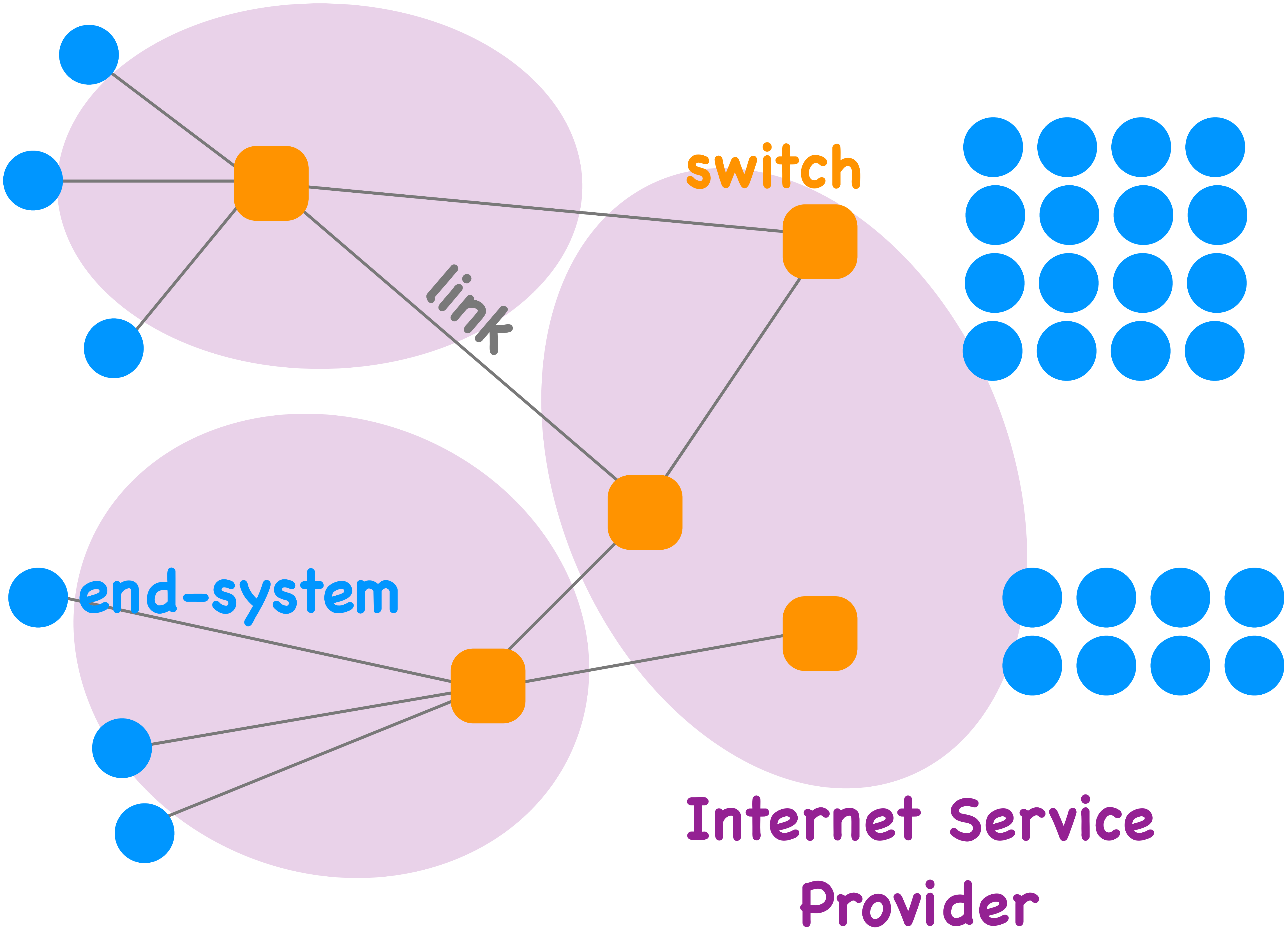
Lecture 2:

Introduction (part 2)

Katerina Argyraki, EPFL

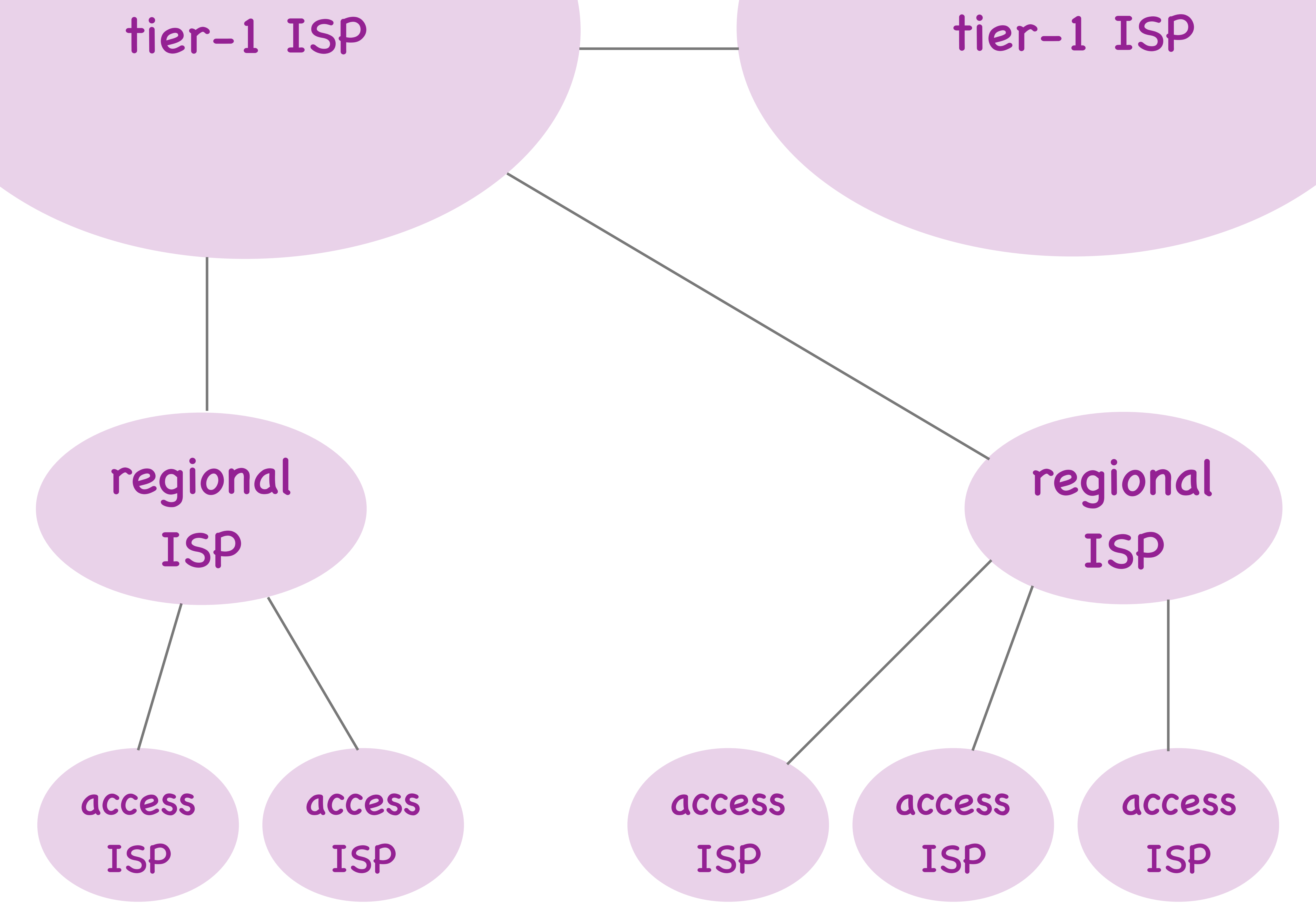
Questions

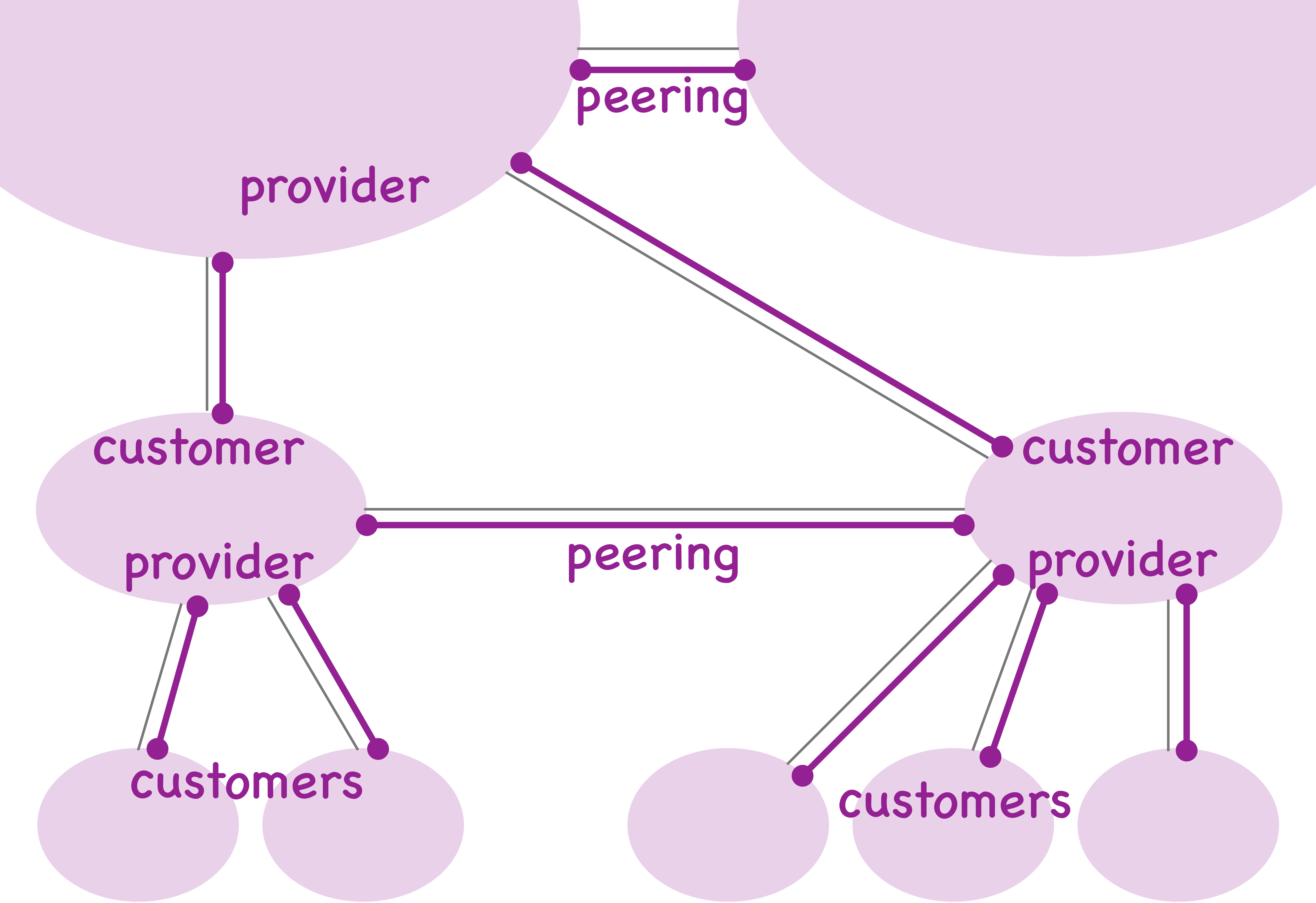
- What's underneath?

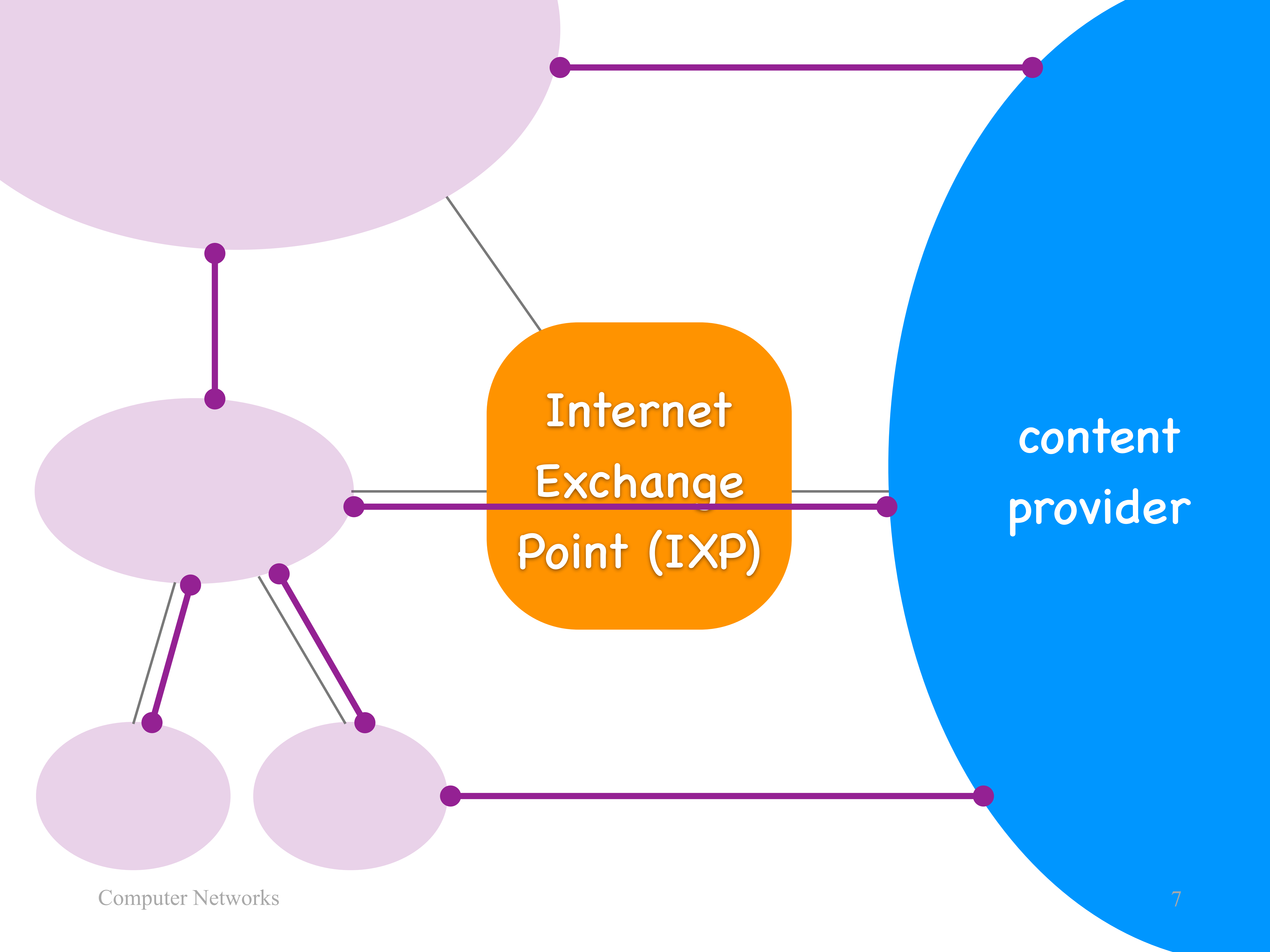

switch
link
end-system
Internet Service Provider

Questions

- What's underneath?
- Who owns what?


tier-1 ISP
tier-1 ISP
regional
ISP
regional
ISP
access
ISP
access
ISP
access
ISP
access
ISP
access
ISP


peering
provider
customer
customer
peering
provider
provider
customers
customers


Internet Exchange Point (IXP)
content provider

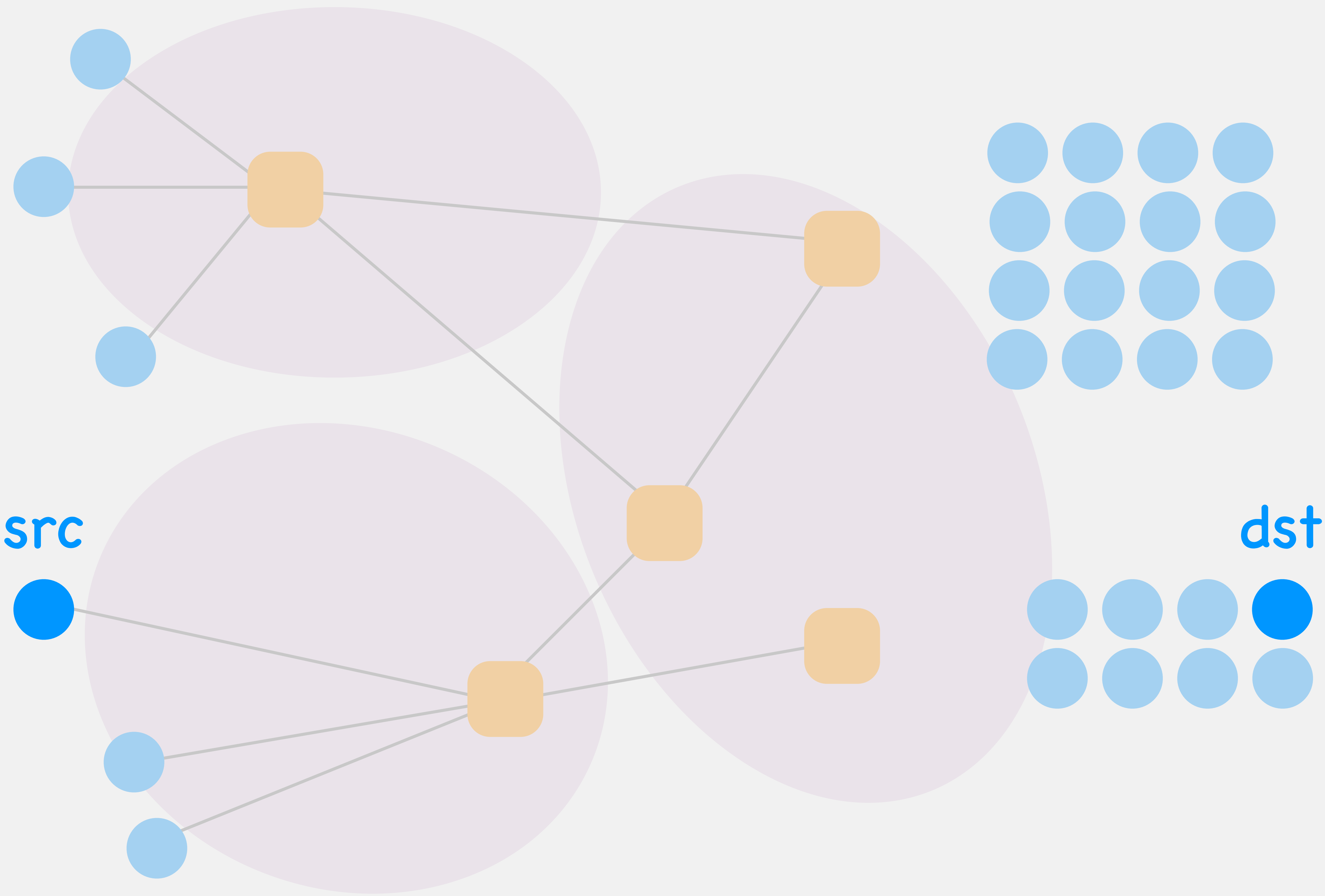
Questions

- What's underneath?
- Who owns what?
- How does it work?

application	web BitTorrent email DNS
transport	TCP UDP
network	IP
link	DSL Cable Ethernet WiFi Cellular Optical
physical	copper fiber wireless

Questions

- What's underneath?
- Who owns what?
- How does it work?
- **How does one *evaluate* it?**
- How do end-systems share it?


src
dst

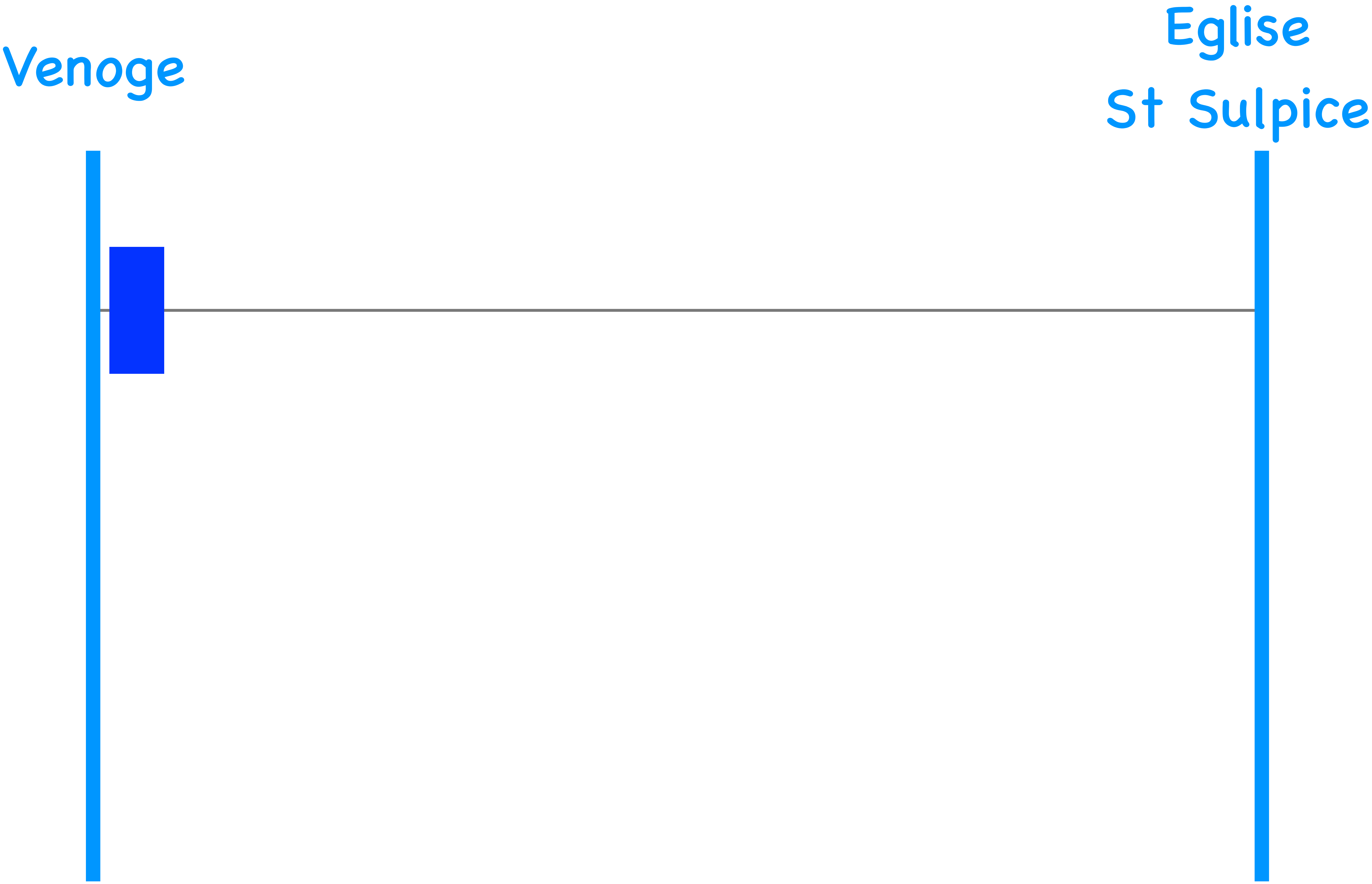
Basic performance metrics

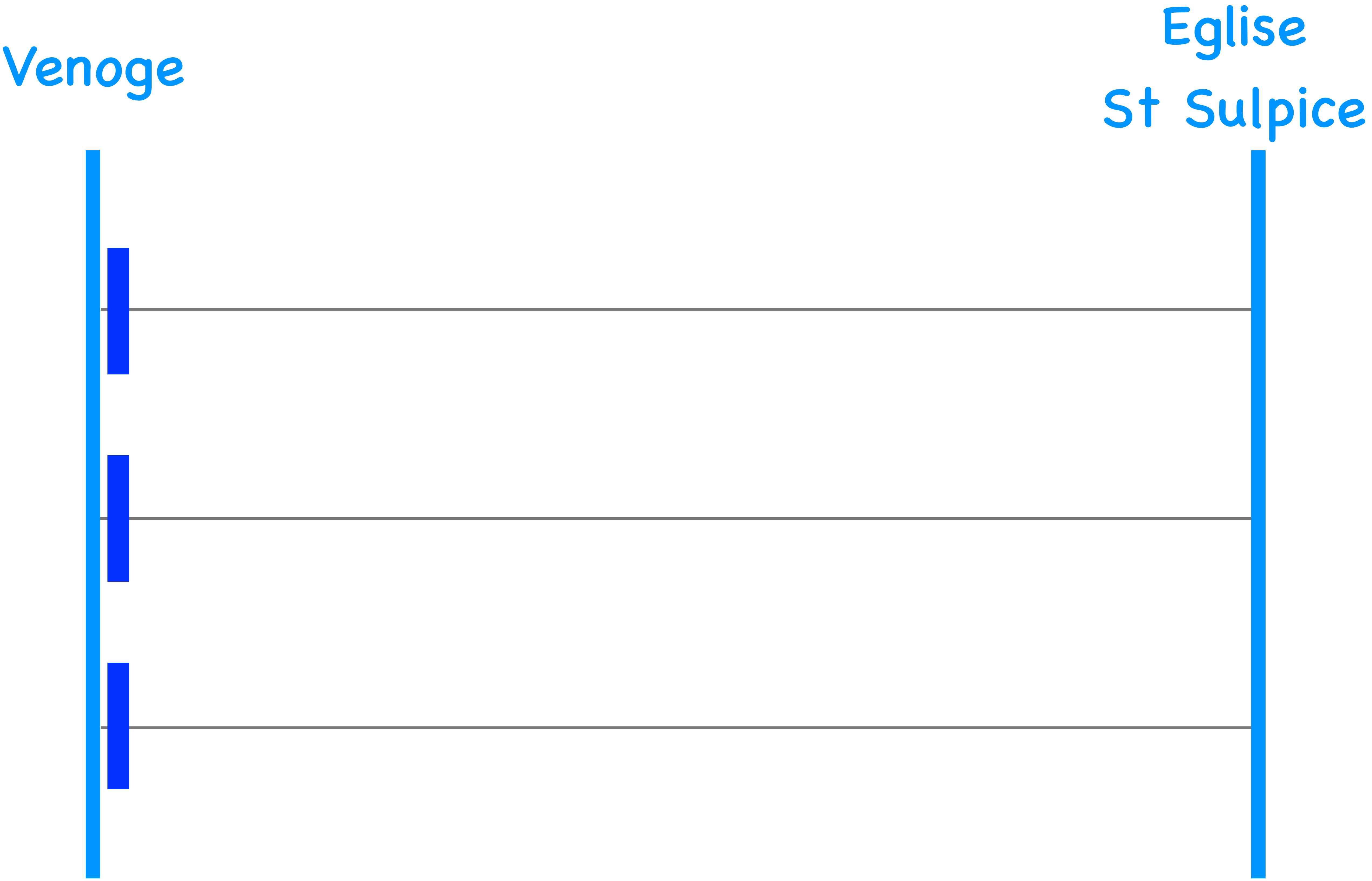
- Packet loss
  - the fraction of packets from src to dst that are lost on the way
  - in %, e.g., 1% packet loss

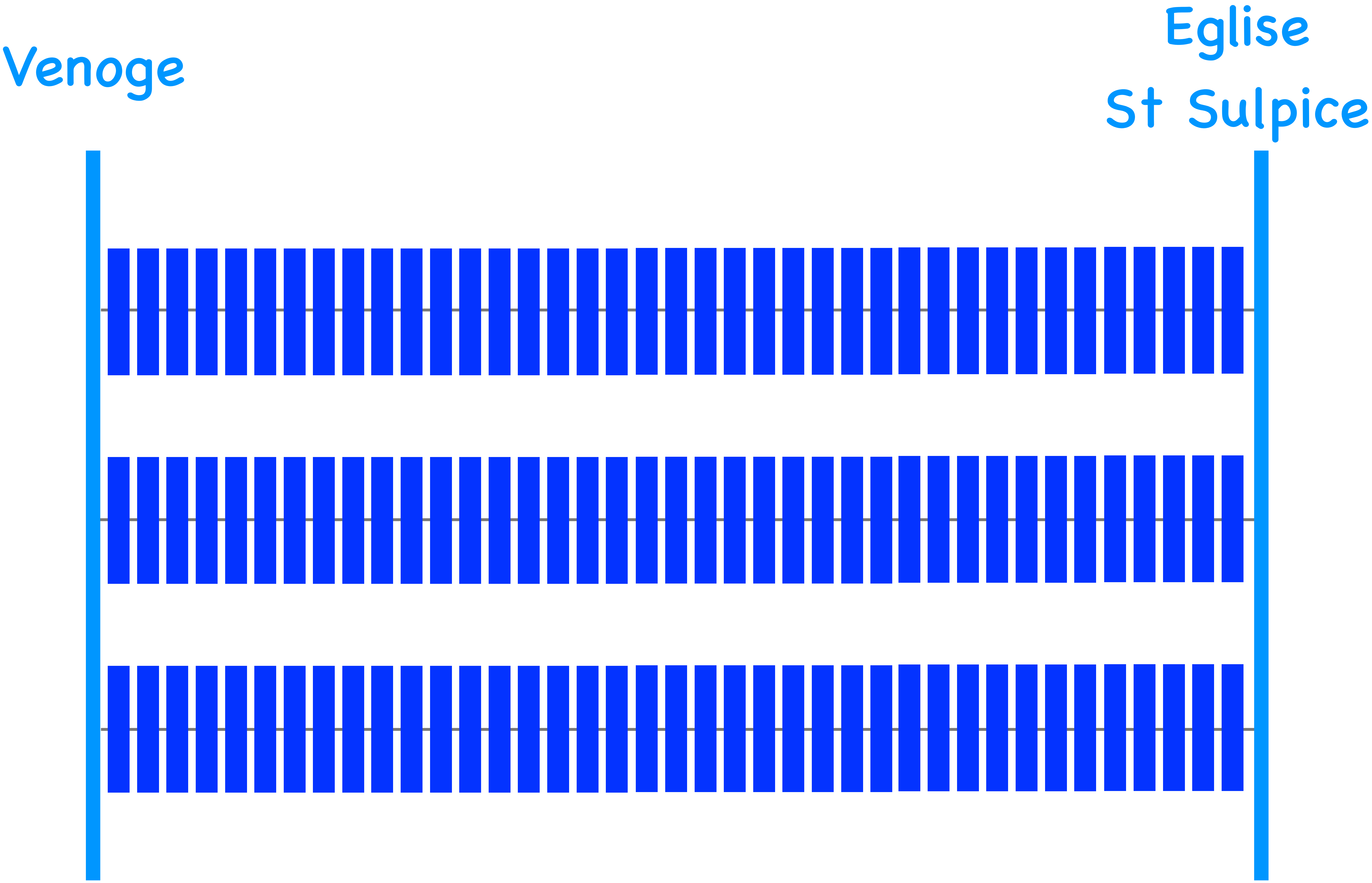
- Packet delay
  - the time it takes for a packet to get from src to dst
  - in time units, e.g., 10 msec

Basic performance metrics

- Average throughput
  - the average rate at which dst receives data
  - in bits per second (bps)
  - e.g., dst receives 1 GB of data in 1 min;
    average throughput = $8\ 10^9$ bits / 60 sec = $133.34\ 10^6$ bps = 133.34 Mbps


Venoge
Eglise
St Sulpice


Venoge
Eglise
St Sulpice


Venoge
Eglise
St Sulpice

Delay vs. throughput

- Packet delay matters for small messages
- Average throughput matters for bulk transfers
- They are related to each other, but not in an obvious way

transmission delay

$$= \frac{\text{packet size}}{\text{link transmission rate}}$$

$$= \frac{3 \text{ bits}}{1 \text{ Gbps}} = 3 \text{ nsec}$$

propagation delay

$$= \frac{\text{link length}}{\text{link propagation speed}}$$

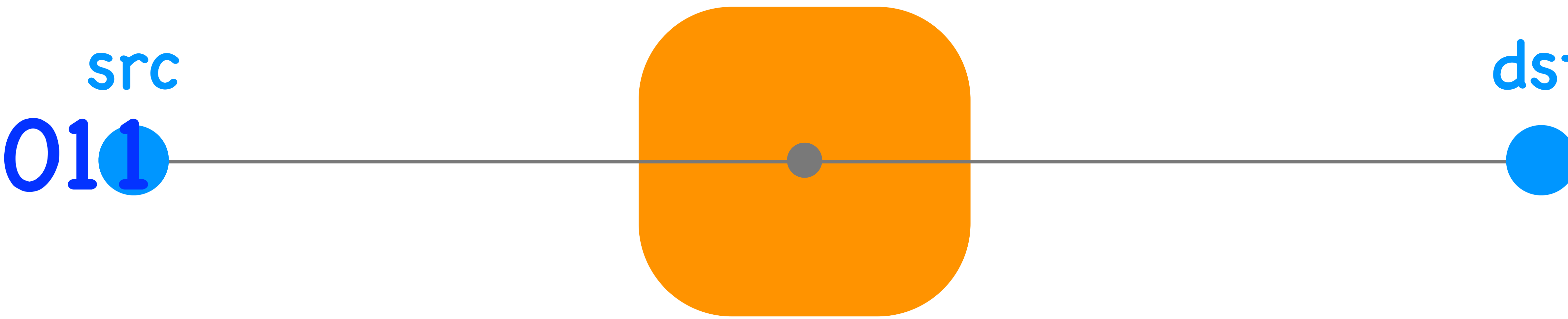
$$= \frac{1 \text{ meter}}{3 \;\; 10^{8} \text{ meters per sec}} = 3.34 \text{ nsec}$$

packet delay =

transmission delay

+ propagation delay

circuit switch



packet delay =

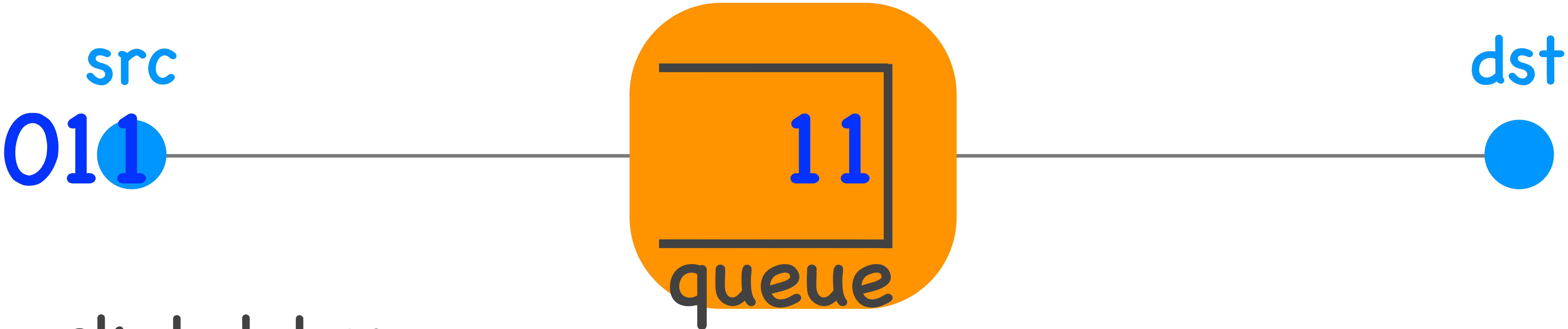
transmission delay over 1st link

+ propagation delay of 1st + 2nd link

(+ delay to establish circuit,

amortized over multiple packets)

store & forward switch

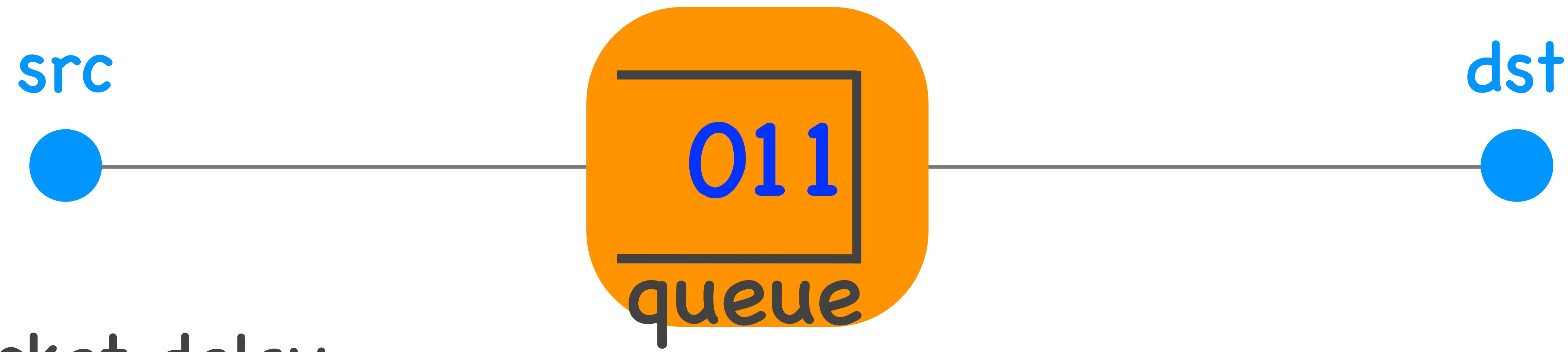


packet delay =

transmission delay over 1st link

+ propagation delay of 1st link

store & forward switch



packet delay =

transmission delay over 1st link

+ propagation delay of 1st link

+ queuing delay

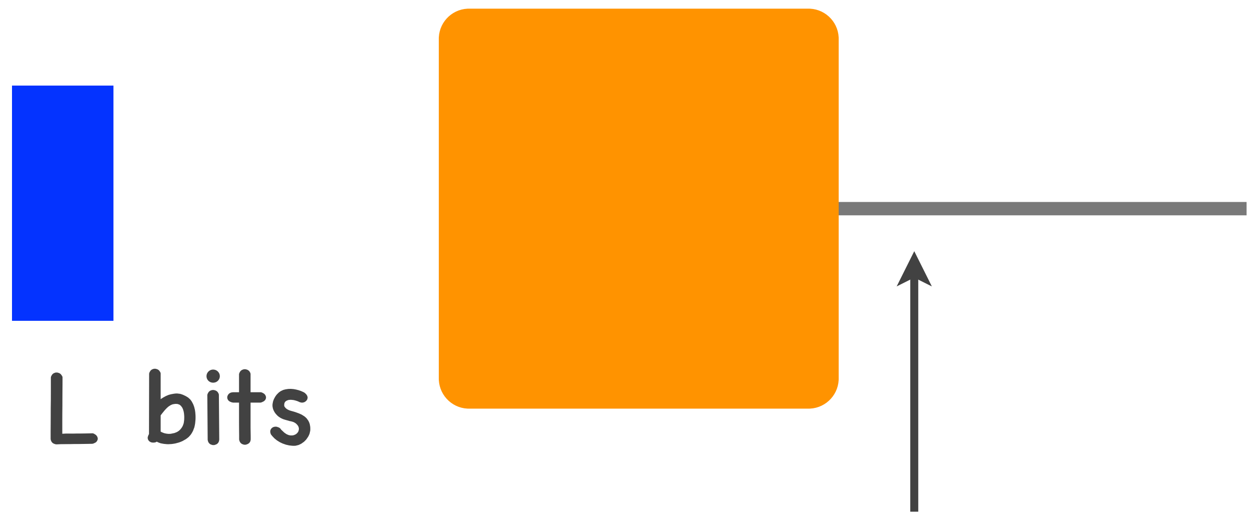
+ processing delay

+ transmission delay over 2nd link

+ propagation delay of 2nd link

Queuing delay

- Given info on traffic pattern
  - arrival rate at the queue
  - nature of arriving traffic (bursty or not?)
- Characterized with statistical measures
  - average queuing delay
  - variance of queuing delay
  - probability that it exceeds a certain value



packet size: L bits

transmission rate:

R bits/sec

packet arrival rate:

A packets/sec

bit arrival rate:

LA bits/sec

bit departure rate:

R bits/sec

bit arrival rate: LA bits/sec $>$ bit departure rate: R bits/sec

Queuing delay

- (Assuming infinite queue)
- Approaches infinity,
  if arrival rate > departure rate

bit arrival rate:

LA bits/sec

<=

bit departure rate:

R bits/sec

0 usec
1 usec
2 usec
3 usec

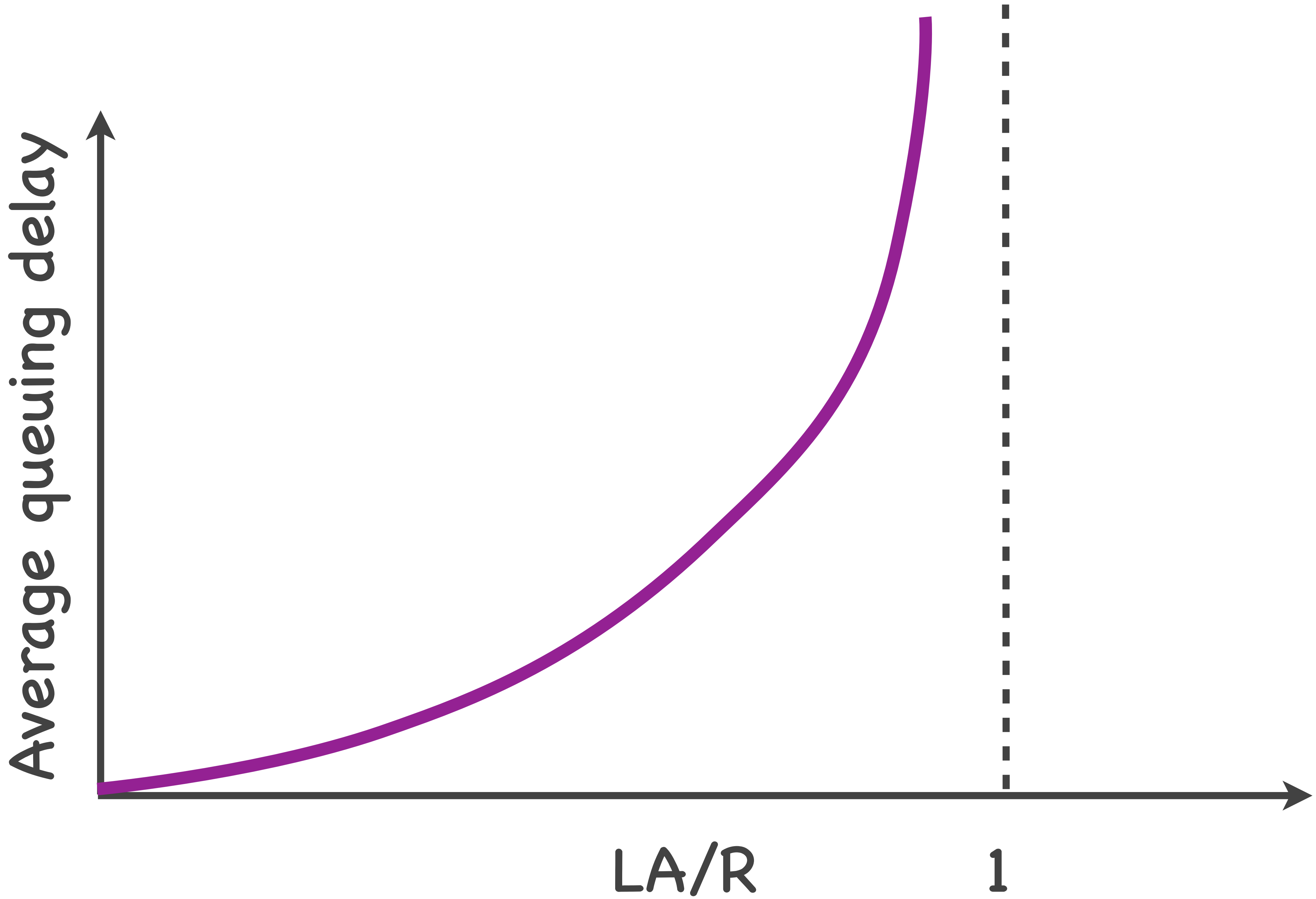
bit arrival rate:
LA bits/sec

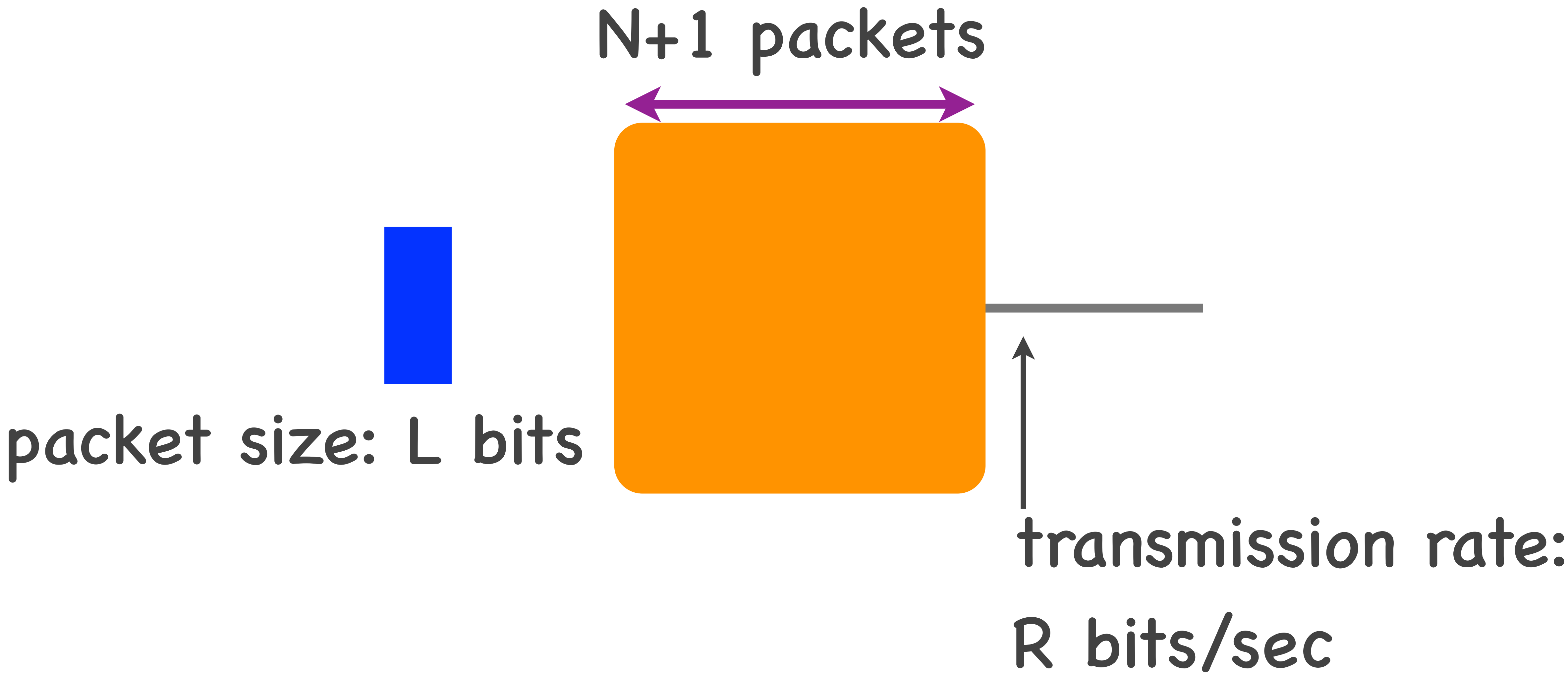
<=

bit departure rate:
R bits/sec

Queuing delay

- (Assuming infinite queue)
- Approaches infinity,
  if arrival rate > departure rate
- Depends on burst size, otherwise


Average queuing delay
LA/R
1



Queuing delay upper bound: **N** L/R

Packet delay

- Many components: transmission, propagation, queuing, processing

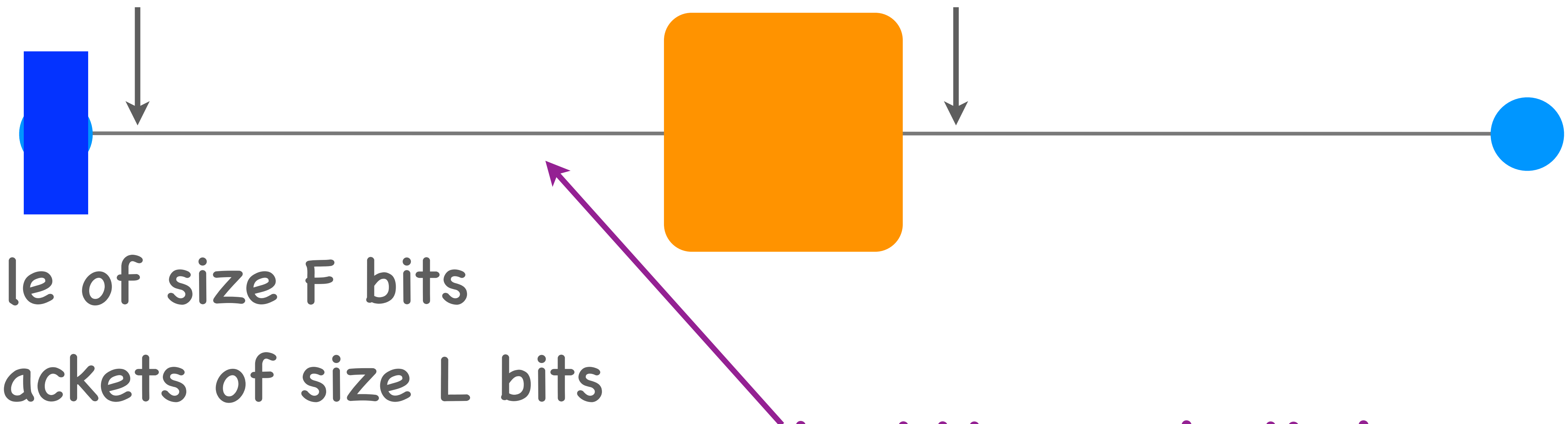
- Depends on network topology, link properties, switch operation, queue capacity, other traffic transmission rate R bits/sec

file of size F bits

packets of size L bits

Transfer time = F/R + propagation delay

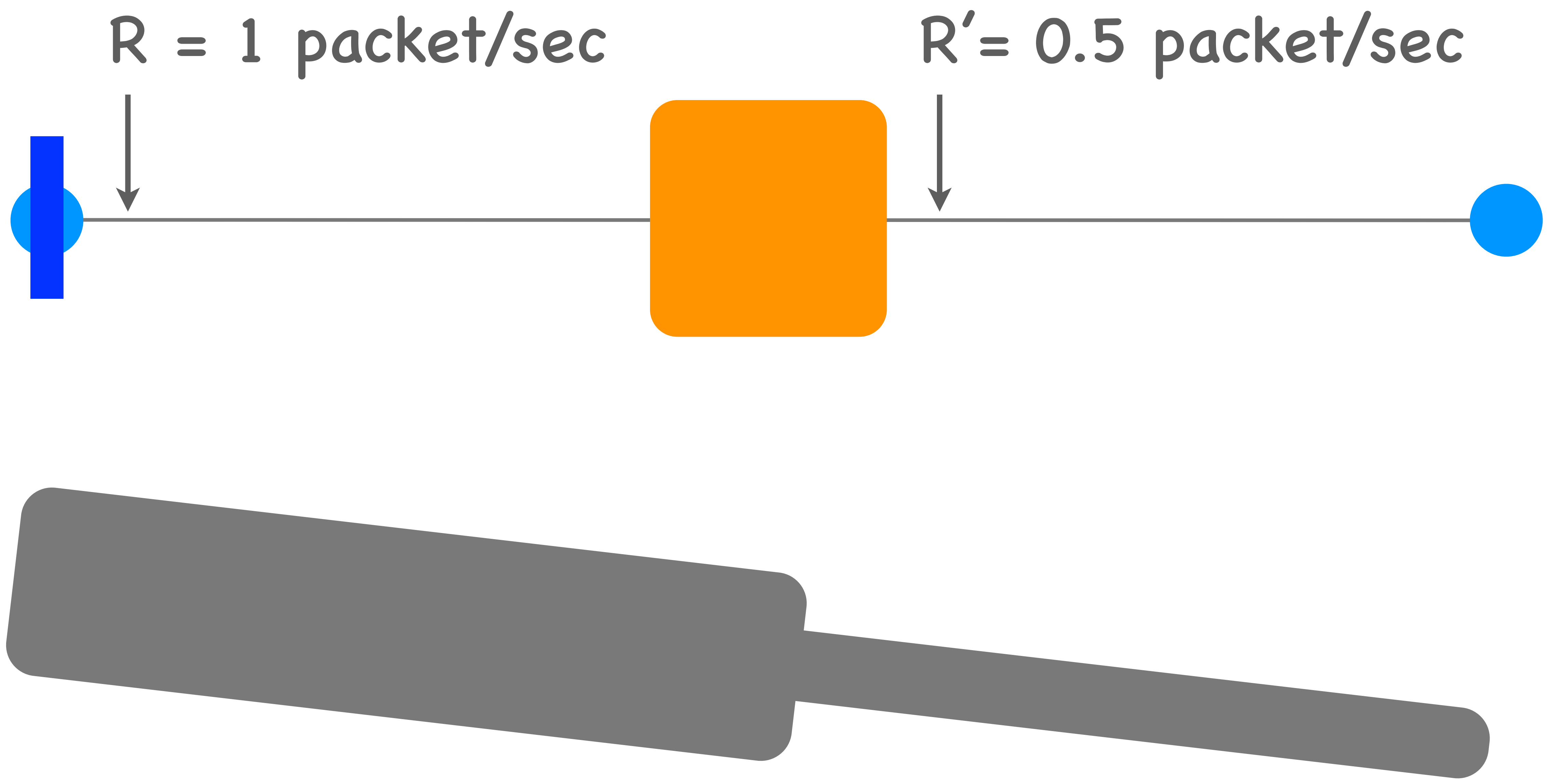
Average throughput = R

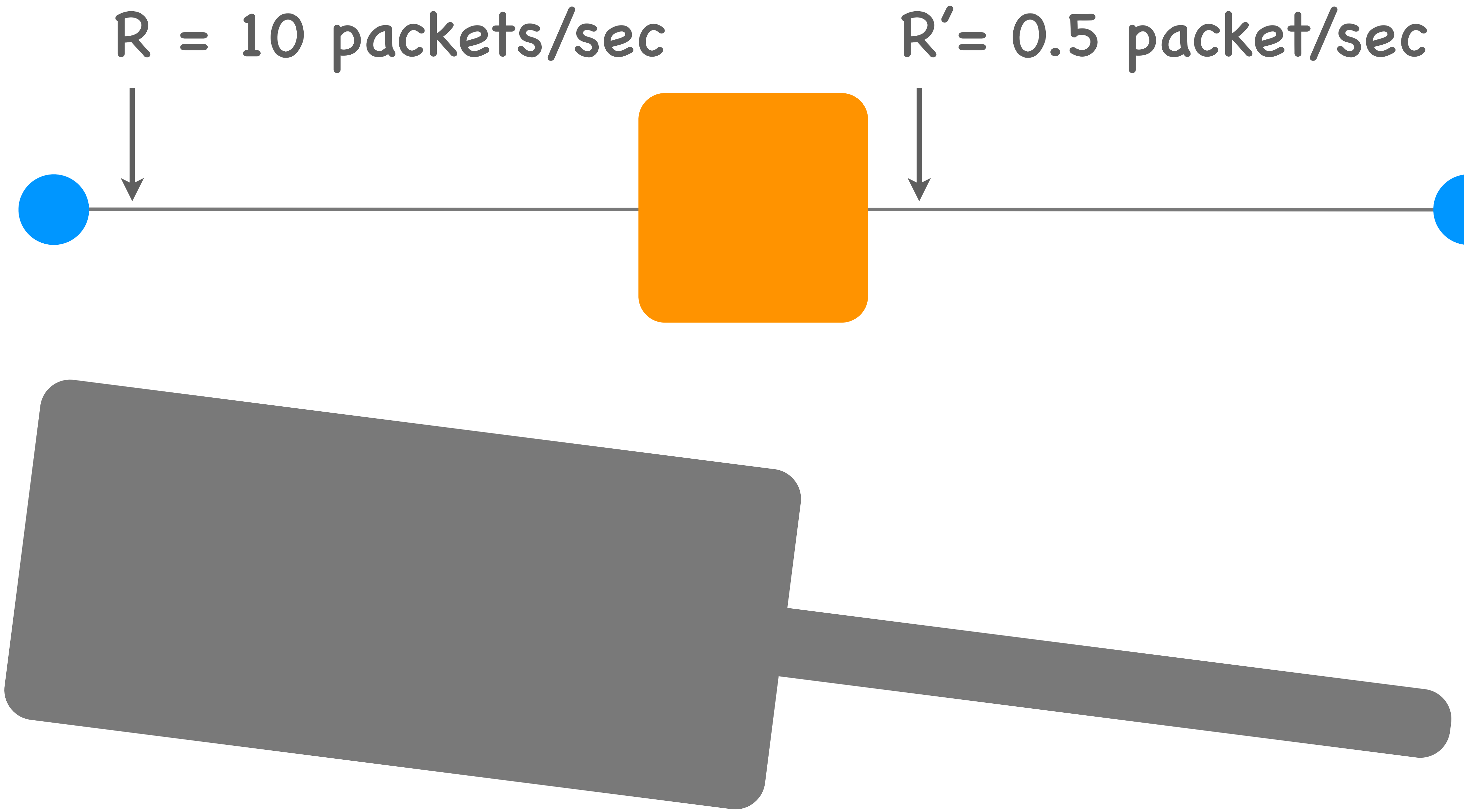


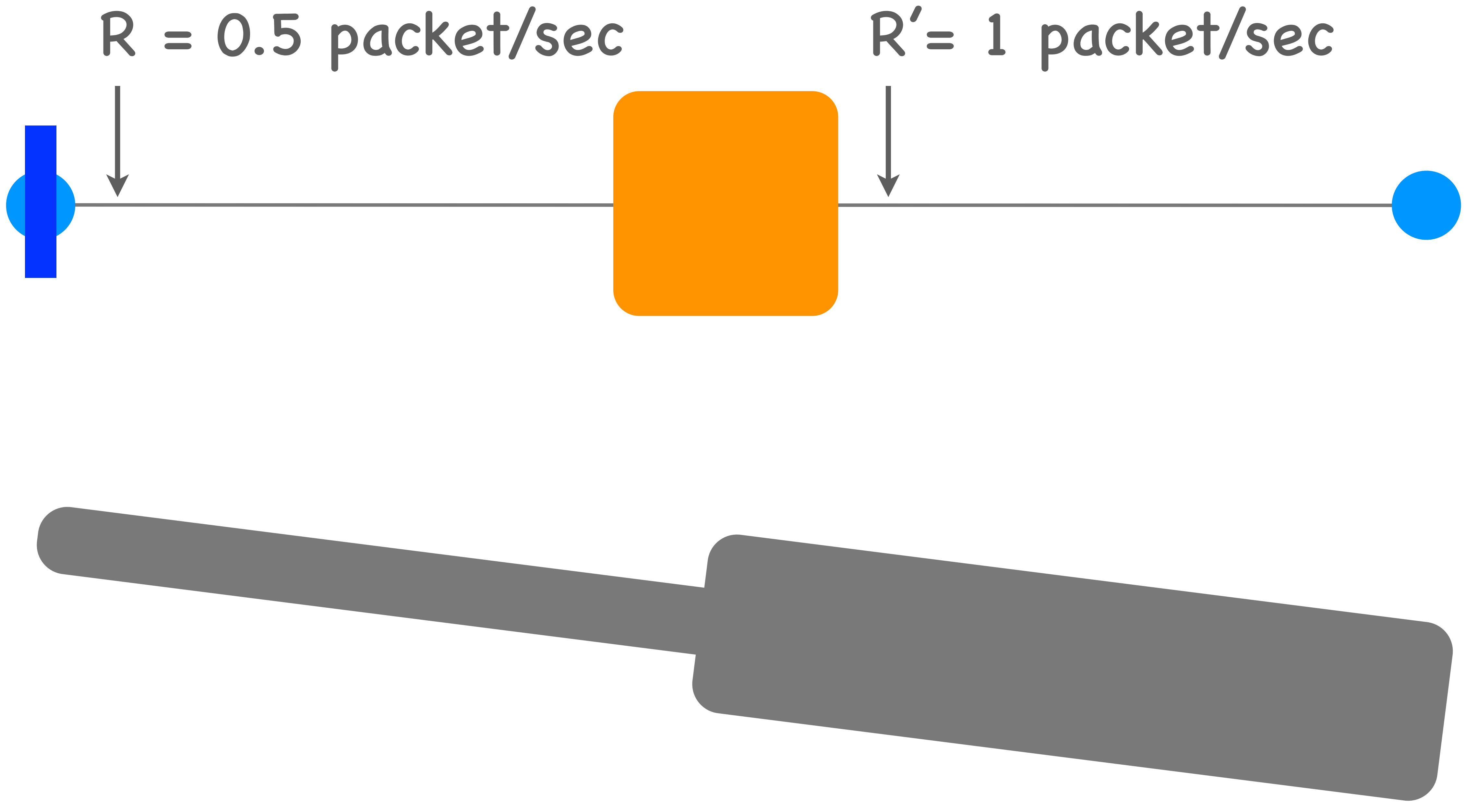
Transfer time = F/R + propagation delay 1st link

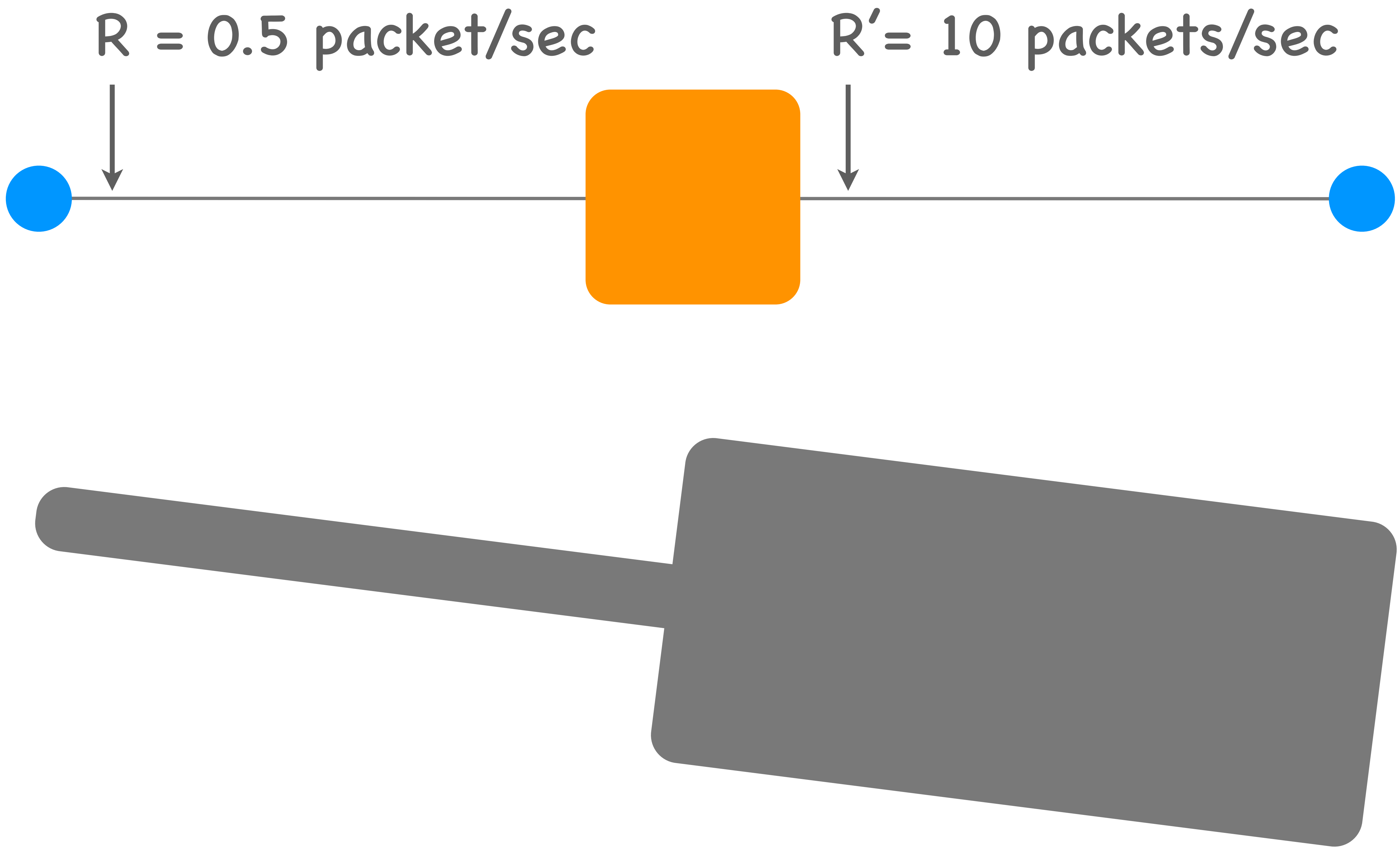
+ L/R′ + propagation delay 2nd link

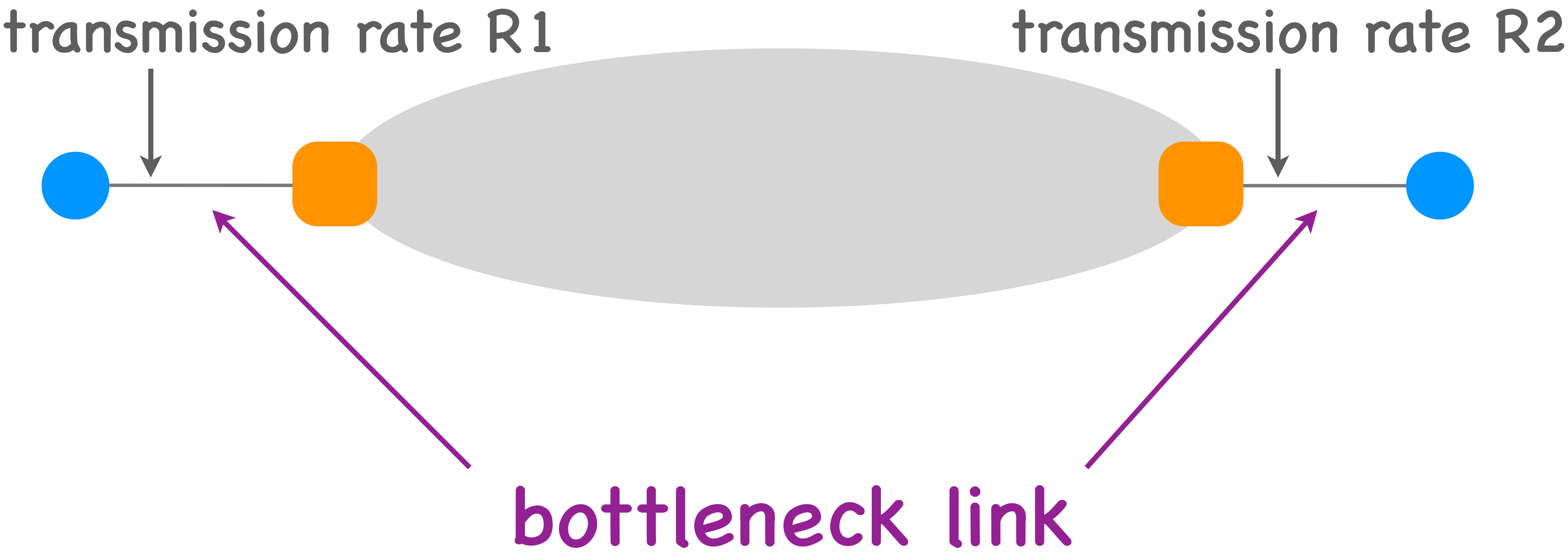
Average throughput = min { R, R′ } = R

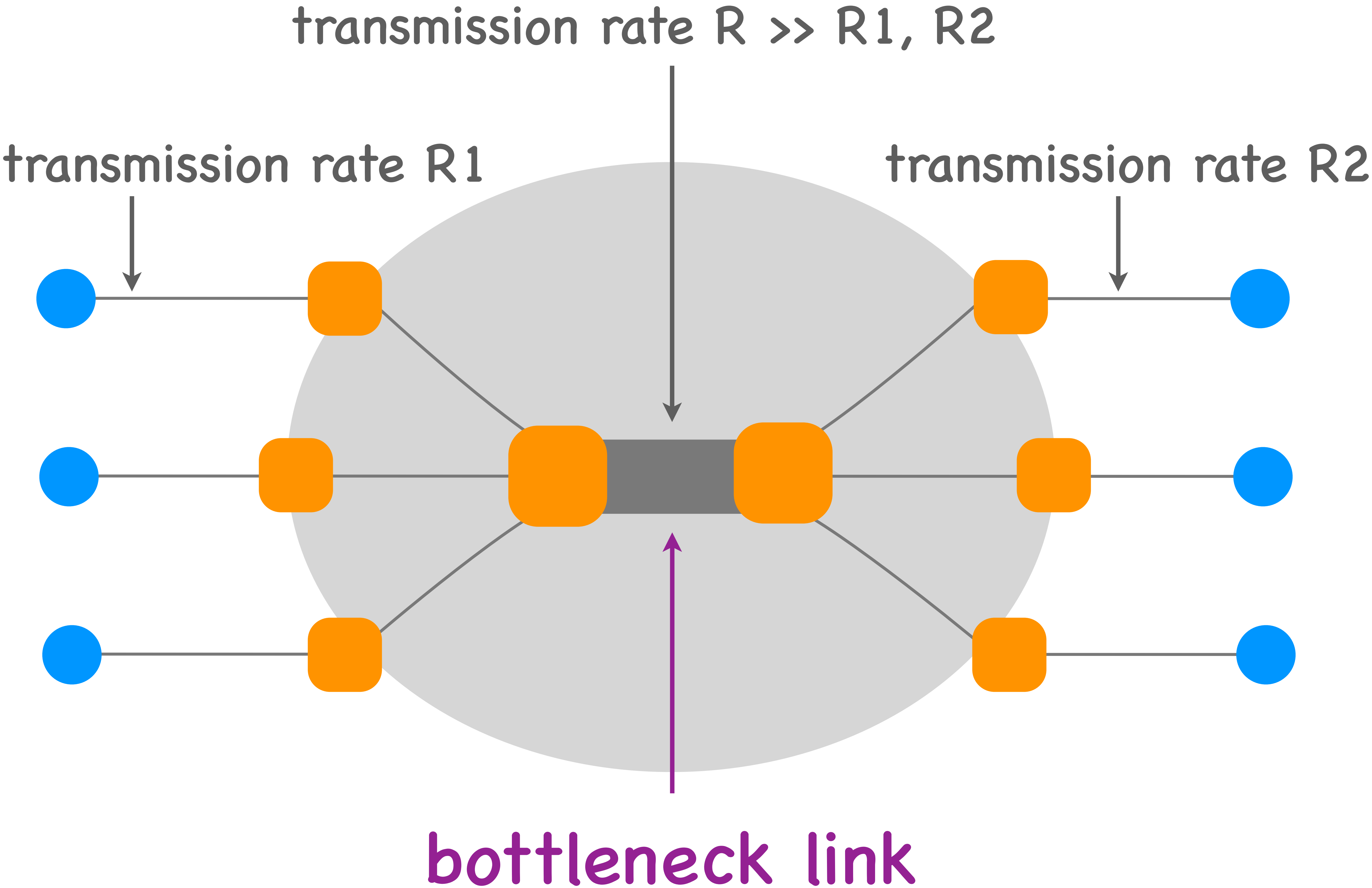

R = 1 packet/sec
R'= 0.5 packet/sec


R = 10 packets/sec
R’= 0.5 packet/sec


R = 0.5 packet/sec
R'= 1 packet/sec


R = 0.5 packet/sec
R'= 10 packets/sec


transmission rate R1
transmission rate R2
bottleneck link


transmission rate R >> R1, R2
transmission rate R1
transmission rate R2
bottleneck link

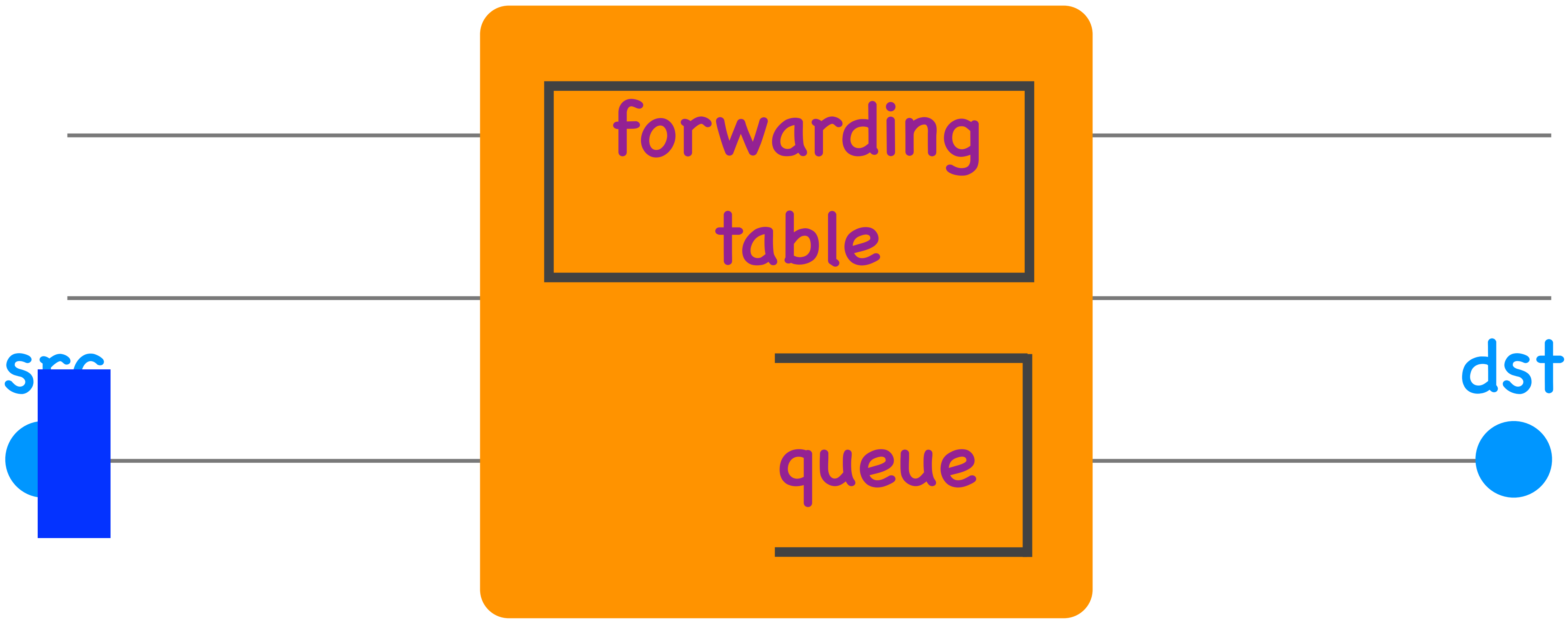
Bottleneck link

- The link where traffic flows at the slowest rate

- Could be because of the link's transmission rate or because of queuing delay

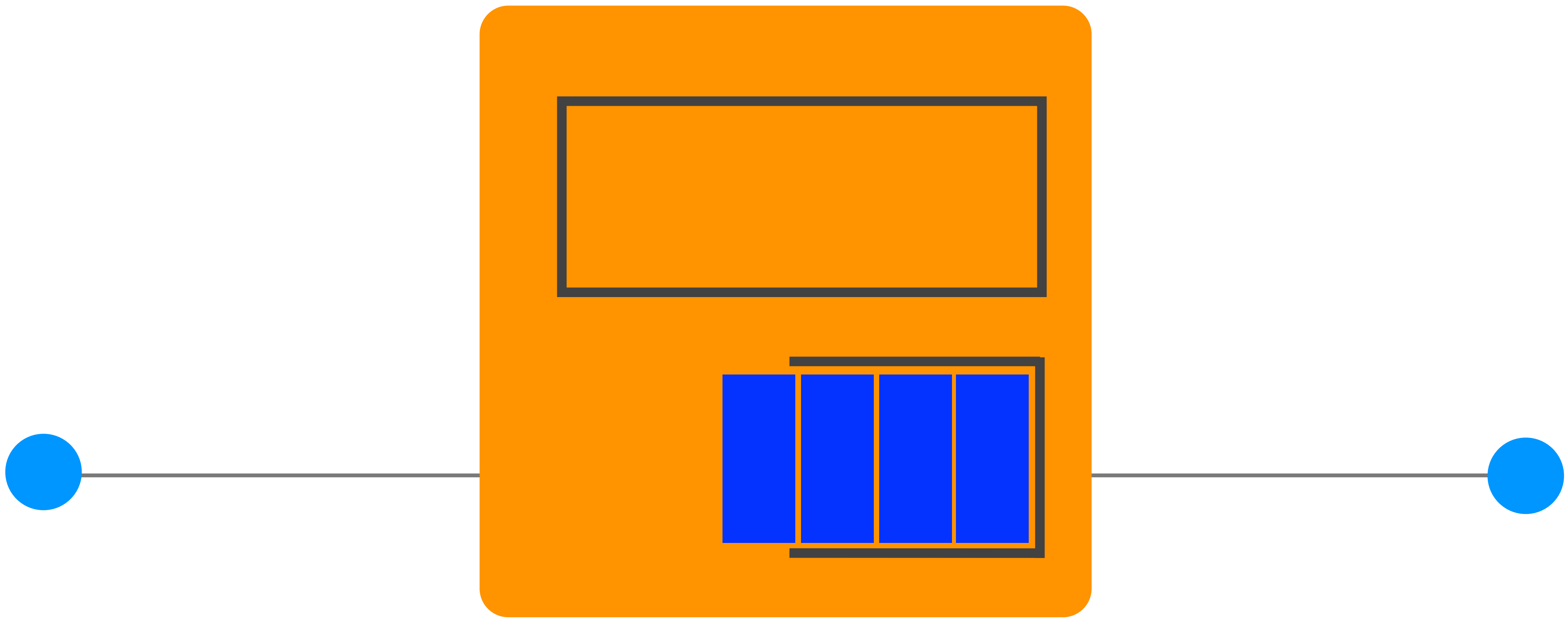
Questions

- What's underneath?
- Who owns what?
- How does it work?
- How does one evaluate it?
- How do end-systems share it?


forwarding
table
queue
src
dst

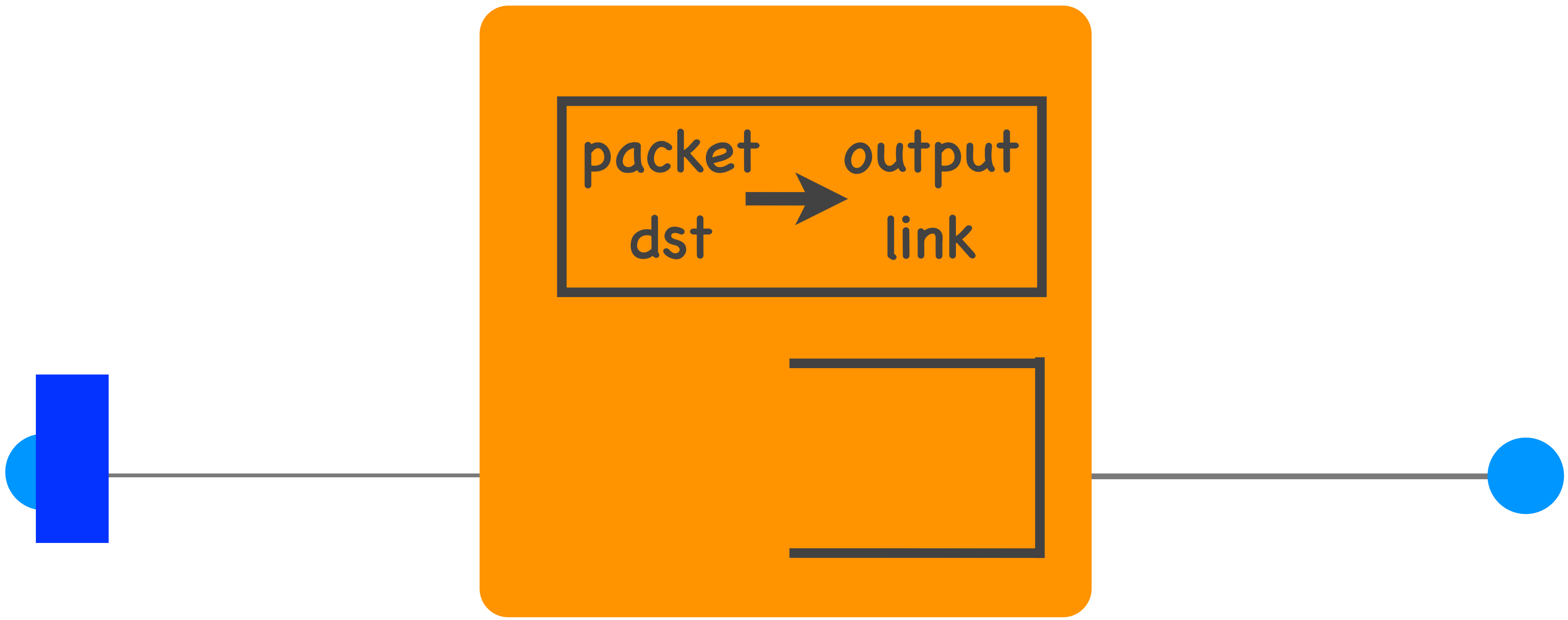
Switch contents

- Queue
  - stores packets
- Forwarding table
  - store meta-data
  - indicate where to send each packet



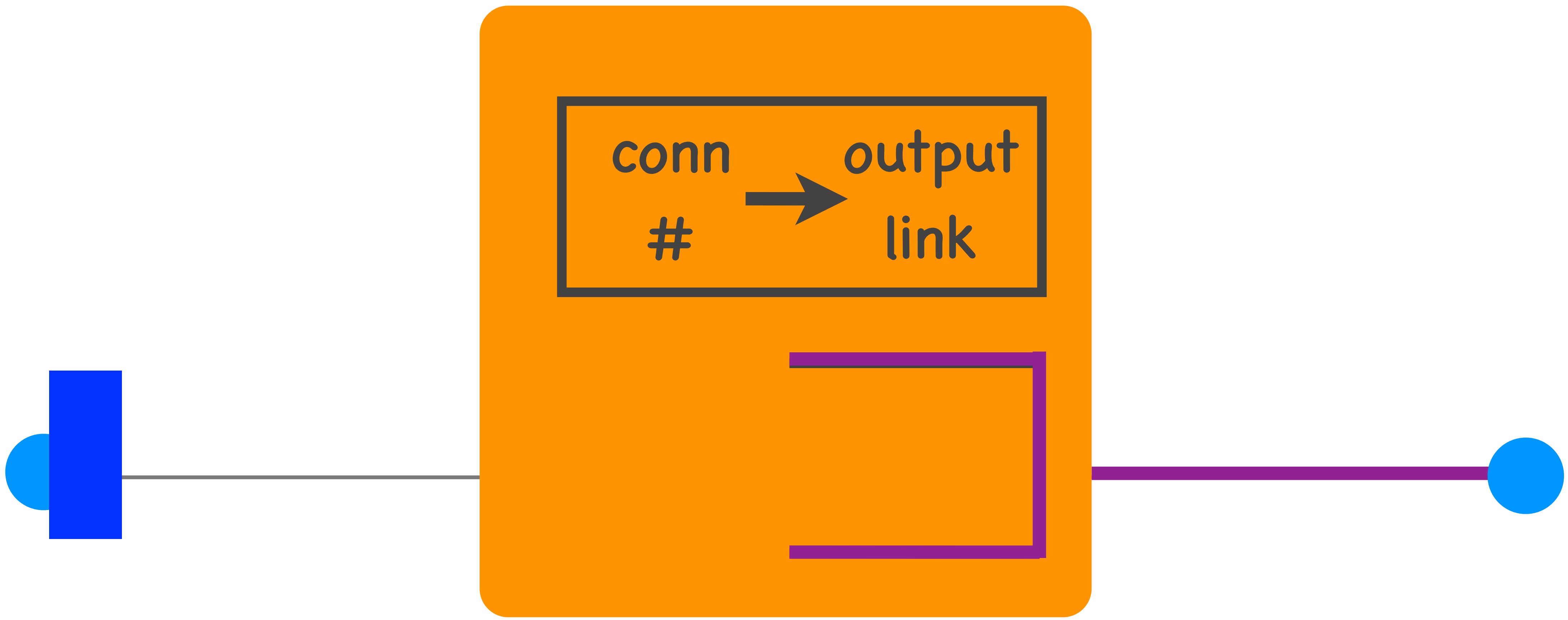
packet loss queuing delay

Packet switching



Packets treated on demand

“Connection switching”



Resources reserved in advance

Resource management

- Packet switching
  - packets treated on demand
  - admission control & forwarding decision: per packet
- "Connection switching"
  - resources reserved per active connection
  - admission control & forwarding decision: per connection

Treat on demand or reserve?

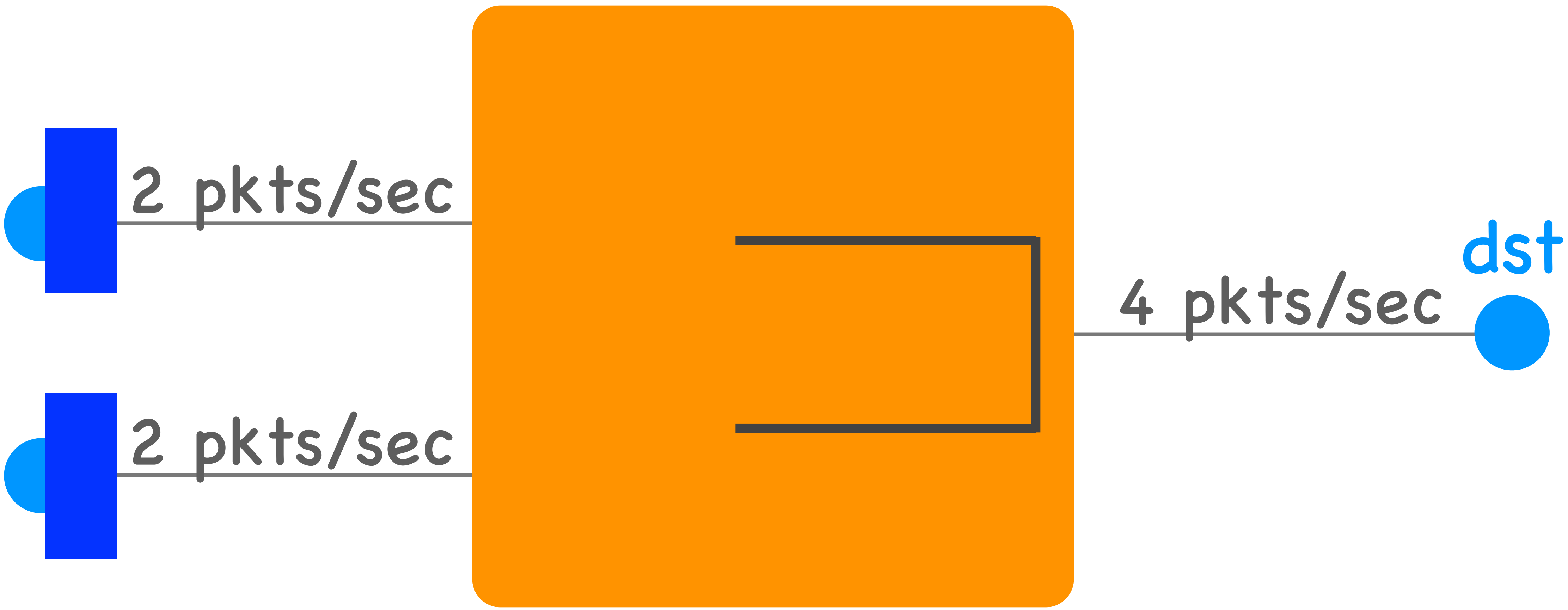
"Connection switching"

Predictable performance

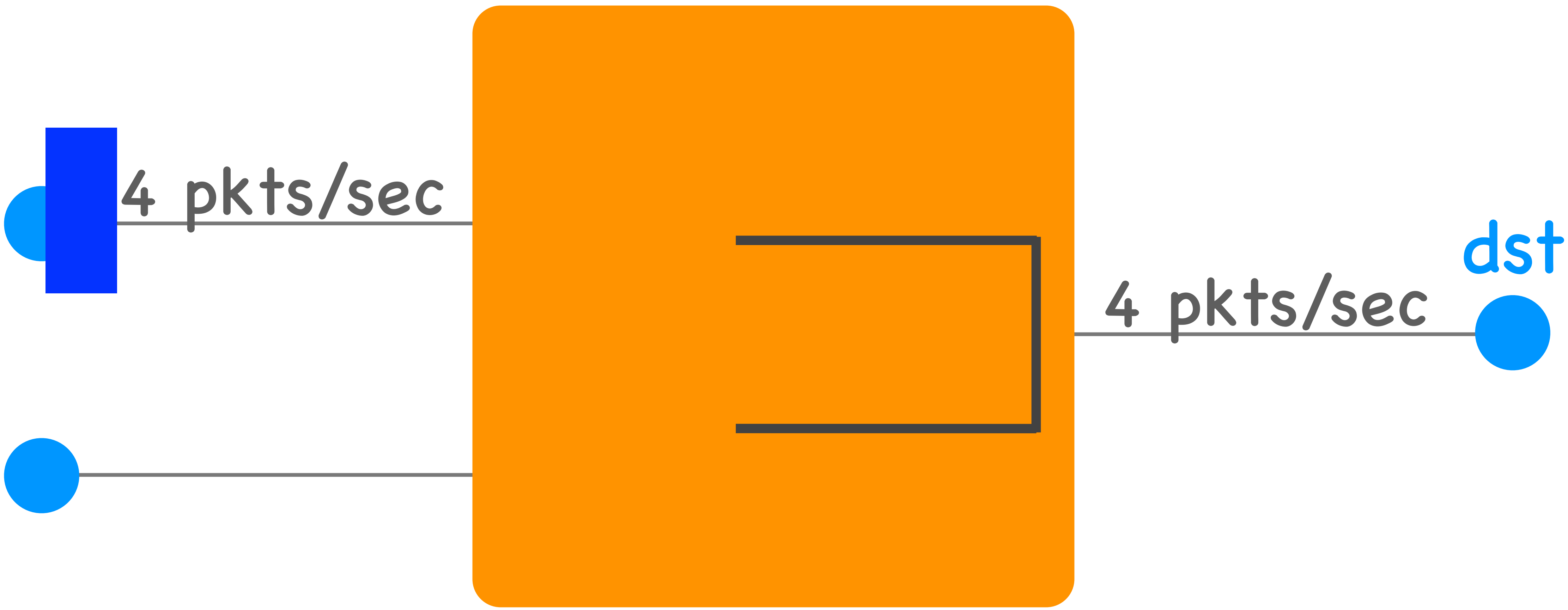
"Connection switching"

Inefficient use of resources

Packet switching

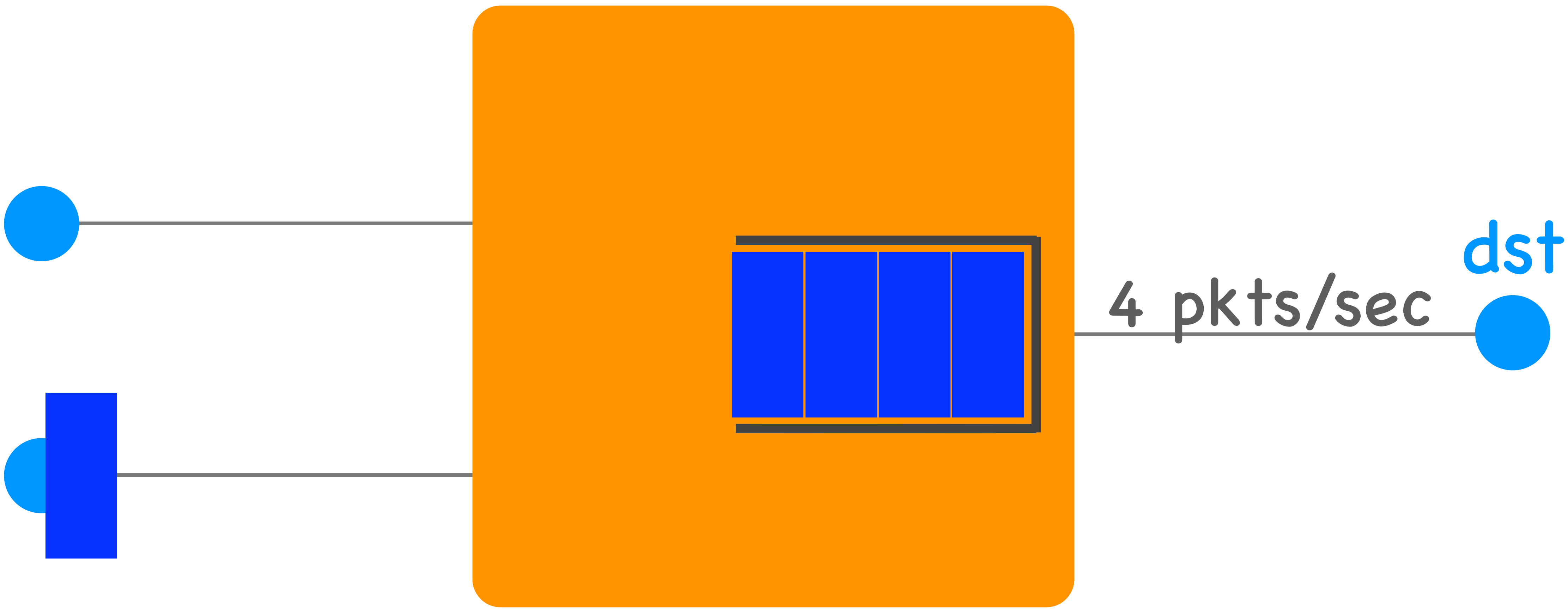


Packet switching



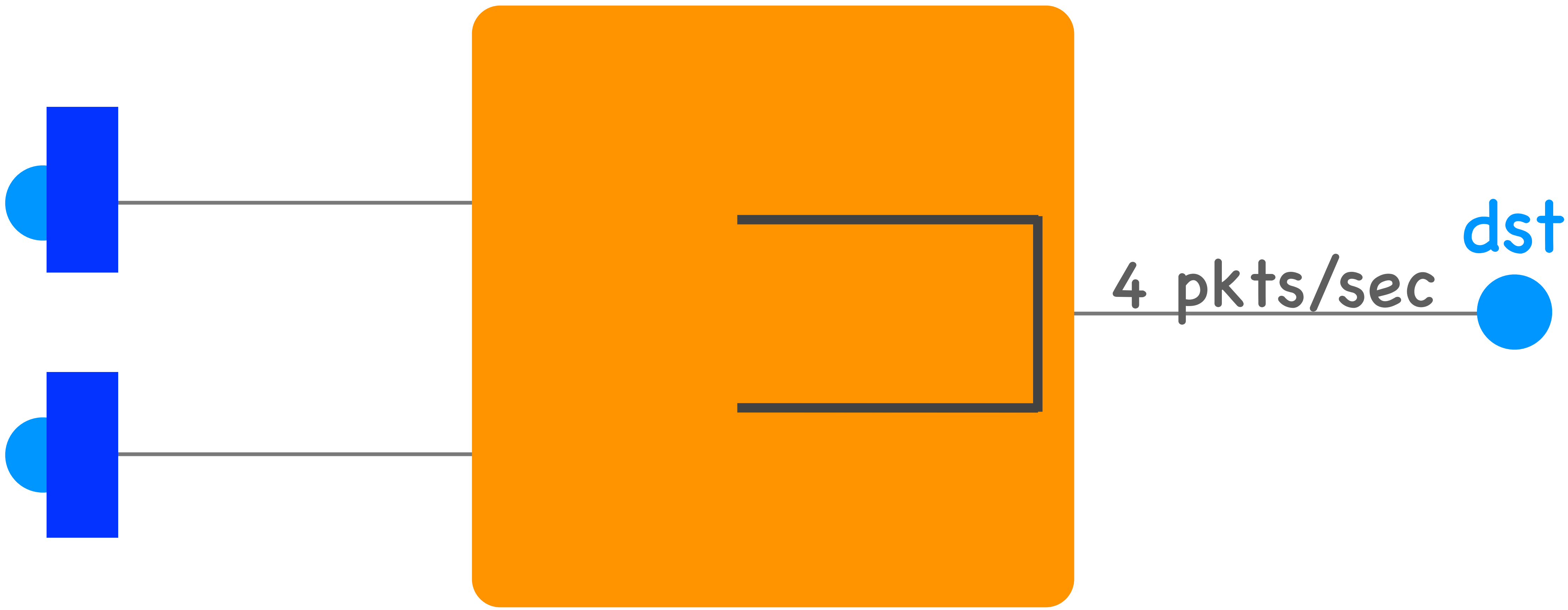
Efficient use of resources

Packet switching



Unpredictable performance

Packet switching



We need congestion control

Resource management

- Packet switching
  - efficient resource use
  - no performance guarantees
  - simpler to implement,
    but requires congestion control
- "Connection switching"
  - performance guarantees
  - inefficient resource use

Each user is active w.p. 10%

With 35 users, 10 or fewer users are active w.p. 99.96%

Connection switching: 10 users
Packet switching: about 35 users

Statistical multiplexing

- Many users share the same resource
- Not all of them can share it at the same time…
- but we do not expect them to be all active at the same time

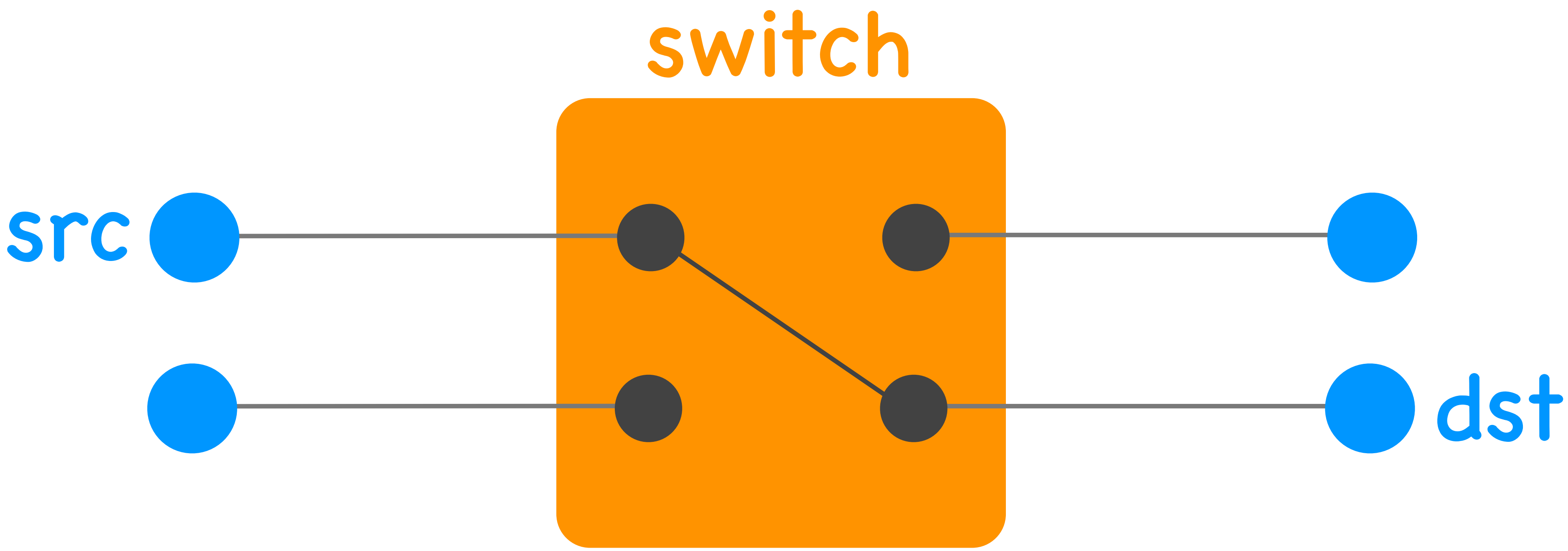
Only 1 user active

Downloading a 10 Gbit video file

Connection switching: 10 seconds

Packet switching: 1 second

Circuit switching



Connection switching
through physical circuits

Many kinds of “circuits”

- Physical circuits
  - separate sequence of physical links per connection
- Virtual circuits
  - manage resources as if there was a separate sequence of physical links per connection

Many kinds of “circuits”

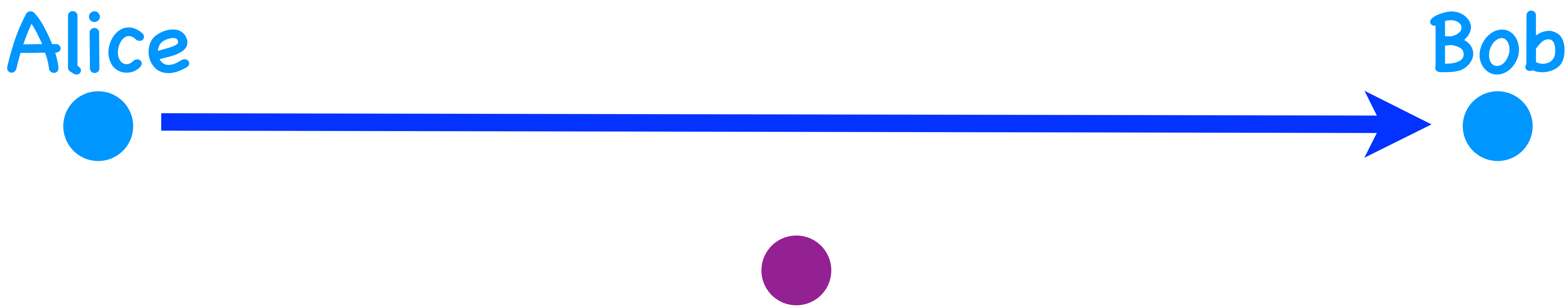
- Time division multiplexing
  - divide time in time slots
  - separate time slot per connection
- Frequency division multiplexing
  - divide frequency spectrum in frequency bands
  - separate frequency band per connection

Many kinds of “circuits”

- Different ways to implement “connection switching”

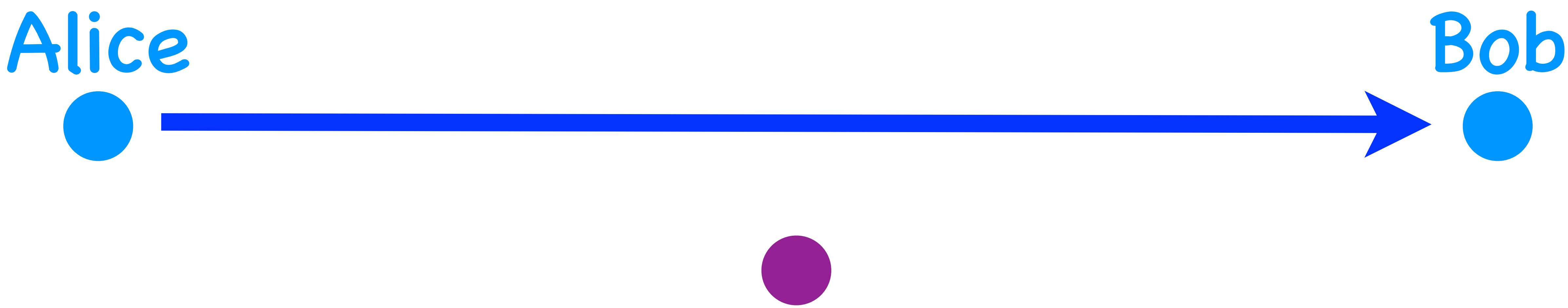
- Create the illusion of a separate physical circuit per connection

Treat on demand or take reservations?



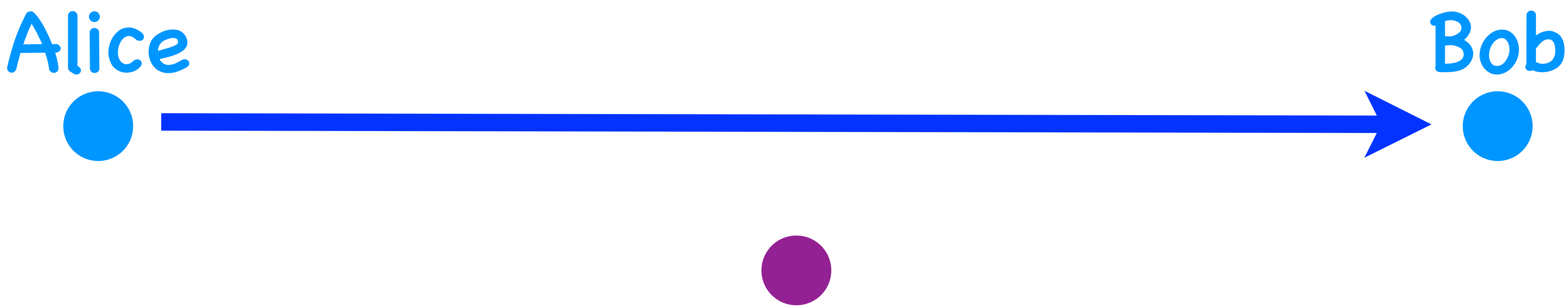
Eve (the eavesdropper)

tries to listen in on the communication, i.e., obtain copies of the data



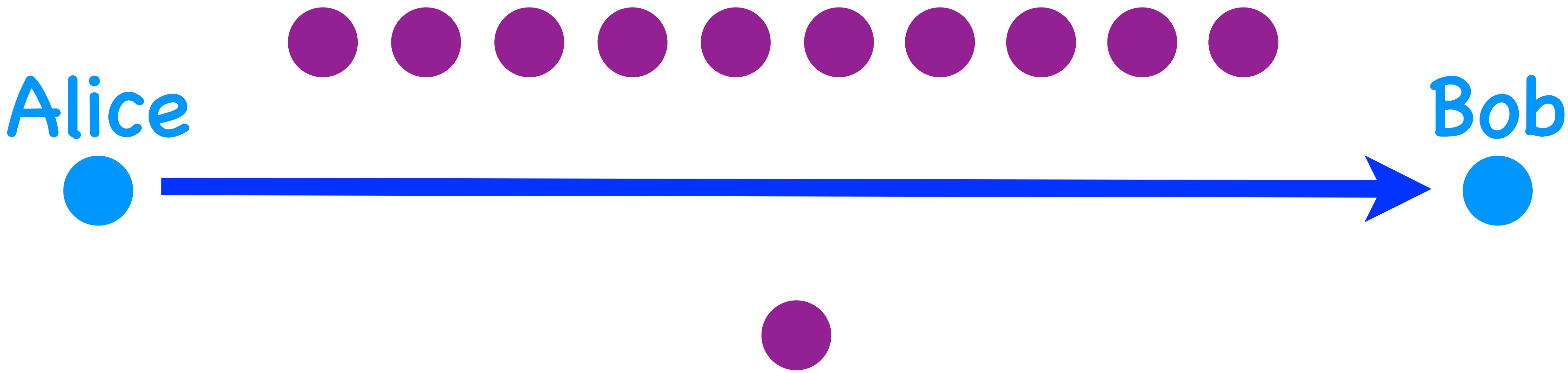
Persa (the impersonator)

pretends that she is Alice to extract information from Bob



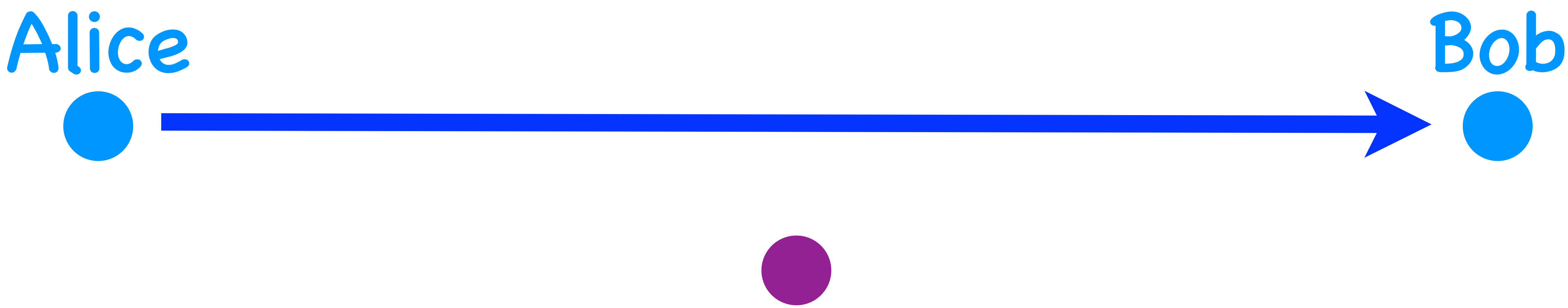
Denis (the denial-of-service attacker)

disrupts the communication
between Alice and Bob



distributed denial-of-service attack

disrupts the communication
between Alice and Bob



Malik (the malware master)

infects Alice and/or Bob

with malware = bad software

Internet vulnerabilities

- Eavesdropping (sniffing)
- Impersonation (spoofing)
- Denial of service (dos-ing)
- Malware

What trust model to design for?

What physical infrastructure is already available?

What modularity & hierarchy?

What layers to define?

Treat on demand or take reservations?

What trust model to design for?