Communication Networks

Spring 2022

Tobias Bühler

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

ETH Zürich

May 05 2022

Communication Networks

Exercise 9

Last week's exercise

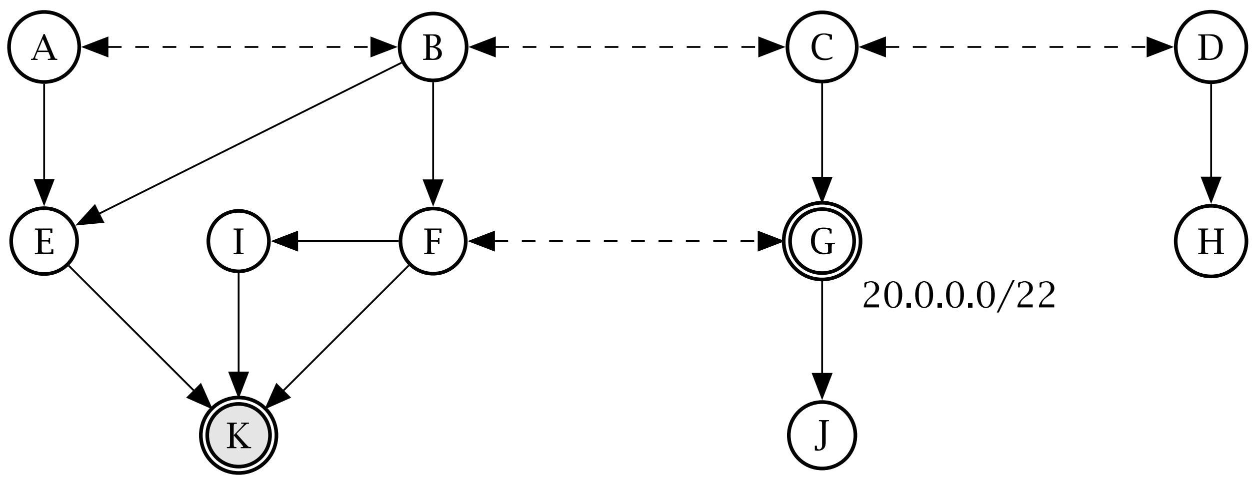
Important lecture topics

Introduction to this week's exercise

Time to solve the exercise

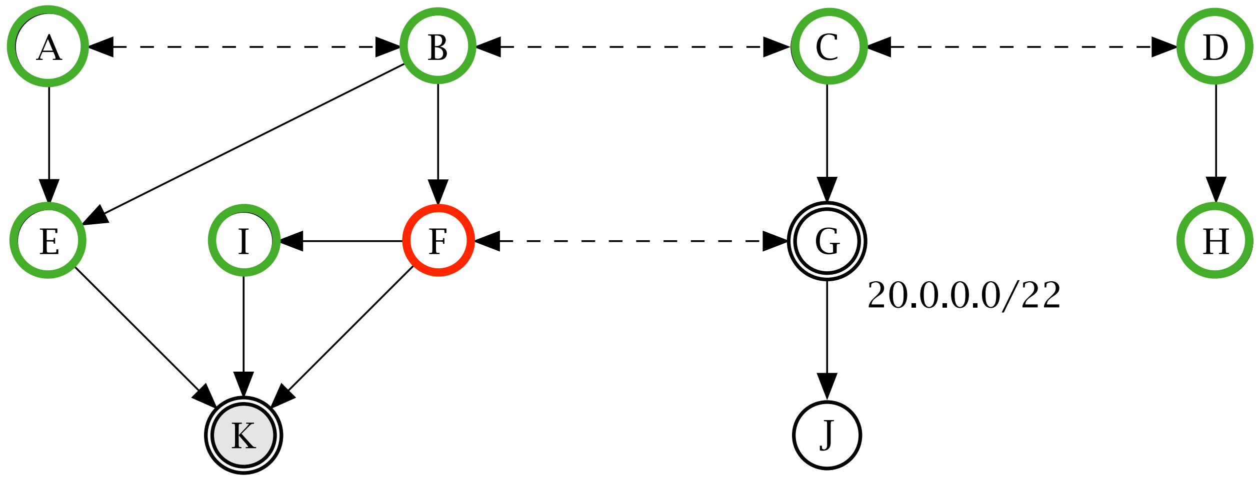
Task 8.3: BGP Hijack

AS path poisoning gives the hijacker some control over which ASes are/are not affected by the hijack



Task 8.3: BGP Hijack

AS path poisoning gives the hijacker some control over which ASes are/are not affected by the hijack

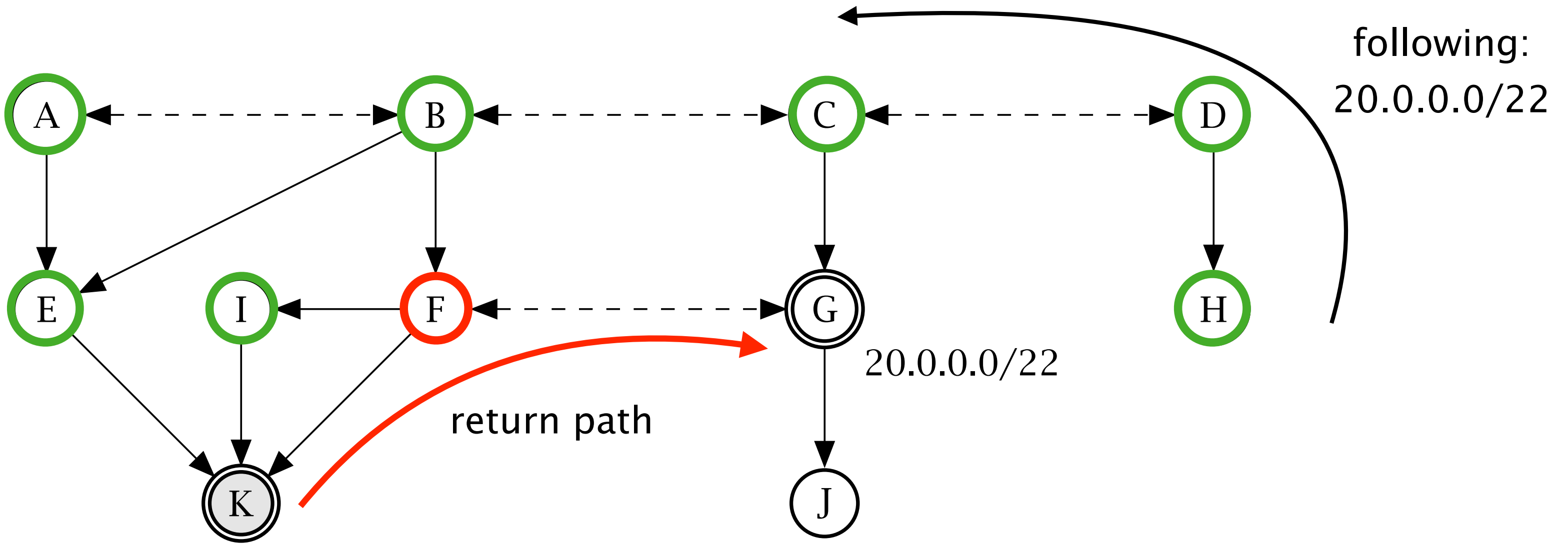


20.0.0.0/23 - AS path: F

20.0.2.0/23 - AS path: F

Task 8.3: BGP Hijack

AS path poisoning gives the hijacker some control over which ASes are/are not affected by the hijack



20.0.0.0/23 - AS path: F

20.0.2.0/23 - AS path: F

Communication Networks

Exercise 9

Last week's exercise

Important lecture topics

Introduction to this week's exercise

Time to solve the exercise

The Go-Back-N protocol

The Go-Back-N Protocol

The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions

The Go-Back-N Protocol

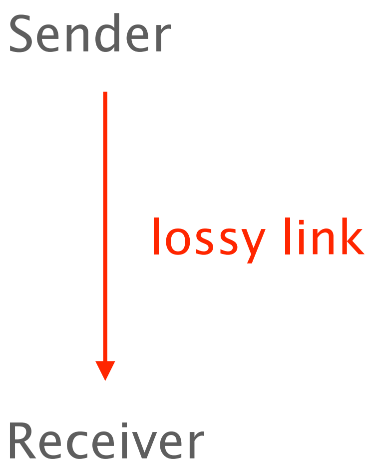
a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions

Sender

Receiver

The Go-Back-N Protocol

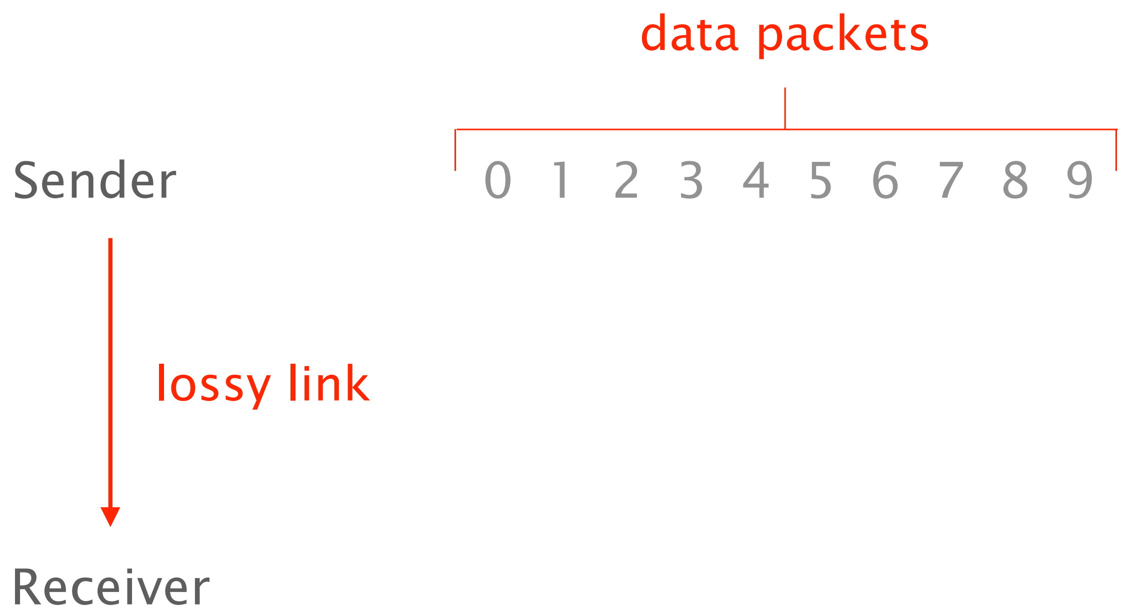
a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

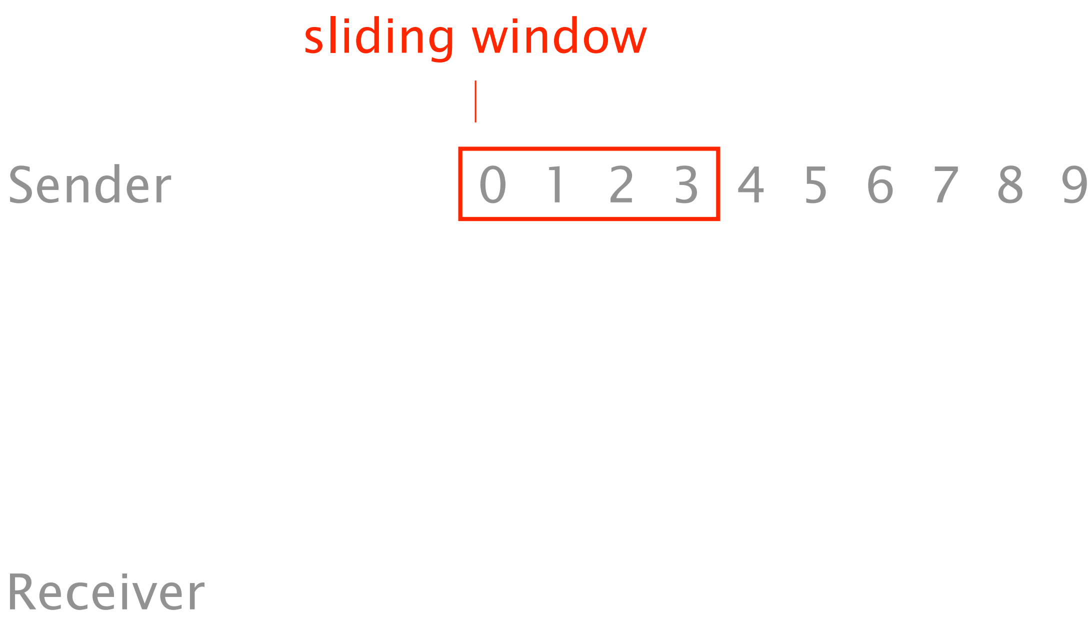
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

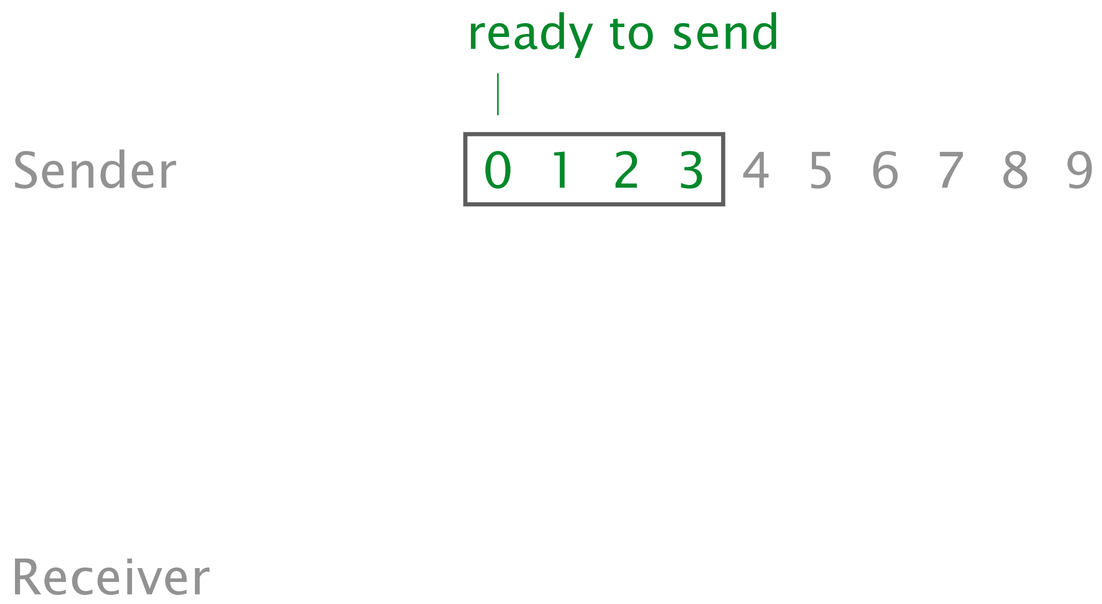
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

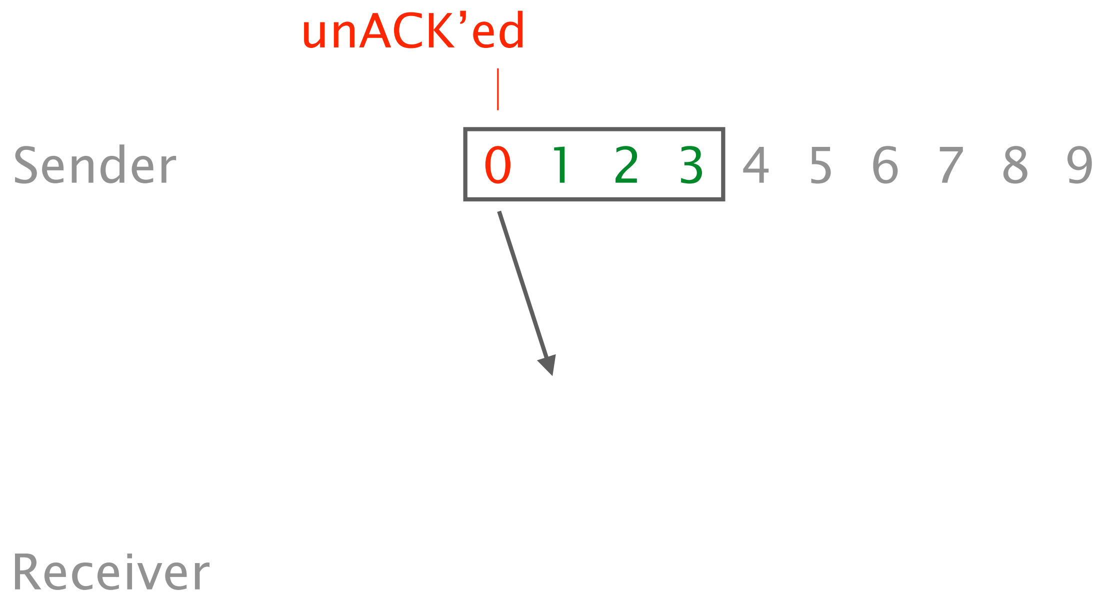
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

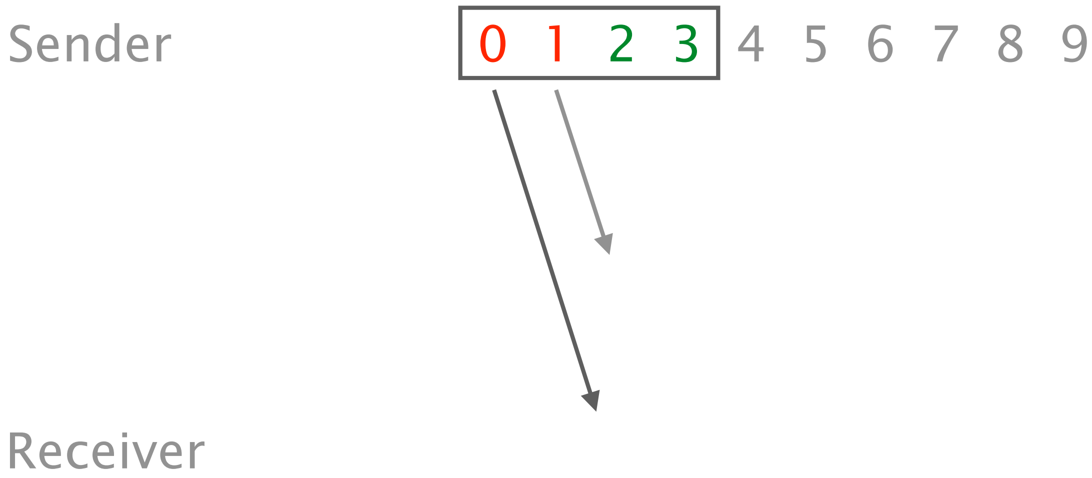
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

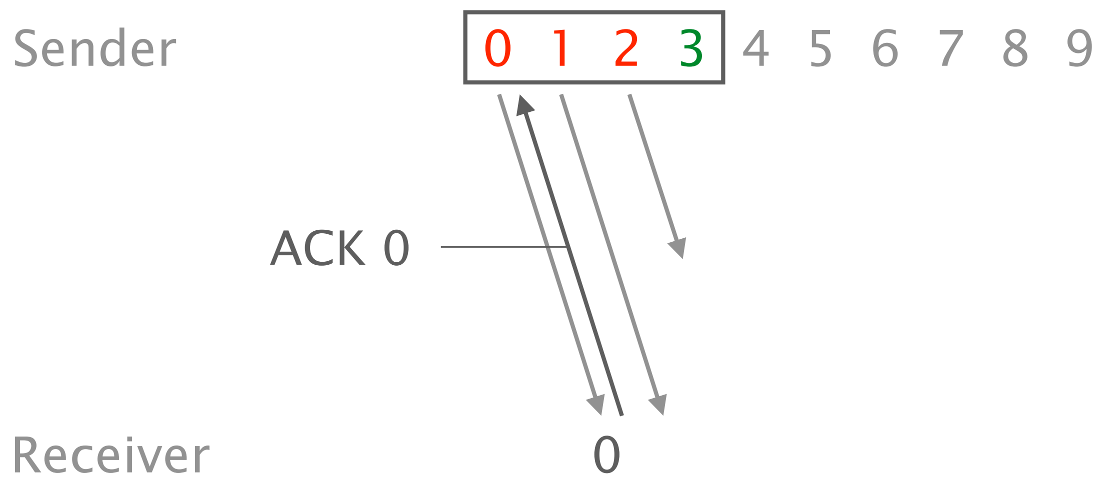
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

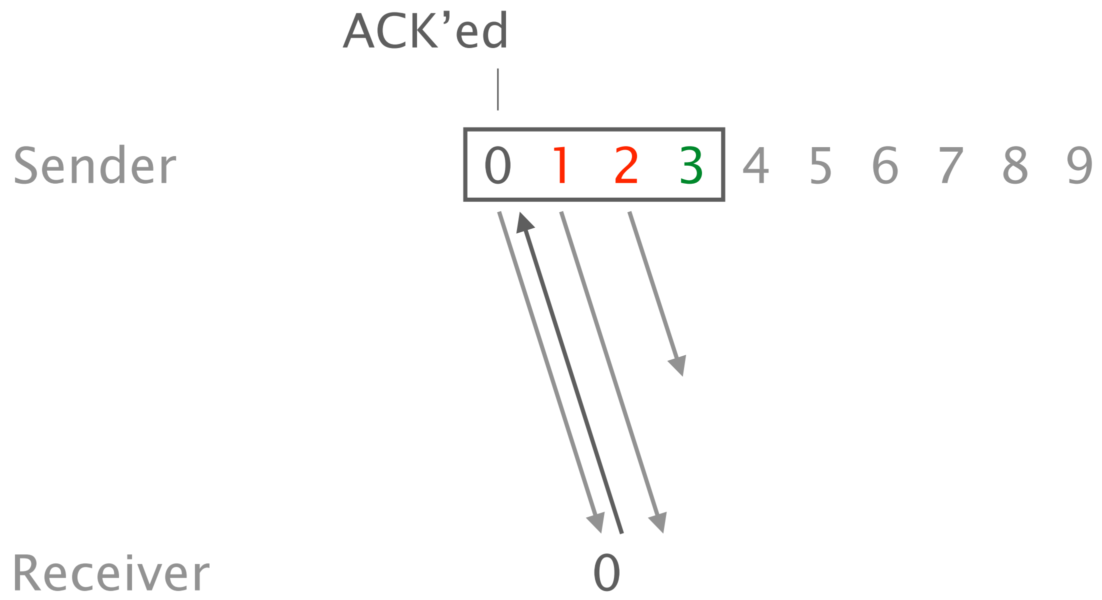
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

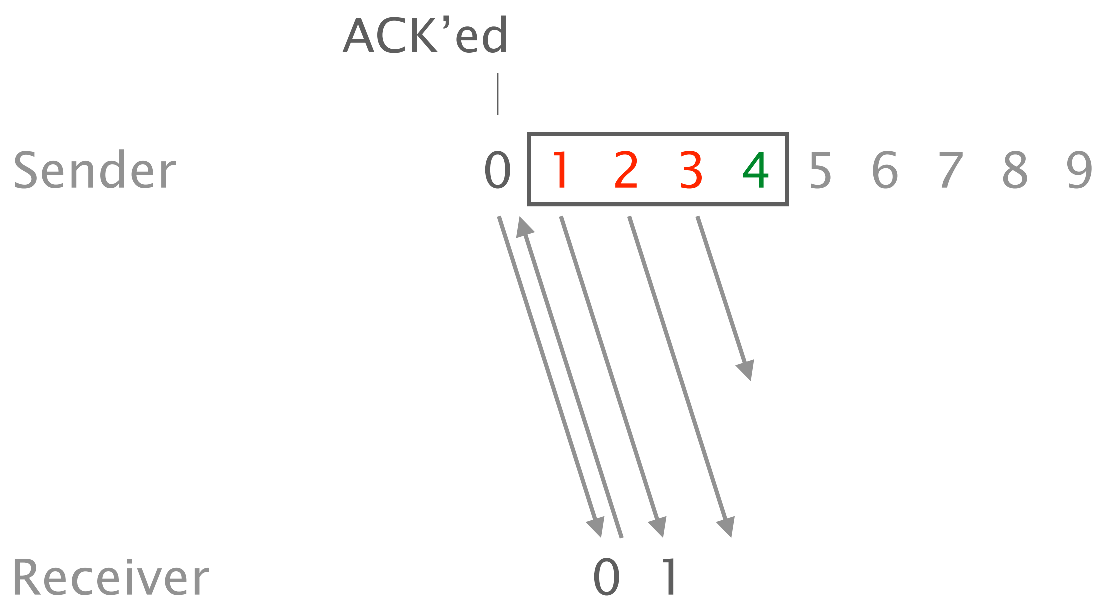
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

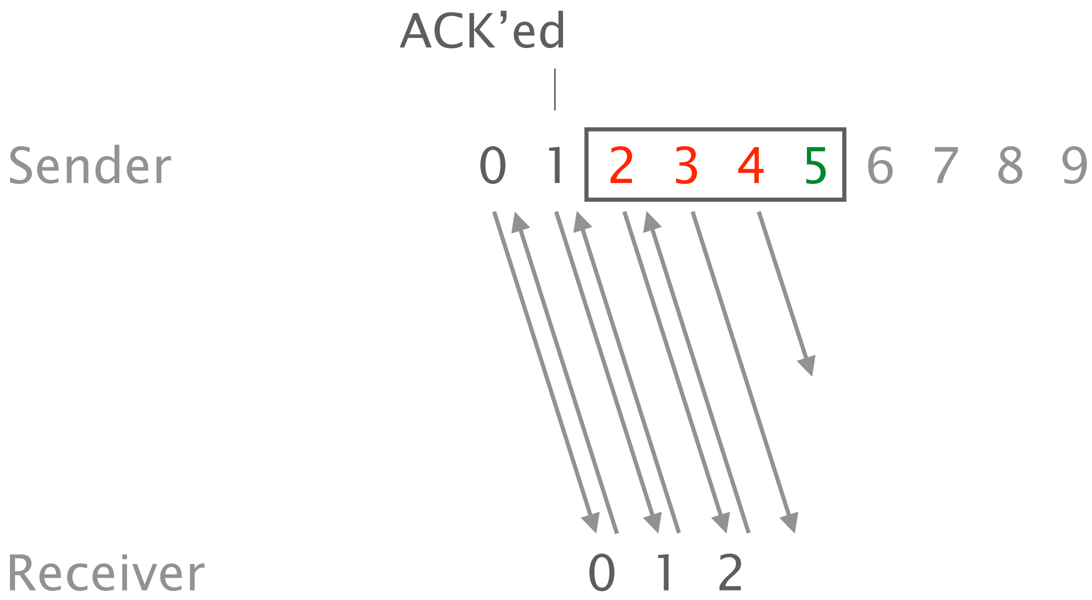
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

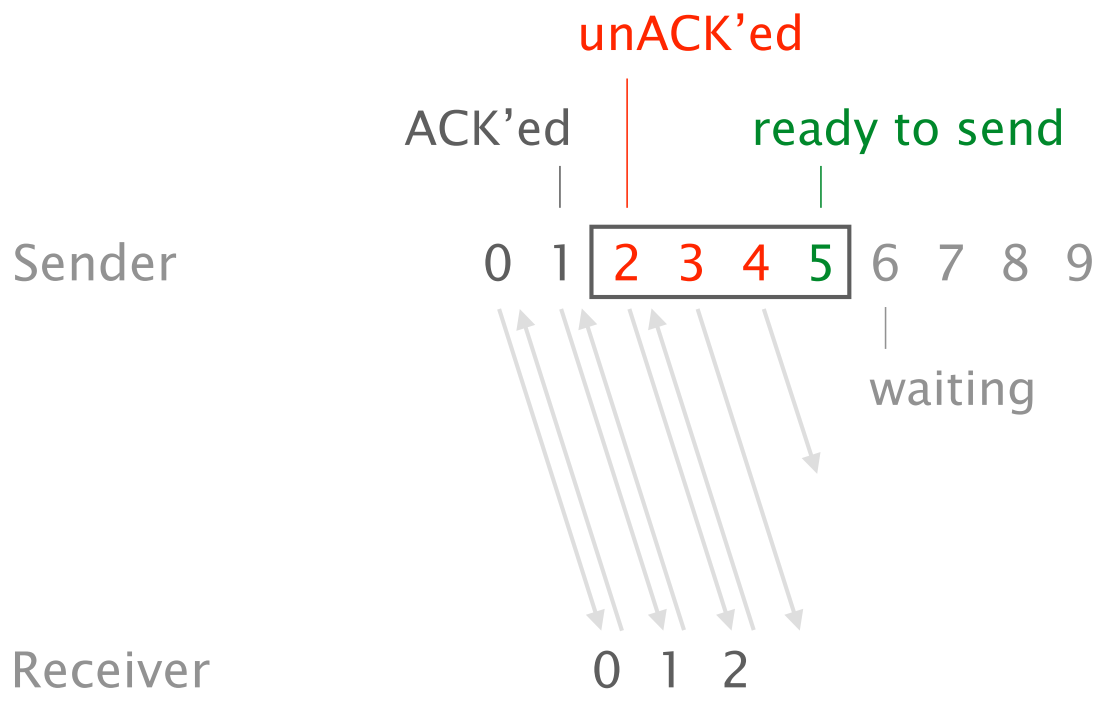
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

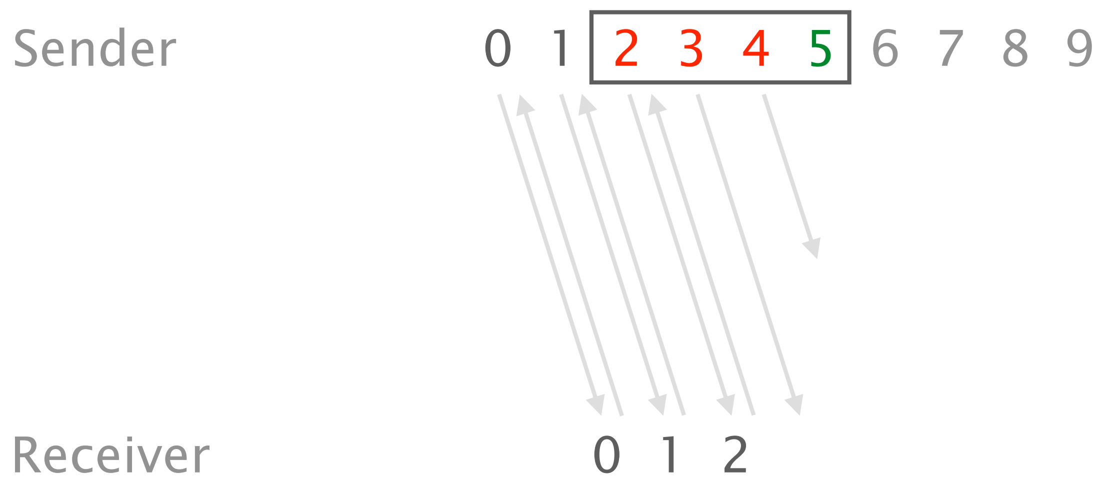
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

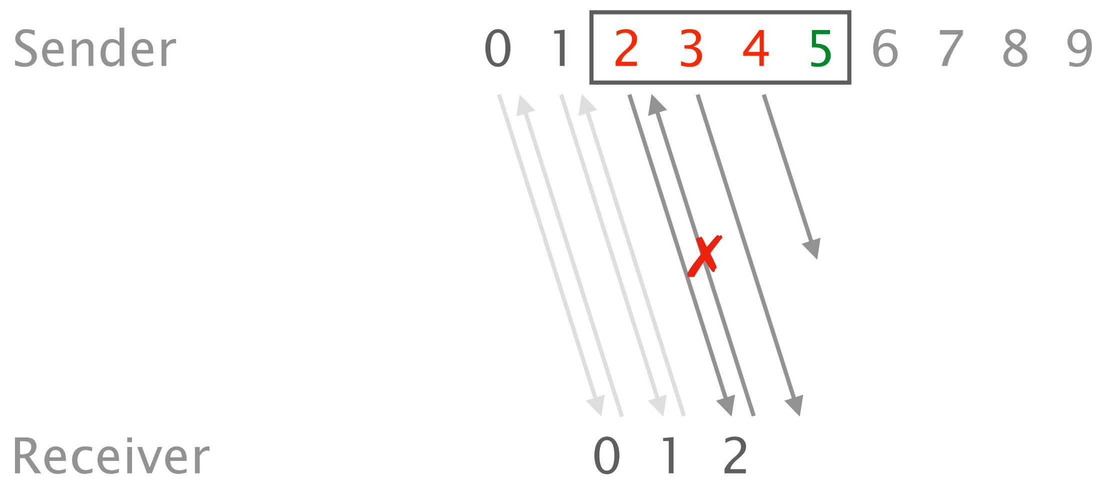
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

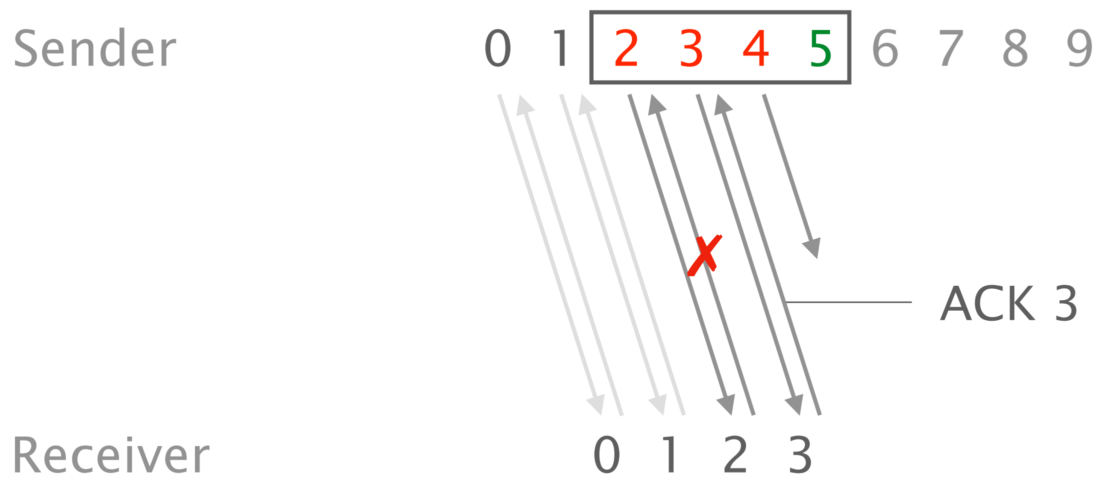
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

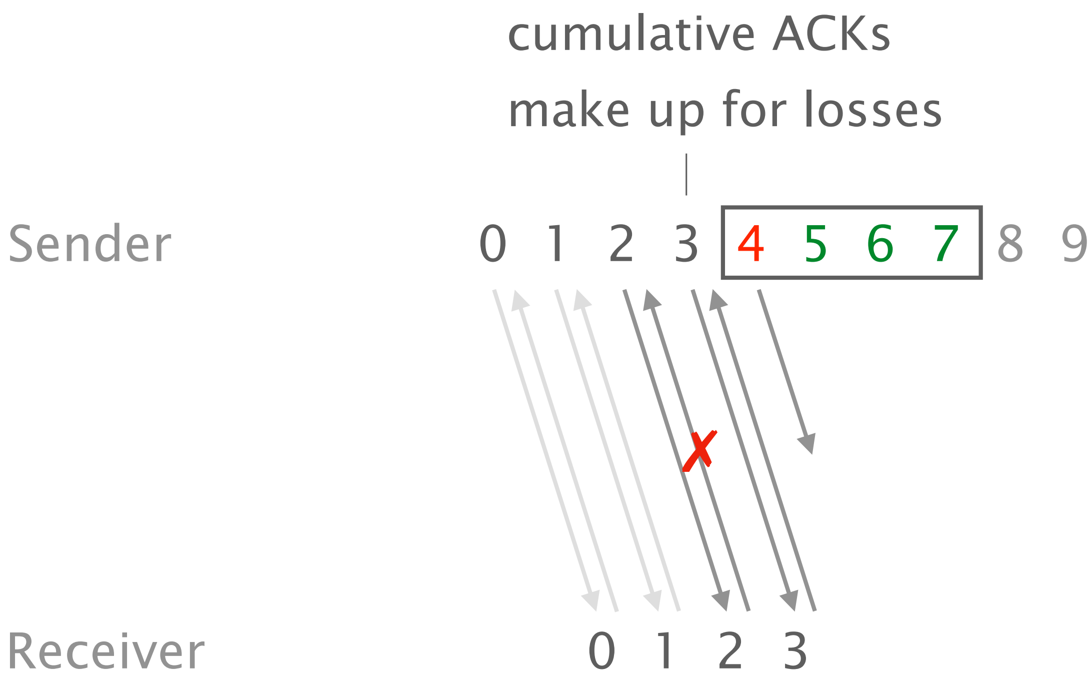
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

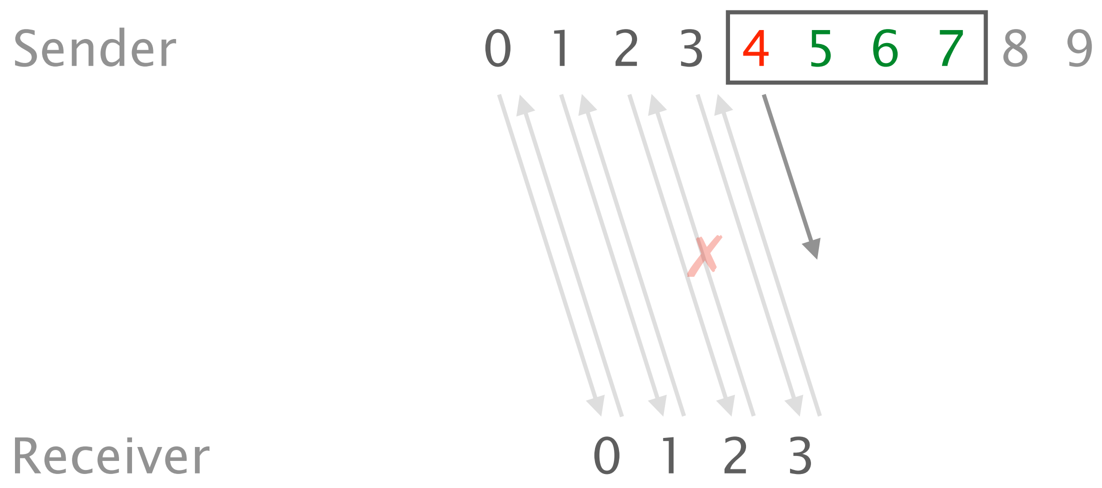
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

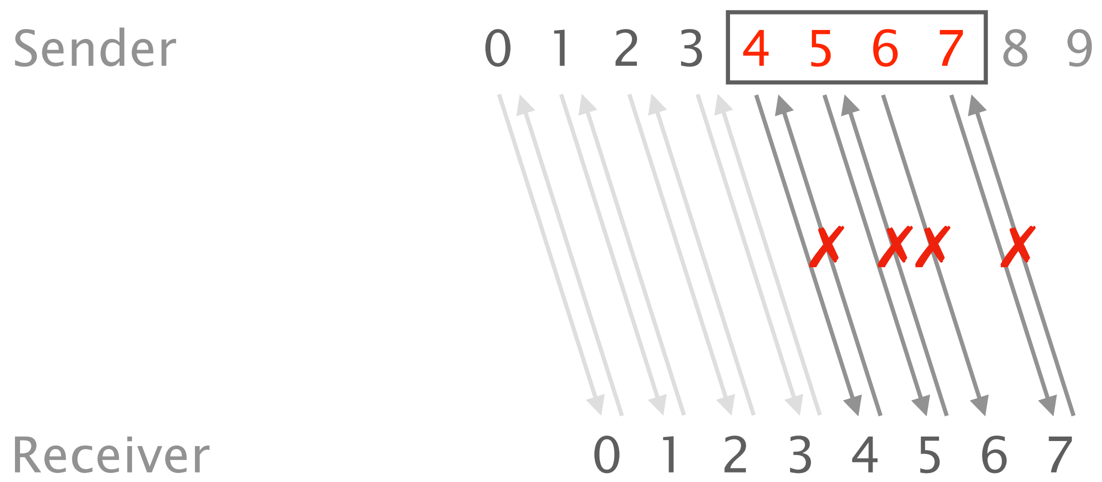
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

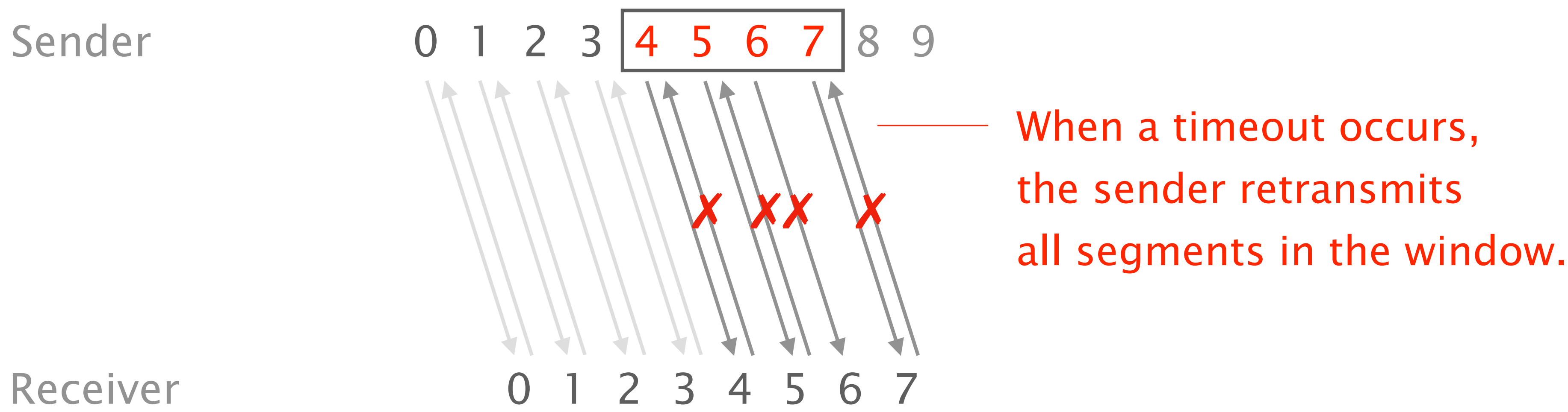
a simple reliable transport protocol with

a sliding window, cumulative ACKs, timeouts and retransmissions



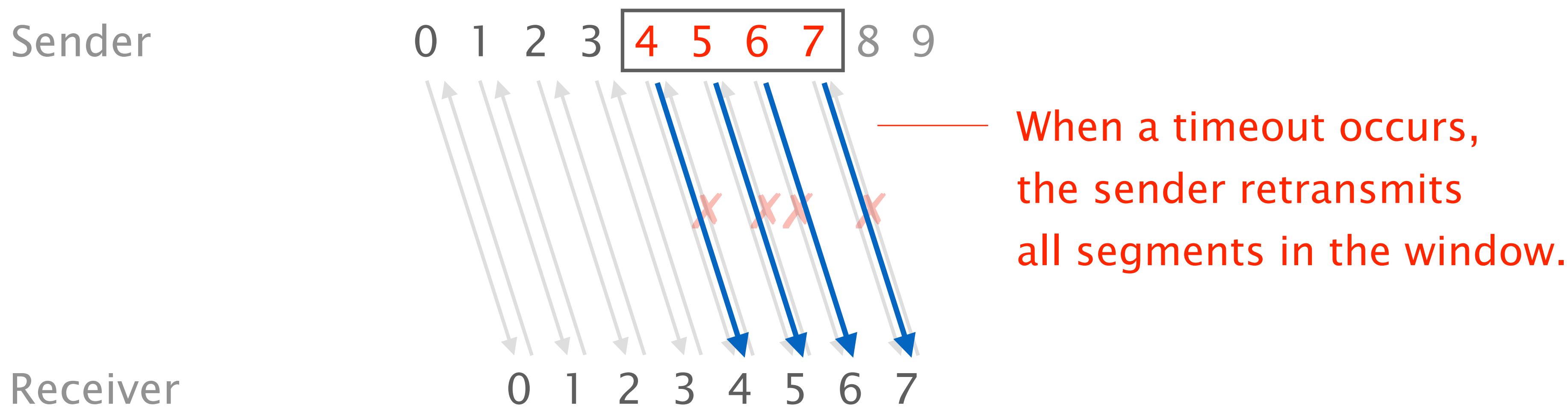
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

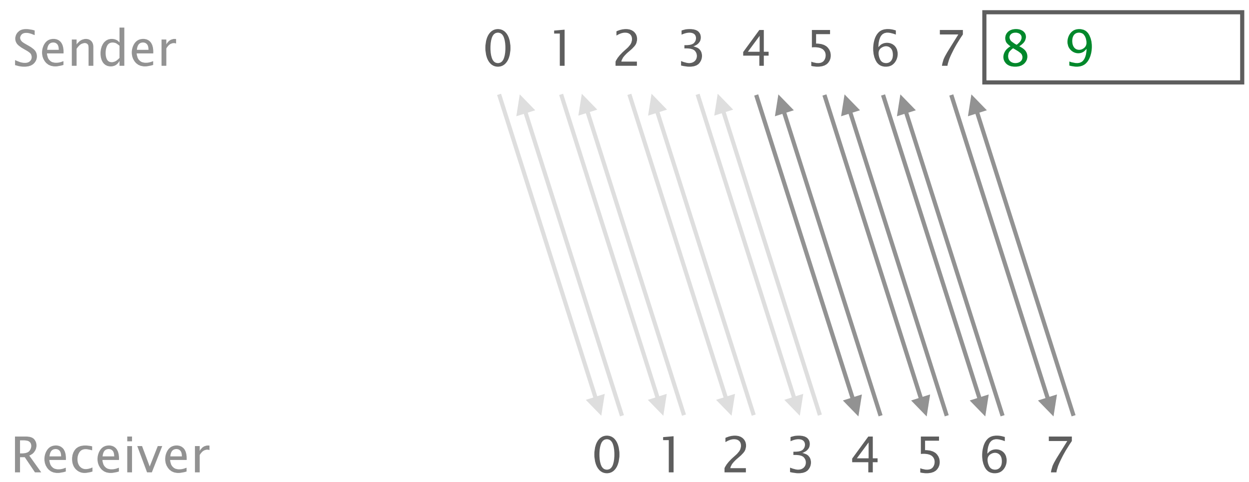
a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions



The Go-Back-N Protocol

a simple reliable transport protocol with

a sliding window, cumulative ACKs, timeouts and retransmissions



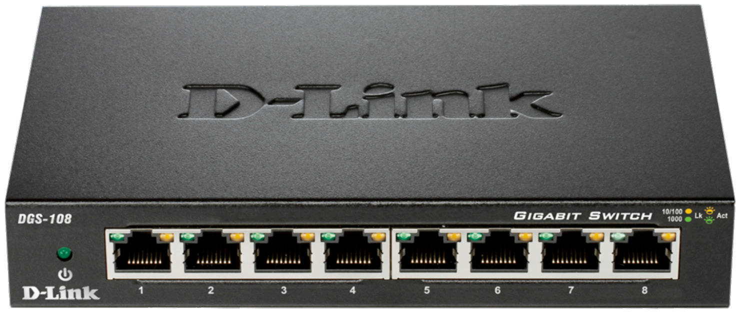
Physical and virtual ports

A **port** can describe two completely different concepts

A physical port on a switch or router (interface)

A logical (virtual) port on a host to demultiplex incoming data

Physical ports



Physical ports

Physical interface on a device

Often numbered from 1…N

Physical ports

Physical interface on a device

Often numbered from 1…N

Important if you configure a device (compare routing project)

These ports are normally **not** visible in a packet header

We also saw these ports in the Spanning Tree algorithm

Constructing a Spanning Tree in a nutshell

Switches…

elect a root switch

the one with the smallest identifier

determine if each interface is

on the shortest-path from the root

and disable it if not

For this switches exchange
Bridge Protocol Data Unit (BDPU) messages

Each switch *X* iteratively sends

BPDU (Y, d, X) to each neighboring switch

Y: the switch ID it considers as root

d: the # hops to reach it

initially

Each switch proposes itself as root

sends (X,0,X) on all its interfaces

Upon receiving (Y, d, X), checks if Y is a better root

if so, considers Y as the new root, flood updated message

Switches compute their distance to the root, for each port

simply add 1 to the distance received, if shorter, flood

Switches disable interfaces not on shortest-path tie-breaking

Upon receiving ≠ BPDUs from ≠ switches with = cost

Pick the BPDU with the lower switch sender ID

Upon receiving ≠ BPDUs from a neighboring switch

Pick the BPDU with the lowest port ID (e.g. port 2 < port 3) tie-breaking

Upon receiving ≠ BPDUs from ≠ switches with = cost

Pick the BPDU with the lower switch sender ID

Upon receiving ≠ BPDUs from a neighboring switch

Pick the BPDU with the lowest port ID (e.g. port 2 < port 3)

This switch receives two BPDUs from its neighbor

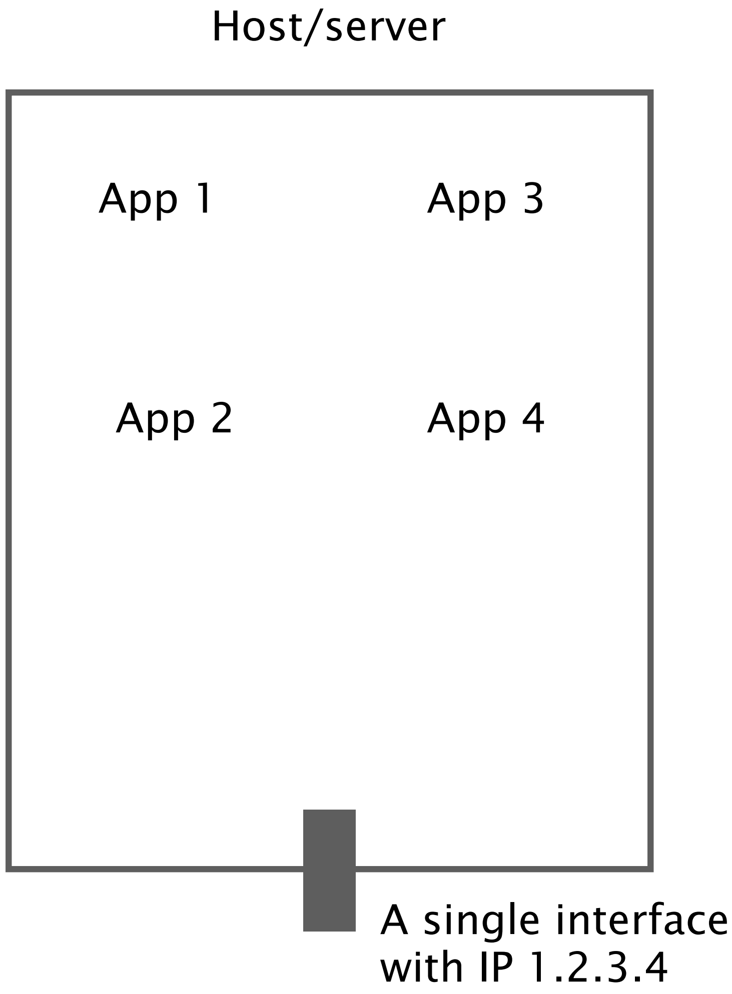
Both have the same cost

One is received over port 1, the other over port 4

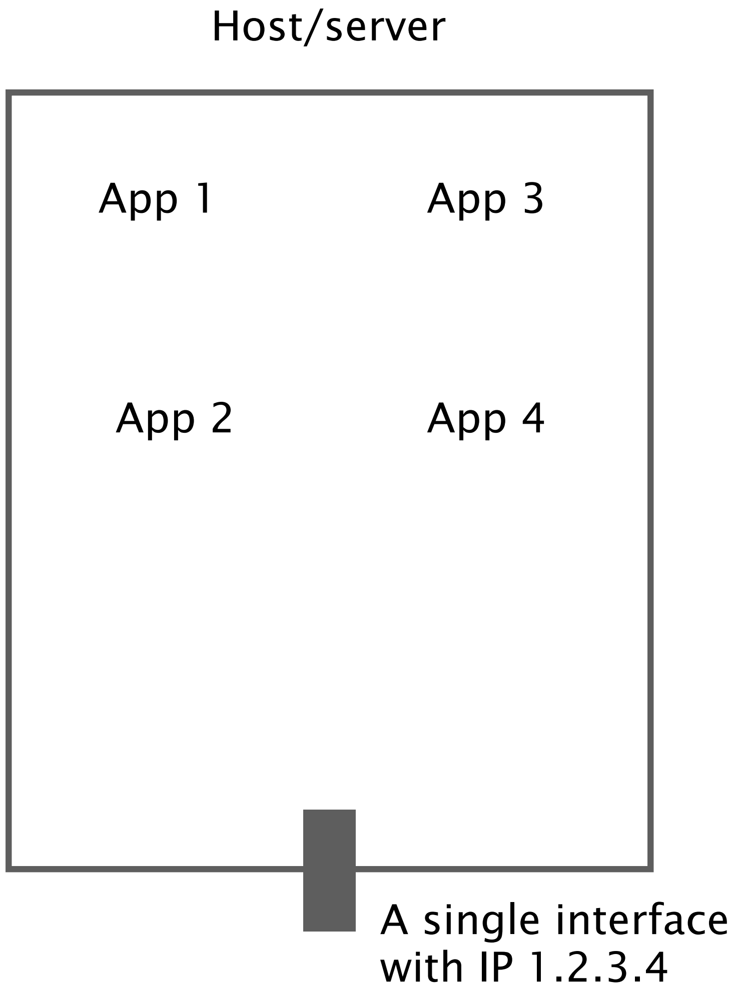
The switch picks the one from port 1

Logical (virtual) ports on a host

Logical (virtual) ports on a host

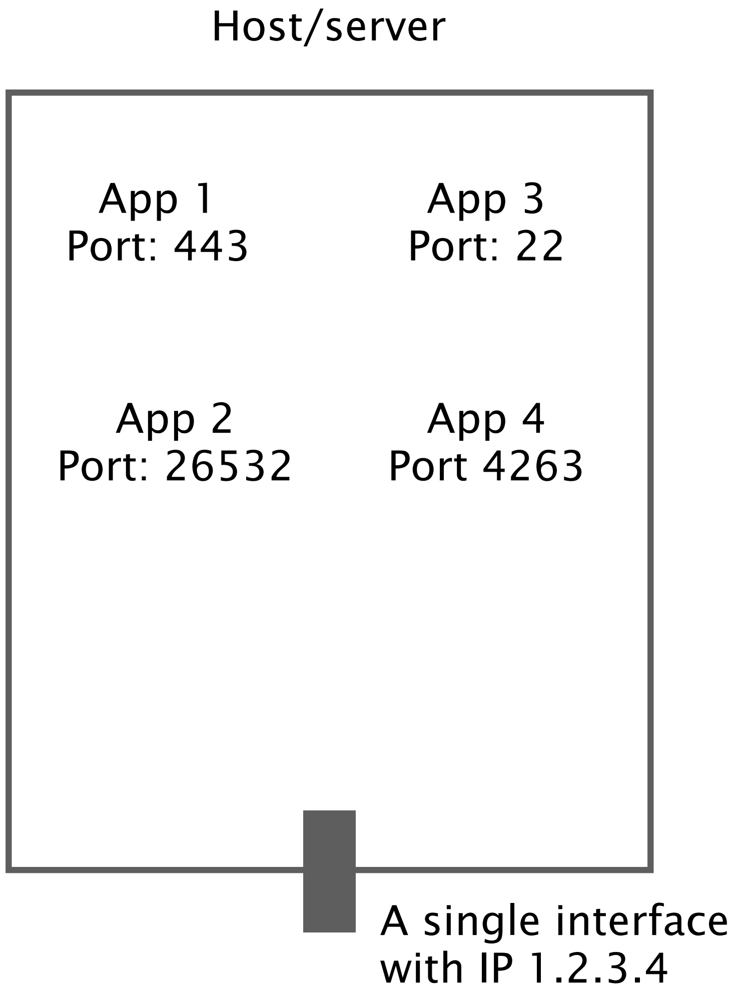


Logical (virtual) ports on a host



How does the host (the transport layer) know to which application it has to forward incoming packets?

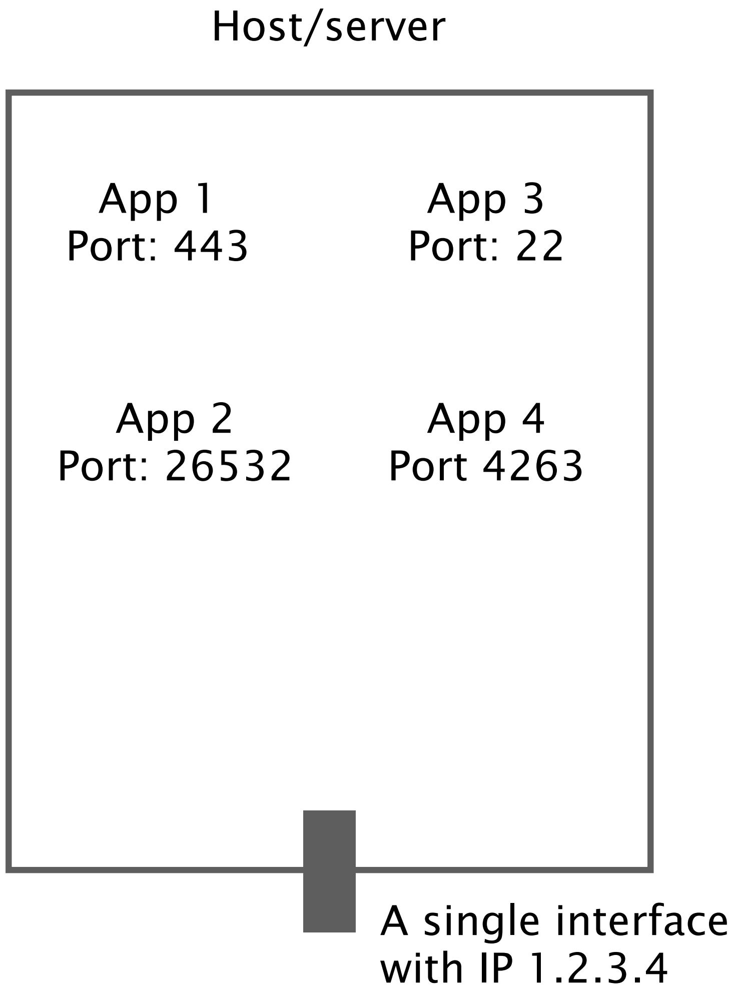
Logical (virtual) ports on a host



How does the host (the transport layer) know to which application it has to forward incoming packets?

Each application listens on a different logical **port**.

Logical (virtual) ports on a host



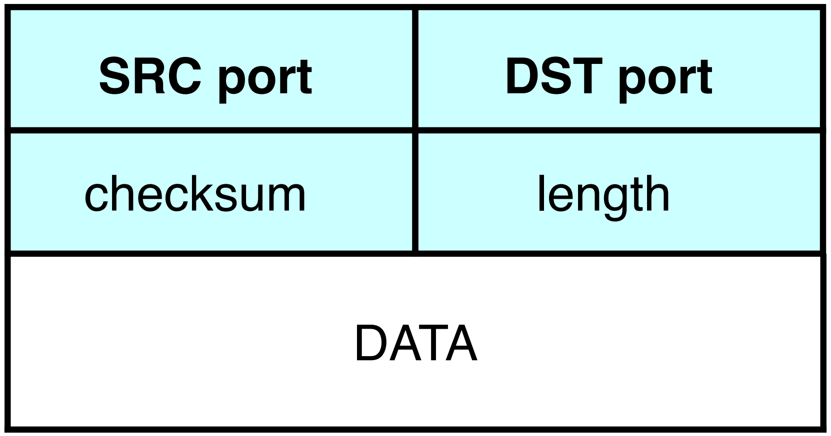
How does the host (the transport layer) know to which application it has to forward incoming packets?

Each application listens on a different logical **port**.

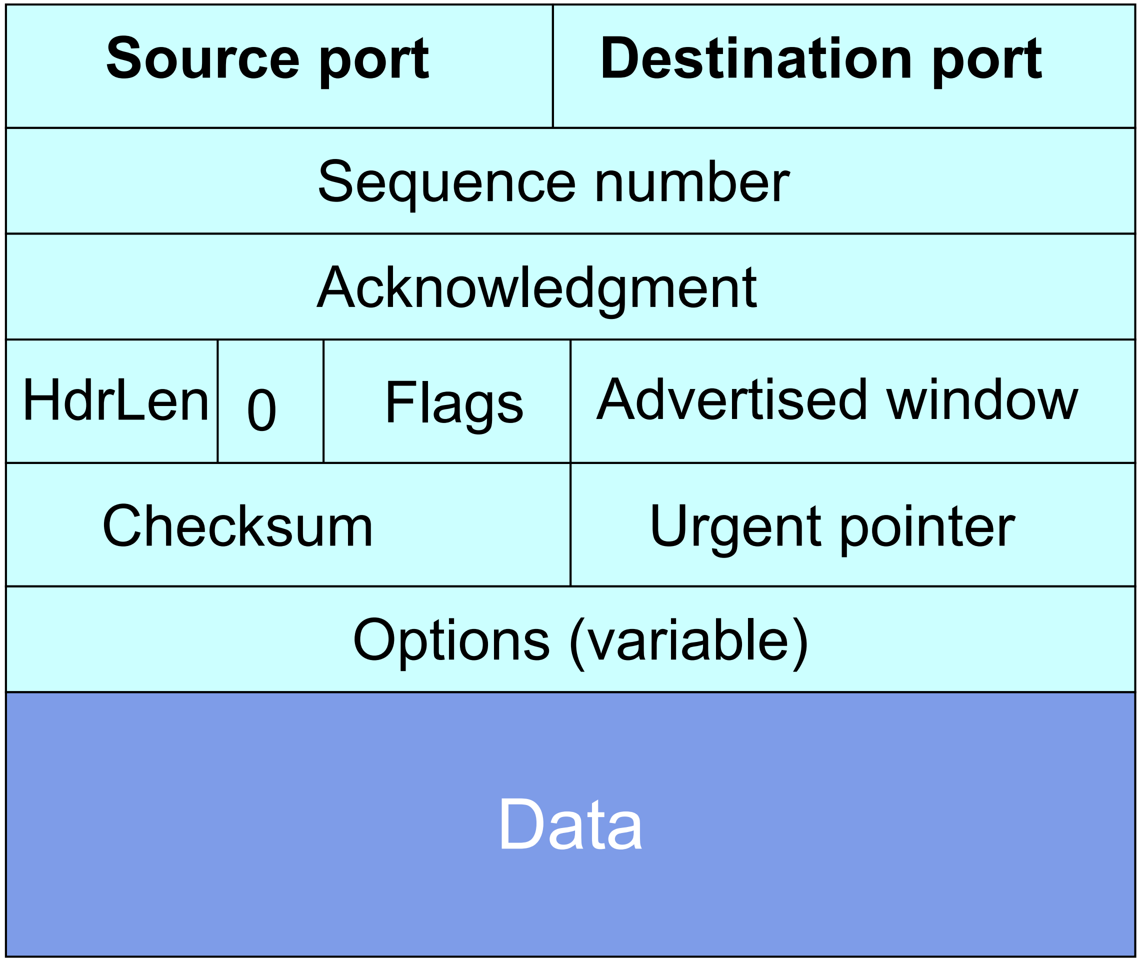
Transport protocol headers contain these port numbers.

Ports in UDP/TCP packets

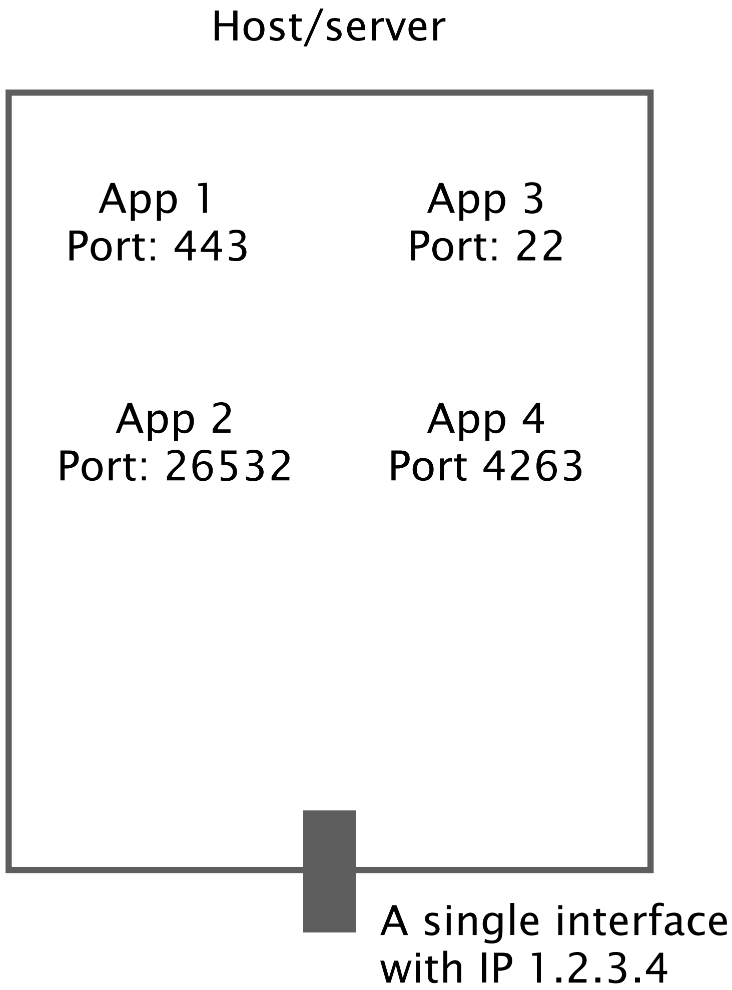
UDP



TCP

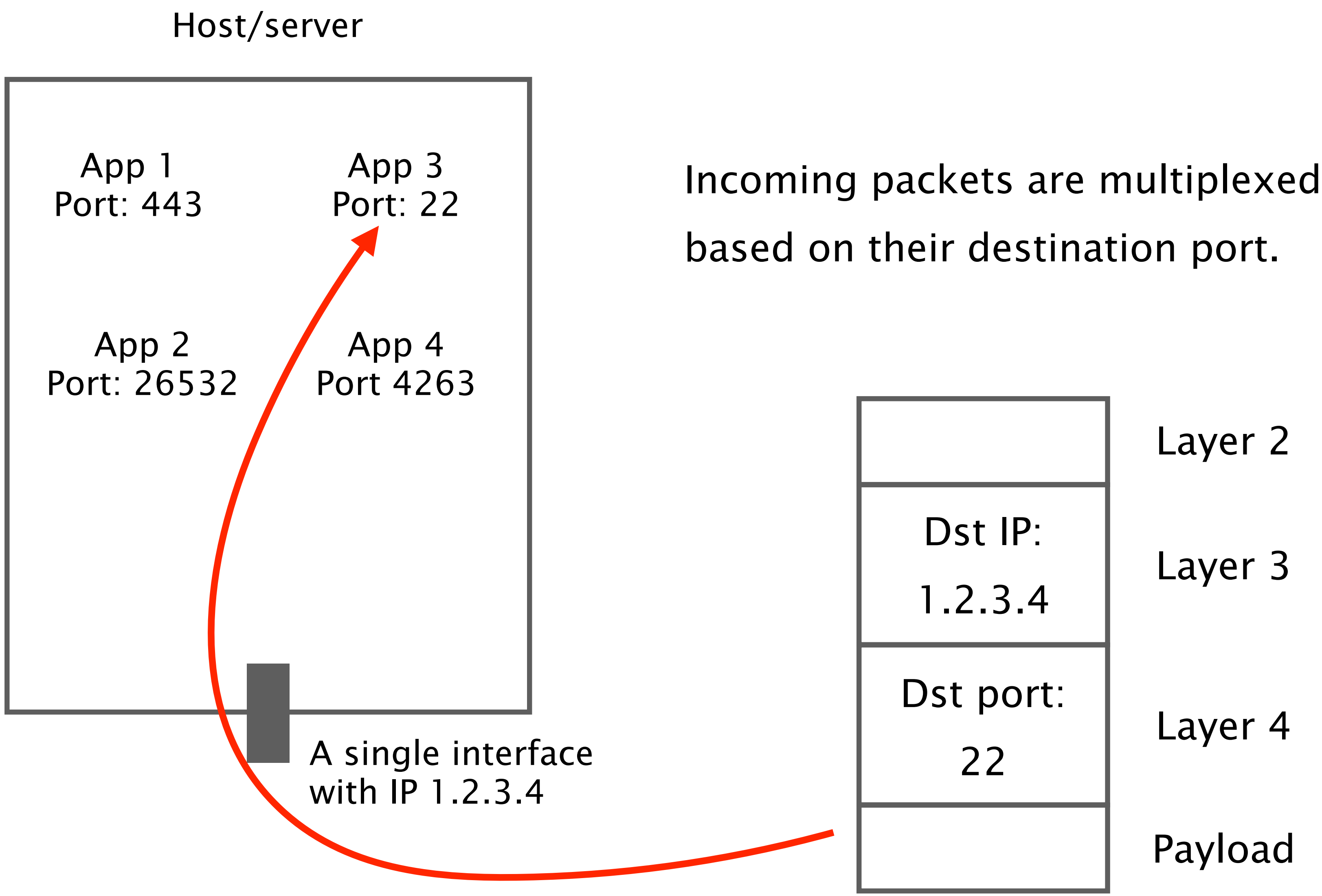


Logical (virtual) ports on a host



Incoming packets are multiplexed based on their destination port.

Logical (virtual) ports on a host



Incoming packets are multiplexed based on their destination port.

More on ports

Ports are 16-bit header fields (max port number: 2**16 - 1)

Ports 0-1023 are „well-known“

for example port 443 for HTTPS

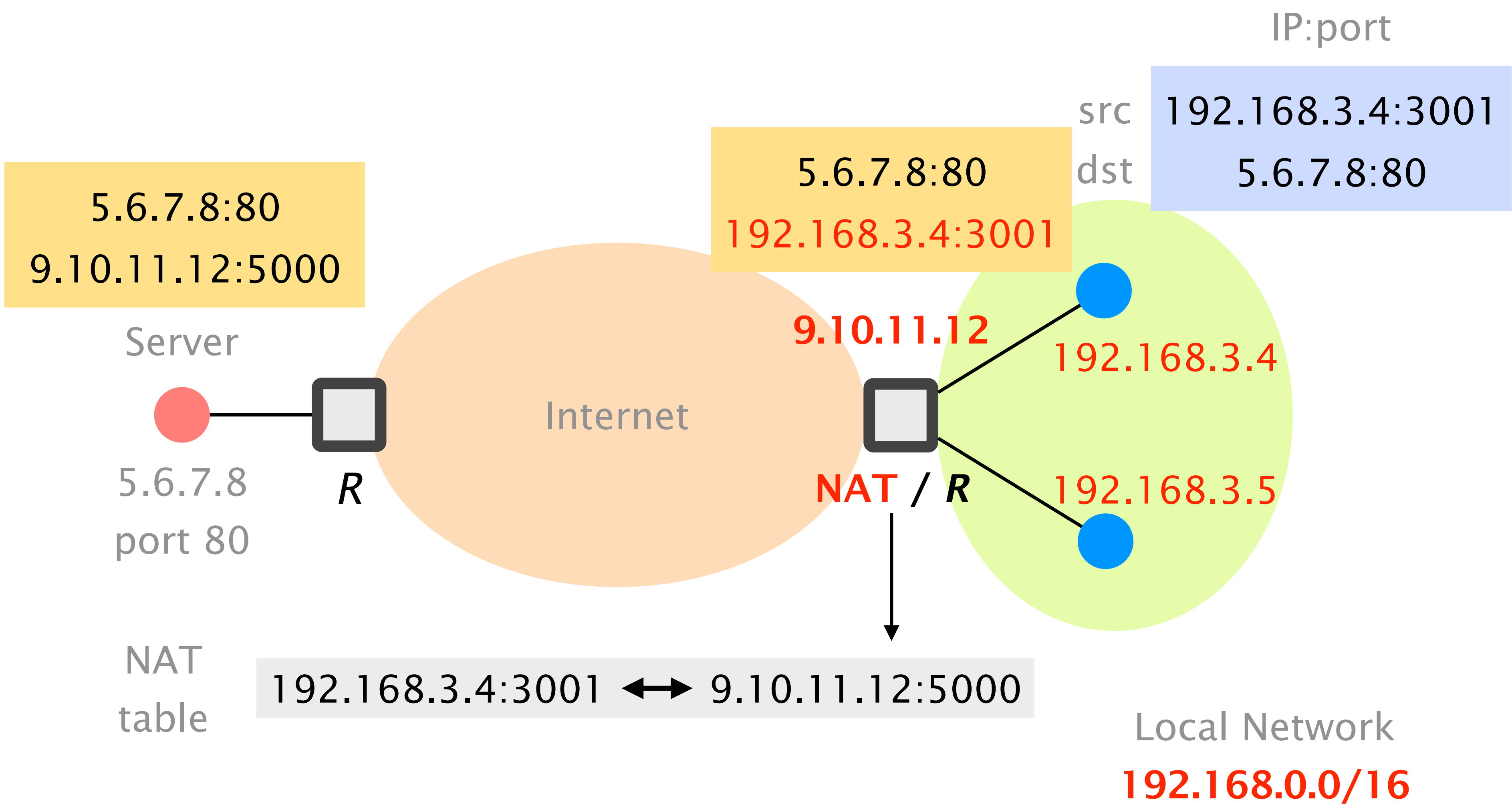
Ports 1024-65535 are so-called „ephemeral“ ports

given to clients (picked at random)

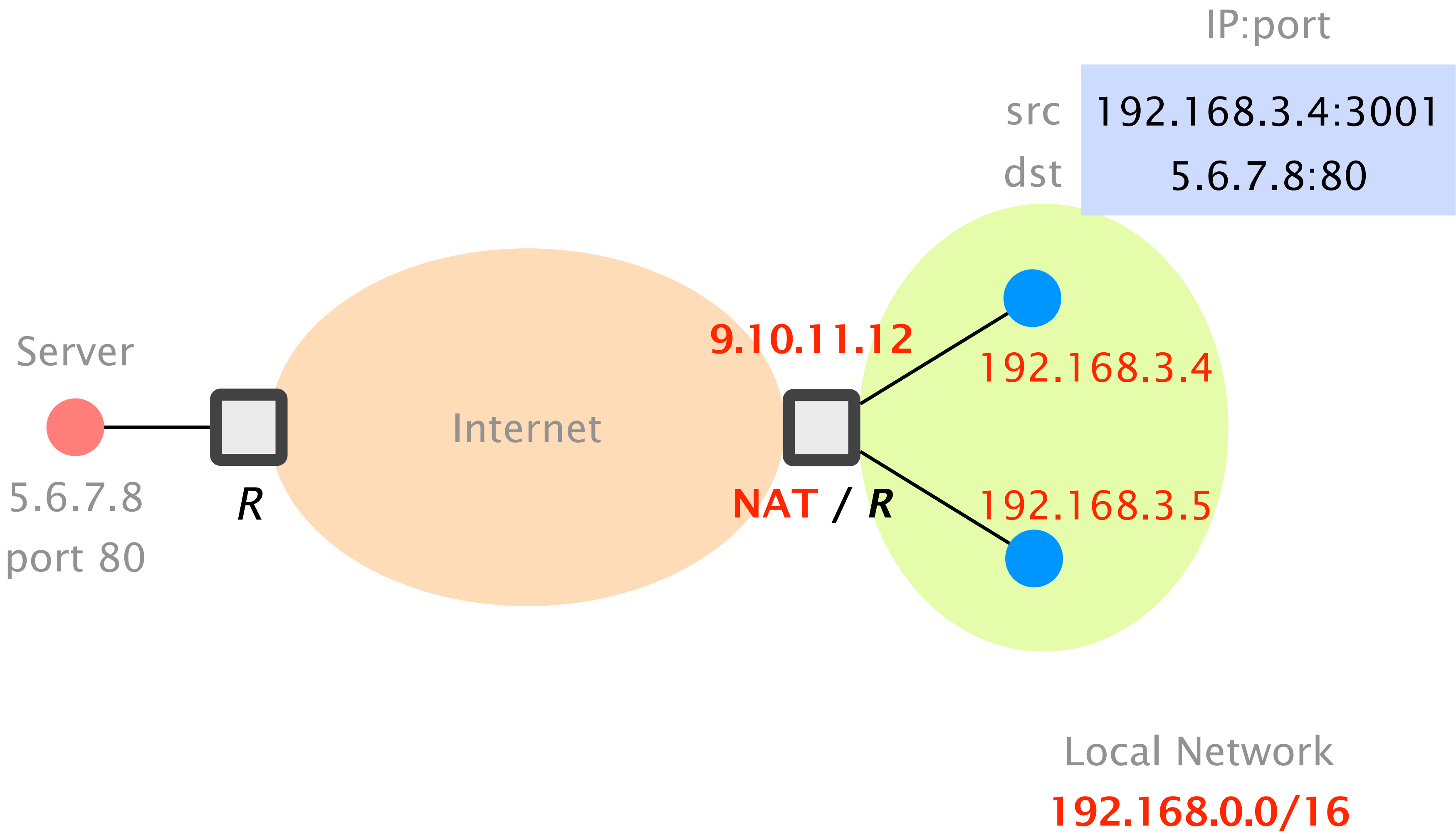
For more details look at the lecture slides

UDP and TCP, keywords: ports and sockets

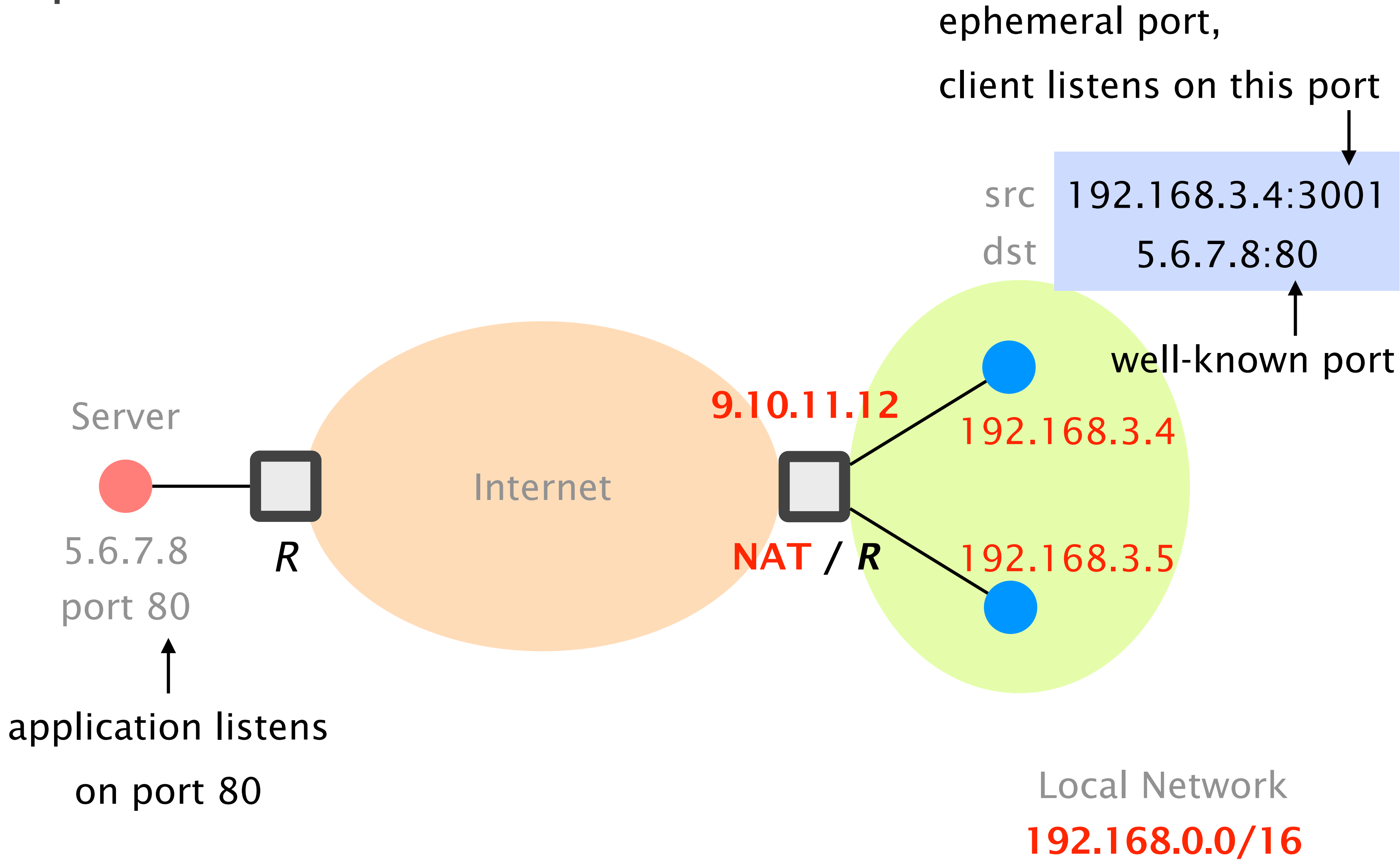
Example: The Internet with NAT (lecture week 5)



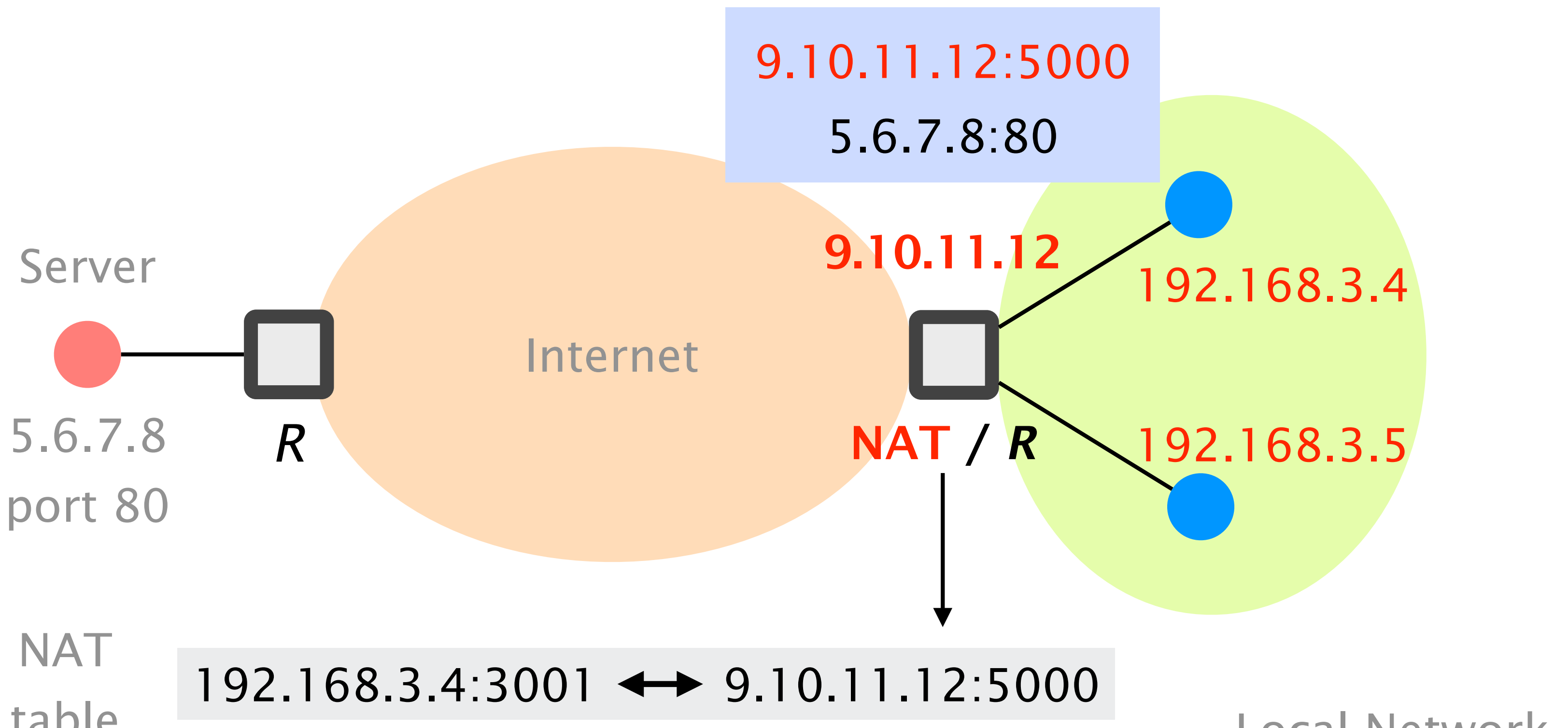
Example: The Internet with NAT



Example: The Internet with NAT



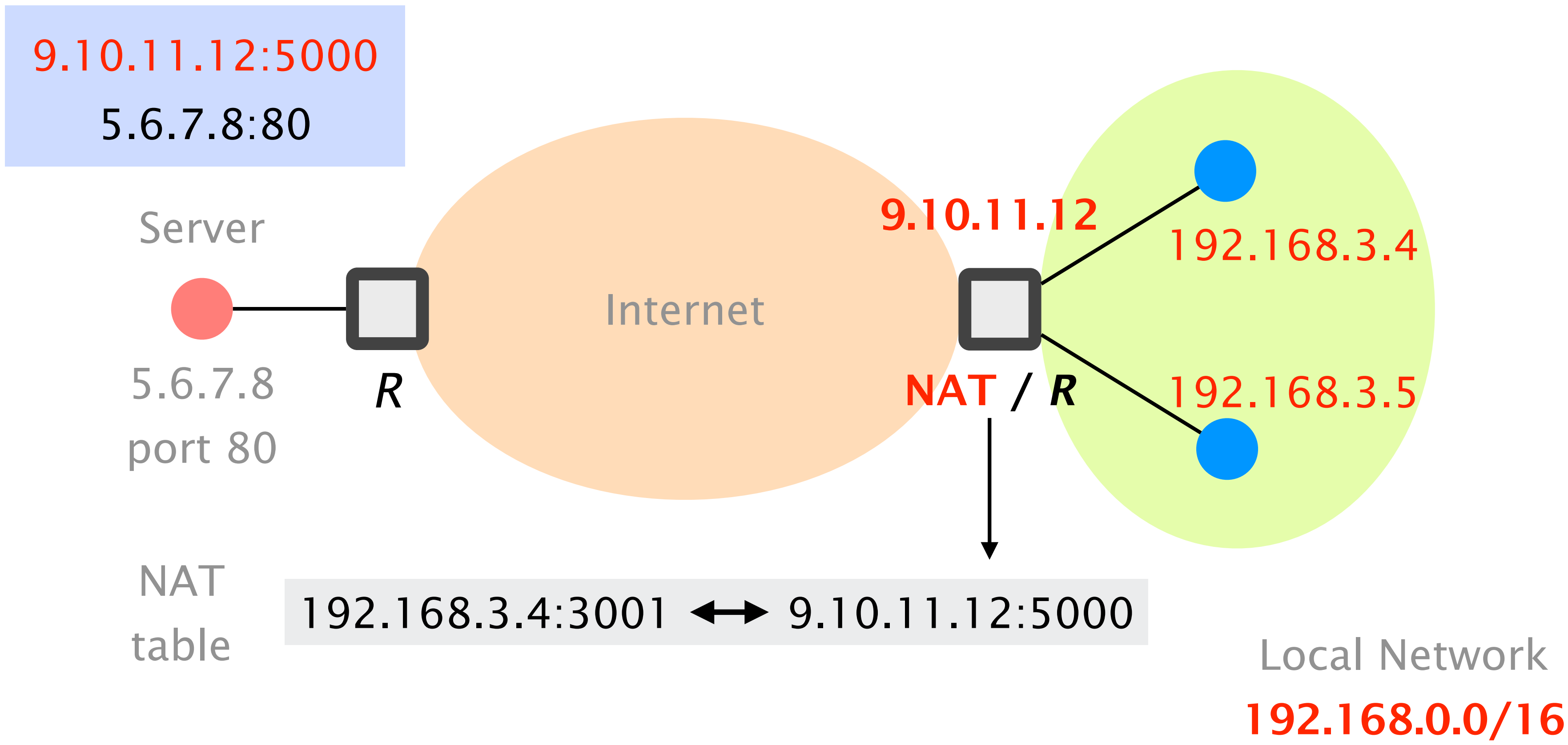
Example: The Internet with NAT



The NAT replaces source port and IP.
Port 3001 with a new port 5000 and
IP 192.168.3.4 with a public IP 9.10.11.12

Example: The Internet with NAT

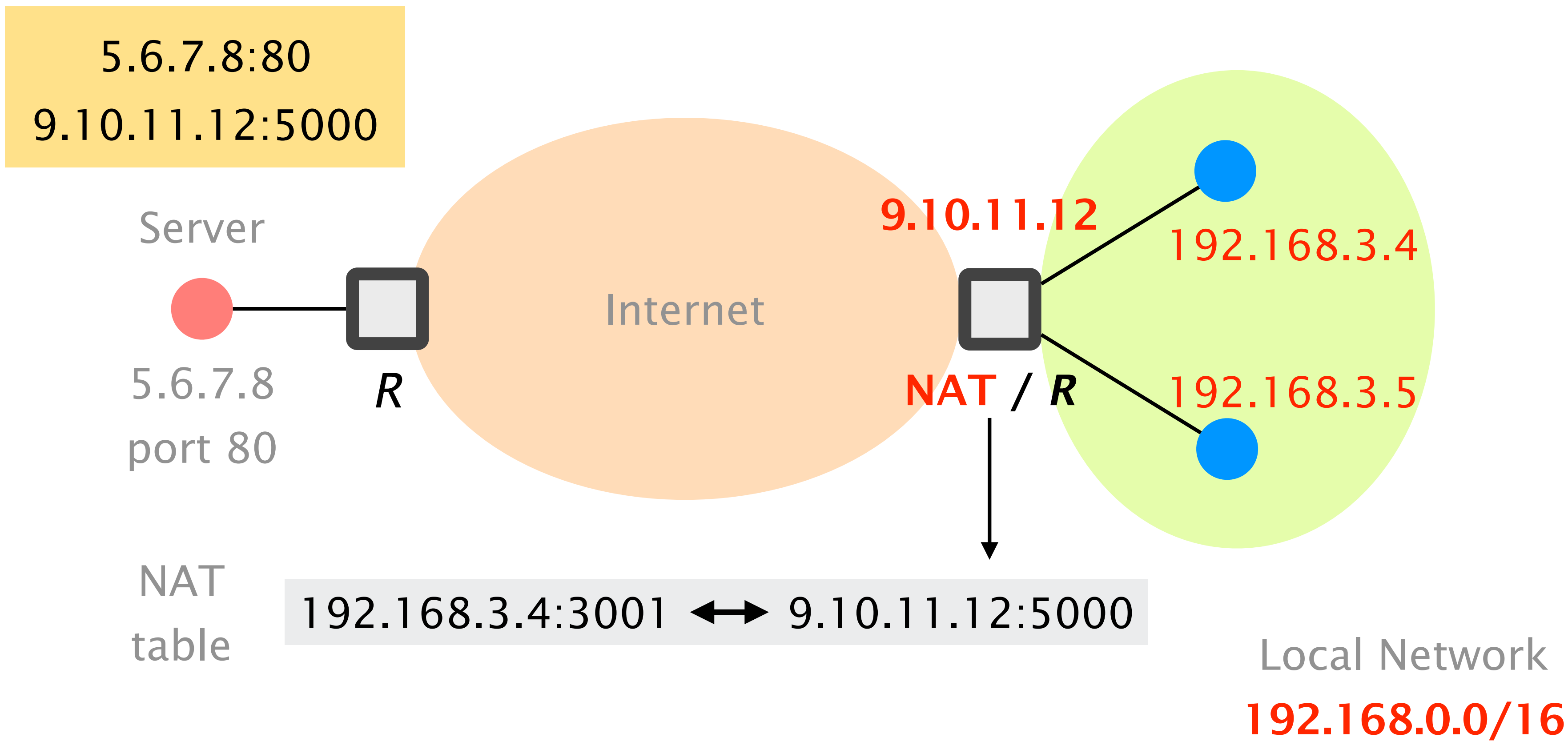
The packet reaches the correct application as it contains destination port 80



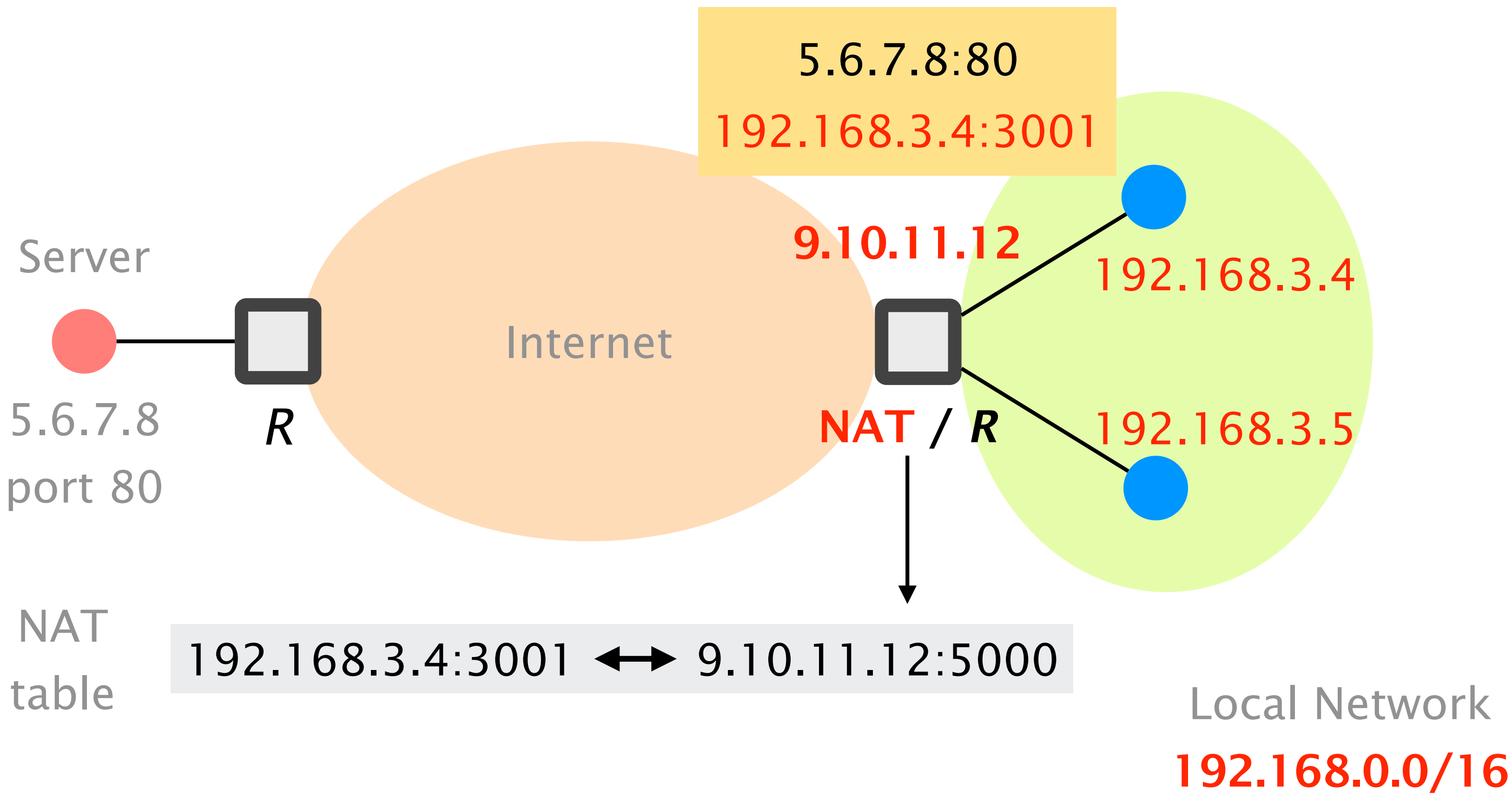
Example: The Internet with NAT

The answer from the server

goes towards destination port 5000



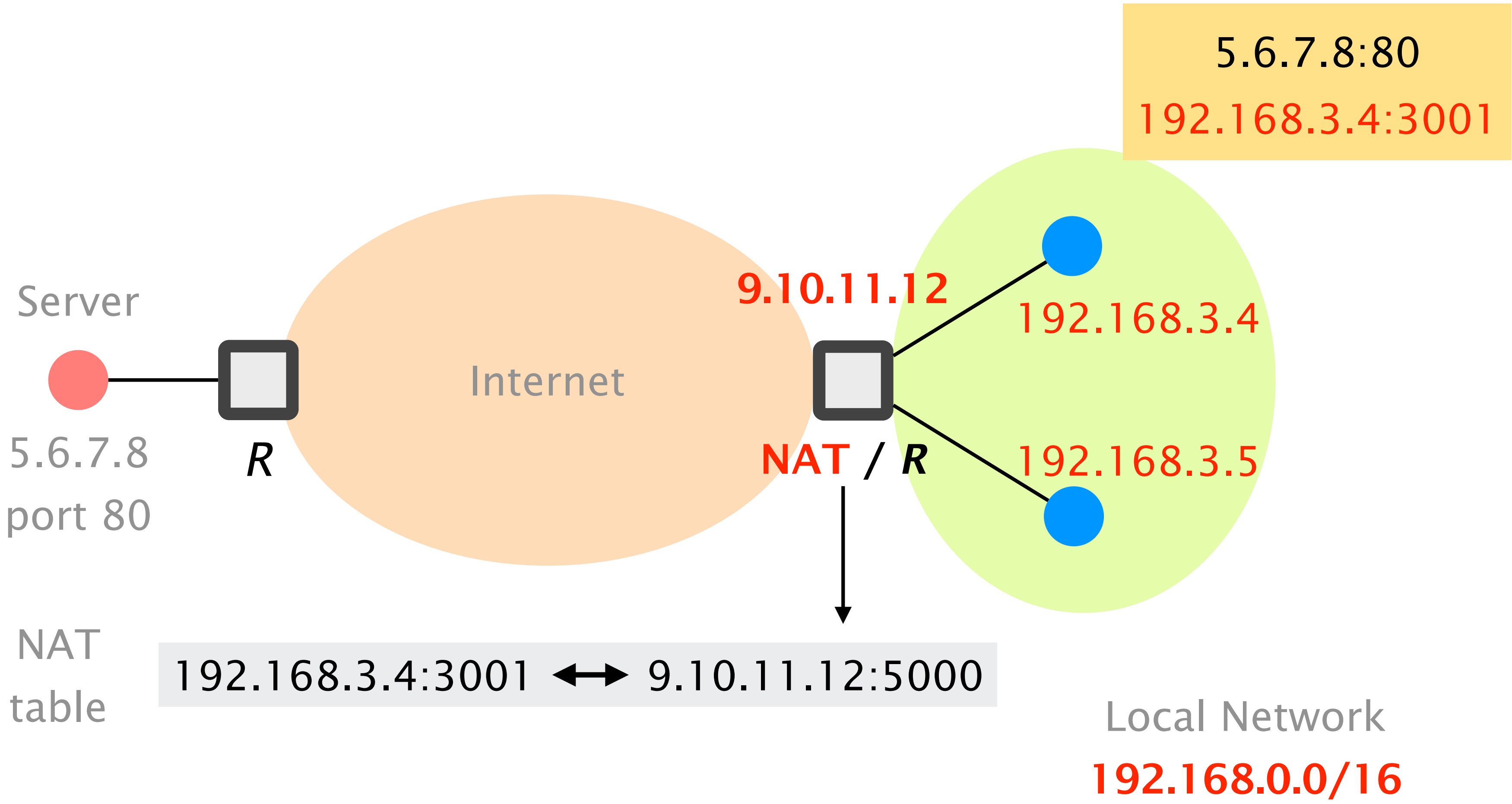
Example: The Internet with NAT



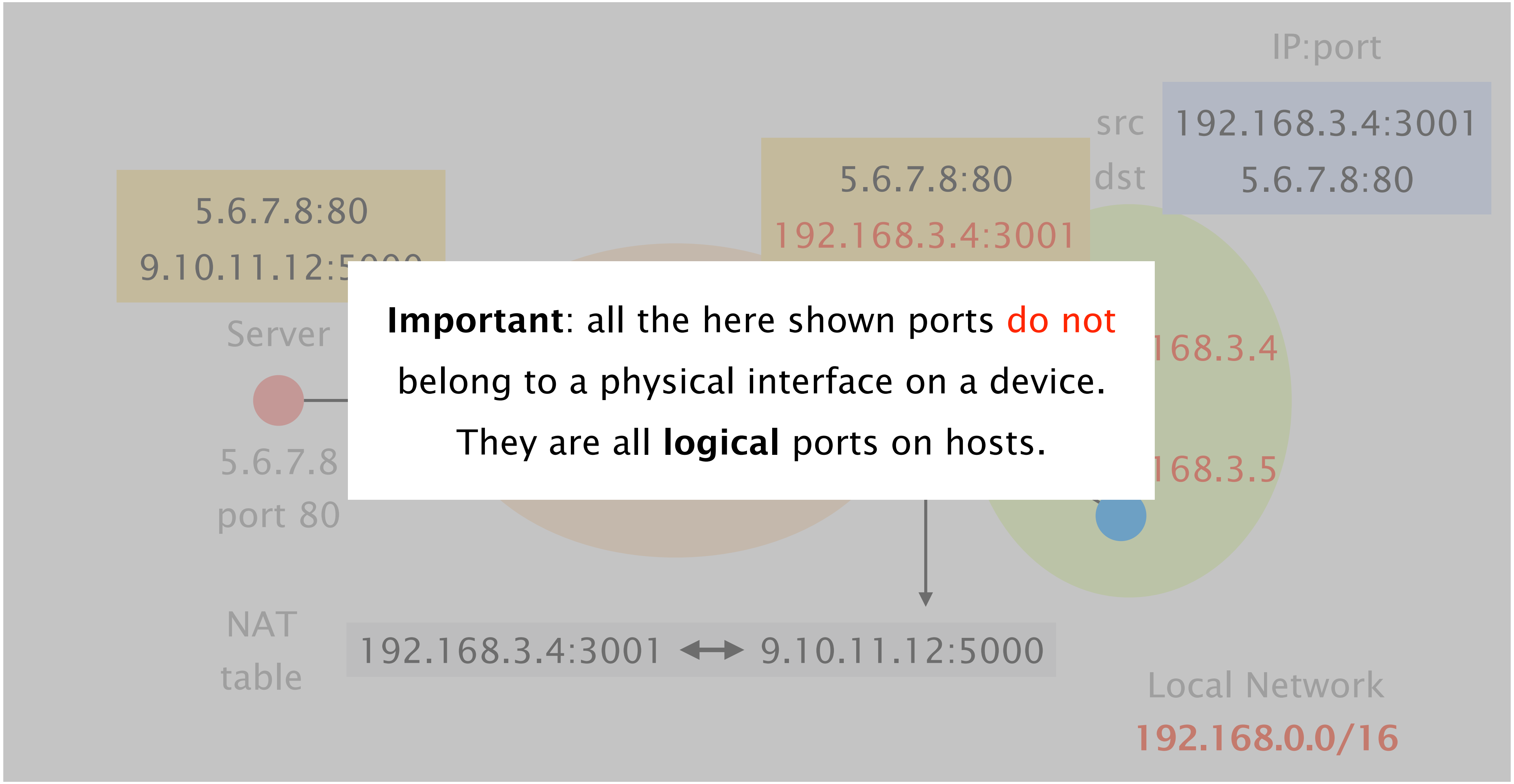
The NAT performs the reverse translation

Example: The Internet with NAT

The packet reaches the correct application on the client listening on port 3001



Example: The Internet with NAT



Important: all the here shown ports do not belong to a physical interface on a device. They are all **logical** ports on hosts.

Communication Networks

Exercise 9

Last week's exercise

Important lecture topics

Introduction to this week's exercise

Time to solve the exercise

Task 9.1: Reliable versus Unreliable Transport

Simple introduction question

Consider the information from the lecture slides

Task 9.2: Negative Acknowledgements

Instead of acknowledging what we received …

… the receiver could also acknowledge not-received data

In which scenarios does this (not) work well?

Task 9.3: Fairness

In this question we consider a **max-min fair allocation**

Have a look at lecture slides 78-81 in

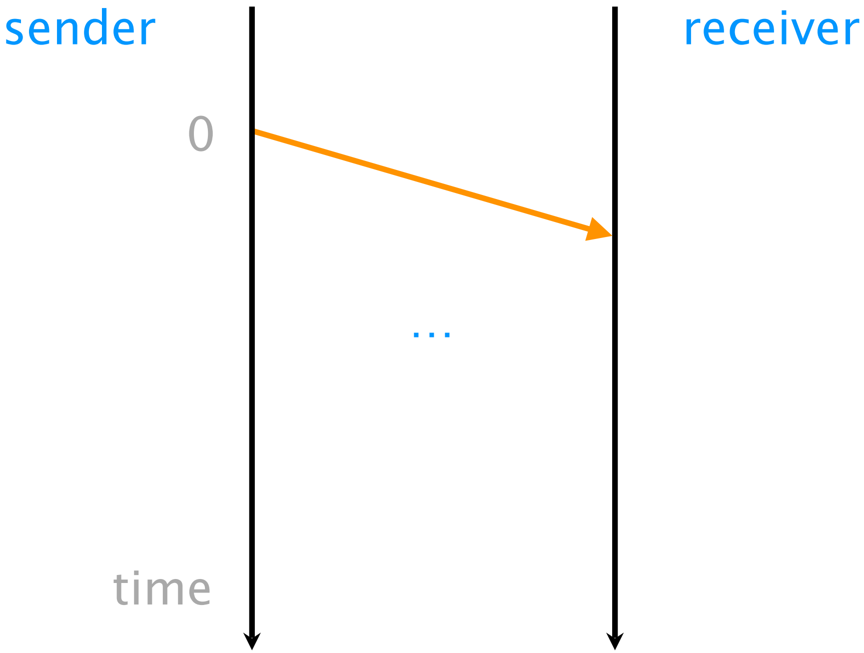
04_concepts_reliable_transport.pdf

Task 9.4: Understanding Go-Back-N's Behavior

Consult the introduction slides we just discussed

Task 9.5: Reliable Transport

Draw time-sequence diagrams



10 Mbps link

100 ms propagation delay

10000 bits in data segment

ACK size very small

Communication Networks

Exercise 9

Last week's exercise

Important lecture topics

Introduction to this week's exercise

Time to solve the exercise