Friday (2/26/24)

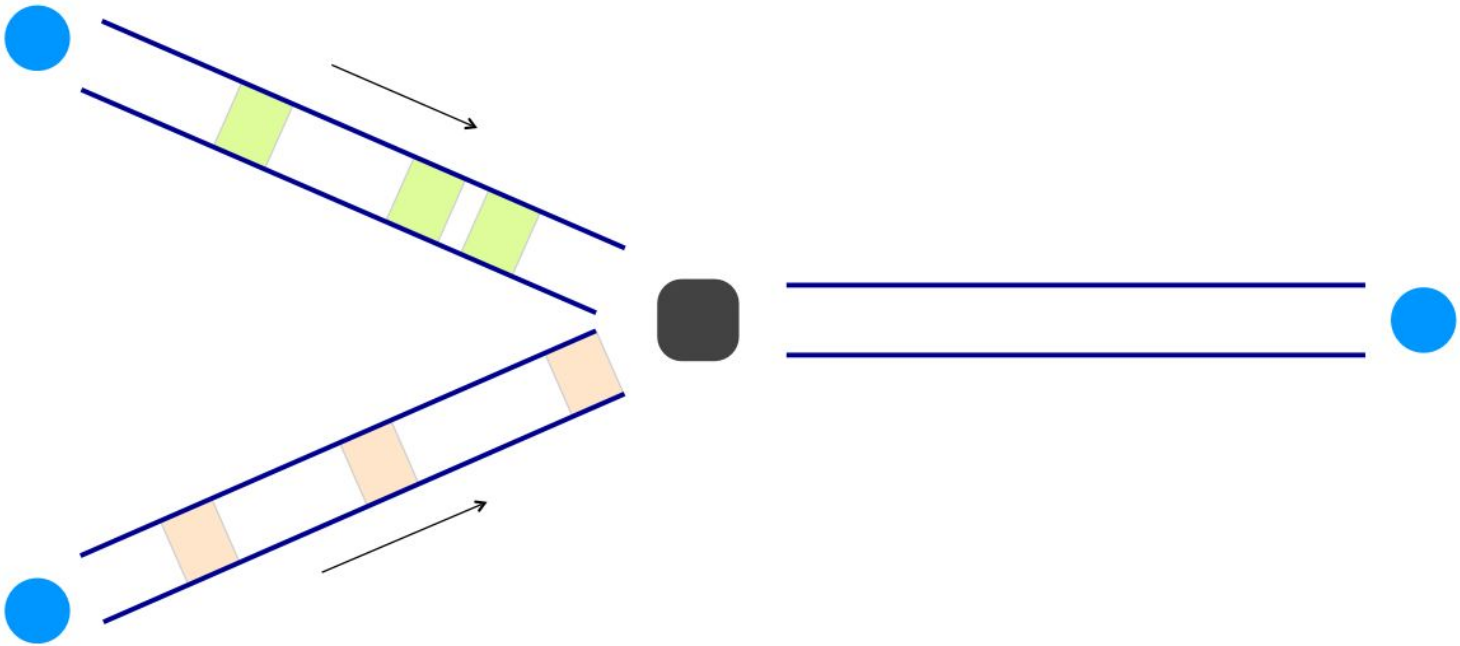
- Weekly assignment will be in-class
- Bring a laptop
- Have netcat installed
  - MacOS - brew install netcat
  - Linux - netcat
  - Windows - ??? nmap maybe include ncat?

Delay, Loss, and Throughput

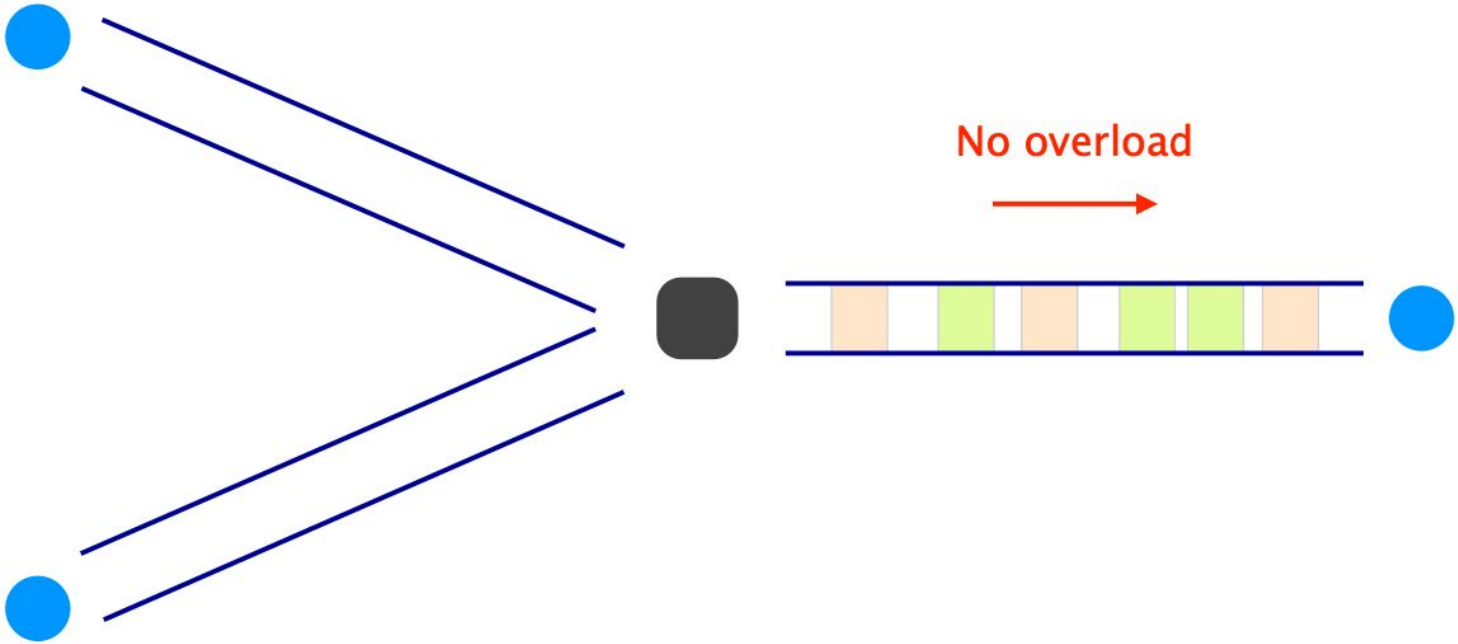
Queueing Delay is the Amount of Time a Packet Waits in a Buffer to be Transmitted

- Hardest to evaluate
  - Varies from packet to packet
- Characterized by statistical measures
  - E.g., average delay, variance, probability of exceeding x

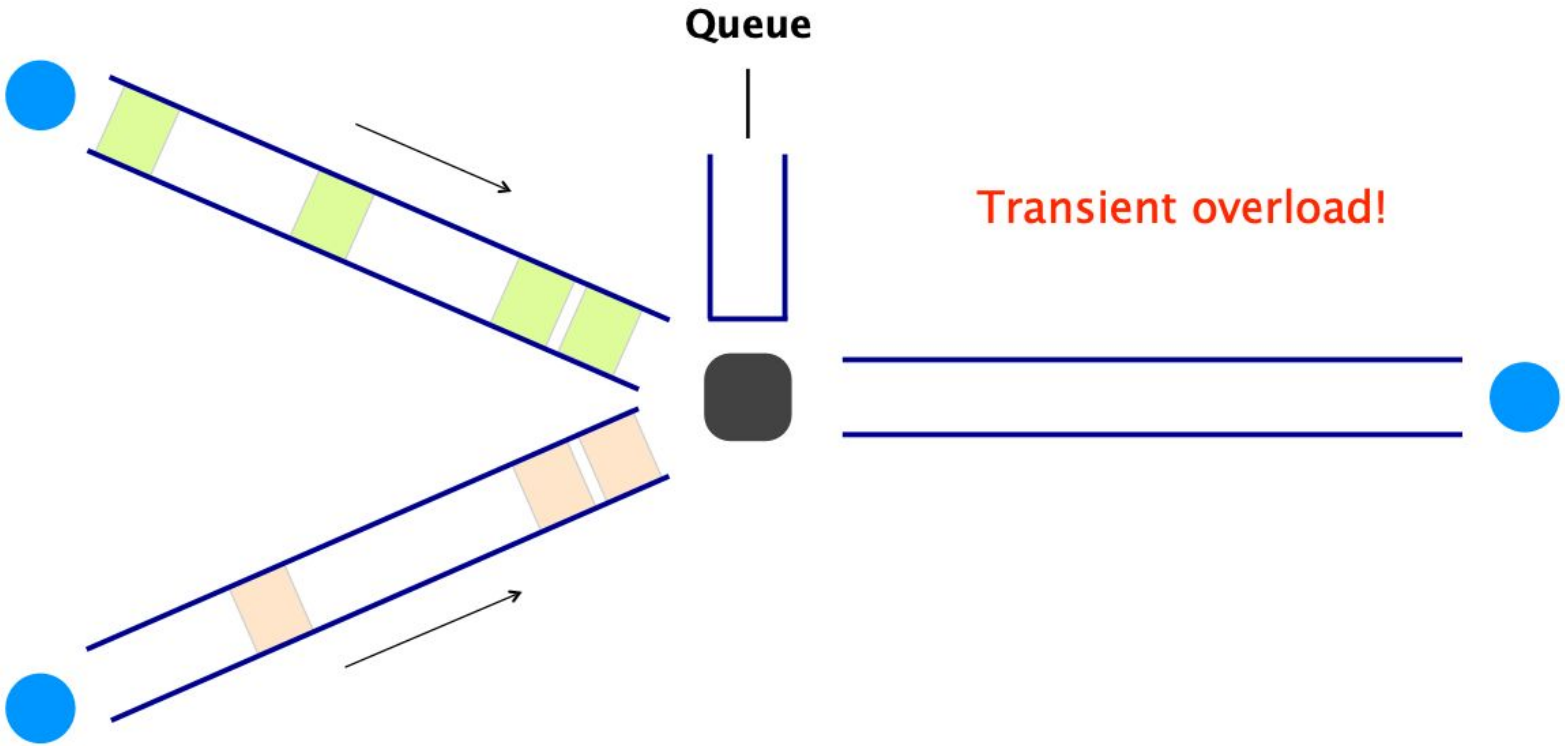
Queueing Delay Depends on the Traffic Pattern



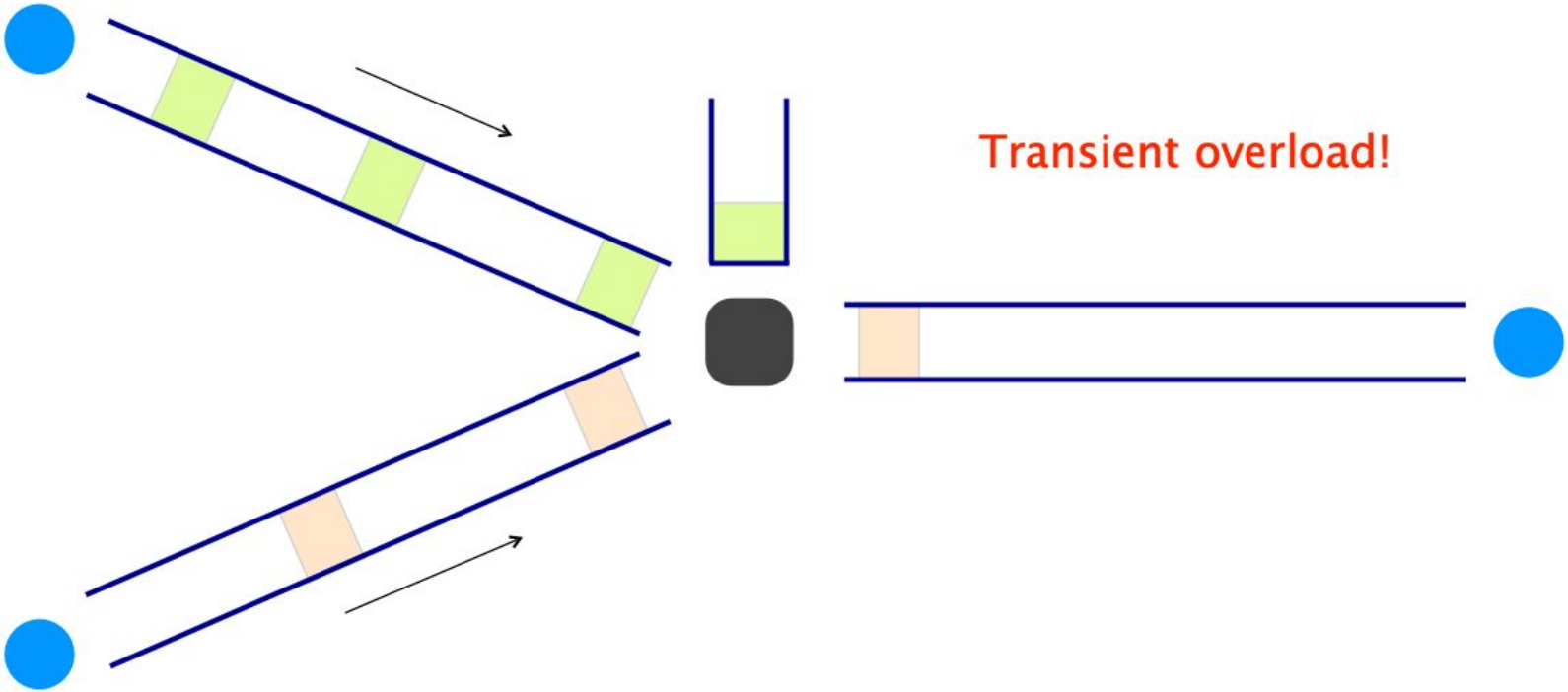
Queueing Delay Depends on the Traffic Pattern



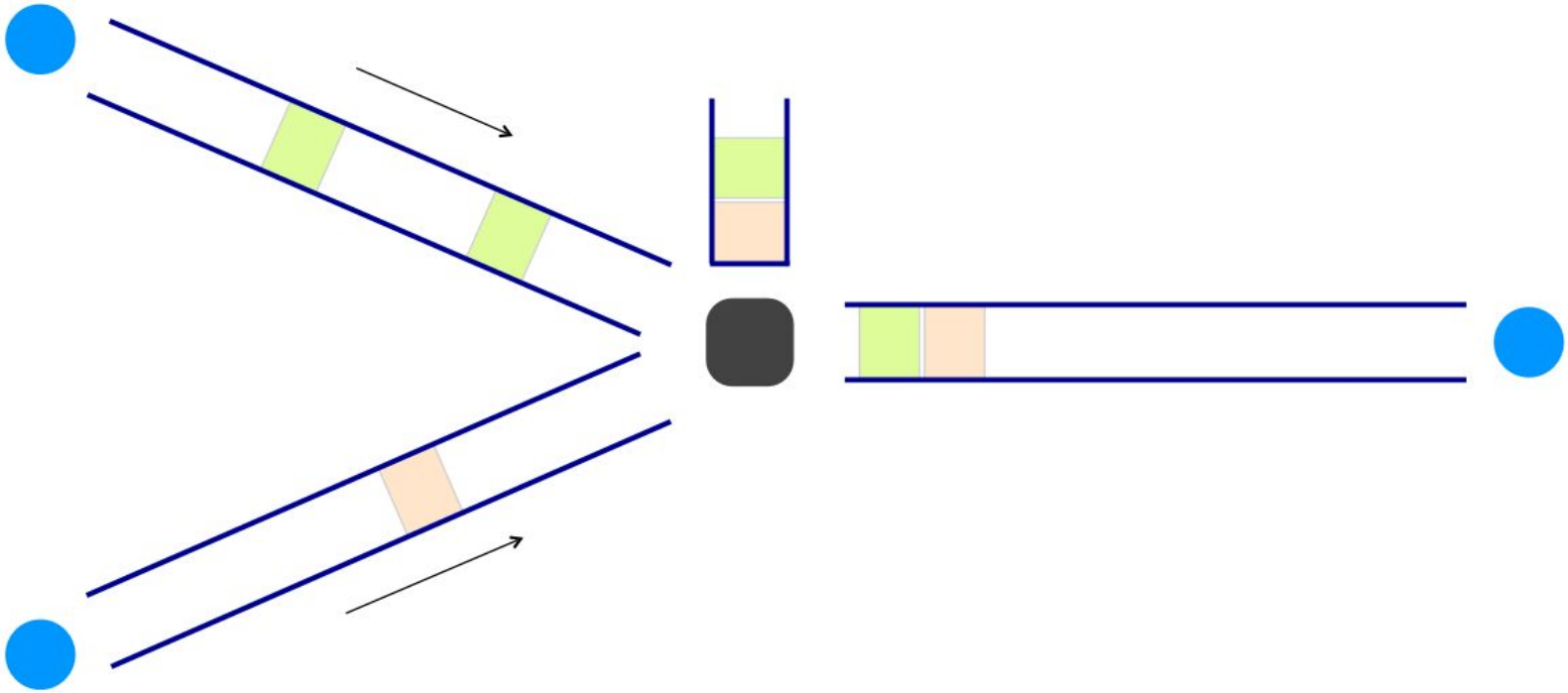
Queueing Delay Depends on the Traffic Pattern



Queueing Delay Depends on the Traffic Pattern



Queueing Delay Depends on the Traffic Pattern



Queueing Delay is Dependent on the Traffic Pattern

- Arrival rate at the queue
- Transmission rate of outgoing link
- Traffic burstiness

average packet arrival rate	a	[packet/sec]
transmission rate of outgoing link	R	[bit/sec]
fixed packets length	L	[bit]
average bits arrival rate	La	[bit/sec]
traffic intensity	La/R	

Queueing Delay is Dependent on the Traffic Pattern

- Arrival rate at the queue
- Tr
- Tr

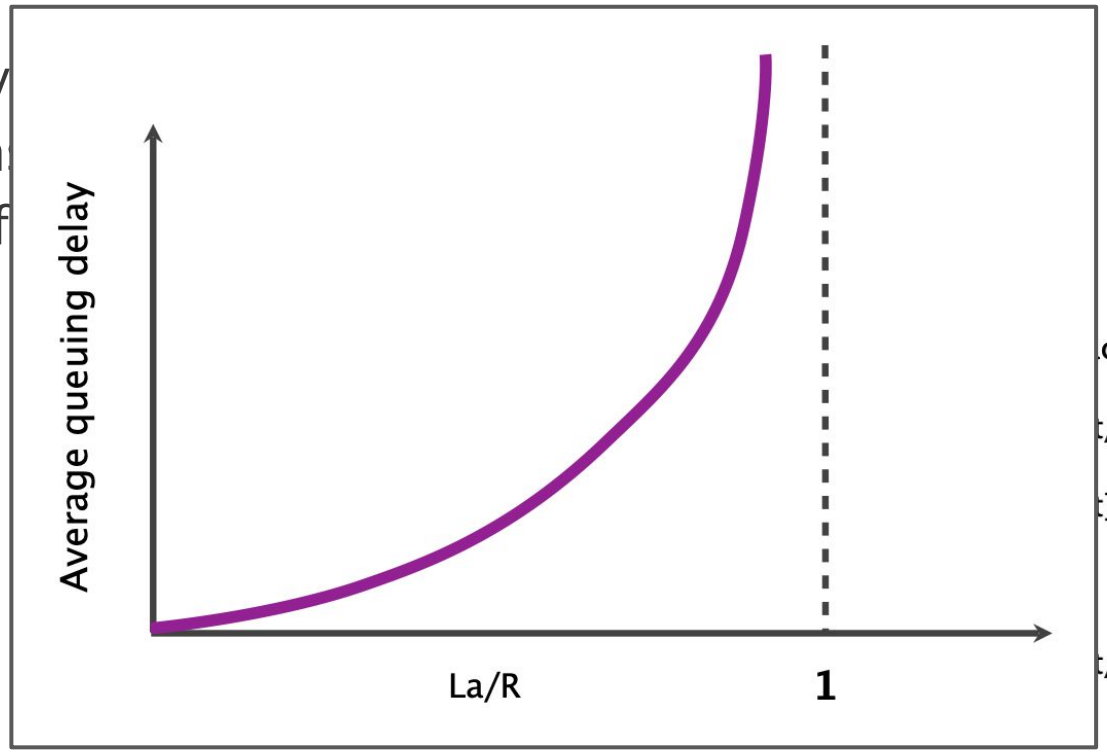
Traffic intensity >1, the queue will increase without bound

Rule: Design queueing system so that you never get near that point

average bits arrival rate	*La*	[bit/sec]
traffic intensity	*La/R*	

Queueing Delay is Dependent on the Traffic Pattern

- Arriv
- Tran
- Traff



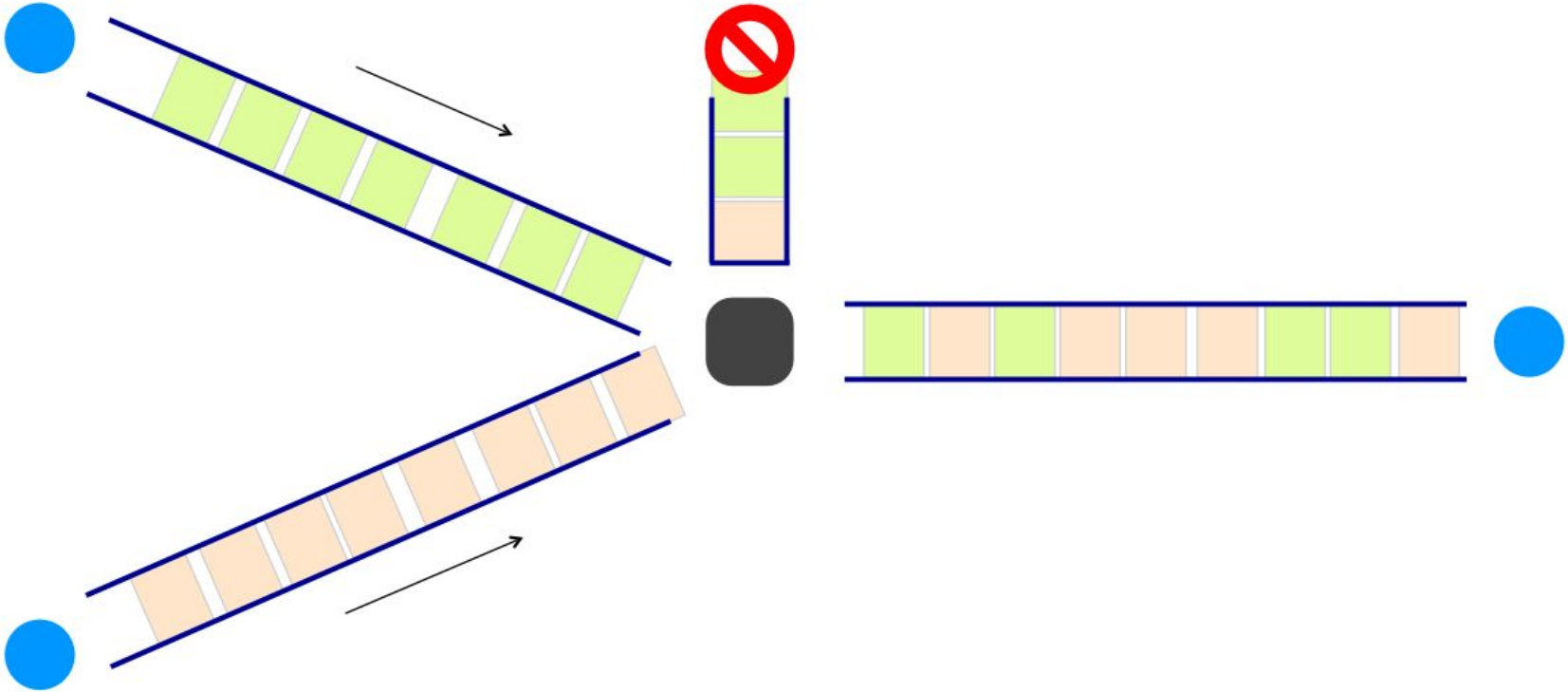
cket/sec]

/sec]

]

/sec]

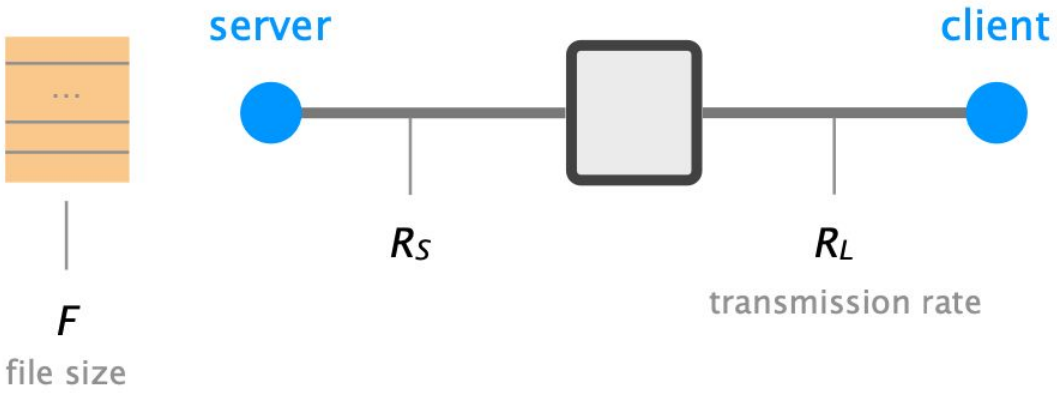
Queues are Finite - Full Queues Lead to Dropped Packets (Loss)



Throughput is the Instantaneous Rate at Which a Host Receives Data

$$\text{Average throughput}\ [\#\text{bits/sec}] = \frac{\text{data size}\ [\#\text{bits}]}{\text{transfer time}\ [\text{sec}]}$$

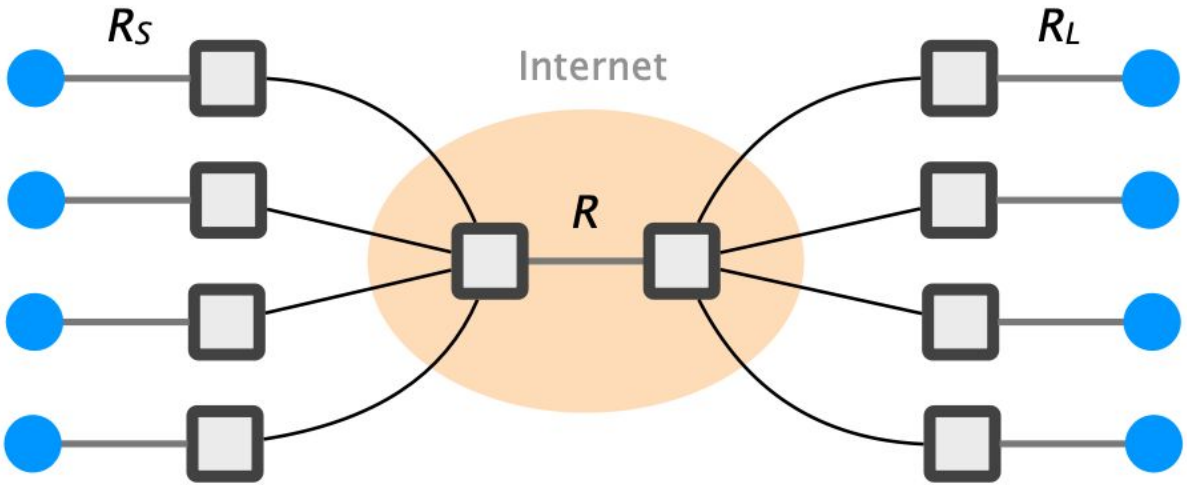
Throughput Depends on the Bottleneck Link



Average throughput $\min(R_S, R_L)$

= transmission rate of the bottleneck link

Bottleneck Links can be Impacted by Intervening Traffic



if 4*min(R_S,R_L) > R

the bottleneck is now in the core,
providing each download R/4 of throughput

Delays

When accessing a website your data has to travel from your computer through different networks to the server on which the website resides and back.

- You want to load a website from Sydney (sydney.edu.au). The round-trip speed of light to Sydney is ~54.5 ms

- Ping `sydney.edu.au` in a terminal

Delays

When accessing a website your data has to travel from your computer through different networks to the server on which the website resides and back.

a) You want to load a website from Sydney (sydney.edu.au). The round-trip speed of light to Sydney is ~54.5 ms

b) Ping sydney.edu.au in a terminal

```
PING sydney.edu.au (20.248.131.216): 56 data bytes
64 bytes from 20.248.131.216: icmp_seq=0 ttl=113 time=198.522 ms
64 bytes from 20.248.131.216: icmp_seq=1 ttl=113 time=197.944 ms
64 bytes from 20.248.131.216: icmp_seq=2 ttl=113 time=202.526 ms
64 bytes from 20.248.131.216: icmp_seq=3 ttl=113 time=198.746 ms
64 bytes from 20.248.131.216: icmp_seq=4 ttl=113 time=202.440 ms
64 bytes from 20.248.131.216: icmp_seq=5 ttl=113 time=201.853 ms
```

Why do we observe such a difference?

Delays

When accessing a website your data has to travel from your computer through different networks to the server on which the website resides and back.

a) You want to load a website from Sydney (sydney.edu.au). The round-trip speed of light to Sydney is ~54.5 ms

b) Ping sydney.edu.au in a terminal

```
PING sydney.edu.au (20.248.131.216): 56 data bytes
64 bytes from 20.248.131.216: icmp_seq=0 ttl=113 time=198.522 ms
64 bytes from 20.248.131.216: icmp_seq=1 ttl=113 time=197.944 ms
64 bytes from 20.248.131.216: icmp_seq=2 ttl=113 time=202.526 ms
64 bytes from 20.248.131.216: icmp_seq=3 ttl=113 time=198.746 ms
64 bytes from 20.248.131.216: icmp_seq=4 ttl=113 time=202.440 ms
64 bytes from 20.248.131.216: icmp_seq=5 ttl=113 time=201.853 ms
```

Why do we observe such a difference?

- Solution: The time calculated in the first task only accounts for the propagation delay assuming a straight-line connection. In the following, we list some points which have been neglected:
  - As we have seen in the lecture, there is not only the propagation, but also the transmission, processing and queuing delay.
  - The cables usually don't follow the straight-line between the two locations. Hence, the real distance is longer.
  - The speed of light in fiber cables is reduced by about 30%.

Bandwidth and Delay

Calculate and compare the bandwidth and the delay for different communication methods. (For this exercise, assume that the delay only consists of propagation delay.)

1. Pigeon post: Pigeons can be used as messengers. They are trained to transport messages from one location to another. Assuming you want to send a USB flash drive with 16 GB from Los Angeles to a friend in San Francisco (~600 km) (only in one direction). Calculate the bandwidth and the delay for one pigeon carrying the USB drive and traveling at an average speed of 80 km/h.

2. Pneumatic tube: These systems were introduced in the late 19th century to transport small, urgent items within buildings or even within cities. The capsules travel at an average speed of 8 meters per second. Assuming you send an external hard drive with 2 TB of storage through a tube from WEB to Holmes (distance 400m), calculate the bandwidth and delay.

3. AWS Truck: Amazon uses a truck to move data to their data center. The truck houses a container which can store 100PB of data. Assuming the truck is transporting data from New York to an AWS data center in San Francisco (distance 4700 km) at an average speed of 100 km per hour, calculate the bandwidth and the delay.

*1GB = 10^9 bytes, 1TB = 10^12 bytes, 1PB = 10^15 bytes

Bandwidth and Delay

Calculate and compare the bandwidth and the delay for different communication methods. (For this exercise, assume that the delay only consists of propagation delay.)

1. Pigeon post: Pigeons can be used as messengers. They are trained to transport messages from one location to another. Assuming you want to send a USB flash drive with 16 GB from Los Angeles to a friend in San Francisco (~600 km) (only in one direction). Calculate the bandwidth and the delay for one pigeon carrying the USB drive and traveling at an average speed of 80 km/h.
   Solution: Delay: d = (600 km / 80kmph) = 7.5 h = 27,000 s
   Bandwidth: bw = (1.28×10^11 bits / 27,000s) ≈ 4.7 Mbps

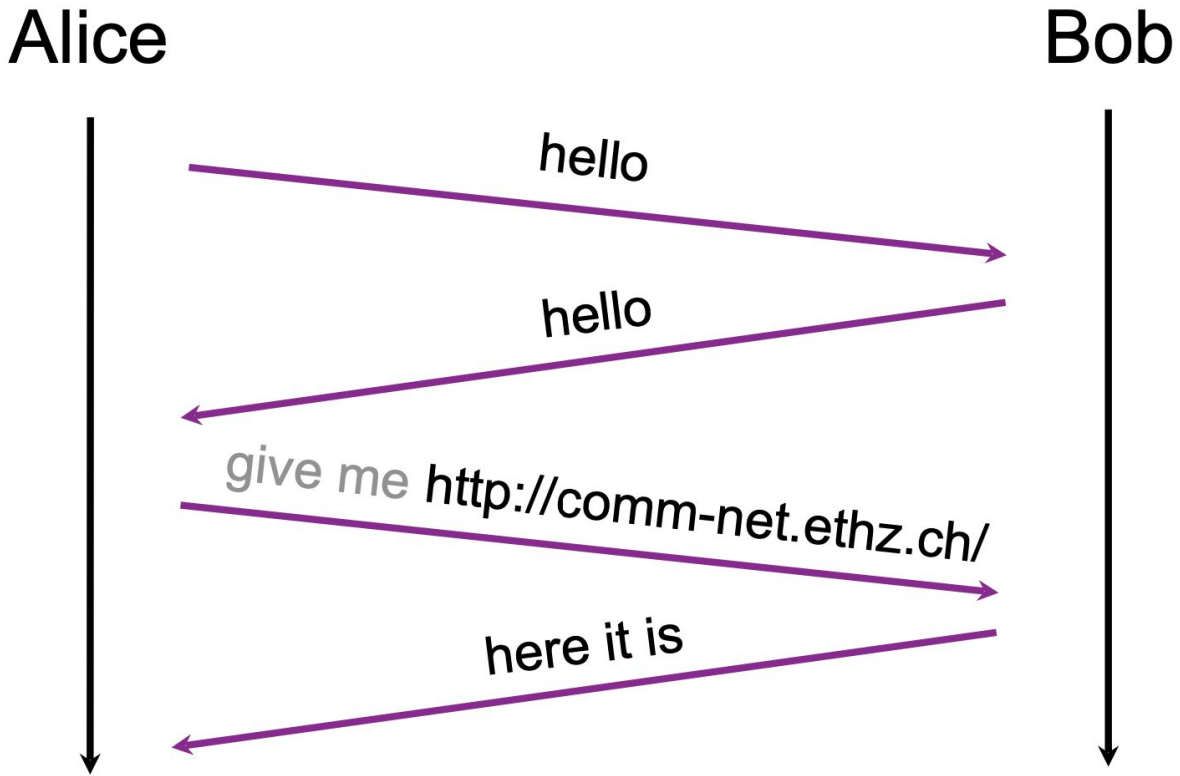
2. Pneumatic tube: These systems were introduced in the late 19th century to transport small, urgent items within buildings or even within cities. The capsules travel at an average speed of 8 meters per second. Assuming you send an external hard drive with 2 TB of storage through a tube from WEB to Holmes (distance 400m), calculate the bandwidth and delay.
   Solution: Delay: d = (400m / 8mps) = 50 s
   Bandwidth: bw = (1.6×10^13 bits / 50s) ≈ 320 Gbps

3. AWS Truck: Amazon uses a truck to move data to their data center. The truck houses a container which can store 100PB of data. Assuming the truck is transporting data from New York to an AWS data center in San Francisco (distance 4700 km) at an average speed of 100 km per hour, calculate the bandwidth and the delay.
   Solution: Delay: d = (4700 km / 100 kmph) = 47 h = 169,200s
   Bandwidth: bw = (8×10^17 bits / 169,200s) ≈ 4.7 Tbps

*1GB = 10^9 bytes, 1TB = 10^12 bytes, 1PB = 10^15 bytes

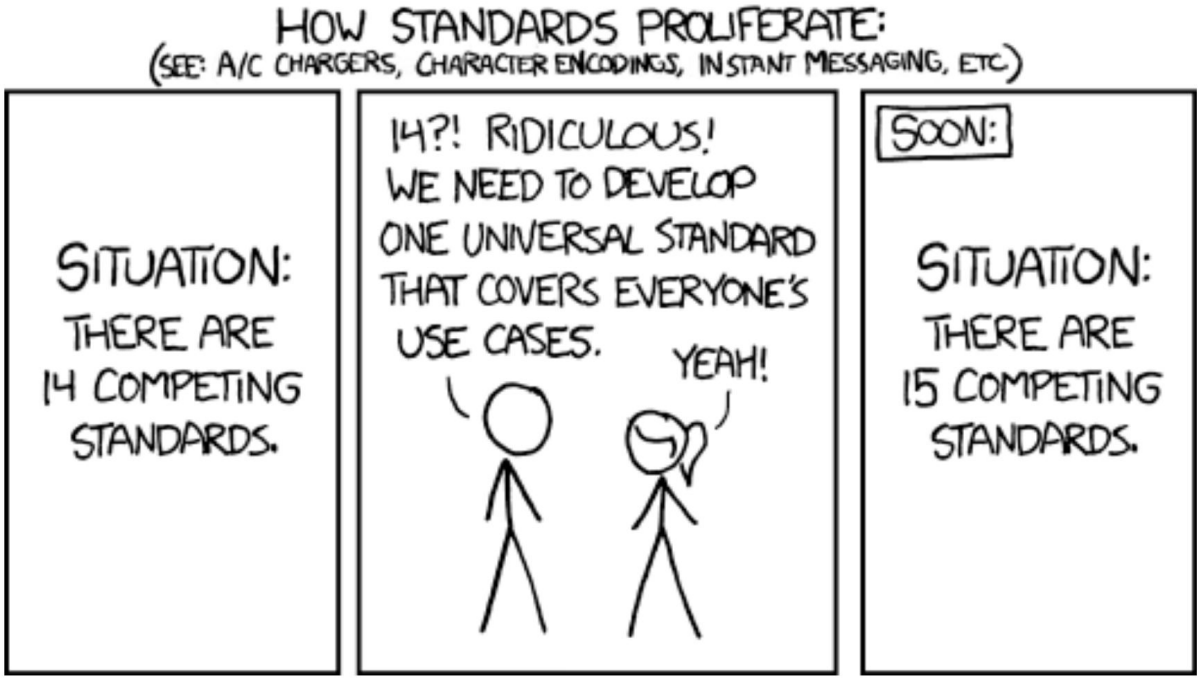
Protocol Layers

Protocols Define How to Communicate



There are a Lot of Protocols. How Do We Organize?

Poorly



https://xkcd.com/927/

Modularity is Key

- can’t build large systems out of spaghetti code
  - hard (if not, impossible) to understand, debug, update
- need to bound the scope of changes
  - evolve the system without rewriting it from scratch
- Modularity is how we do it
  - …and understand the system at a higher-level

Internet communication can be decomposed in 5 independent layers (or 7 layers for the OSI model)

	layer
L5	Application
L4	Transport
L3	Network
L2	Link
L1	Physical

Internet communication can be decomposed in 5 independent layers (or 7 layers for the OSI model)

	layer	service provided:
L5	Application	network access
L4	Transport	end-to-end delivery (reliable or not)
L3	Network	global best-effort delivery
L2	Link	local best-effort delivery
L1	Physical	physical transfer of bits

Each layer provides a service to the layer above by using the services of the layer directly below it

Applications

…built on…

Reliable (or unreliable) transport

…built on…

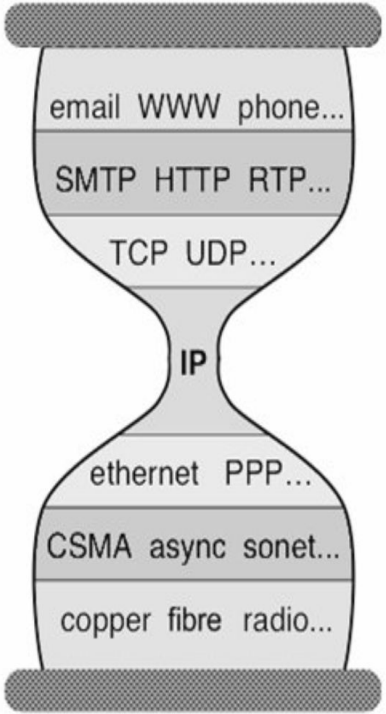
Best-effort global packet delivery

…built on…

Best-effort local packet delivery

…built on…

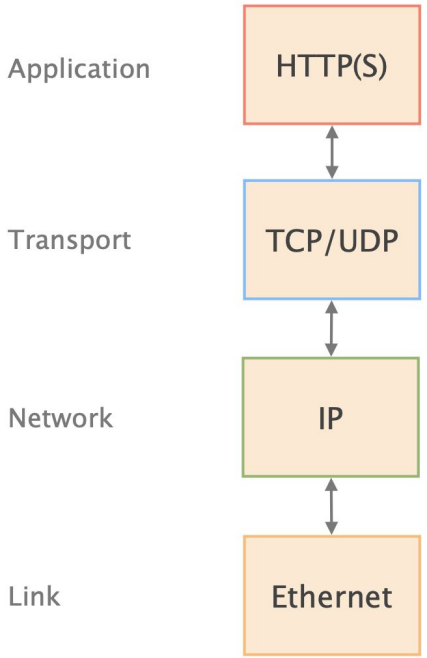
Physical transfer of bits



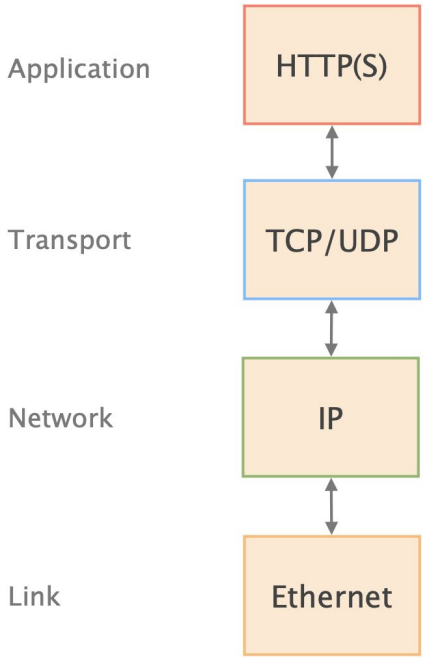
Each layer Has a Unit of Data

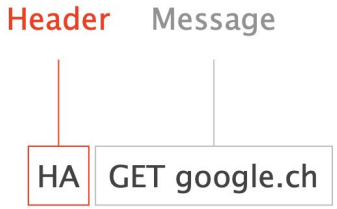
	layer	role
L5	Application	exchanges messages between processes
L4	Transport	transports segments between end systems
L3	Network	moves packets around the network
L2	Link	moves frames across a link
L1	Physical	moves bits across a physical medium

Each layer takes messages from the layer above, and encapsulates with its own header and/or trailer

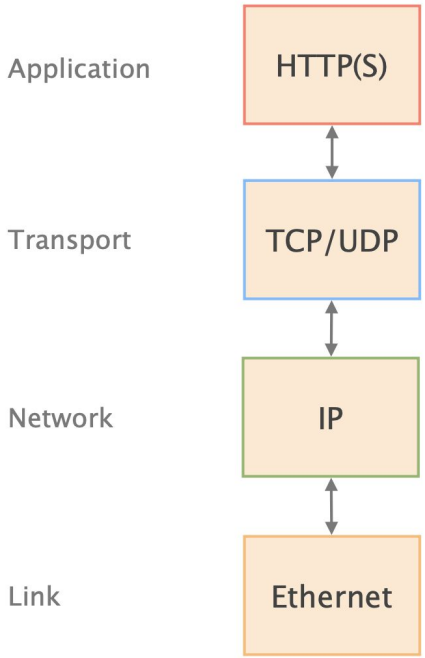


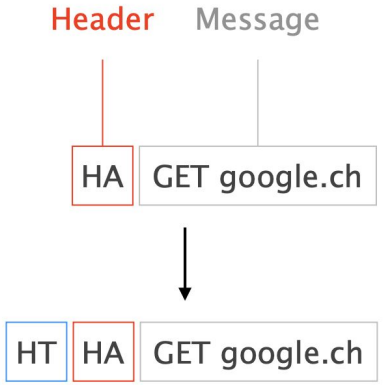
Each layer takes messages from the layer above, and encapsulates with its own header and/or trailer



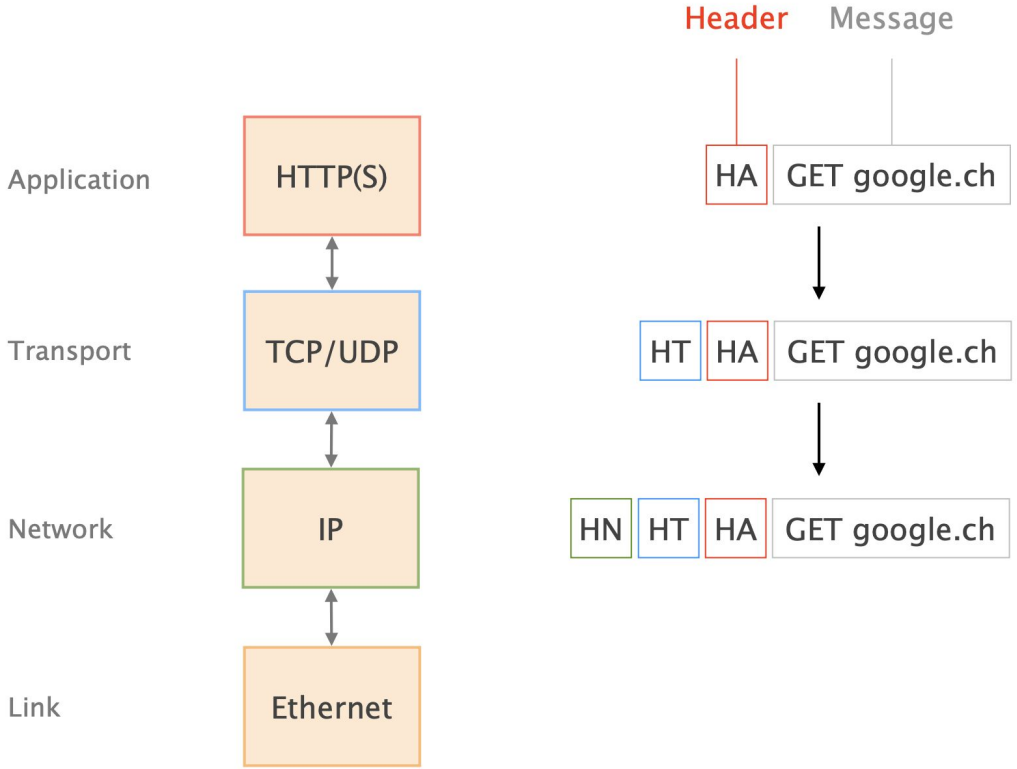


Each layer takes messages from the layer above, and encapsulates with its own header and/or trailer





Each layer takes messages from the layer above, and encapsulates with its own header and/or trailer



Each layer takes messages from the layer above, and encapsulates with its own header and/or trailer