EPFL

POCS Recitation: Review of Networking Basics

Katerina Argyraki

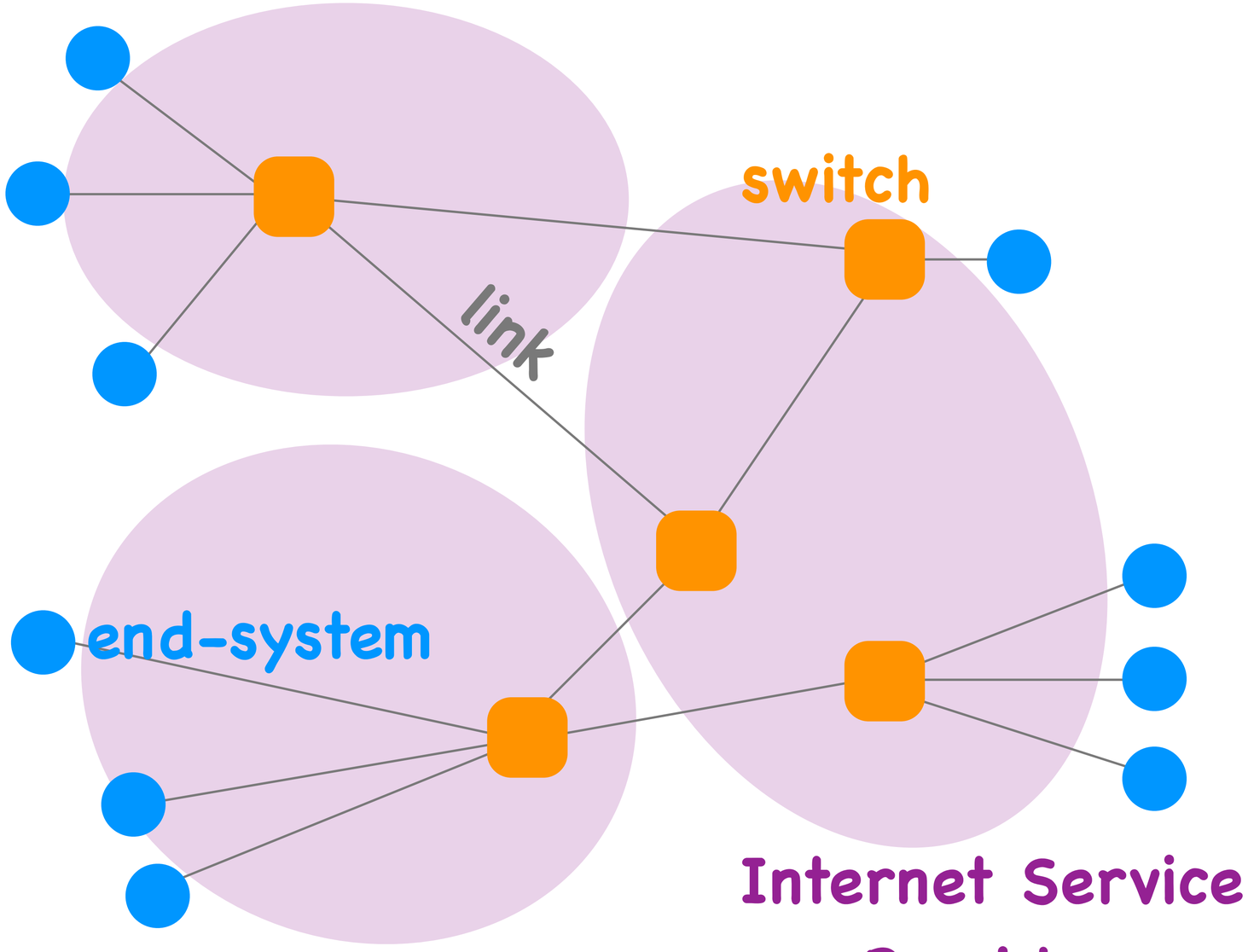
School of Computer & Communication Sciences

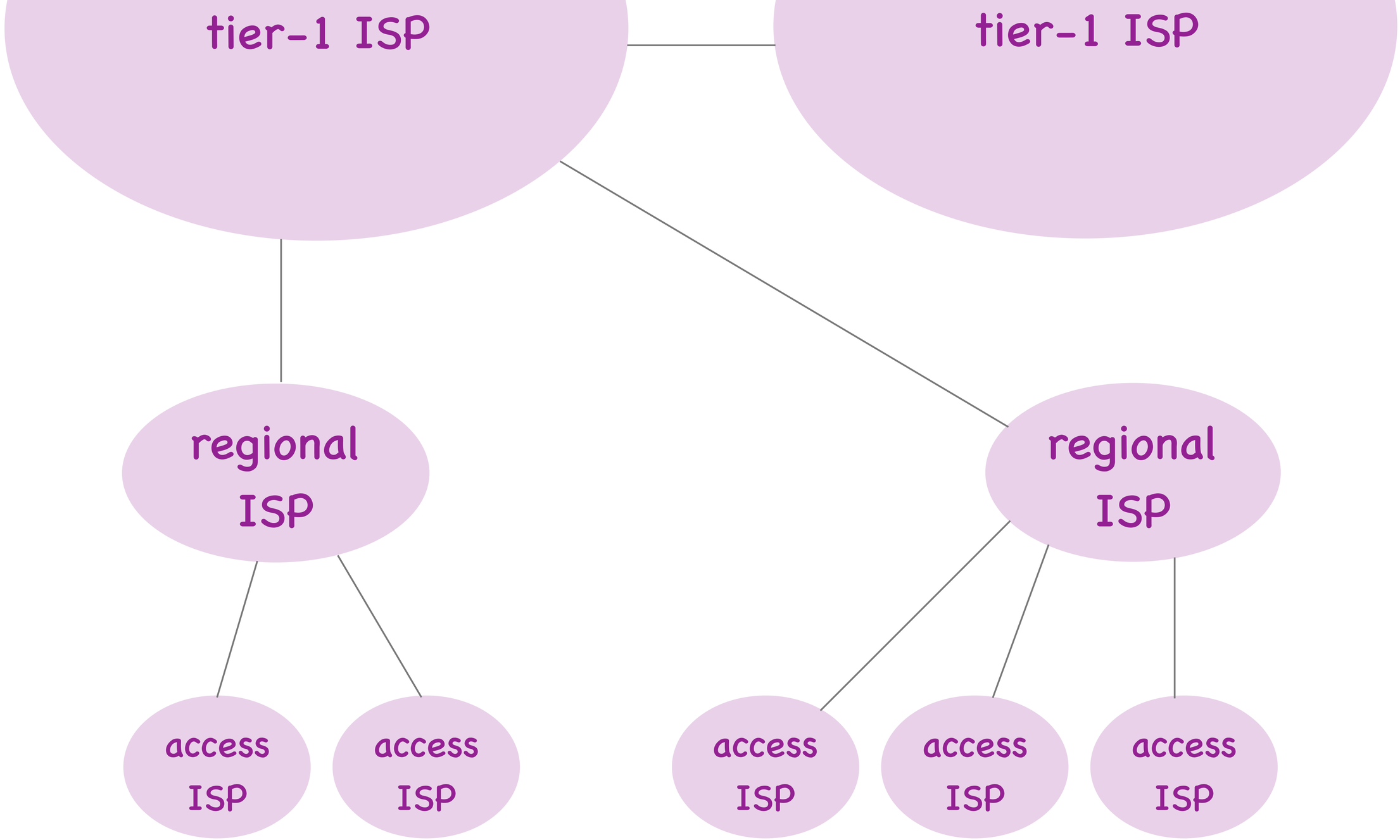
Topics

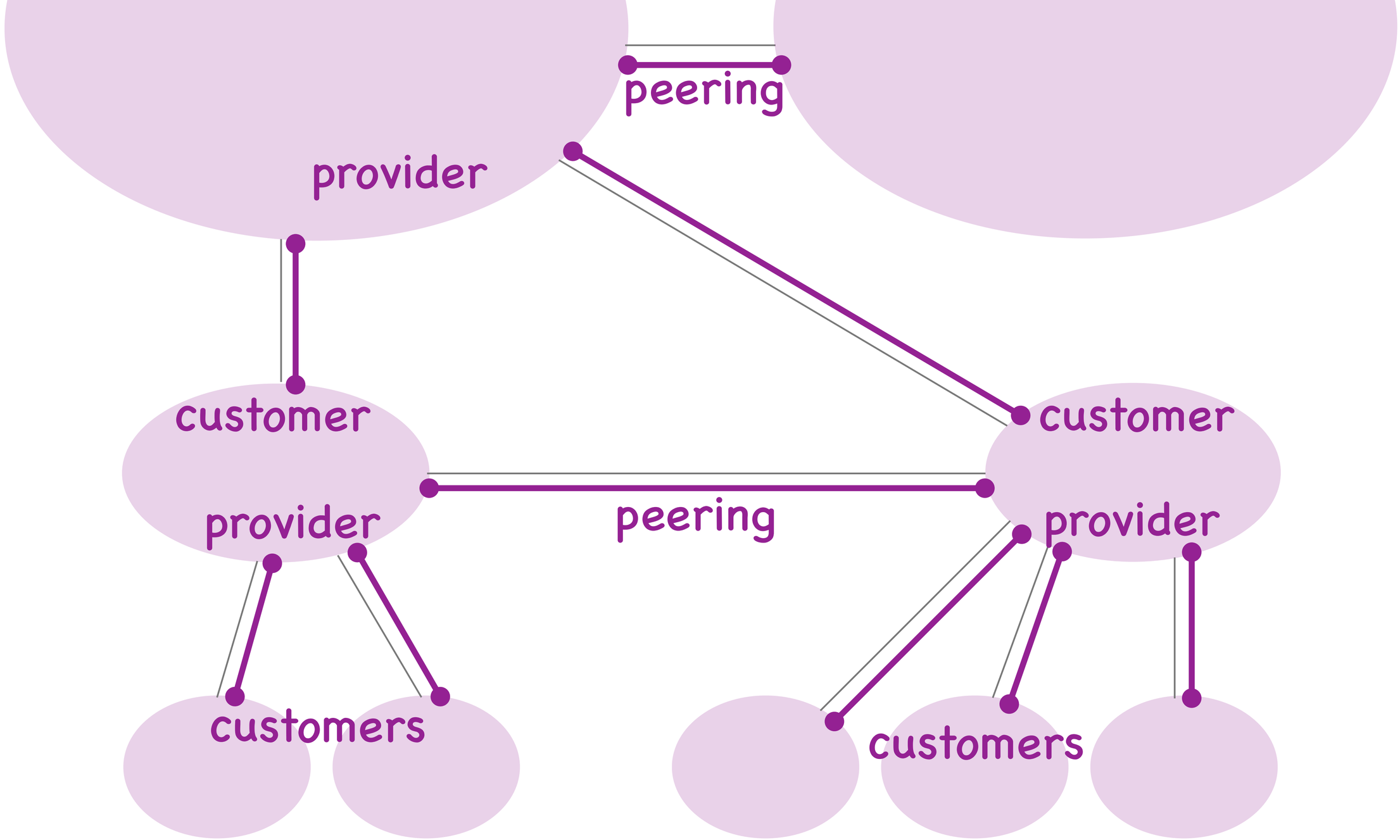
- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer

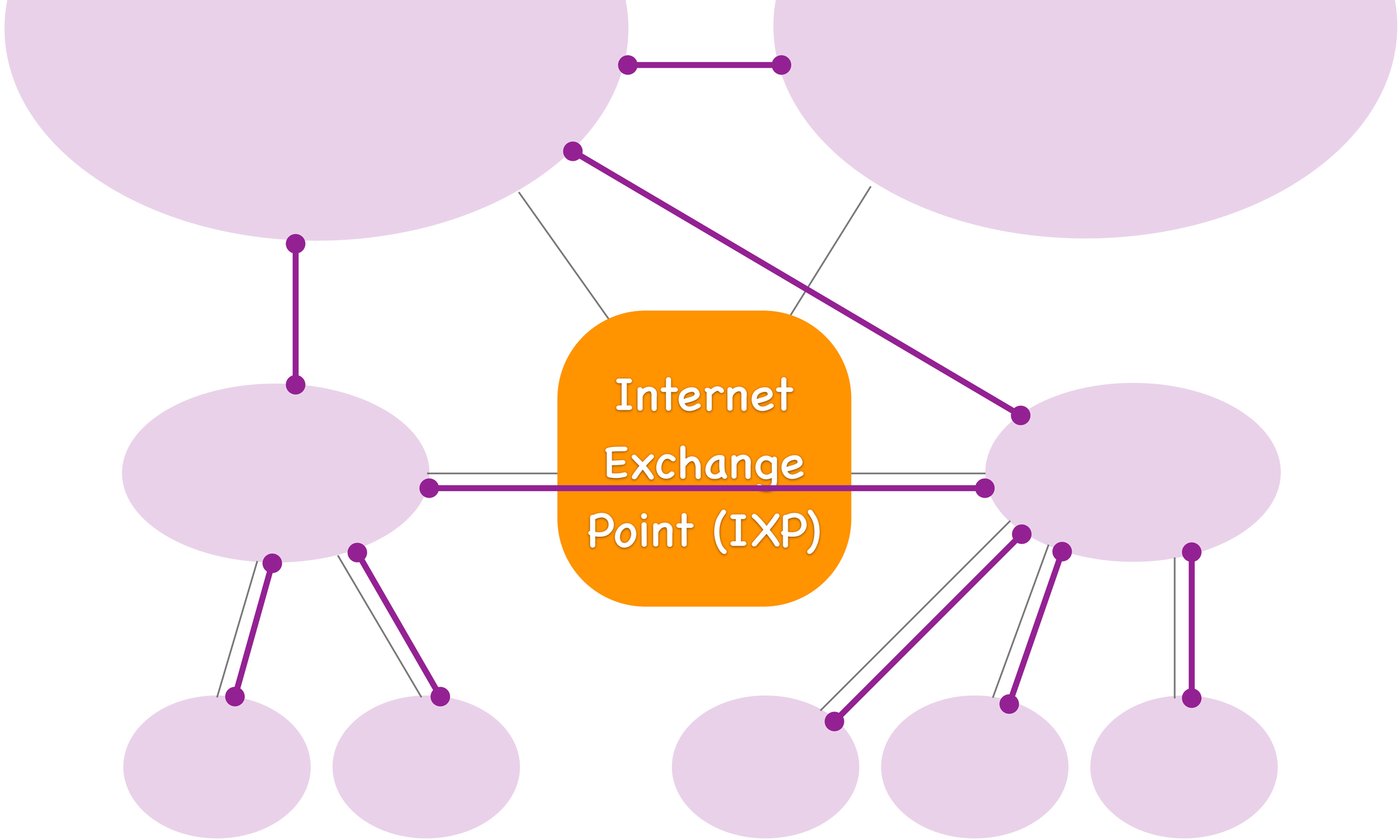
Topics

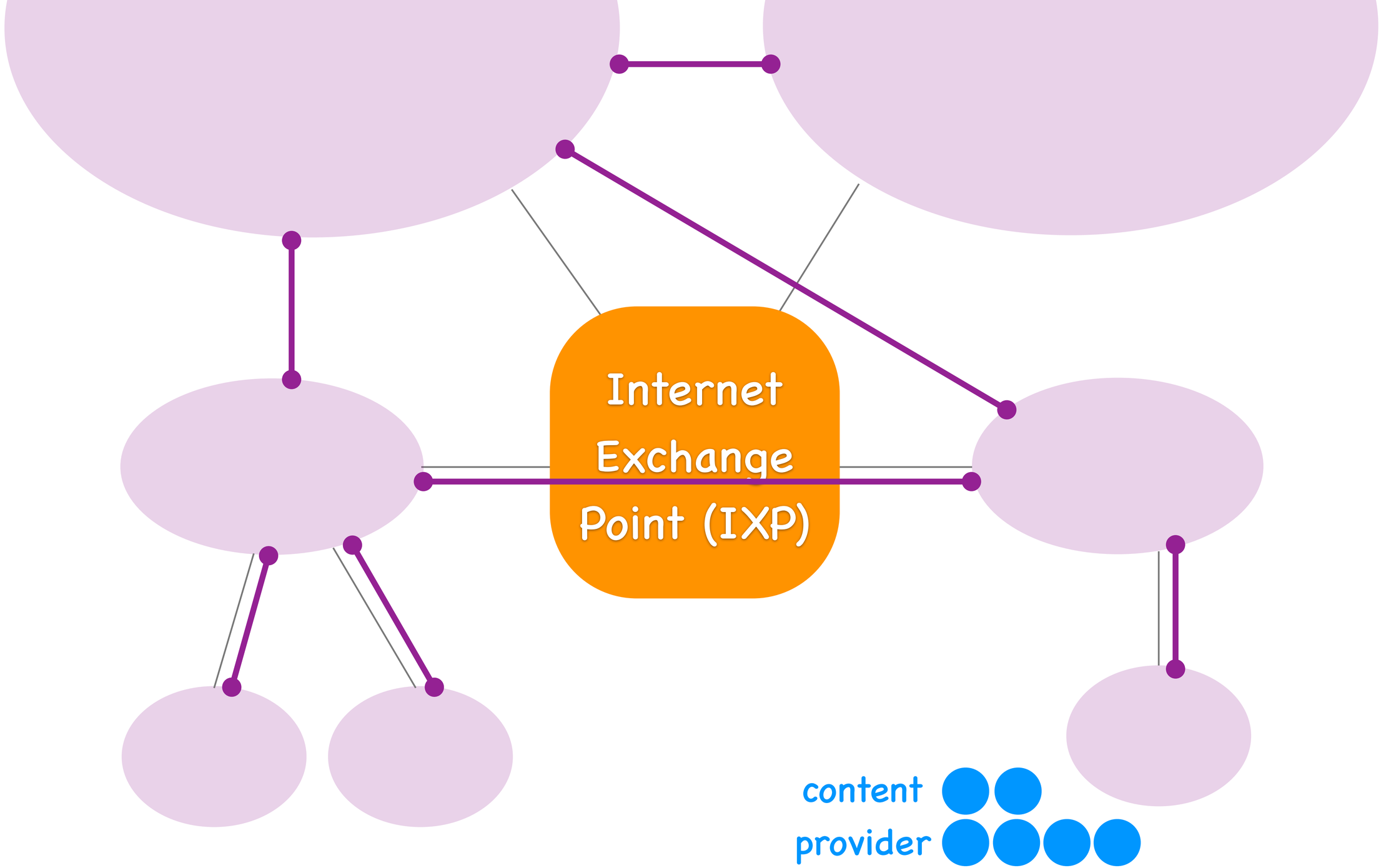
- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer

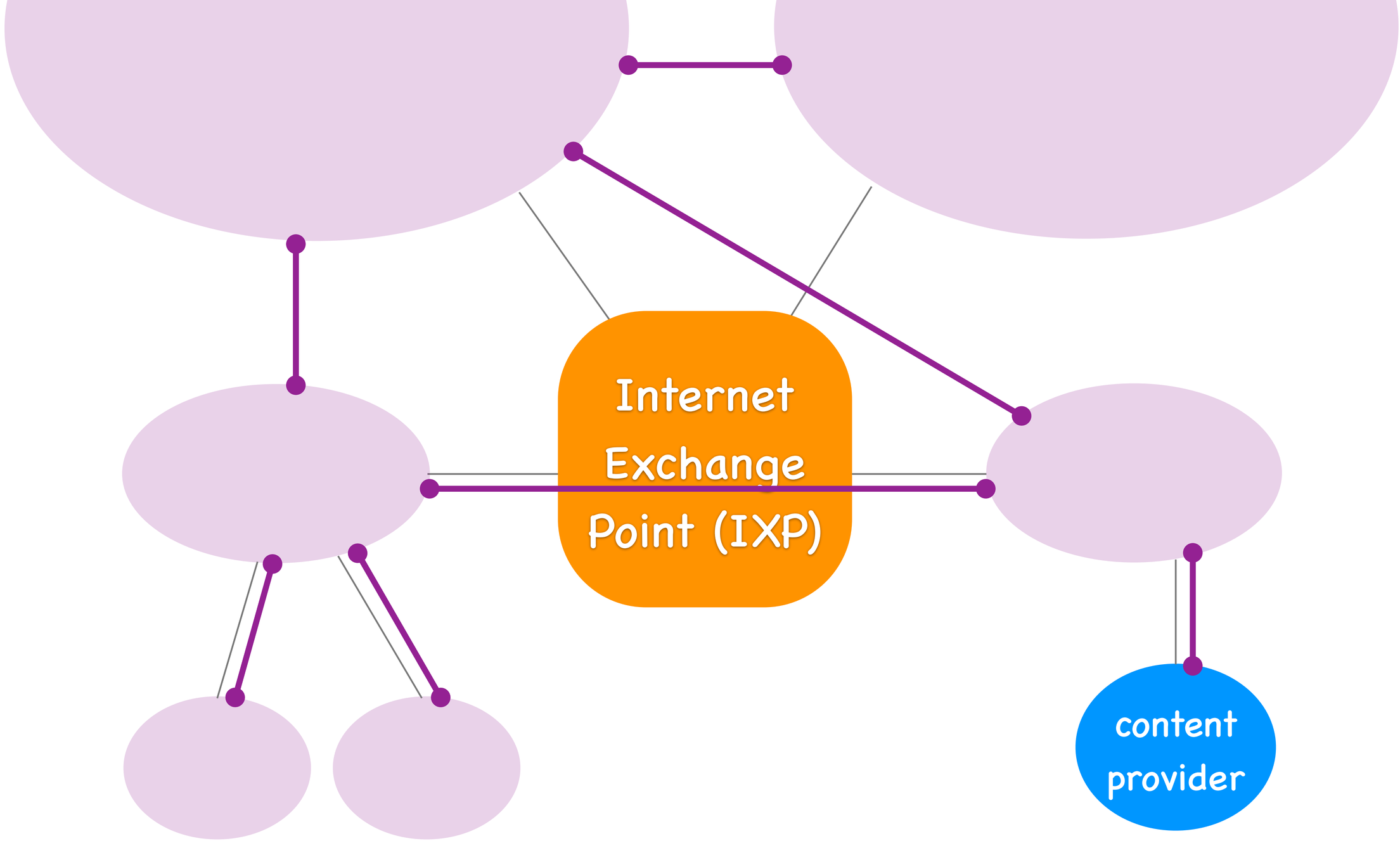

switch
link
end-system
Internet Service Provider

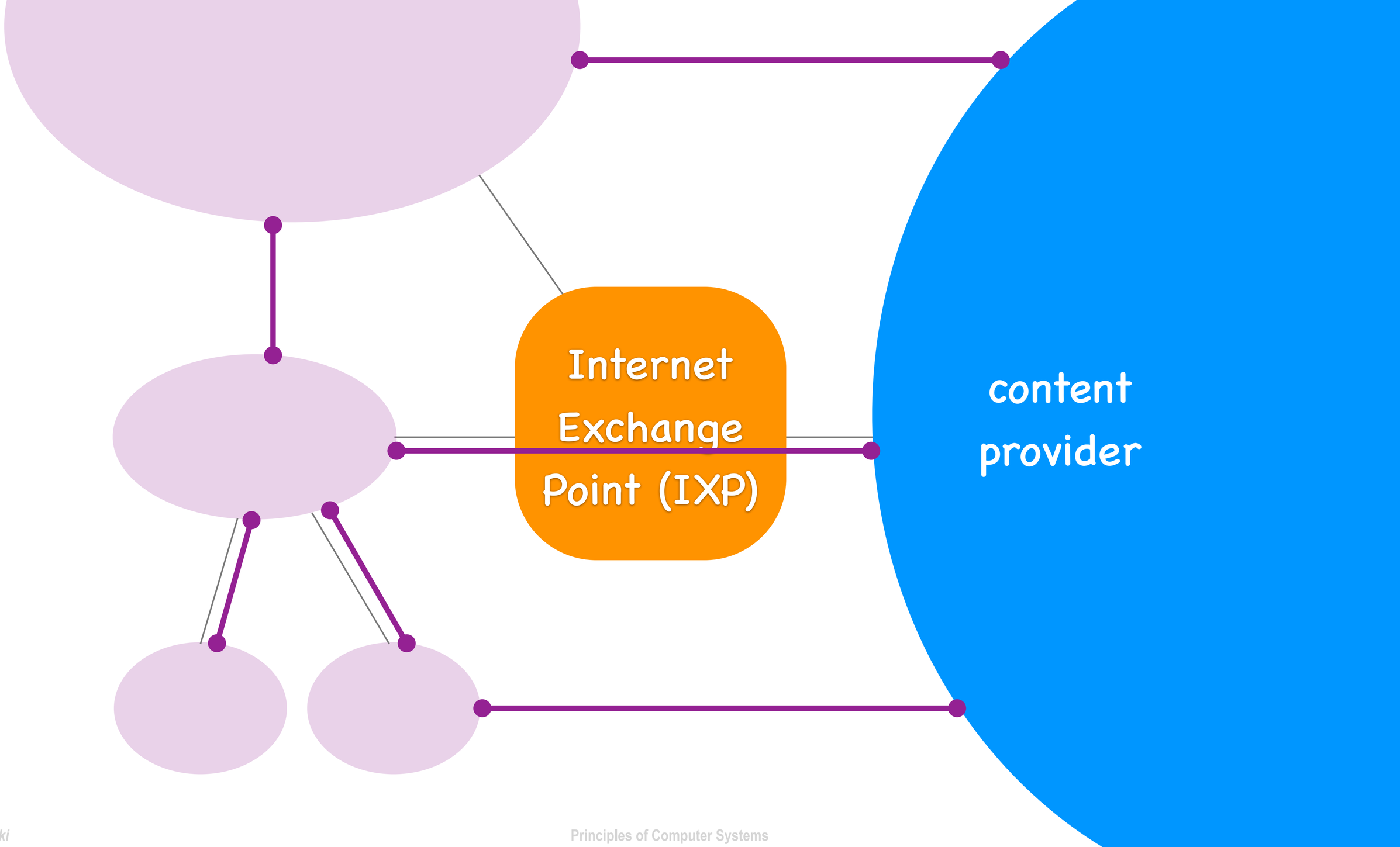

tier-1 ISP
tier-1 ISP
regional ISP
regional ISP
access ISP
access ISP
access ISP
access ISP
access ISP

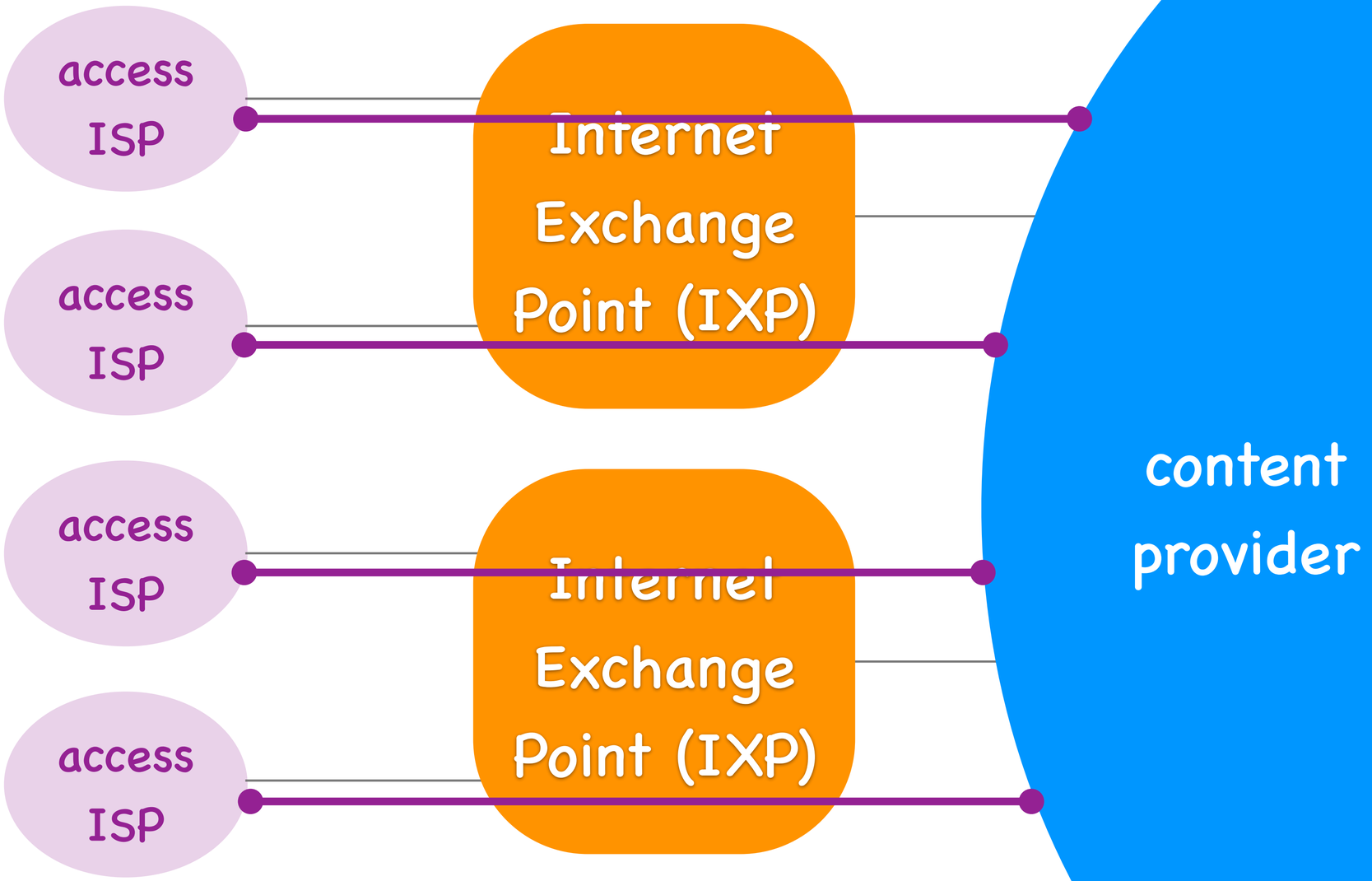

peering
provider
customer
customer
peering
provider
provider
customers
customers

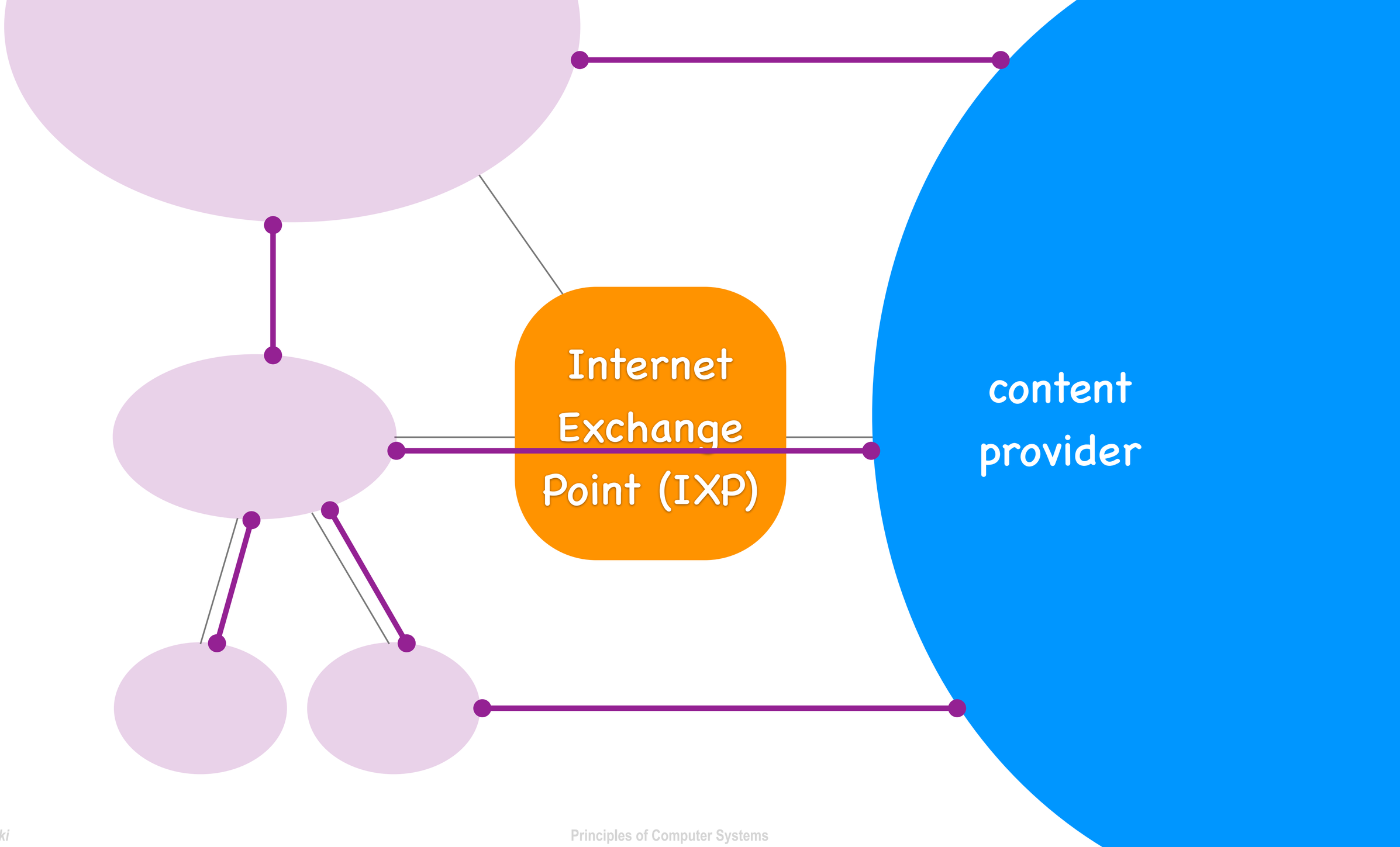

Internet
Exchange
Point (IXP)


Internet
Exchange
Point (IXP)
content
provider


Internet
Exchange
Point (IXP)
content
provider


Internet
Exchange
Point (IXP)
content
provider


access ISP
access ISP
access ISP
access ISP
Internet Exchange Point (IXP)
Internet Exchange Point (IXP)
content provider

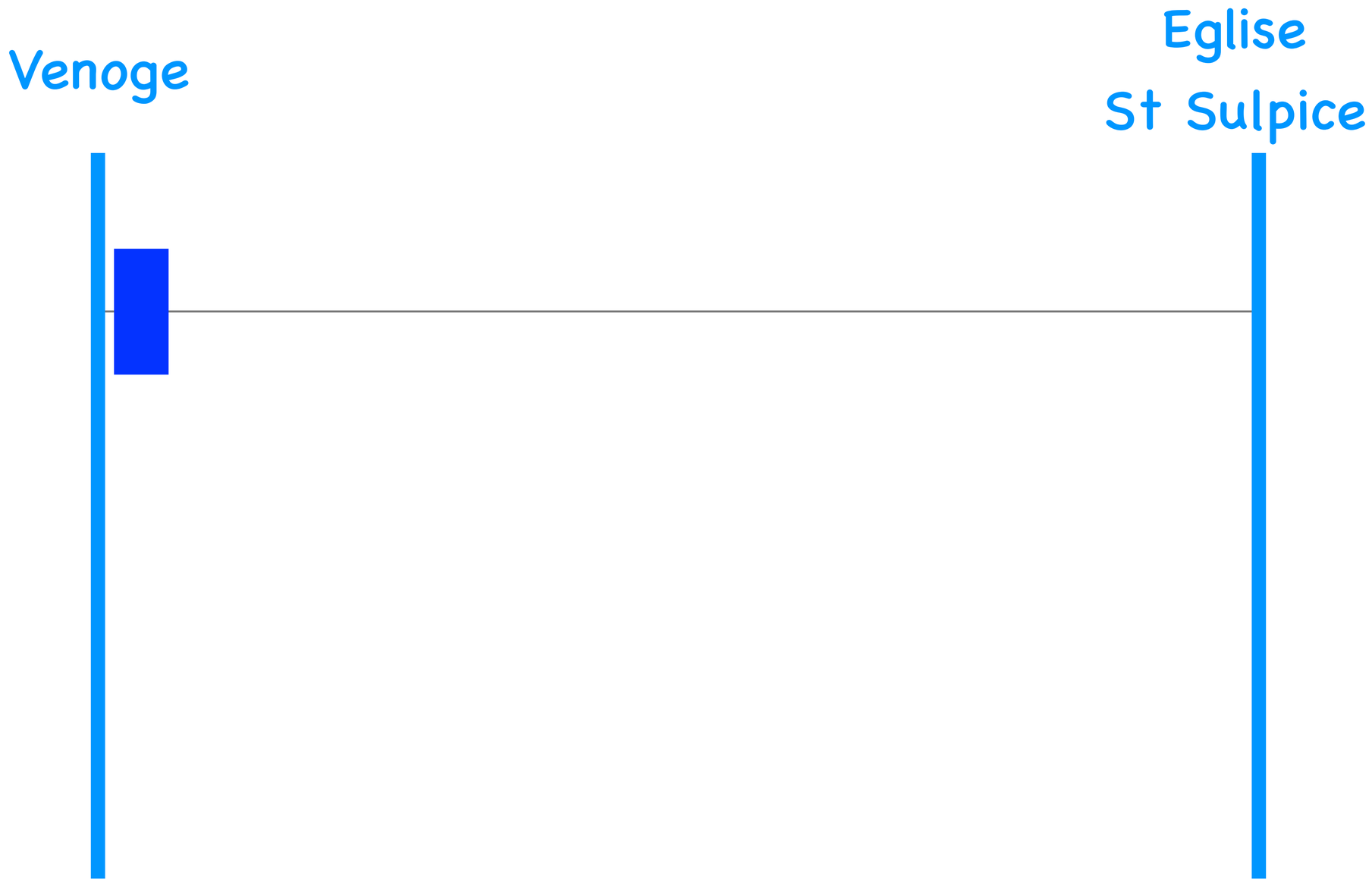

Internet
Exchange
Point (IXP)
content
provider

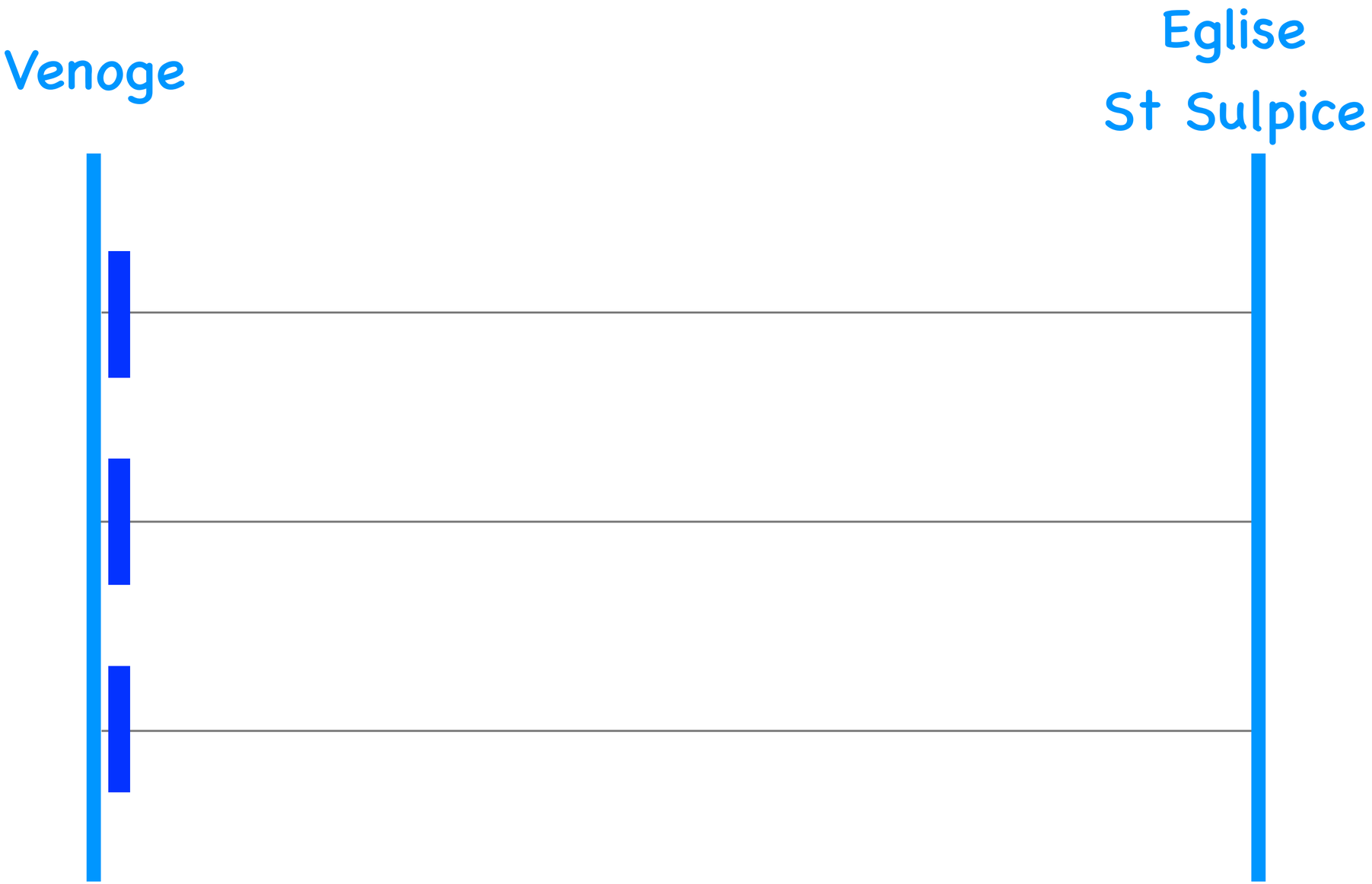
Topics

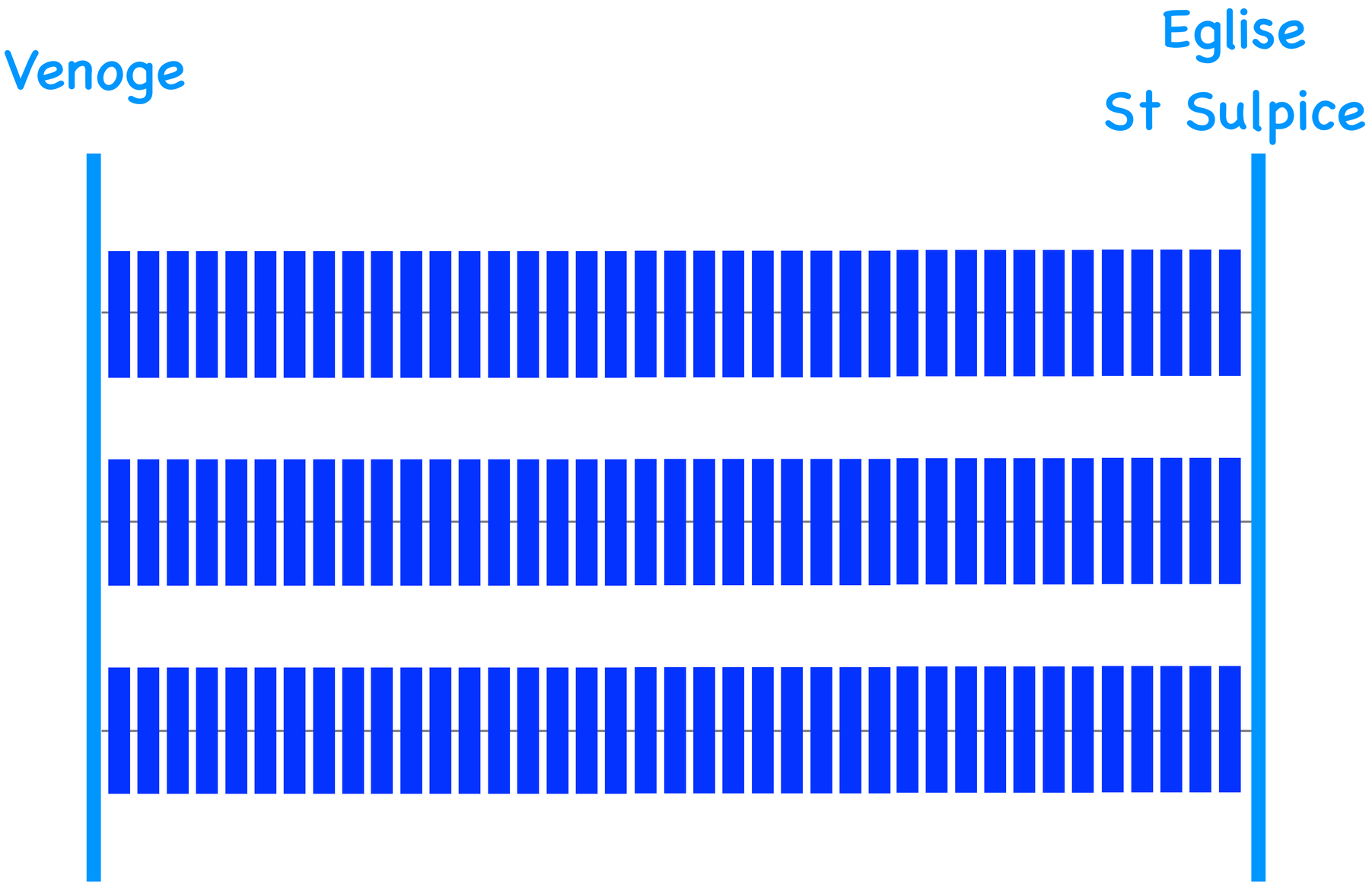
- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer

Network performance metrics

- Packet loss
  - *the fraction of packets from the source to the destination that are lost on the way*
  - *in %, e.g., 1% packet loss*
- Packet delay
  - *the time it takes for a packet to get from the source to the destination*
  - *in time units, e.g., 10 msec*
- Average throughput
  - *the average rate at which the destination receives data from the source*
  - *in bits per second (bps)*


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 messages

- They are related to each other, but not in an obvious way

transmission delay

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

propagation delay

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

packet delay =

transmission delay

+ propagation delay

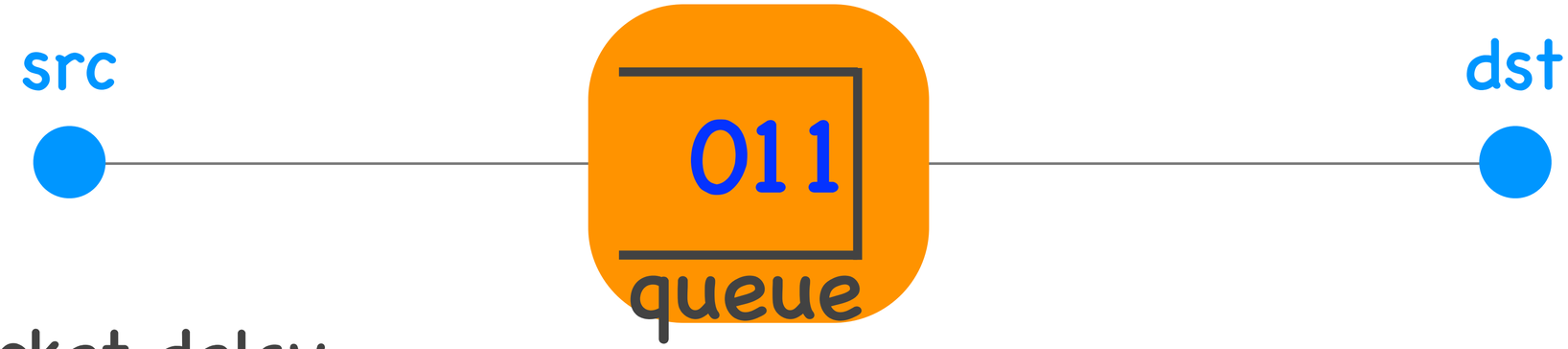
store & forward switch

packet delay =

transmission delay over 1st link

+ propagation delay of 1st link

store & forward switch



packet delay =

transmission delay over 1st link

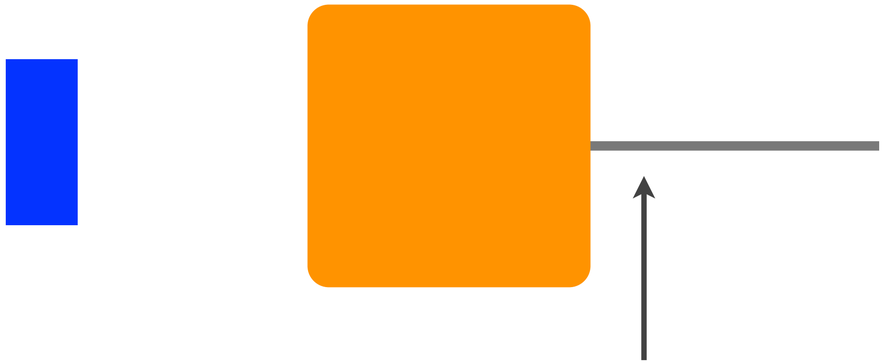
+ propagation delay of 1st link

+ queuing delay

+ processing delay

+ transmission delay over 2nd link

+ propagation delay of 2nd link



transmission rate:

R bits/sec

bit arrival rate:

A bits/sec

bit departure rate:

R bits/sec

bit arrival rate:

A bits/sec

>

bit departure rate:

R bits/sec

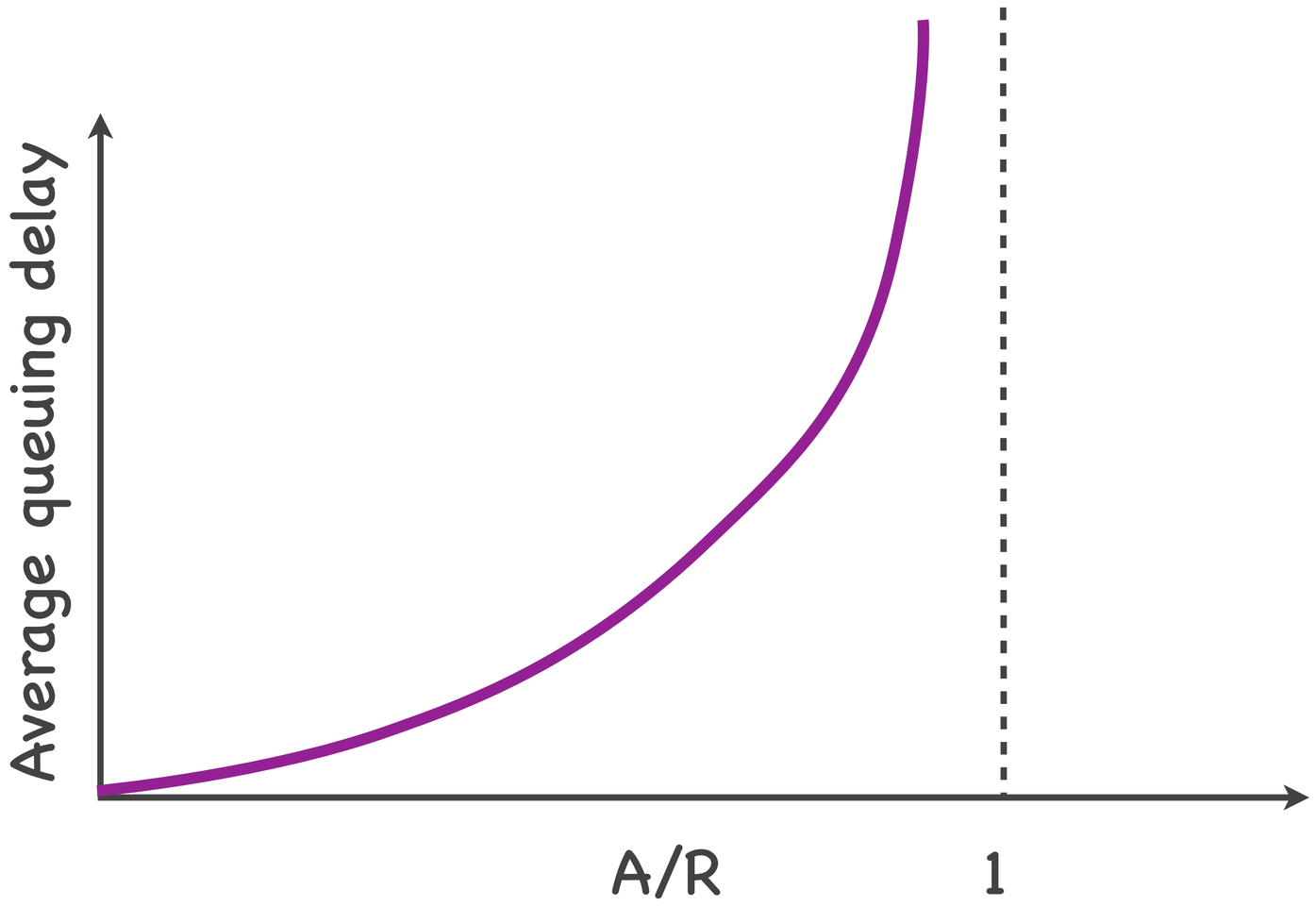
bit arrival rate:

A bits/sec

<=

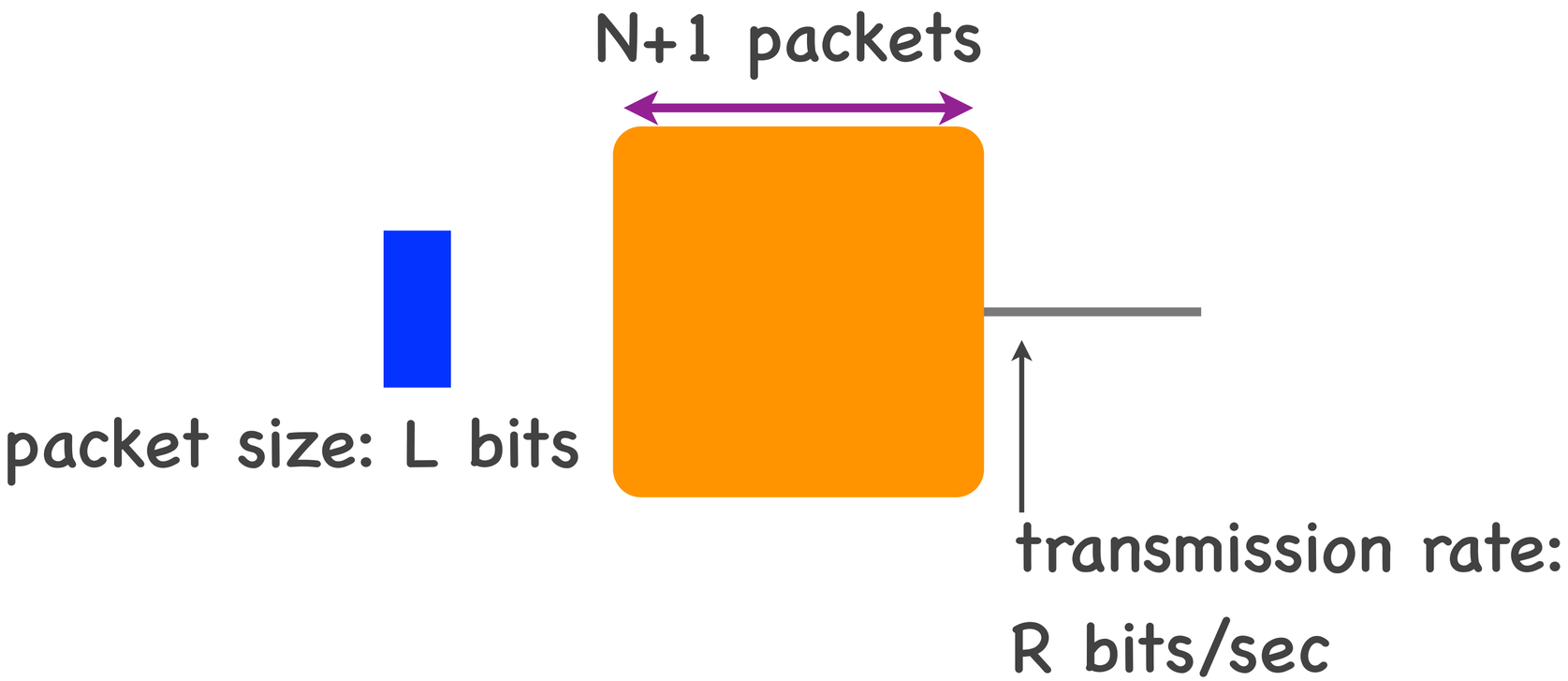
bit departure rate:

R bits/sec


Average queuing delay
A/R
1

Queuing delay

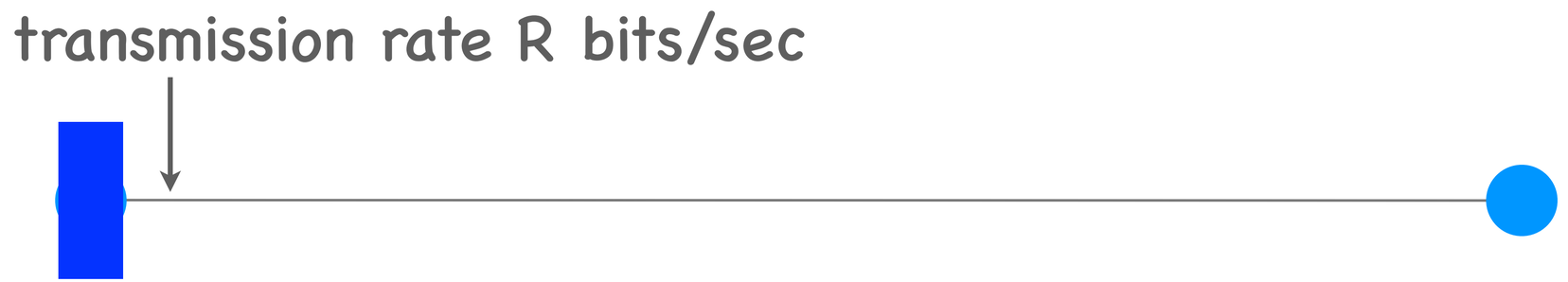
- (Assuming infinite queue size)
- Approaches infinity, if arrival rate > departure rate
- Depends on burst size, otherwise



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

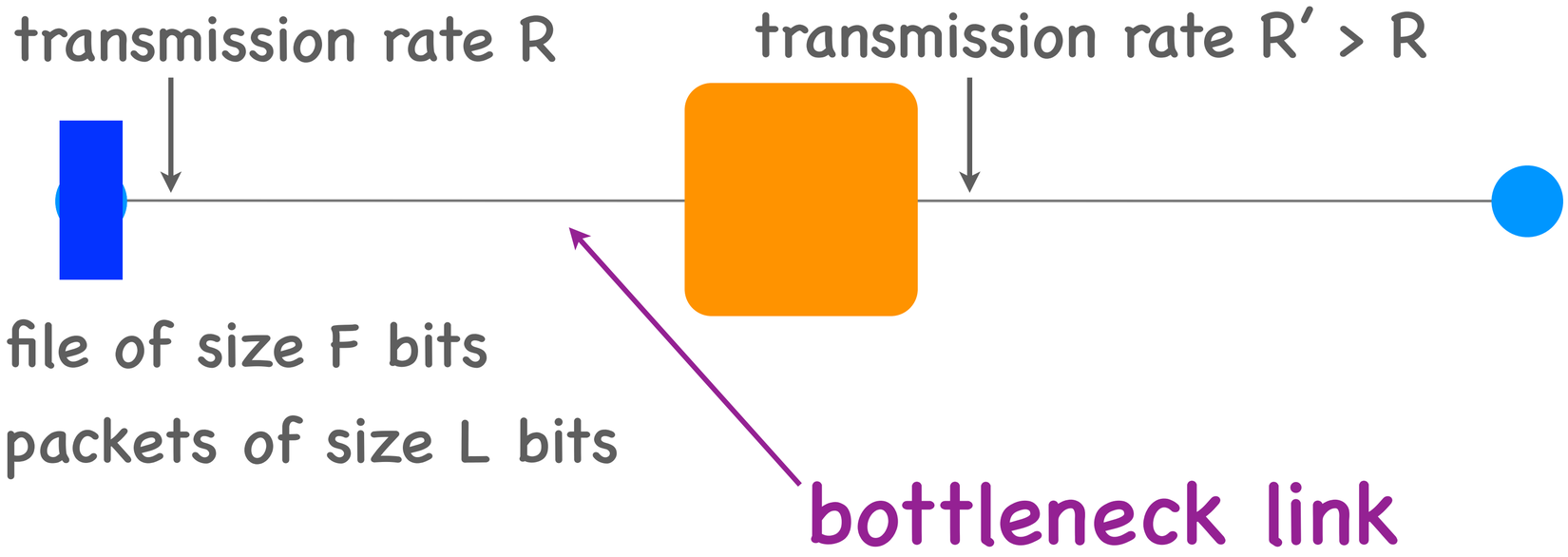


file of size F bits

packets of size L bits

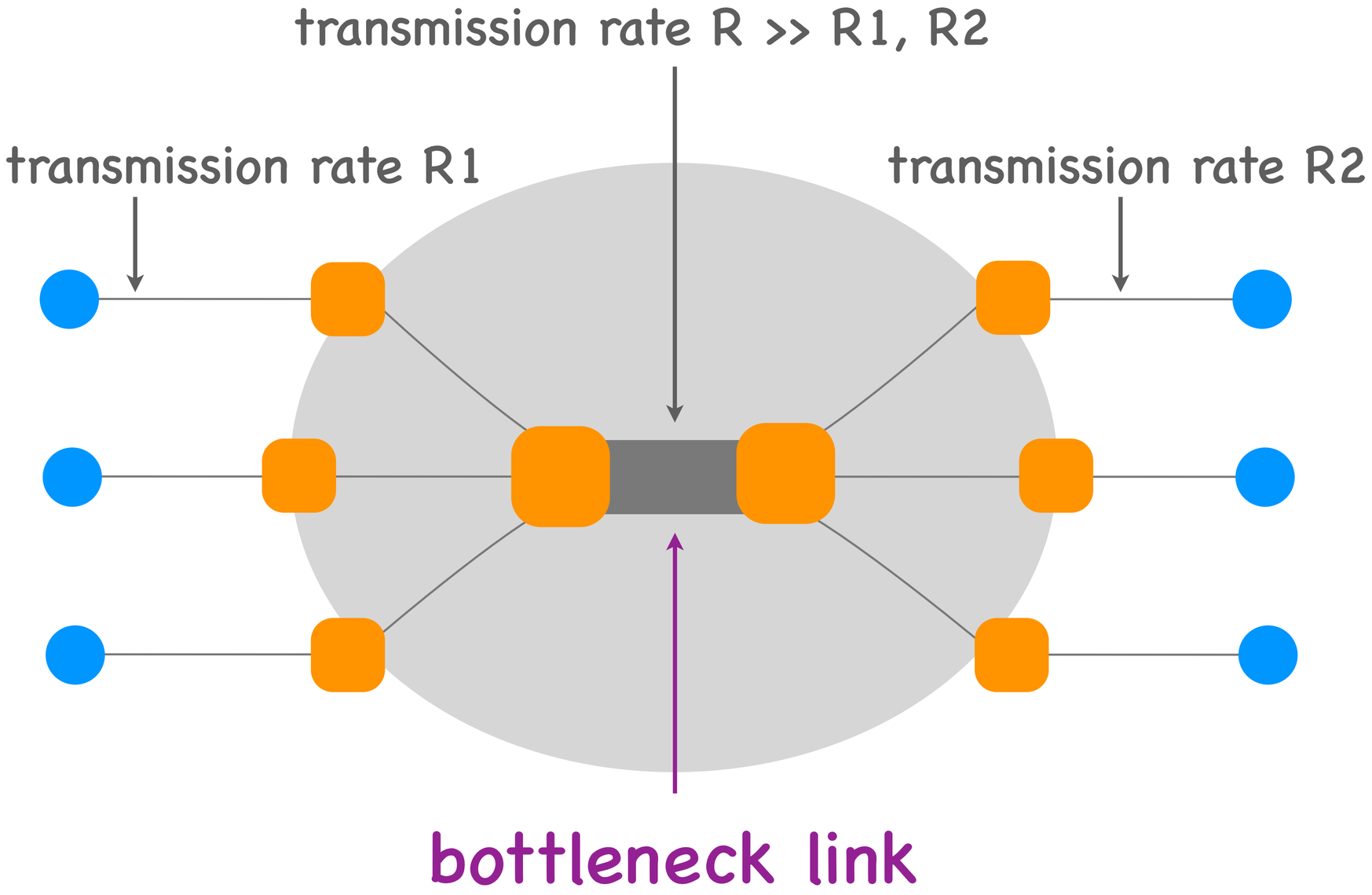
Transfer time = F/R + propagation delay

Average throughput = R



Transfer time = F/R + propagation delay 1st link
+ L/R' + propagation delay 2nd link

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

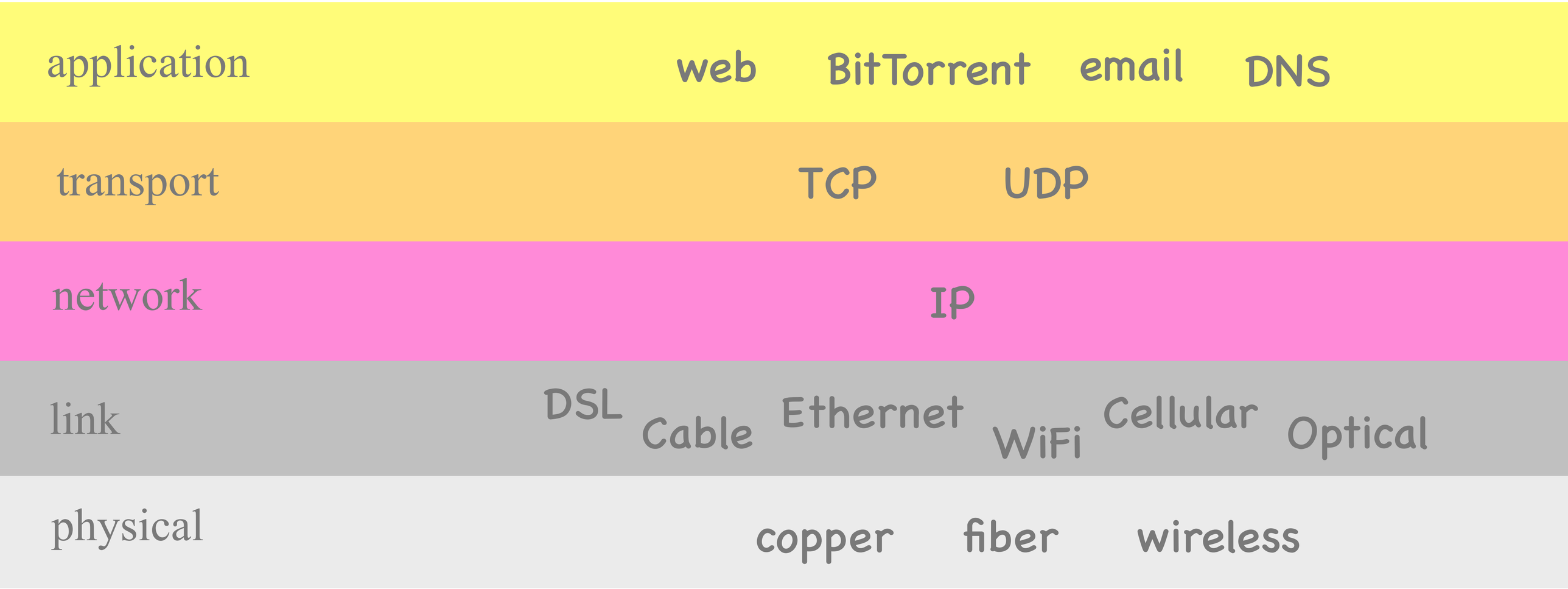

transmission rate R >> R1, R2
transmission rate R1
transmission rate R2
bottleneck link

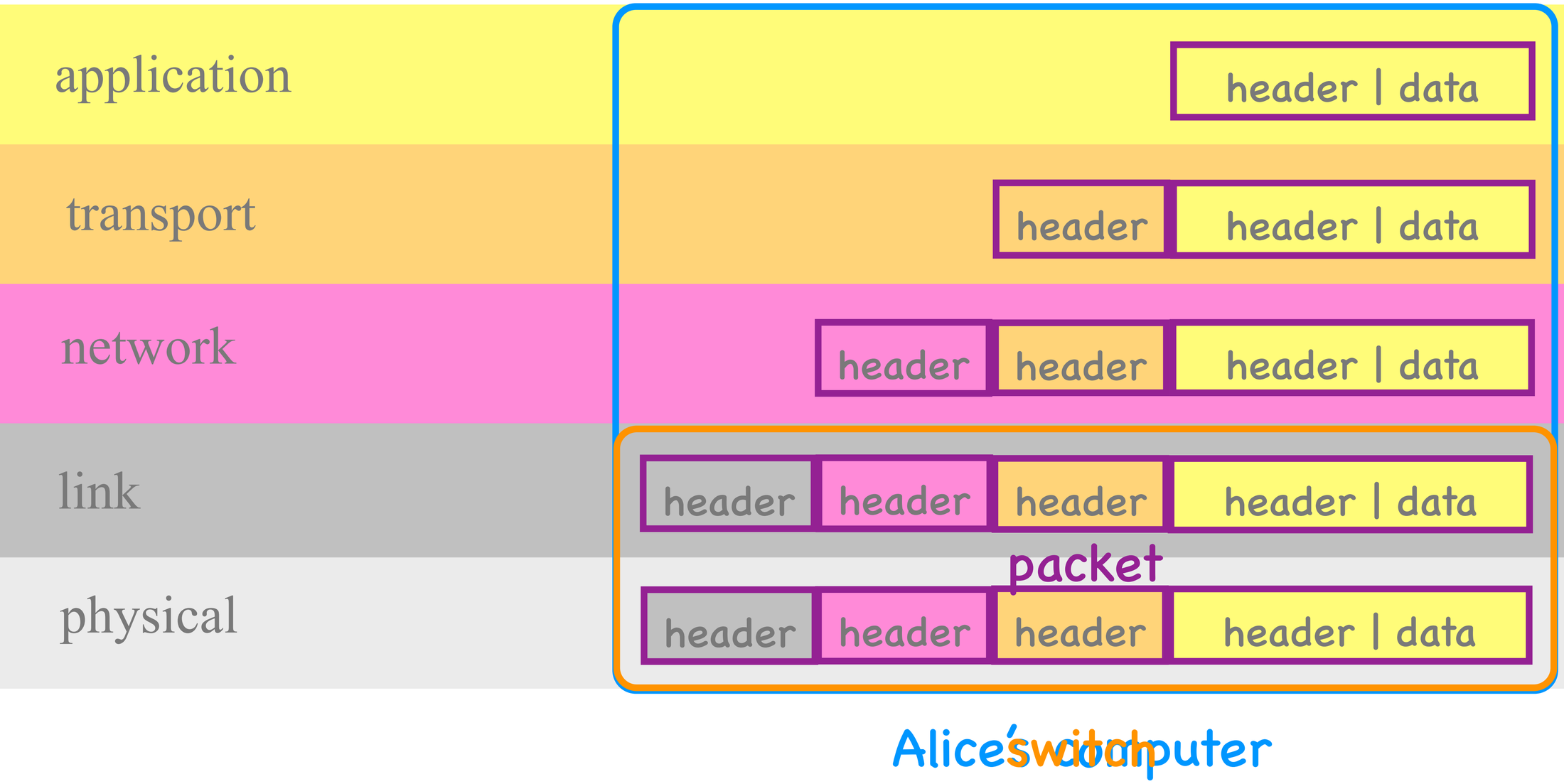
Average throughput

- Determined by the bottleneck link
  between the source and the destination
- = the link where traffic between the source and the destination
  flows at the slowest rate
- Could be because of the link's transmission rate
  or because of queuing delay

Topics

- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer


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


application
transport
network
link
physical
header | data
header
header | data
header
header
header | data
header
header
header
header | data
packet
header
header
header
header | data
Alice's computer
switch

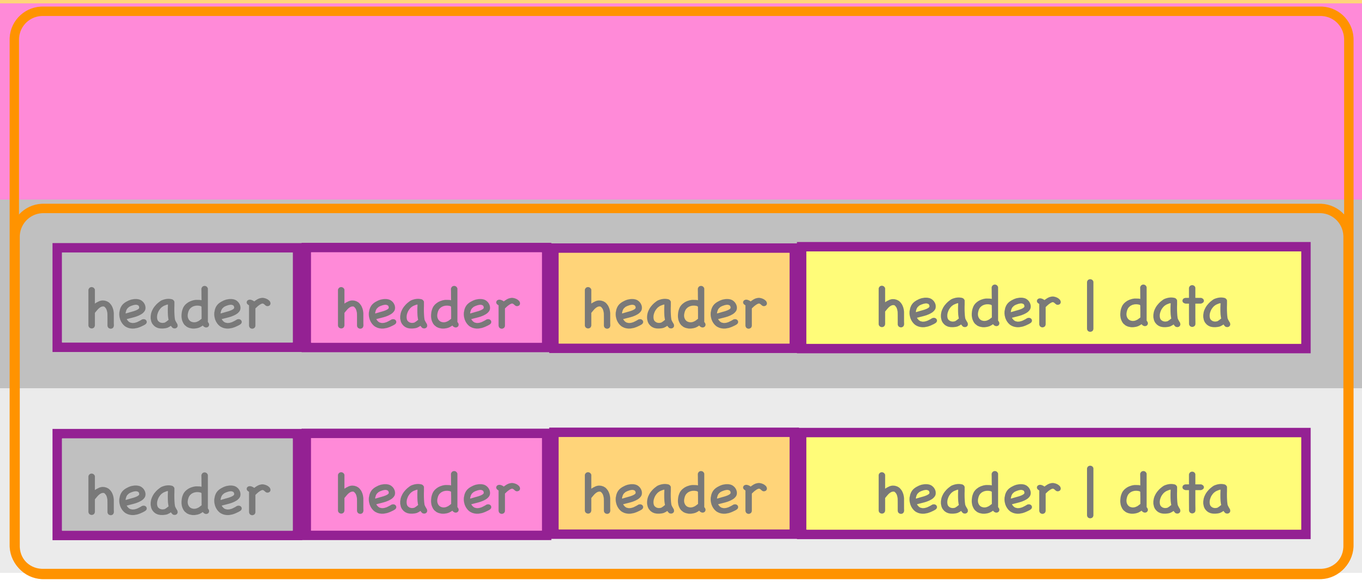
application

transport

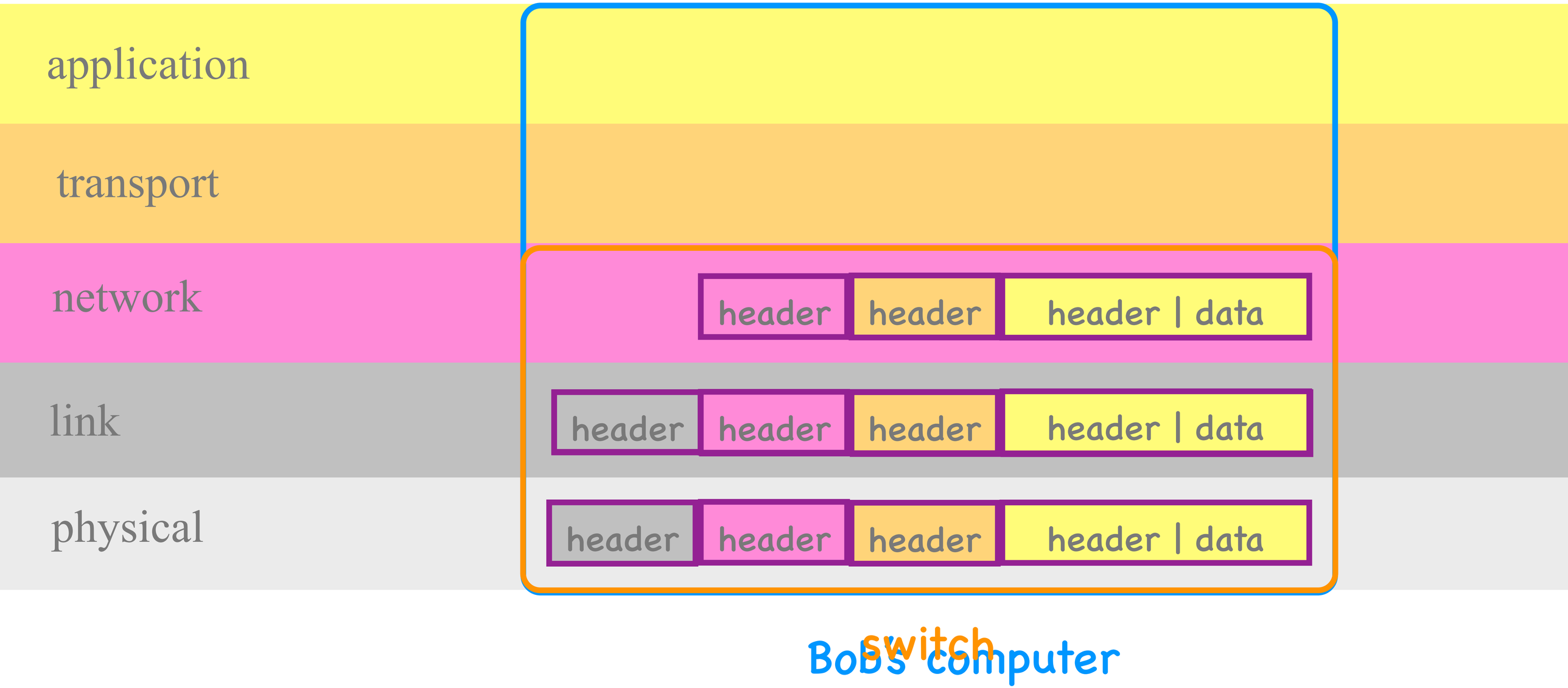
network

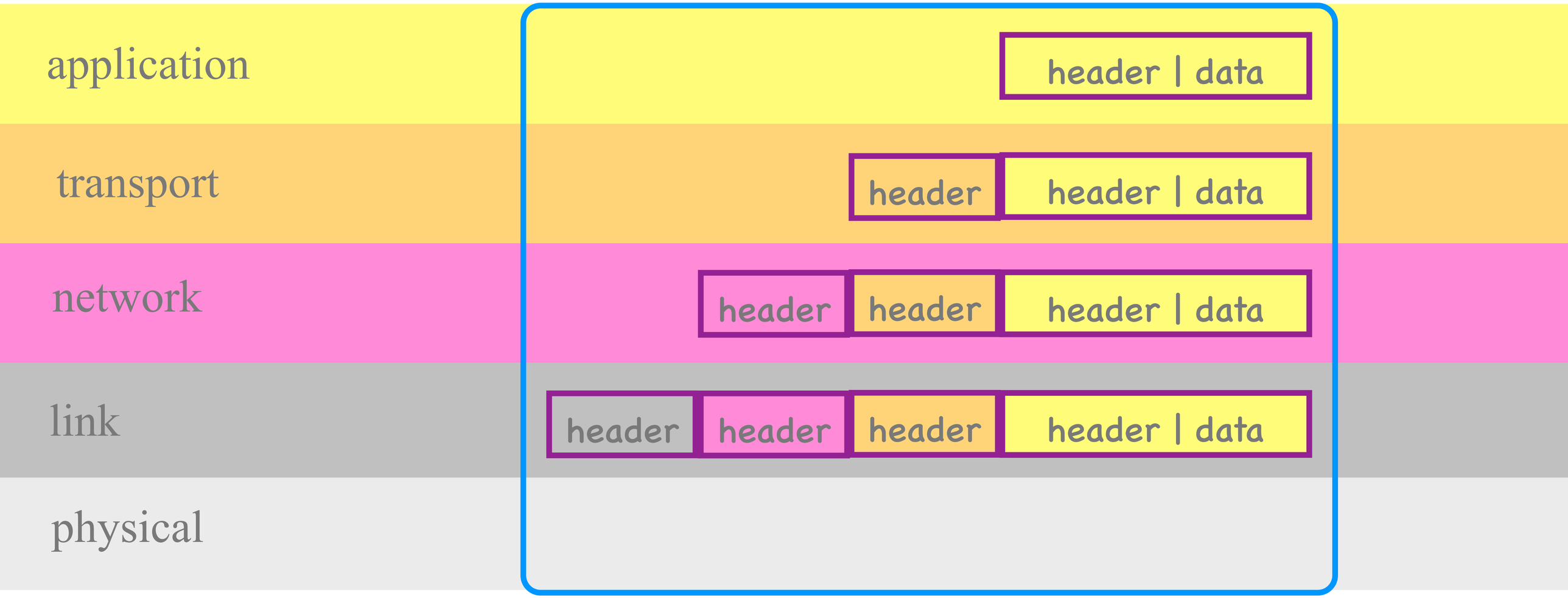
link

physical



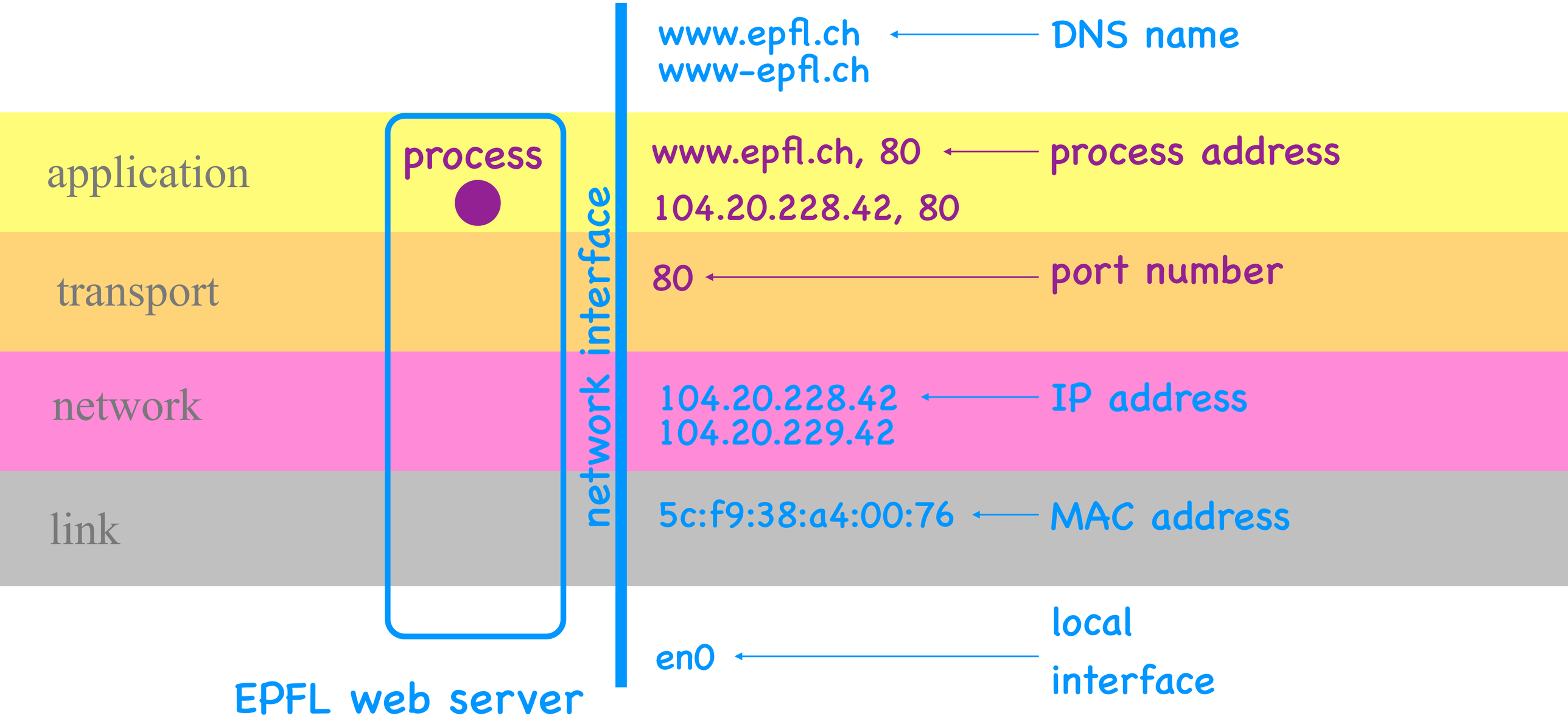
switch


application
transport
network
link
physical
header
header
header | data
header
header
header
header | data
header
header
header
header | data
switch
BoBS computer


application
transport
network
link
physical
header | data
header
header | data
header
header
header | data
header
header
header
header | data
Bob's computer

Layers

- Each layer touches only the header of the same layer
- May add a new header = encapsulation
- May remove the header = decapsulation


www.epfl.ch
www-epfl.ch
DNS name
application
process
www.epfl.ch, 80
process address
104.20.228.42, 80
transport
80
port number
network
104.20.228.42
104.20.229.42
IP address
link
5c:f9:38:a4:00:76
MAC address
network interface
en0
local
interface
name
EPFL web server

Topics

- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer

Domain Name Service (DNS)

- Maps DNS names to IP addresses (among other things)
- Hierarchy of DNS servers for scalability:
  root, top-level domain (TLD), authoritative
- Caching of mappings at DNS servers and clients for performance
- Expiration dates associated with each mapping for consistency between cached and authoritative data

Name lookup

- The client sends the name lookup to a local DNS server
  - *the local DNS server is outside the root/TLD/auth. hierarchy*

- The local DNS server may forward the name lookup to another server
  - *recursive query: each server that receives the name lookup either responds with the answer or forwards the name lookup to another server*
  - *iterative query: each server that receives the name lookup responds, either with the answer or with the name of another server that can provide the answer*

Topics

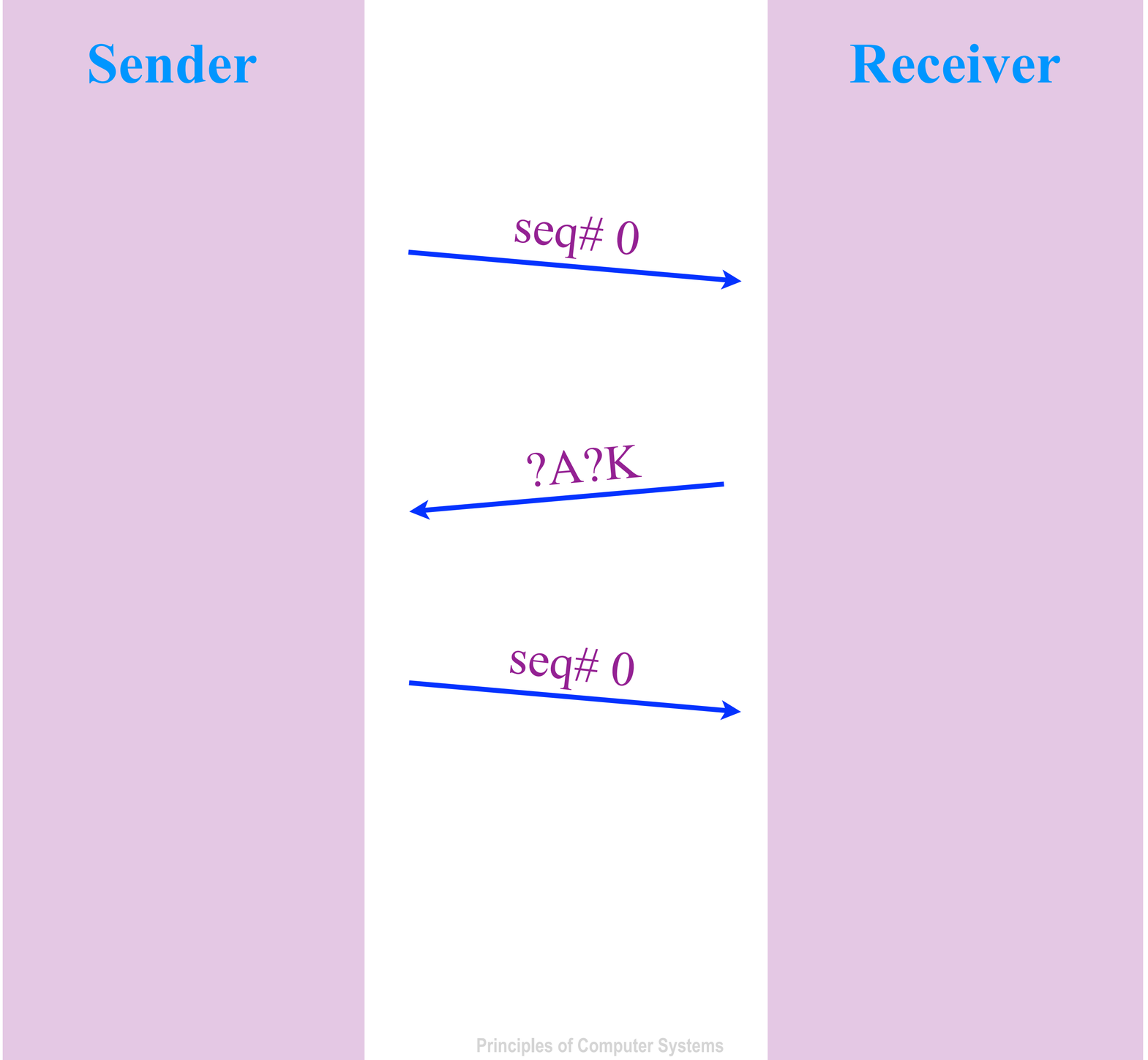
- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer

TCP services

- Reliable data delivery: to recover from corruption and loss
- Flow control: to avoid overloading the receiver
- Congestion control: to avoid overloading the network

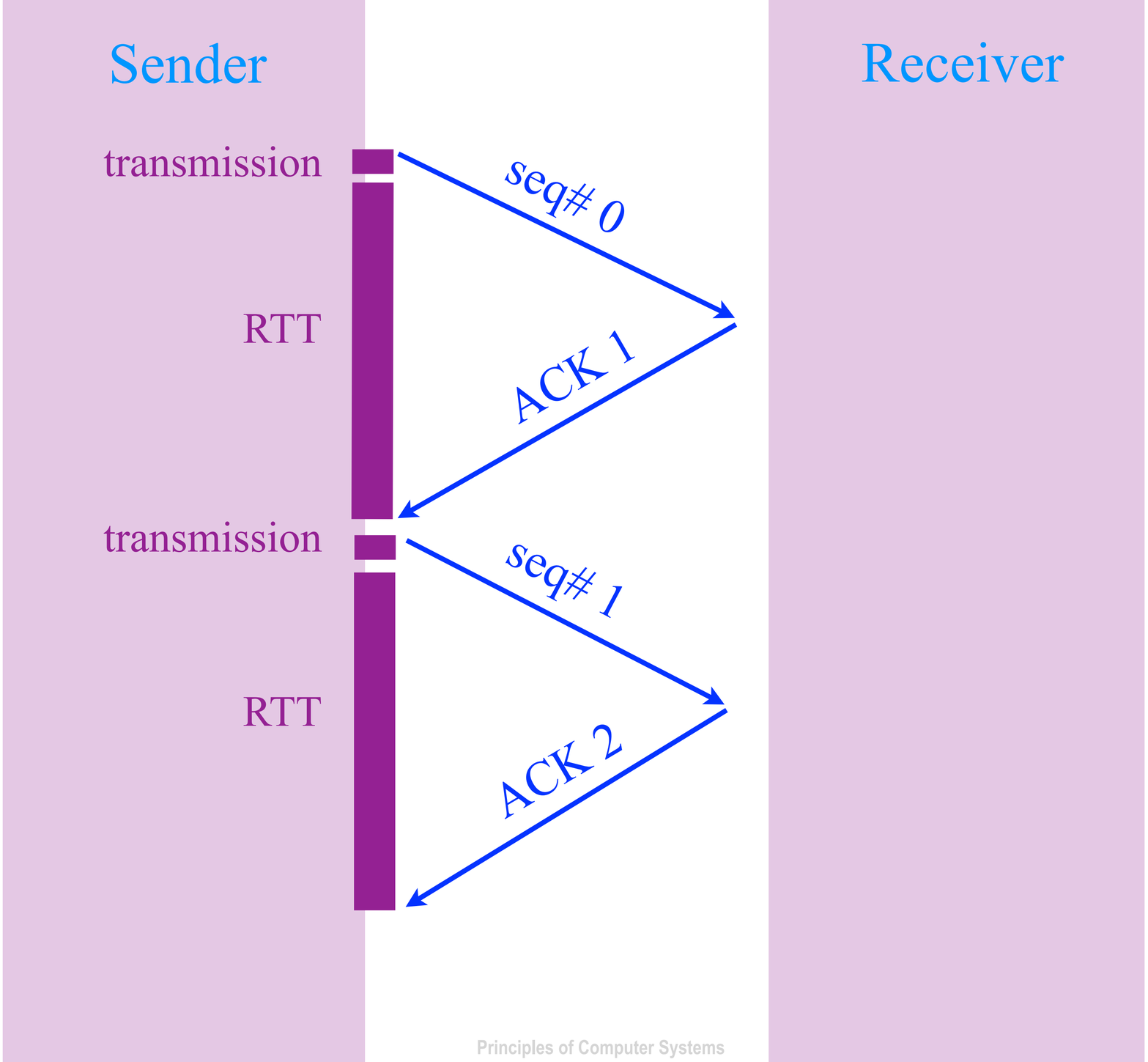
Reliable data delivery

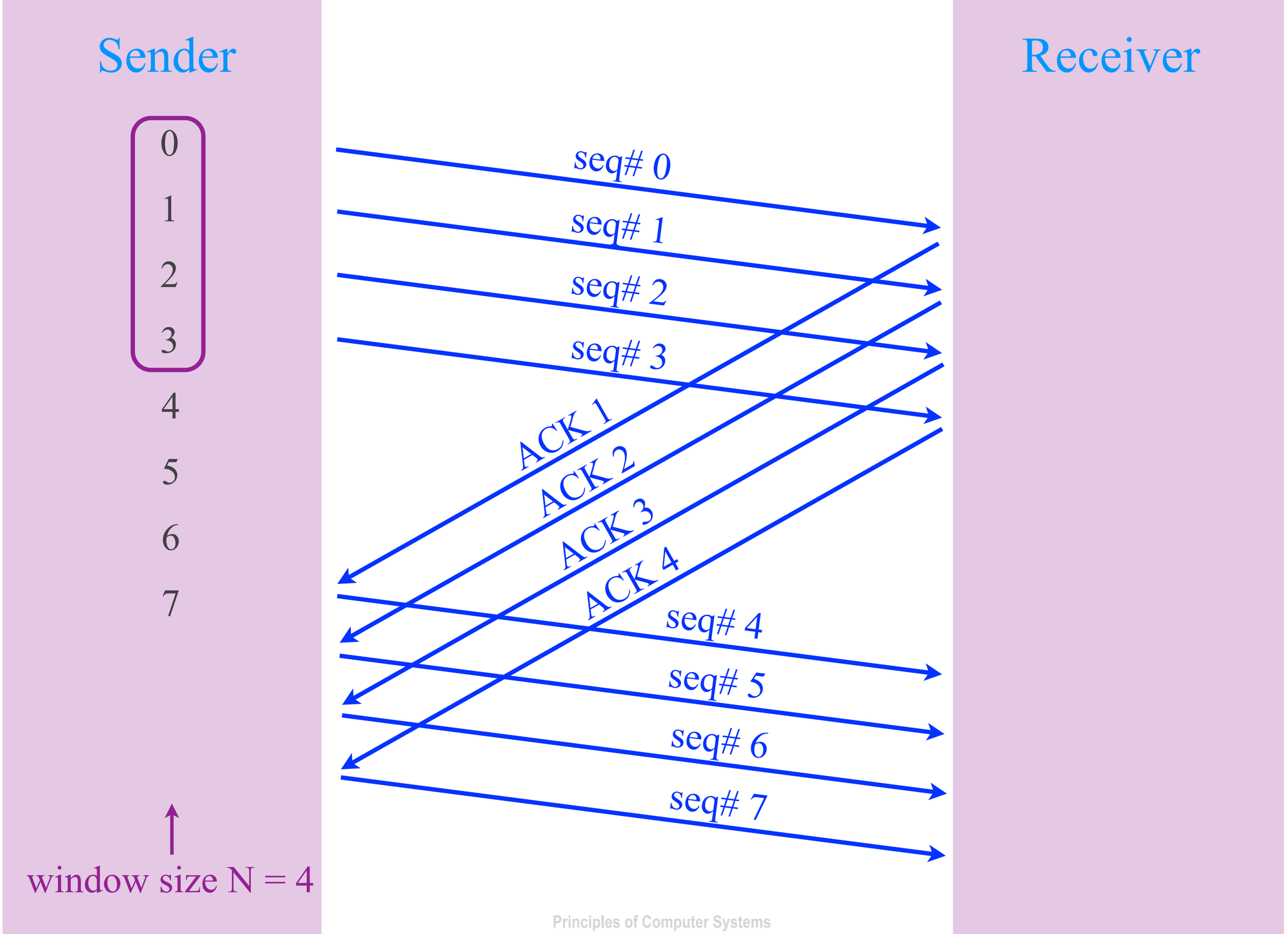
- Checksums to detect packet corruption
- Timeouts to detect packet loss
- Acknowledgments and retransmissions to recover from packet loss and corruption


Sender
Receiver
seq# 0
?A?K
seq# 0

Reliable data delivery

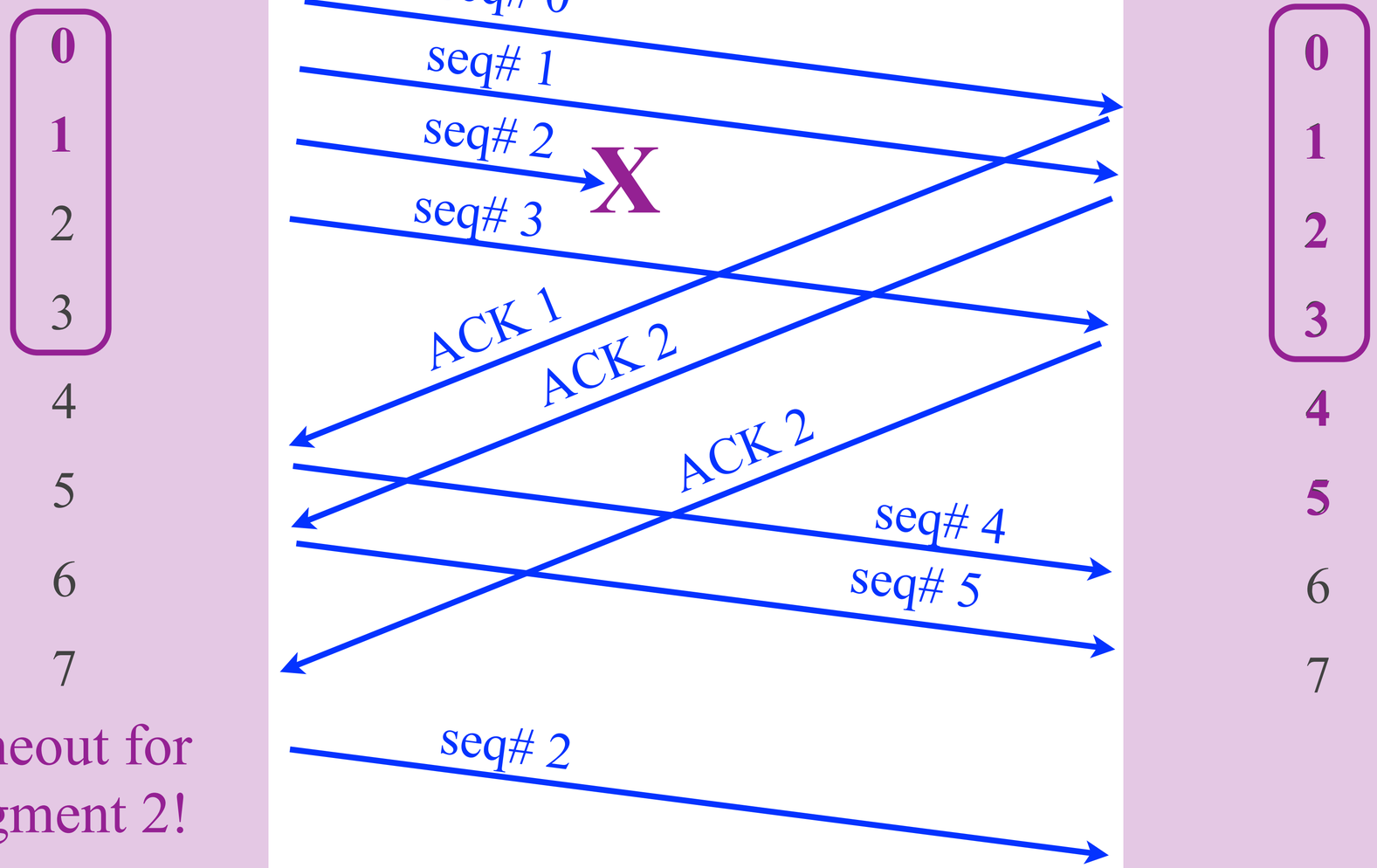
- Checksums to detect packet corruption
- Timeouts to detect packet loss
- Acknowledgments and retransmissions to recover from packet loss and corruption
- Sequence numbers to disambiguate between packets and ACKs


Sender
Receiver
transmission
seq# 0
RTT
ACK 1
transmission
seq# 1
RTT
ACK 2


Sender
Receiver
0
1
2
3
4
5
6
7
seq# 0
seq# 1
seq# 2
seq# 3
ACK 1
ACK 2
ACK 3
ACK 4
seq# 4
seq# 5
seq# 6
seq# 7
window size N = 4

Sender

Receiver



Reliable data delivery

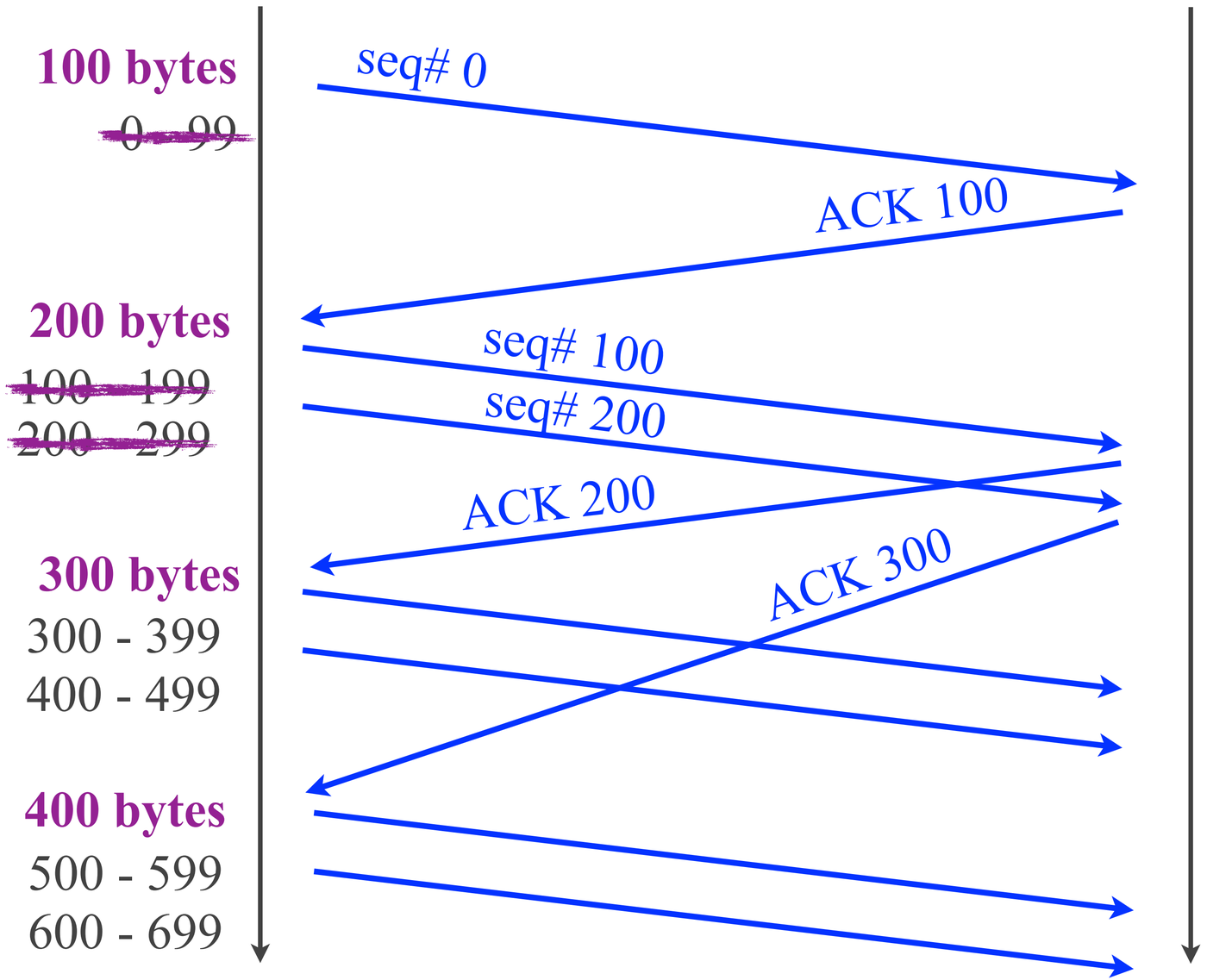
- Pipelining to achieve max throughput
- Window size to control useless transmissions
  - flow control
  - congestion control

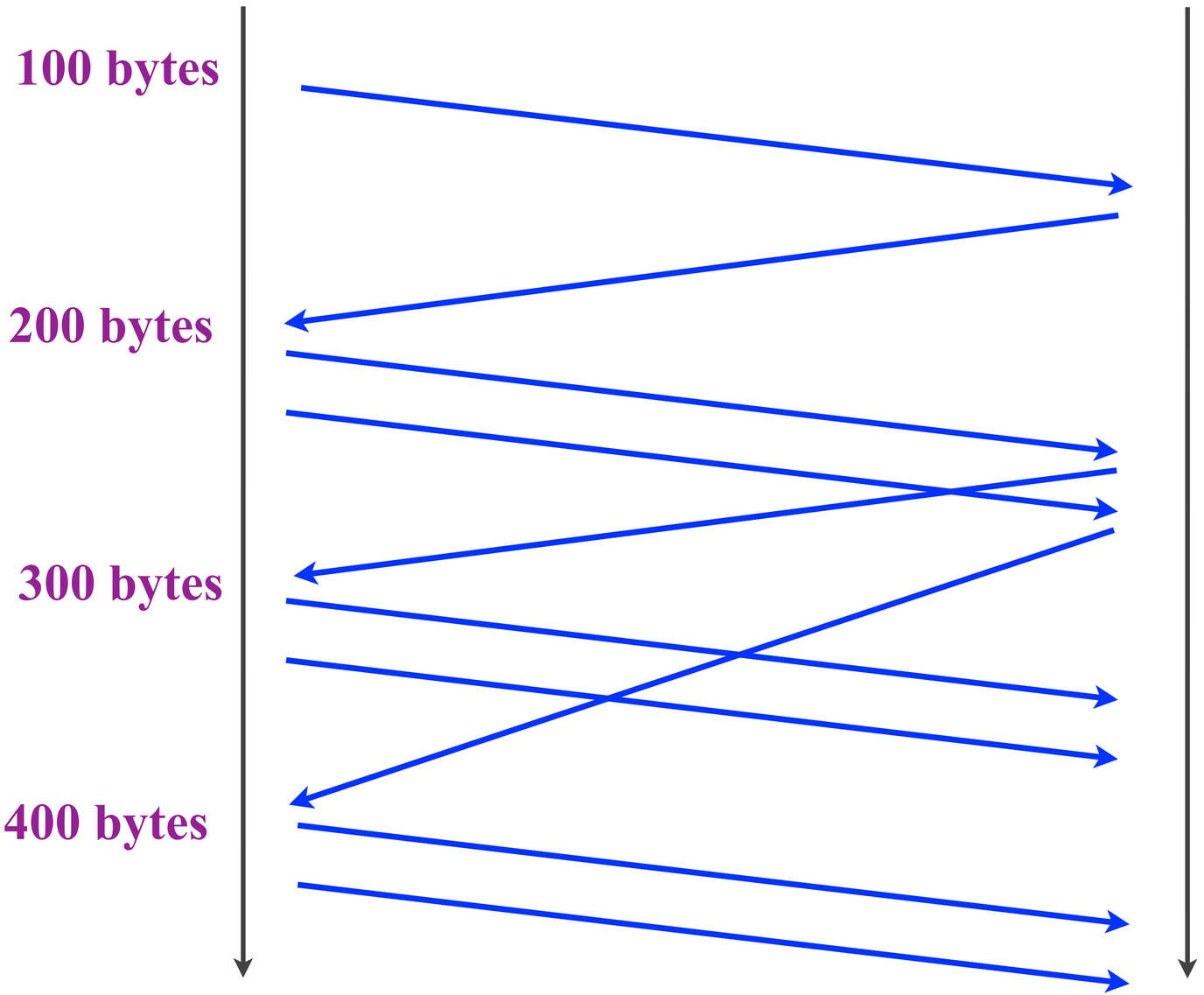
Flow control

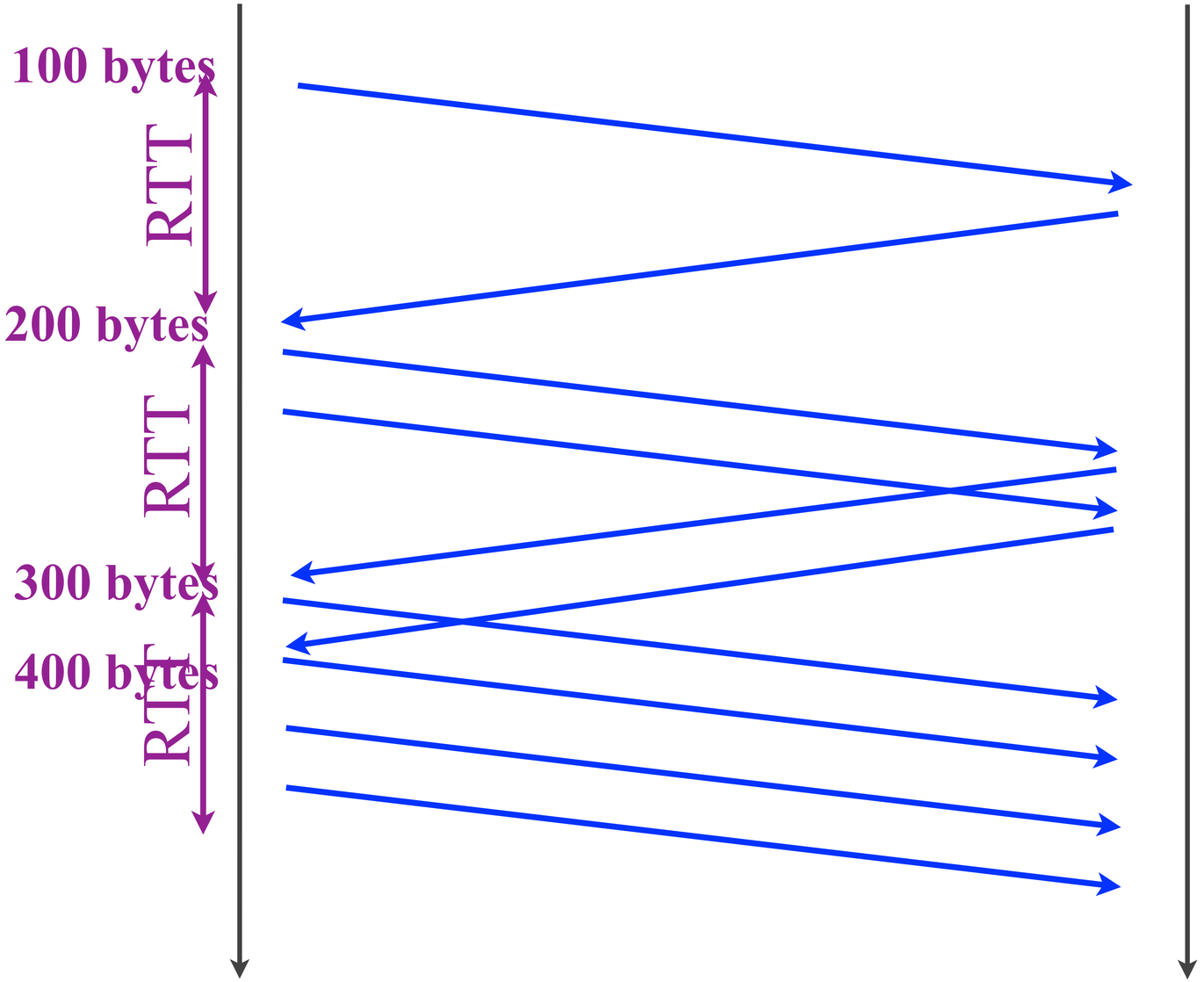
- The receiver communicates to the sender
  the max acceptable window size from its point of view
- The sender uses this to cap its window size

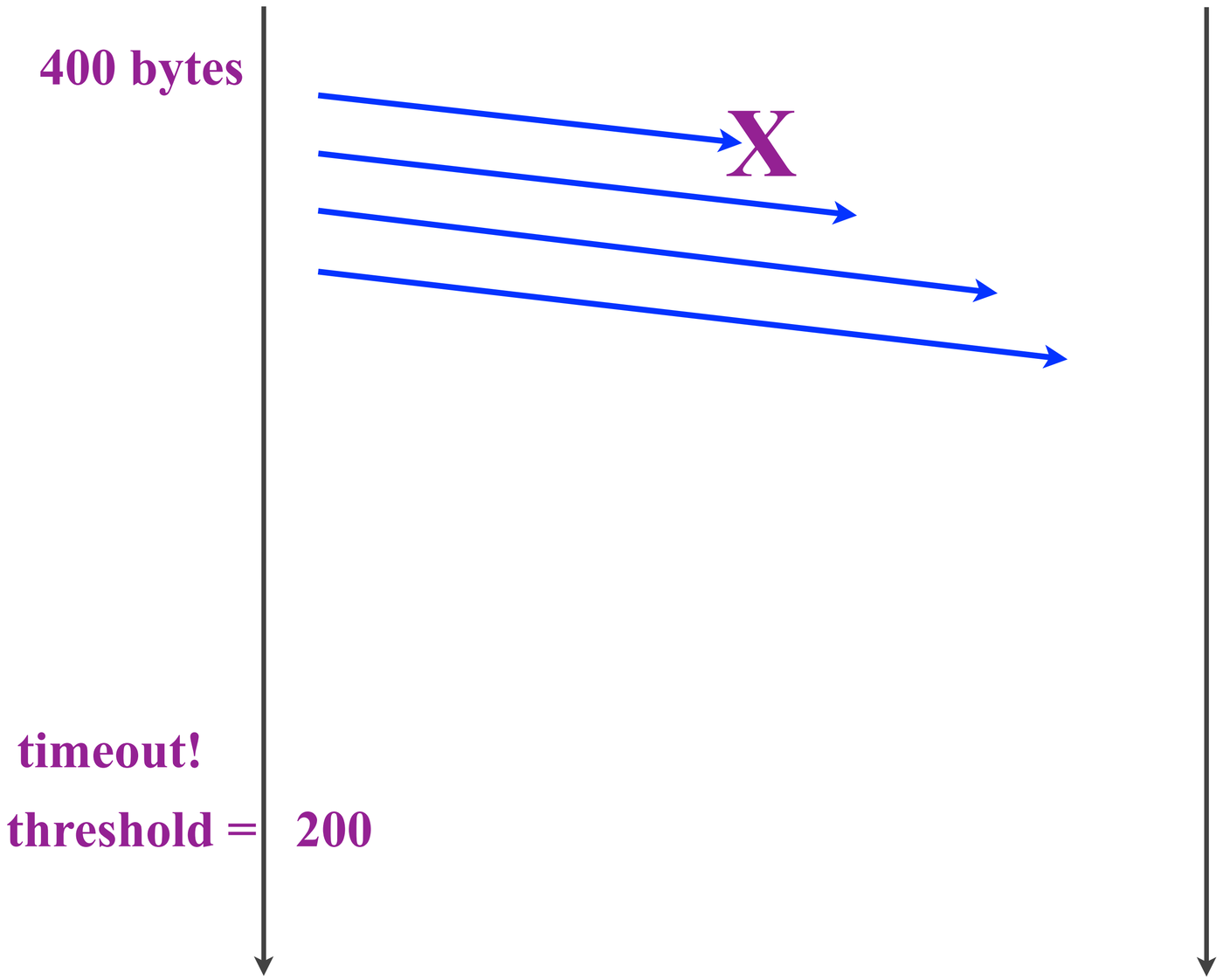
Congestion control

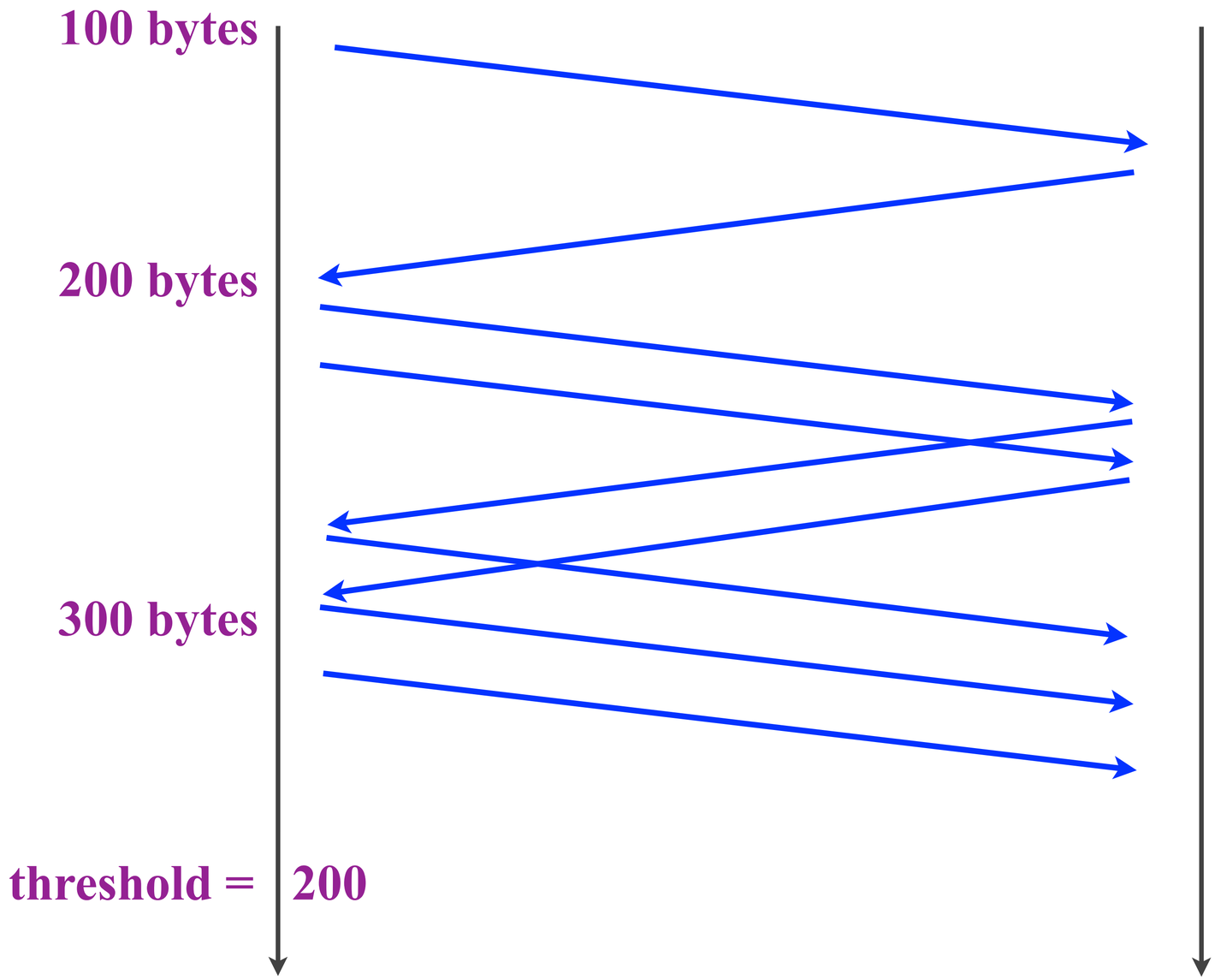
- The sender infers the max window size that will not overload the network through trial and error
- The sender uses this to cap its window size

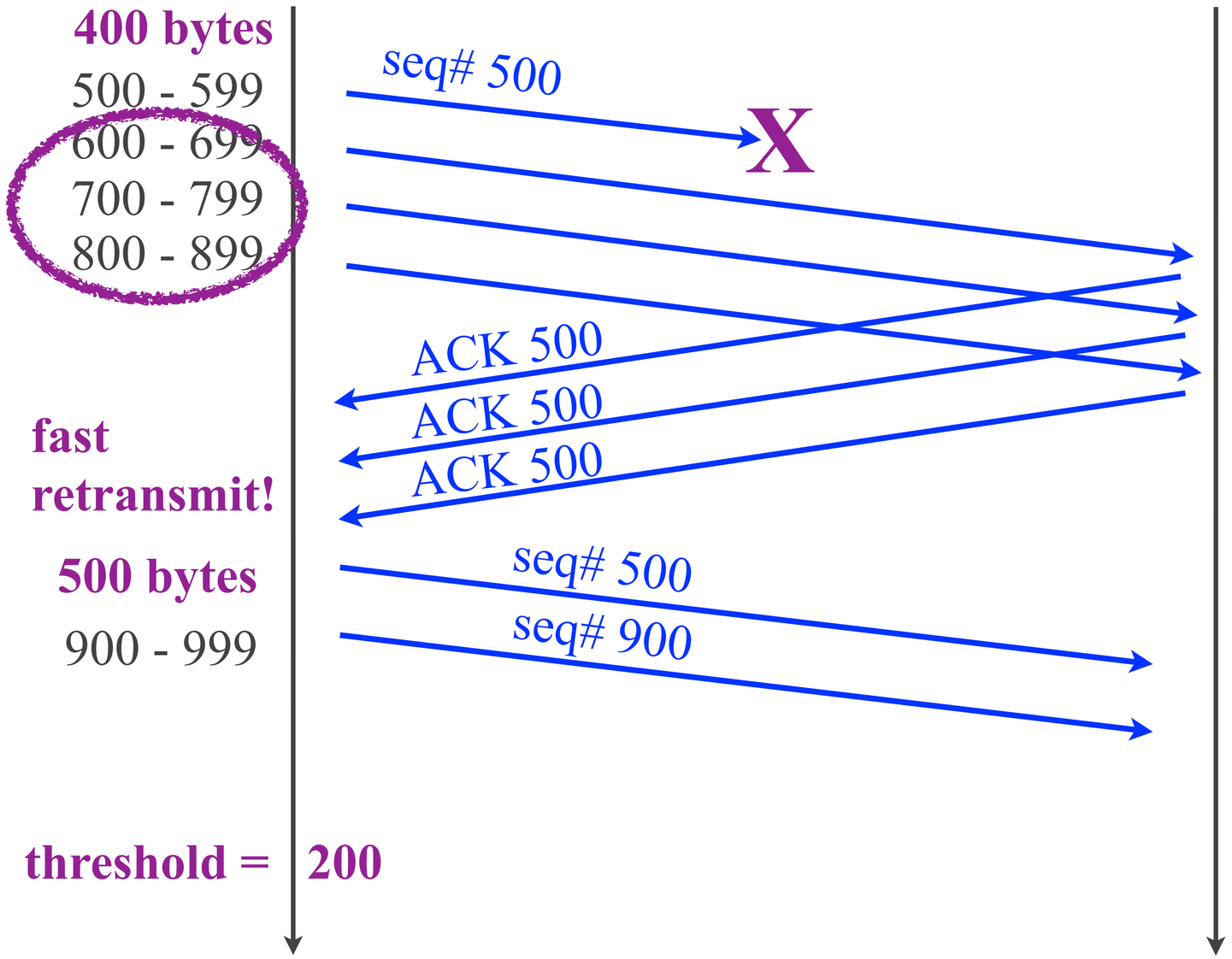

100 bytes
0 - 99
200 bytes
100 - 199
200 - 299
300 bytes
300 - 399
400 - 499
400 bytes
500 - 599
600 - 699
seq# 0
ACK 100
seq# 100
seq# 200
ACK 200
ACK 300


100 bytes
200 bytes
300 bytes
400 bytes

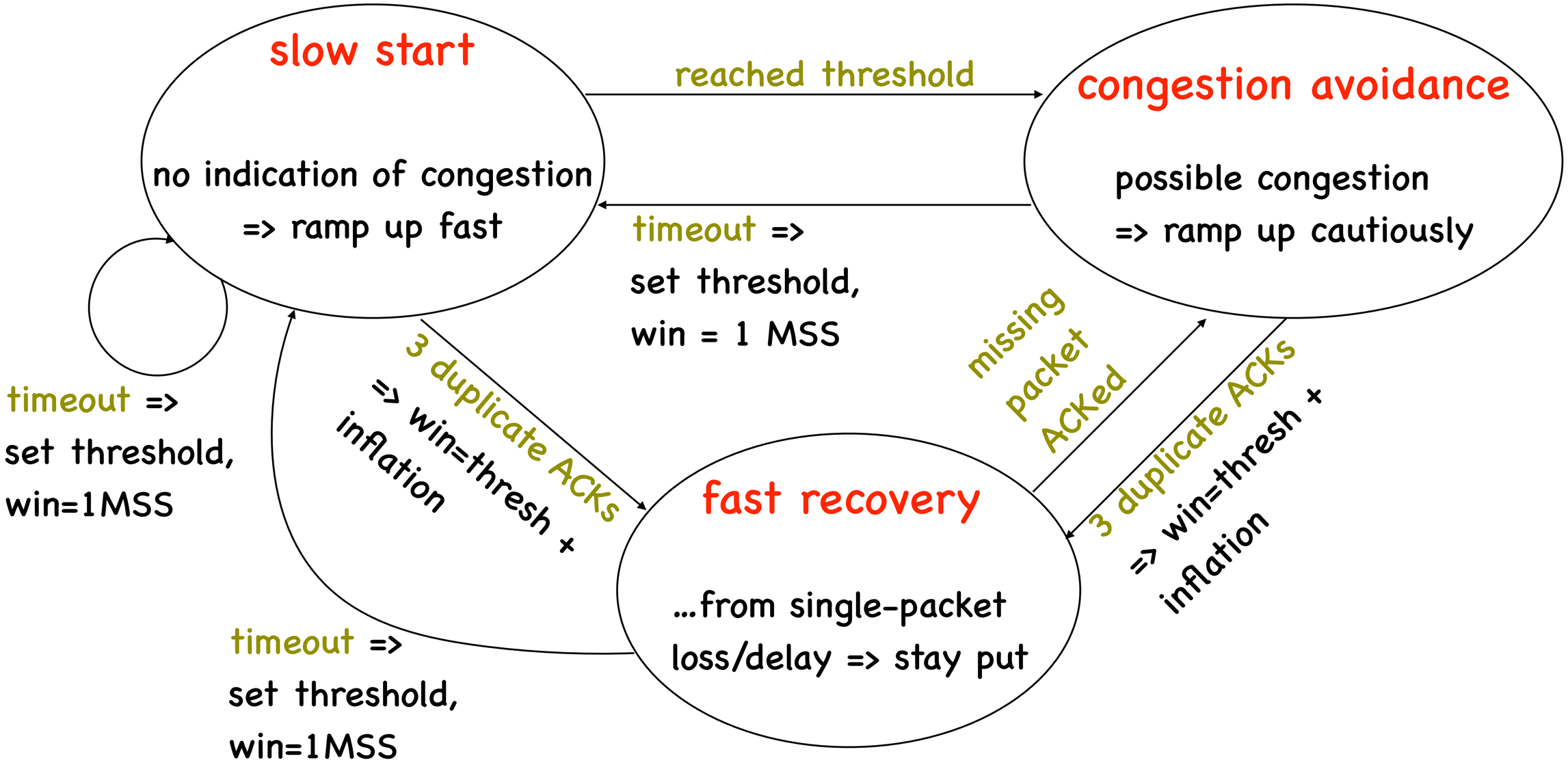

100 bytes
RTT
200 bytes
RTT
300 bytes
400 bytes
RTT


400 bytes
X
timeout!
threshold = 200


100 bytes
200 bytes
300 bytes
threshold = 200


400 bytes
500 - 599
600 - 699
700 - 799
800 - 899
seq# 500
X
ACK 500
ACK 500
ACK 500
fast
retransmit!
500 bytes
900 - 999
seq# 500
seq# 900
threshold = 200

Congestion control algorithm



Topics

- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer

Network-layer functions

- IP forwarding
- IP routing

IP forwarding (basic elements)

- IP address: hierarchical 32-bit name that refers to a network interface
- IP header: carried by every IP packet,
  specifies source and destination IP address
- IP router (= L3 switch): packet switch that forwards IP packets
  based on their destination IP addresses
- IP forwarding table: data structure inside an IP router
  that maps each IP address to an output link

dest. address range			output link
0 - 3	0000 - 0011	00**	1
4 - 7	0100 - 0111	01**	2
8 - 11	1000 - 1011	10**	3
12 - 15	1100 - 1111	11**	4

0100

dest. address range			output link
0 - 2	0000 - 0010	00**	1
3	0011	0011	2
4, 6, 7	0100, 0110, 0111	01**	3
5	0101	0101	2
8 - 15	1000 - 1111	1***	4

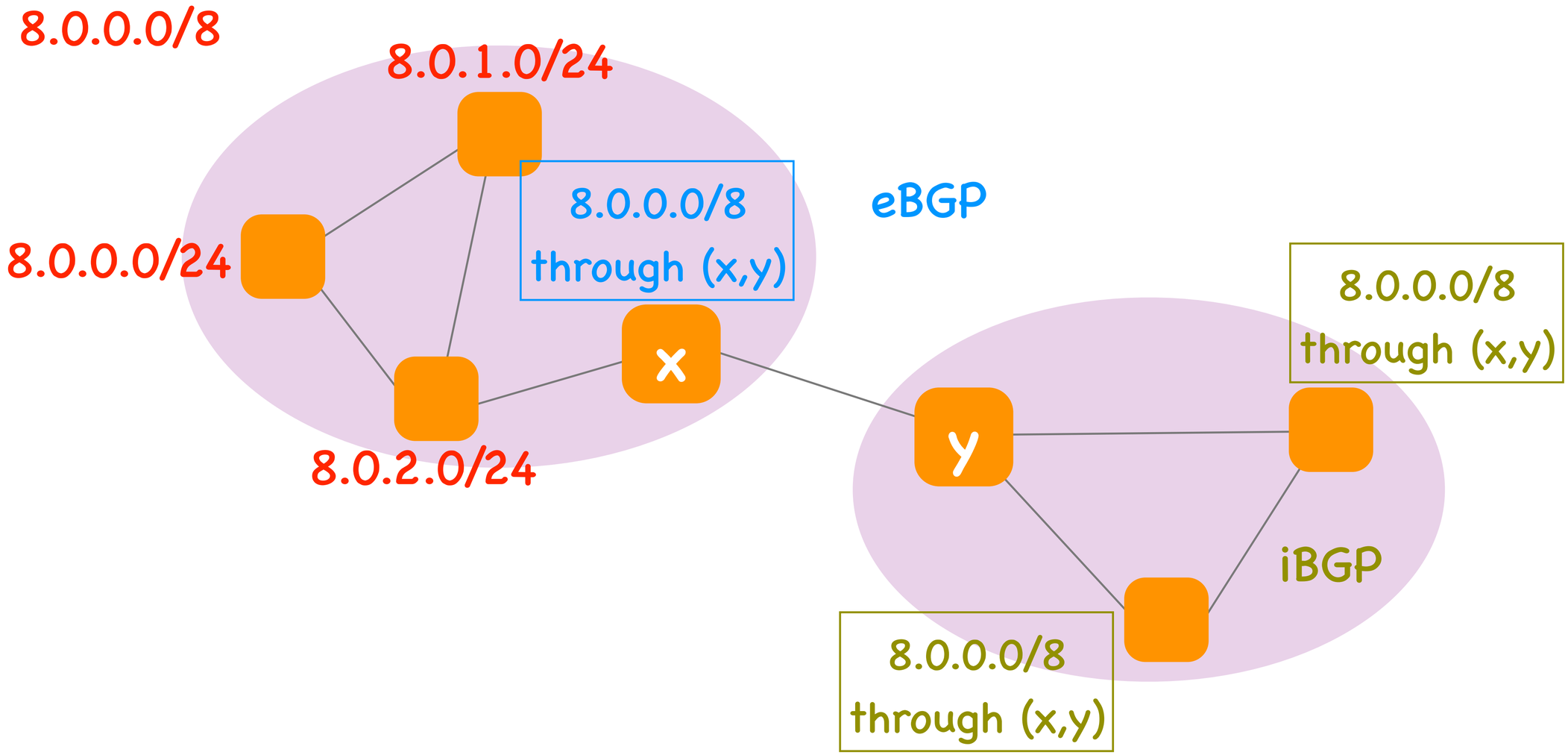
0011

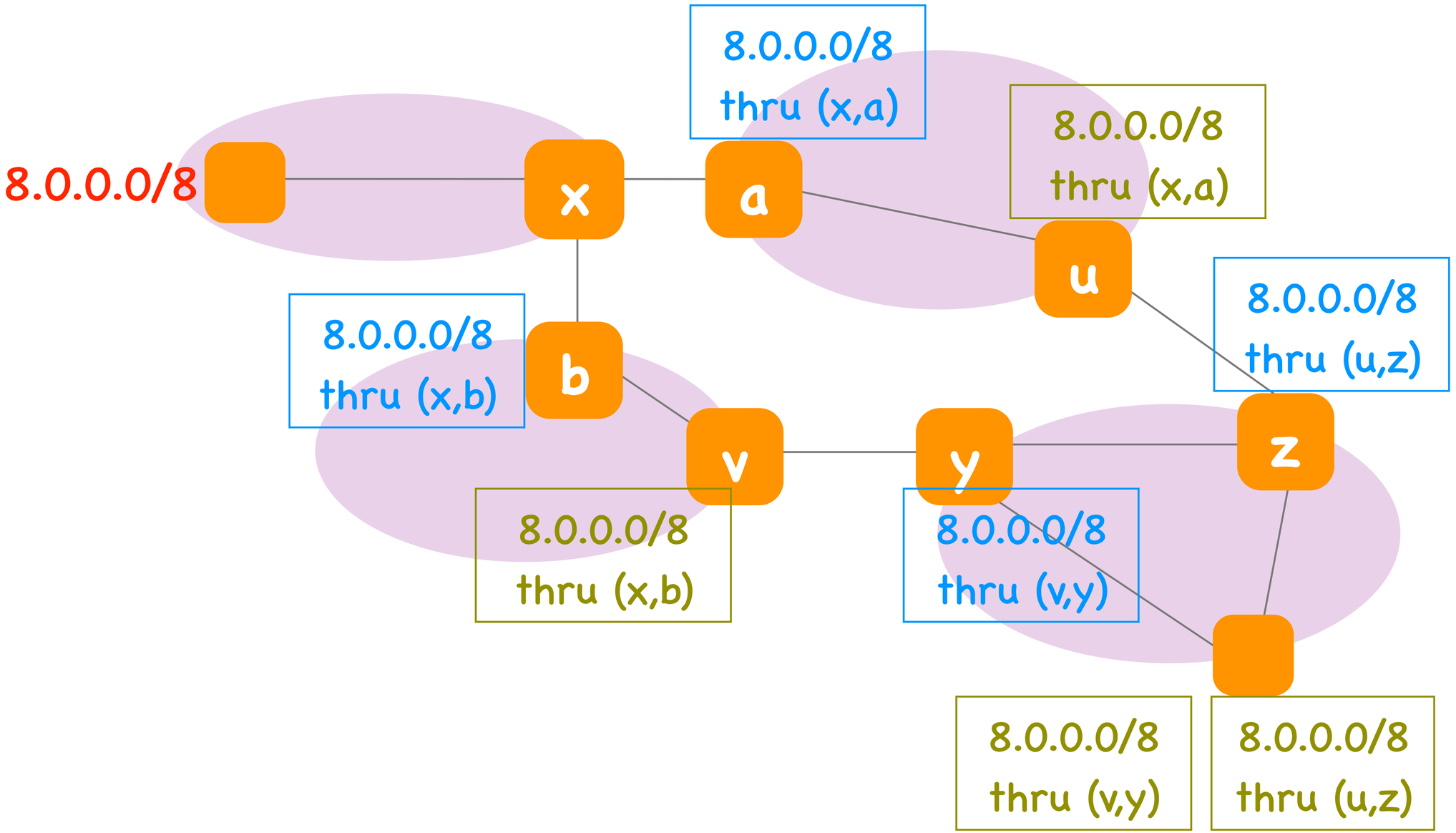
IP forwarding

- IP forwarding maintains state per IP prefix

IP routing

- Routing algorithm: populates the forwarding tables of all the routers
- Least cost path routing: picks the least cost path to each destination
- Link-state routing: each router first learns the entire network graph, then computes the least-cost path from itself to each destination
- Path-vector routing: each router exchanges information with its neighbors and converges to the least-cost path from itself to each destination


8.0.0.0/8
8.0.1.0/24
8.0.0.0/24
8.0.0.0/8
through (x,y)
eBGP
x
8.0.2.0/24
8.0.0.0/8
through (x,y)
y
iBGP
8.0.0.0/8
through (x,y)

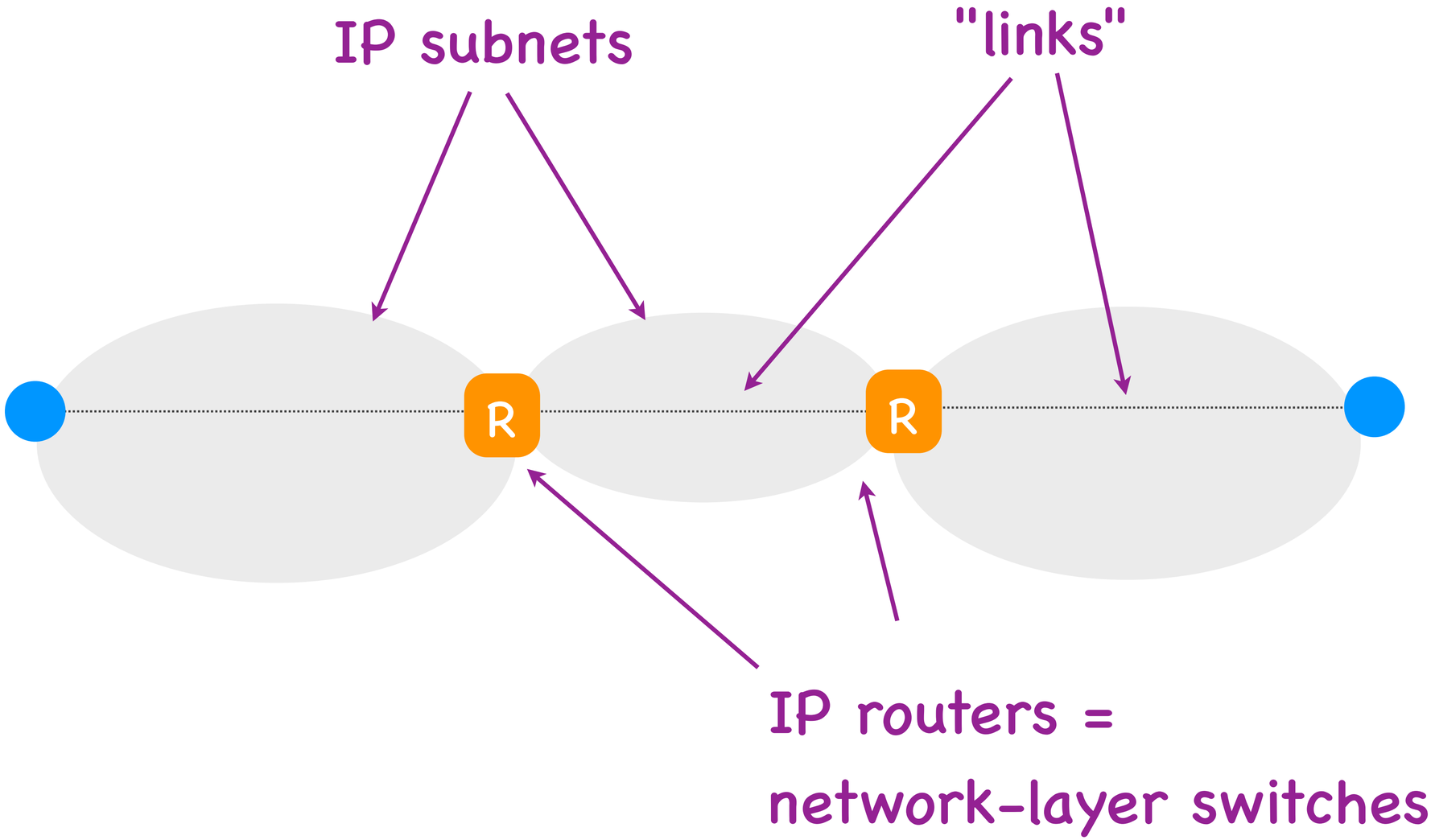

8.0.0.0/8
x
a
8.0.0.0/8
thru (x,a)
8.0.0.0/8
thru (x,a)
u
8.0.0.0/8
thru (u,z)
z
8.0.0.0/8
thru (x,b)
b
v
y
8.0.0.0/8
thru (x,b)
8.0.0.0/8
thru (v,y)
8.0.0.0/8
thru (v,y)
8.0.0.0/8
thru (u,z)

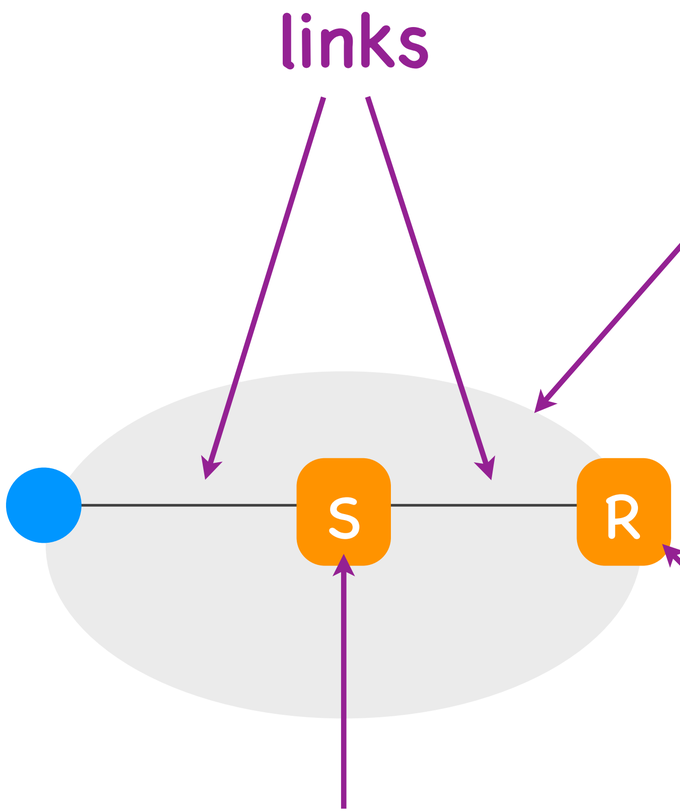
IP routing

- Autonomous System (AS): contiguous network under the same admin
- Intra-AS routing: algorithm run by all the routers of the same AS;
  each router learns a path to every local IP prefix (every other local router)
- Inter-AS routing: algorithm run by all the routers of all ASes;
  each router learns a path to every foreign IP prefix

Topics

- Internet architecture
- Network performance metrics
- Internet layers
- DNS
- TCP
- Network layer
- Link layer


IP subnets
"links"
R
R
IP routers =
network-layer switches



Switch =
link-layer switch

IP router =
network-layer switch

Internet point of view

- Network layer: takes packet from one end of the Internet to the other end
- Link layer: takes packet from one end of one IP subnet to the other end

IP subnet point of view

- Network layer: takes packet from one end of the IP subnet to the other end
- Link layer: takes packet from one end of one physical link to the other end

Link-layer services (IP subnet point of view)

- Error detection
  - *receiver detects and drops corrupted packets*
  - *with checksums*
- Reliable data delivery
  - *sender/receiver detect corruption and loss and try to recover*
  - *with checksums, ACKs, timeouts, retransmissions, & sequence numbers*
  - *only for error-prone links, typically wireless*
- Medium access control (MAC)
  - *sender manages access to shared medium (typically wireless link)*
  - *listens for ongoing transmissions or "collisions"*
  - *backs off and retries later*

Link-layer services (Internet point of view)

- L2 forwarding
- “L2 routing”

L2 forwarding (basic elements)

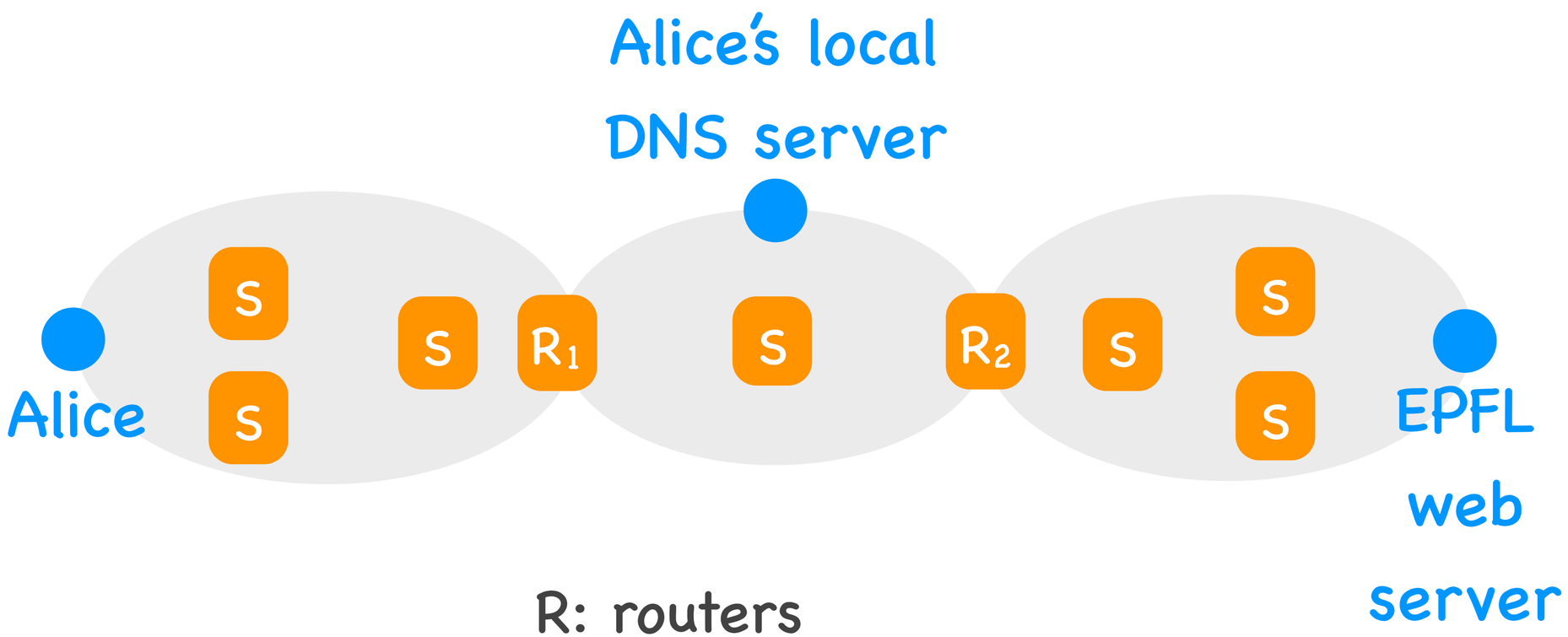
- Ethernet/MAC address: flat 48-bit name that refers to a network interface
- Ethernet header: carried by every Ethernet packet,
  specifies source and destination Ethernet address
- Ethernet switch (= L2 switch): packet switch that forwards Ethernet packets
  based on their destination Ethernet addresses
- MAC forwarding table: data structure inside an Ethernet switch
  that maps each MAC address to an output link

L2 forwarding

- L2 forwarding maintains state per active destination MAC address

“L2 routing”

- Self-learning: each switch learns the output link for each destination by observing incoming traffic
- Broadcasting: until it learns the output link for a given destination, it broadcasts packets for that destination
- Spanning tree: prevents loops when broadcasting



R: routers

S: switches

gray circles: IP subnets

A types http://www.epfl.ch in her browser

At least 4 packets:

- A's DNS request to local DNS server
- local DNS server's response to A
- A's HTTP GET request to web server
- web server's response to A

A types http://www.epfl.ch in her browser

At least 4 packets:

A's DNS request to local DNS server

local DNS server's response to A

A's HTTP GET request to web server

web server's response to A

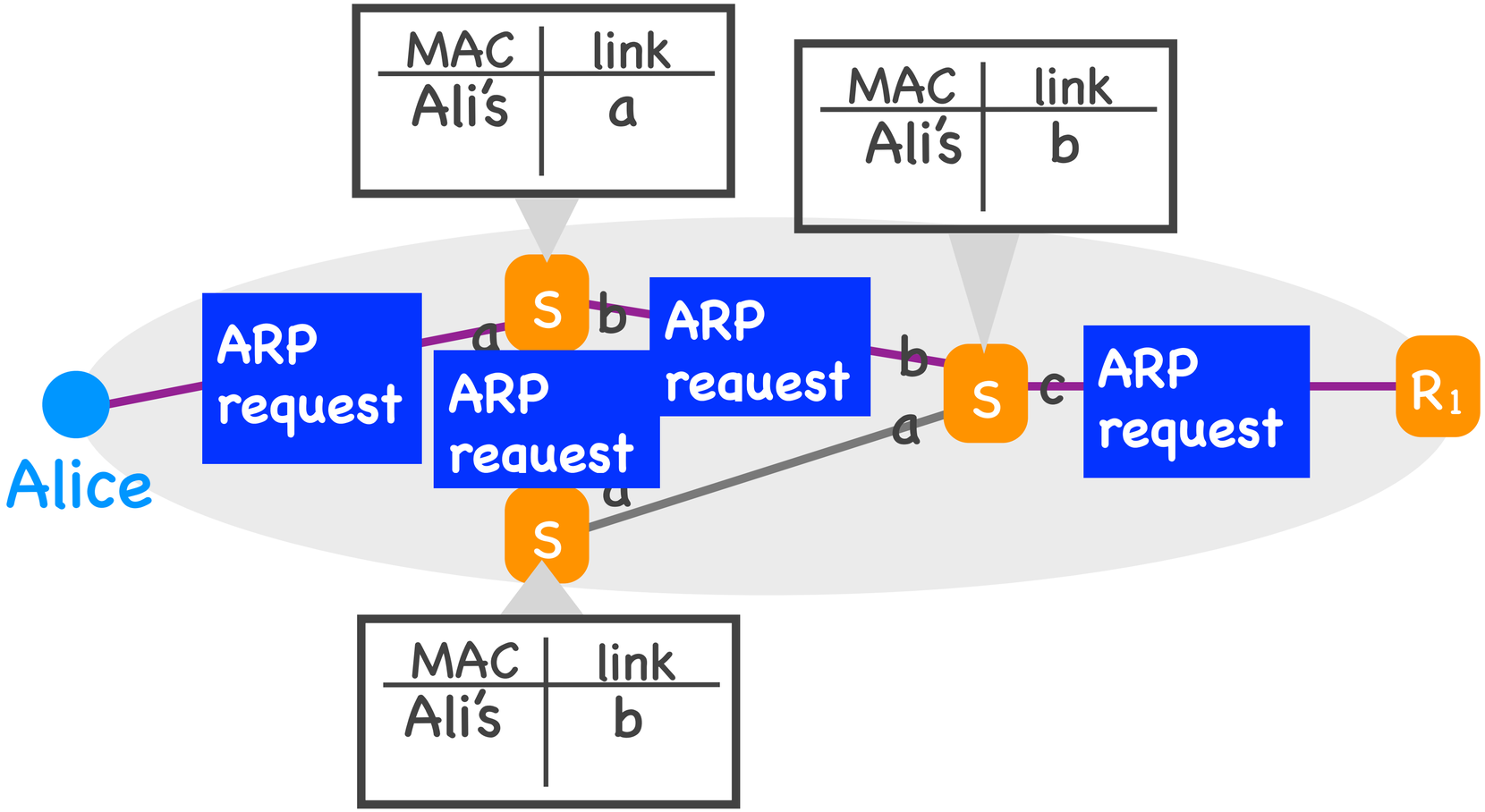
1. A's DNS client process creates DNS request

2. Passed down to transport, network layer

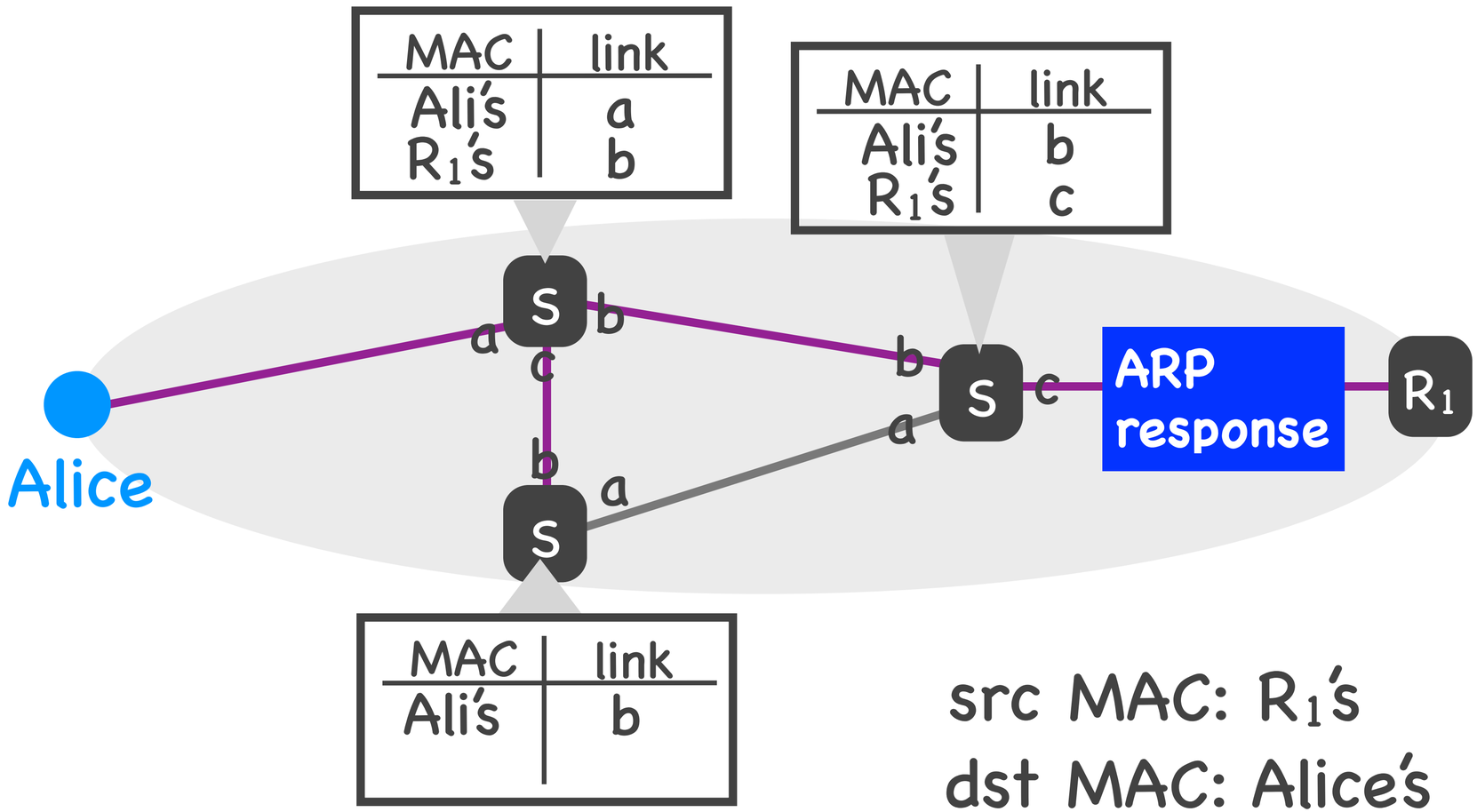
3. A's network layer sends ARP request
   - to resolve DNS server's IP address

src MAC: Alice's

dst MAC: broadcast



4. R_1's network layer sends ARP response

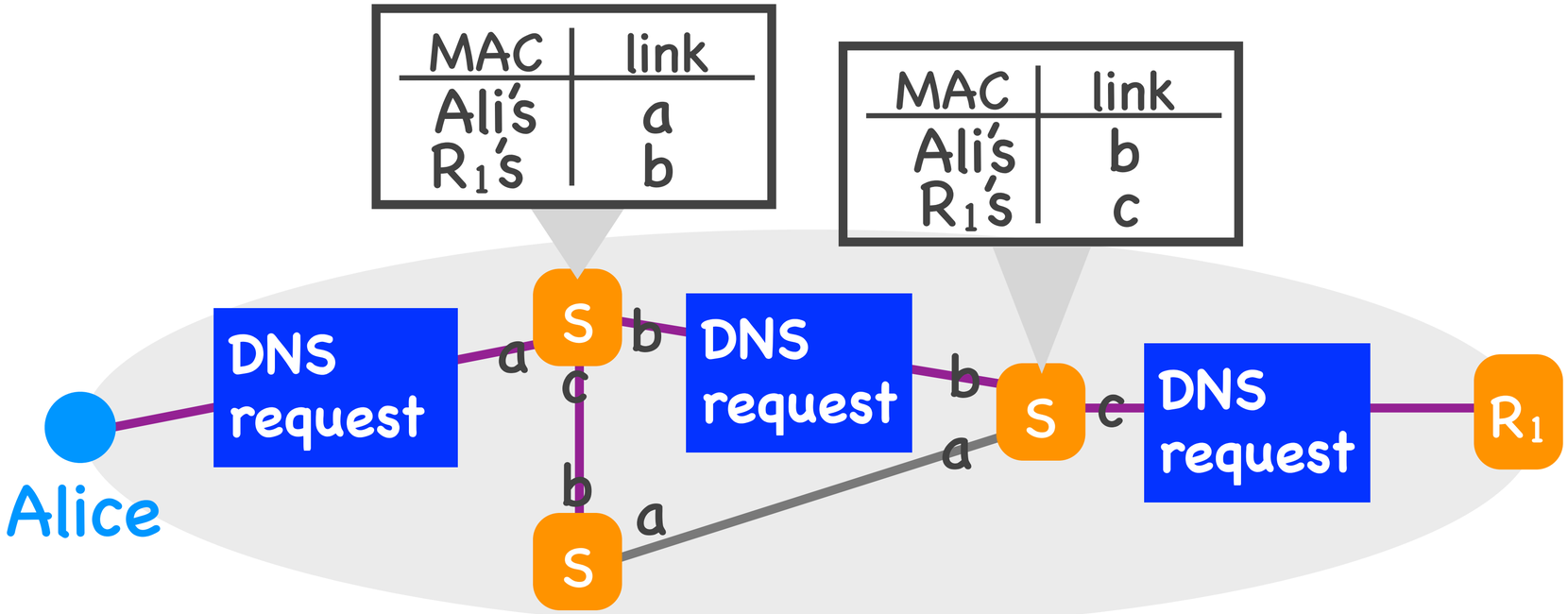

MAC | link
Ali's | a
R1's | b
MAC | link
Ali's | b
R1's | c
S
S
S
a
b
c
b
a
b
c
a
Alice
ARP response
R1
MAC | link
Ali's | b
src MAC: R1's
dst MAC: Alice's

5. A's network layer sends DNS request

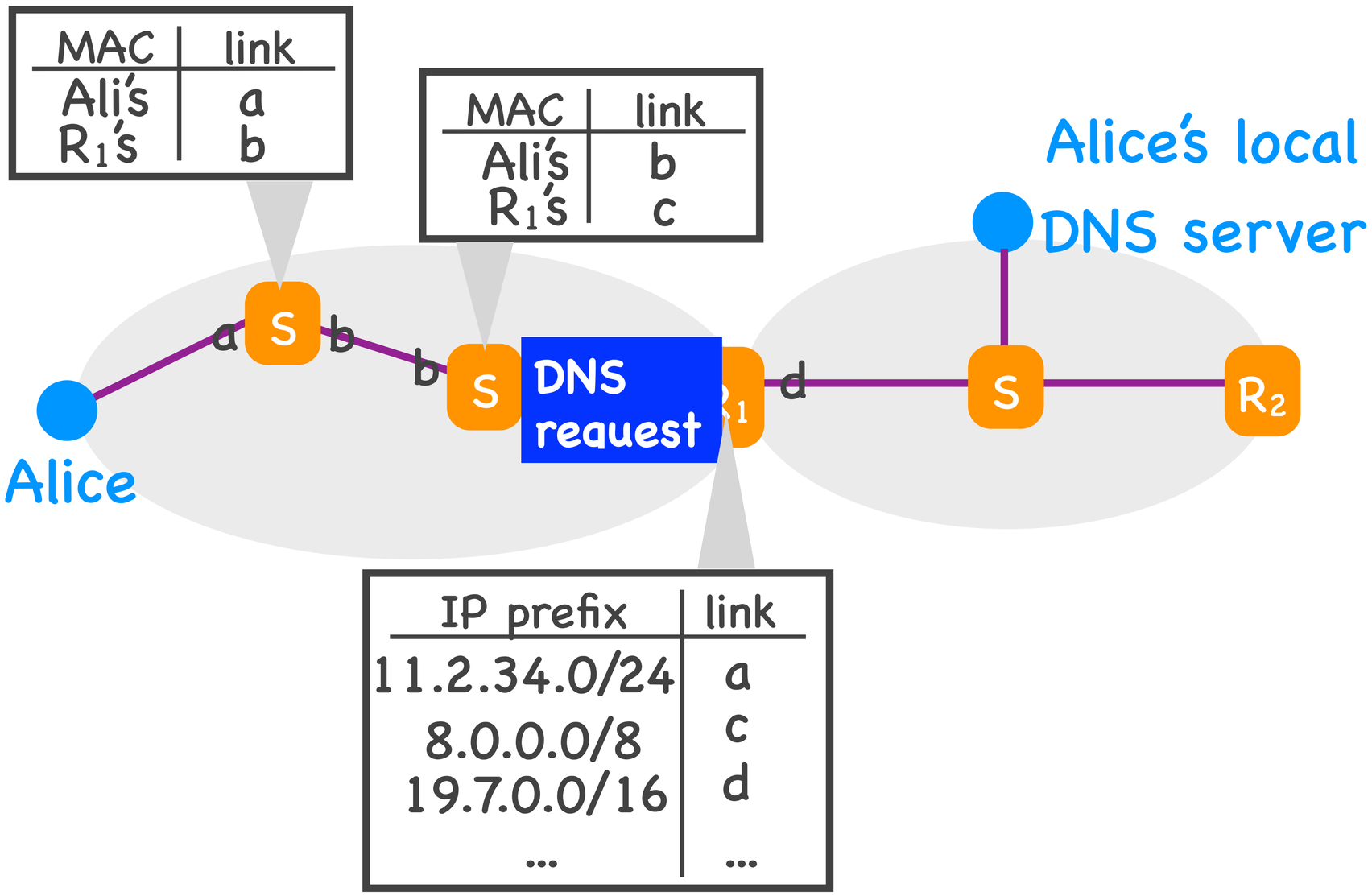
- it now knows the right MAC address to use

src MAC: Alice's src IP: Alice's

dst MAC: R_1's dst IP: DNS server's



6. R_1's network layer performs IP forwarding

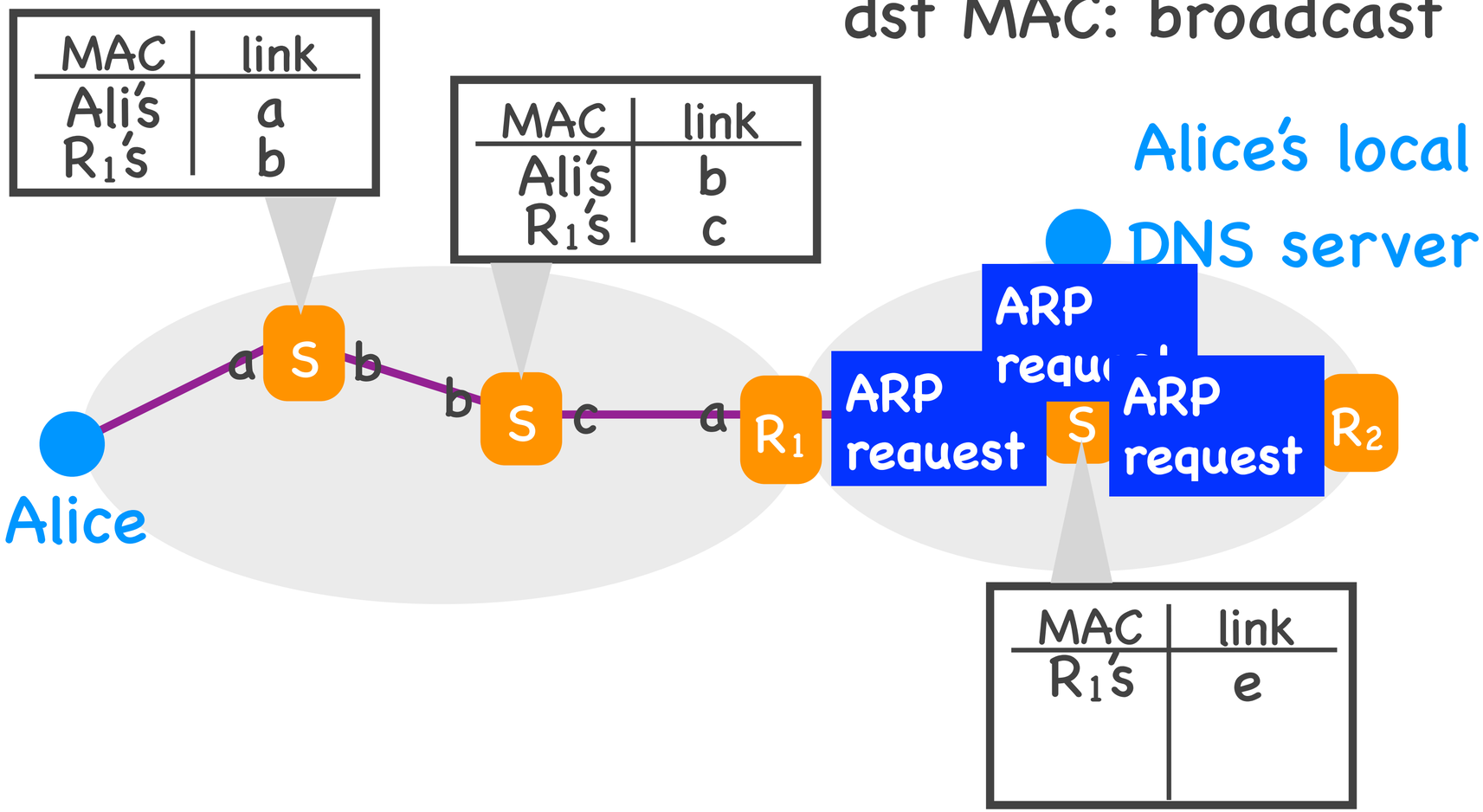

MAC | link
Ali's | a
R1's | b
MAC | link
Ali's | b
R1's | c
Alice's local
DNS server
a
S
b
b
S
DNS
request
R1
d
S
R2
Alice
IP prefix | link
11.2.34.0/24 | a
8.0.0.0/8 | c
19.7.0.0/16 | d
... | ...

7. R_1's network layer sends ARP request

- to resolve DNS server's IP address

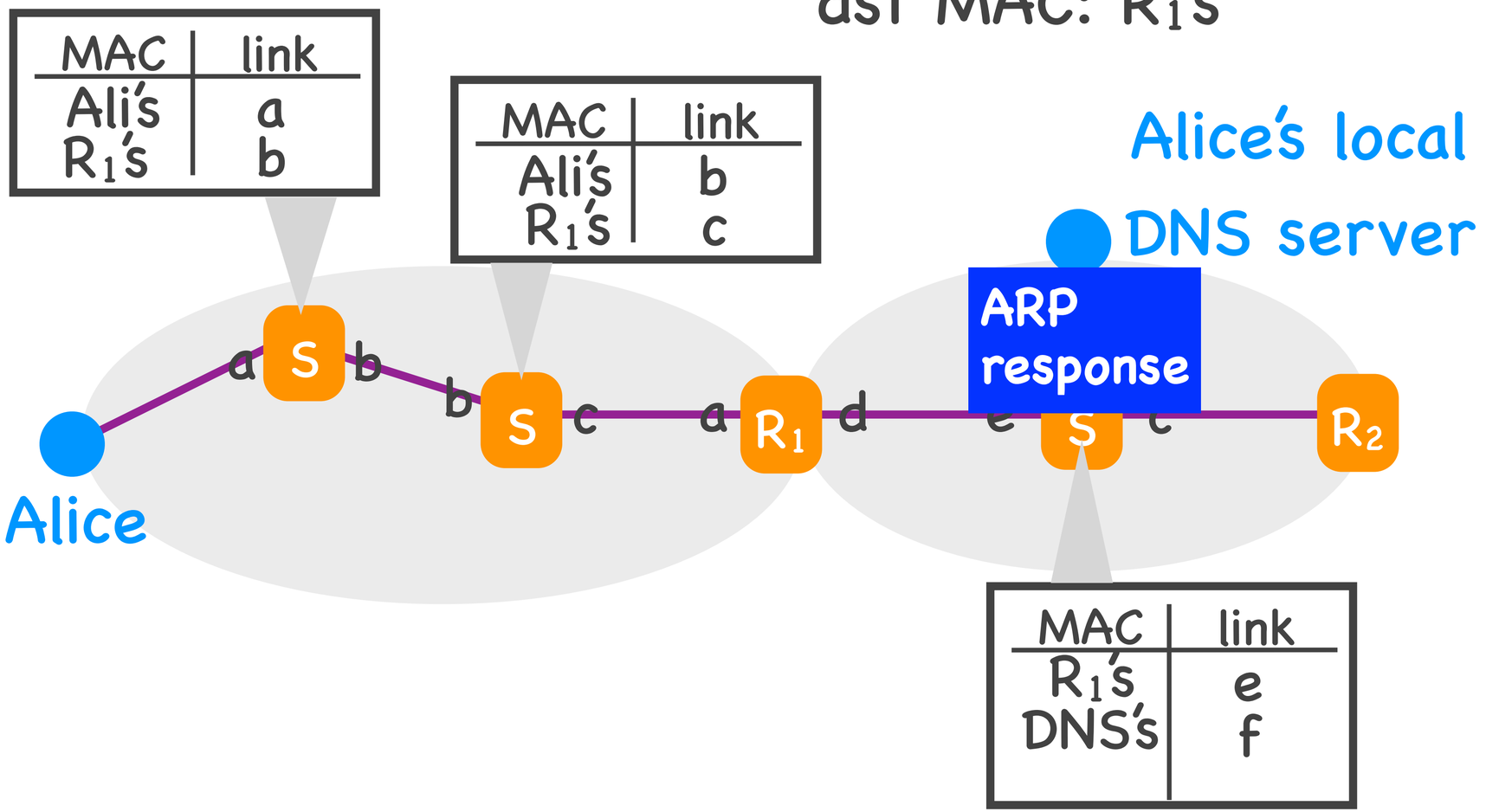
src MAC: R_1's

dst MAC: broadcast



8. DNS server's network layer sends ARP response

src MAC: DNS servers's
dst MAC: R_1's



9. R_1's network layer forwards DNS request

- it now knows the right MAC address to use

src MAC: R_1's
dst MAC: DNS server's

src IP: Alice's
dst IP: DNS server's