Computer Networks and Internet Technology

2021W703033 VO Rechnernetze und Internettechnik
Winter Semester 2021/22

Jan Beutel

universität innsbruck

With materials and inspiration from Jim Kurose, Scott Shenker, Jennifer Rexford, Phil Levis, Adrian Perrig and Laurent Vanbever

Lecture Recap

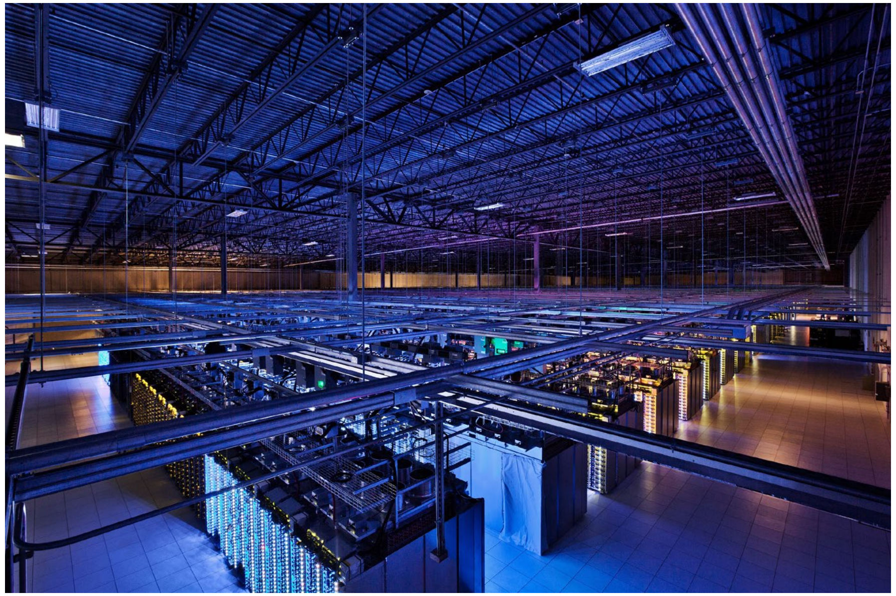
Communication Networks

So what?!

Knowledge

Understand how the Internet works and why

from your network plug...



...to Google's data-center

You should now be able to …

List any

technologies, principles, applications…

used after typing in:

> www.google.at

and pressing enter in your browser

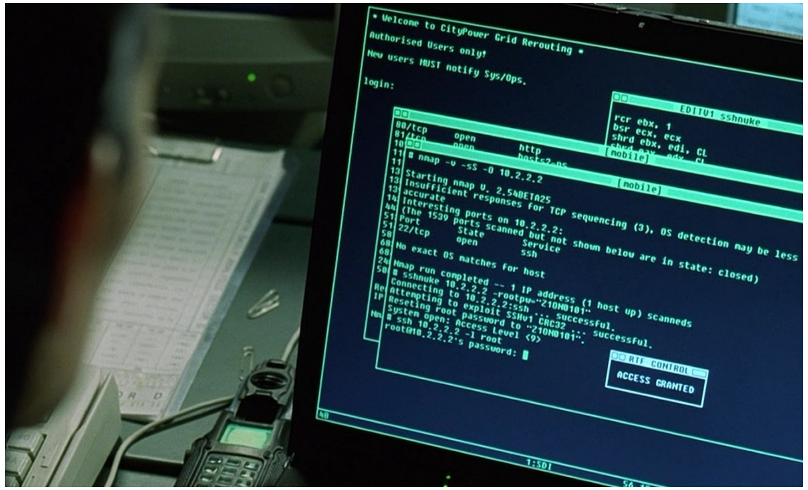
Insight

Key concepts and problems in Networking

Naming Layering Routing Reliability Sharing

Skill

Build, operate and configure networks



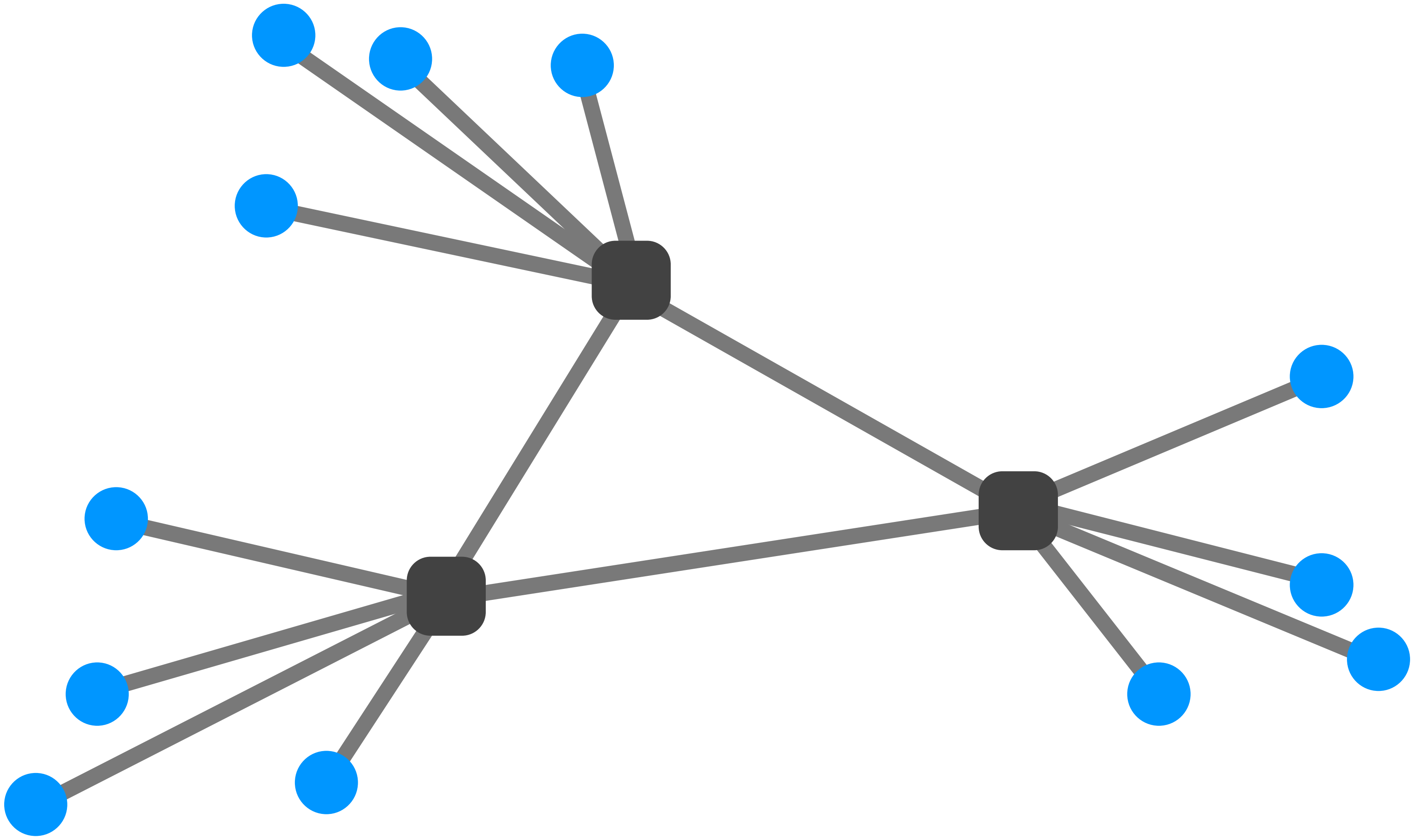
Trinity using a port scanner (`nmap`) in Matrix Reloaded™

Communication Networks and Internet Technology

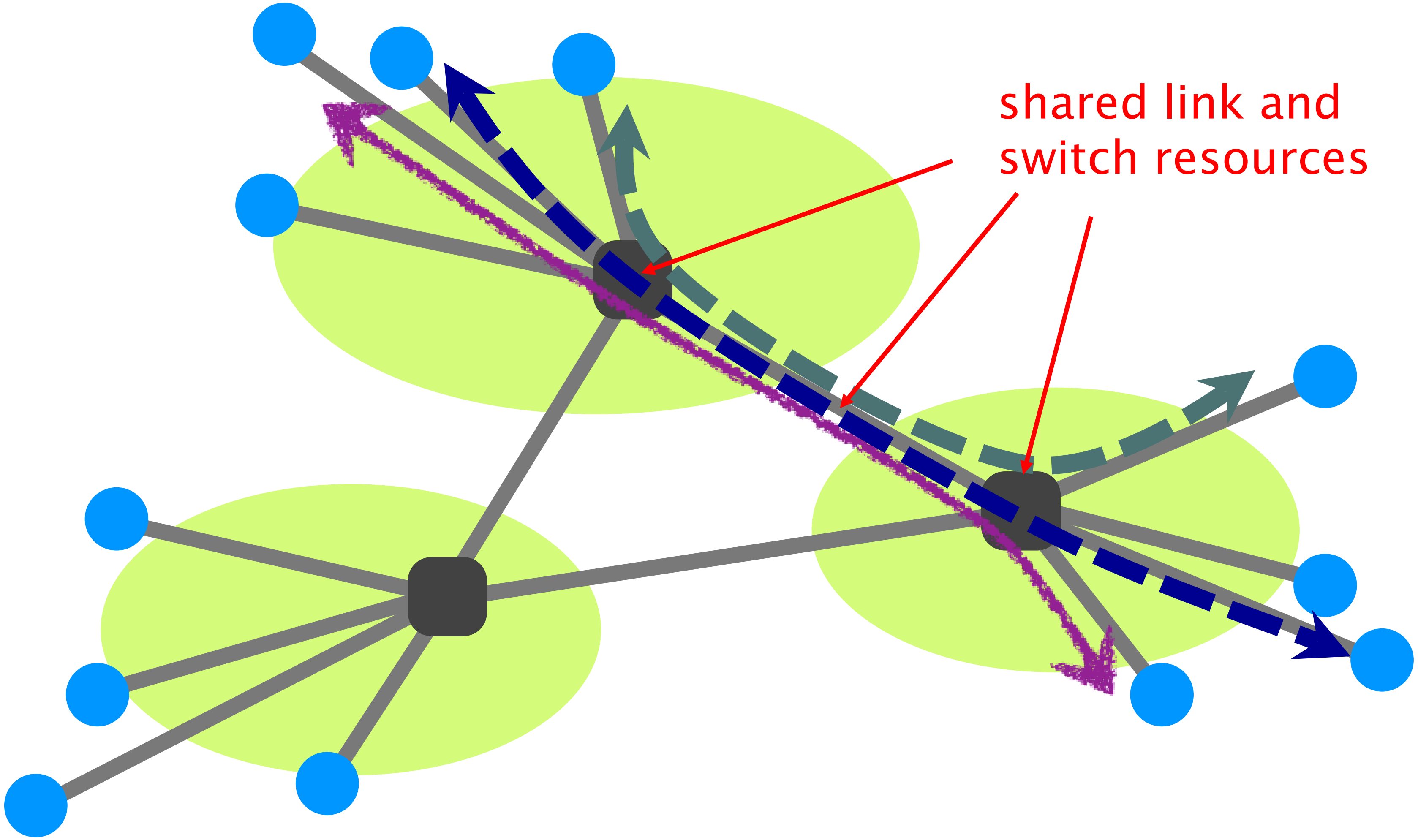
Part 1: Overview

#1	What is a network made of?
#2	How is it shared?
#3	How is it organized?
#4	How does communication happen?
#5	How do we characterize it?

Networks are composed of three basic components

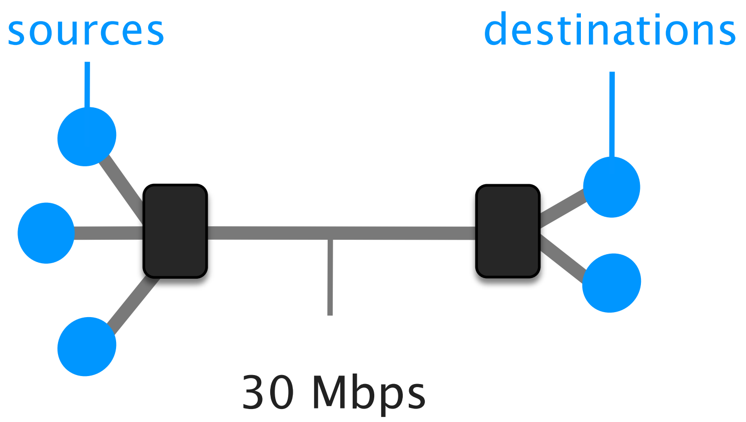


Links and switches are shared between flows



Between reservation and on-demand:

Which one do you pick?

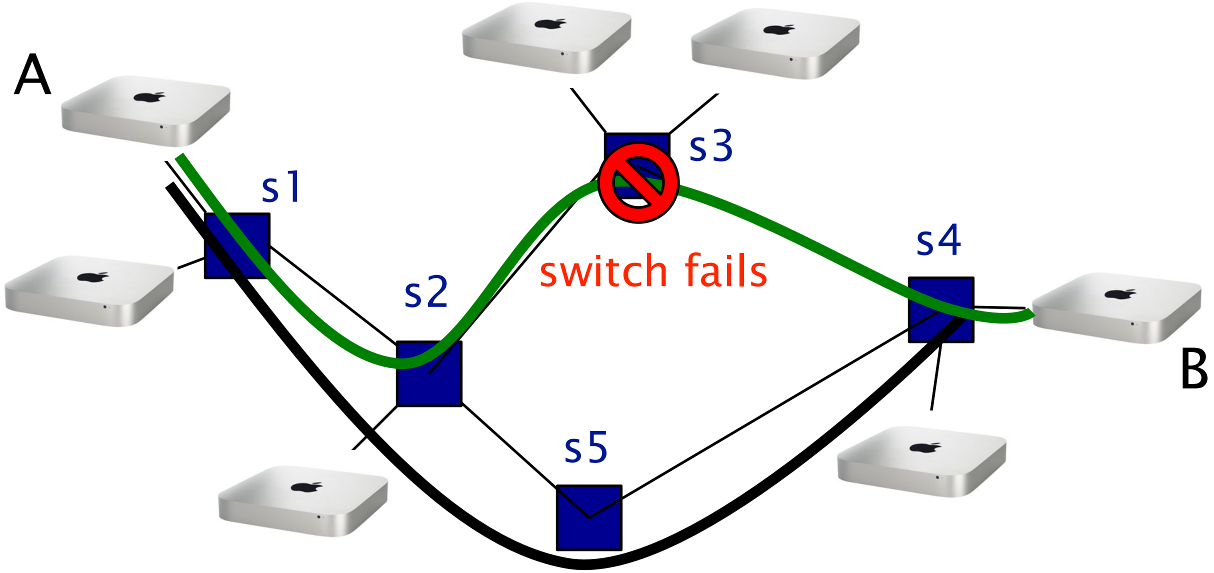


Consider that each source needs 10 Mbps

What do they get with:

- reservation
- on-demand

Packet switching routes around trouble

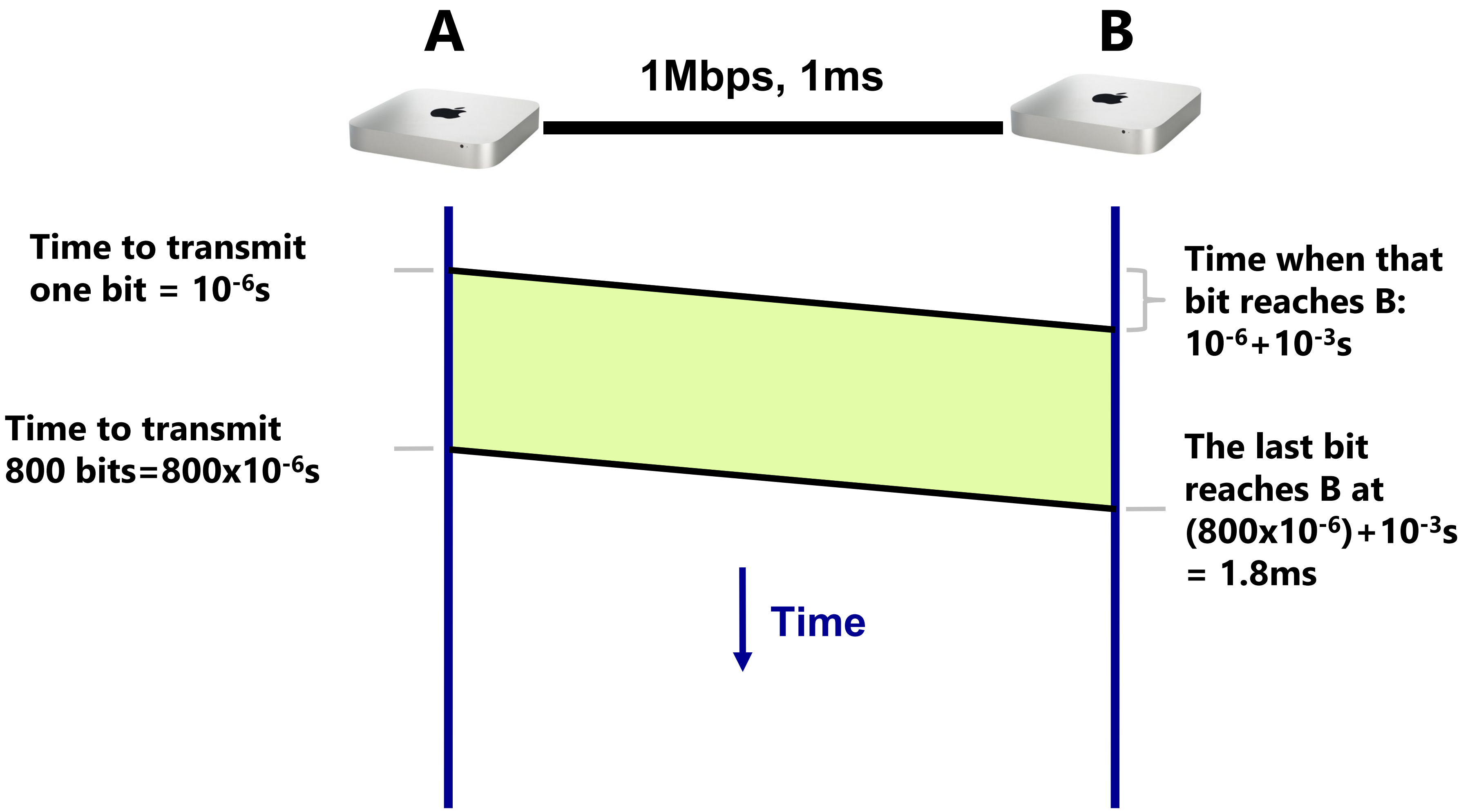


route is recomputed on the fly by s2

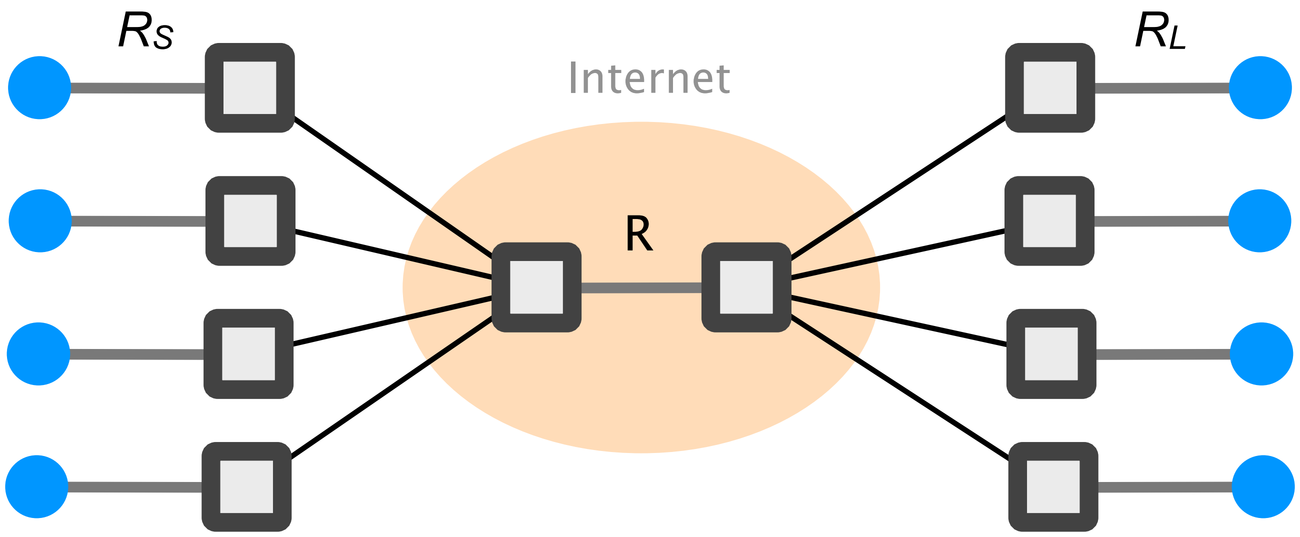
Each packet suffers from several types of delays at *each node* along the path

	transmission	delay	↕	due to link properties
⊕	propagation	delay		
⊕	processing	delay	↕	due to traffic mix & switch internals
⊕	queuing	delay		
=	total	delay		

How long does it take to exchange 100 Bytes packet?



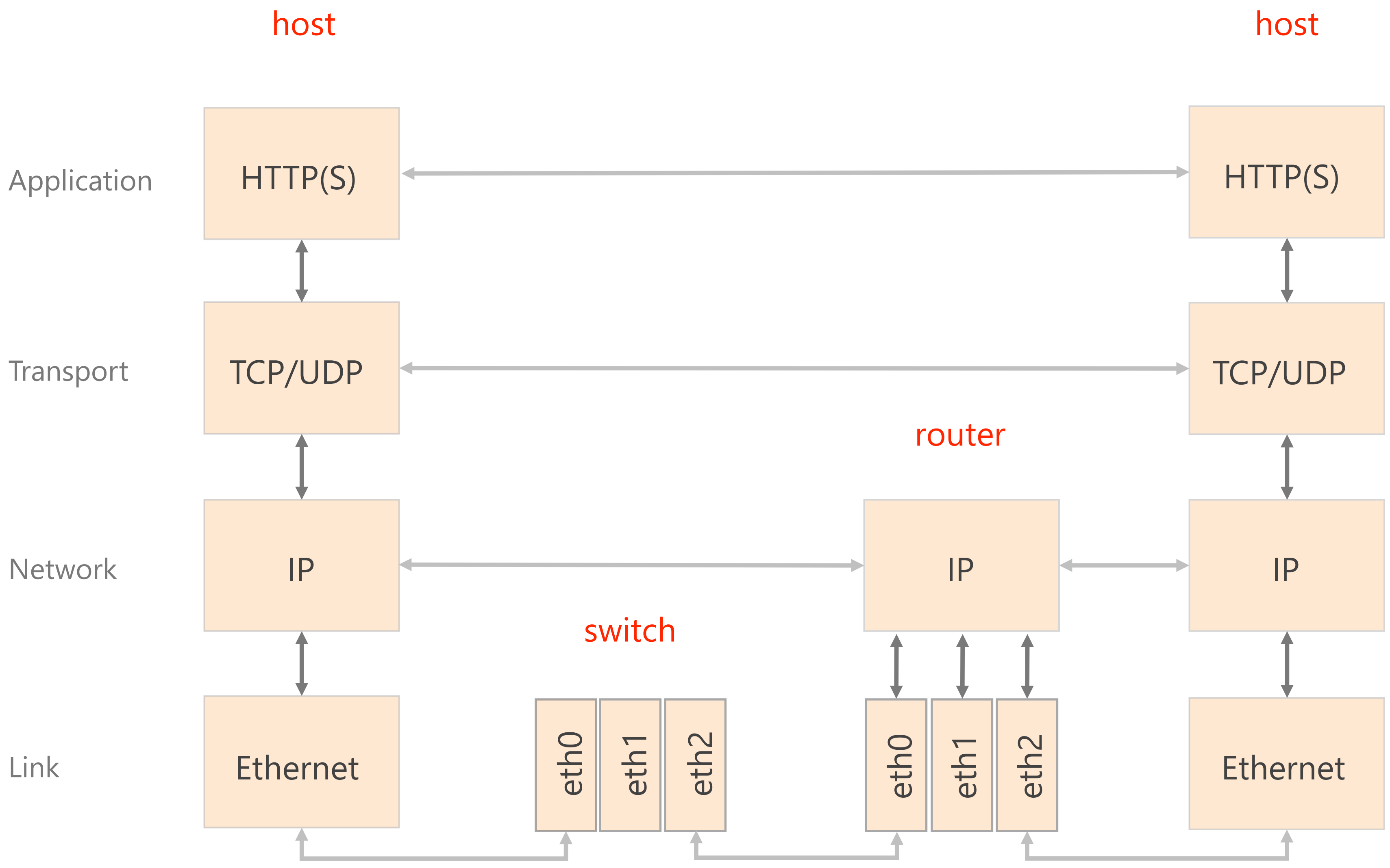
To compute throughput, one has to consider the bottleneck link… and the intervening traffic



if $4 \cdot \min(R_S, R_L) > R$ — the bottleneck is now in the core, providing each download R/4 of throughput

The Internet is organized as layers, providing a set of services

	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


host
host
Application
HTTP(S)
HTTP(S)
Transport
TCP/UDP
TCP/UDP
router
Network
IP
IP
IP
switch
Link
Ethernet
eth0
eth1
eth2
eth0
eth1
eth2
Ethernet

We started with the fundamentals of
routing and reliable transport

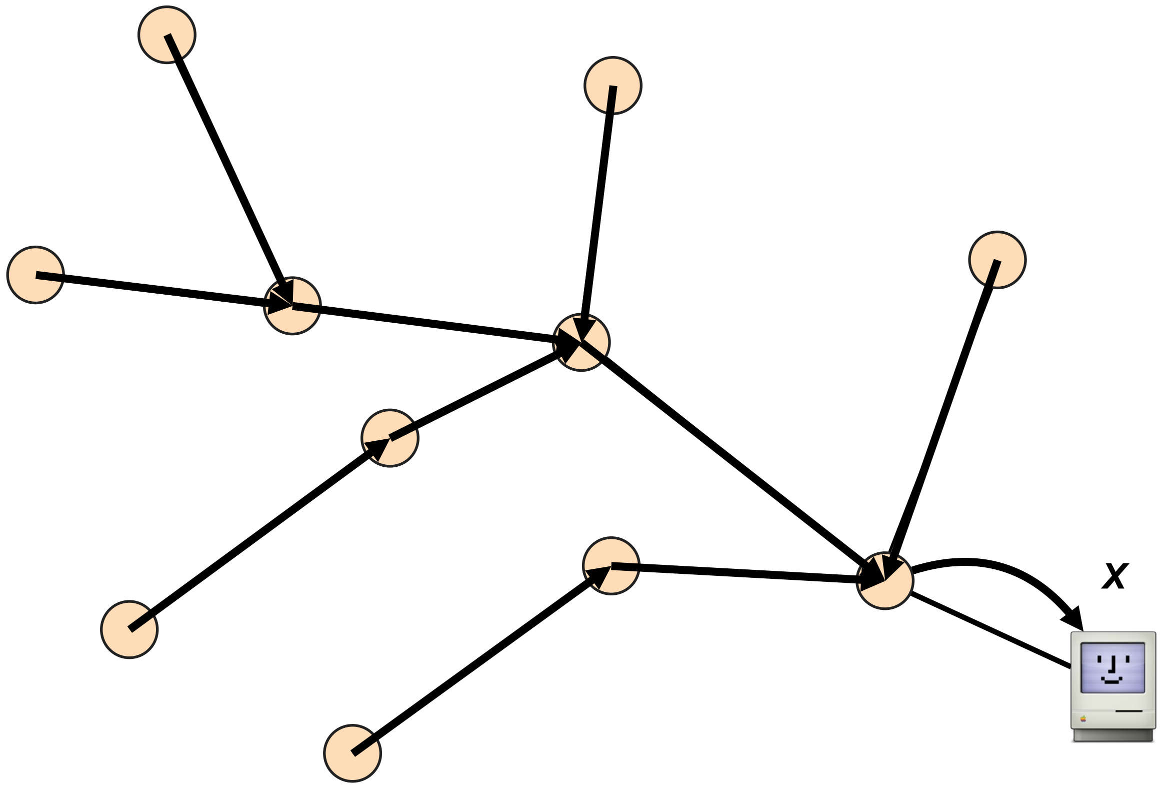
	Application	network access
L4	Transport	end-to-end delivery (reliable or not)
L3	Network	*global best-effort delivery*
	Link	*local best-effort delivery*
	Physical	physical transfer of bits

We saw three ways to compute valid routing state

	Intuition	Example
#1	Use tree-like topologies	Spanning-tree
#2	Rely on a global network view	Link-State SDN
#3	Rely on distributed computation	Distance-Vector BGP

The result is a spanning tree.

This is a valid routing state



Once a node *u* knows the entire topology,
it can compute shortest-paths using **Dijkstra's algorithm**

Initialization

S = {*u*}

for all nodes *v*:

    if (*v* is adjacent to *u*):

        D(*v*) = c(*u*,*v*)

    else:

        D(*v*) = ∞

Loop

while *not* all nodes in S:

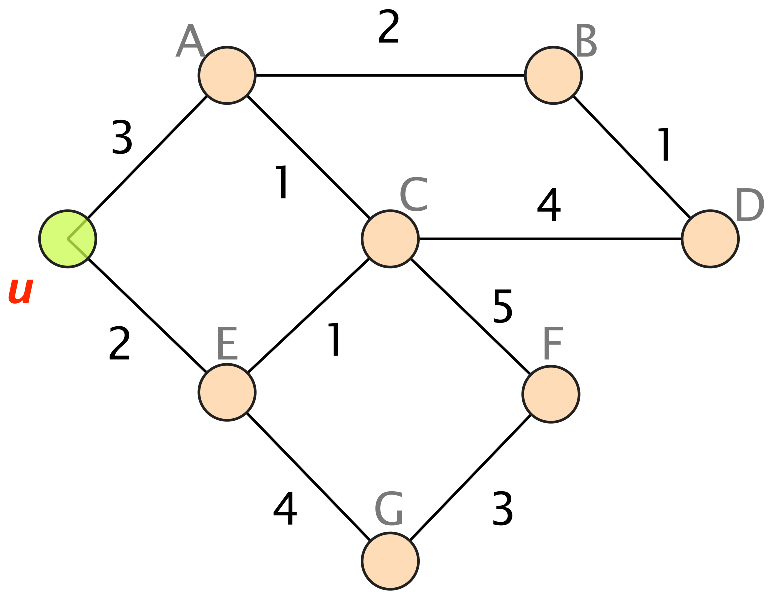
    add *w* with the smallest D(*w*) to S

    update D(*v*) for all adjacent *v* not in S:

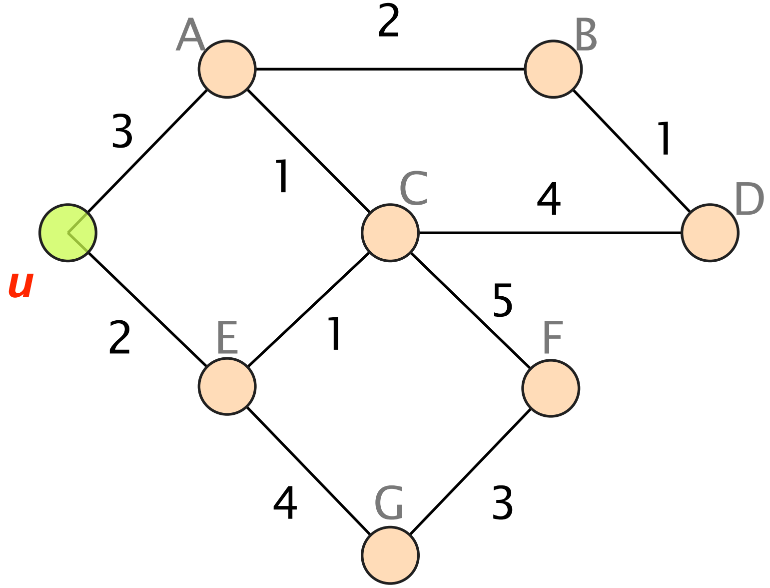
        D(*v*) = min{D(*v*), D(*w*) + c(*w*,*v*)}

Let's compute the shortest-paths

from u



The values computed by a node *u*
depends on what it learns from its neighbors (A and E)



$$d_x(y) = \min\{\ c(x,v) + d_v(y)\ \}$$

over all neighbors *v*

$$d_u(D) = \min\{\ c(u,A) + d_A(D),\ c(u,E) + d_E(D)\ \}$$

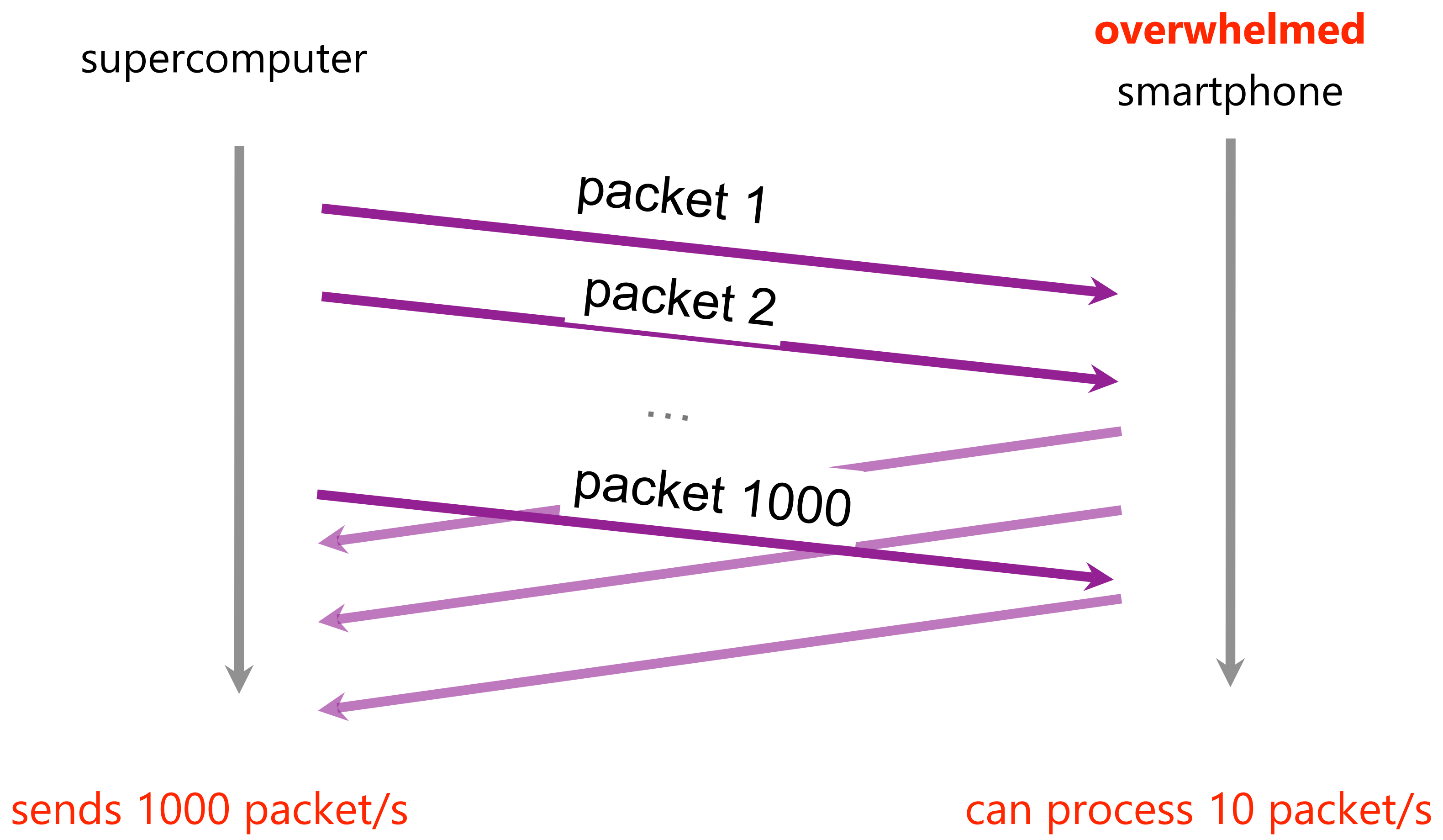
We saw how to design a reliable transport protocol

and you implemented one yourself

goals

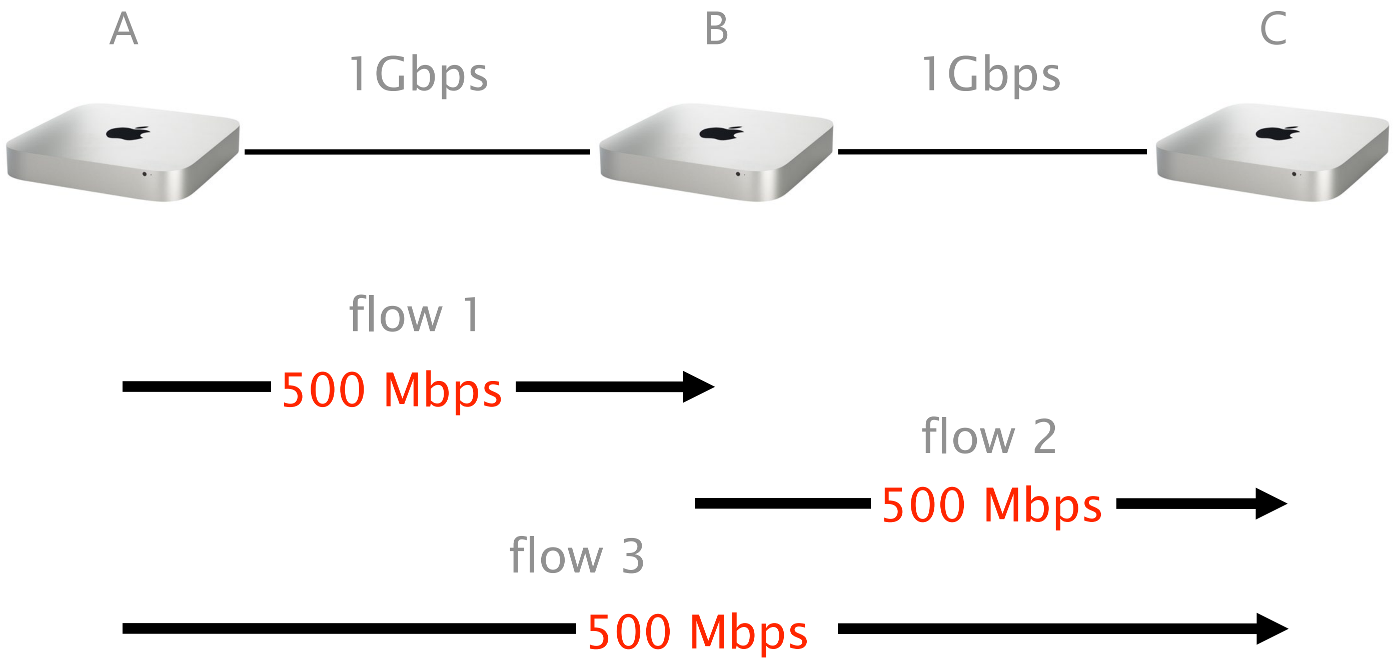
correctness	ensure data is delivered, in order, and untouched
timeliness	minimize time until data is transferred
efficiency	optimal use of bandwidth
fairness	play well with other concurrent communications

Sending multiple packets improves timeliness, but it can also overwhelm the receiver



An equal allocation is certainly "fair",

but what about the **efficiency of the network**?



Total traffic is 1.5 Gbps

In each case, we explored the rationale behind each protocol and why they came to be

Why did the protocols end up looking like this?

minimum set of features required

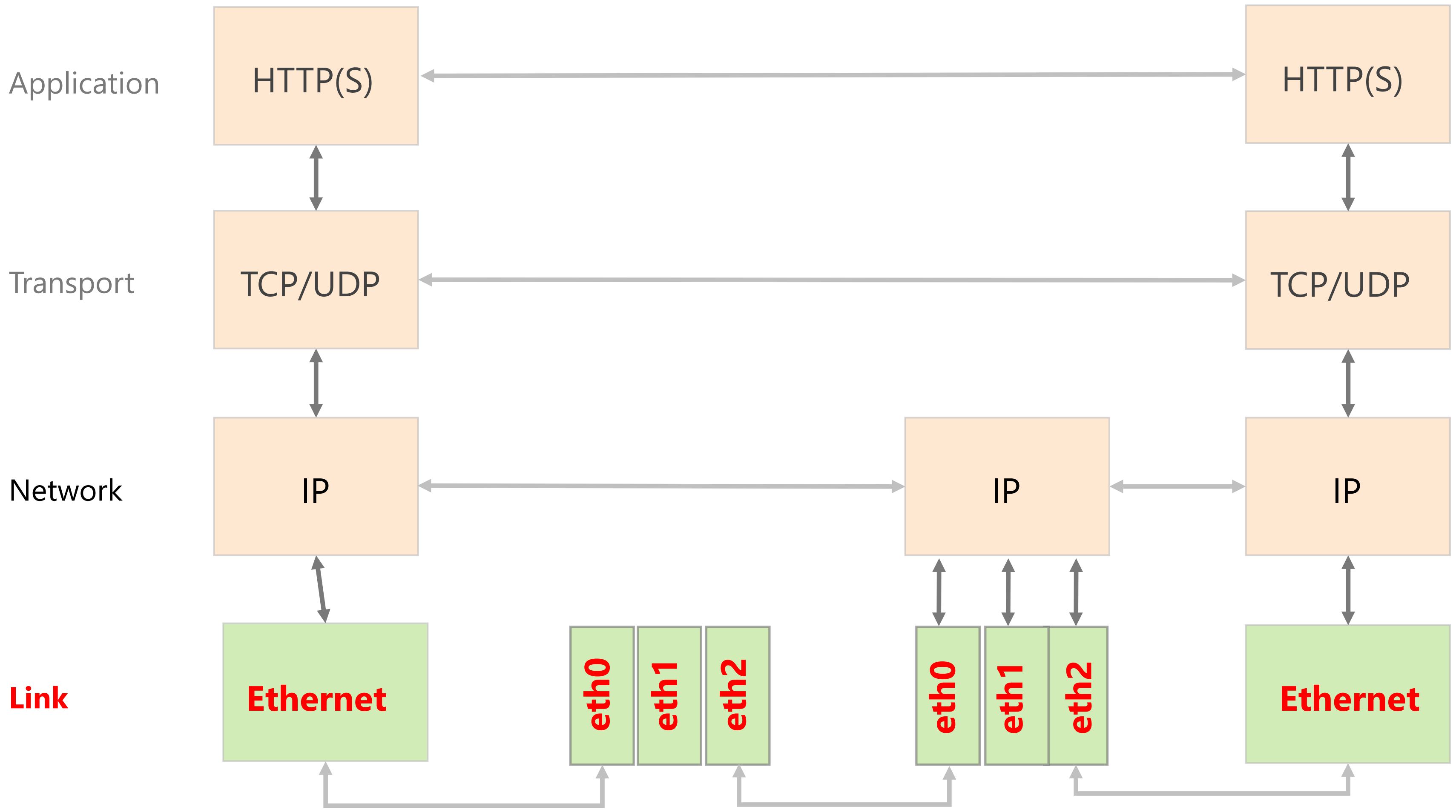
What tradeoffs do they achieve?

efficiency, cost,…

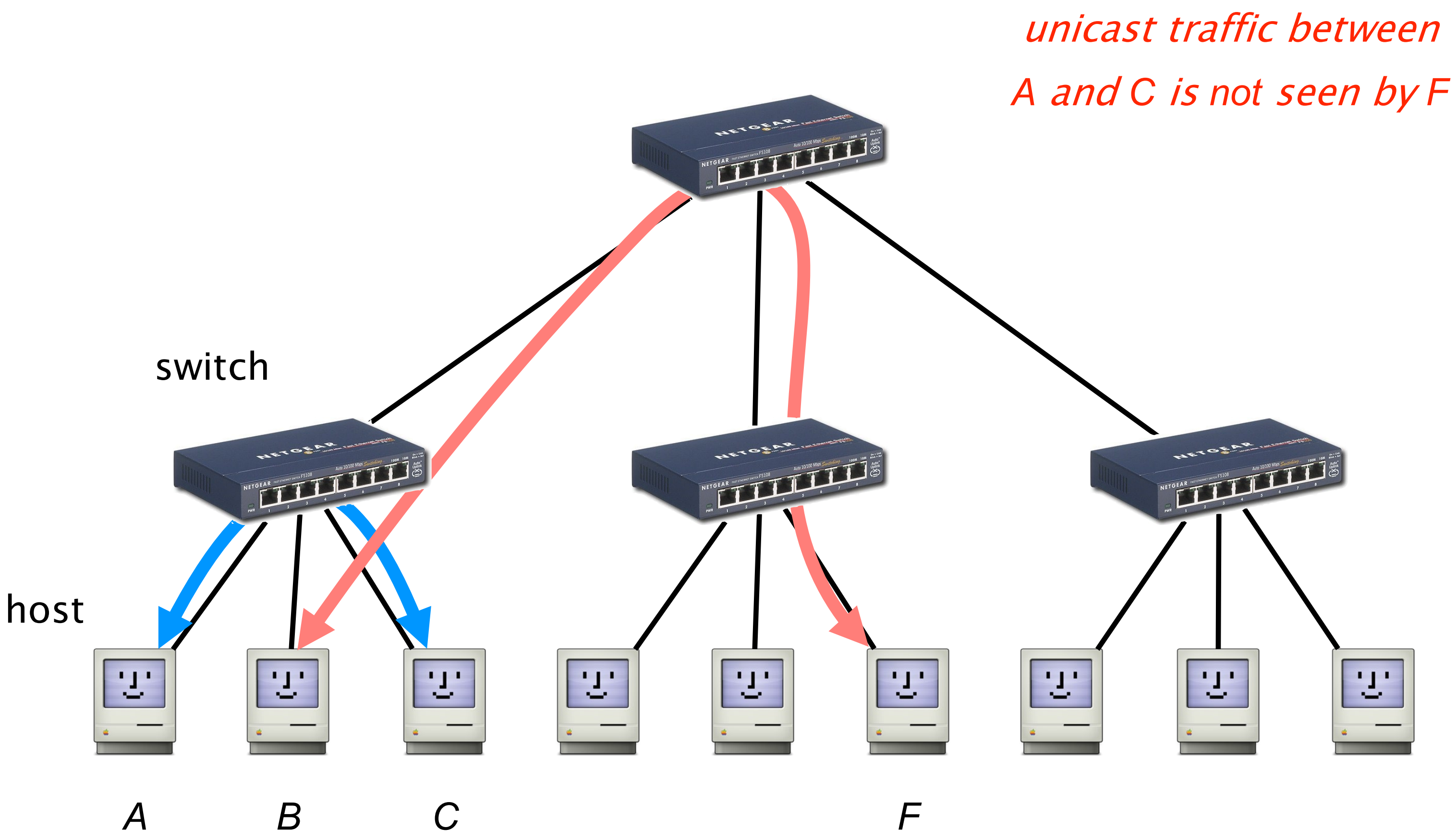
When is one design more adapted than another?

packet switching *vs* circuit switching, DV *vs* LS,…

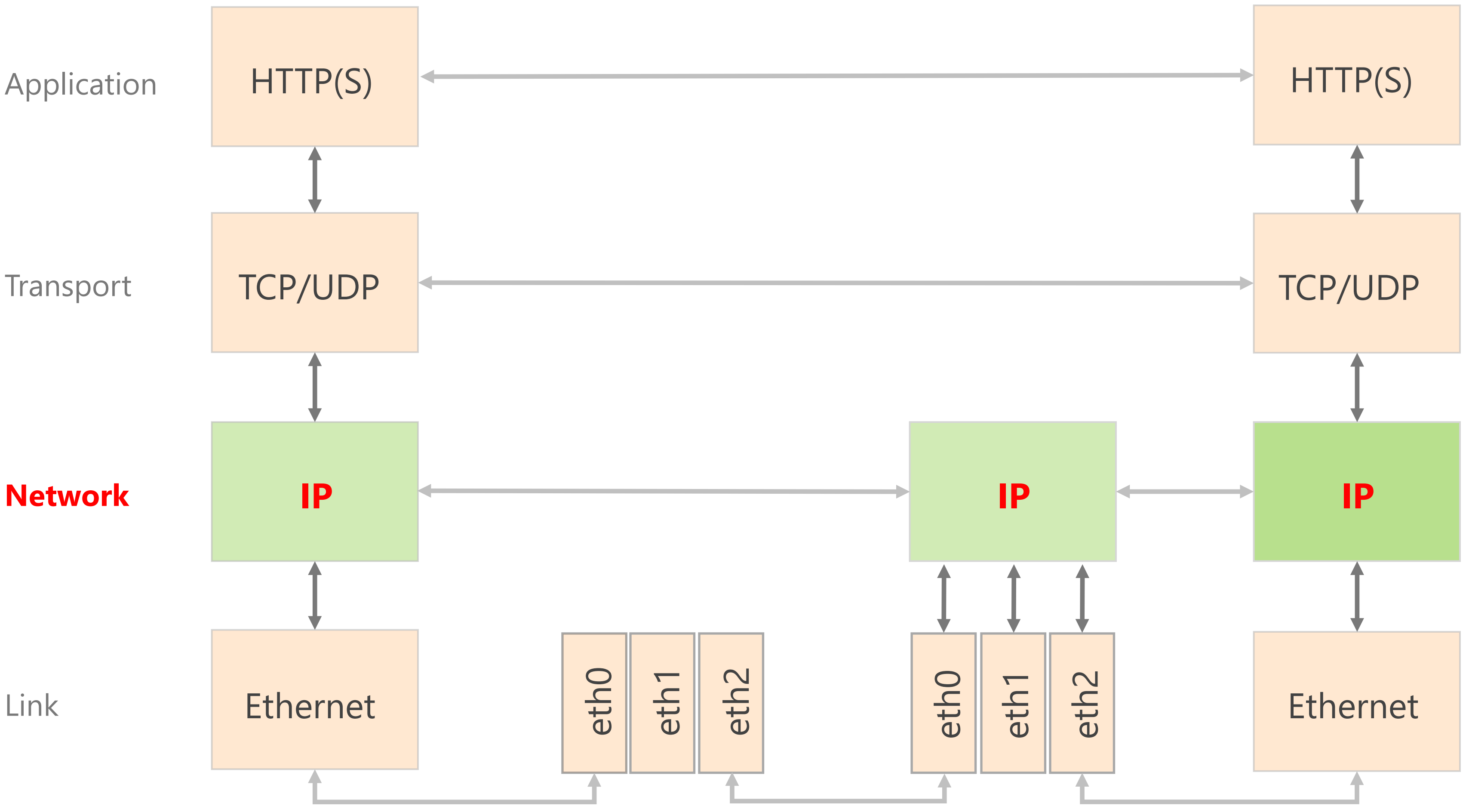
We then climbed up the layers, starting from layer 2



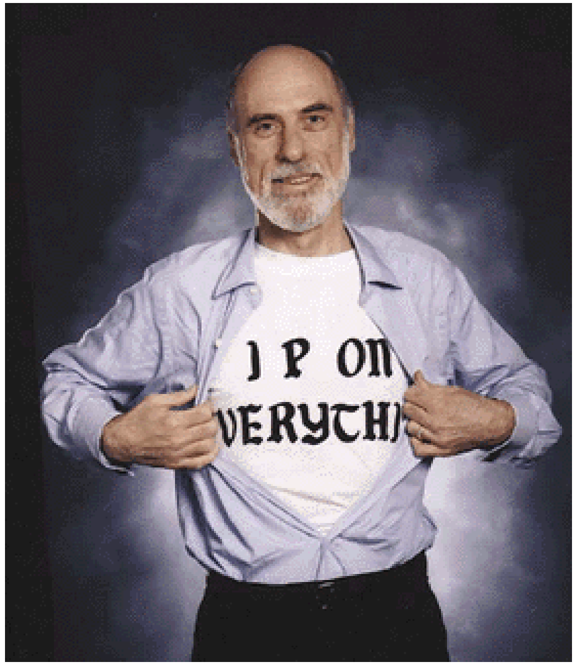
Unlike with hubs, switches enable each LAN segment to carry its own traffic



We then spent multiple weeks on layer 3



Internet Protocol and Forwarding



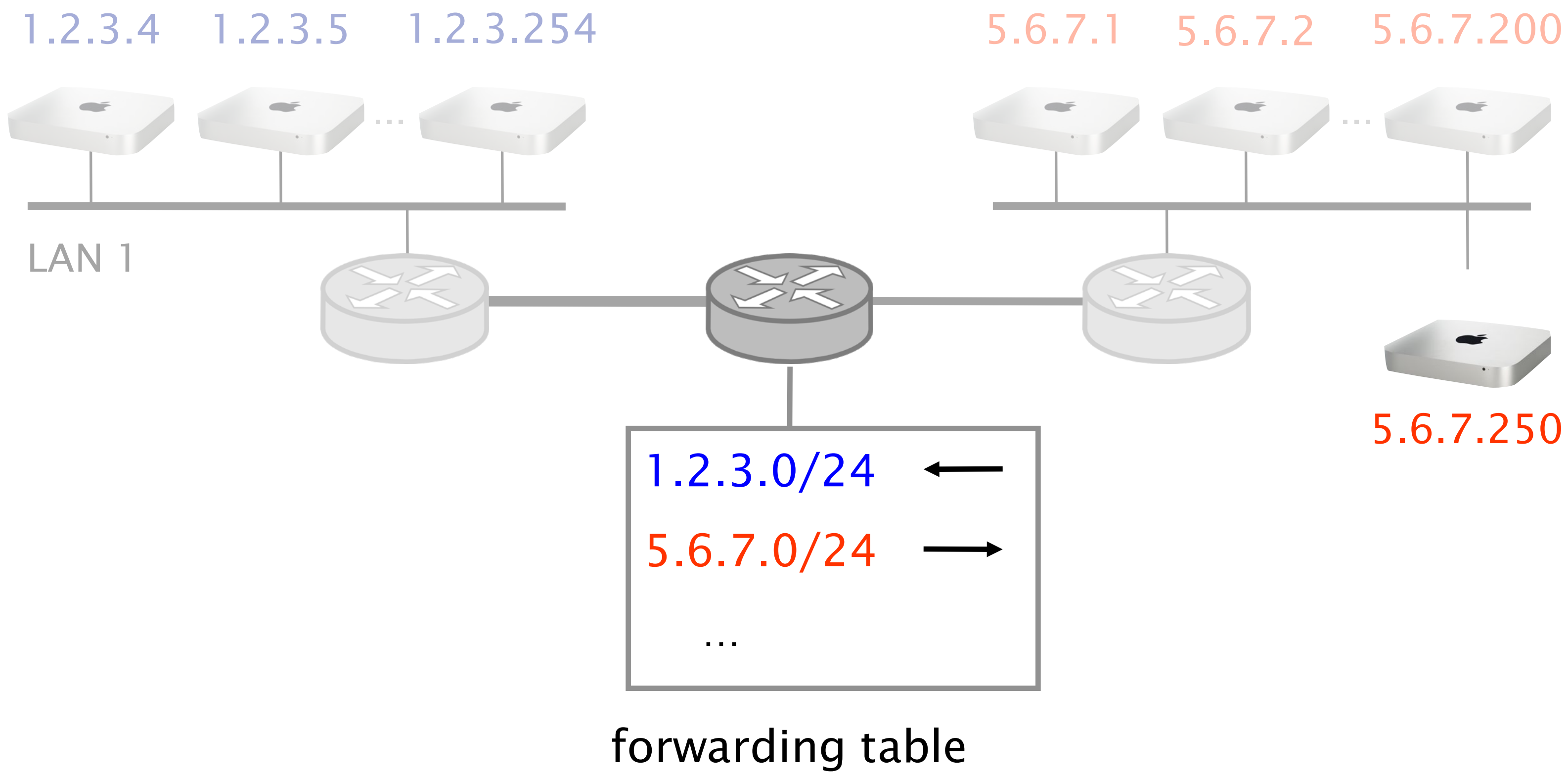
source: Boardwatch Magazine

1 **IP addresses**
use, structure, allocation

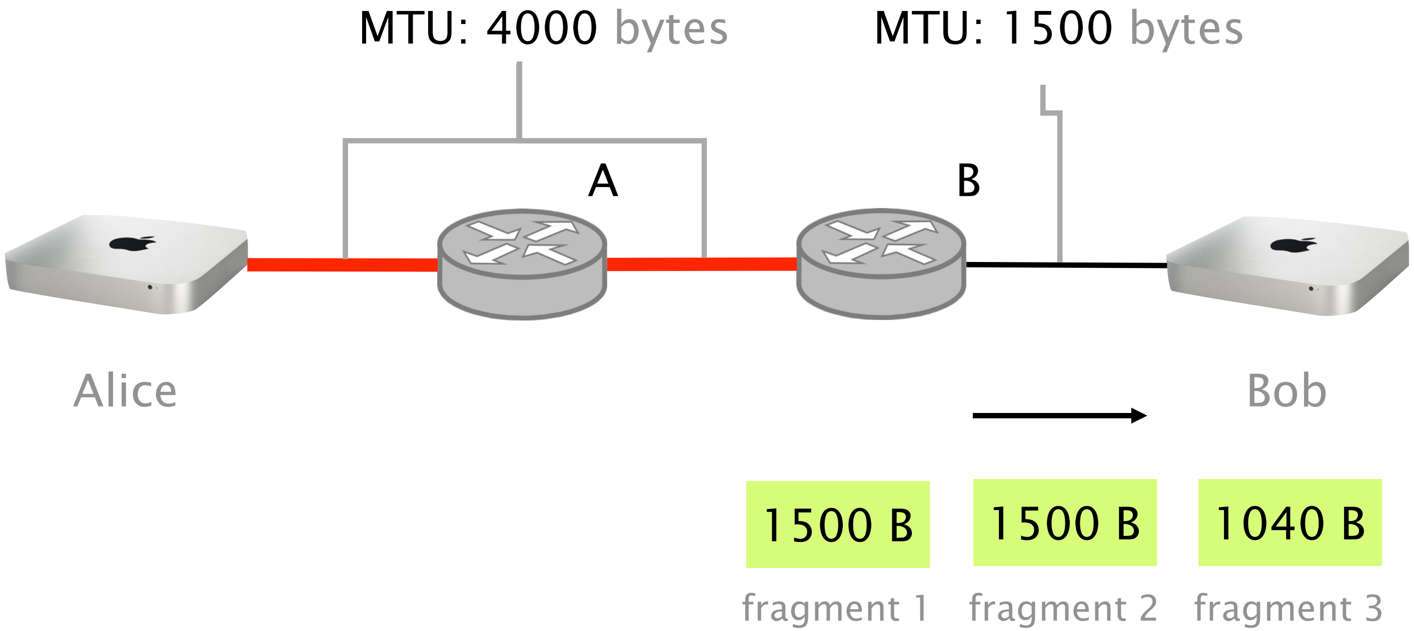
2 **IP forwarding**
longest prefix match rule

3 **IP header**
IPv4 and IPv6, wire format

Hierarchical addressing enables to add new hosts without changing or adding forwarding rules



Because the packet is larger than the MTU,
router B will split the packet into fragments



We also talked about **IPv6**

Internet routing

from here to there, and back

1 **Intra-domain routing**

Link-state protocols

Distance-vector protocols

2 **Inter-domain routing**

Path-vector protocols

Internet routing comes into two flavors:

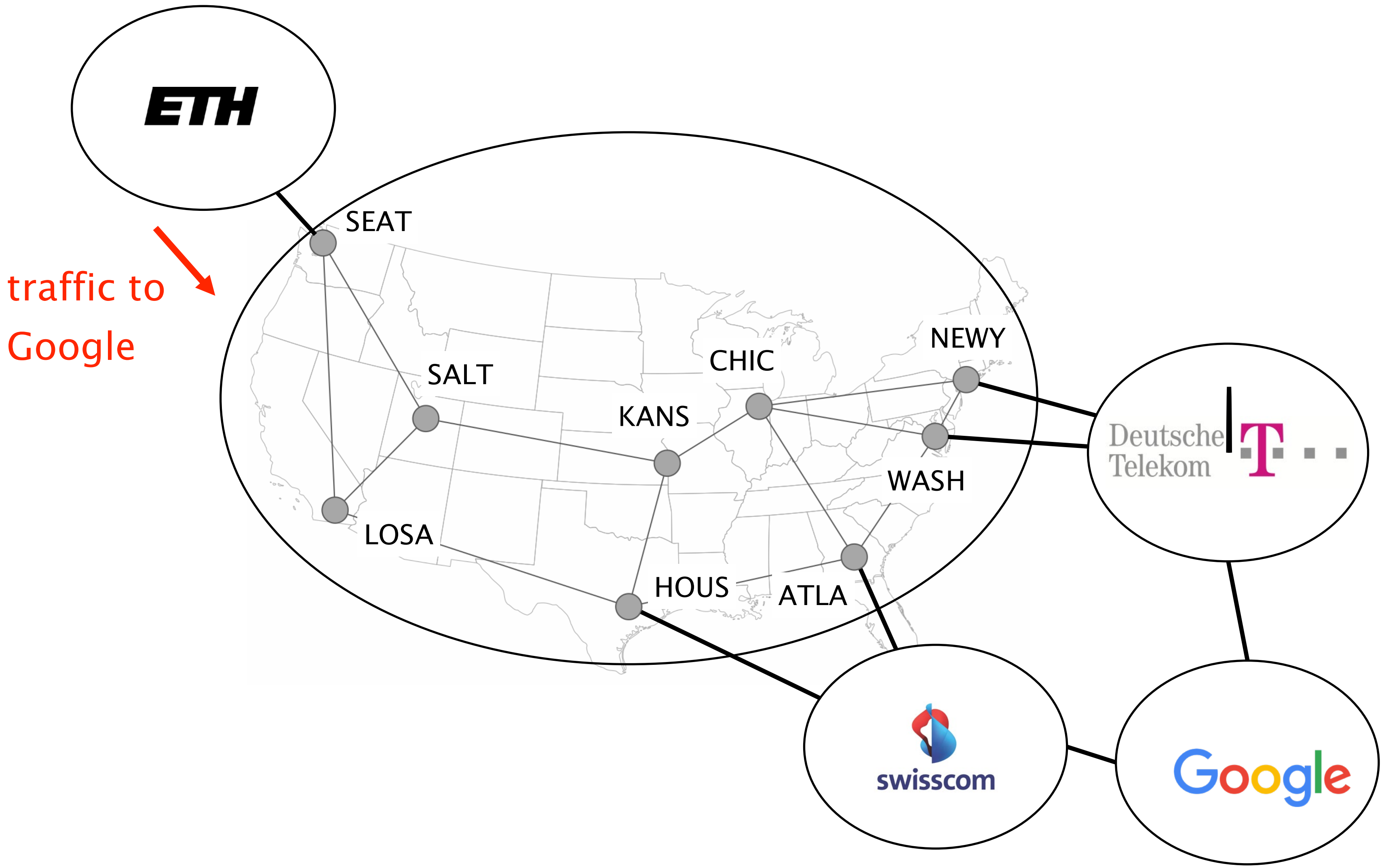
intra- and *inter-domain* routing

inter-domain routing	intra-domain routing
Find paths between networks	Find paths within a network

inter-domain
routing

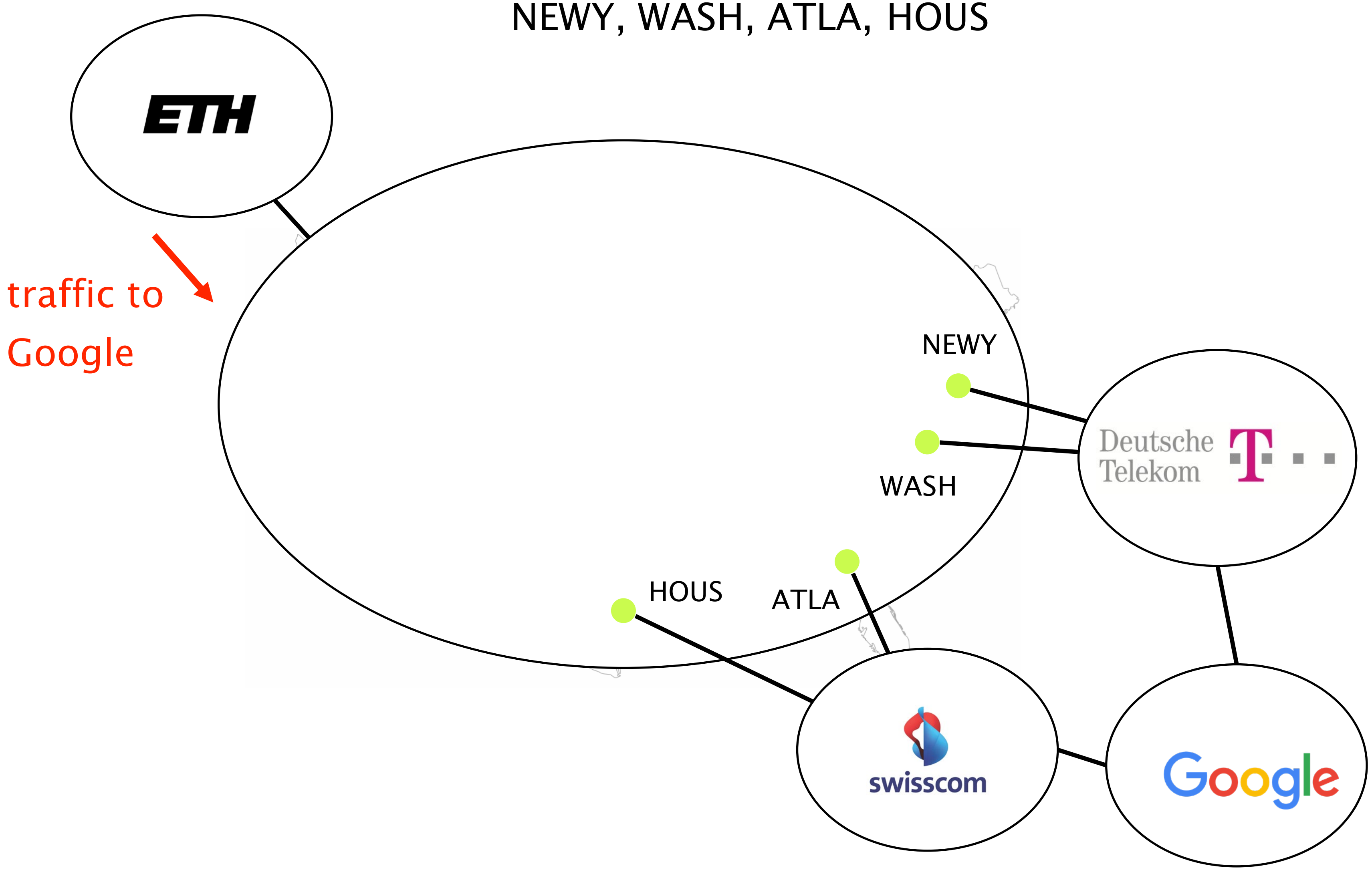
intra-domain
routing

Find paths between networks



traffic to
Google
SEAT
SALT
KANS
CHIC
NEWY
WASH
LOSA
HOUS
ATLA
Deutsche
Telekom
swisscom
Google

Google can be reached via
NEWY, WASH, ATLA, HOUS



inter-domain

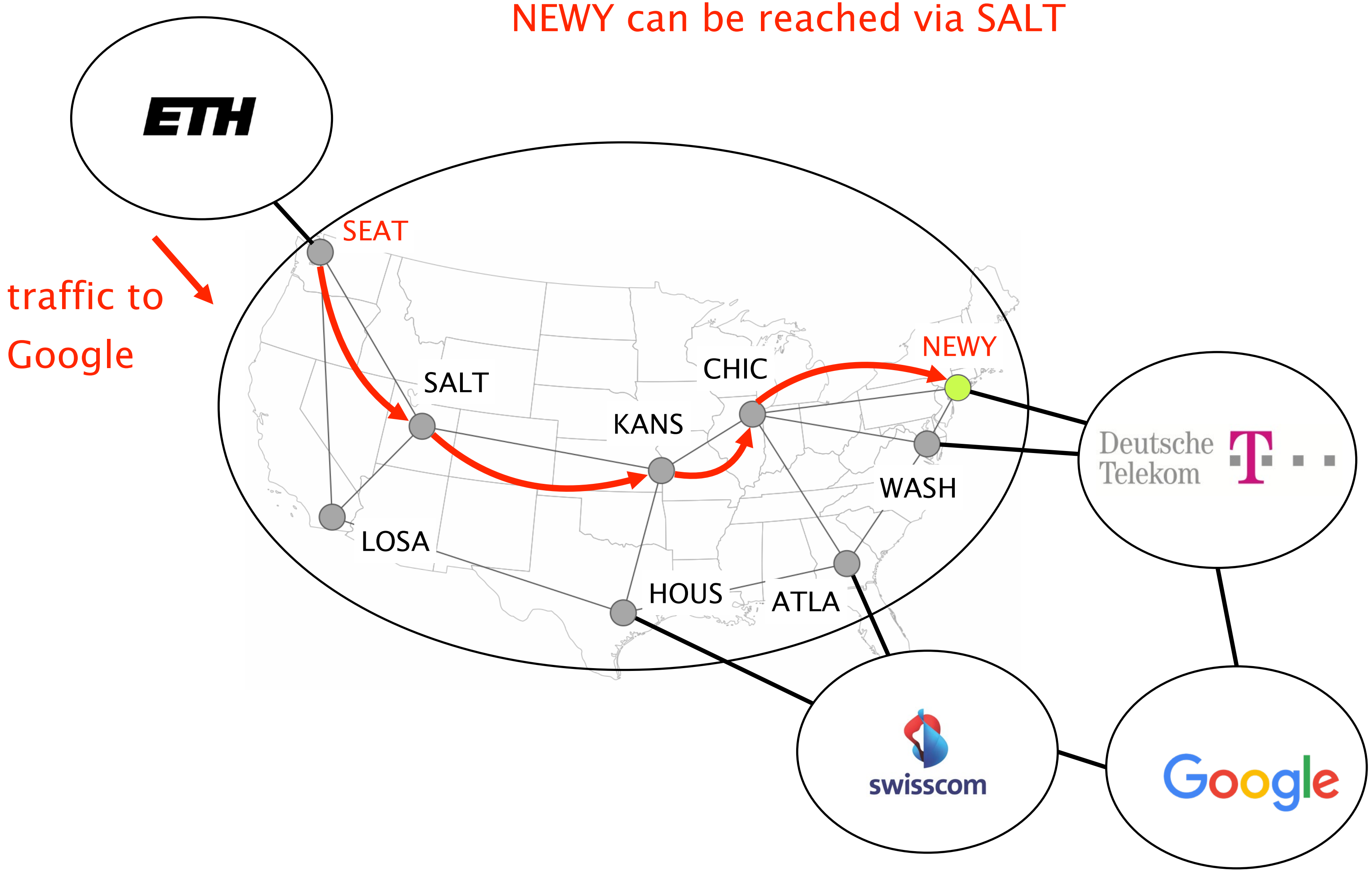
routing

intra-domain

routing

Find paths within a network

NEWY can be reached via SALT



Border Gateway Protocol

policies and more

1 **BGP Policies**

Follow the money

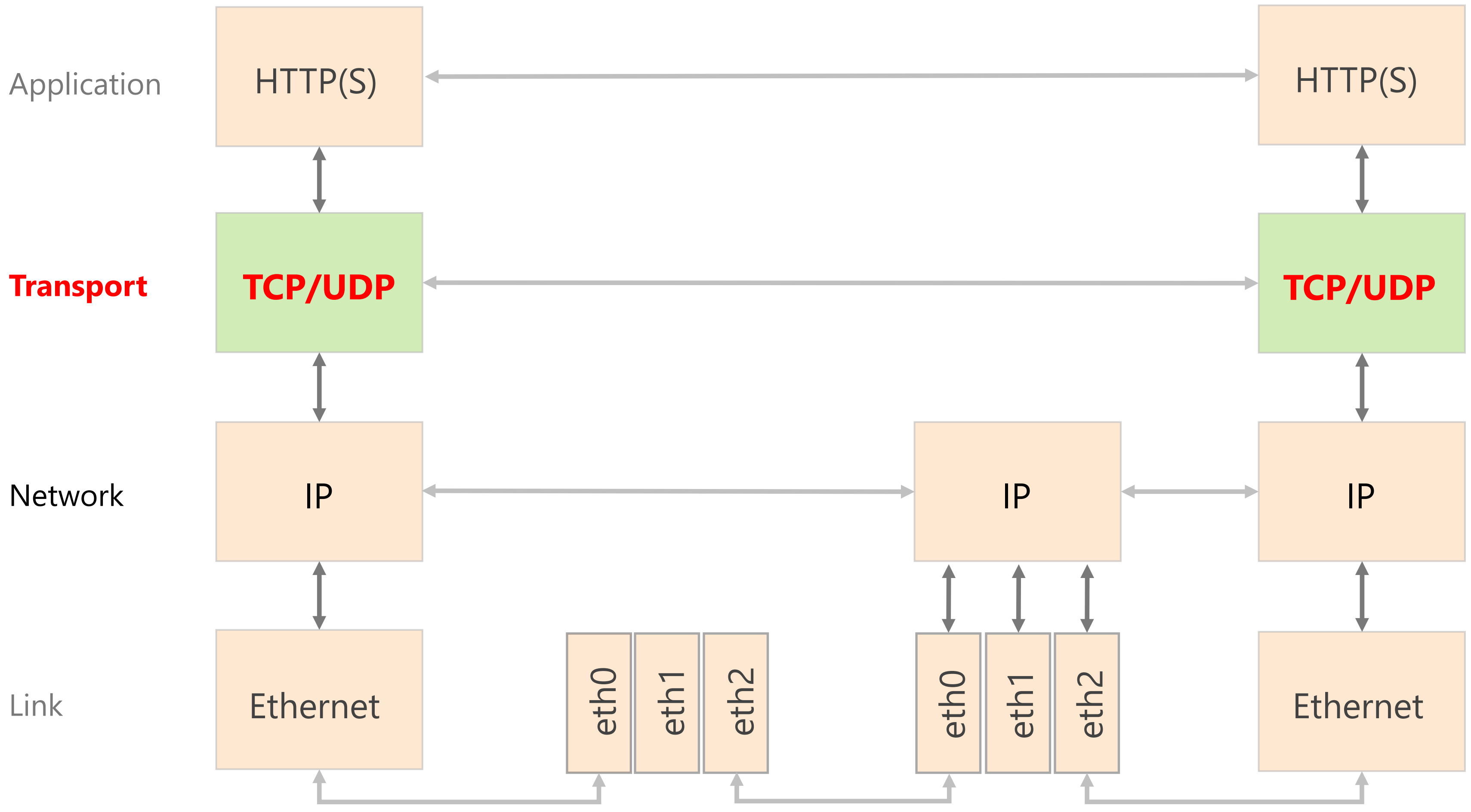
2 **Protocol**

How does it work?

3 **Problems**

security, performance, …

Layer 4 Transport Abstractions



Important Context: Sockets and Ports

- **Sockets**: an operating system abstraction

- **Ports**: a networking abstraction
  - This is not a port on a switch (which is an interface)
  - Think of it as a *logical interface* on a host

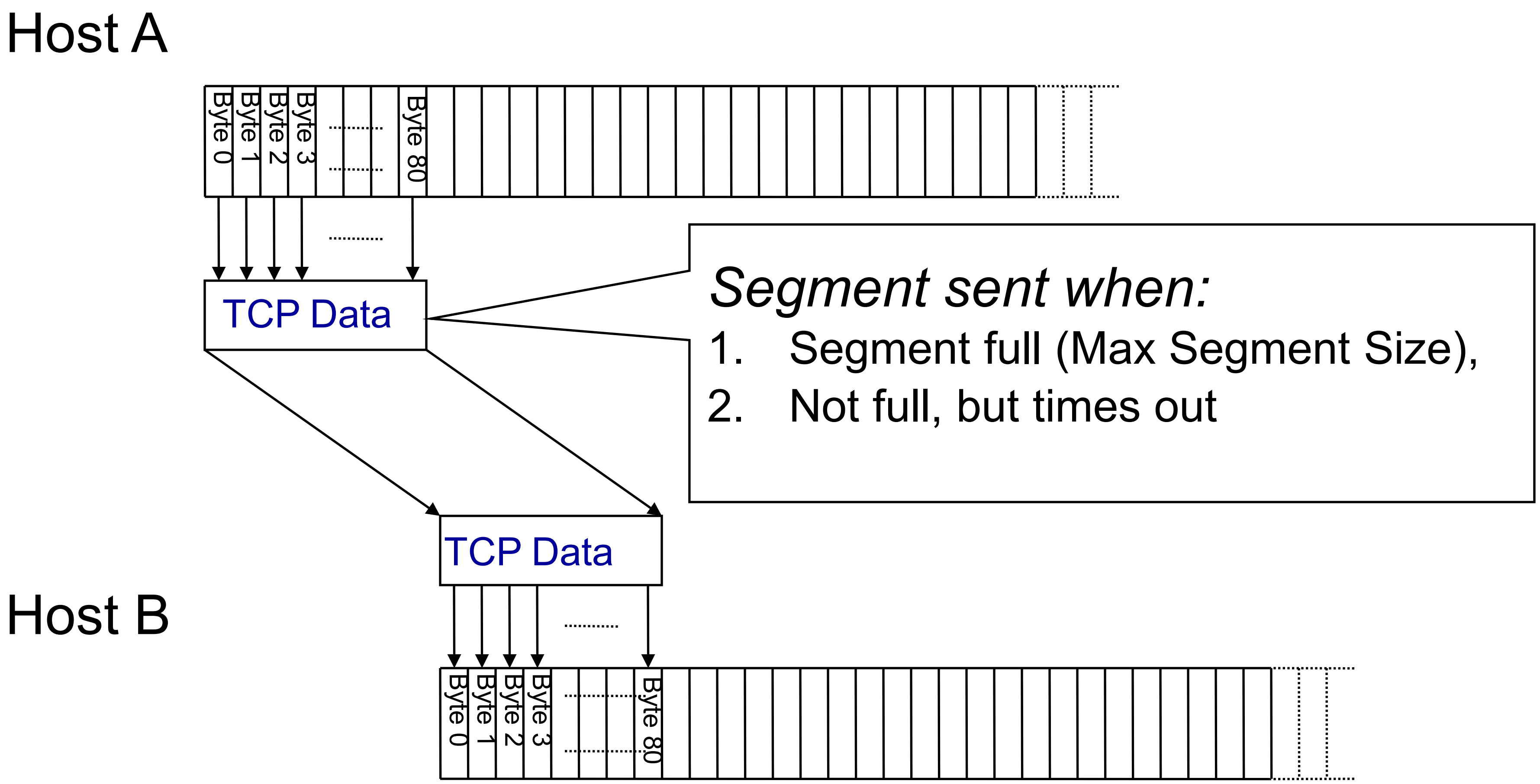
Client OS		src IP	src port	dest IP	dest port
socket	1	129.132.19.1	54001	172.217.168.3	443
	2	129.132.19.1	55240	172.217.168.3	443
	3	129.132.19.1	48472	172.217.168.3	443
	4	129.132.19.1	35456	172.217.168.3	443
	5	129.132.19.1	42001	172.217.168.3	443

Server OS		src IP	src port	dest IP	dest port
socket	1	172.217.168.3	443	129.132.19.1	54001
	2	172.217.168.3	443	129.132.19.1	55240
	3	172.217.168.3	443	129.132.19.1	48472
	4	172.217.168.3	443	129.132.19.1	35456
	5	172.217.168.3	443	129.132.19.1	42001

We looked at the requirements and implementation of transport protocols (UDP/TCP)

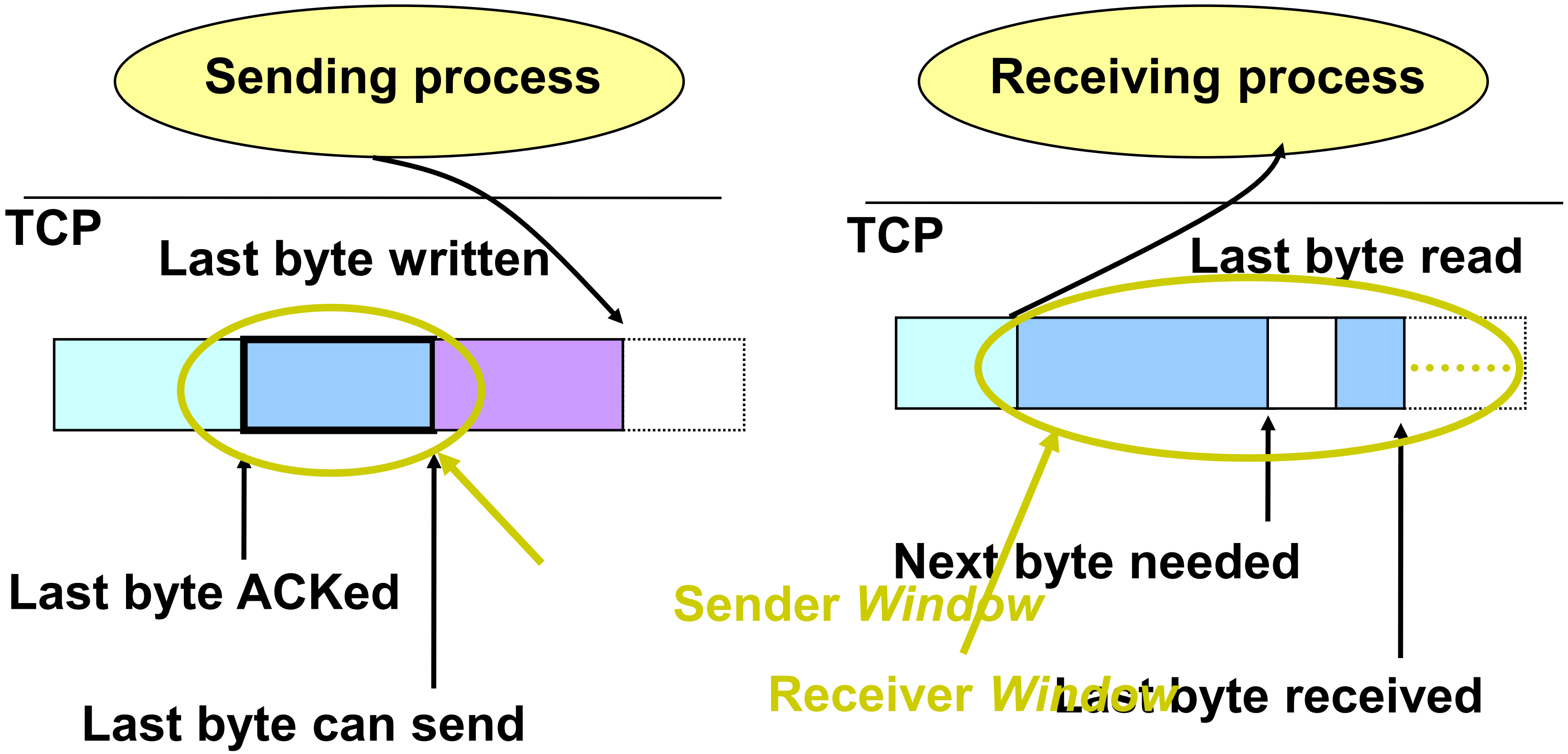
- Data delivering, to the *correct* application
  - IP just points towards next protocol
  - *Transport needs to demultiplex incoming data (ports)*
- Files or bytestreams abstractions for the applications
  - Network deals with packets
  - *Transport layer needs to translate between them*
- Reliable transfer (if needed)
- Not overloading the receiver
- Not overloading the network

… Provided Using TCP "Segments"

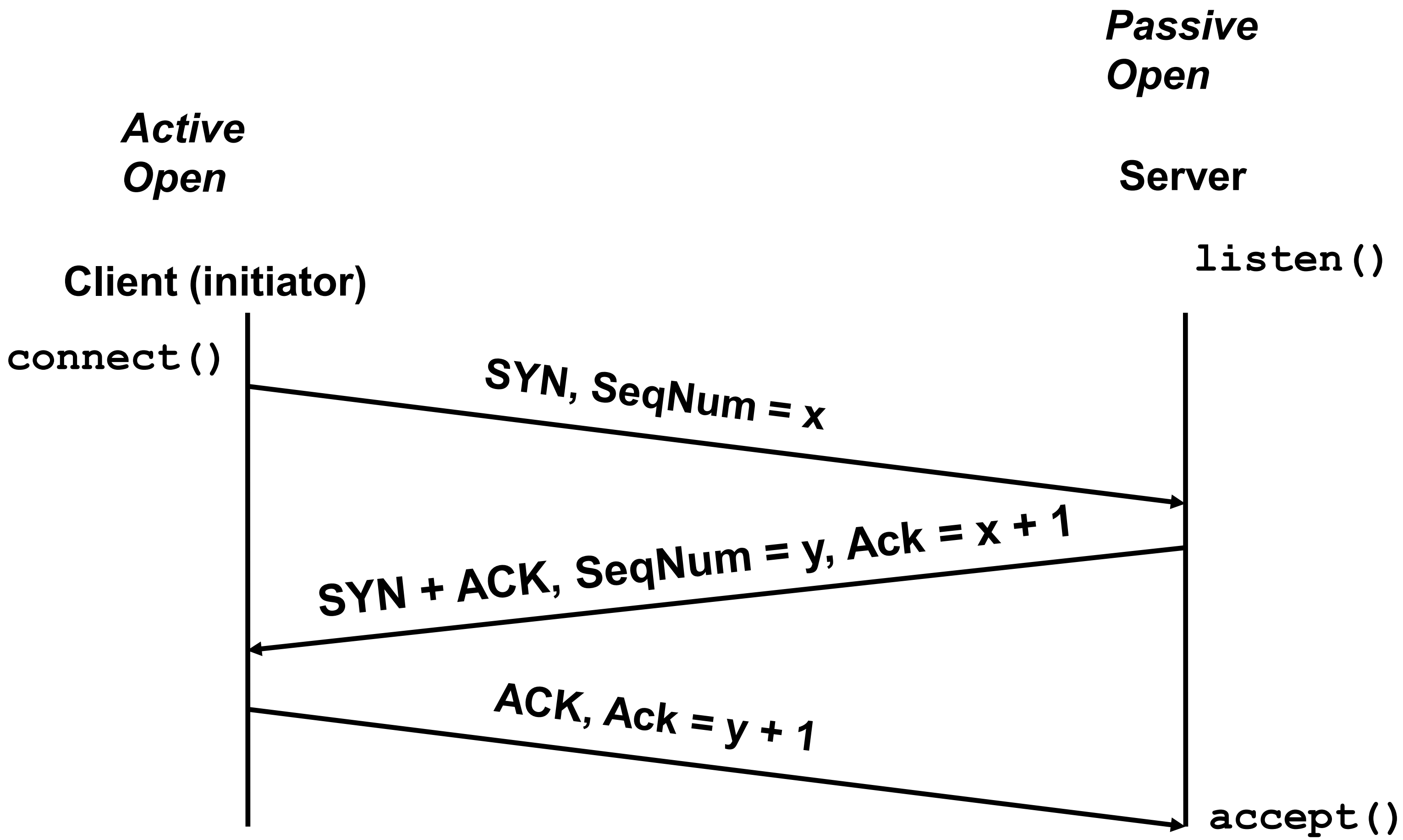


Sliding Window

- Allow a larger amount of data "in flight"
  - Allow sender to get ahead of the receiver
  - … though not *too far* ahead



Timing Diagram: 3-Way Handshaking



We then looked at Congestion Control
and how it solves three fundamental problems

#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps…
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth “fairly" among flows, without overloading the network

… by combining two key mechanisms

detecting
congestion

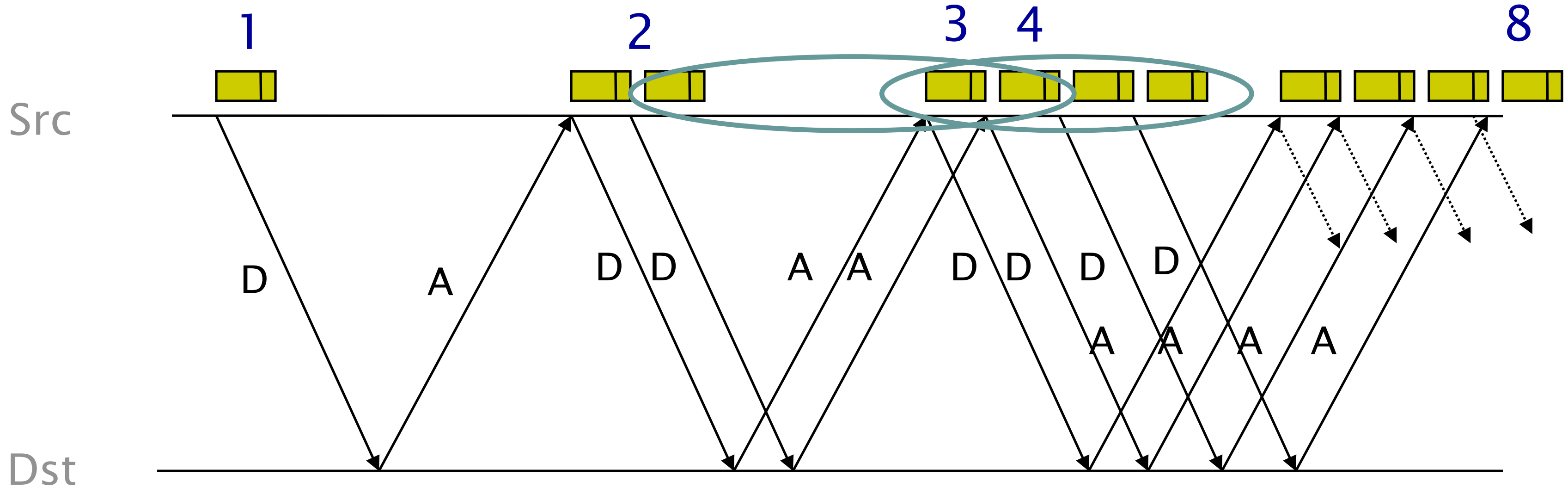
reacting to
congestion

TCP congestion control (almost complete)

```
Initially:
    cwnd = 1
    ssthresh = infinite
New ACK received:
    if (cwnd < ssthresh):
        /* Slow Start*/
        cwnd = cwnd + 1
    else:
        /* Congestion Avoidance
*/
        cwnd = cwnd + 1/cwnd
    dup_ack = 0
Timeout:
    /* Multiplicative decrease */
    ssthresh = cwnd/2
    cwnd = 1
```

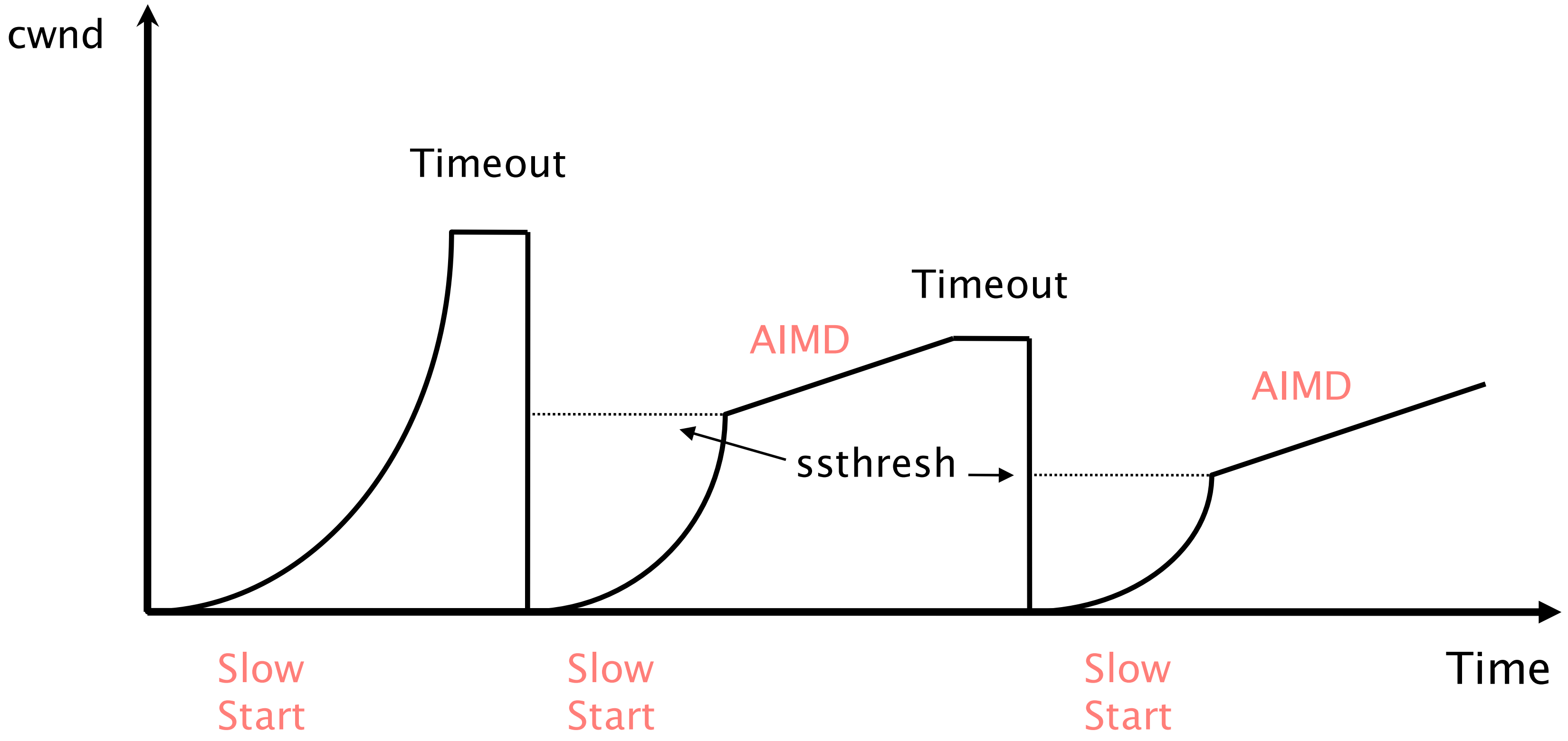
```
Duplicate ACKs received:
    dup_ack ++;
    if (dup_ack >= 3):
        /* Fast Recovery */
        ssthresh = cwnd/2
        cwnd = ssthresh
```

This increase phase, known as slow start,
corresponds to an… exponential increase of CWND!



slow start is called like this only because of starting point

The congestion window of a TCP session typically undergoes multiple cycles of slow-start/AIMD



We then looked at

what's running on top of all this …

DNS

Web

google.ch ⟷ 172.217.16.131

http://www.google.ch

We then looked at
what’s running on top of all this …

Video Streaming

HTTP-based

E-mail

MX, SMTP, POP, IMAP

Your final grade

Exam

written, open book

18.11.2021 (repeat exam for 20/21 students; written >10 Pax)
02.02.2022
03.03.2022

Proseminar

continuous performance assessments

1st exercise + quiz questions, CTF
lab sessions + 10% bonus for class input (challenges)
1 missed session

Final Exam

- 02.02.2022 – 10:15h
- Online exam
- Duration **120 min**
- Open book
- Individual problem solving
- No (electronic) communication allowed
- Random spot checks

- Register online via lfuonline

Allowed are…

- All written notes are allowed (open book).
- The exam must be completed in person.
- All forms of communication during the exam is prohibited.
- (Interactive) calculators on your computer are ok…

- Regulations concerning exams/plagiarism of UIBK apply.

During the Exam

- Individual start of the exam in OLAT
- 120 min duration + 5 min grace period to start up exam
- Questions are answered linearly
- Questions have variable number of points. 1 point is approximately 1 minute
- Possibility to ask technical questions concerning OLAT via zoom/matrix chat
- You may choose to scan & submit your notes after completion of the exam via OLAT.

Questions With Data Input

- Some questions require textual/numerical input. Typically, a syntax/rounding/unit/prefix information is given for each question, otherwise the input are integer numbers without units.

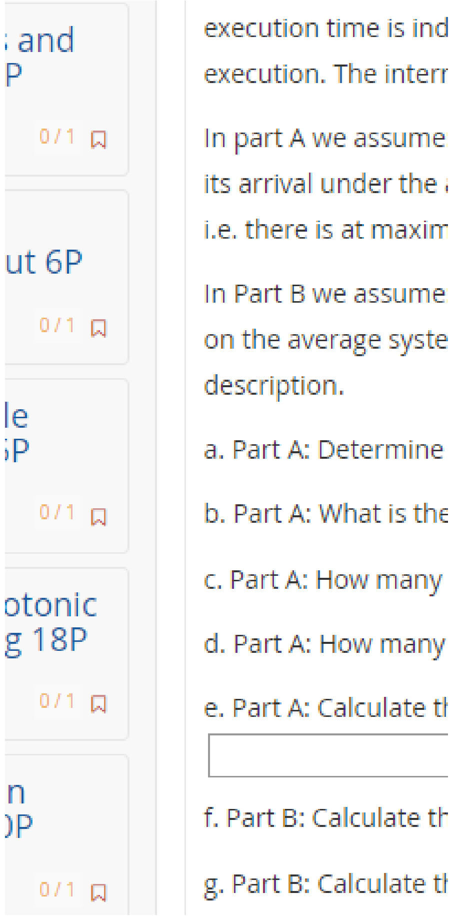
- Wrong syntax, e.g., "3 byte" instead of "3" are counted wrong on all problems where the syntax is specified.

- Units and 10E notation are not entered, e.g. 10 microsecond is entered as 10, not as 0.00001 and not as 10E-6 if microseconds are asked for.

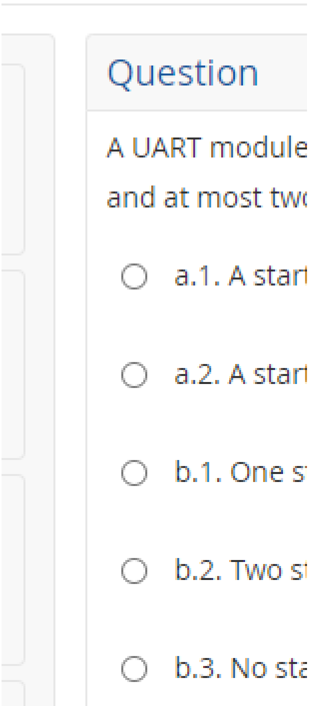
- Examples
  - Enter the hex value with the prefix 0x and separating underscores using lower case alphanumerical characters, e.g., enter 0xff00_0000 for a hex value of ff000000.
  - Round the value to three decimal places.
  - Be sure to only input a single integer number for each answer.

Points are noted on the sidebar

Some questions have an A and B part scenario



Some multiple/single choice questions have funny numbering



After you finish you can upload your scanned notes