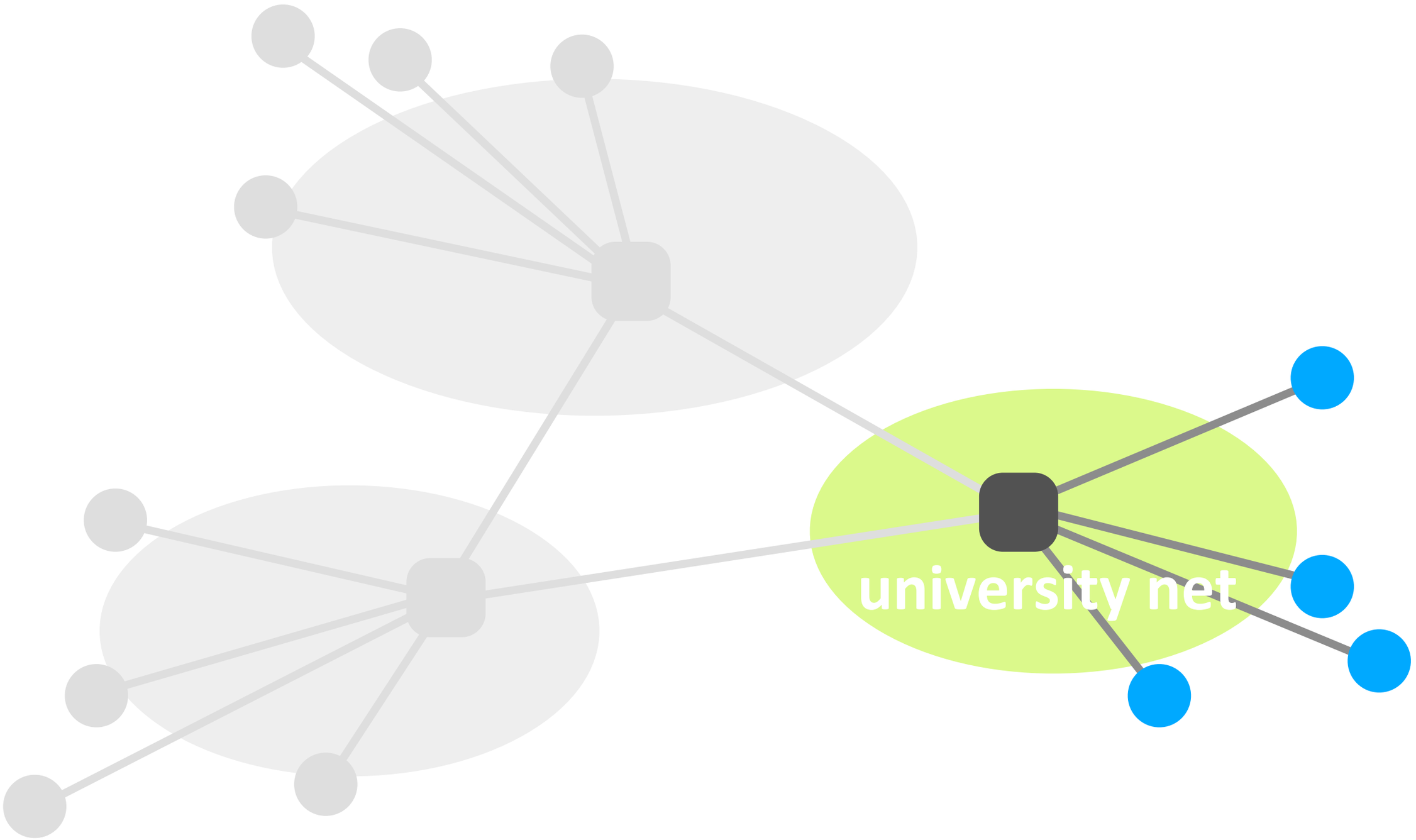

home PC
switch

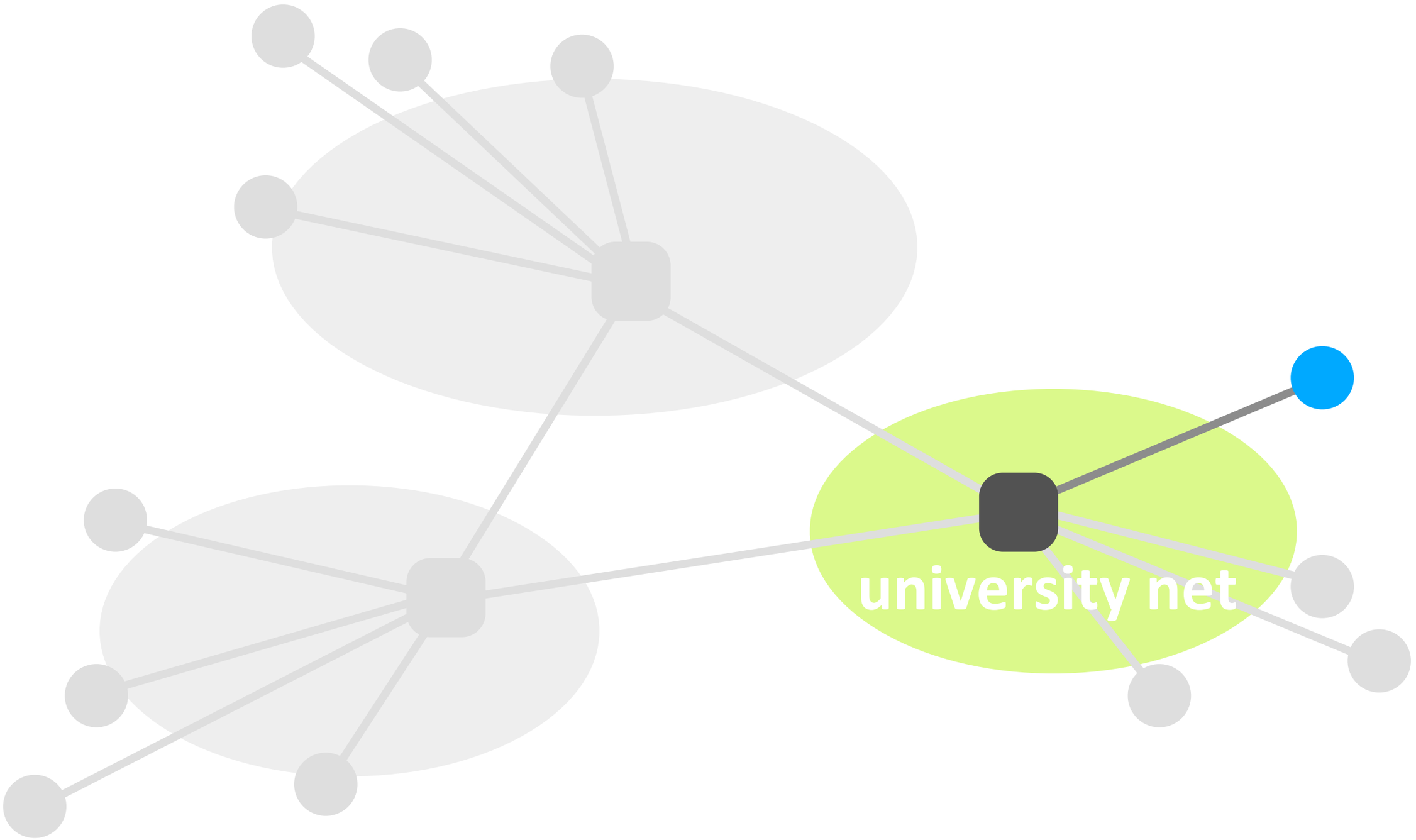

cable
modem
cable head
end
CMTS
copper
fiber
...
home PC
switch

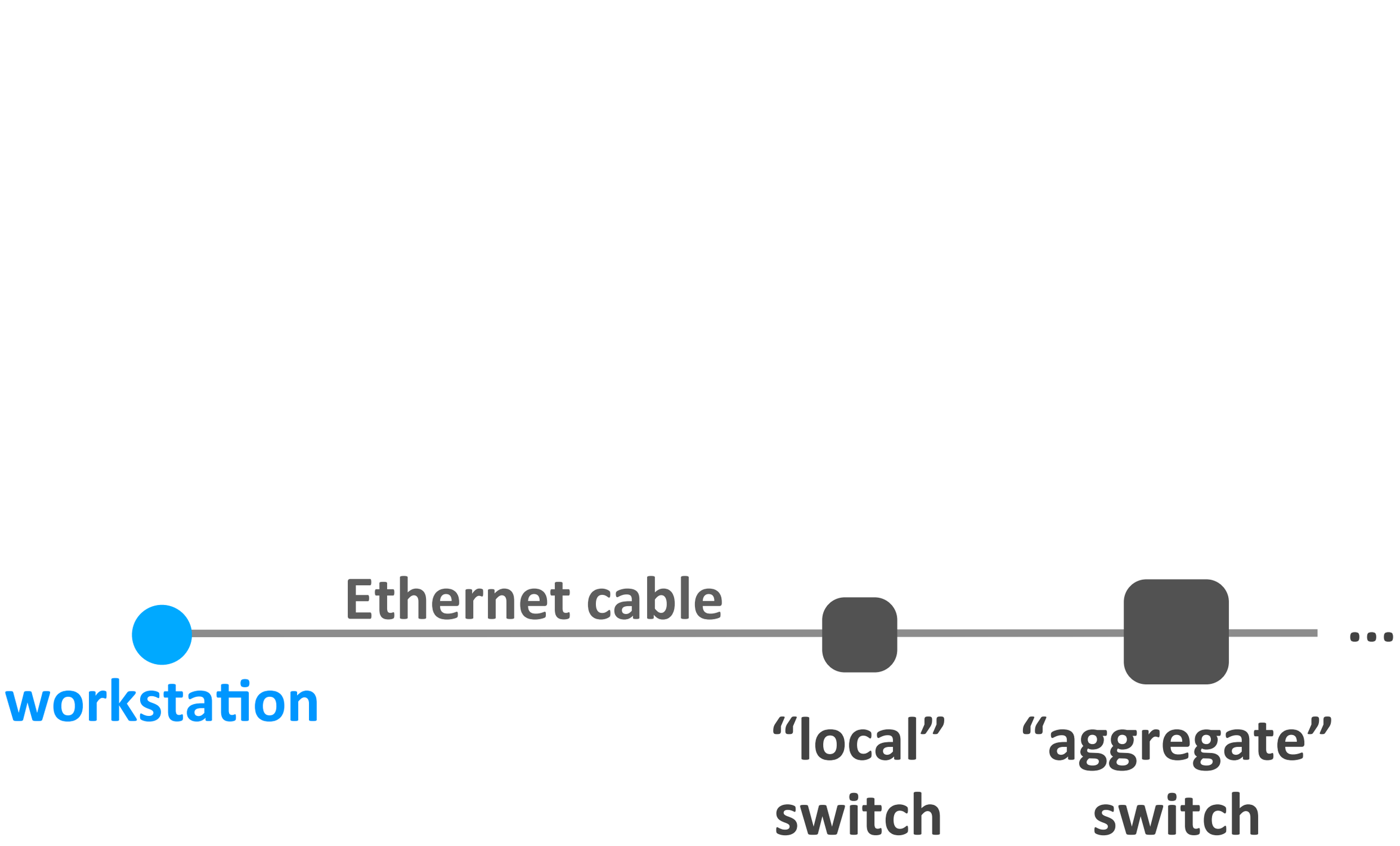

cable head
end
CMTS
fiber
switch

Cable

- Coaxial copper & fiber
- up to 42.8 Mbps downstream
- up to 30.7 Mbps upstream
- shared broadcast medium


university net


university net

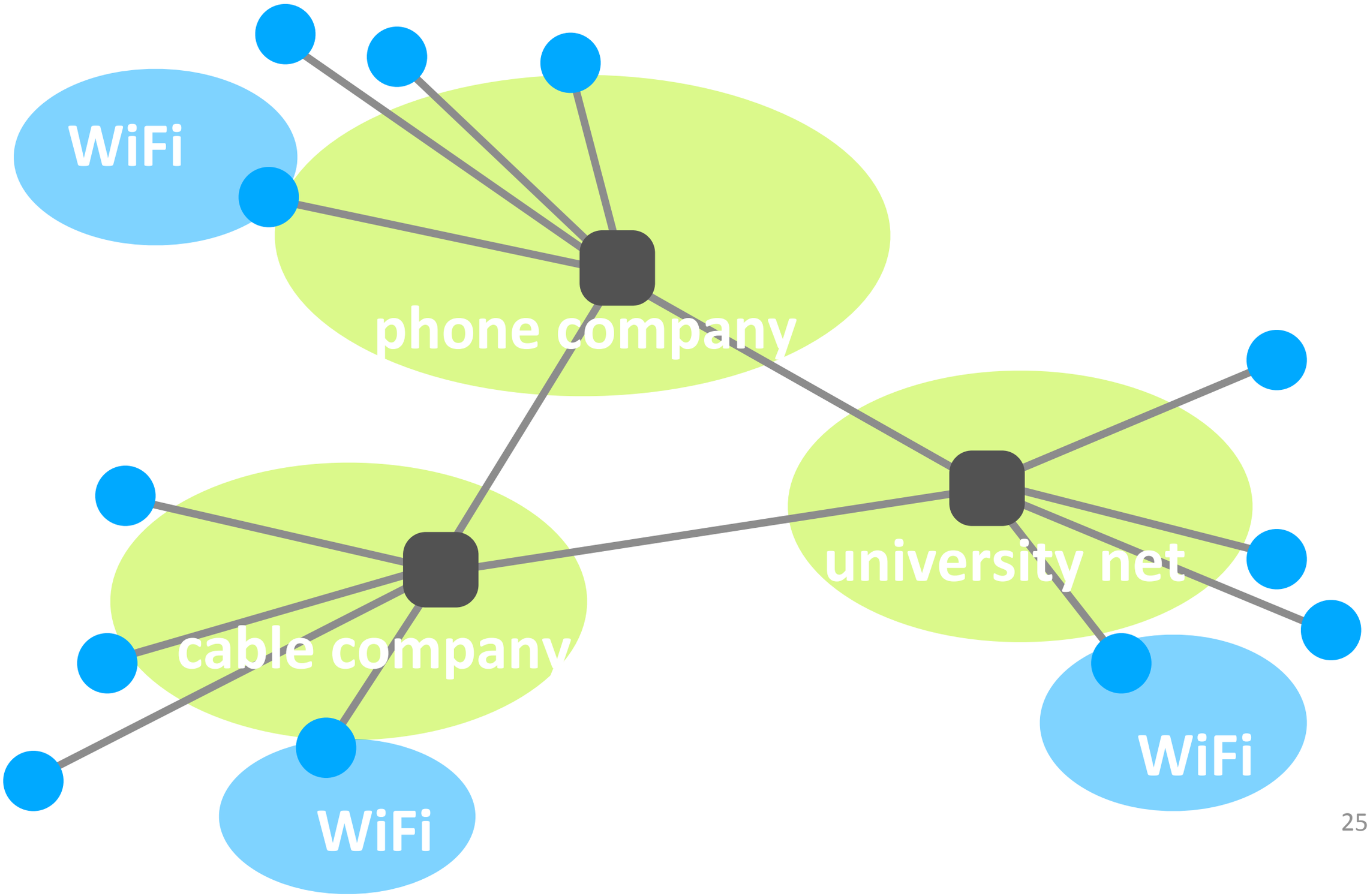

Ethernet cable
workstation
“local” switch
“aggregate” switch
…

Ethernet

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

& more

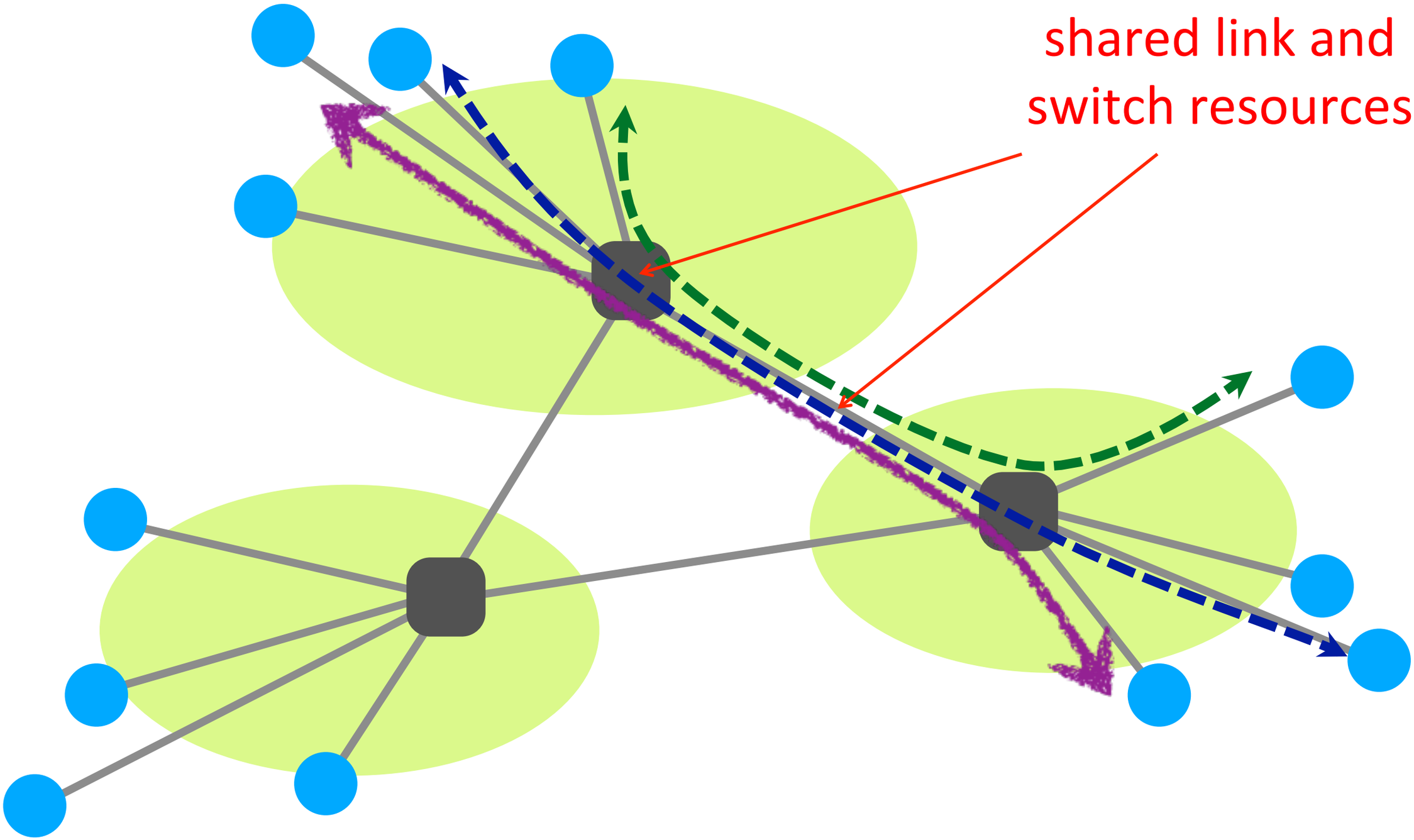
- Cellular (smart phones)
- Satellite (remote areas)
- Fiber to the Home (home)
- Optical carrier (Internet backbone)


WiFi
phone company
cable company
university net
WiFi
WiFi

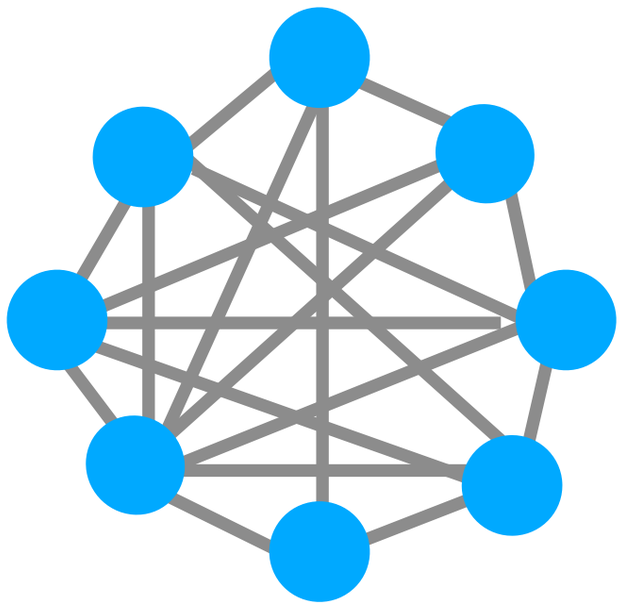
- What physical infrastructure is already available?

Today

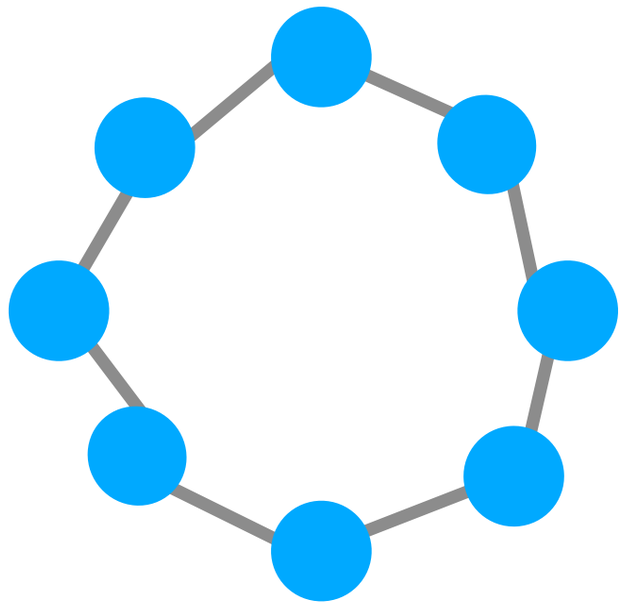
- What is a network made of?
- How is it shared?
- How do we evaluate a network?


shared link and
switch resources

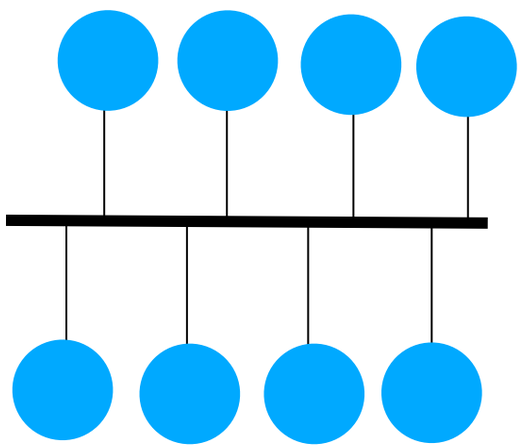
How do we scale a network to many end-systems?



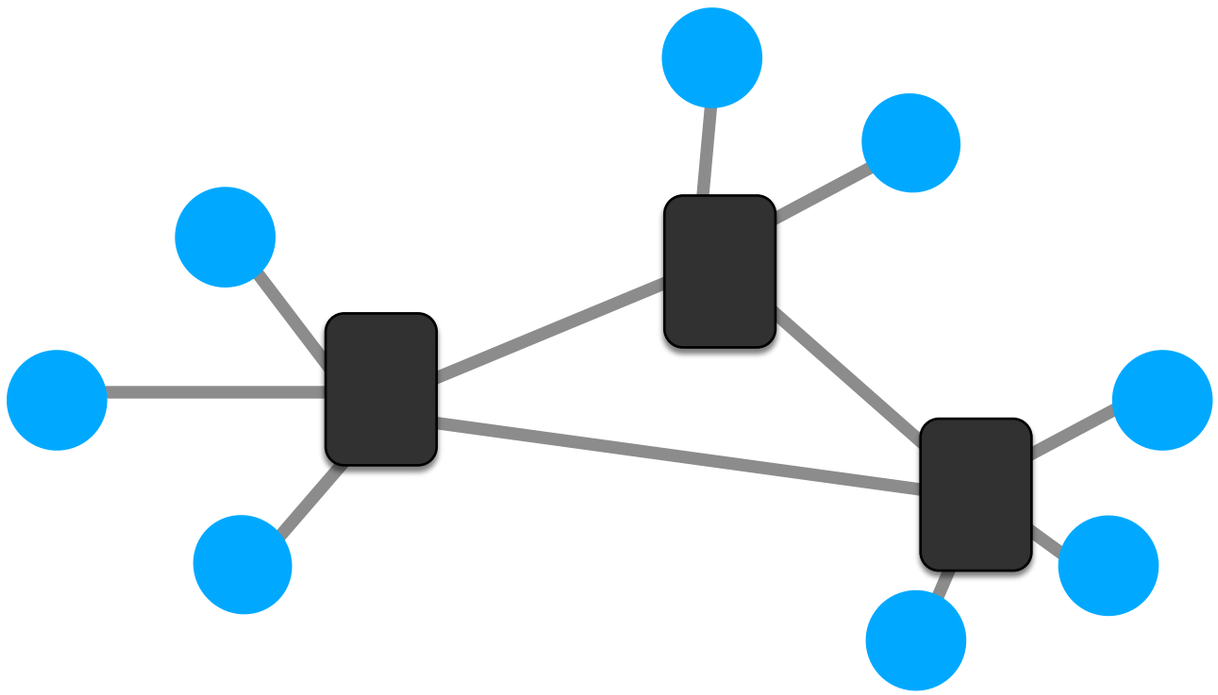
$O(n^2)$ links



per-node capacity scales as 1/n



How do we scale a network to many end-systems?

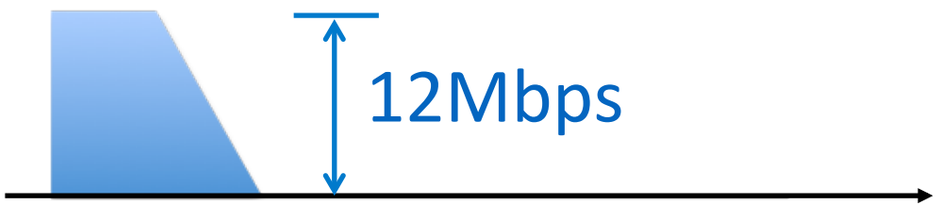


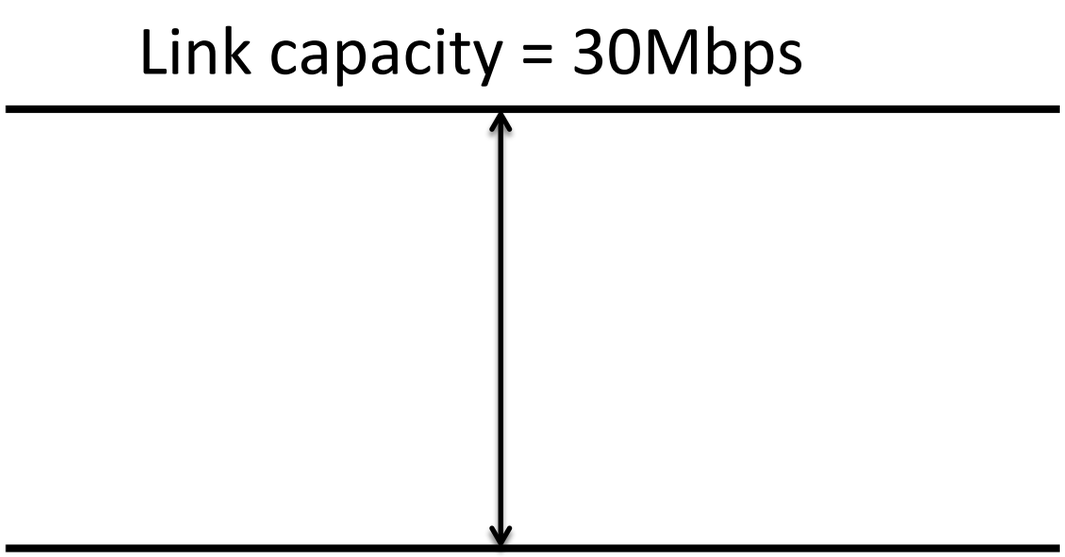
Switched networks enable efficient scaling!

Two approaches to sharing

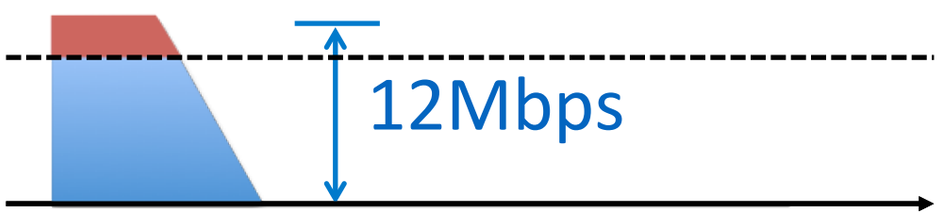
- Reservations
- On demand

Intuition: Three sources of “bursty” traffic

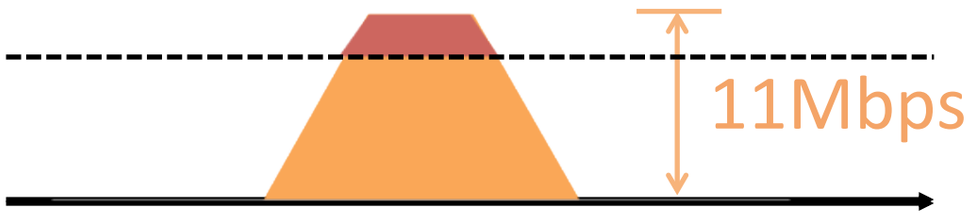


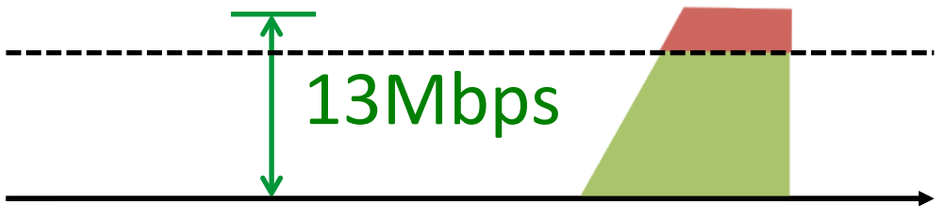


Intuition: reservations



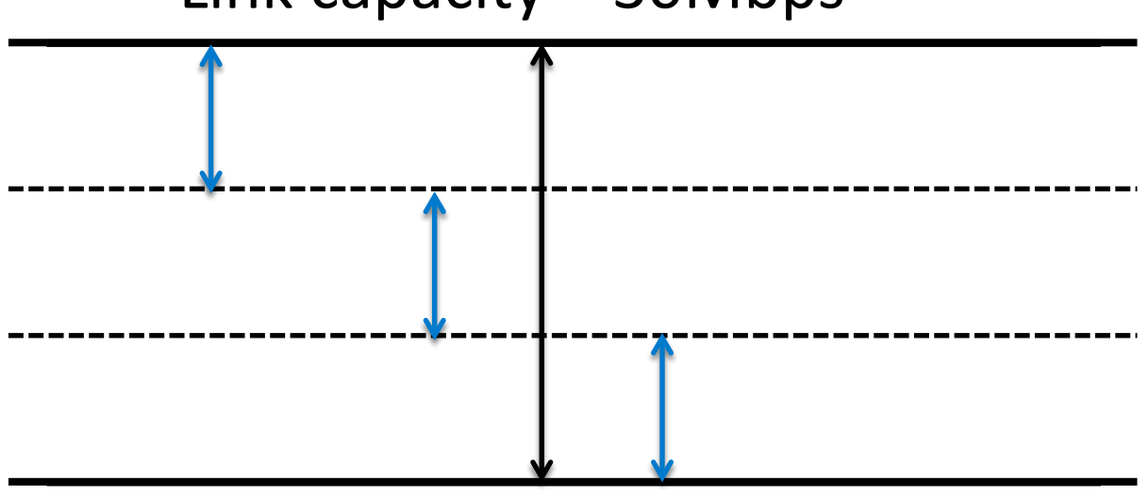
Frequent overloading





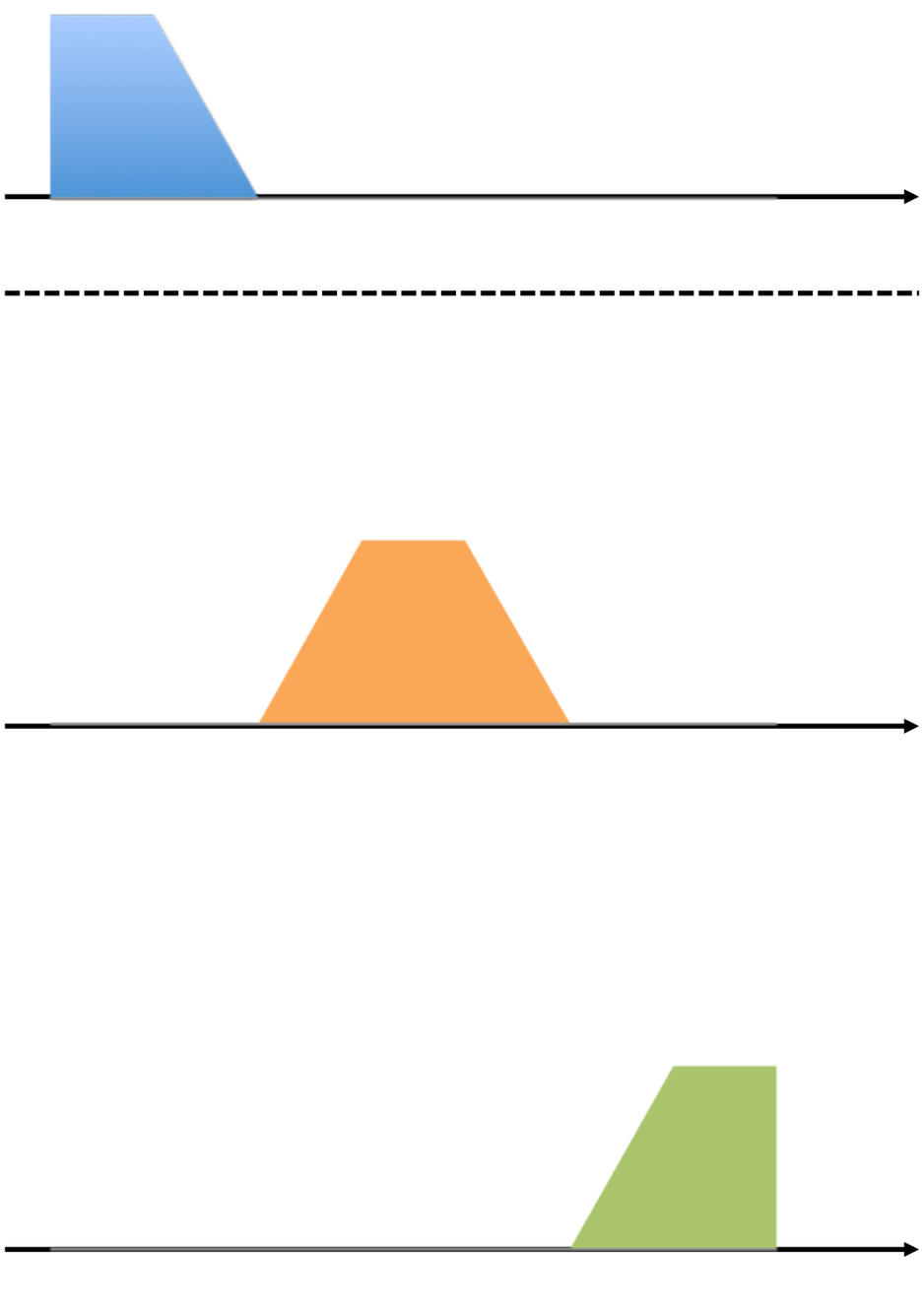
Time

Link capacity = 30Mbps



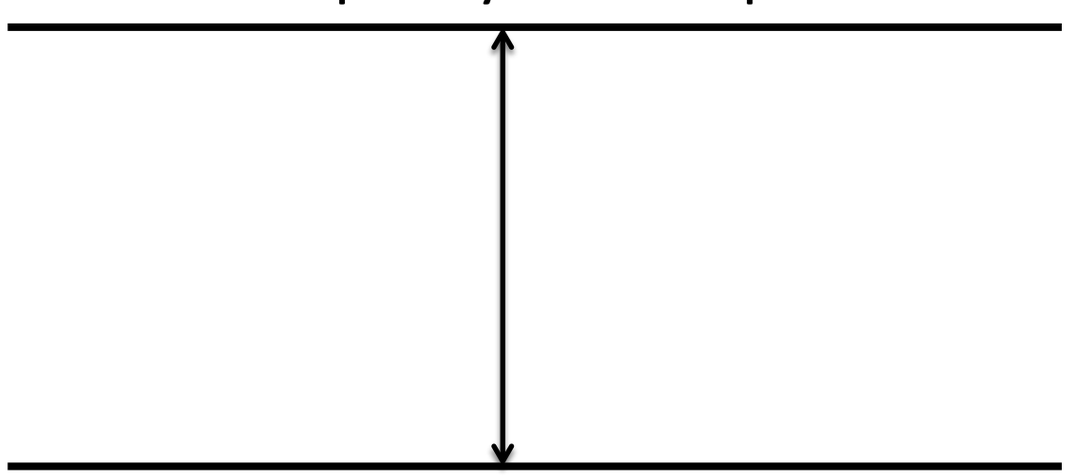
Each source gets 10Mbps

Intuition: on demand



No overloading

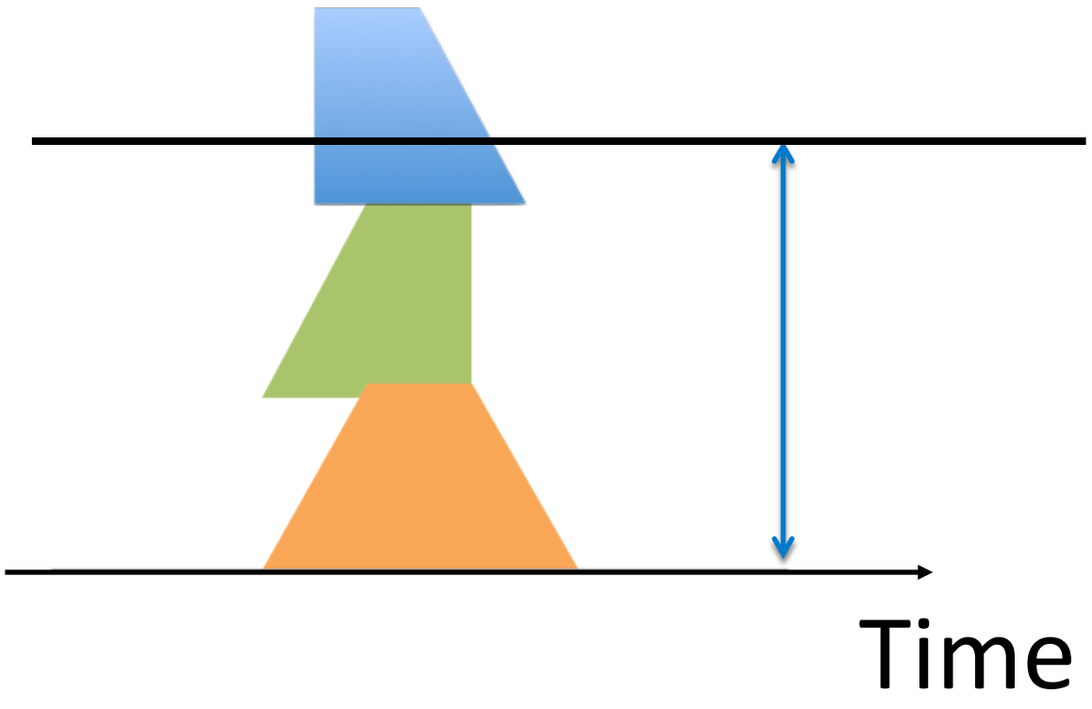
Link capacity = 30Mbps

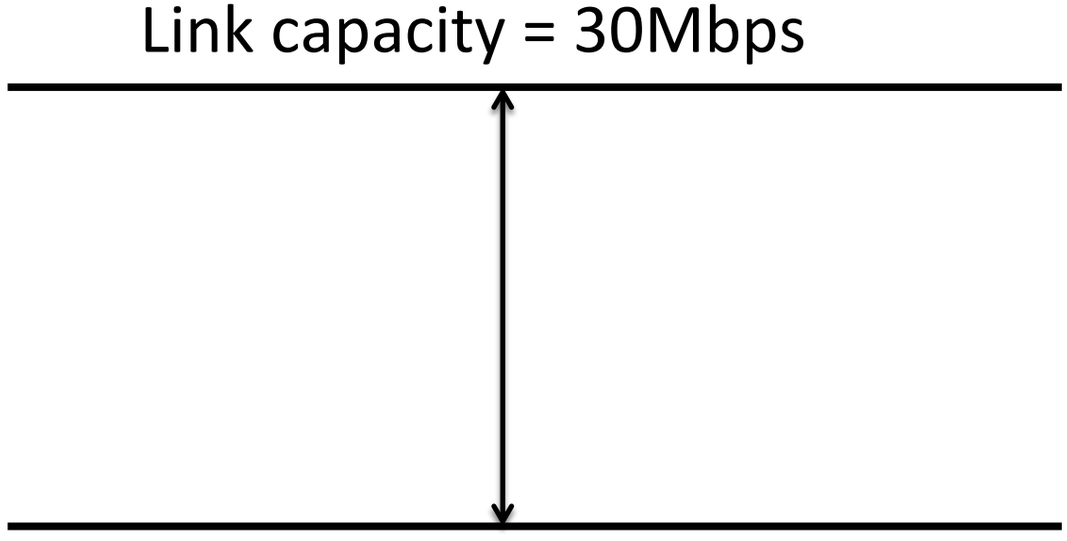


Switching on-demand exploits *statistical multiplexing* better than reservations

- Sharing using the statistics of demand
- Good for bursty traffic (average << peak demand)
- Similar to insurance, with the same failure mode

Intuition: on demand





What do we do under overload?

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 *vs.* dedicated machines

Two approaches to sharing

- Reservations
- On demand

How are these implemented?

Two approaches to sharing

- Reservations → circuit switching
- On demand → packet switching

How are these implemented?

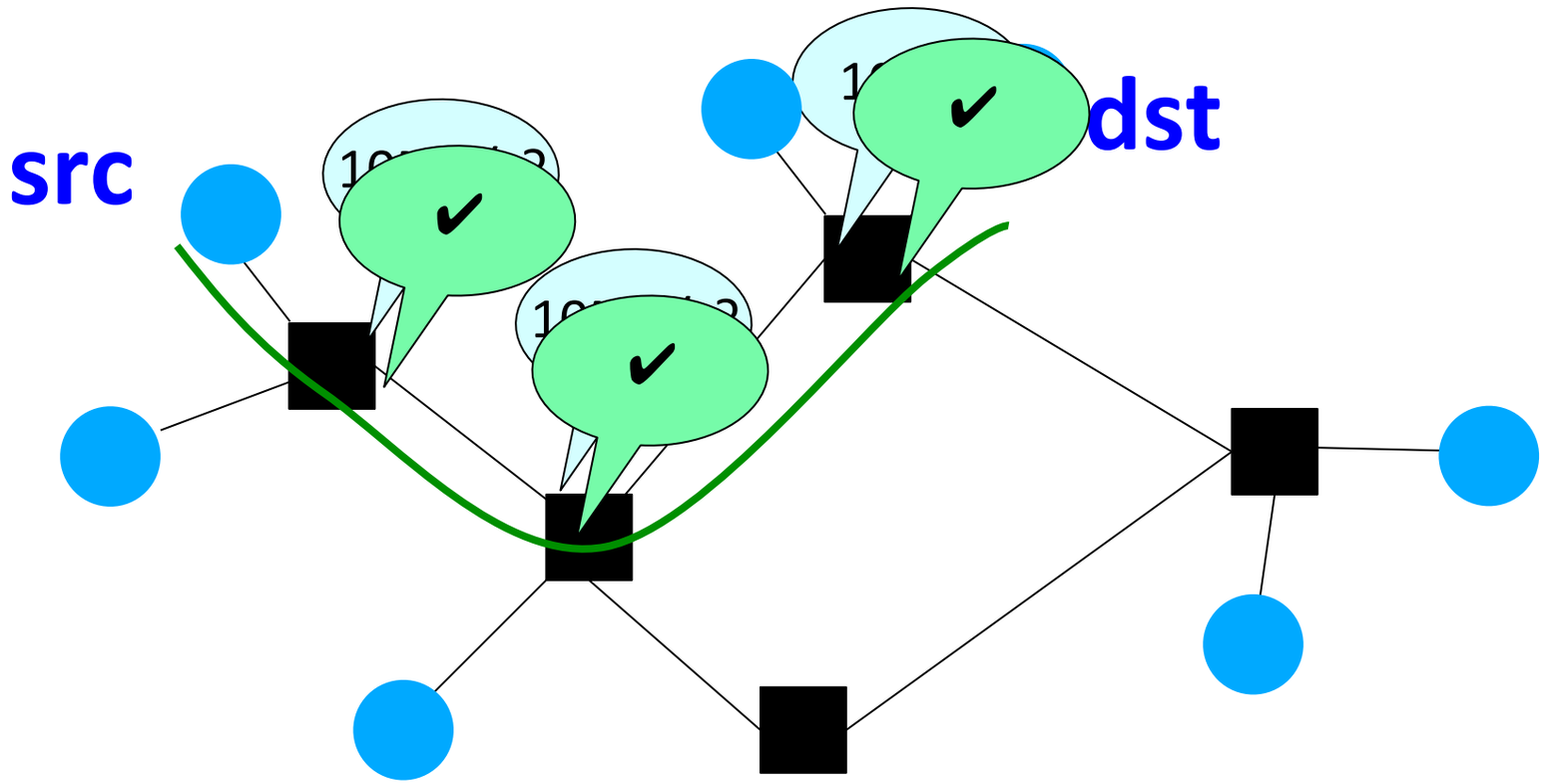
Two approaches to sharing

- Packet switching
  - packets treated on demand
  - admission control: per packet

- Circuit switching
  - resources reserved per active "connection"
  - admission control: per connection

- A hybrid: virtual circuits
  - emulating “circuit” switching with packets (see text)

Circuit Switching



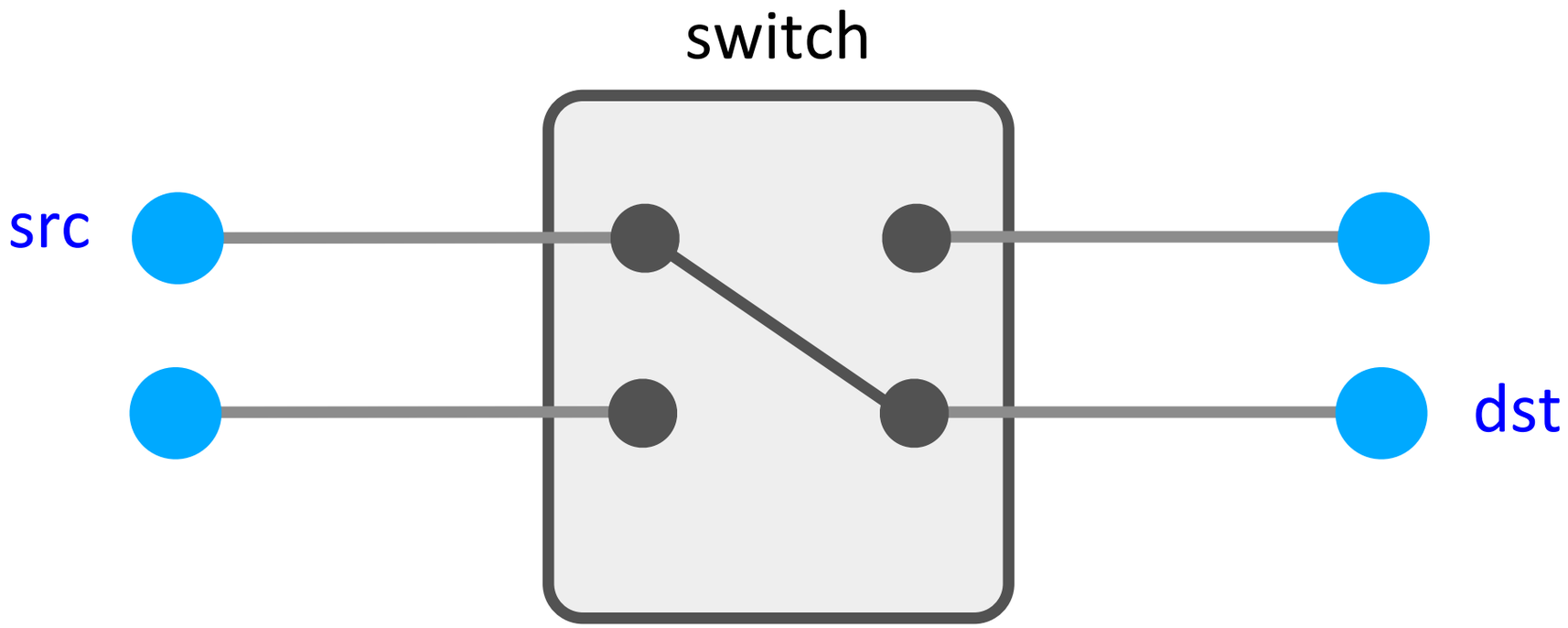
(1) src sends a reservation request to dst

(2) Switches “establish a circuit”

(3) src starts sending data

(4) src sends a “teardown circuit” message

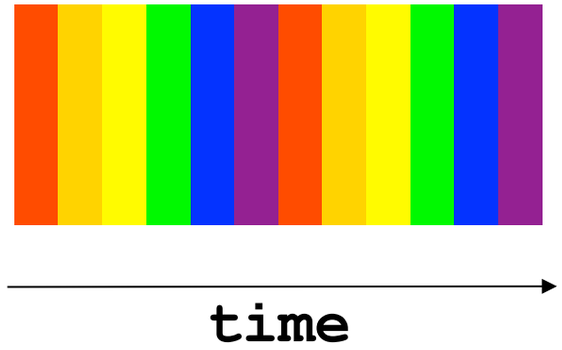
Circuit Switching



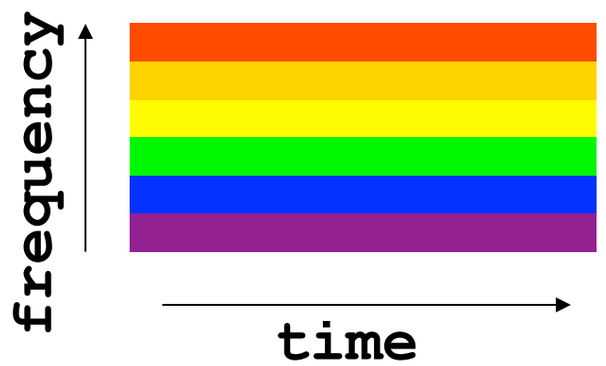
Reservation establishes a “circuit” within a switch

Many kinds of “circuits”

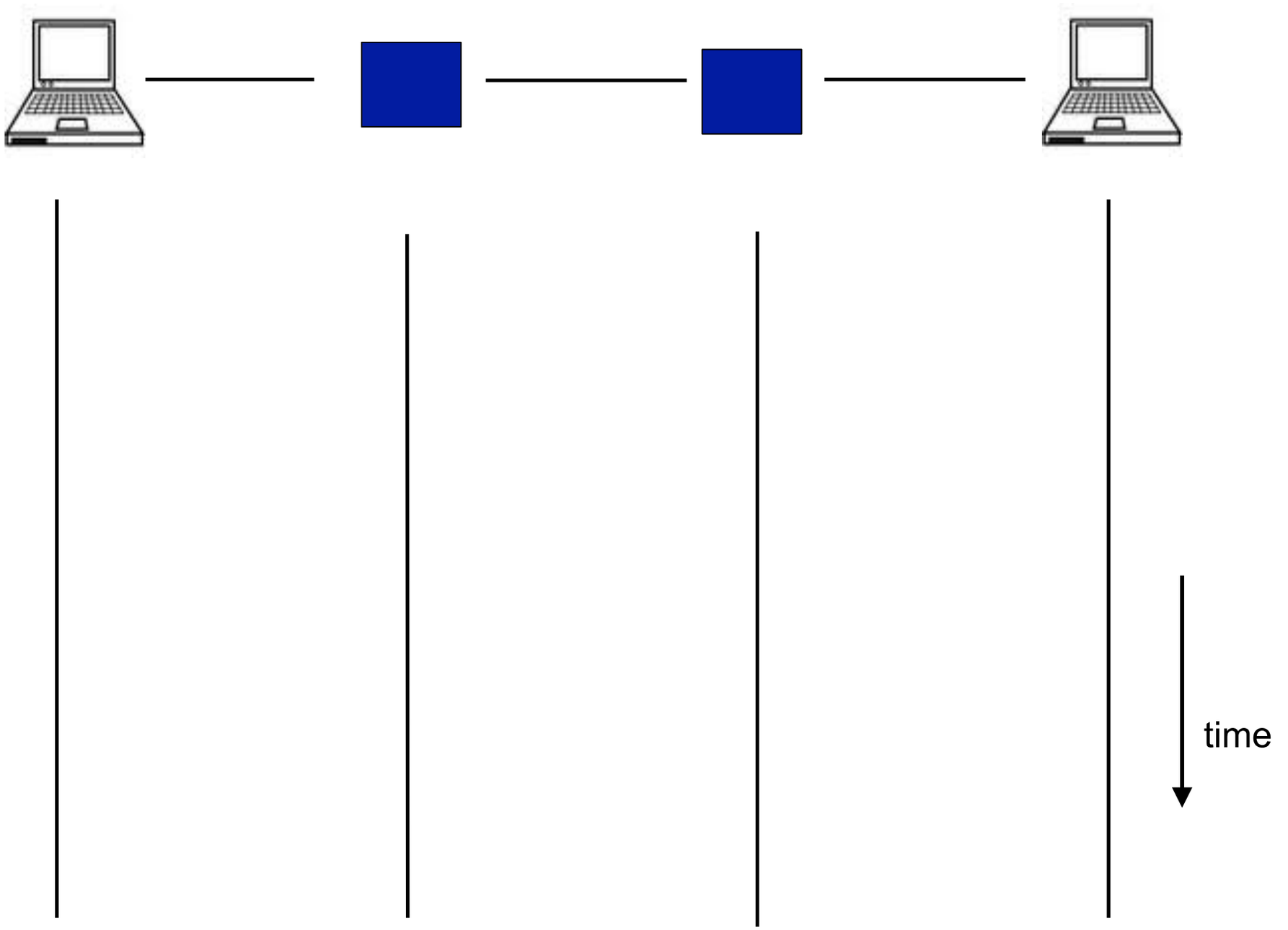
- Time division multiplexing
  - divide time in time slots
  - separate time slot per circuit



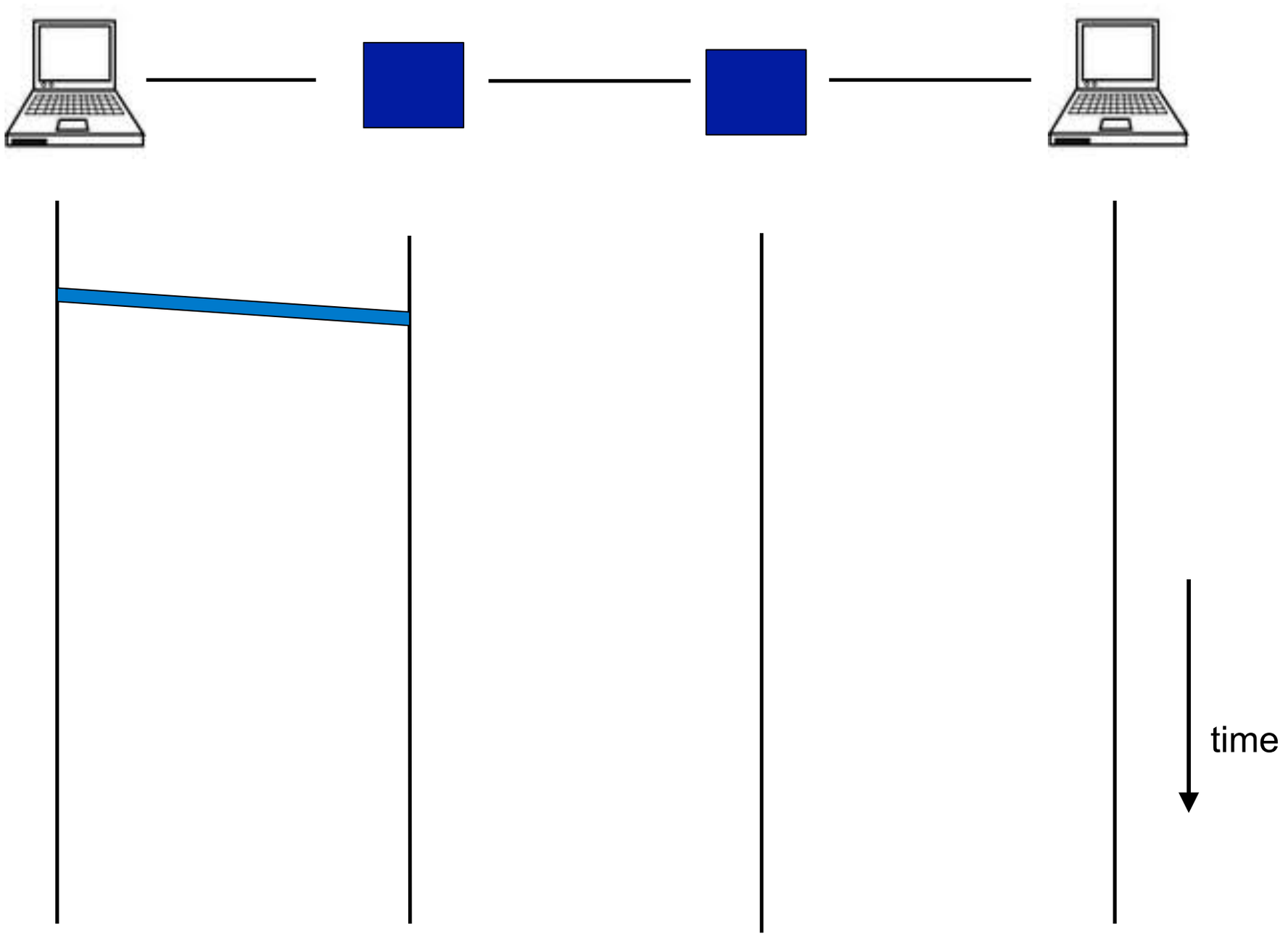
- Frequency division multiplexing
  - divide frequency spectrum in frequency bands
  - separate frequency band per circuit



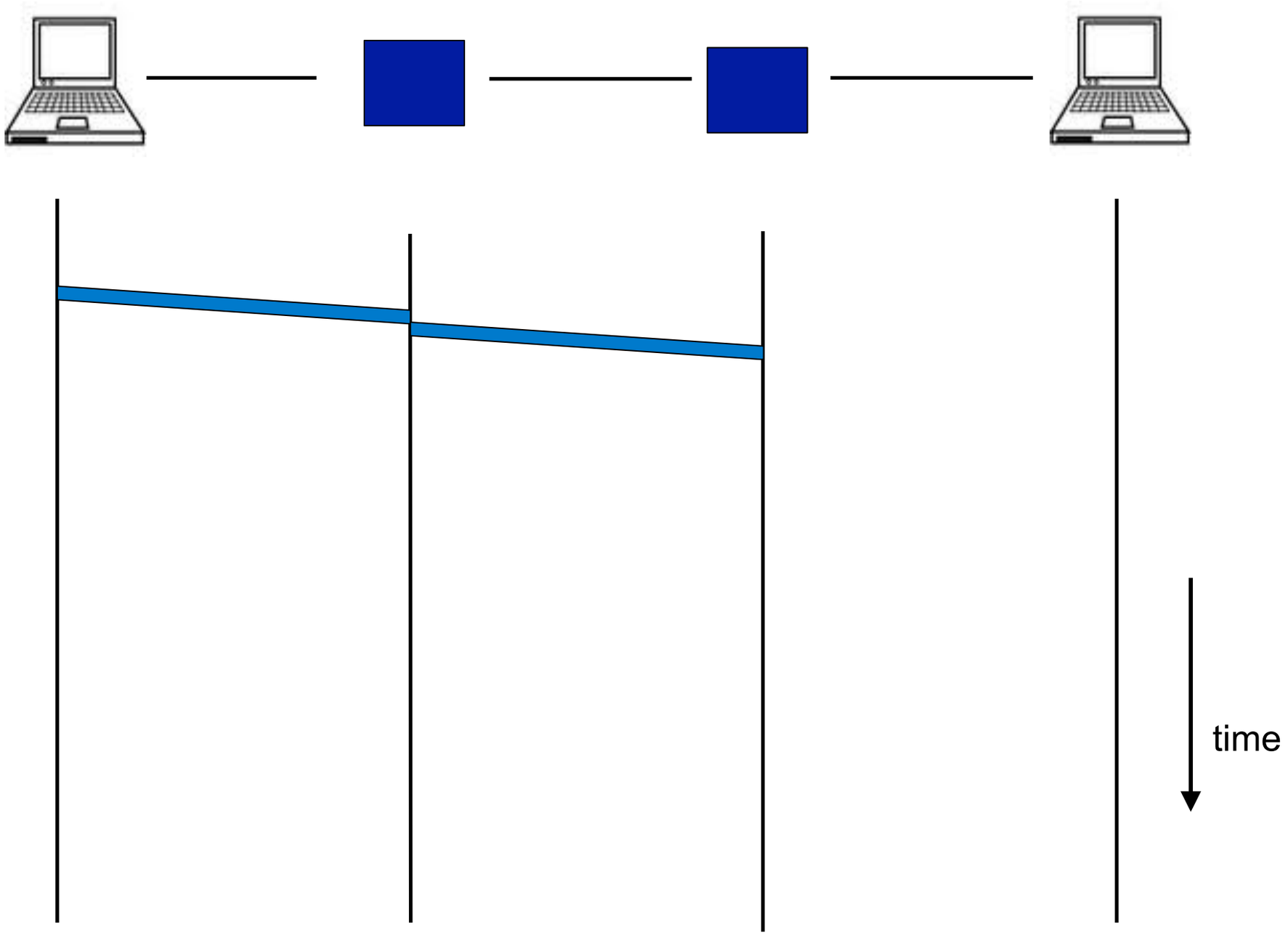
Timing in Circuit Switching



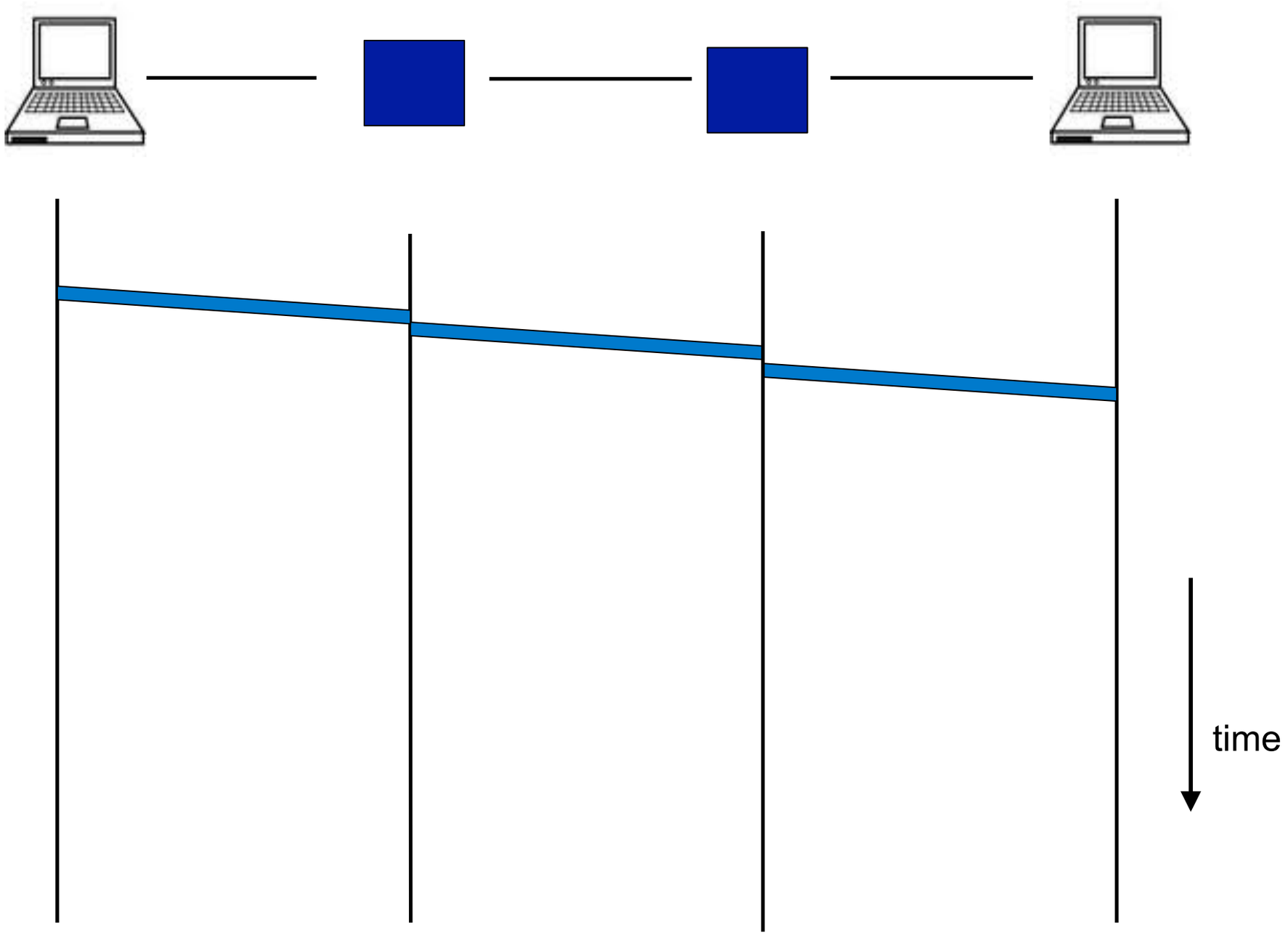
Timing in Circuit Switching



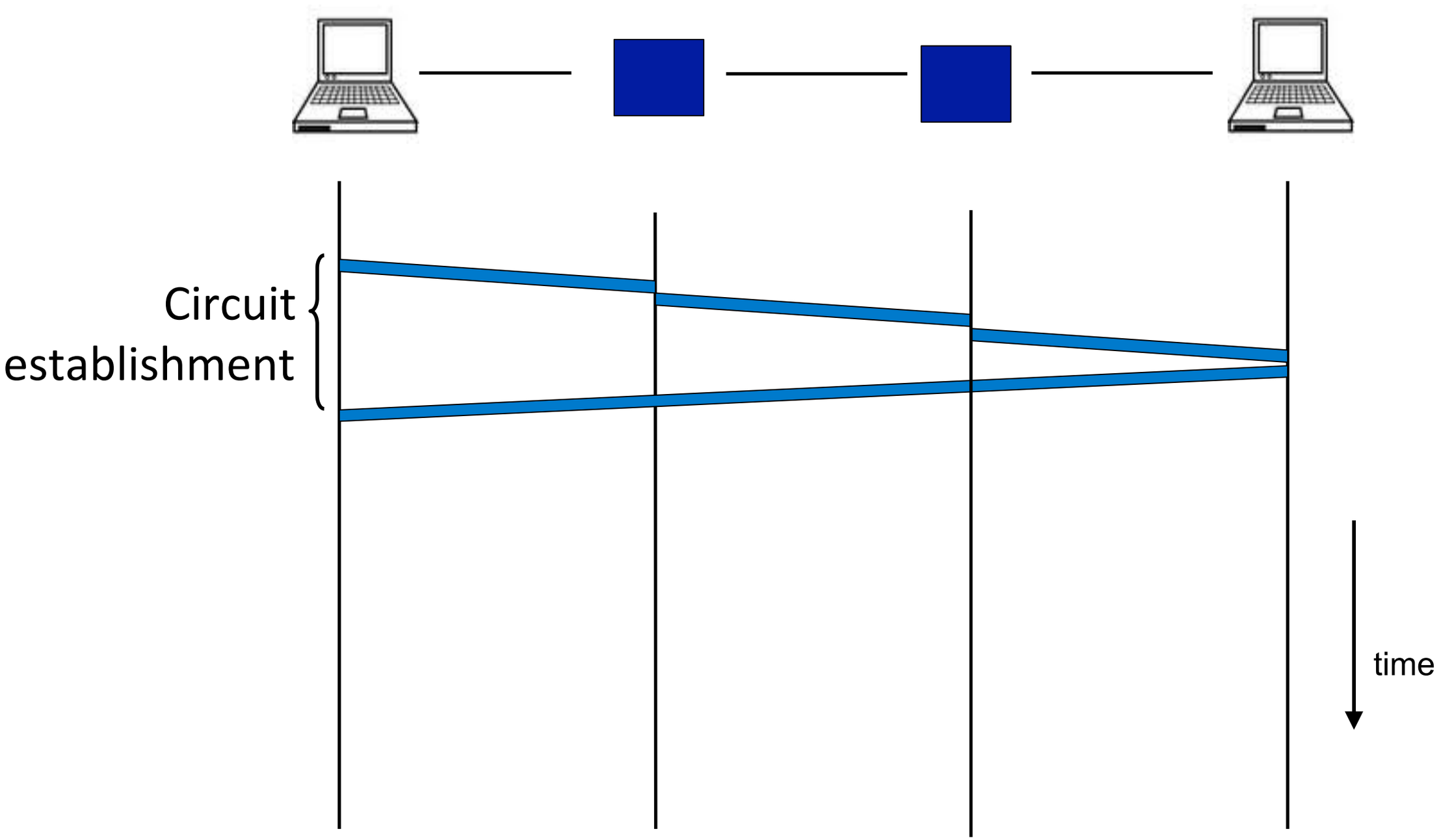
Timing in Circuit Switching



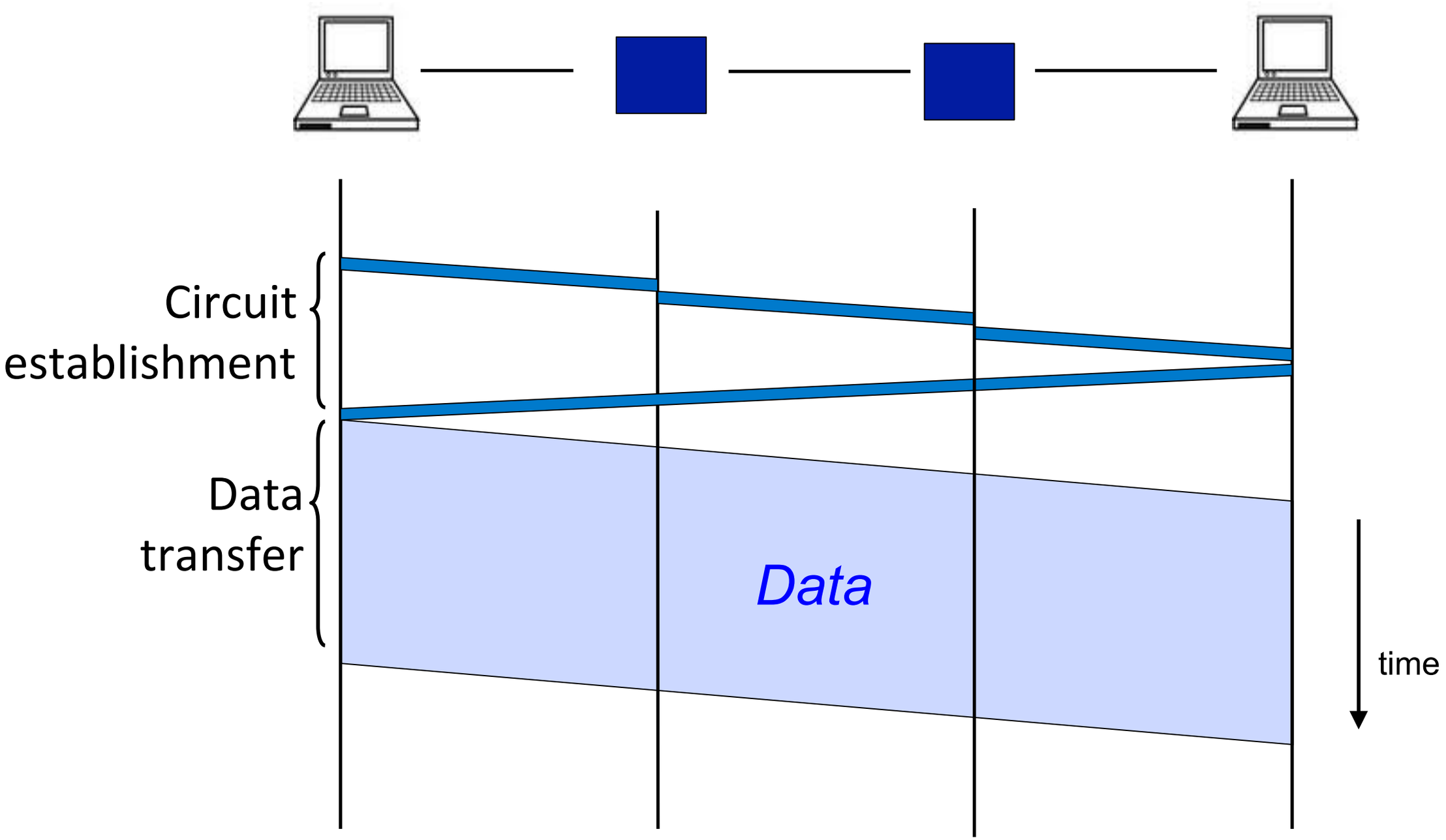
Timing in Circuit Switching



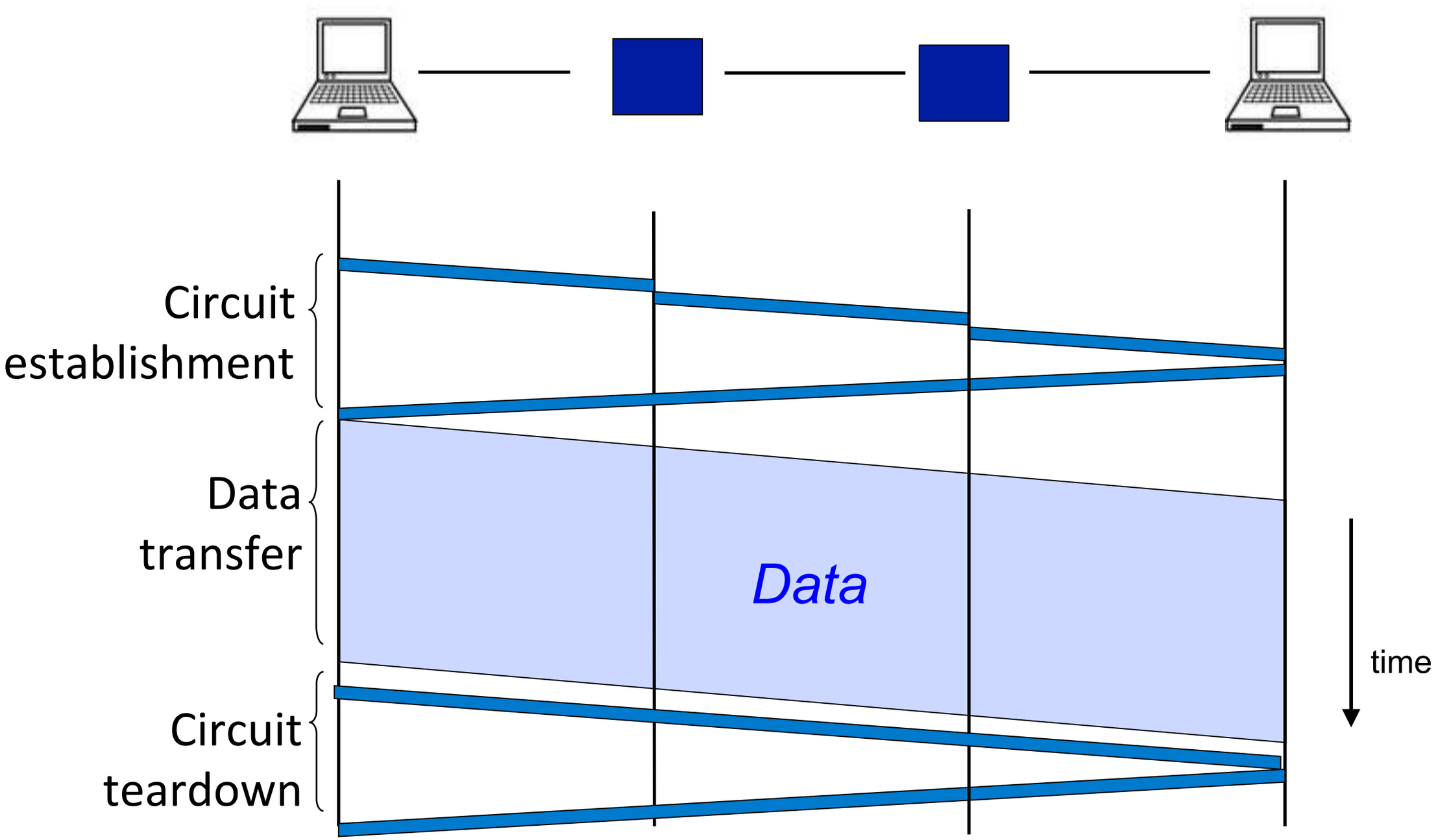
Timing in Circuit Switching



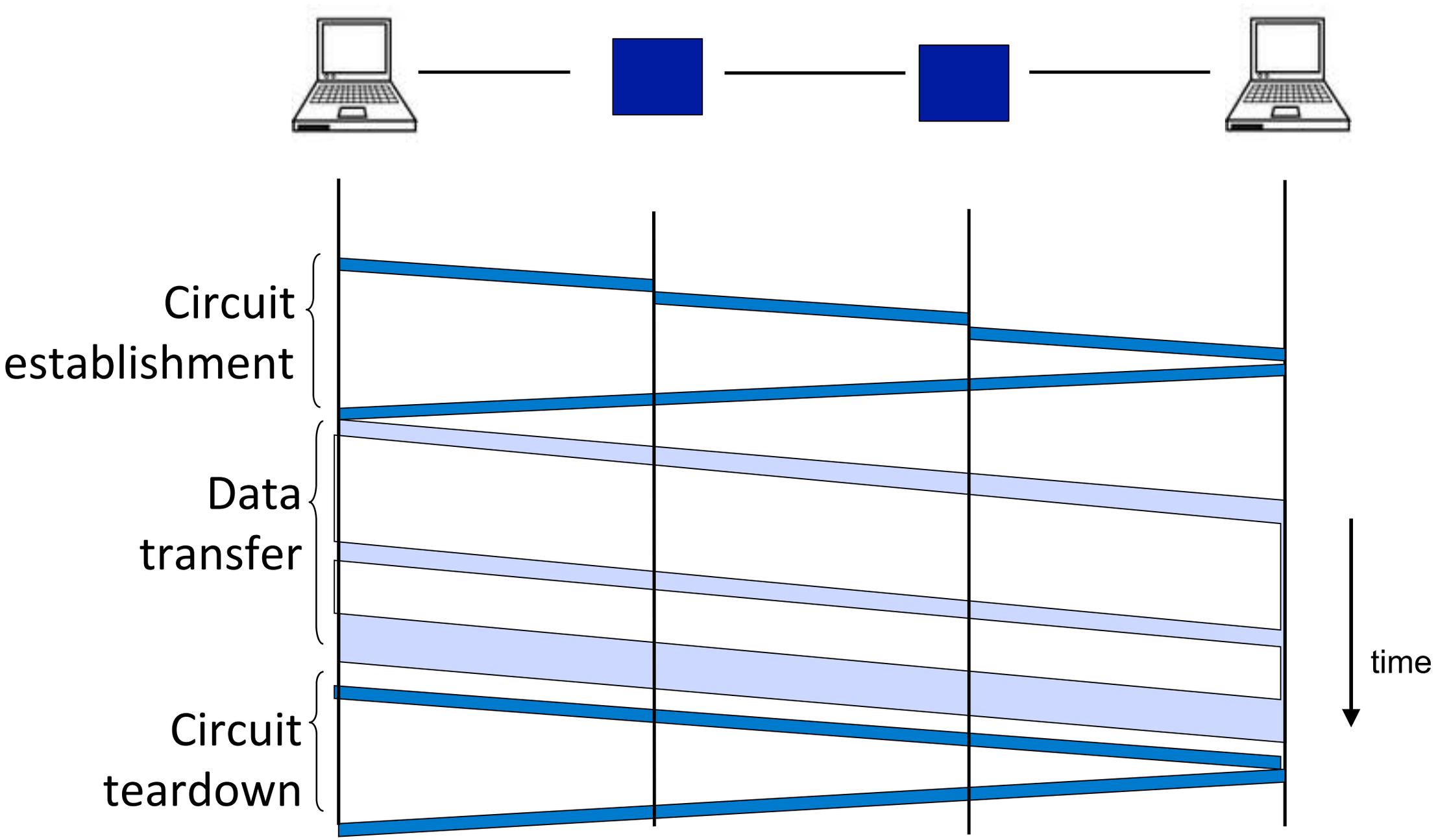
Timing in Circuit Switching



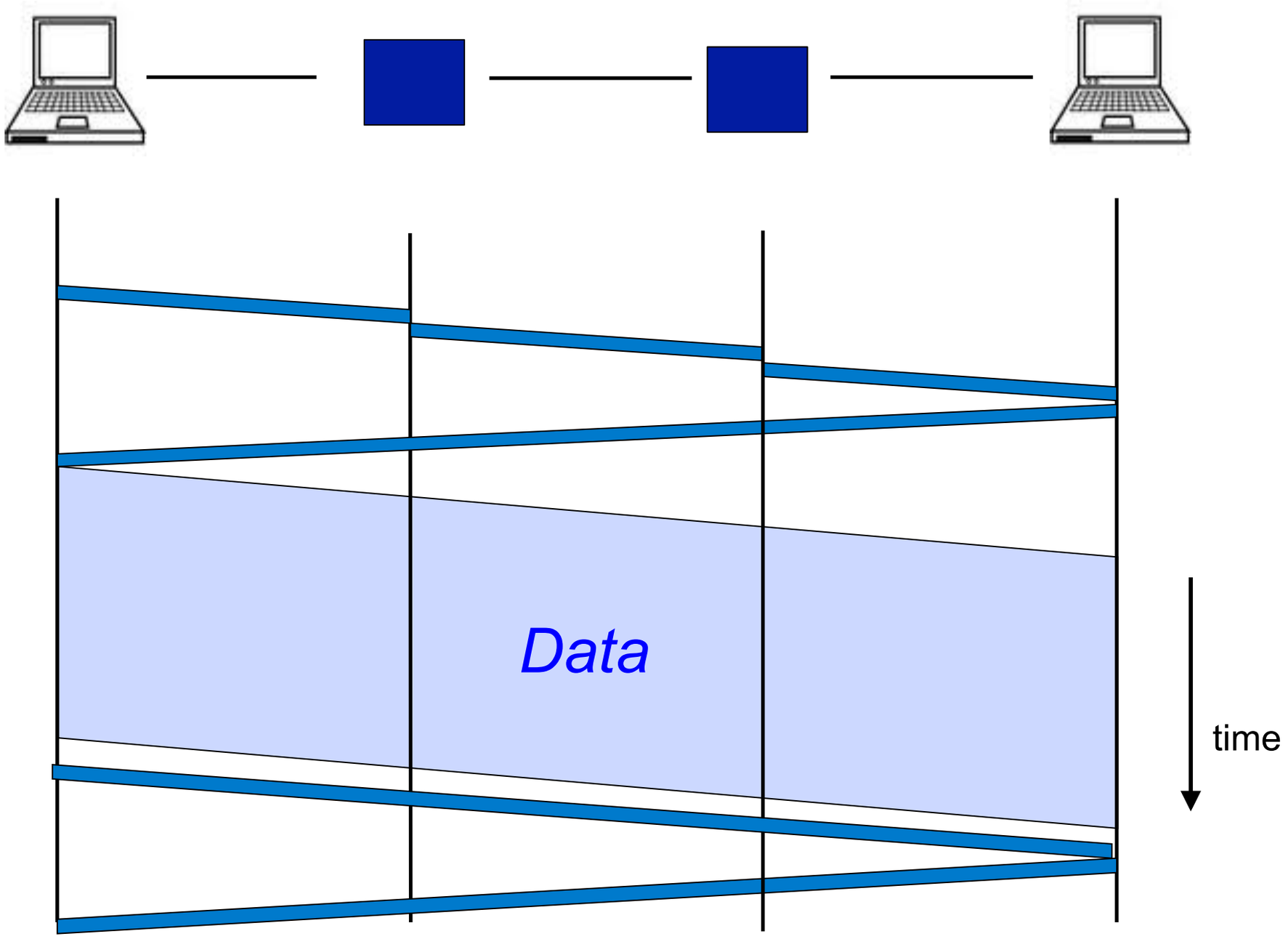
Timing in Circuit Switching



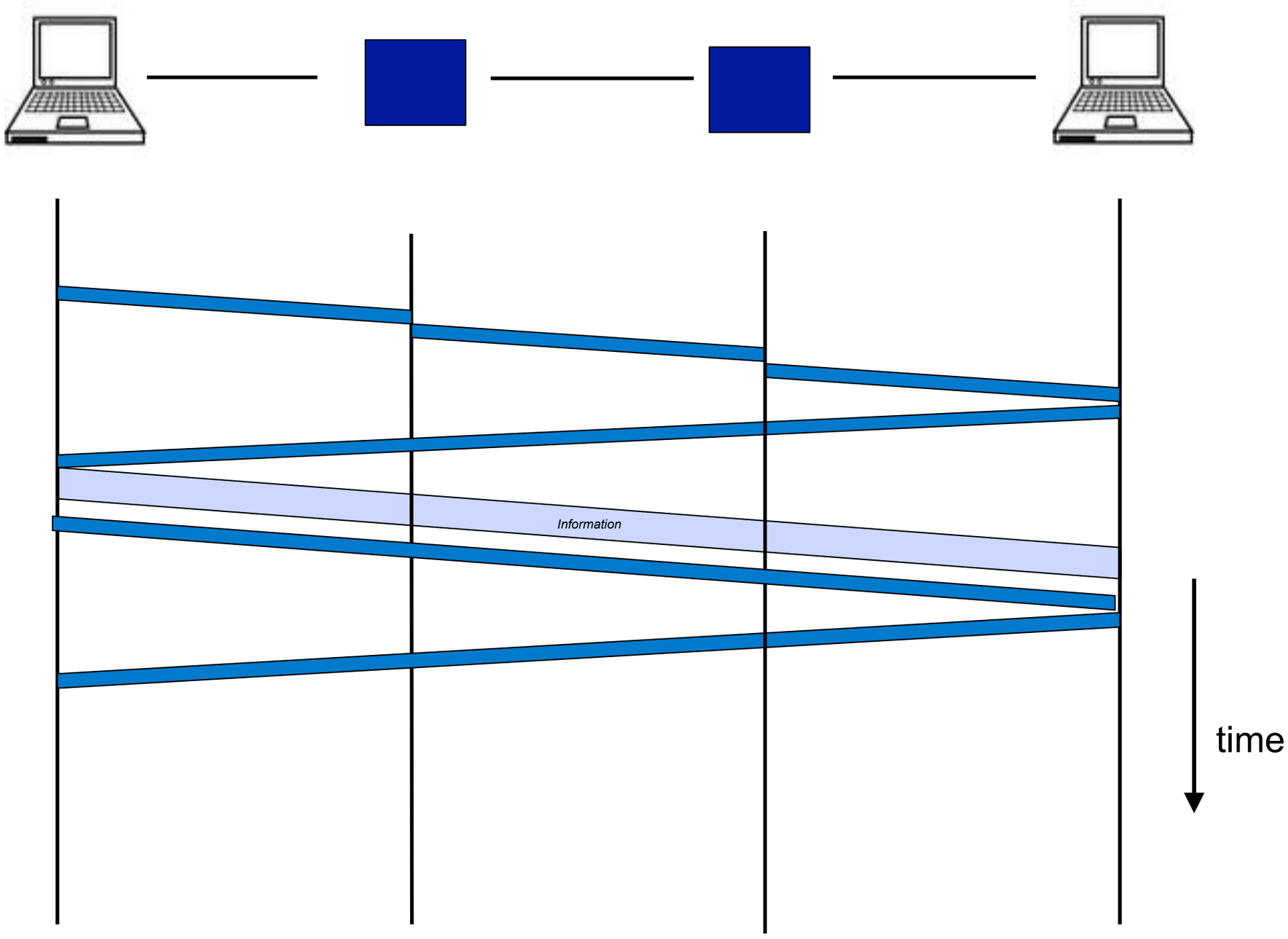
Timing in Circuit Switching



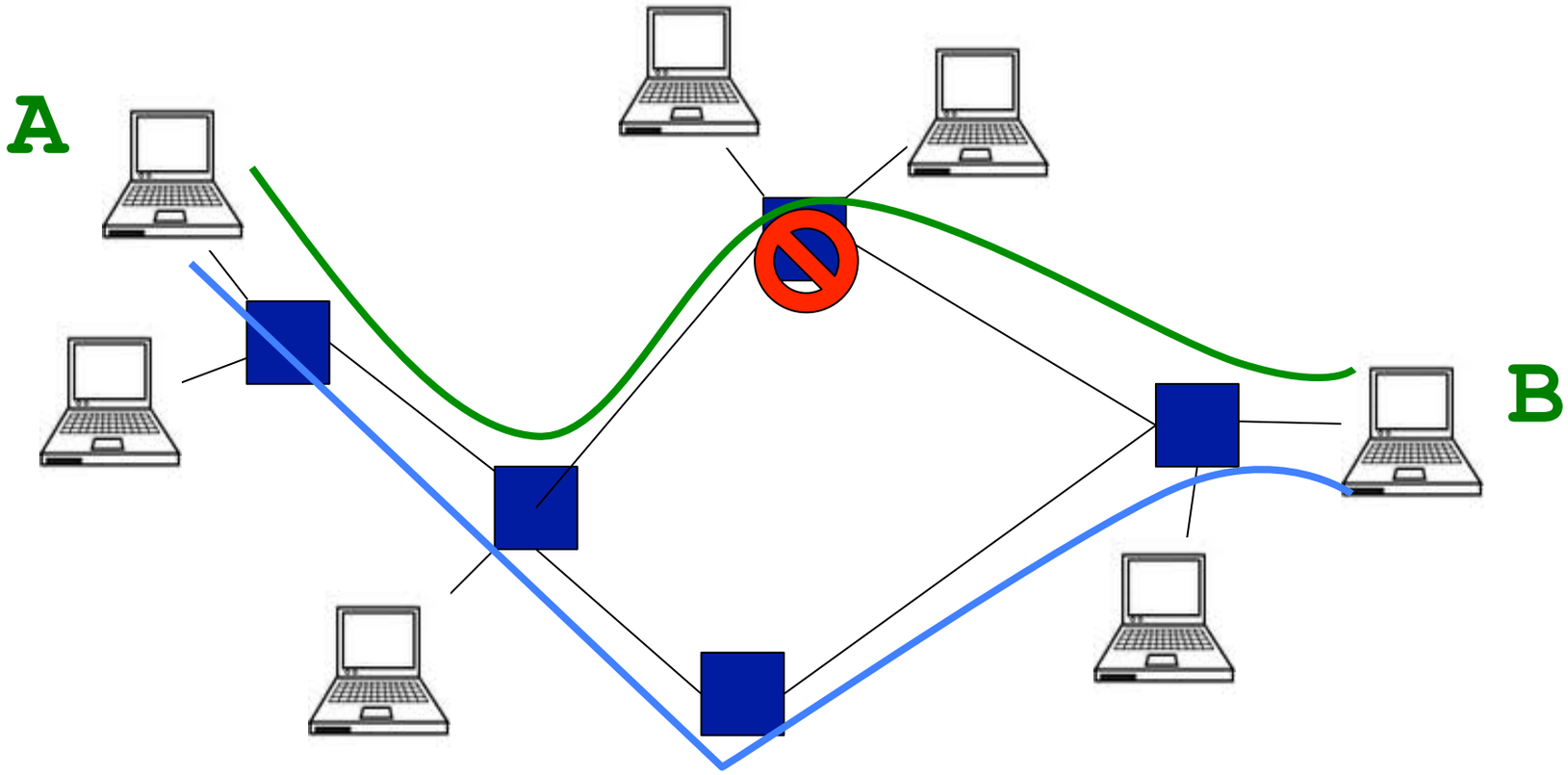
Timing in Circuit Switching



Timing in Circuit Switching



Circuit Switching



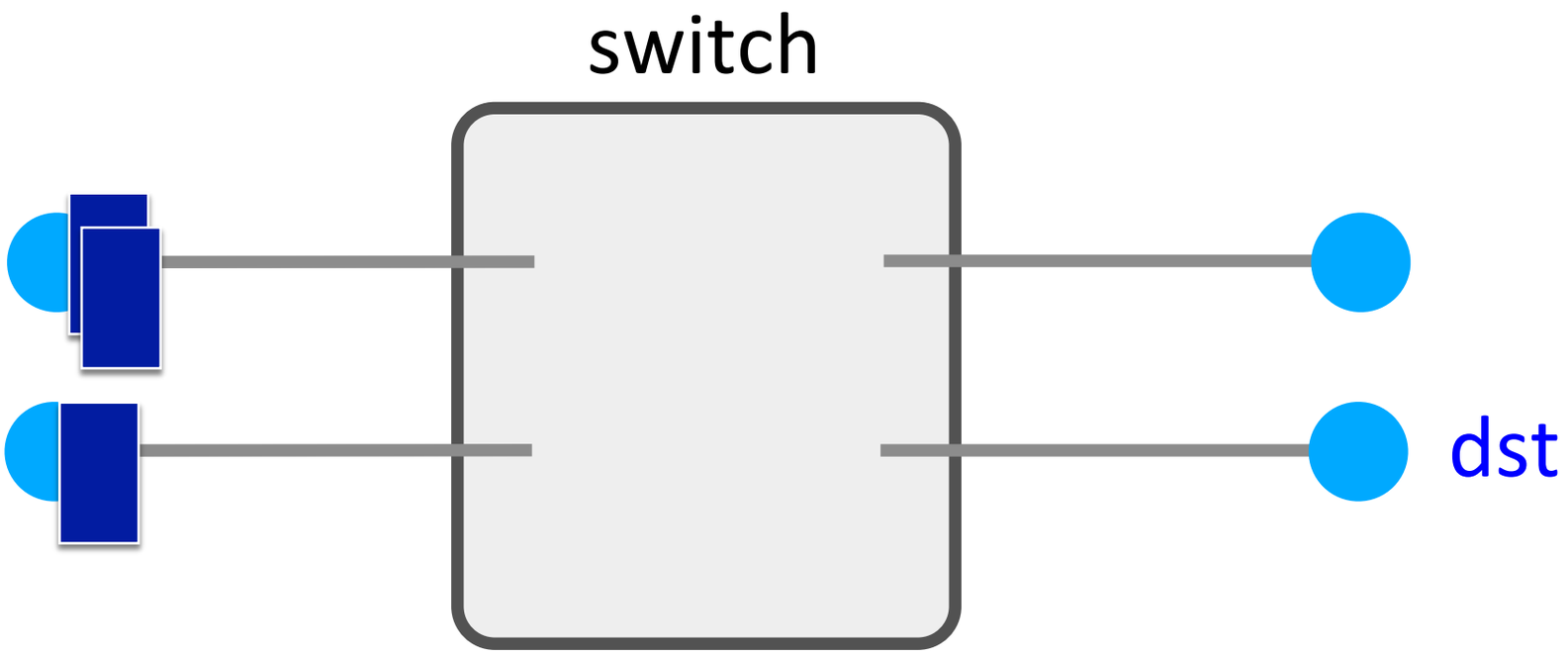
Circuit switching doesn’t “route around trouble”

Circuit Switching

- Pros
  - *predictable performance*
  - *simple/fast switching (once circuit established)*

- Cons
  - *complexity of circuit setup/teardown*
  - *inefficient when traffic is bursty*
  - *circuit setup adds delay*
  - *switch fails → its circuit(s) fails*

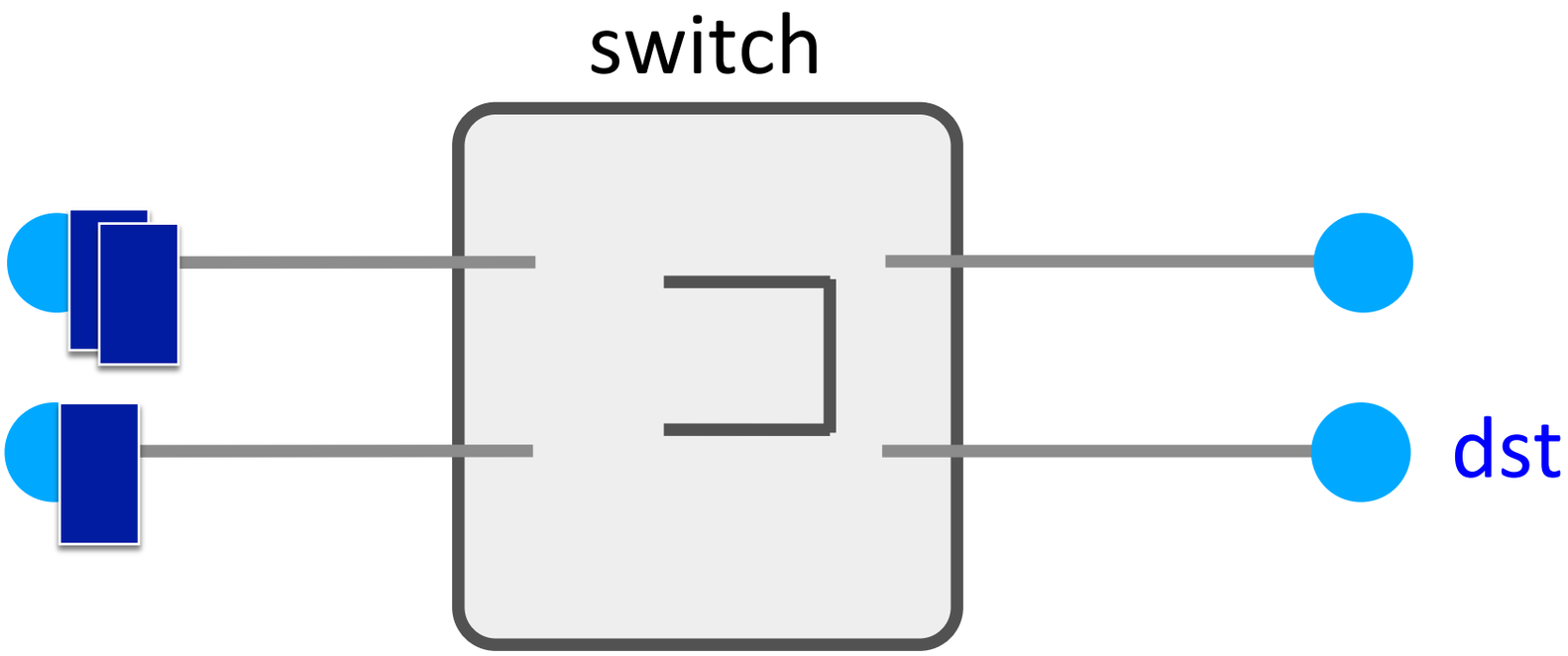
Packet switching



Each packet contains destination (dst)

Each packet treated independently

Packet switching



Each packet contains destination (dst)
Each packet treated independently

With buffers to absorb 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*

Today

- What is a network made of?
- How is it shared?
  - *will assume packet switching from here on*
- How do we evaluate a network?

Today

- What is a network made of?
- How is it shared?
- 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?*

Loss

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

Throughput

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

Delay

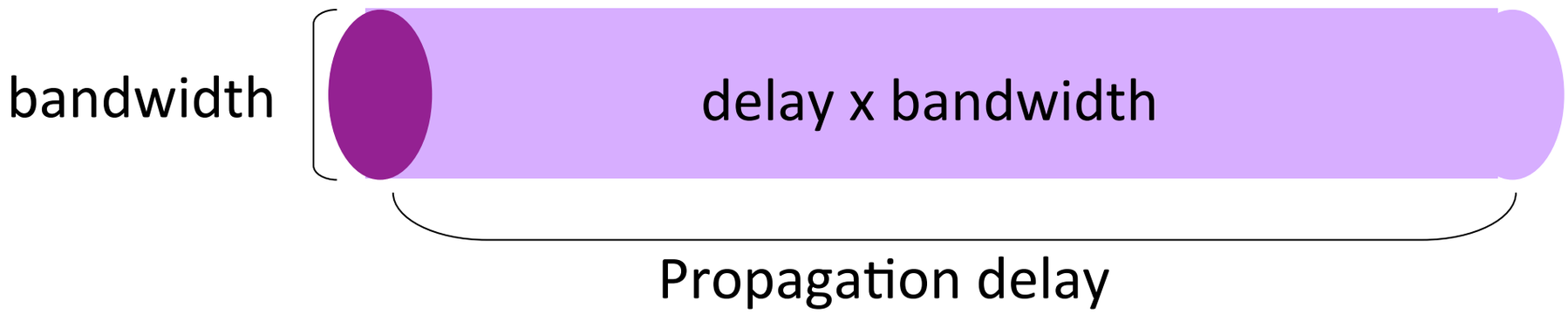
- Consists of four components
  - *transmission delay*
  - *propagation delay*

  } *due to link properties*

  - *queuing delay*
  - *processing delay*

  } *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)
- Bandwidth-Delay Product (BDP)
  - number of bits “in flight” at any time
  - BDP = bandwidth × propagation delay

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)

1. Transmission delay

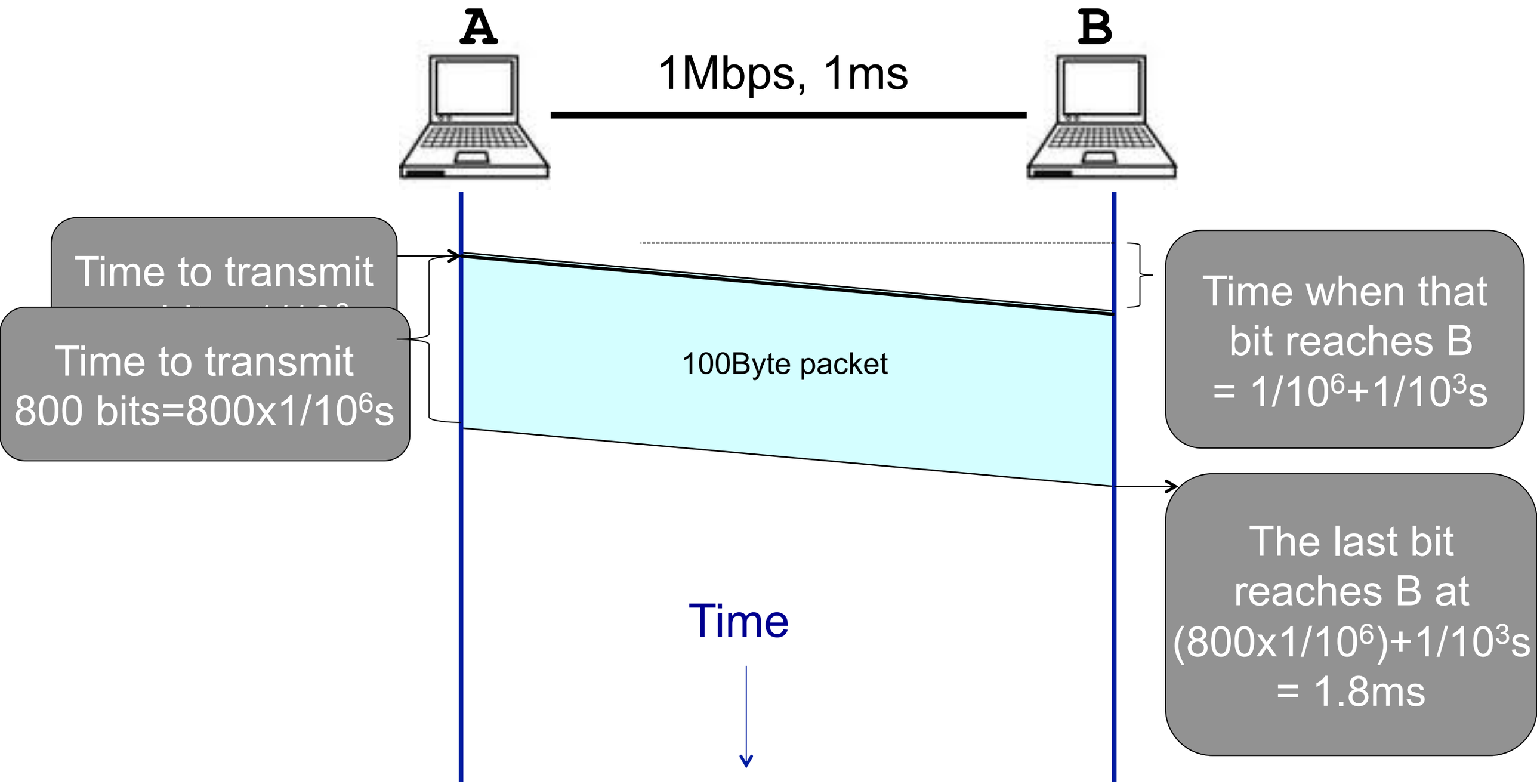
- How long does it take to push all the bits of a packet into a link?
- Packet size / Transmission rate of the link
  - *e.g. 1000 bits / 100 Mbits per sec = 10^{-5} sec*

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

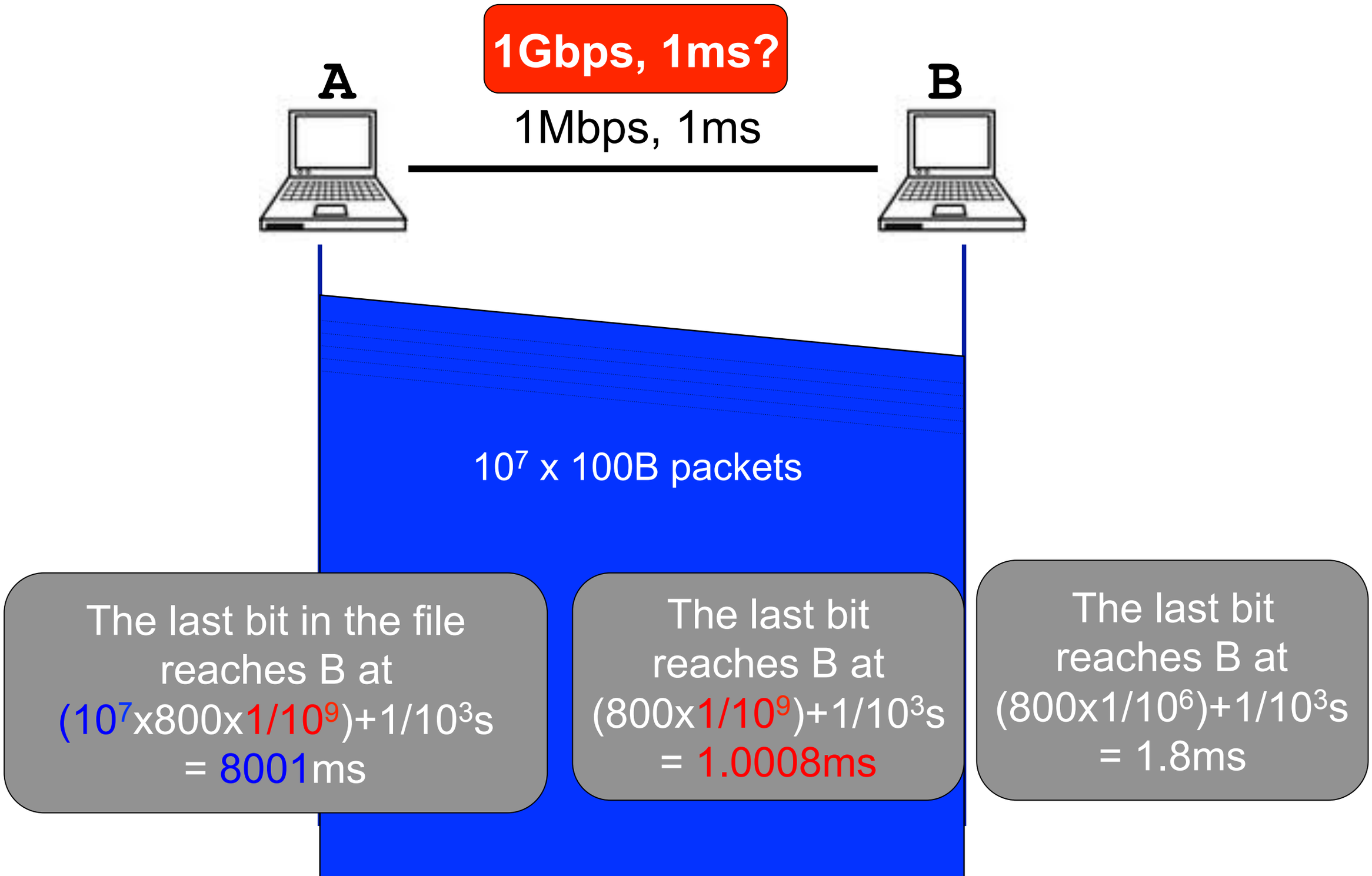
Sending 100B packets from A to B?



1GB file in 100B packets

ay

Sending 100B packets from A to B?



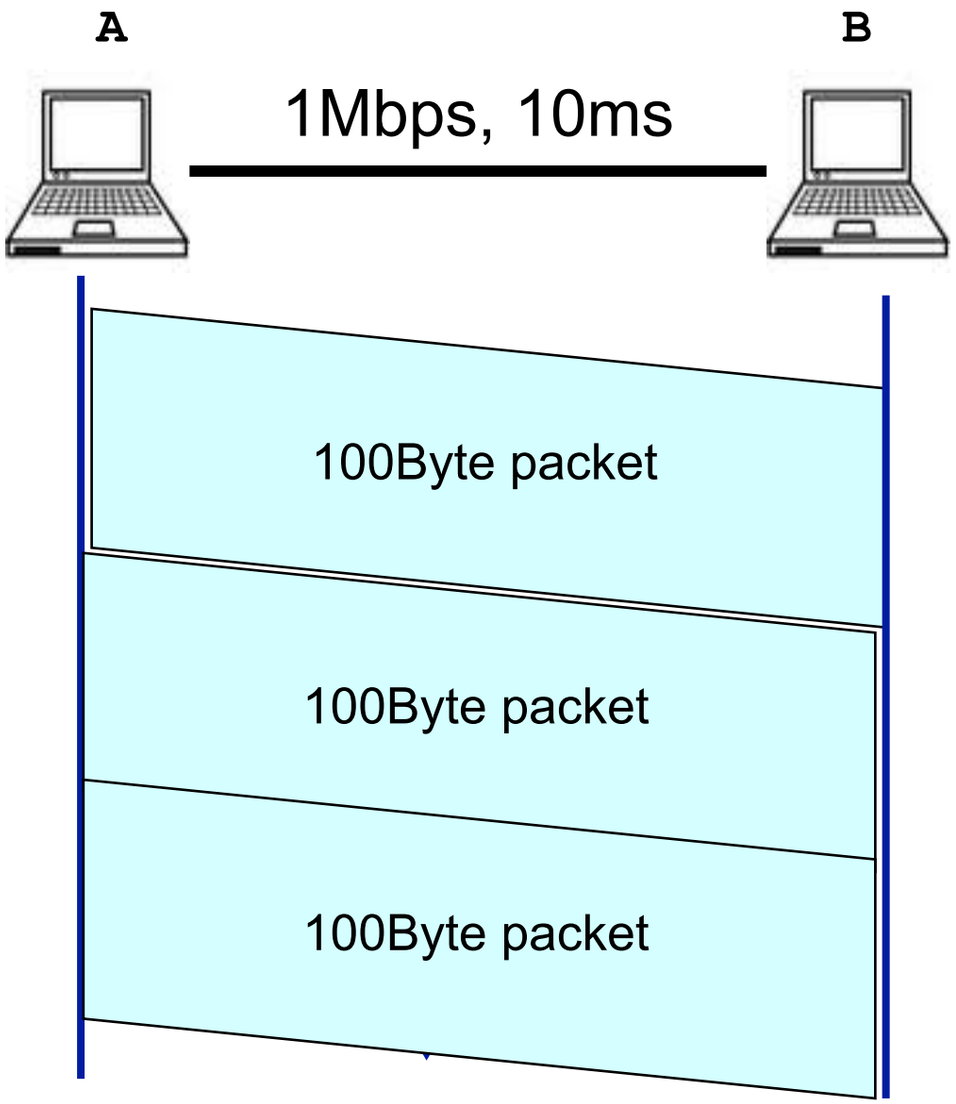
The last bit in the file reaches B at $(10^7 \times 800 \times 1/10^9)+1/10^3$s = 8001ms

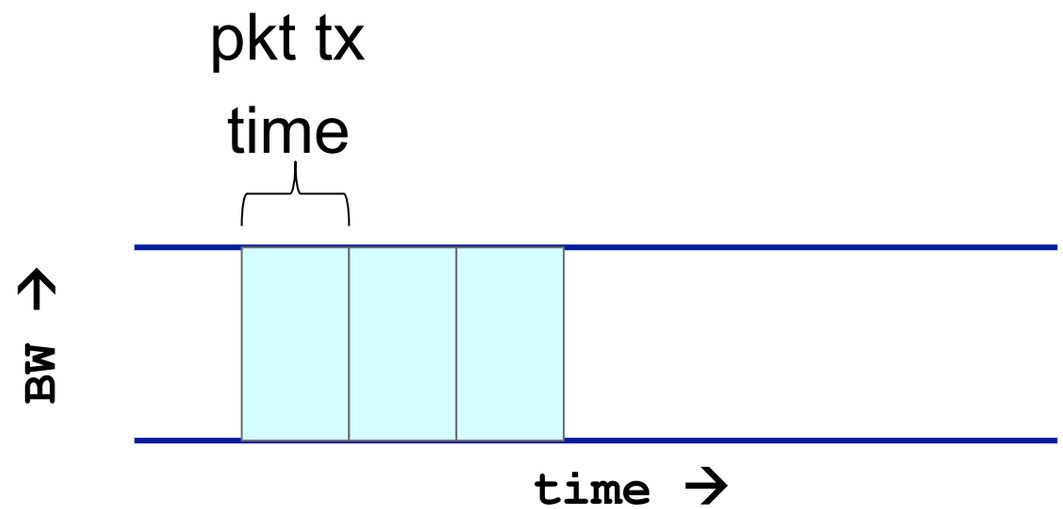
The last bit reaches B at $(800 \times 1/10^9)+1/10^3$s = 1.0008ms

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?

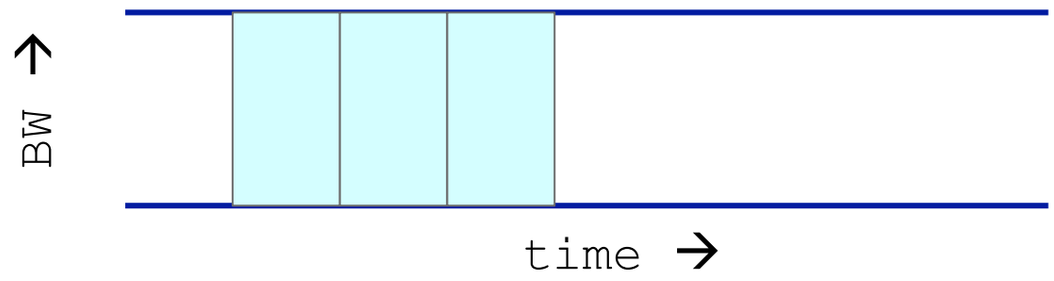




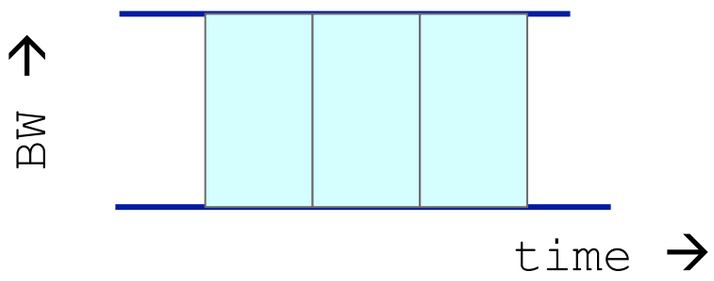
Packet Delay: The "pipe" view

Sending 100B packets from A to B?

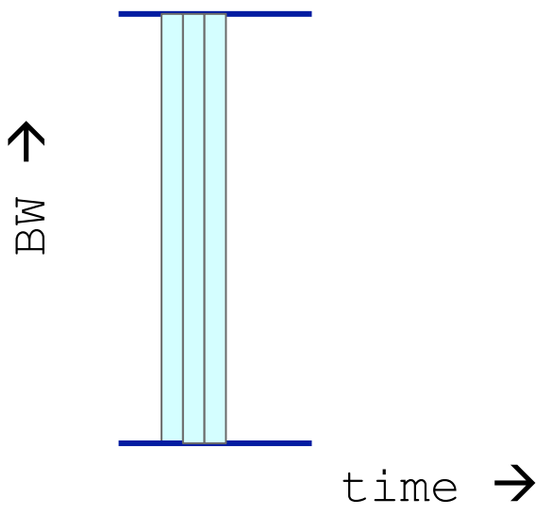
1Mbps, 10ms (BDP=10,000)



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

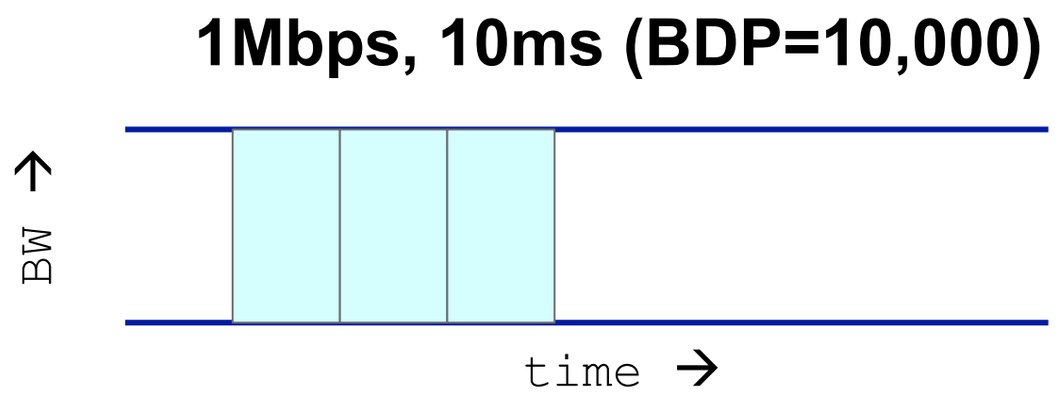


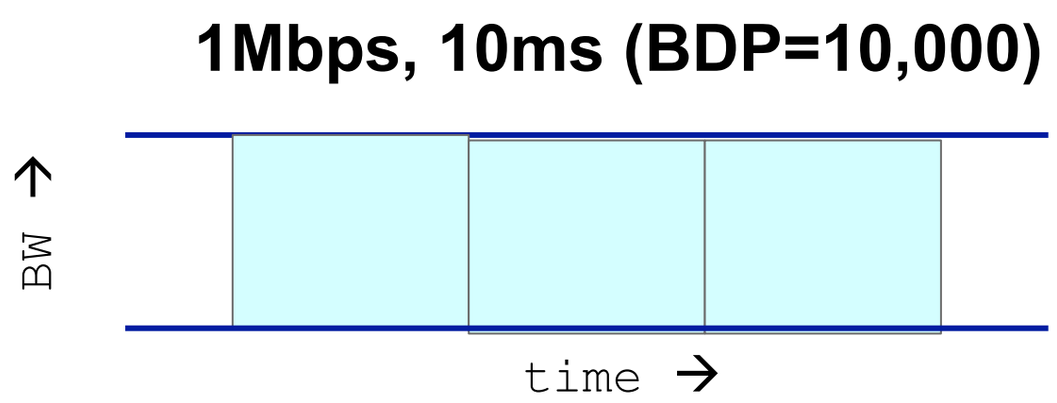
10Mbps, 1ms (BDP=10,000)



Packe 200B?: The “pipe” view

Sending 100B packets from A to B?

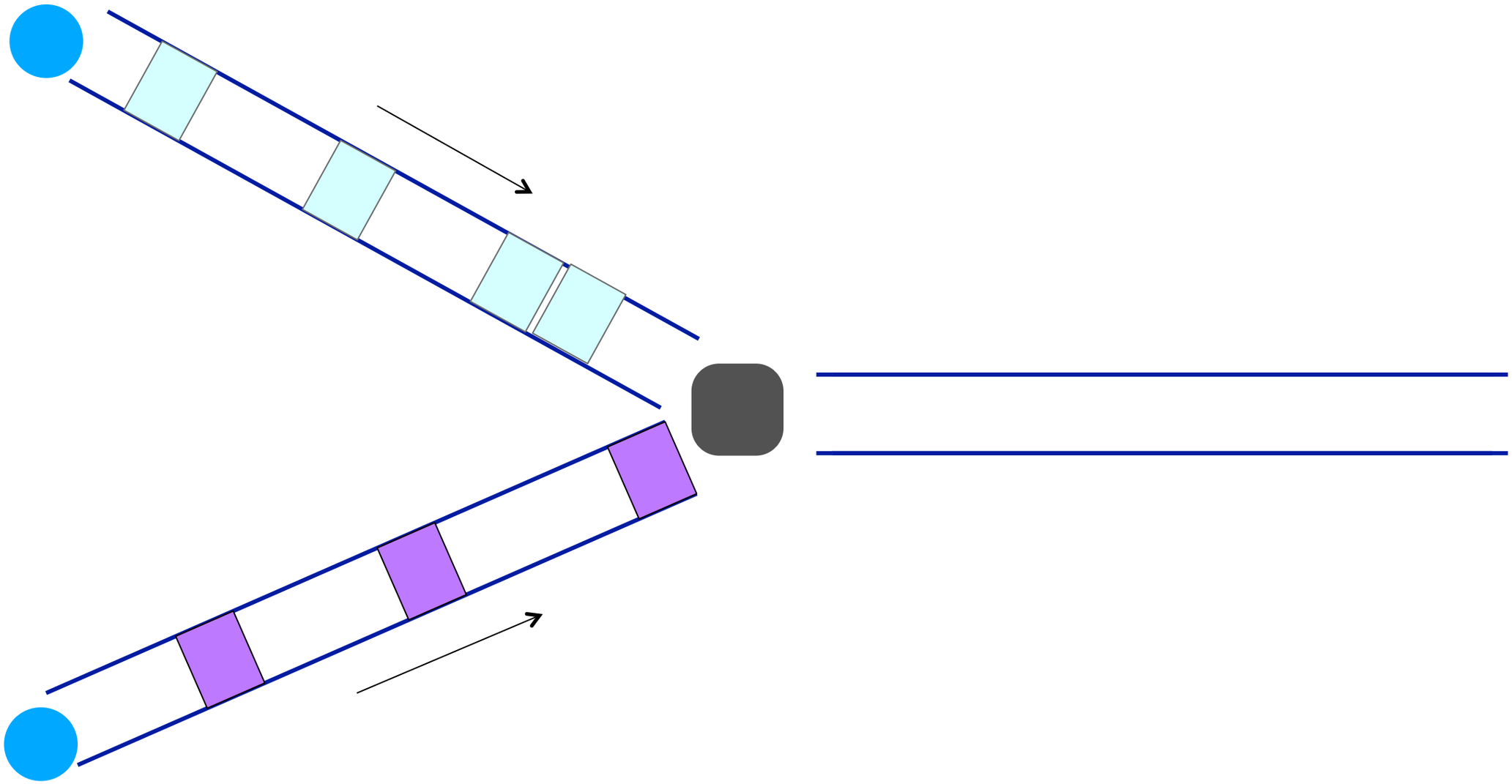




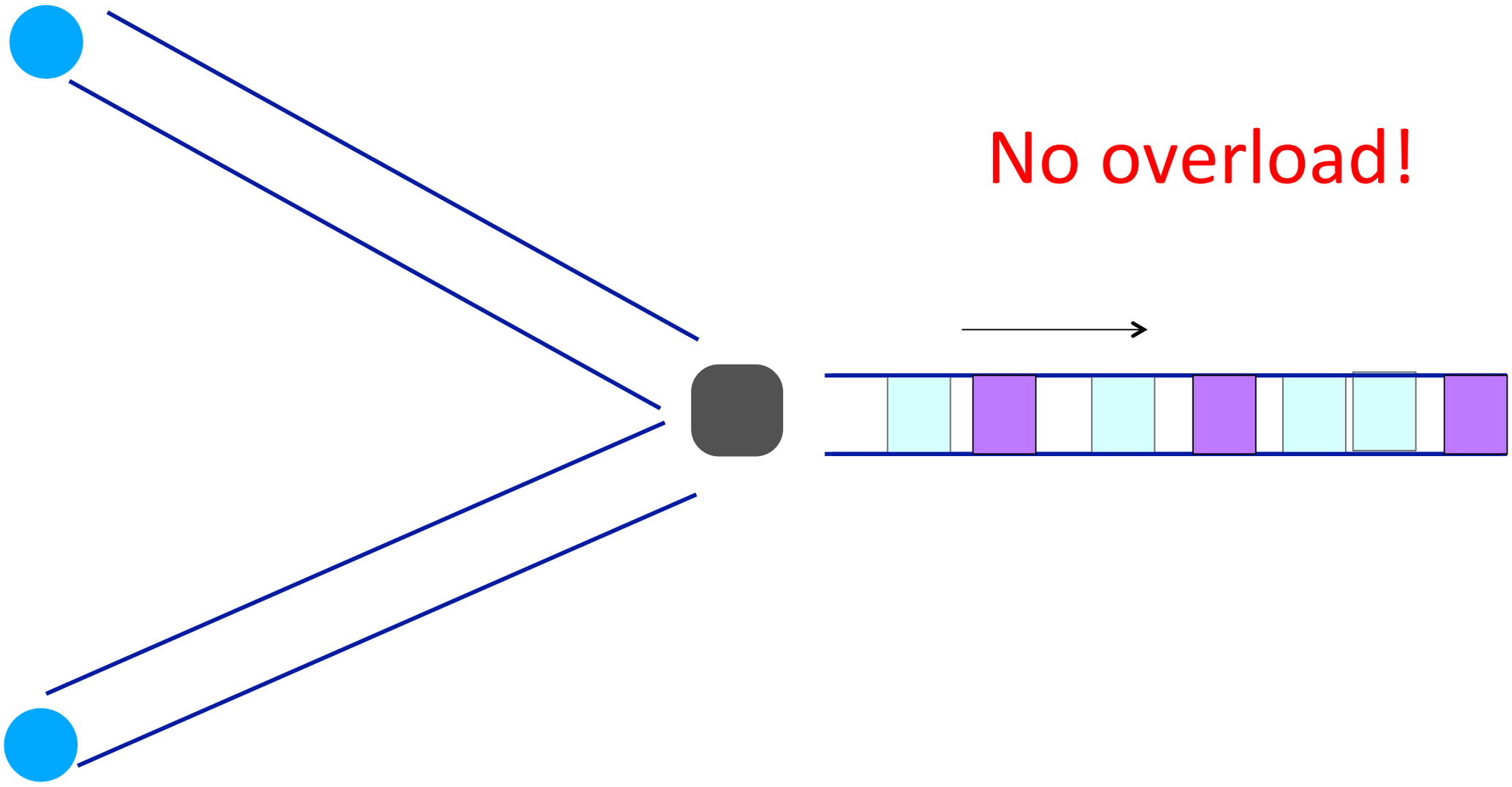
3. Queuing delay

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

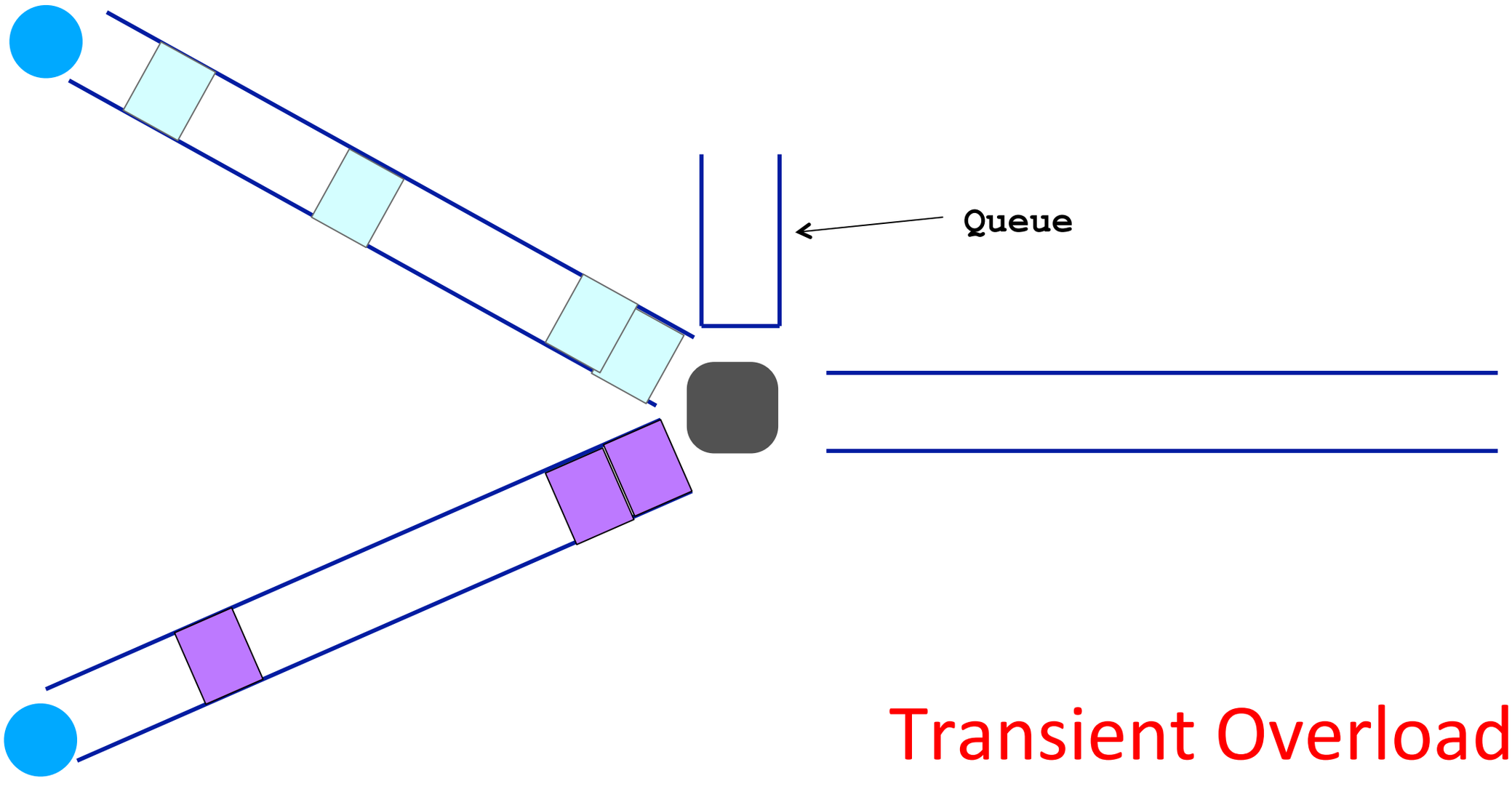
Queuing delay: “pipe” view



Queuing delay: “pipe” view



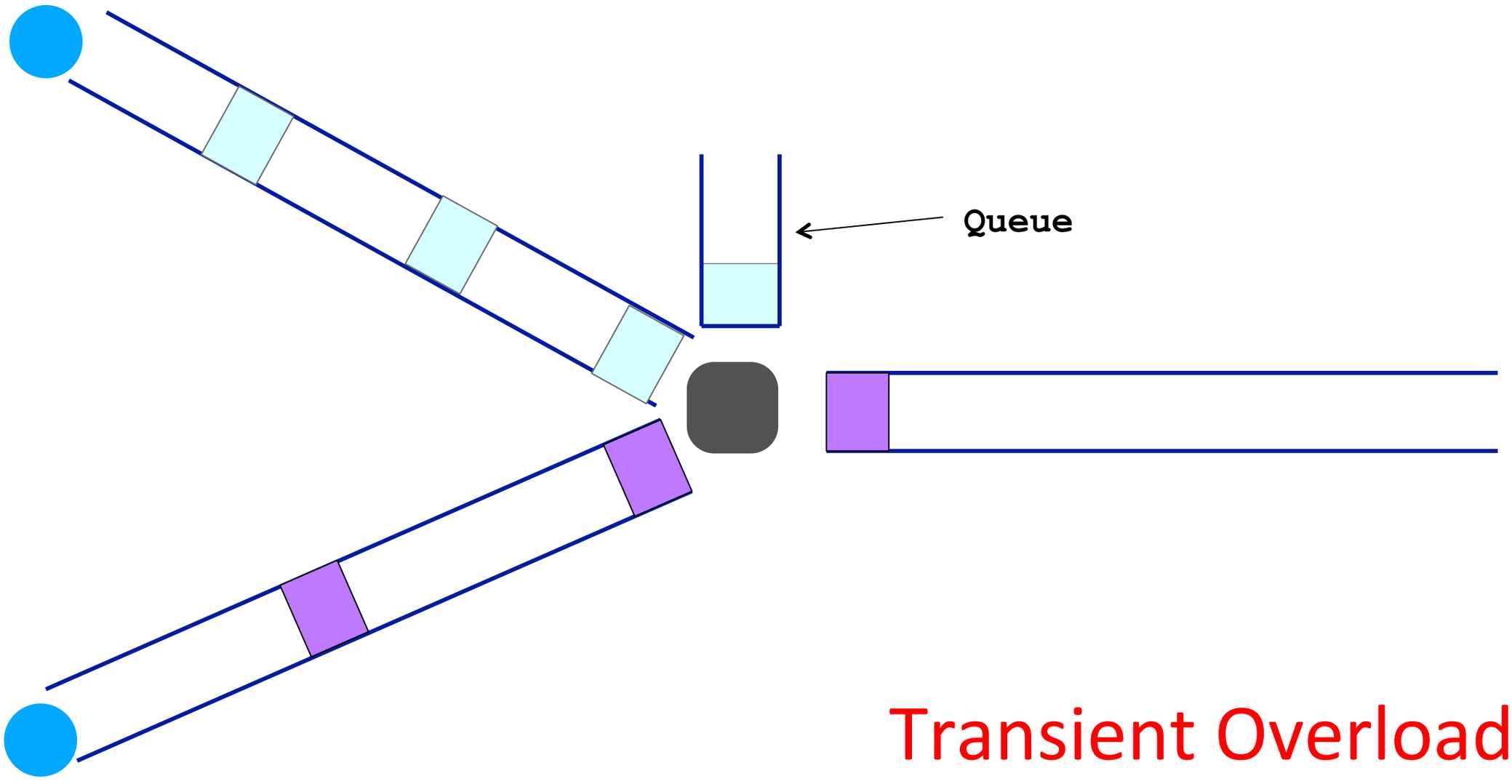
Queuing delay: “pipe” view



Transient Overload

Not a rare event!

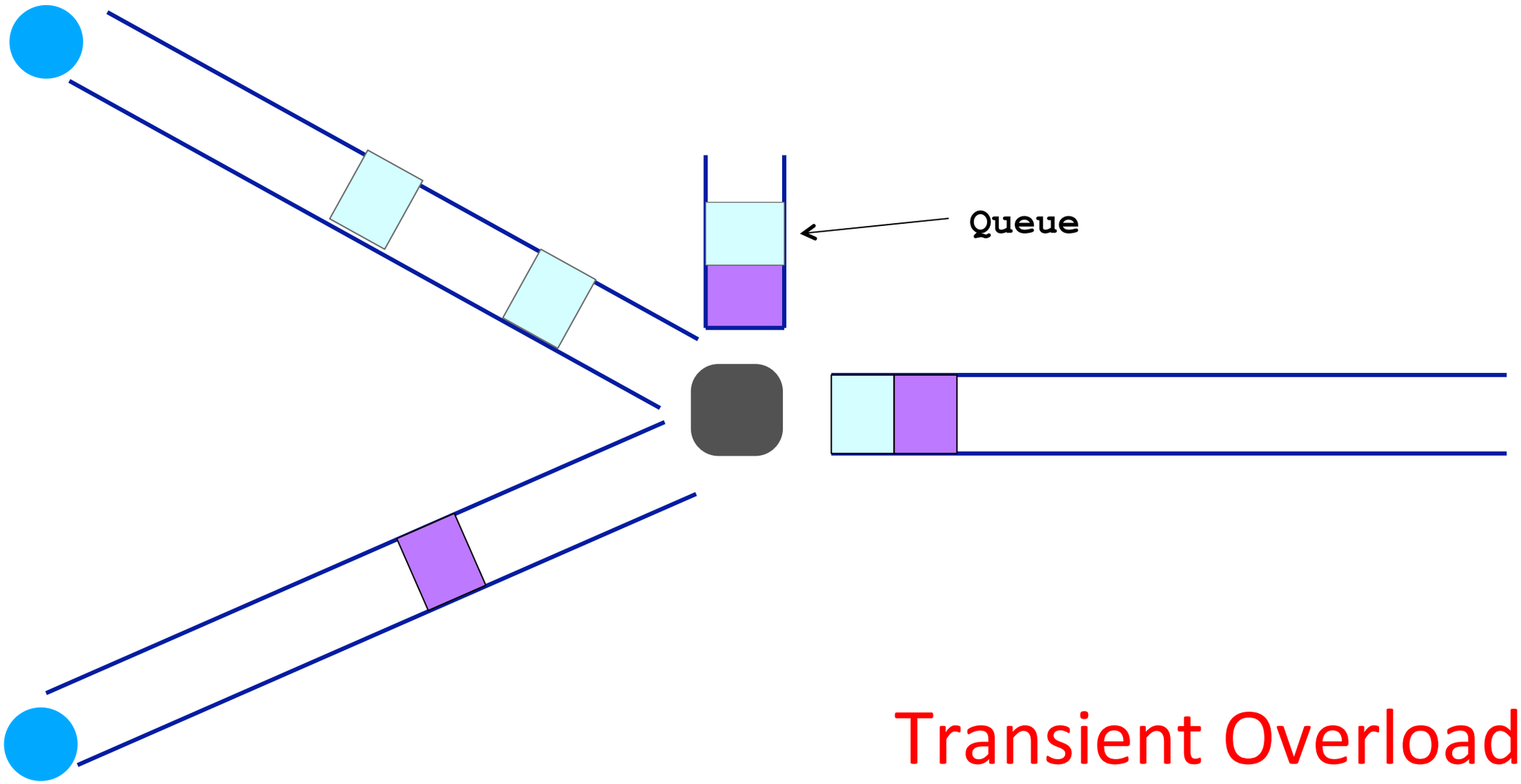
Queuing delay: “pipe” view



Transient Overload

Not a rare event!

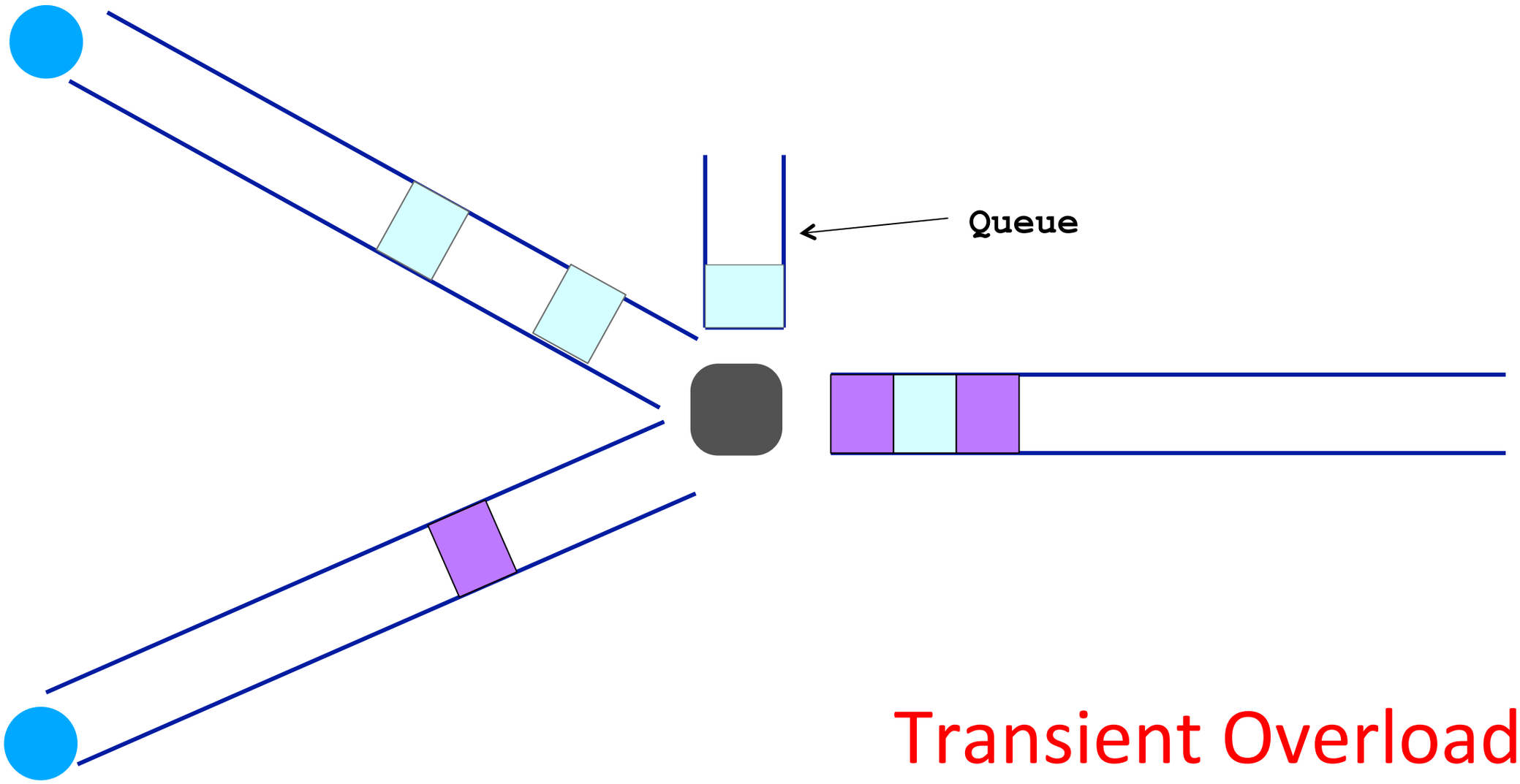
Queuing delay: “pipe” view



Transient Overload

Not a rare event!

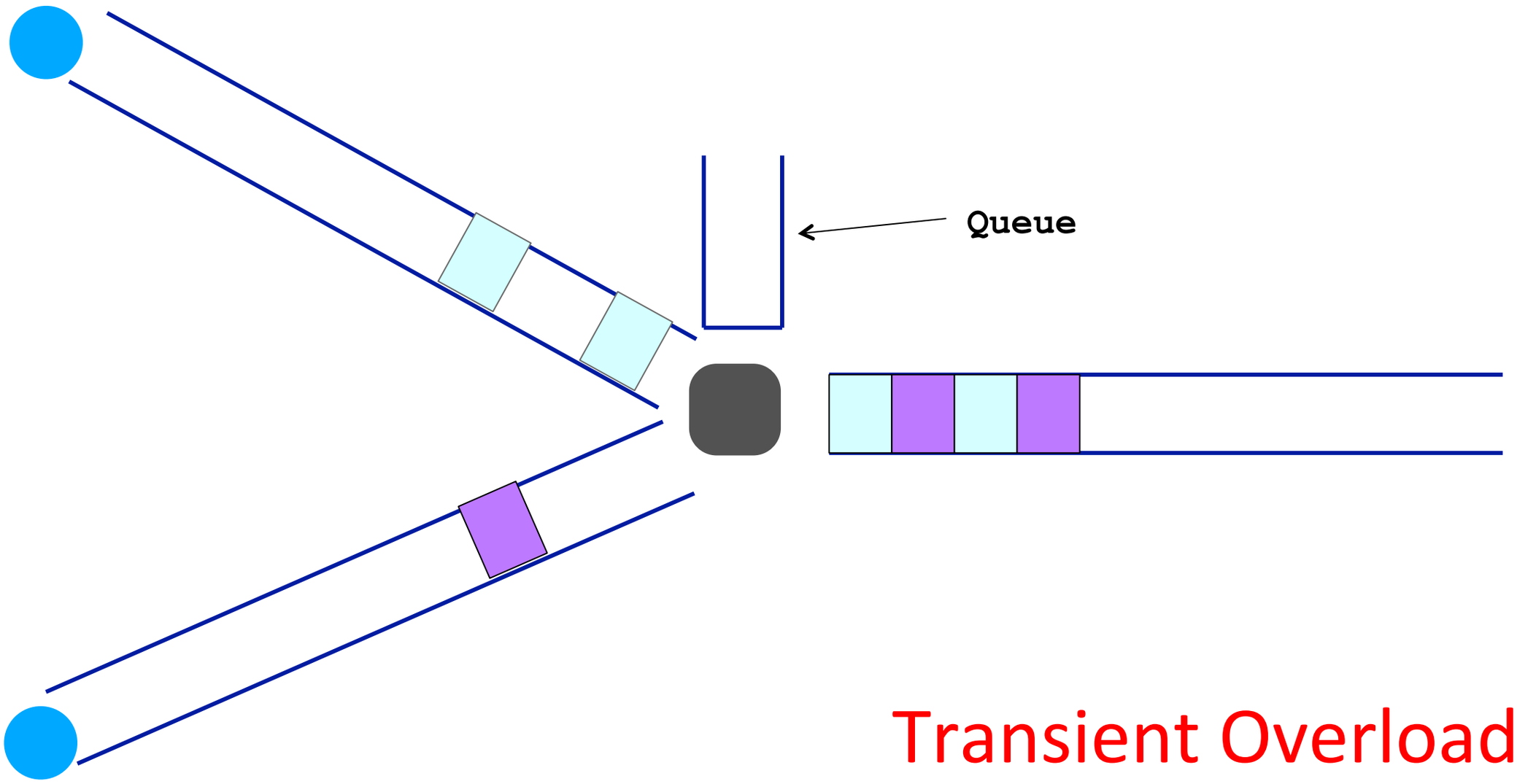
Queuing delay: “pipe” view



Transient Overload

Not a rare event!

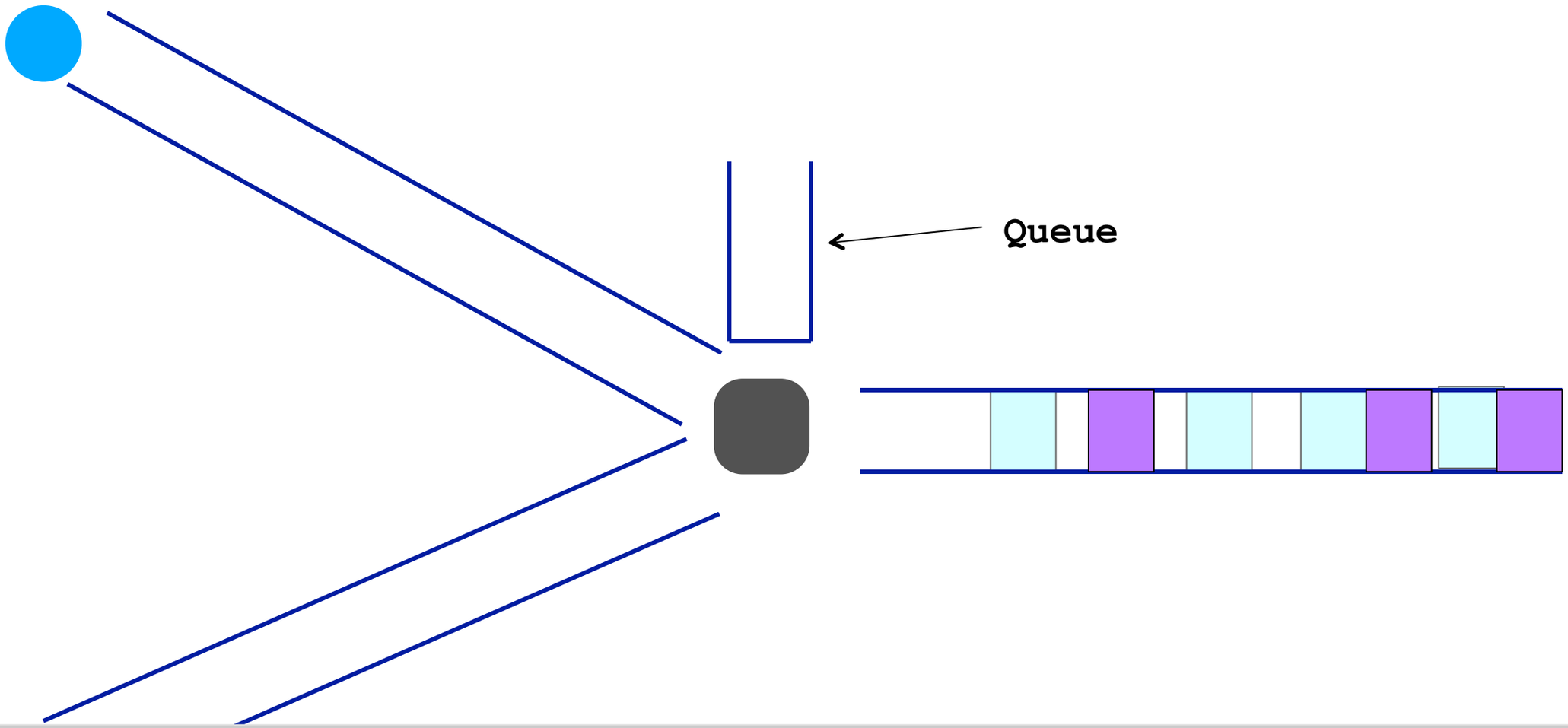
Queuing delay: “pipe” view



Transient Overload

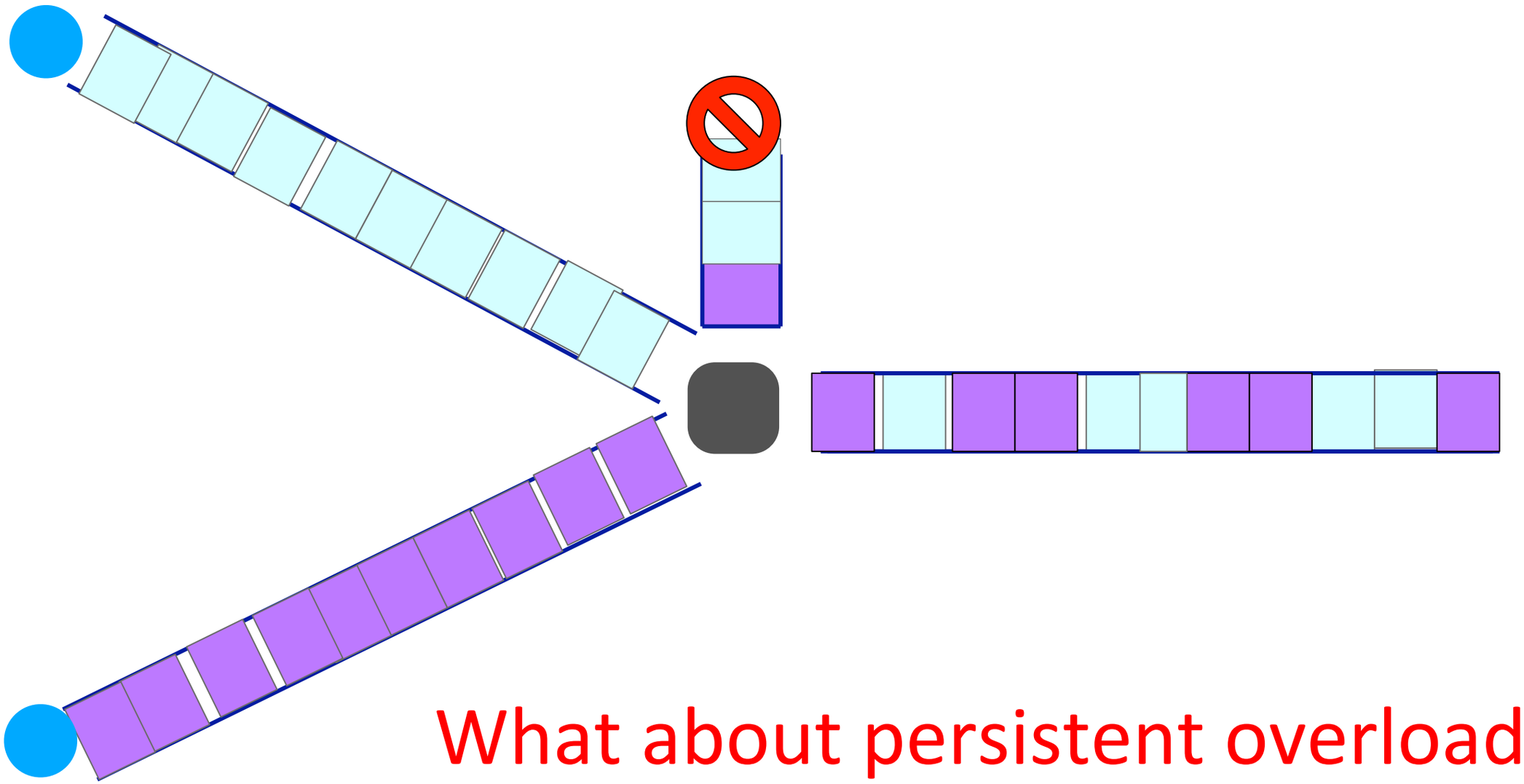
Not a rare event!

Queuing delay: “pipe” view



Queues absorb transient bursts but introduce queuing delay

Queuing delay: “pipe” view



What about persistent overload?

Will eventually drop packets (“loss”)

Queuing Delay

- *How long does a packet have to sit in a buffer before it is processed?*
- Depends on traffic pattern
  - arrival rate at the queue
  - nature of arriving traffic (bursty or not?)
  - transmission rate of outgoing link

Queuing Delay

- *How long does a packet have to sit in a buffer before it is processed?*
- Characterized with statistical measures
  - average queuing delay
  - variance of queuing delay
  - probability delay exceeds a threshold value

Basic Queuing Theory Terminology

- Arrival process: how packets arrive
  - Average rate A
  - Peak rate P

- 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”

Little's Law (1961)

$$L = A \times W$$

- Compute L: count packets in queue every second
  - How often does a single packet get counted? W times

- 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?*
  - typically assume this is negligible

Recap

- What is a network made of?
  - *What physical infrastructure exists?*
- How is it shared?
  - *On-demand or reserve?*
- How do we evaluate a network?
  - *to be contd.*