Netzwerk - Programmierung

Netzwerke

Jan Krüger
jkrueger@cebitec.uni-bielefeld.de

Alexander Sczyrba
asczyrba@cebitec.uni-bielefeld.de

Übersicht

- Netzwerk Protokolle
- Protokoll Familie - TCP/IP
- Transmission Control Protocol (TCP)
- Erste Schritte mit Sockets

Computer Netzwerke

- Problem:
  - Daten von Computer A nach Computer schicken
  - A und B sind über ein Netzwerk (indirekt) verknüpft
- Lösung:
  - **ein allgemeines Protokoll**
  - spezialisierte Protokolle die darauf aufbauen bzw. darin verpackt werden.

Netzwerk Schichtenmodell (TCP/IP)

4	Applikation	ssh,SMTP,HTTP
3	Transport	TCP, UDP
2	Netzwerk	IP
1	Link	Ethernet

Verbindungsschicht

- models physical connection
- Physikalische Verbindung
- Hardware Adressen
- Datenframes
- fehleranfällig
- Checksummen
- Beispiel: Ethernet, . . .

Netzworkschicht

- verbindet unterschiedliche Netzwerke
- Pakete innerhalb von Frames
- verbindungslos, unzuverlässig, „best effort“
- Routing
- Eigener Addressierungsraum
- Beispiel: Internet Protokol (IP)
- Domain Name System (DNS) für „lesbare Addressen“

Transportschicht

- Demultiplexing durch Ports
- bekannte Ports → `/etc/services`
- User Datagram Protocol (UDP):
  - Datagramme, verbindungslos, unzuverlässig
  - schnell (kaum Overhead)
- Transmission Control Protocol (TCP):
  - Datastrom, verbindungsorientiert, zuverlässig
  - langsam(er)
- Fluss- und Staukontrolle

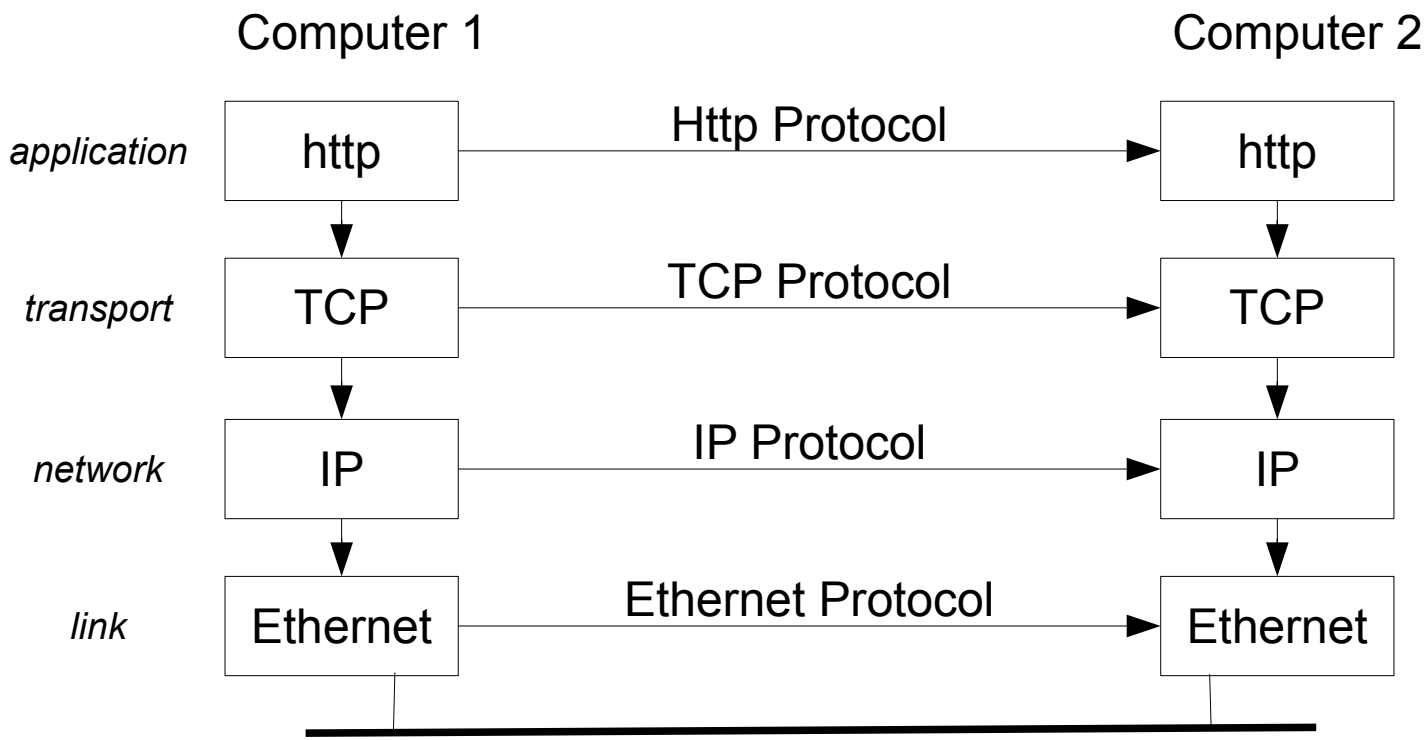
Applikationschicht

- Baut auf der Transportschicht auf
- TCP oder UDP hängt vom Anwendungsfall ab
- APIs:
  - Berkeley sockets
  - X/Open Transport Interface
  - . . .

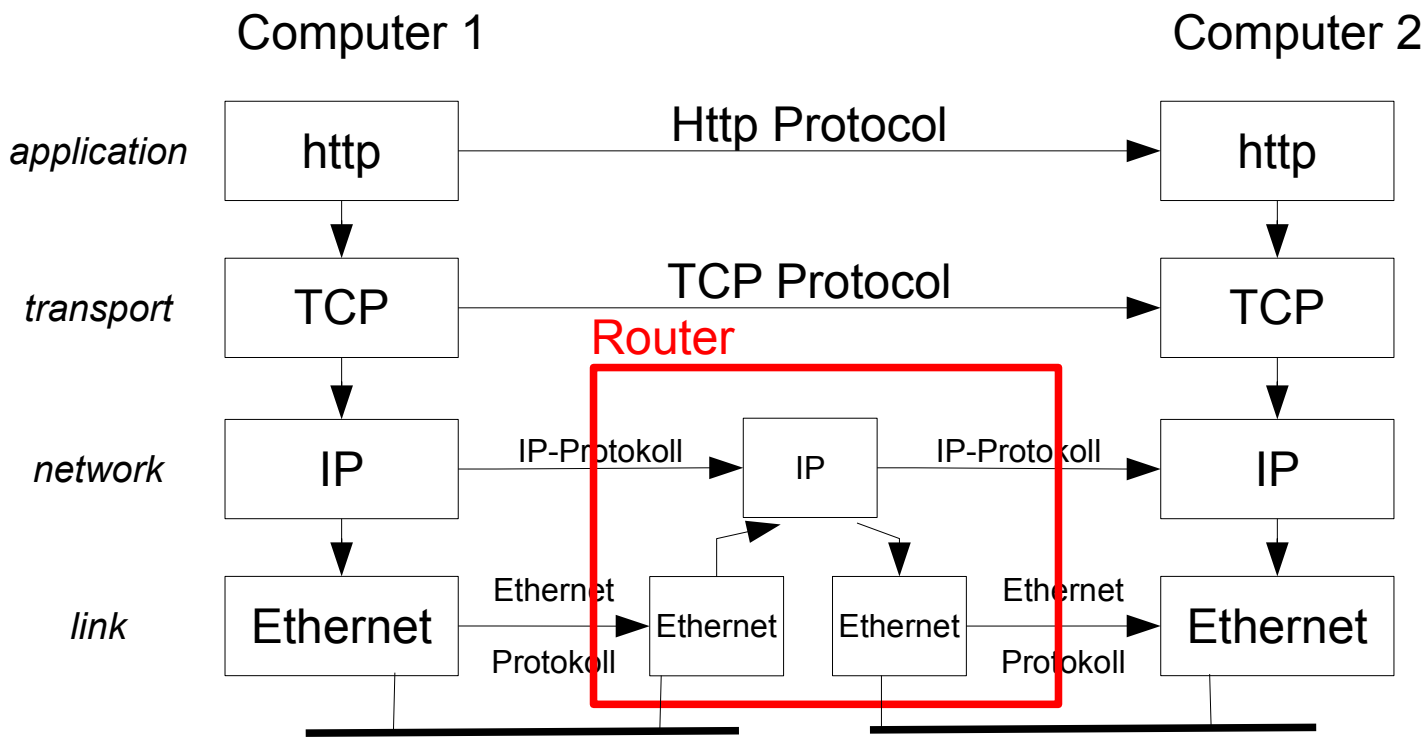
Kapslung


Ethernet header
IP header
TCP header
Application header
User data
Ethernet trailer

Kommunikation zwischen den Netzwerkschichten

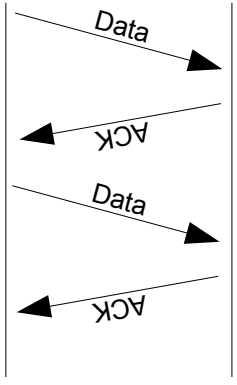


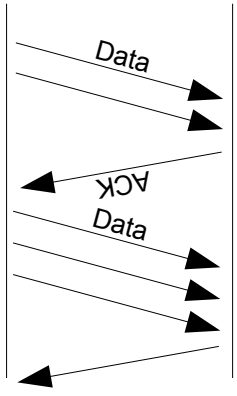
Routing


Computer 1
Computer 2
application
transport
network
link
http
Http Protocol
http
TCP
TCP Protocol
TCP
Router
IP
IP-Protokoll
IP
IP-Protokoll
IP
Ethernet
Ethernet
Protokoll
Ethernet
Ethernet
Ethernet
Protokoll
Ethernet

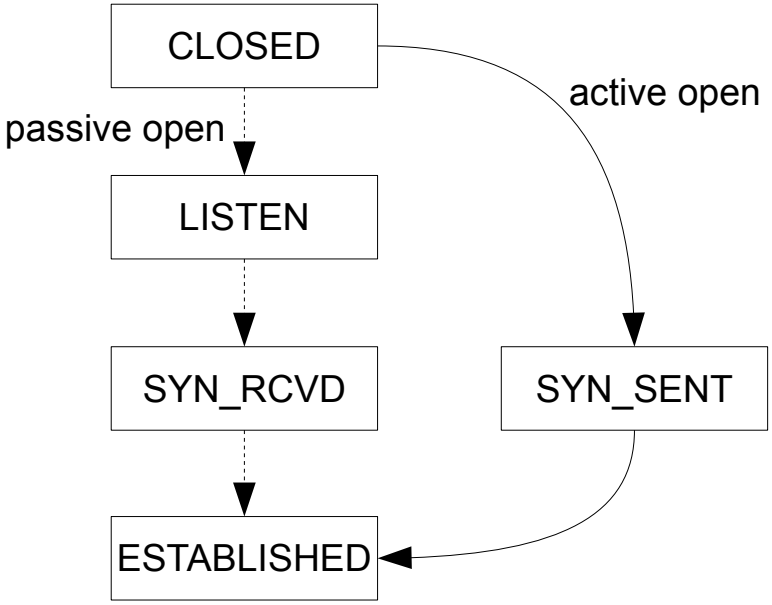
Transmission Control Protocol (TCP)

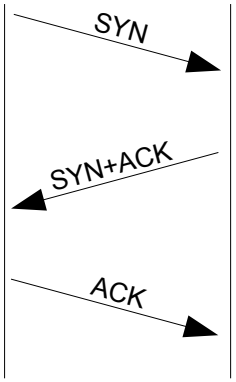
- Bestätigung von erhaltenen Paketen
- Fehlende Pakete werden erneut verschickt
- Pakete werden nummeriert



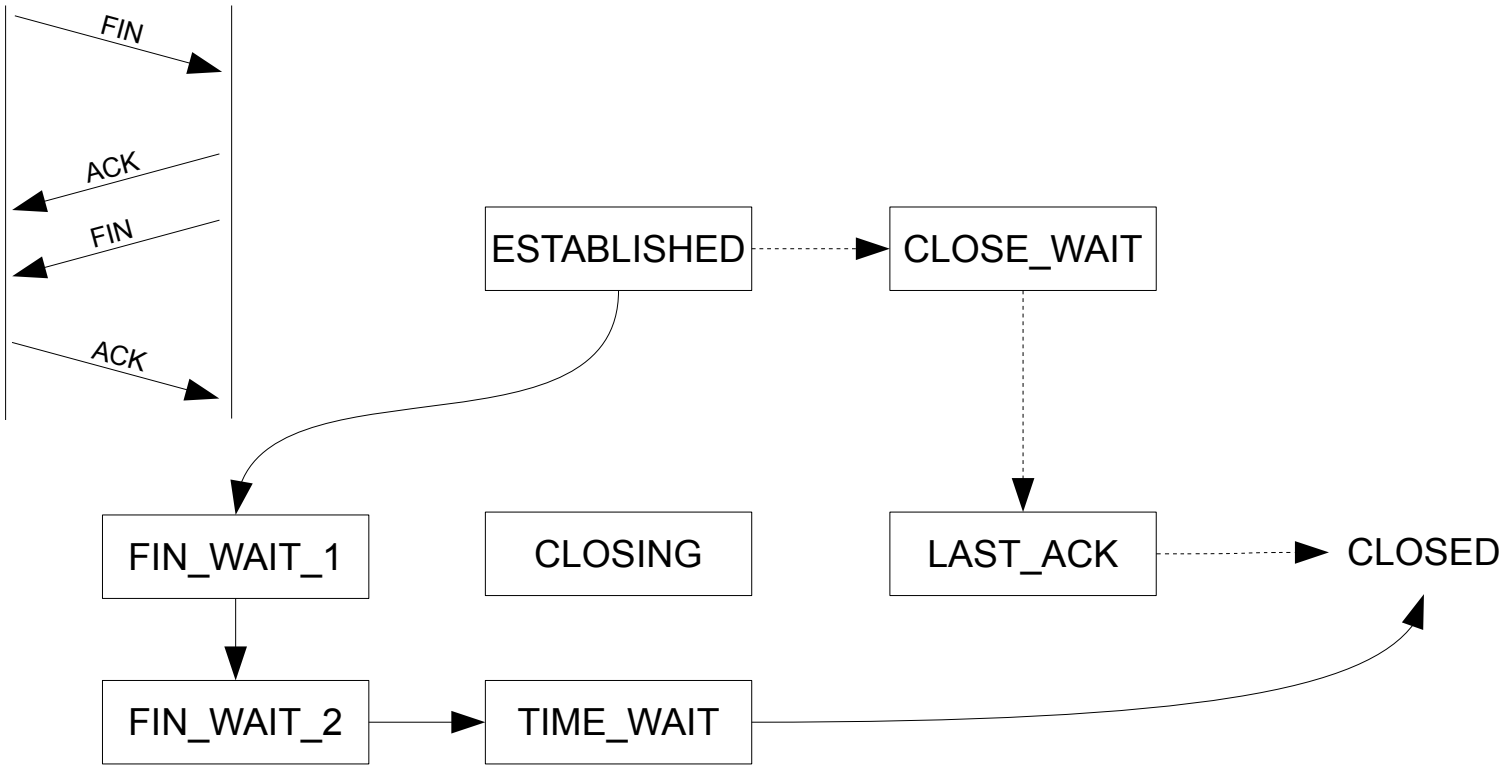


Verbindungsaufbau





Verbindungsabbau



TCP

- Verbindungen werden durch ein Socket Paar repräsentiert
  (lokale IP-Addresse, lokaler Port), (remote IP-Addresse, remote Port)
- Verbindungsdetails bzw. der Netzwerkstatus lassen sich durch `netstat` im Terminal analysieren.
  z.B. : Anzeigen aller IPv4 TCP Verbindungen
  - BSD Style:
    ```
    $ netstat -a -f inet -p tcp -n
    ```
  - GNU Style:
    ```
    $ netstat -a -t -n -4
    ```

Übung

- Mach eine ssh Verbindung zu einen anderen Computer (z.B. compute.linux auf. Untersuche die Verbindung in zwei Terminals mittels **`netstat`**. Welche Verbindungen w(e|u)rden hinzugefügt ?
- Schliess die SSH-Sitzung. Wie ändert sich die Ausgabe ?
- *Hinweis: Für diese Übung sind die Shell Befehle **`netstat`** und **`watch`** hilfreich (man netcat bzw. man watch)*

Übung!

- In dem Material (`Netzwerke.tar.gz`) zu den Folien findest Du zwei Skripte `server.pl` and `client.pl`.

  Ab jetzt brauchst Du 3 Terminals (Server, Client und Monitor)

- Starte den Server. Du musst einen Port angeben.

```
$ perl server.pl 54321
```

- Schau Dir den Netzwerkstatus an.

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

Übung !!

- Lies mit Hilfe des Clients ein paar Daten vom Server.

  ```
  $ perl client.pl 54321
  ```

  Beobachte währendessen den Netzwerkstatus
- Starte den Client mehrmals nacheinander ? Was ändert sich ?
- Starte den Client mit einem festen Port:

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

  Wiederhole den Vorgang ? Was ist der Unterschied zu den vorherigen Aufruf ?

Übung !!!

- Starte mehre Clients gleichzeitig

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

  Was stellt Du fest ?

- Starte zwei Client mit einem gleichen feste Port gleichzeitig ?

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

  Was passiert ?

- Beende den Server und starte in sofort wieder neu .
- Was passiert, wenn Du den Server beendest während Daten gelesen werden ?