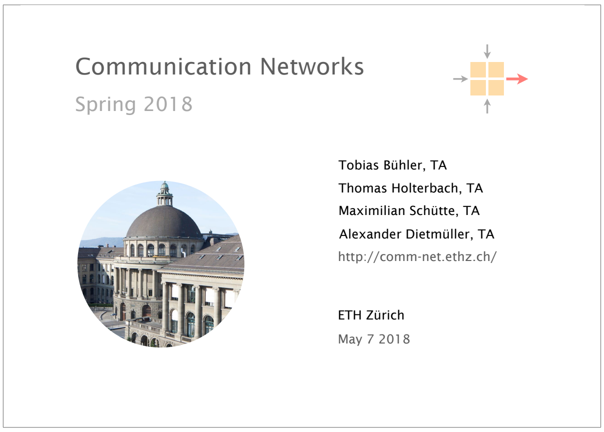
Communication Networks

Prof. Laurent Vanbever

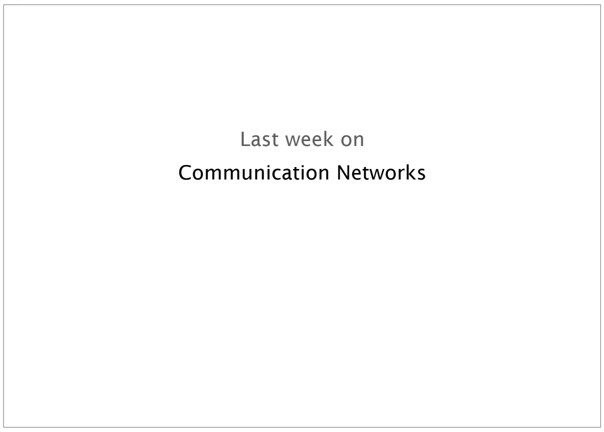


Communication Networks

Spring 2018

Tobias Bühler, TA
Thomas Holterbach, TA
Maximilian Schütte, TA
Alexander Dietmüller, TA
http://comm-net.ethz.ch/

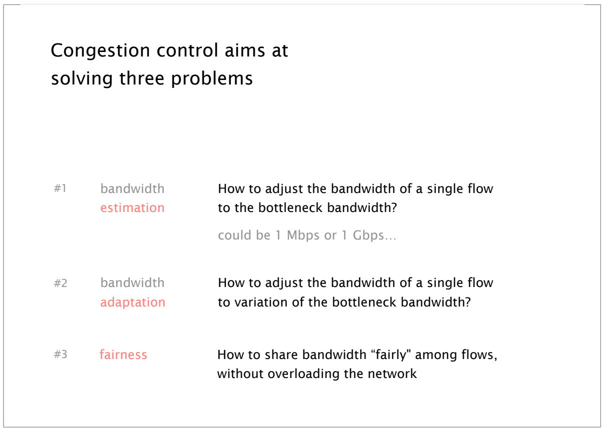
ETH Zürich
May 7 2018



Last week on
Communication Networks

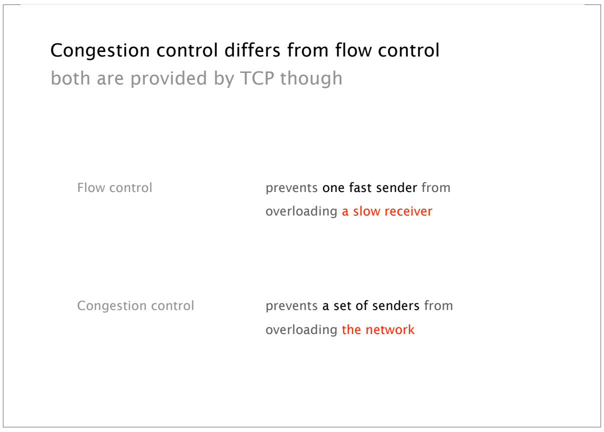
TCP Congestion Control

CONGESTION
AHEAD
EXPECT DELAYS



Congestion control aims at solving three problems

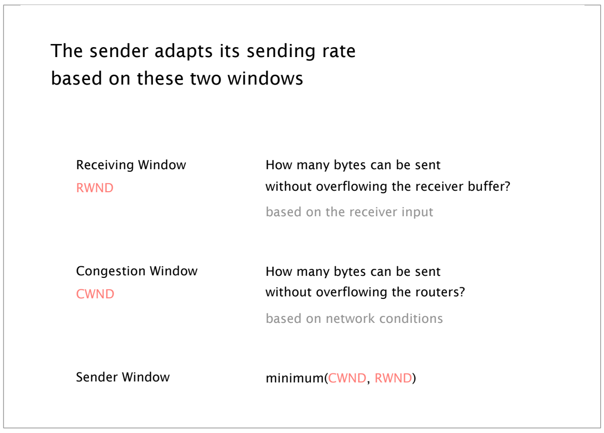
#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps…
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth "fairly" among flows, without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



Congestion control differs from flow control

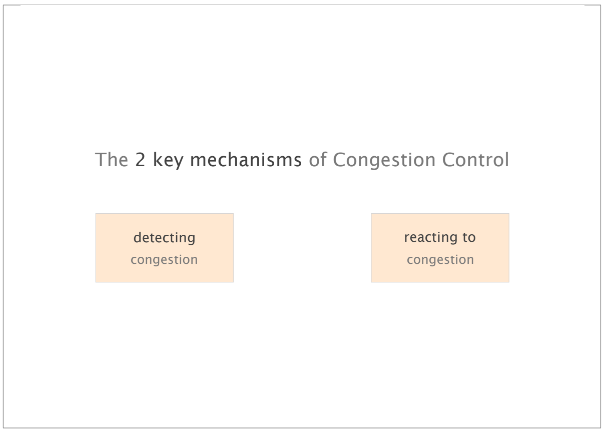
both are provided by TCP though

Flow control	prevents one fast sender from overloading a slow receiver
Congestion control	prevents a set of senders from overloading the network



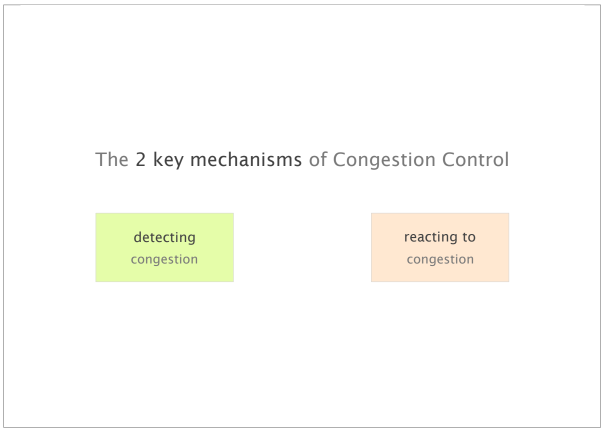
The sender adapts its sending rate based on these two windows

Receiving Window RWND	How many bytes can be sent without overflowing the receiver buffer? based on the receiver input
Congestion Window CWND	How many bytes can be sent without overflowing the routers? based on network conditions
Sender Window	minimum(CWND, RWND)



The 2 key mechanisms of Congestion Control

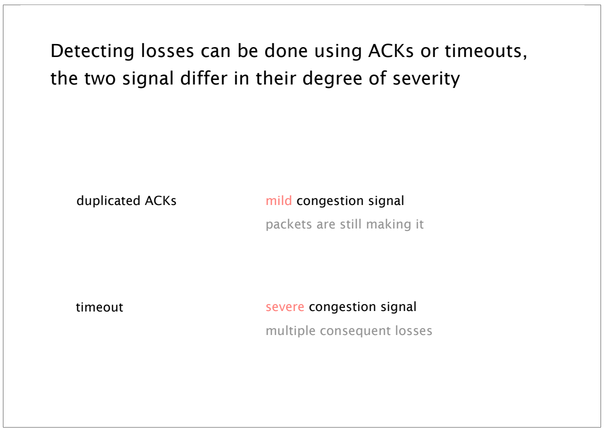
detecting
congestion
reacting to
congestion



The 2 key mechanisms of Congestion Control

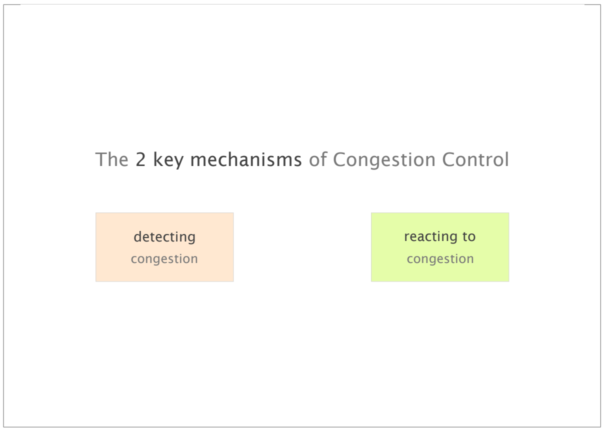
detecting congestion

reacting to congestion



Detecting losses can be done using ACKs or timeouts, the two signal differ in their degree of severity

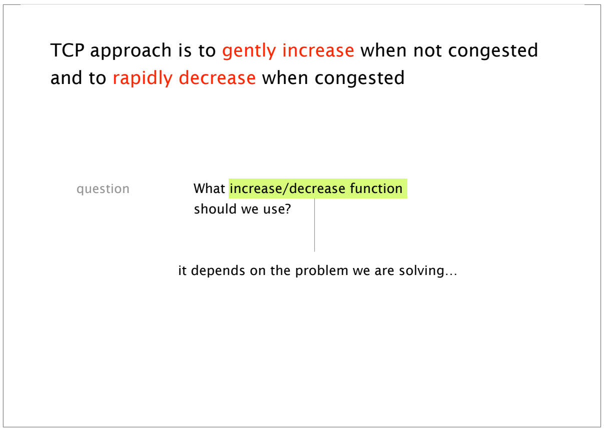
duplicated ACKs	mild congestion signal packets are still making it
timeout	severe congestion signal multiple consequent losses



The 2 key mechanisms of Congestion Control

detecting congestion

reacting to congestion

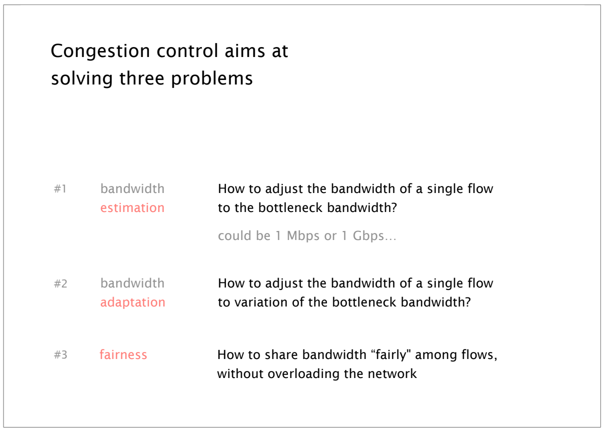


TCP approach is to gently increase when not congested and to rapidly decrease when congested

question

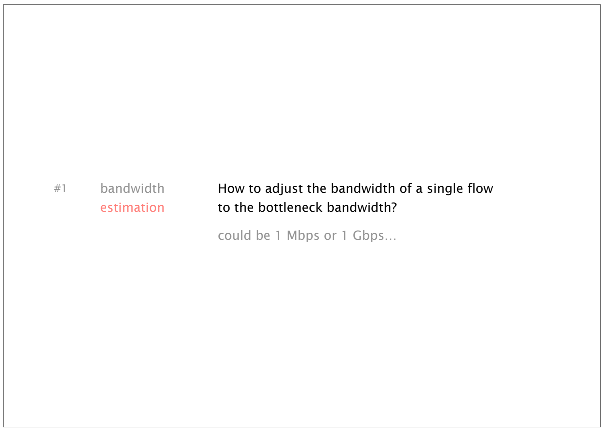
What increase/decrease function should we use?

it depends on the problem we are solving...

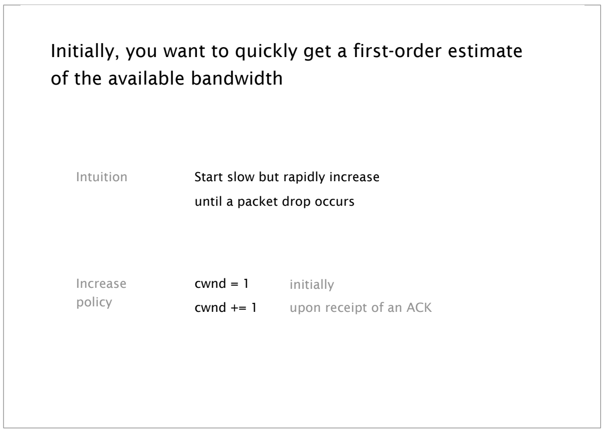


Congestion control aims at solving three problems

#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

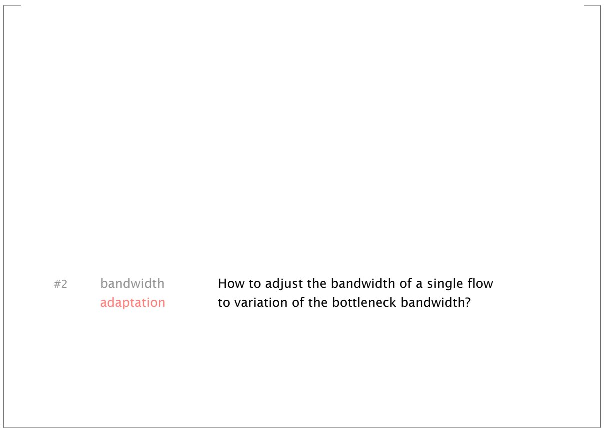


#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps...



Initially, you want to quickly get a first-order estimate of the available bandwidth

Intuition	Start slow but rapidly increase until a packet drop occurs	
Increase policy	cwnd = 1	initially
	cwnd += 1	upon receipt of an ACK

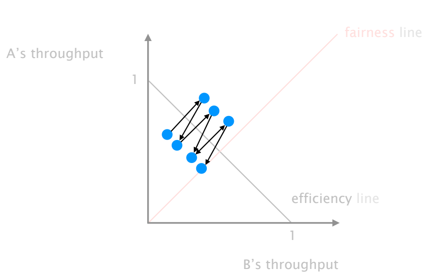


#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?

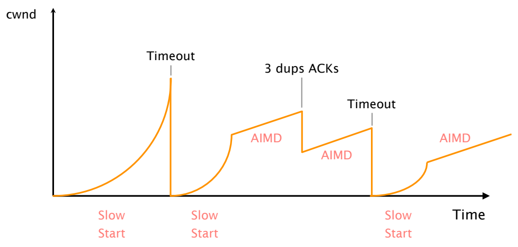
	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

#3 fairness How to share bandwidth "fairly" among flows, without overloading the network

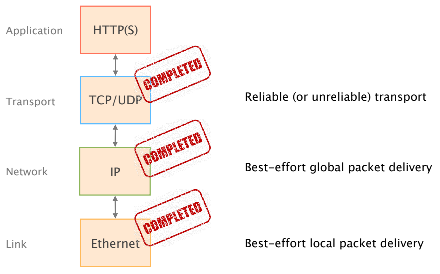
AIMD converge to fairness and efficiency,
it then fluctuates around the optimum (in a stable way)



Congestion control makes TCP throughput
look like a "sawtooth"



We now have completed the transport layer (!)



Application
HTTP(S)
Transport
TCP/UDP
COMPLETED
Network
IP
COMPLETED
Link
Ethernet
COMPLETED
Reliable (or unreliable) transport
Best-effort global packet delivery
Best-effort local packet delivery

This week on
Communication Networks

Routing Project

Recap, demo and final results

Reliable Transport Project

Introduction and demo

Python and Git tutorial

Routing Project

Recap, demo and final results

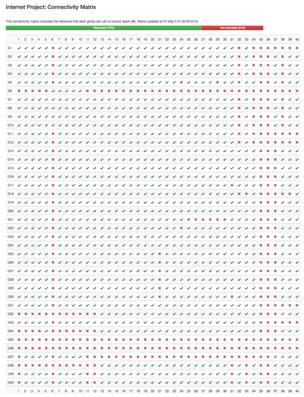
Reliable Transport Project

Introduction and demo

Python and Git tutorial

Communication Networks 2018
Routing Project

Recap



Internet Project: Connectivity Matrix

This connectivity matrix indicates the networks that each group can (✓) or cannot reach (✗). Matrix updated at Fri May 4 01:30:28 2018.

Reachable (76%) Not reachable (24%)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
G1	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
G2	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G3	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G4	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G5	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗	✓	✓
G6	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
G7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓
G11	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
G12	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
G13	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G14	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G15	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G16	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G17	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓
G18	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
G19	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G20	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G21	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G22	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G23	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G24	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G25	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G26	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G27	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G28	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G29	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G30	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓
G31	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
G32	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G33	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
G34	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
G35	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
G36	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
G37	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
G38	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓
G39	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓
G40	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

73%

Proportion of valid BGP paths
in your mini-Internet

*From 15 traceroutes launched
between random pairs of ASes

Your mini-Internet works!

and common services
can run on top of it

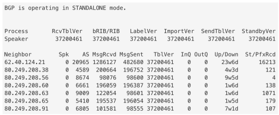
For this project, you basically did what an actual network operator has to do

Including debugging and monitoring your configuration and connectivity

For this project, you basically did what an actual network operator has to do

Including debugging and monitoring your configuration and connectivity

Looking glass



```
BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker        37200461   37200461   37200461   37200461    37200461    37200461

Neighbor        Spk    AS MsgRcvd MsgSent    TblVer  InQ OutQ  Up/Down  St/PfxRcd
62.40.124.21      0 20965 1286127  482680 37200461    0    0    23w6d      16213
80.249.208.38     0  4589  200664  196752 37200461    0    0     4w3d        121
80.249.208.56     0  8674   98076   98600 37200461    0    0     9w5d          4
80.249.208.60     0  6661  196059  196387 37200461    0    0     1w6d        138
80.249.208.63     0  9009  122054   98601 37200461    0    0     1w6d       1071
80.249.208.65     0  5410  195537  196054 37200461    0    0     1w5d        179
80.249.208.91     0  6805  101581   98555 37200461    0    0     7w1d        107
```

Measurement platform

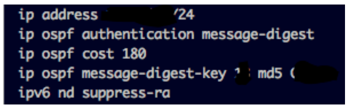
RIPE Atlas

There was often multiple ways
to answer the questions

There was often multiple ways
to answer the questions

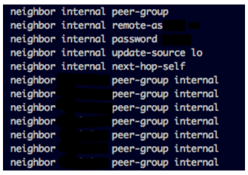
and we found some interesting answers

Enabling authentication in OSPF



```
ip address [redacted]/24
ip ospf authentication message-digest
ip ospf cost 180
ip ospf message-digest-key [redacted] md5 [redacted]
ipv6 nd suppress-ra
```

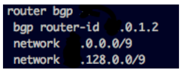
Group BGP neighbors to simplify configuration



```
neighbor internal peer-group
neighbor internal remote-as [redacted]
neighbor internal password [redacted]
neighbor internal update-source lo
neighbor internal next-hop-self
neighbor [redacted] peer-group internal
neighbor [redacted] peer-group internal
neighbor [redacted] peer-group internal
neighbor [redacted] peer-group internal
neighbor [redacted] peer-group internal
neighbor [redacted] peer-group internal
neighbor [redacted] peer-group internal
```

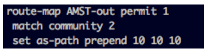
Multiple valid answers for question 3.3

More specific advertisements



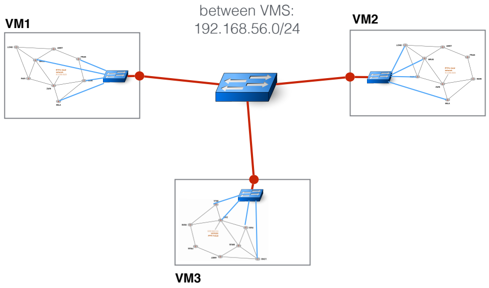
```
router bgp [redacted]
 bgp router-id [redacted].0.1.2
 network [redacted].0.0.0/9
 network [redacted].128.0.0/9
```

AS path prepending

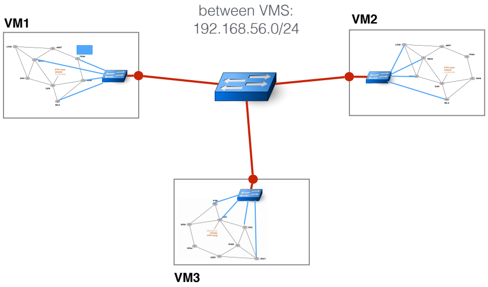


```
route-map AMST-out permit 1
 match community 2
 set as-path prepend 10 10 10
```

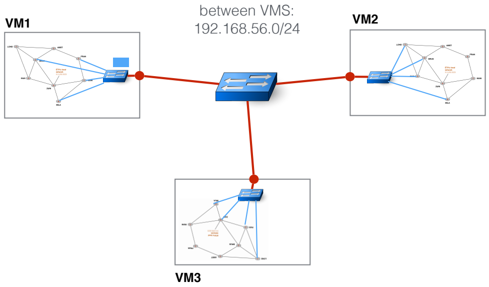
How we have built the mini-Internet



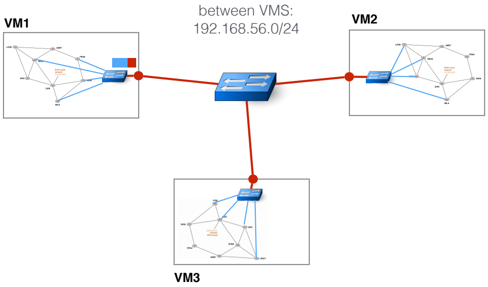
How we have built the mini-Internet



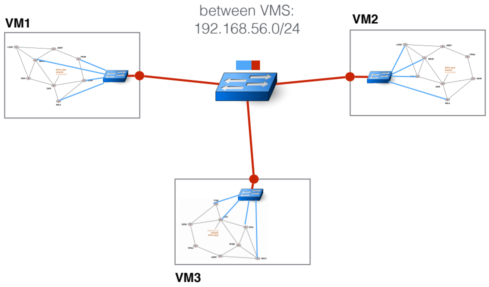
How we have built the mini-Internet



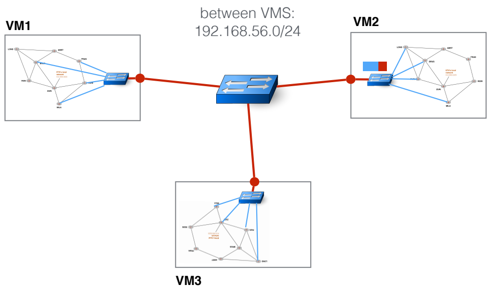
How we have built the mini-Internet



How we have built the mini-Internet

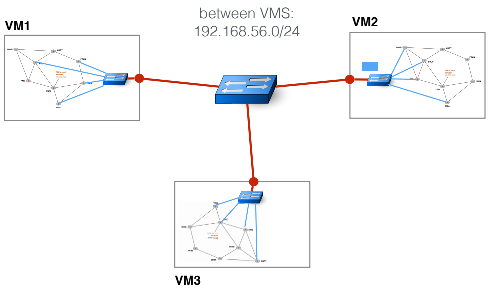


How we have built the mini-Internet



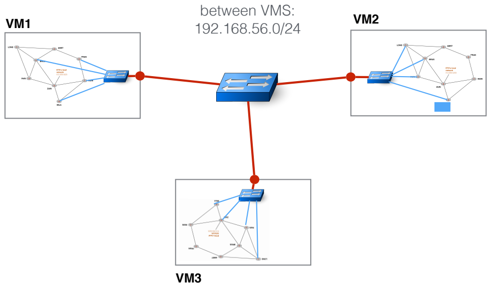
between VMS:
192.168.56.0/24

How we have built the mini-Internet



between VMS:
192.168.56.0/24

How we have built the mini-Internet



between VMS:
192.168.56.0/24

Communication Networks 2018
Routing Project

Except the grades within ~2weeks from now

Routing Project

Recap, demo and final results

Reliable Transport Project

Introduction and demo

Python and Git tutorial

Implement your own Reliable Transport Protocol

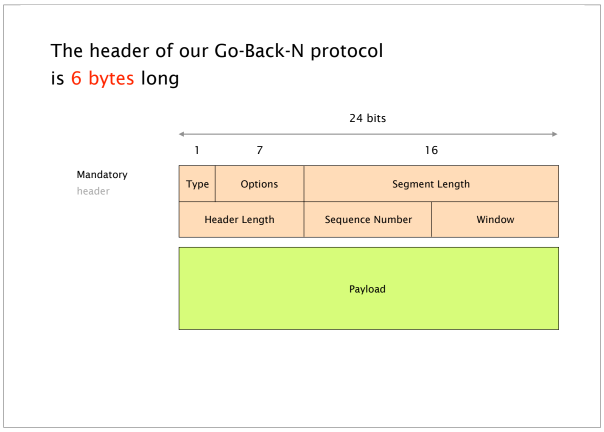
recover from packet loss
and reordering

Implement your own Reliable Transport Protocol

recover from packet loss
and reordering

Part 1	Simple Go-Back-N implementation Retransmit all packets after a timeout
Part 2	Support for Selective Repeat Fast retransmission after repeated ACKs
Part 3	Support for Selective Acknowledgements (SACK) SACK contains blocks of correctly received segments

Let's see how the final sender and receiver should look like



The header of our Go-Back-N protocol is 6 bytes long

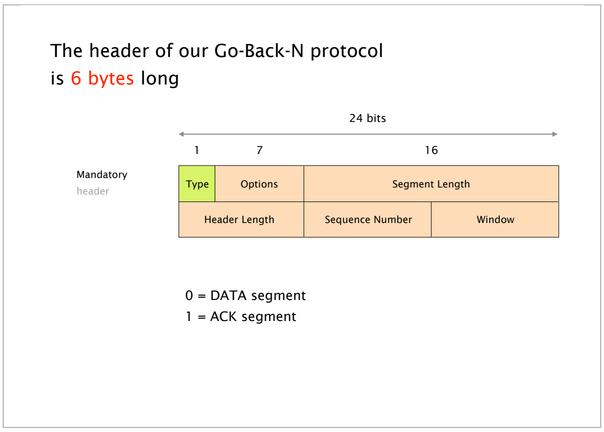
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

Payload



The header of our Go-Back-N protocol is 6 bytes long

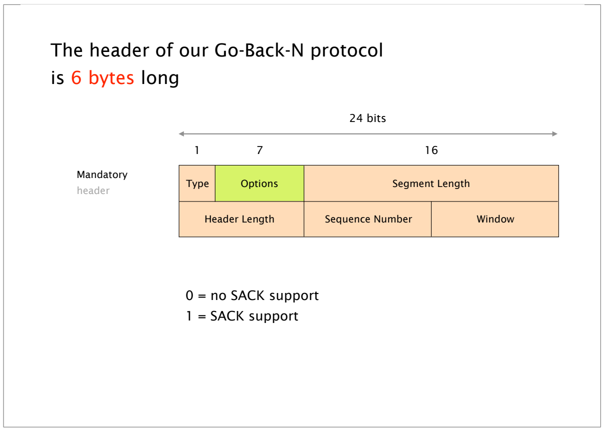
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

0 = DATA segment
1 = ACK segment



The header of our Go-Back-N protocol is 6 bytes long

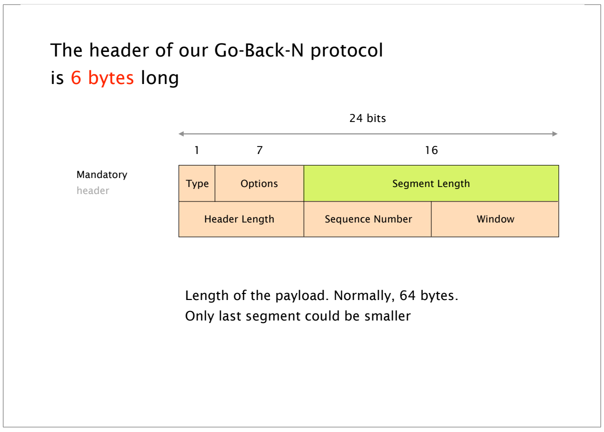
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

0 = no SACK support
1 = SACK support



The header of our Go-Back-N protocol is 6 bytes long

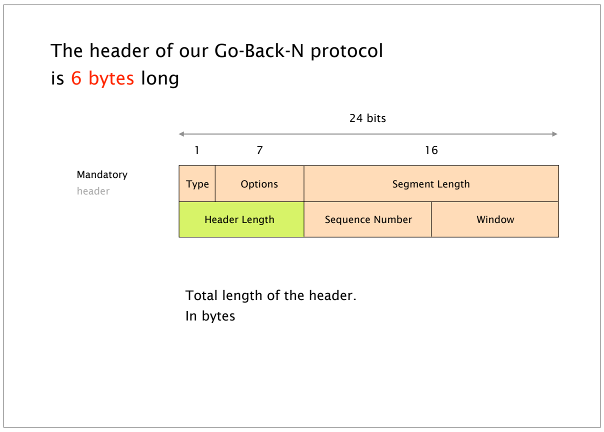
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

Length of the payload. Normally, 64 bytes.
Only last segment could be smaller



The header of our Go-Back-N protocol is 6 bytes long

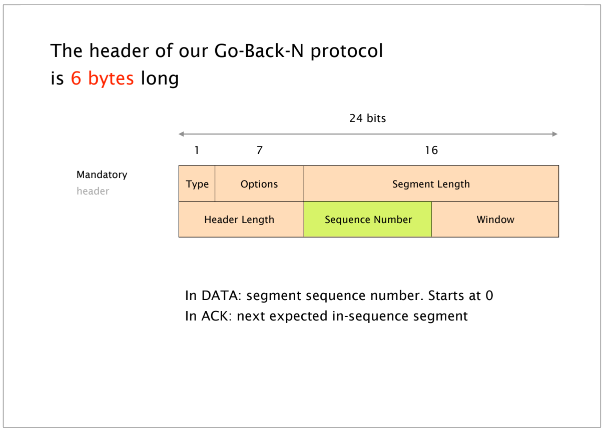
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

Total length of the header.
In bytes



The header of our Go-Back-N protocol is 6 bytes long

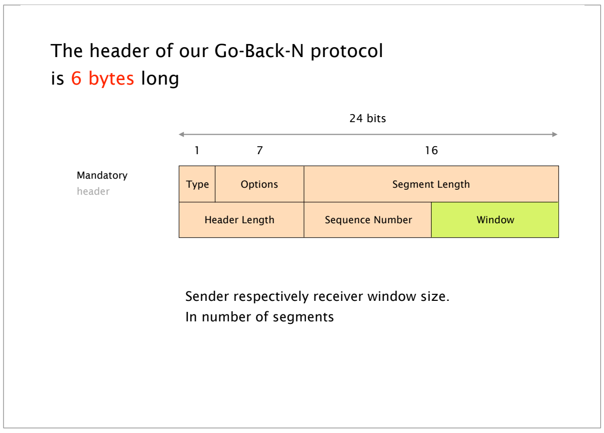
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

In DATA: segment sequence number. Starts at 0
In ACK: next expected in-sequence segment



The header of our Go-Back-N protocol is 6 bytes long

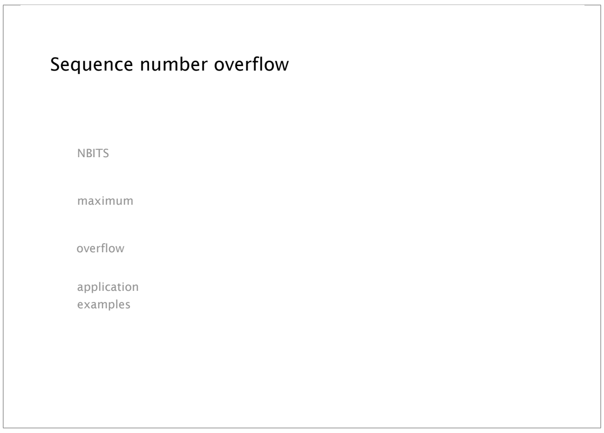
24 bits

1 7 16

Mandatory header

Type	Options	Segment Length
Header Length	Sequence Number	Window

Sender respectively receiver window size.
In number of segments



Sequence number overflow

NBITS

maximum

overflow

application examples

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-or-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

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

Selective Repeat can increase the performance

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

Selective Repeat can increase the performance

Sent segments:	0	~~1~~	2	3	4	5	
Receiver behavior:	0	-	2	3	4	5	Out-or-order segments are **buffered**
Sent ACKs:	1	-	1	1	1	1	
Retransmission:	3 duplicate ACKs 1						timeout 1 2 3 4 5

Selective Repeat can increase the performance

Sent segments:	0	~~1~~	2	3	4	5	1	
Receiver behavior:	0	-	2	3	4	5		
Sent ACKs:	1	-	1	1	1	1	6	
Retransmission:	3 duplicate ACKs 1							timeout 1 2 3 4 5

For SACK we need an optional header

24 bits

	1	7	16	
Mandatory header	Type	Options	Segment Length	
	Header Length		Sequence Number	Window
Optional header	Block Length		Left edge 1st block	Length 1st block
	Padding		Left edge 2nd block	Length 2nd block
	Padding		Left edge 3rd block	Length 3rd block
	Payload			

For SACK we need an optional header

24 bits

	8	8	8
Optional header	Block Length	Left edge 1st block	Length 1st block
	Padding	Left edge 2nd block	Length 2nd block
	Padding	Left edge 3rd block	Length 3rd block

Number of SACK blocks in the optional header
Between 1 and 3

For SACK we need an optional header

24 bits

Optional header	8	8	8
	Block Length	Left edge 1st block	Length 1st block
	Padding	Left edge 2nd block	Length 2nd block
	Padding	Left edge 3rd block	Length 3rd block

Start of the first block

For SACK we need an optional header

24 bits

Optional header	8	8	8
	Block Length	Left edge 1st block	Length 1st block
	Padding	Left edge 2nd block	Length 2nd block
	Padding	Left edge 3rd block	Length 3rd block

Length of the first block. In number of segments
A block with one segment has size 1

For SACK we need an optional header

24 bits

Optional header	8	8	8
	Block Length	Left edge 1st block	Length 1st block
	Padding	Left edge 2nd block	Length 2nd block
	Padding	Left edge 3rd block	Length 3rd block

Padding for better alignment

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

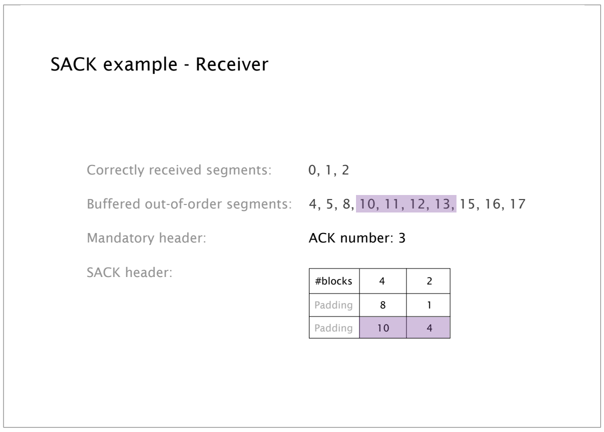
Correctly received segments:	0, 1, 2
Buffered out-of-order segments:	4, 5, 8, 10, 11, 12, 13, 15, 16, 17
Mandatory header:	ACK number: 3
SACK header:	

#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

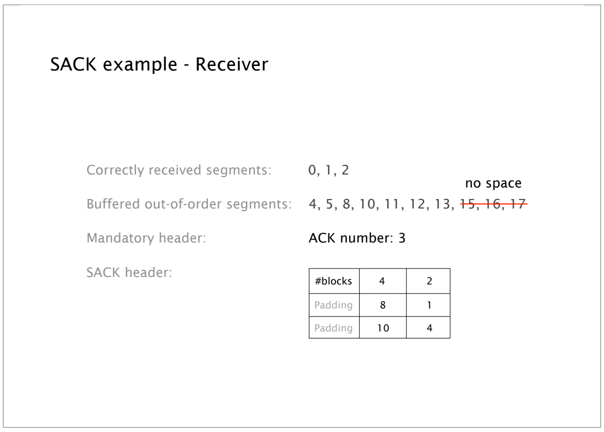
Correctly received segments: 0, 1, 2

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

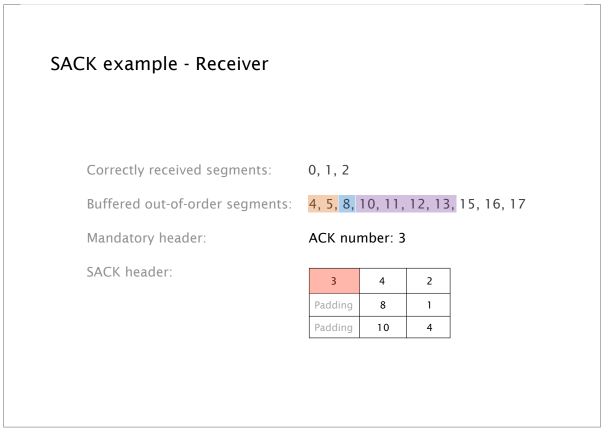
no space

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

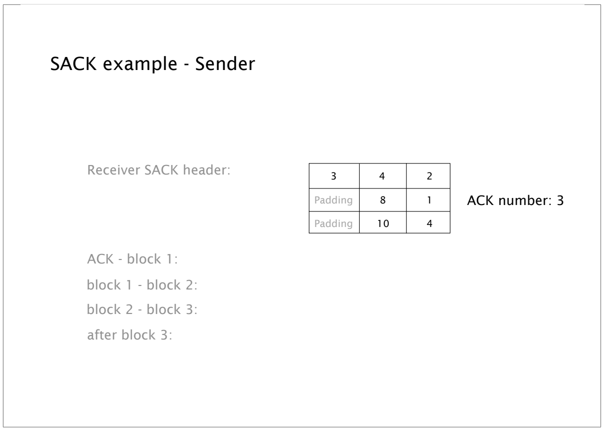
Correctly received segments: 0, 1, 2

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

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

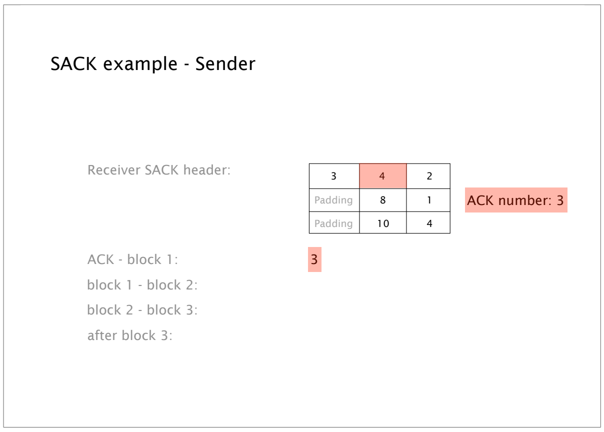
ACK number: 3

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

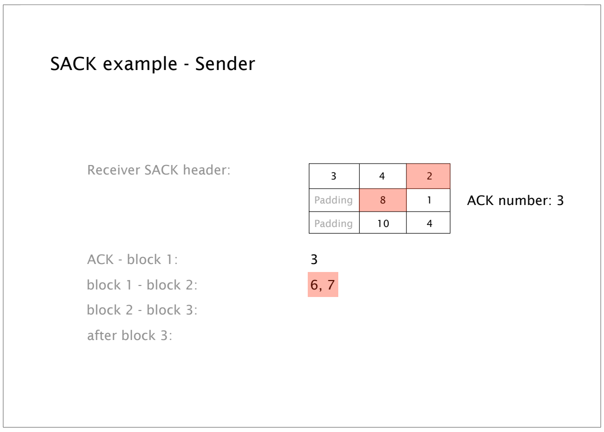
ACK number: 3

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

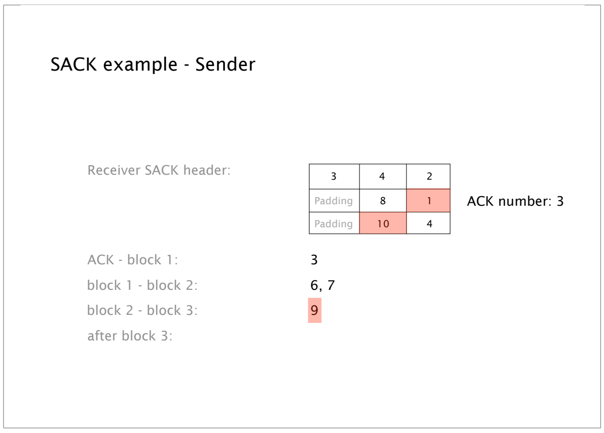
ACK number: 3

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

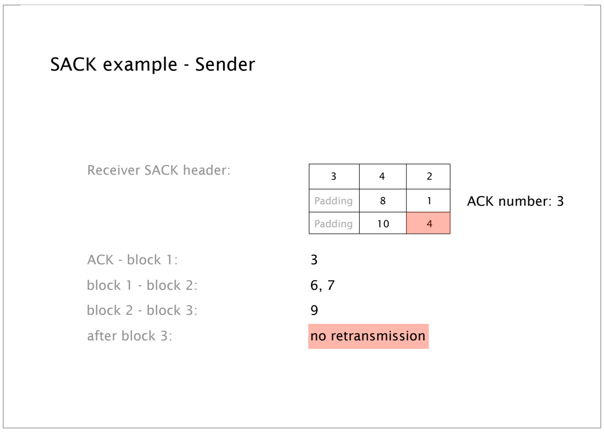
ACK number: 3

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

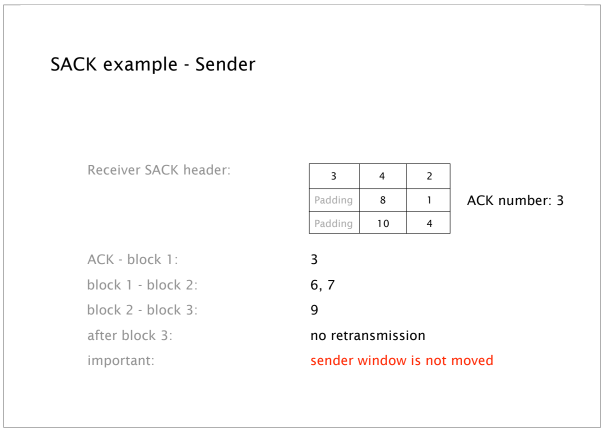
ACK number: 3

ACK - block 1: 3

block 1 - block 2: 6, 7

block 2 - block 3: 9

after block 3: no retransmission



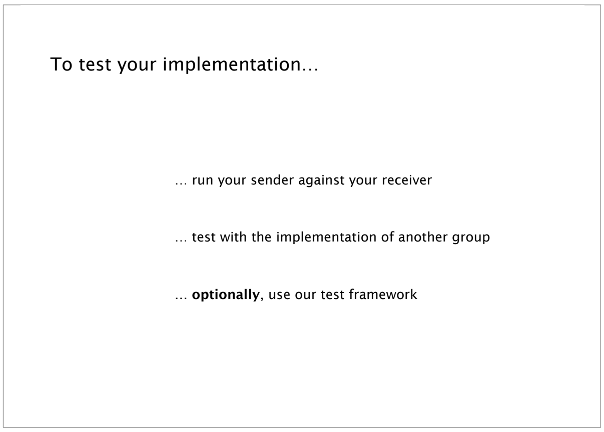
SACK example - Sender

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
important:	sender window is not moved

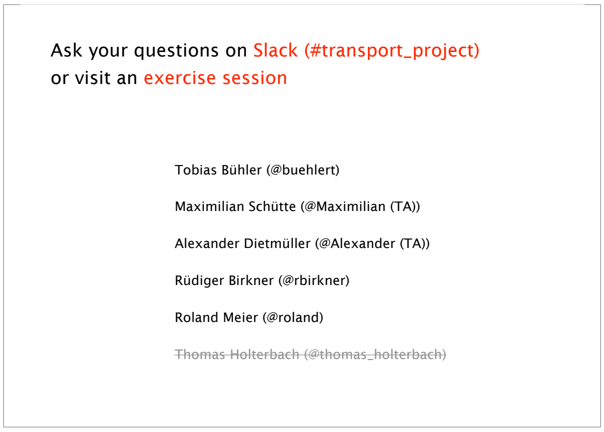


To test your implementation…

… run your sender against your receiver

… test with the implementation of another group

… **optionally**, use our test framework



Ask your questions on Slack (#transport_project) or visit an exercise session

Tobias Bühler (@buehlert)

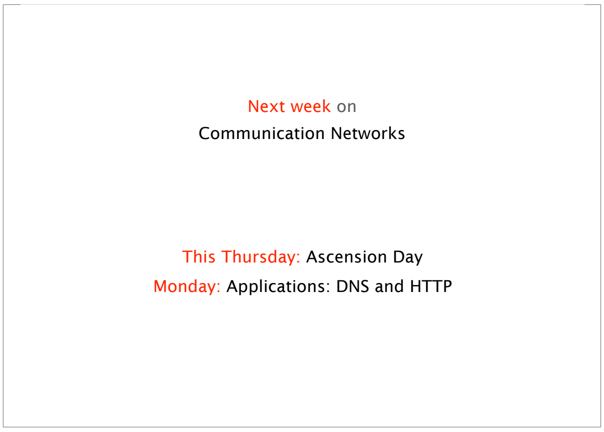
Maximilian Schütte (@Maximilian (TA))

Alexander Dietmüller (@Alexander (TA))

Rüdiger Birkner (@rbirkner)

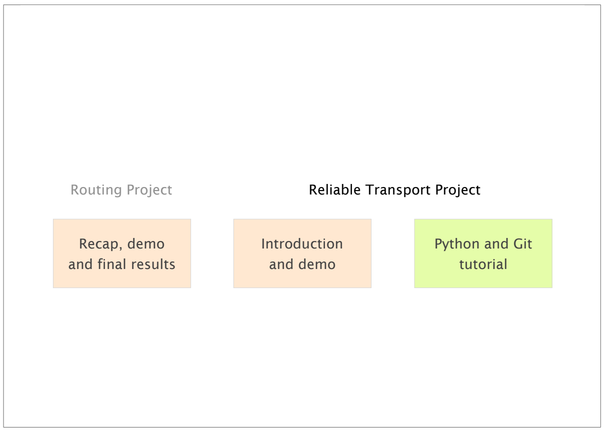
Roland Meier (@roland)

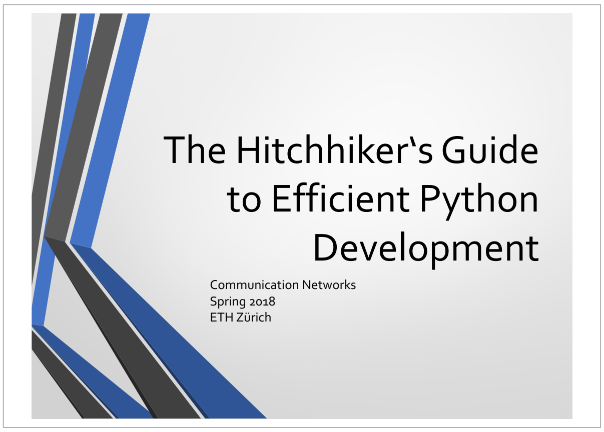
~~Thomas Holterbach (@thomas_holterbach)~~



Next week on
Communication Networks

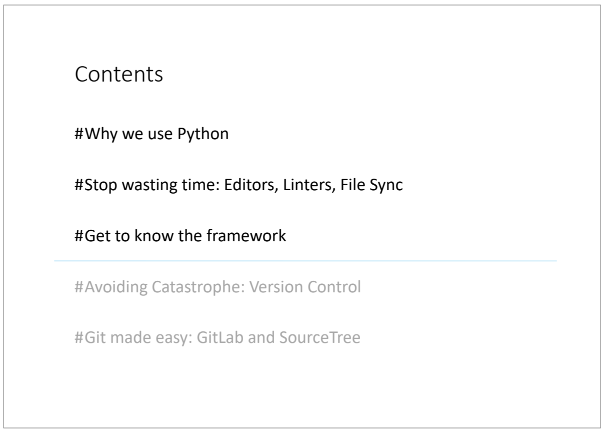
This Thursday: Ascension Day
Monday: Applications: DNS and HTTP





The Hitchhiker's Guide to Efficient Python Development

Communication Networks
Spring 2018
ETH Zürich



Contents

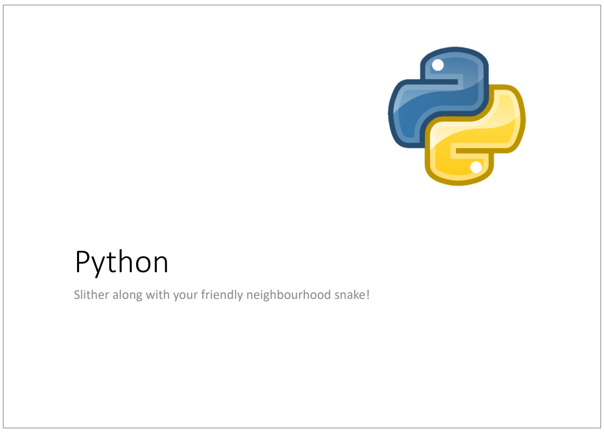
#Why we use Python

#Stop wasting time: Editors, Linters, File Sync

#Get to know the framework

#Avoiding Catastrophe: Version Control

#Git made easy: GitLab and SourceTree



Python

Slither along with your friendly neighbourhood snake!

Reasons to choose Python

#Interpreted Language

#Many packages available

#Simple yet powerful Syntax / Beginner Friendly

#Often used in academia and science

Learn the Basics BEFORE You Start!

We promise the basics will pay off…

Learn the Basics BEFORE You Start!

#One afternoon on learnpython.org should suffice

#If you skip the preparation, bugs may go unnoticed and cost you points

#Also you will spend much more time on debugging than you would have to learn the python basics

Learning Python for Pros

https://learnxinyminutes.com/docs/python3/

Learning Python for Everyone

#Interactive Getting Started Guide
  # http://www.learnpython.org/

#Short Intro
  # https://developers.google.com/edu/python/

#Not So Short Intro
  # http://thepythonguru.com/
  # https://docs.python.org/3/tutorial/index.html

#Detailed Intro
  # https://learnpythonthehardway.org/python3/

#Free Video Series for Beginners
  # https://mva.microsoft.com/en-US/training-courses/introduction-to-programming-with-python-8360

#Udemy Lecture for Beginners
  # https://www.udemy.com/complete-python-bootcamp/

Learning Python for Beginners

http://www.learnpython.org/

Python 2.7 or 3.x?

#Python 2.7 is slowly dying

#Python 3.x is cleaner, better, faster, stronger…

#Details
  # https://wiki.python.org/moin/Python2orPython3
  # https://www.dataquest.io/blog/python-2-or-3/
  # https://www.digitalocean.com/community/tutorials/python-2-vs-python-3-practical-considerations-2
  # http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html

Which Python Shall It Be?

Two major distributions to consider…

Which Python Shall It Be?

CPython from python.org

- The „default“ distribution
- Is installed on the VMs
- Comes only with standard library
- Pip packet manager

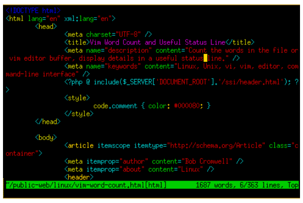
Anaconda by Continuum Analytics

- Optimized for data science and large scale science apps
- Derived from CPython
- Ships with a big library of science related packages
- Uses Conda packet manager
- But also supports pip

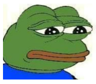
VSCode & PyLint

It's 2018, get your development workflow together!

Editing on the console is cumbersome…



```
<!DOCTYPE html>
<html lang="en" xml:lang="en">
        <head>
                <meta charset="UTF-8" />
                <title>Vim Word Count and Useful Status Line</title>
                <meta name="description" content="Count the words in the file or
 vim editor buffer, display details in a useful status line." />
                <meta name="keywords" content="Linux, Unix, vi, vim, editor, com
mand-line interface" />
                <?php @ include($_SERVER['DOCUMENT_ROOT'].'/ssi/header.html'); ?
>
                <style>
                        code.comment { color: #000080; }
                </style>
        </head>

        <body>
                <article itemscope itemtype="http://schema.org/Article" class="c
ontainer">
                <meta itemprop="author" content="Bob Cromwell" />
                <meta itemprop="about" content="Linux" />
                <header>
~/public-web/linux/vim-word-count.html[html]      1687 words, 6/363 lines, Top
```



#feelsawfulman

… but sometimes useful for quick fixes!

Many good Python IDEs available!

Sublime Text

Visual Studio Code

Atom.io

JetBrains PyCharm

Eclipse PyDev

Many good Python IDEs available!

#Any of the above will do, you the one you know and adapt it to the project!

#Top three are basic and can be used for many programming languages

#PyCharm is the most powerful Python IDE and even free for ETH students (professional edition)

Integrated Development Environment Benefits

#Easy to set up and getting started

#Come with many supporting tools out of the box

#IntelliSense, Syntax Checker / Linter, Auto completion…

#GUI based debugging is much faster and easier

Linter

#A Linter performs static code analysis

#It points out…

#… errors in your code

#… redundant code

#… code that can be optimized

#… changes that improve the readability of your code

#Use it so you don't have to spend hours chasing typos!

Secure File Transfer Protocol (SFTP)

#Available via extension for Visual Studio Code

#Makes transfering files from / to the VM super easy

#Extension shows you differences between local and vm code

Demo Time!

#Install Python

#Install Visual Studio Code & Python / PyLint + sftp extension

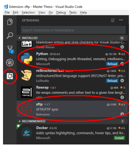
#Configure sftp & Download Project Files

#IntelliSense Demo

#CHECK SLACK FOR VIDEO DEMO! (to be released…)

Step-by-Step Installation Reference

- Install CPython 3.x or Anaconda / Miniconda 3.x
  - https://www.python.org/downloads/
  - https://www.anaconda.com/download/
- Install Visual Studio Code
  - https://code.visualstudio.com/
- Start Visual Studio Code and click on the extensions icon on the left
- Search for and install Python (ms-python.python) and sftp (liximomo.sftp)
- Reload after BOTH installations have finished


Extension: sftp - Master Thesis - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXTENSIONS
Search Extensions in Marketplace
INSTALLED 12
Markdown linting and style checking for Visual Studio C...
David Anson
Python 2018.4.0 10.1M 4.5
Linting, Debugging (multi-threaded, remote), Intellisens...
Microsoft Reload
reStructuredText
reStructuredText language support (RST/ReST linter, pre...
LeXtudio Reload
Rewrap 1.8.1 60K 5
Re-wraps comments and other text to a given line lengt...
sftp 1.1.7 310K 4.5
SFTP/FTP sync
liximomo
RECOMMENDED 1
Docker 0.0.26 3.1M 4.5
Adds syntax highlighting, commands, hover tips, and lin...
Microsoft Install

Configure Python and PyLint in VSCode

- Press F1 and enter “Python: Select Interpreter”
- Choose the python version that you just installed
  - On Mac use the one in /usr/local, NOT the system installation!
- Press F1 again and enter “Python: Selecte Linter” and choose “PyLint”
- The first time you open a python file, you will receive a message box in the bottom right corner saying that PyLint is not installed. Press “install” to do so.
  - On Mac, gcc will be installed if not installed already

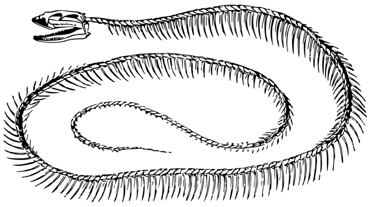
Configure sftp and Download Code Reference

- In VSCode, open a folder where you want your project files to be located.
- Press F1 and enter “SFTP: Config”
- A config file will pop up. Enter the details to your VM, as shown on the next slide. Providing a password is optional.
- The config will be stored a subfolder .vscode and can be edited anytime.
- Right click in the VSCode file browser and use the SFTP features like “download”, “upload”, or “sync”.
- In general, the plugin is conservative when it comes to «destructive» operations. See Extension Info page for more details.

SFTP Example Config

```
{
    "protocol": "sftp",
    "host": "samichlaus.ethz.ch",
    "username": "root",
    "port": 3000+YOUR-GROUP-NUMBER,
    "remotePath": "./",
    "ignore": ["/.*"]
}
```

Don't forget this! It makes sure that you just copy the project related files!



The Project Skeleton

You don't need to start from scratch…

Sending and Receiving Packets in Python

scapy

Sending and Receiving Packets in Python

```
from scapy.all import send, IP, TCP

Payload = b"This is some binary test data."

packet = IP(src="192.168.0.1", dst="8.8.8.8") / TCP() / payload

send(packet)
```

Combine headers with the divison operator

Sending and Receiving Packets in Python

Show summary and details

```
print(packet.summary())

print(packet.show())
```

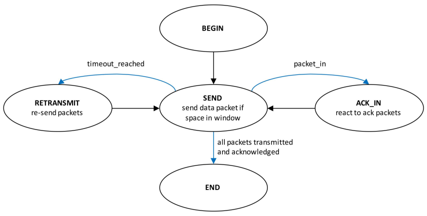
Access headers and data

```
from scapy.all import IP

ip_header = packet.getlayer(IP)
source_address = ip_header.src
payload = ip_header.payload
```

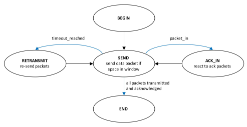
Define Your Own Header

```
from scapy.all import Packet, bind_layers, BitEnumField, BitField

class GBN(Packet):
    name = 'GBN'
    fields_desc = [
        BitEnumField("type", 0, 1, {0: "data", 1: "ack"}),
        BitField("options", 0, 7),
        # other fields ...
    ]

# Tell Scapy where to look for the header when receiving a packet
bind_layers(IP, GBN, frag=0, proto=222)
```

Our GBN Automaton is powered by Scapy



Our GBN Automaton is powered by Scapy



```
from scapy.all import Automaton, ATMT

class GBNSender(Automaton):
    @ATMT.state(initial=1)
    def BEGIN(self):
        raise self.SEND()

    @ATMT.state()
    def SEND(self):
        # Your code!
```

Transitions are triggered by exceptions

Where to start?

The GBN header is already defined...

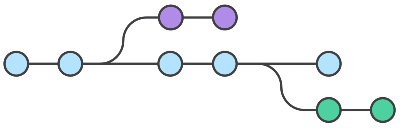
...you'll need to extend it in later questions

The automaton skeleton is fully implemented...

...no new states or transitions needed

The receiver already works for the first question...

...complete the sender, check receiver for inspiration



Version Control

If two people are working on one problem, you get two problems...

git Tracks Changes in Source Code

Without git

Everyone works on the same file and uploads it to the server.

The version uploaded last ***overwrites all other changes.***

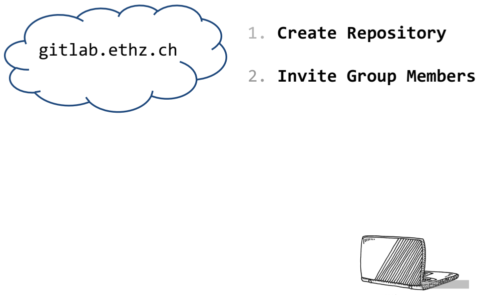
With git

Everyone works on the same file and pushes

the changes to the git repository.

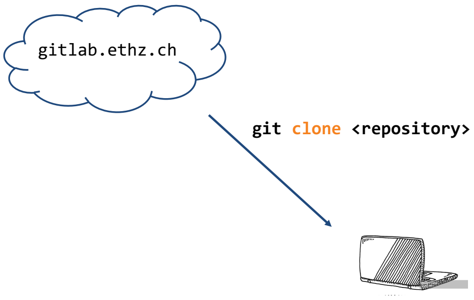
All changes are combined, nothing is lost.

git Workflow



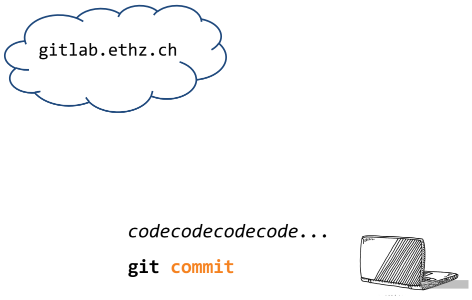
1. **Create Repository**
2. **Invite Group Members**

git Workflow



`git clone <repository>`

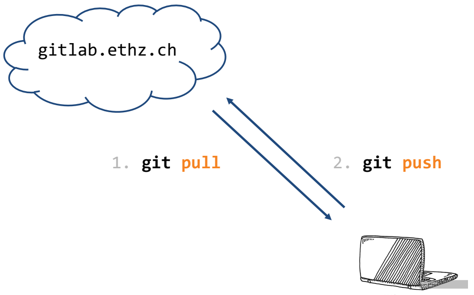
git Workflow



codecodecodecode...

`git commit`

git Workflow



1. `git pull`
2. `git push`

git Workflow

commit → **pull** → **push**

Try it yourself and learn more:

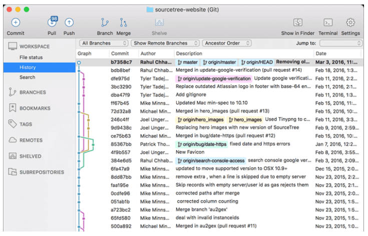
http://try.github.io/
https://backlog.com/git-tutorial/

SourceTree & Gitlab

… because no matter what they say, GUI matters.

SourceTree

See Slack for Video Demo! (To be released)



Graph	Commit	Author	Description	Date
	b7358c7	Rahul Chha...	master origin/master origin/HEAD Removing ol...	Mar 3, 2016, 11:...
	bdb8bef	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:3...
	dfe975d	Tyler Tadej...	origin/update-google-verification Update google verificati...	Feb 11, 2016, 2:2...
	3bc3290	Tyler Tadej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 11, 2016, 2:1...
	dba47f9	Tyler Tadej...	Add gitignore	Feb 11, 2016, 1:3...
	ff67b45	Mike Minns...	Updated Mac min-spec to 10.10	Feb 15, 2016, 11:...
	72d32a8	Michael Min...	Merged in hero_images (pull request #13)	Feb 15, 2016, 10:...
	246c4ff	Joel Unger...	origin/hero_images hero_images Used Tinypng to c...	Feb 11, 2016, 3:3...
	9d9438c	Joel Unger...	Replacing hero images with new version of SourceTree	Feb 9, 2016, 2:59...
	ce75b63	Michael Min...	Merged in bug/date-https (pull request #12)	Feb 15, 2016, 10:...
	85367bb	Patrick Tho...	origin/bug/date-https fixed date and https errors	Jan 7, 2016, 12:2...
	4f9b557	Joel Unger...	New Favicon	Feb 8, 2016, 3:55...
	384e6d5	Rahul Chhab...	origin/search-console-access search console google ver...	Feb 3, 2016, 2:09...
	6fa47a9	Mike Minns...	updated to move supported version to OSX 10.9+	Dec 15, 2015, 2:0...
	8dd87bb	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 2:2...
	faa195e	Mike Minns...	Skip records with empty server/user id as gas rejects them	Nov 23, 2015, 2:1...
	0cdfe96	Mike Minns...	corrected paths after merge	Nov 23, 2015, 2:0...
	051ab1b	Mike Minns...	corrected column counting	Nov 23, 2015, 1:5...
	a723bc2	Mike Minns...	Merge branch 'au2gex'	Nov 23, 2015, 1:5...
	65fd580	Mike Minns...	deal with invalid instanceids	Nov 23, 2015, 1:5...
	500a892	Michael Min...	Merged in au2gex (pull request #11)	Nov 23, 2015, 1:0...