Communication Networks

Spring 2022

Tobias Bühler

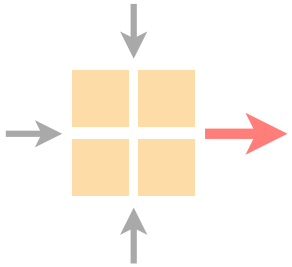
https://comm-net.ethz.ch/

ETH Zürich

March 17 2022

Communication Networks

Exercise 3



General information

Last week’s exercise

Important lecture topics

Introduction to this week’s exercise

Time to solve the exercise

Group registration is open

Please register your groups for the routing project:
https://comm-net.ethz.ch/registration/php/index.php

Use the #group_search channel on Slack
if you look for other group members

Let us know via Slack or email if there are any problems

GitLab setup

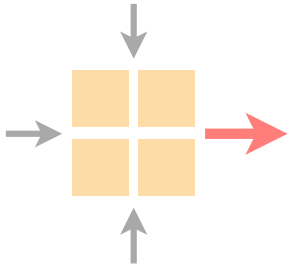
We will give each group a GitLab repository

If you did not already use the ETH GitLab in the past,
please log in once: https://gitlab.ethz.ch

More information to the GitLab usage once the project starts

Communication Networks

Exercise 3



General information

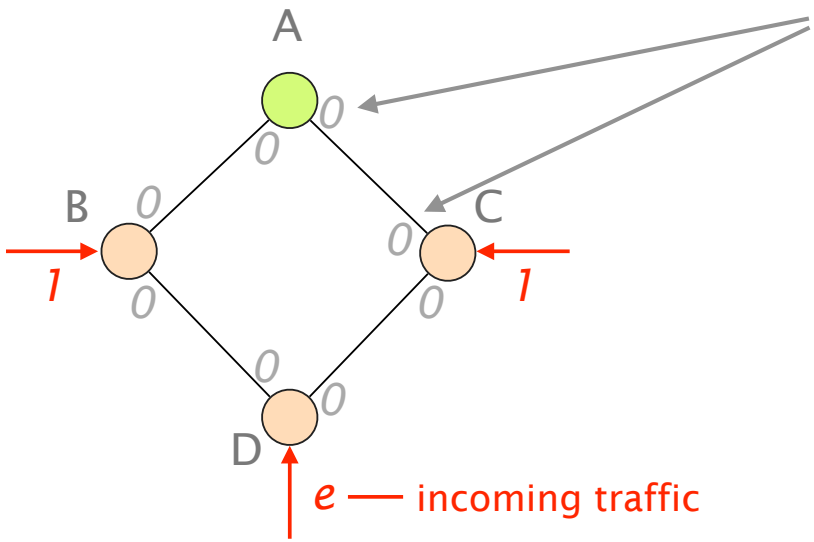
Last week’s exercise

Important lecture topics

Introduction to this week’s exercise

Time to solve the exercise

Task 2.2 - Changing Weights



Network topology with directional link weights.

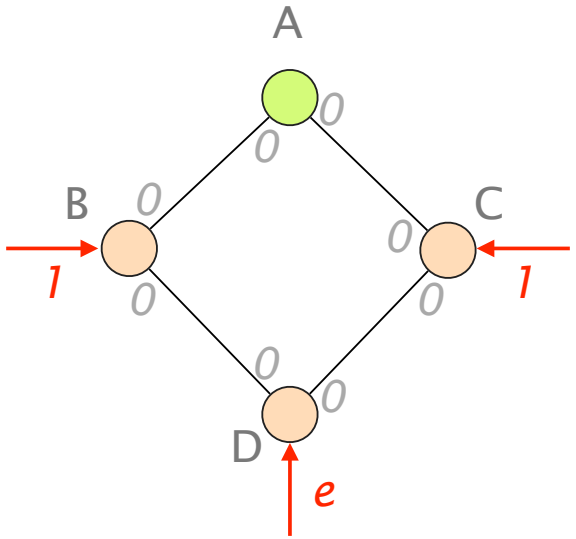
Link weight from A to C can be different than from C to A

Weights are dynamic and always represent the link load

Tie-breaking: path with lower (alphabetically) next hop

e is much bigger than 1

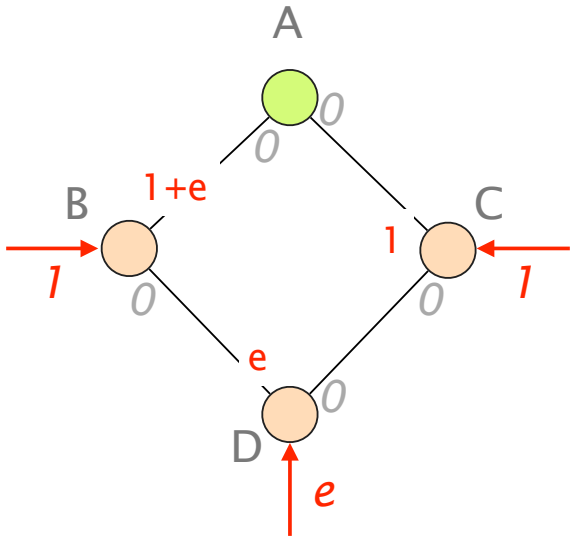
Task 2.2 - Changing Weights



	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0

Next Hop		
B	C	D
A	A	B

Task 2.2 - Changing Weights

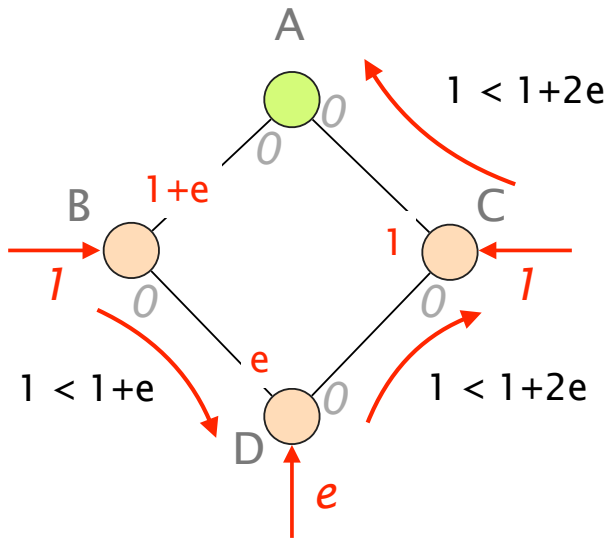


First, find the current next hops

	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0

Next Hop		
B	C	D
A	A	B

Task 2.2 - Changing Weights

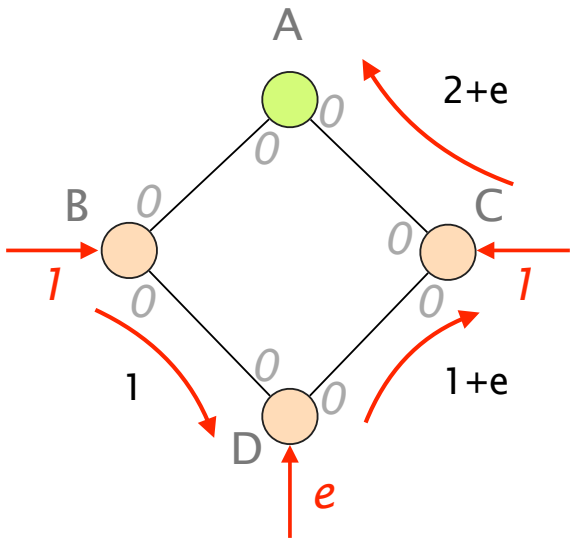


First, find the current next hops

	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0

Next Hop		
B	C	D
A	A	B
D	A	C

Task 2.2 - Changing Weights

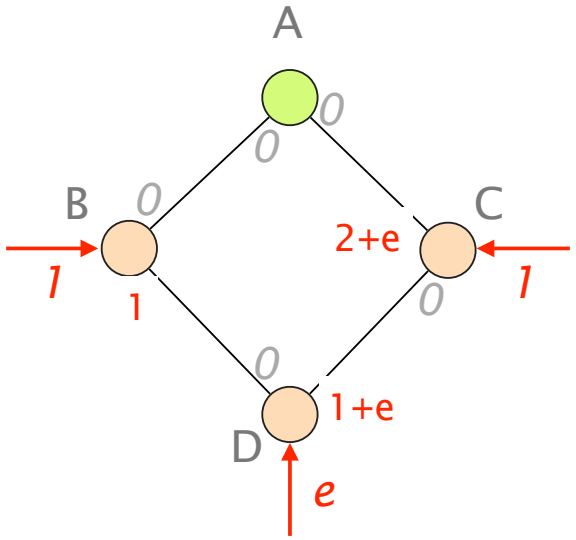


Then update the current link loads

	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0
2	0	0	0	1	2 + e	0	0	1 + e

Next Hop		
B	C	D
A	A	B
D	A	C

Task 2.2 - Changing Weights

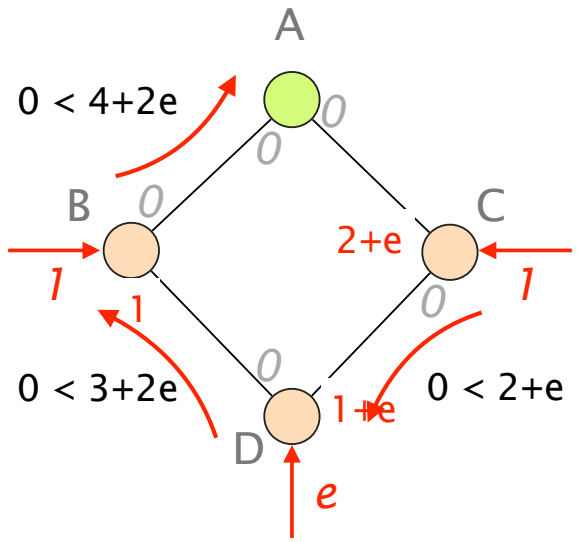


Once again find the next hops

	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0
2	0	0	0	1	2 + e	0	0	1 + e

Next Hop		
B	C	D
A	A	B
D	A	C

Task 2.2 - Changing Weights

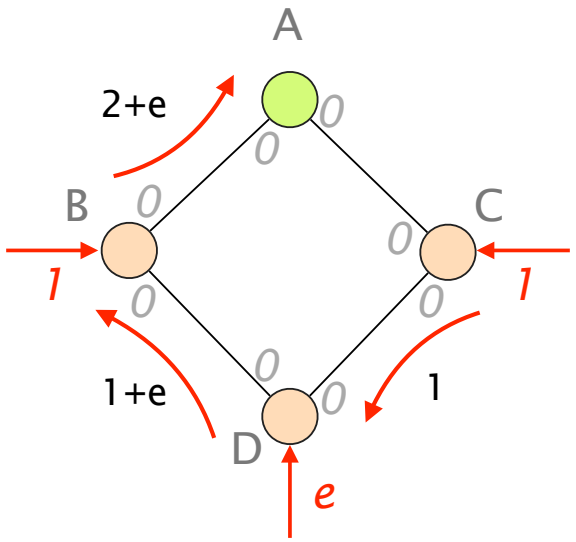


Once again find the next hops

	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0
2	0	0	0	1	2 + e	0	0	1 + e

Next Hop		
B	C	D
A	A	B
D	A	C
A	D	B

Task 2.2 - Changing Weights

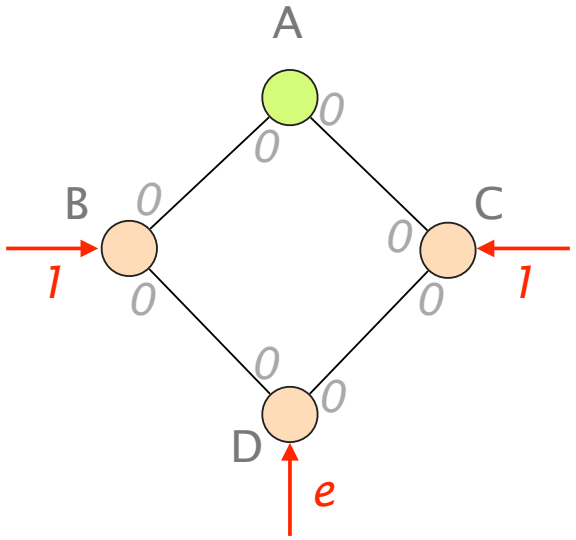


Update the link loads

	Link Load							
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C
0	0	0	0	0	0	0	0	0
1	0	0	1 + e	0	1	0	e	0
2	0	0	0	1	2 + e	0	0	1 + e
3	0	0	2 + e	0	0	1	1 + e	0

Next Hop		
B	C	D
A	A	B
D	A	C
A	D	B

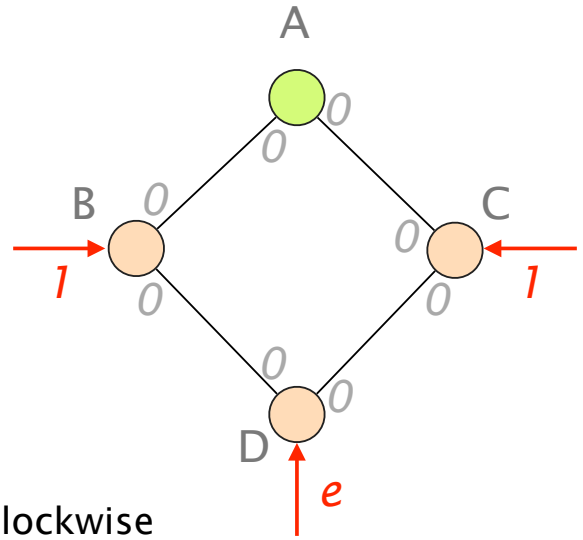
Task 2.2 - Changing Weights



Continue in the same way

	Link Load								Next Hop		
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C	B	C	D
0	0	0	0	0	0	0	0	0	A	A	B
1	0	0	1 + e	0	1	0	e	0	D	A	C
2	0	0	0	1	2 + e	0	0	1 + e	A	D	B
3	0	0	2 + e	0	0	1	1 + e	0	D	A	C
4	0	0	0	1	2 + e	0	0	1 + e	A	D	B

Task 2.2 - Changing Weights



We already see the oscillation

In general, such a frequent weight adaptation is way too sensitive

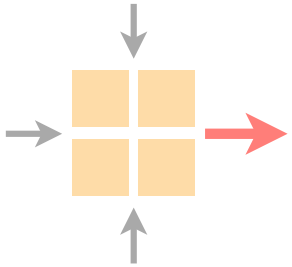
All traffic anti-clockwise

All traffic clockwise

	Link Load								Next Hop		
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C	B	C	D
0	0	0	0	0	0	0	0	0	A	A	B
1	0	0	1 + e	0	1	0	e	0	D	A	C
2	0	0	0	1	2 + e	0	0	1 + e	A	D	B
3	0	0	2 + e	0	0	1	1 + e	0	D	A	C
4	0	0	0	1	2 + e	0	0	1 + e	A	D	B

Communication Networks

Exercise 3



General information

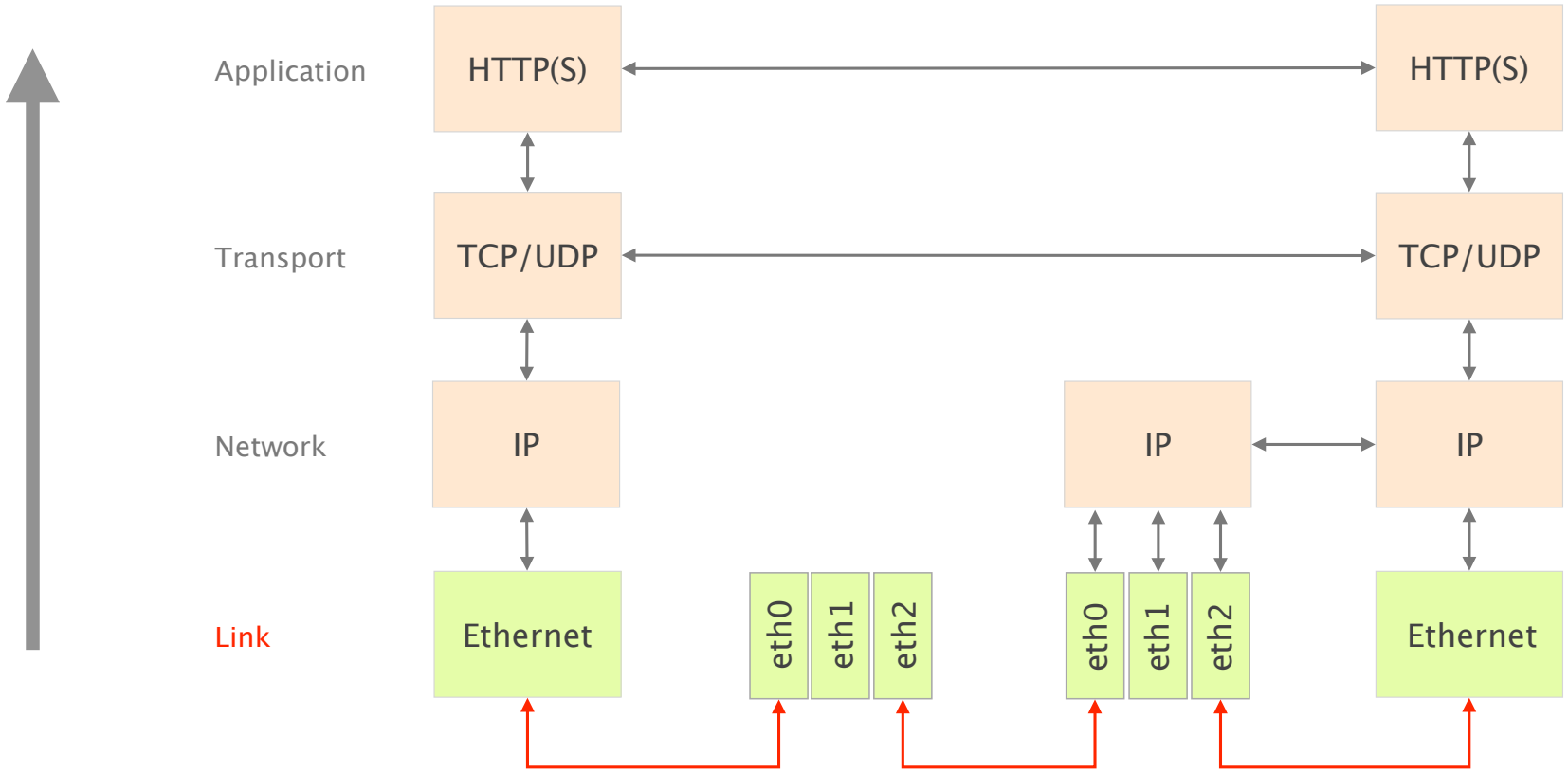
Last week's exercise

Important lecture topics

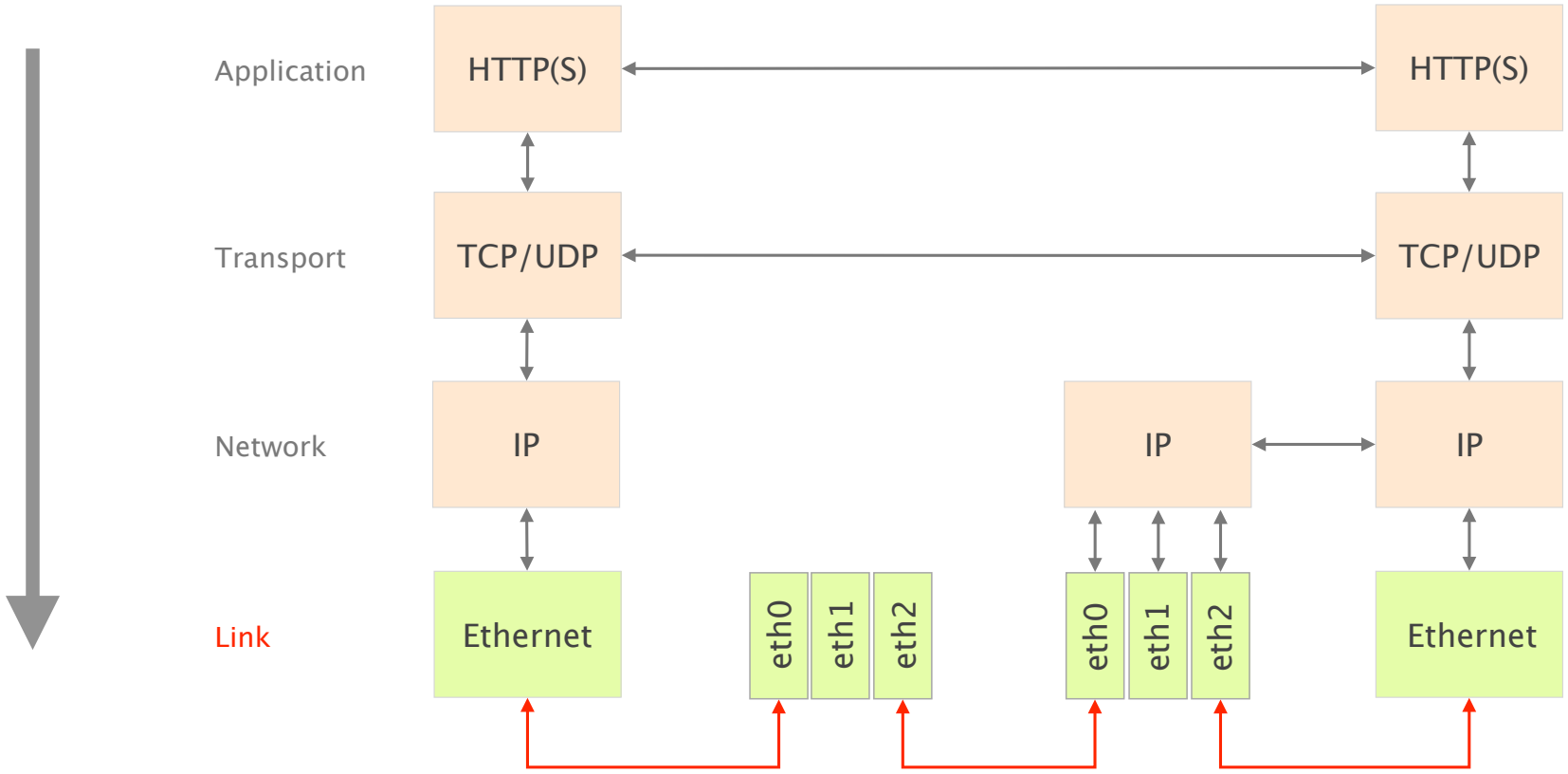
Introduction to this week's exercise

Time to solve the exercise

In the lecture we go through the layers bottom-up



Another possible approach would be top-down



We face a common problem

No matter the direction, often concepts of other layers are needed to understand the current one

Unfortunately, we cannot prevent that completely

We saw that when speaking about MAC addresses, suddenly we also care about IP addresses

MAC addresses identify sender and receiver adapters

MAC addresses identify sender and receiver adapters

used on a single link

MAC addresses identify sender and receiver adapters

used on a single link

In general, we therefore use IP addresses (L3)
to address arbitrary hosts

MAC addresses are then used on a hop-by-hop basis
to eventually reach the corresponding host

In fact, for humans domain names are even easier to remember

domain name → DNS (L5) → IP (L3) → ARP → MAC (L2)

of destination — of destination — of next hop

We currently only consider IP addresses which are reachable over a given link

That simplifies the whole process, we only need to be able to translate from IP to MAC address

IP (L3) of destination → ARP → MAC (L2) of destination

Who are you?

IP-to-MAC binding

Given an IP address reachable on a link, How do I find out what MAC to use?

Address Resolution Protocol

That can only work if hosts can get an IP address

Who am I?

MAC-to-IP binding

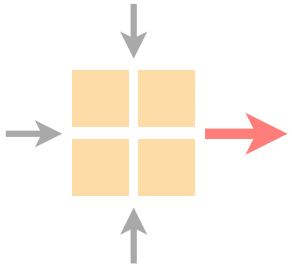
How do I acquire an IP address?

Dynamic Host Configuration Protocol

We will explore both concepts
(ARP and DHCP) in today exercise

Communication Networks

Exercise 3



General information

Last week's exercise

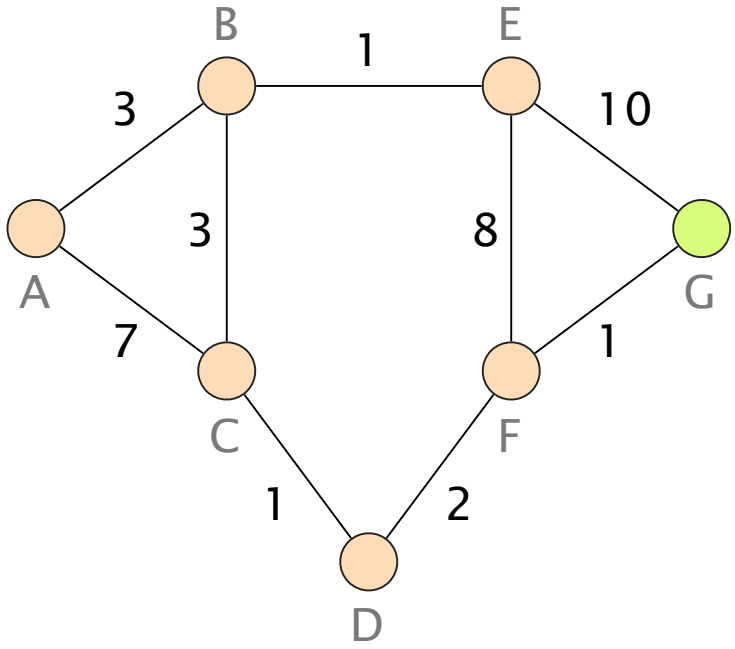
Important lecture topics

Introduction to this week's exercise

Time to solve the exercise

Two more questions related to routing concepts

Task 3.1 Distance Vector

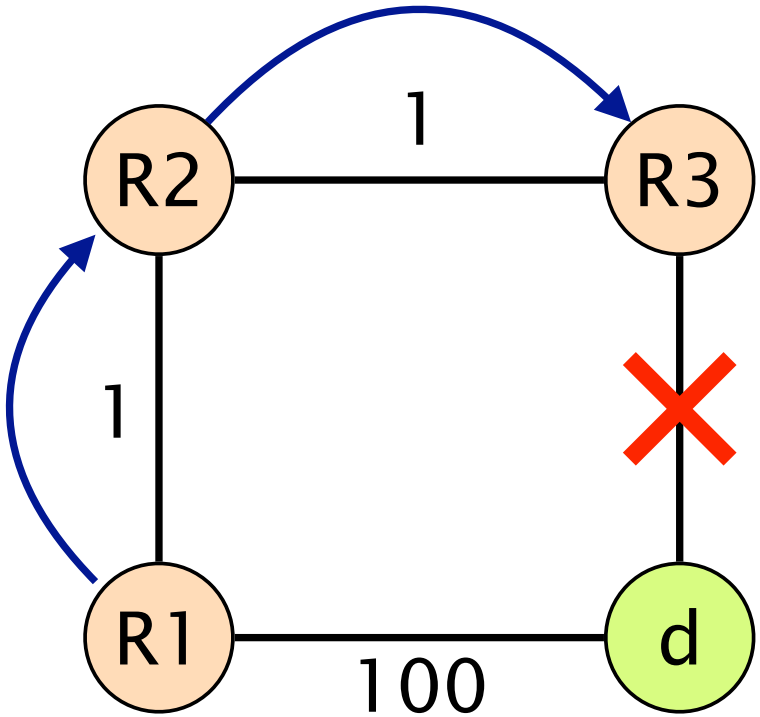


Compute shortest-paths using a distance vector algorithm

Tie-breaking: path with lower amount of links

Compared to link-state algorithms, paths are now computed in a distributed fashion

Task 3.2 Dijkstra's Algorithm with Link Failure



Back to Dijkstra (link-state)

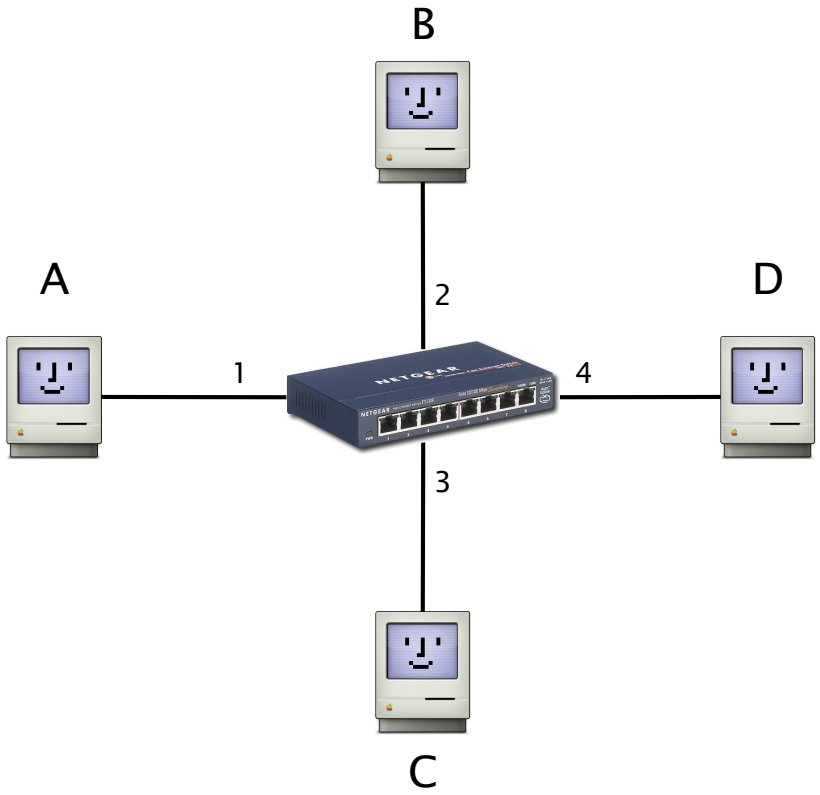
We assume that the link between d and R3 fails

R3 detects that quickly but what about the other nodes?

What happens if the local network view does not match with the reality?

And three questions related to Ethernet & Switching

Task 3.3: Duplicate MAC Address

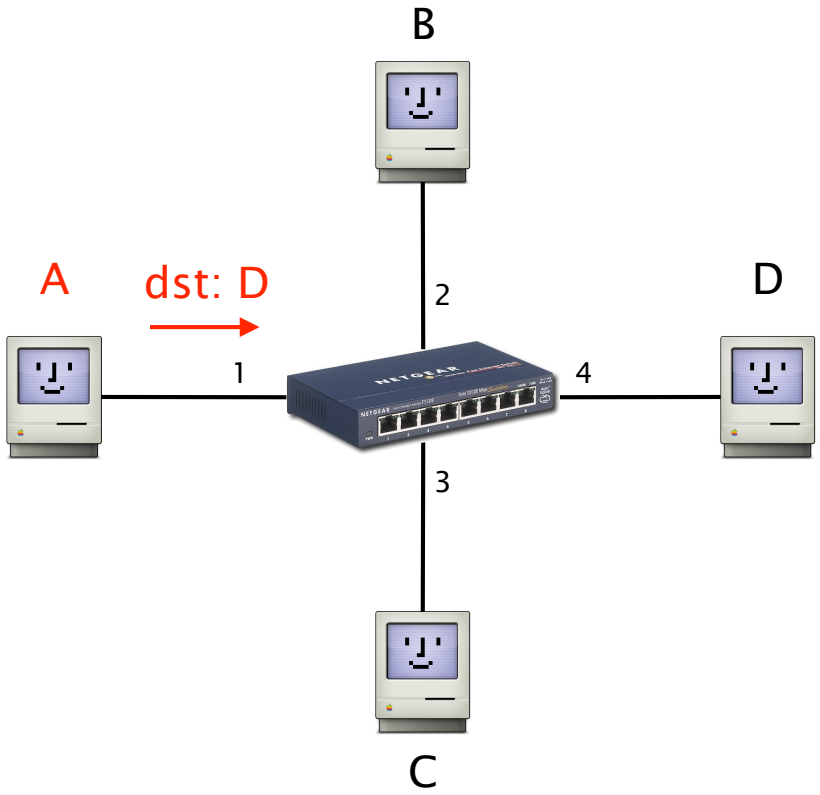


dst	port

As a reminder, let's look at this simple example

A switch learns how to map MACs to ports

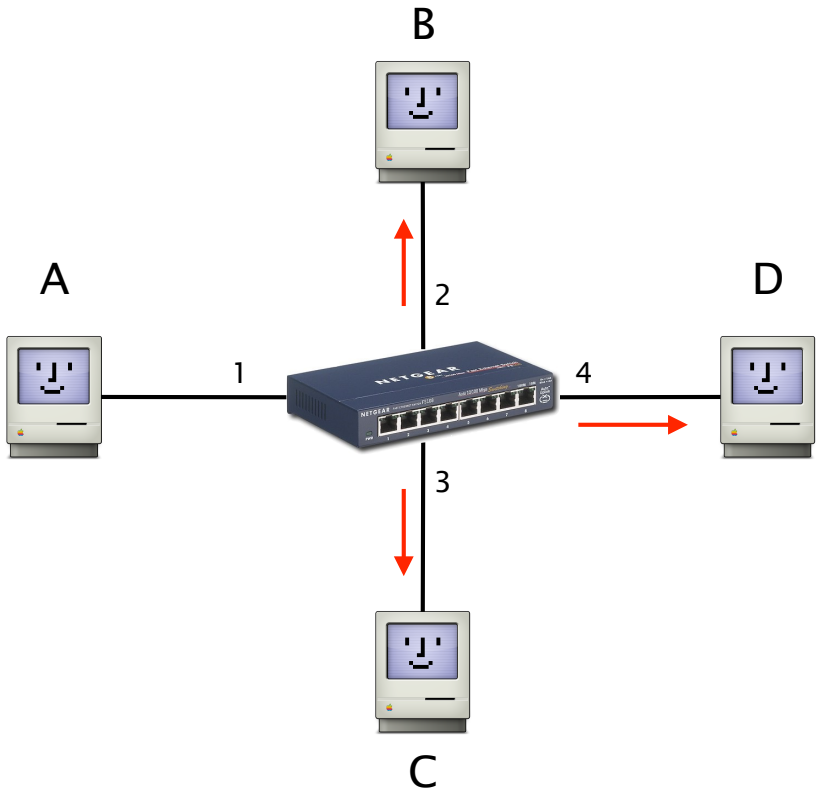
Task 3.3: Duplicate MAC Address



dst	port
A	1

Switch learns how to map A to port 1

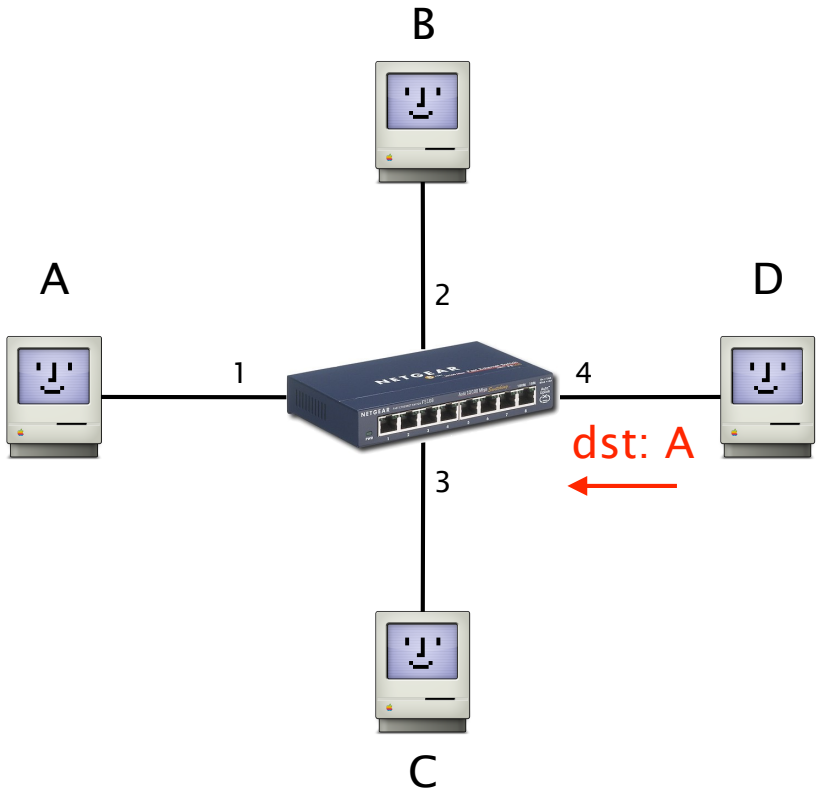
Task 3.3: Duplicate MAC Address



dst	port
A	1

Dst D unknown: broadcast

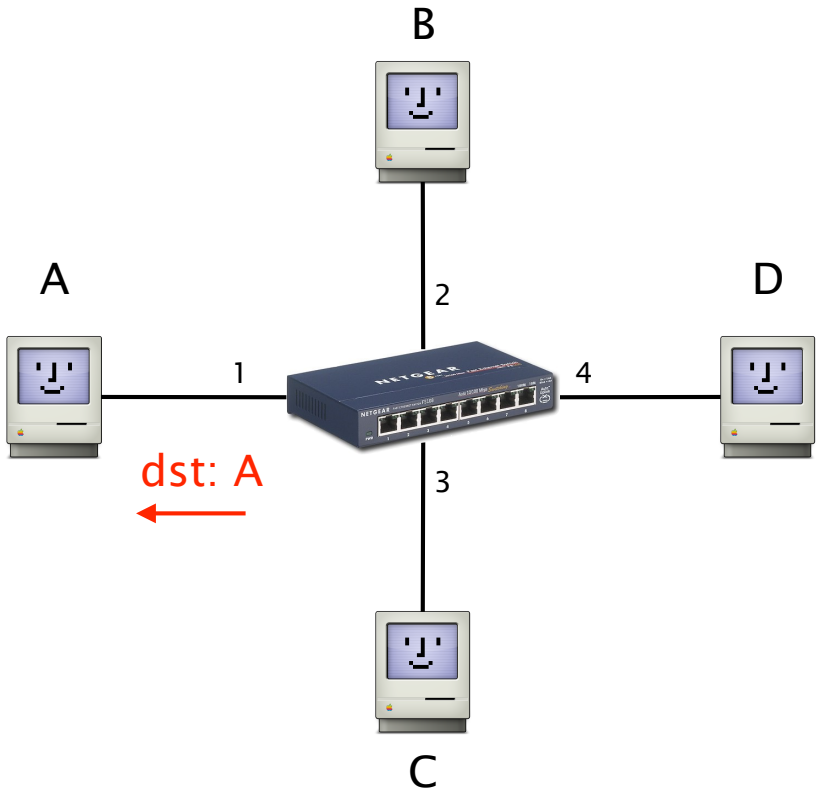
Task 3.3: Duplicate MAC Address



dst	port
A	1
D	4

Switch learns how to map D to port 4

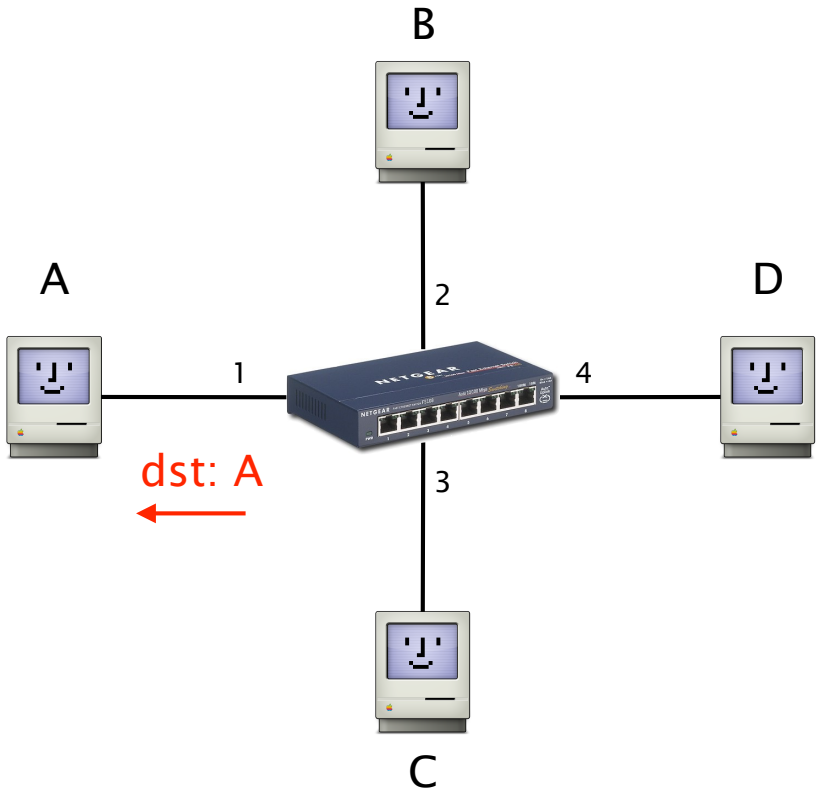
Task 3.3: Duplicate MAC Address



dst	port
A	1
D	4

Dst A known, no broadcast required

Task 3.3: Duplicate MAC Address



dst	port
A	1
D	4

What happens if you have duplicated MAC addresses?

Task 3.4: Imposter

Put your knowledge about DHCP and ARP together

Who am I?	How do I acquire an IP address?
MAC-to-IP binding	Dynamic Host Configuration Protocol (DHCP)
Who are you?	Given an IP address reachable on a link,
IP-to-MAC binding	how do I find out what MAC to use?
	Address Resolution Protocol (ARP)

Task 3.5: MAC-Learning (exam question 2021)

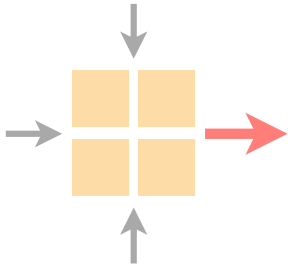
We asked this question in the summer exam of 2021

You find the full exam (as well as other ones) on the website

Use your knowledge from task 3.3 to solve this one

Communication Networks

Exercise 3



General information

Last week’s exercise

Important lecture topics

Introduction to this week’s exercise

Time to solve the exercise