

Network - Programming

Networks

Jan Krueger

jkrueger@cebitec.uni-bielefeld.de

Alexander Sczyrba

asczyrba@cebitec.uni-bielefeld.de

Overview

- Network-Protocols
- Protocol family TCP/IP
- Transmission Control Protocol (TCP)
- First steps with sockets

Computer Network

- Problem:
 - Send data from computer A to computer B
 - A and B are (indirectly) connected via a network
- Solutions:
 - One protocol for everything
 - Family of specialized protocols, that are stacked

Network Layers

application	ssh,SMTP,HTTP
transport	TCP, UDP
network	IP
link	Ethernet

link layer

- models physical connection
- Hardware addresses
- Data in frames
- Sensitive to errors
- Checksums
- Example: Ethernet, . . .

network layer

- modells link between different networks
- Packets inside of frames
- connectionless, unreliable, best effort
- routing
- Own address space
- Example: Internet Protocol (IP)
- Domain Name System (DNS) for “readable” addresses

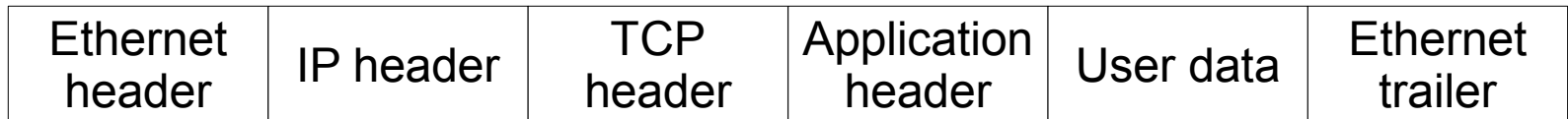
transport layer

- demultiplexing, port numbers
- well known ports, `/etc/services`
- User Datagram Protocol (UDP):
 - Similar to IP: datagramme, connectionless, unreliable
- Transmission Control Protocol (TCP):
 - Data stream
 - “Connection”
- Reliability through billing procedure
- flow control, congestion avoidance

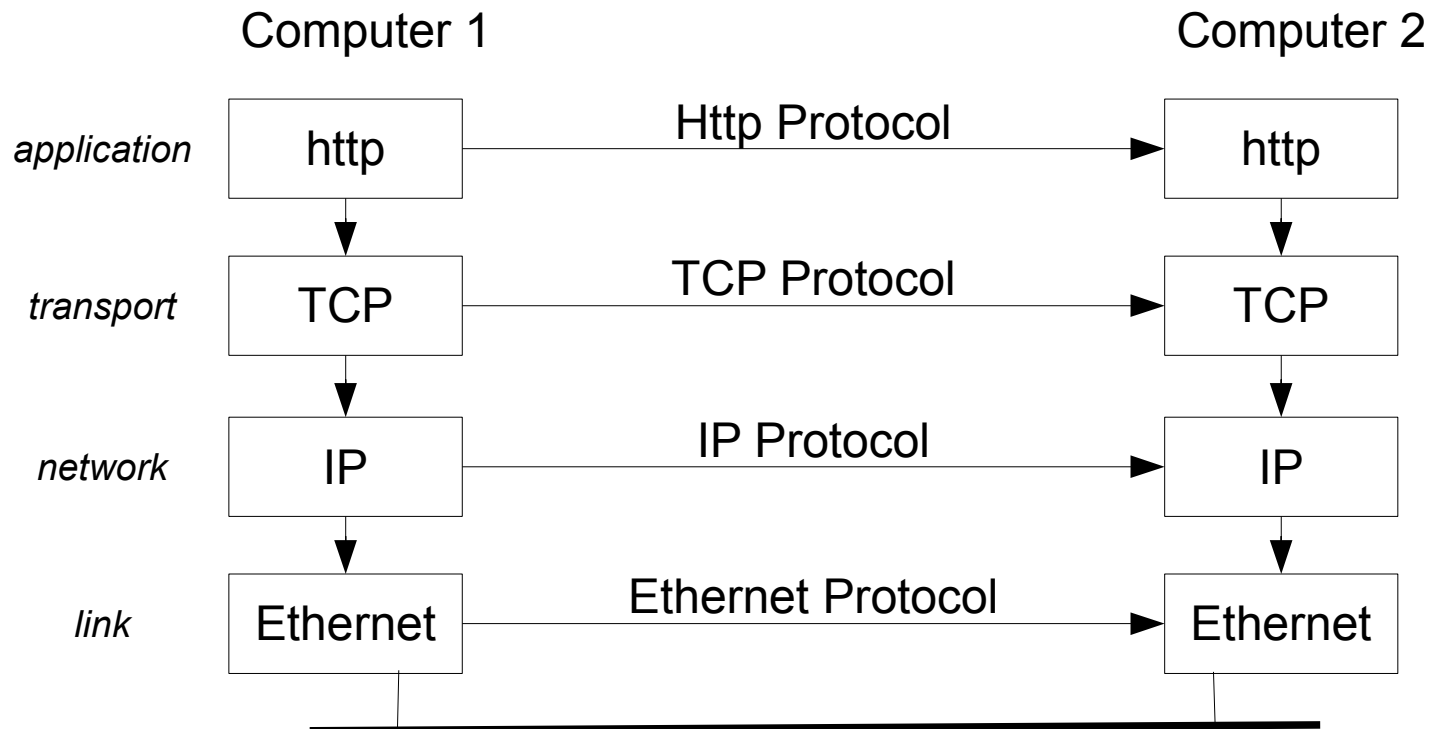
application layer

- Uses transport layer
- TCP or UDP relies on use case
- APIs:
 - Berkeley sockets
 - X/Open Transport Interface
 - . . .

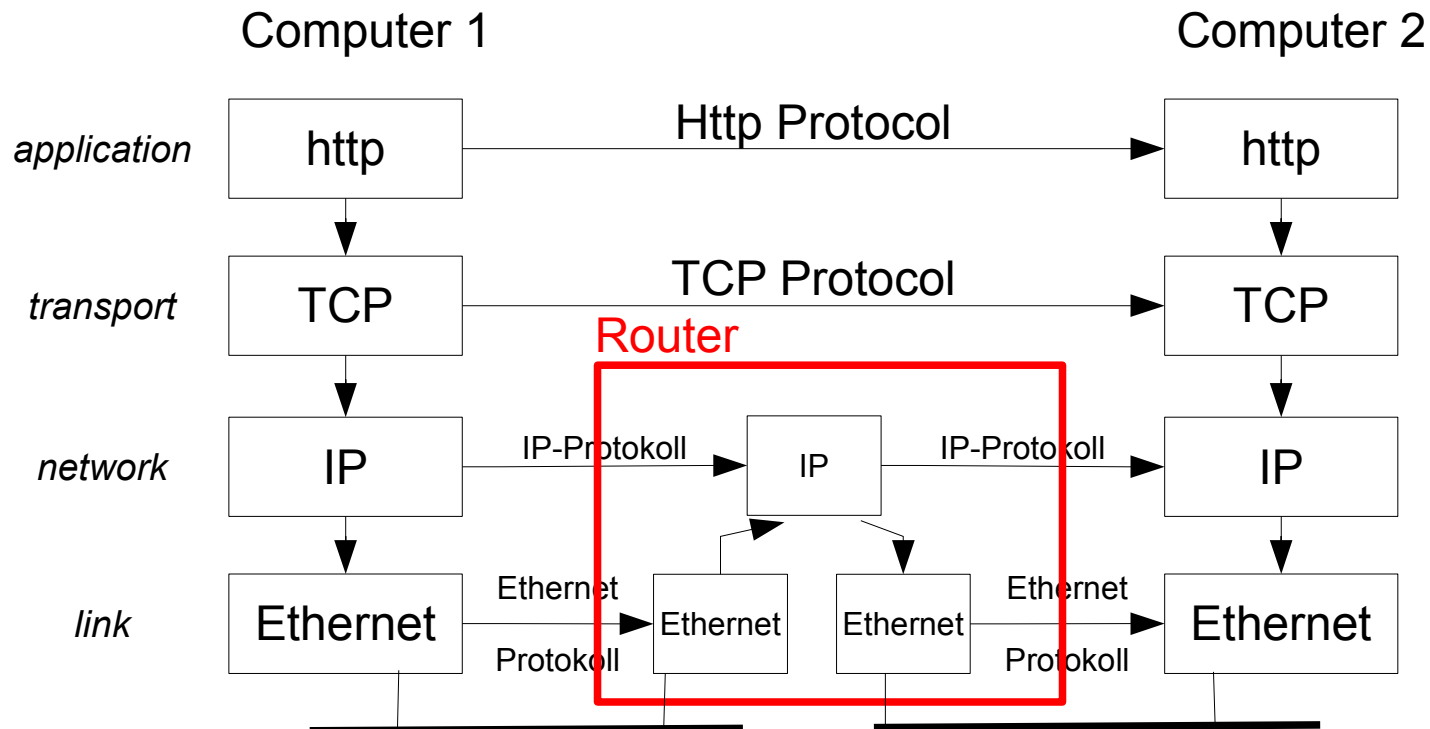
encapsulation



Communication in different Layers

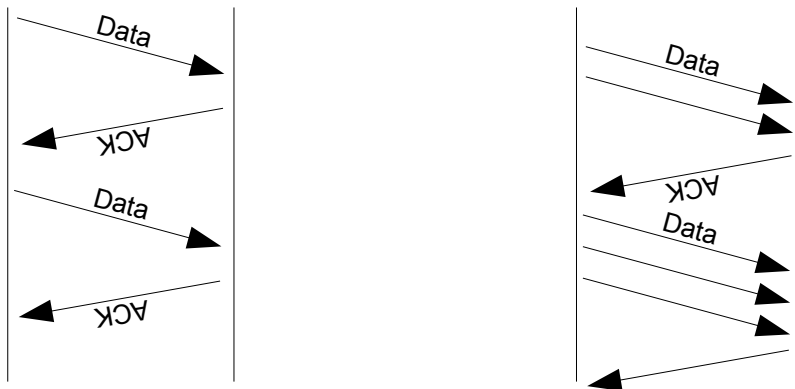


Routing

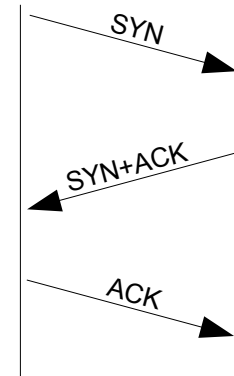
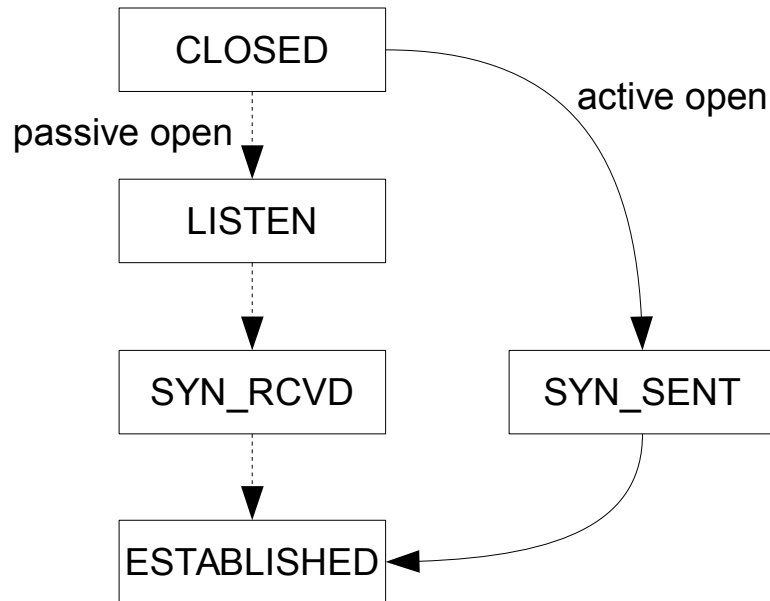


Transmission Control Protocol

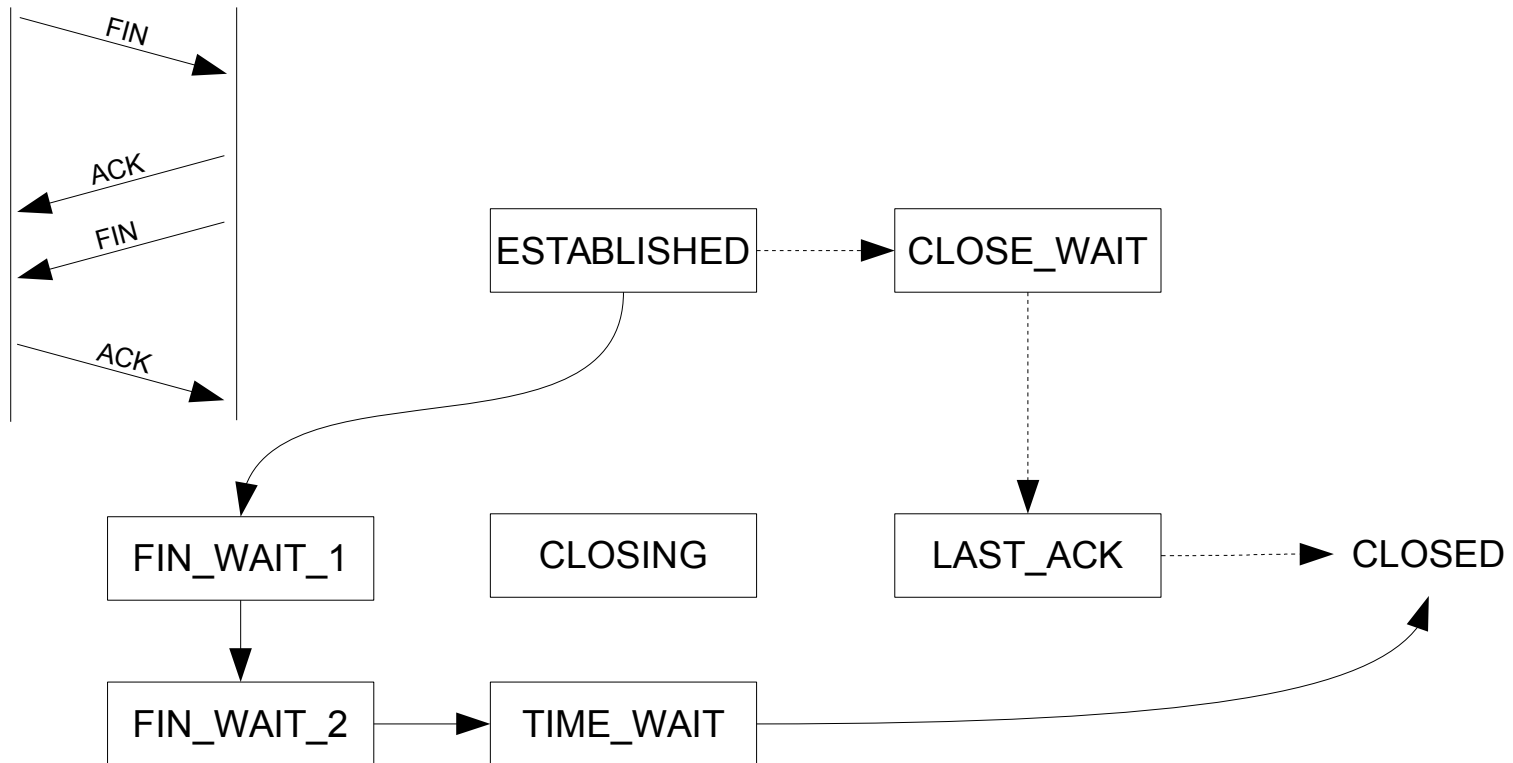
- Confirmation of recieved packets
- New delivery of lost packets
- sequence numbers



Starting a Connection



Closing a Connection



TCP Usage

- Connection identified by a socket pair:
(IP-AdresseL, PortL, IP-AdresseR, PortR)
- Show connection details:

```
$ netstat -A inet | more
```

```
...
```

```
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 leonardo.TF:1018 kaylee.TF:nfs ESTABLISHED
tcp 0 0 leonardo.TF:50657 simon.TF:nfs TIME_WAIT
tcp 0 0 leonardo.TF:38896 donatello.TF:ssh ESTABLISHED
```

```
...
```

```
(TF = TechFak.Uni-Bielefeld.DE)
```

Hands on!

- Open a ssh connection to another computer, e.g. from goldfinger to compute.linux. Observe the output in two different terminals on every computer of netstat. Which entries will be added?
- Close the connection. How is the output of netstat altered?

Hands on!

- You can find the scripts `server.pl` and `client.pl` in the archive `uebung4.tar.gz`. (Extract the content of the archive with `tar -xvf uebung4.tar.gz`)

From now on you need three open terminals.

- Start the server. You have to specify a port number:

```
$ server.pl 54321
```

- Have a look at the current status of the server:

```
$ netstat -a | grep 54321
```

Hands on!

- Let the client read data from the server:

```
$ client.pl 54321
```

Use meanwhile netstat for analyzing the connection.

- Start the client two times more. What will be changed in every call?
- Start the client with a fixed port number:

```
$ client.pl 54321 55443
```

Repeat the call again twice.

Hands on!

- Start more clients simultaneously:

```
client.pl 54321 & [enter]  
client.pl 54321 & [enter]  
client.pl 54321 & [enter]
```

What can you observe?

- Start two clients with the same fixed port number simultaneously:

```
client.pl 54321 55443 & [enter]  
client.pl 54321 55443 & [enter]
```

What happens?

- Quit the server and try restart it instantly. What happens?
- What will happen, if you quit the client with CTRL+C while reading data from the server?