Communication Networks

Spring 2020

Tobias Bühler

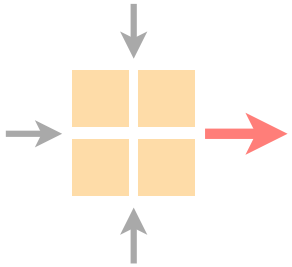
https://comm-net.ethz.ch/

ETH Zürich

May 7 2020

Communication Networks

Exercise 10



Transport project

Overview current assignment

Old exam questions

Solutions will be published next week

Soon at the halfway point

Deadline: **May 22 2020**

We will organize an additional Q&A session
next Monday (11.05.2020) at 3-4pm (voice chat available)

To transfer your files to/from the VM

Use scp (from your local machine):

```
scp -P 3000+X sender.py root@snowball.ethz.ch:.
```

Push and pull from your private GitLab repository

Use editors such as VS Code or PyCharm
which allow you to modify files remotely

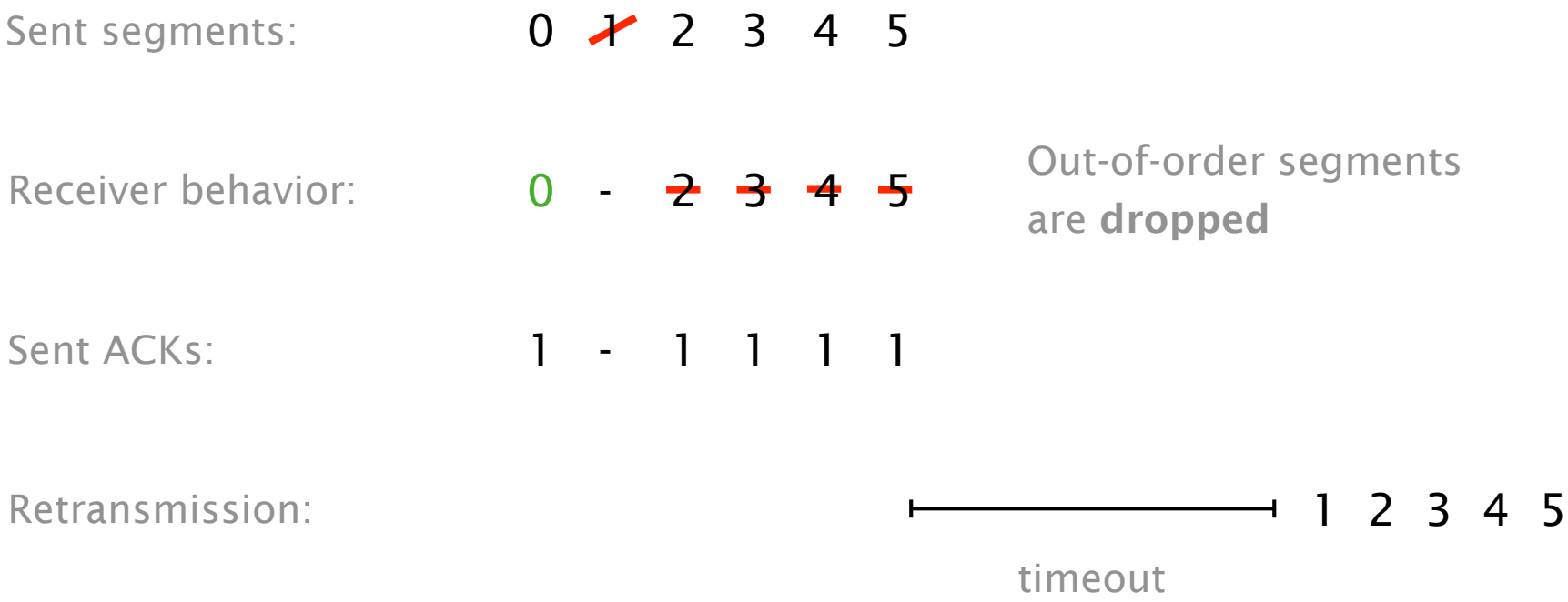
Sequence number overflow

NBITS	controls the maximum sequence number
maximum	assuming NBITS=3: $2^{NBITS} - 1 = 7$
overflow	… 5, 6, 7, 0, 1, 2, …
application examples	ACK number, SACK header blocks, retransmission, …

The Go-Back-N sender waits for a timeout before segments are retransmitted

Sent segments:	0 ~~1~~ 2 3 4 5	
Receiver behavior:	0 - ~~2~~ ~~3~~ ~~4~~ ~~5~~	Out-of-order segments are **dropped**
Sent ACKs:	1 - 1 1 1 1	
Retransmission:		

The Go-Back-N sender waits for a timeout before segments are retransmitted

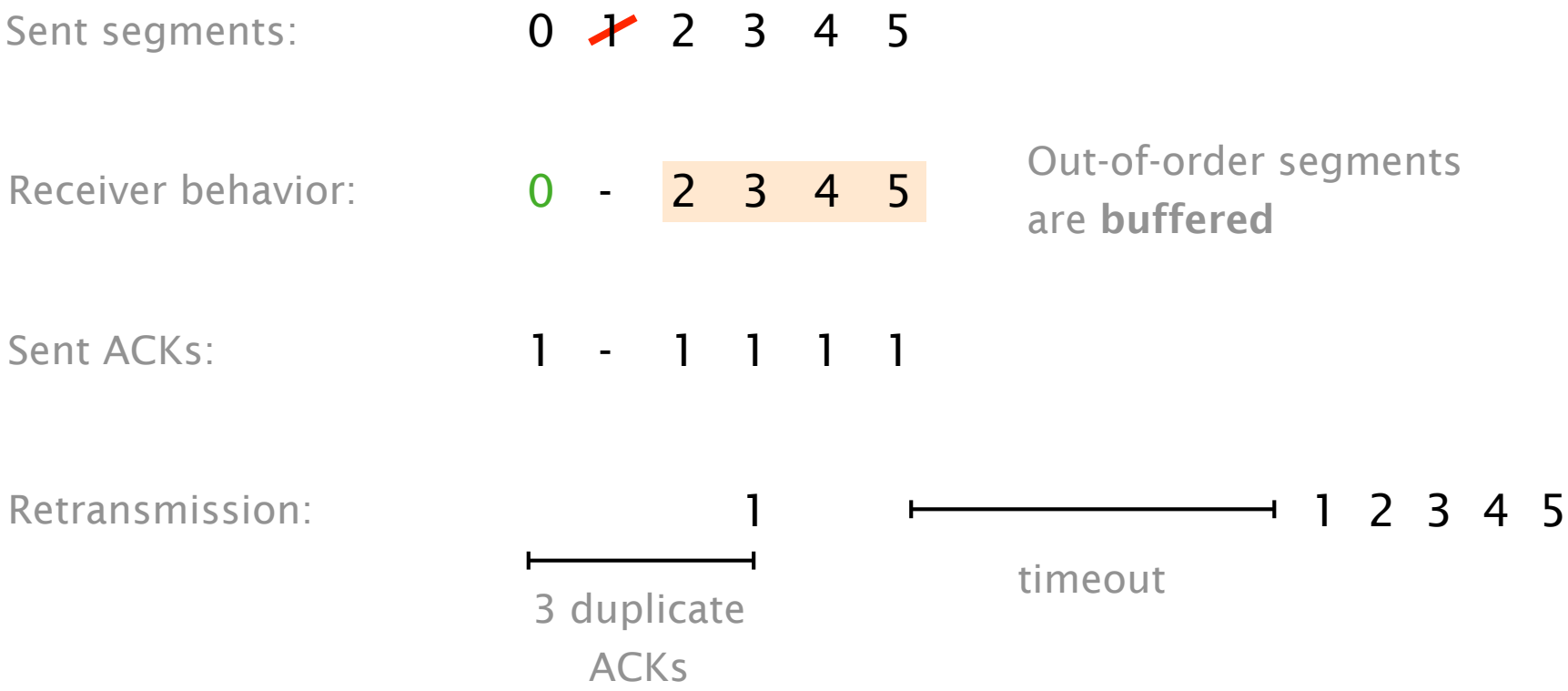


Selective Repeat can increase the performance

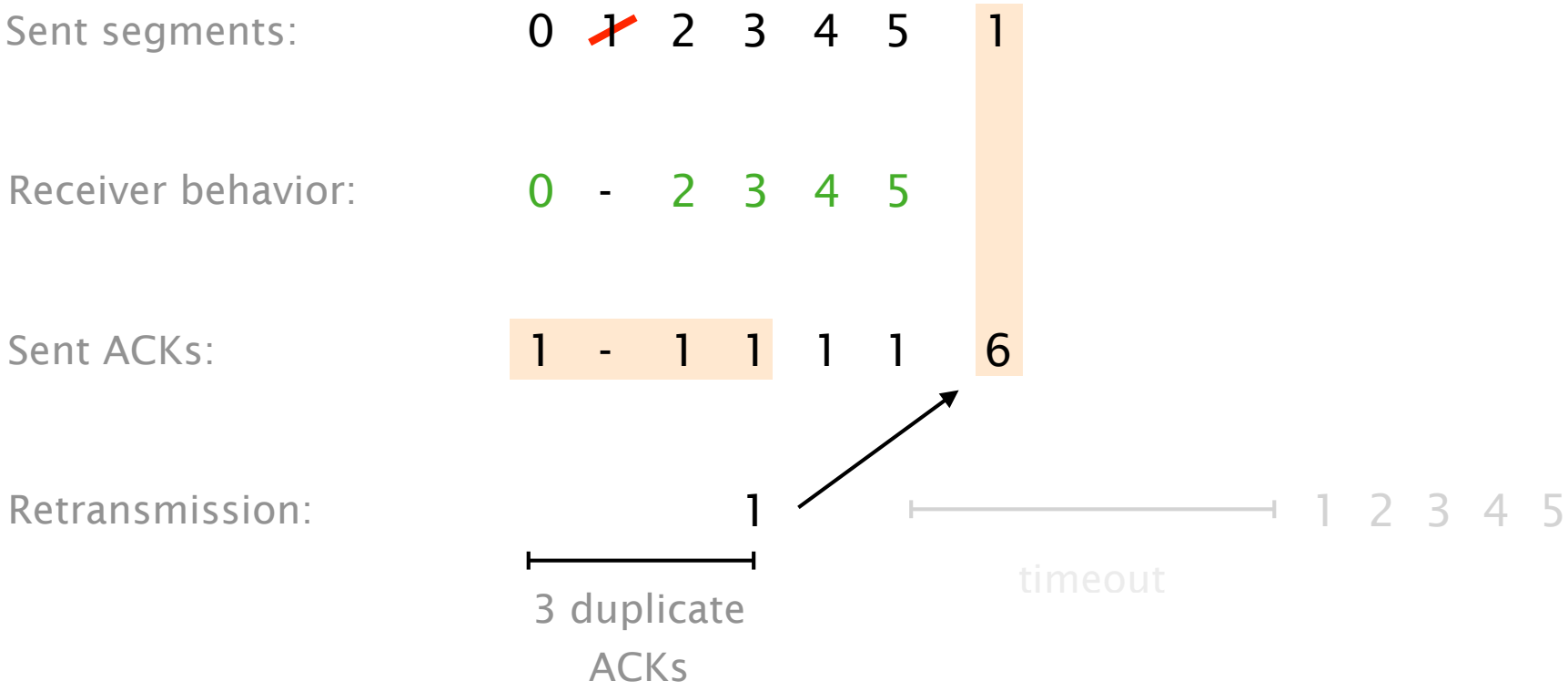
Sent segments:	0	~~1~~	2	3	4	5
Receiver behavior:	0	-	2	3	4	5
Sent ACKs:	1	-	1	1	1	1
Retransmission:						

Out-of-order segments are **buffered**

Selective Repeat can increase the performance

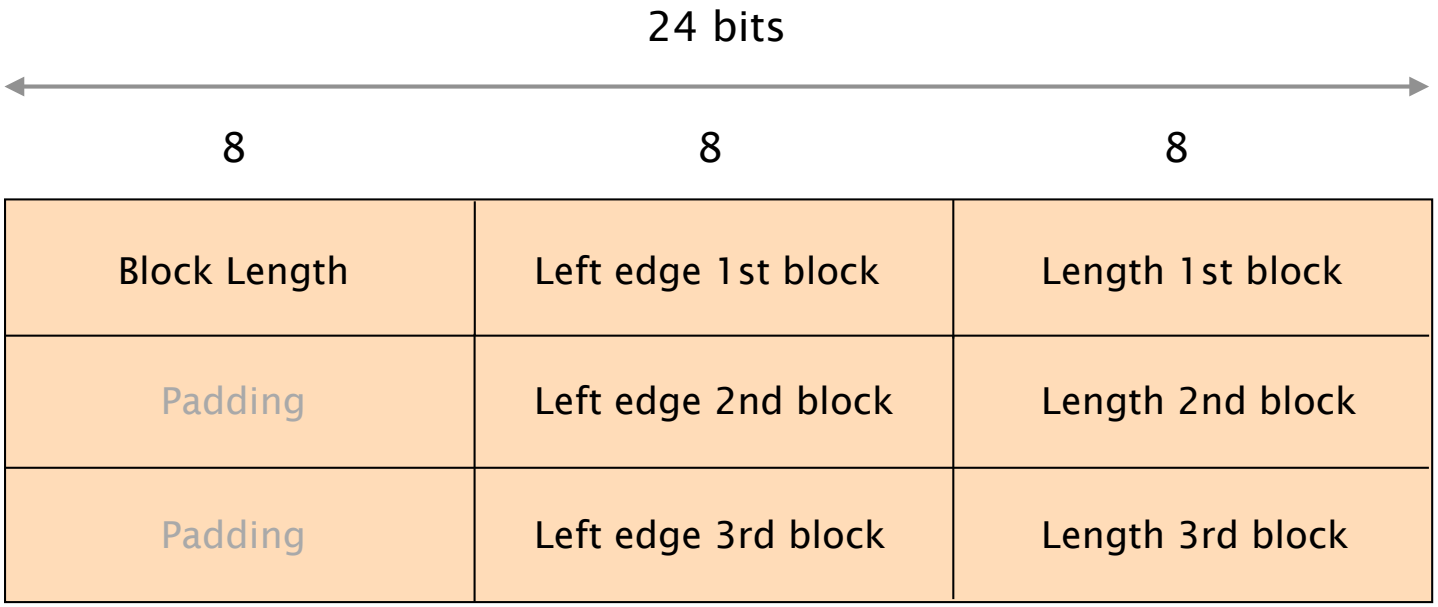


Selective Repeat can increase the performance



For SACK we need an optional header

Optional header



Maximal 3 SACK blocks in the optional header

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header:

SACK header:

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header: ACK number: 3

SACK header:

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header: ACK number: 3

SACK header:

#blocks	start b1	size b1
Padding	start b2	size b2
Padding	start b3	size b3

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header: ACK number: 3

SACK header:

#blocks	4	2
Padding	start b2	size b2
Padding	start b3	size b3

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header: ACK number: 3

SACK header:

#blocks	4	2
Padding	8	1
Padding	start b3	size b3

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header: ACK number: 3

SACK header:

#blocks	4	2
Padding	8	1
Padding	10	4

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, ~~15, 16, 17~~

no space

Mandatory header: ACK number: 3

SACK header:

#blocks	4	2
Padding	8	1
Padding	10	4

SACK example - Receiver

Correctly received segments: 0, 1, 2

Buffered out-of-order segments: 4, 5, 8, 10, 11, 12, 13, 15, 16, 17

Mandatory header: ACK number: 3

SACK header:

3	4	2
Padding	8	1
Padding	10	4

SACK example - Sender

Receiver SACK header:

3	4	2
Padding	8	1
Padding	10	4

ACK number: 3

ACK - block 1:

block 1 - block 2:

block 2 - block 3:

after block 3:

SACK example - Sender

Receiver SACK header:

3	4	2
Padding	8	1
Padding	10	4

ACK number: 3

ACK - block 1: 3

block 1 - block 2:

block 2 - block 3:

after block 3:

SACK example - Sender

Receiver SACK header:

3	4	2
Padding	8	1
Padding	10	4

ACK number: 3

ACK - block 1: 3

block 1 - block 2: 6, 7

block 2 - block 3:

after block 3:

SACK example - Sender

Receiver SACK header:

3	4	2
Padding	8	1
Padding	10	4

ACK number: 3

ACK - block 1: 3

block 1 - block 2: 6, 7

block 2 - block 3: 9

after block 3:

SACK example - Sender

Receiver SACK header:

3	4	2
Padding	8	1
Padding	10	4

ACK number: 3

ACK - block 1: 3

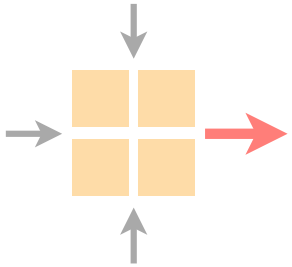
block 1 - block 2: 6, 7

block 2 - block 3: 9

after block 3: no retransmission

Communication Networks

Exercise 10



Transport project

Overview current assignment

Old exam questions

Solutions will be published next week

10.1-10.3 Compare DNS responses from different DNS servers

On Linux/Mac use the `dig` tool

`nslookup` on Windows

Carefully study all the output from the DNS server.
What can e.g., the TTL field tell you?

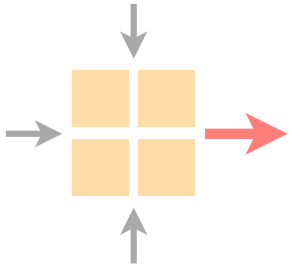
10.4 HTTP host header

How can we host multiple websites on the same server?

Perform a hand-crafted HTTP GET request with `telnet`

Communication Networks

Exercise 10



Transport project

Overview current assignment

Old exam questions

Solutions will be published next week

Exam 2016 - Task 5 b): In naming we trust

Some rather precisely defined open questions

Make sure that you read the questions carefully

Often there are multiple correct answers

Exam 2016 - Task 5 b) - (i): In naming we trust

Explain **two** distinct reasons why many DNS server operators disable recursive queries and only allow iterative queries instead. (2 Points)

Exam 2016 - Task 5 b) - (i): In naming we trust

one

Explain ~~two~~ distinct reasons why many DNS server operators disable recursive queries and only allow iterative queries instead. (2 Points)

(Lecture content changes over the years)

Exam 2016 - Task 5 b) - (i): In naming we trust

one

Explain ~~two~~ distinct reasons why many DNS server operators disable recursive queries and only allow iterative queries instead. (2 Points)

Reason 1: Increase memory and processing requirements on the DNS server for recursive queries.

Reason 2: Security concerns. A recursive DNS server could be used to perform reflection-based DoS attacks.

Exam 2016 - Task 5 b) - (ii): In naming we trust

As we saw in the course, DNS operators often rely on BGP Anycast to distribute the load on multiple servers spread across the Internet. With BGP Anycast, it is possible for different packets (and therefore, requests), sent by the same client (e.g., your laptop at ETH), to reach servers located in different locations. Explain: *(i)* why it is possible; and *(ii)* whether it is a problem or not. (4 Points)

Exam 2016 - Task 5 b) - (ii): In naming we trust

As we saw in the course, DNS operators often rely on BGP Anycast to distribute the load on multiple servers spread across the Internet. With BGP Anycast, it is possible for different packets (and therefore, requests), sent by the same client (e.g., your laptop at ETH), to reach servers located in different locations. Explain: *(i)* why it is possible; and *(ii)* whether it is a problem or not. (4 Points)

(i): BGP Anycast routes to the currently „nearest“ server/destination which can change over time.

(ii): DNS request are UDP-based. UDP is inter-packet state-less and it does not matter if packets from the same host arrive at different servers.

Exam 2016 - Task 5 b) - (iii): In naming we trust

Consider now a CDN which considers replicating static Web content using BGP Anycast. Explain whether having packets from the same client going to distinct replicas would be acceptable or not. (3 Points)

Exam 2016 - Task 5 b) - (iii): In naming we trust

Consider now a CDN which considers replicating static Web content using BGP Anycast. Explain whether having packets from the same client going to distinct replicas would be acceptable or not. (3 Points)

Answer: This is not acceptable. Web content is served through TCP connections which are stateful. Changing the destination within a connection breaks the current connection

Exam 2016 - Task 5 b) - (iii): In naming we trust

Consider now a CDN which considers replicating static Web content using BGP Anycast. Explain whether having packets from the same client going to distinct replicas would be acceptable or not. (3 Points)

Answer: This is not acceptable. Web content is served through TCP connections which are stateful. Changing the destination within a connection breaks the current connection

Bonus question: For some websites it could even be problematic if two consecutive connections go to different servers. Why?

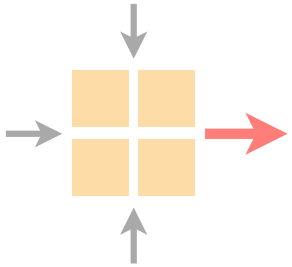
Exam 2016 - Task 5 b) - (iii): In naming we trust

Bonus question: For some websites it could even be problematic if two consecutive connections go to different servers. Why?

Answer: Think about dynamic websites which change based on the user input (i.e., your Facebook page).
What if the new server did not yet get the changes done in the first connection towards the initial server?

Communication Networks

Exercise 10



Transport project

Overview current assignment

Old exam questions

Solutions will be published next week