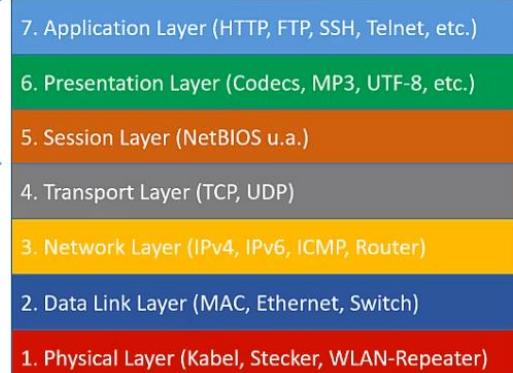


## Zusammenfassung Tag 25

### Einführung in die Netzwerk-Kommunikation

- Protokolle regeln Kommunikationsabläufe
- TCP/IP ist ein Protokollstapel (Stack)
- Kommuniziert wird über Netzwerkadressen (MAC, IP)
- Namen erleichtern das Merken von Adressen
- DNS ist der wichtigste Namensraum
- DHCP ermöglicht die automatische Zuweisung von IPv4- und IPv6-Adresskonfigurationen
- Private Adressen werden in internen Netzwerken eingesetzt:  
192.168.x.y, 172.16-31.x.y, 10.x.y.z
- Network Address Translation (NAT) übersetzt die privaten Adressen in öffentliche Adressen im Internet
- Router vermitteln zwischen Netzwerken
- Switches dienen zum Anschluss von Endgeräten im kabelgebundenen Netzwerk
- Kabelgebundene lokalen Netze (LANs) werden über Ethernet (IEEE 802.3) realisiert
- WLAN (IEEE 802.11a/b/g/n/...) ist die kabellose Alternative

ISO-OSI-Referenzmodell



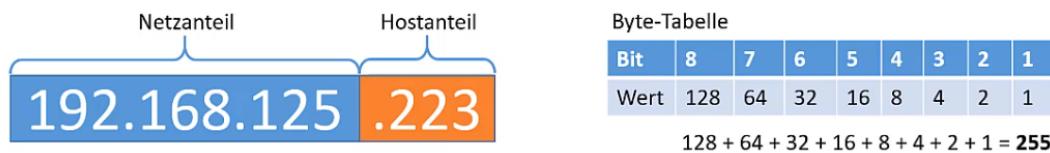
### Wireshark installieren und starten

- Wireshark ist ein Netzwerksniffer, der alle Datenpakete auf dem Netzwerk mit schneiden kann, die an der eigenen Schnittstelle hereinkommen oder herausgehen
- Wireshark wird für Windows, Linux/Unix und MacOS angeboten
- Wireshark erfordert eine grafische Oberfläche (z.B. GNOME oder KDE)
- Wireshark unterstützt zahlreiche Filterregeln, sowohl während des Mitschnitts (Capture-Filter) als auch danach (Display-Filter)
- Wireshark stellt mitgeschnittene Pakete in drei Fenster dar:
  - Paketliste: enthält jedes einzelne Paket
  - Details: Enthält die Interpretation der Daten
  - Rohdaten: Enthält eine hexadezimale Darstellung
- ICMP wird für die Übermittlung von Fehler- und Statusmeldungen genutzt
- Ping nutzt ICMP Echo Request (Typ 8) und Echo Reply (Typ 0)

## Einführung in die IPv4-Adressierung

# Einführung in die IPv4-Adressierung

- Jede IPv4-Adresse besteht aus 4 Oktetten (=Bytes) durch Punkte voneinander getrennt
- Jedes Oktett kann Werte zwischen 0 und 255 annehmen
- Die IPv4-Adresse ist in einen Netz- und einen Hostanteil aufgeteilt
- Systeme, deren IP-Adressen denselben Netzanteil haben, befinden sich im selben Subnetz
- Systeme in unterschiedlichen Subnetzen müssen über Router kommunizieren



Netzklassen nach RFC 791

Klasse	High Order Bits	1. Oktett	Netzanteil	Beispiel	Bemerkung	Hosts pro Netz
A	0	1-127	8 Bit	109.23.17.222	regulär nutzbar	16,7 Mio
B	10	128-191	16 Bit	160.0.113.7	regulär nutzbar	65534
C	110	192-223	24 Bit	213.255.5.117	regulär nutzbar	254
D	1110	224-239	n.a.	224.0.0.5	Multicast	n.a.
E	1111	240-255	n.a.	Nicht relevant	experimentell	n.a.

# Einführung in die IPv4-Adressierung

- Die erste und die letzte Adresse in einem Netz sind reserviert:
- 1. Adresse = Subnetzadresse
- Letzte Adresse = Broadcastadresse

### Beispiel 1:

Subnetzadresse: 192.168.125.0

Reguläre Adressen: 192.168.125.1 bis 192.168.125.254

Broadcastadresse: 192.168.125.255

### Beispiel 2:

Subnetzadresse: 172.16.0.0

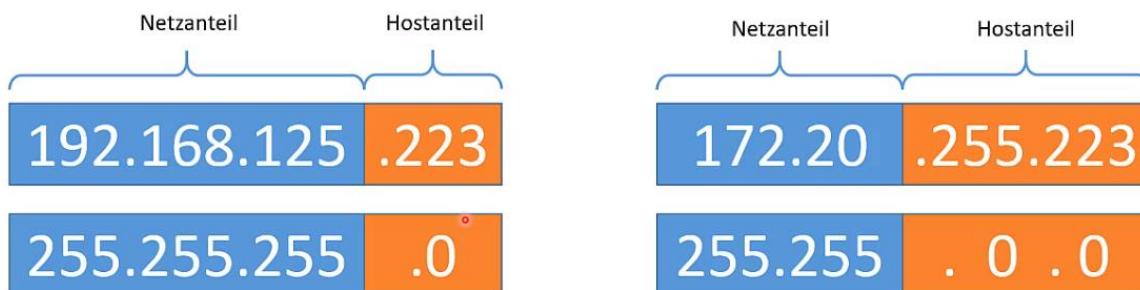
Reguläre Adressen: 172.16.0.1 bis 172.16.255.254

Broadcastadresse: 172.16.255.255

## Einführung in die IPv4-Adressierung

- Subnetzmasken bestimmen den Netzanteil einer IP-Adresse
- Sie werden über die IP-Adresse gelegt
- Zu jeder IP-Konfiguration gehören:
  - IP-Adresse
  - Subnetzmaske
  - Default-Gateway

Netzklasse	Standard-Subnetzmaske
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0



## Einführung in die IPv4-Adressierung

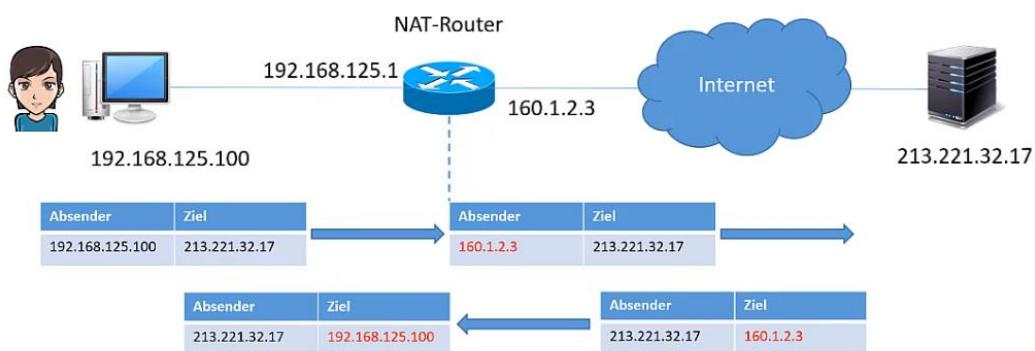
Private IP-Adressbereiche nach RFC 1918

Netzklasse	Privater Adressbereich	Standard-Subnetzmaske
A	10.0.0.0 – 10.255.255.255	255.0.0.0
B	172.16.0.0 – 172.31.255.255	255.255.0.0
C	192.168.0.0 – 192.168.255.255	255.255.255.0

- Private IP-Adressen können in jedem lokalen Netzwerk nach Belieben eingesetzt werden
- Private IP-Adressen werden im Internet nicht geroutet

## Einführung in die IPv4-Adressierung

**Network Address Translation (NAT)** ermöglicht die Kommunikation interner Systeme mit privaten IP-Adressen mit Systemen aus dem Internet



## Einführung in die IPv4-Adressierung

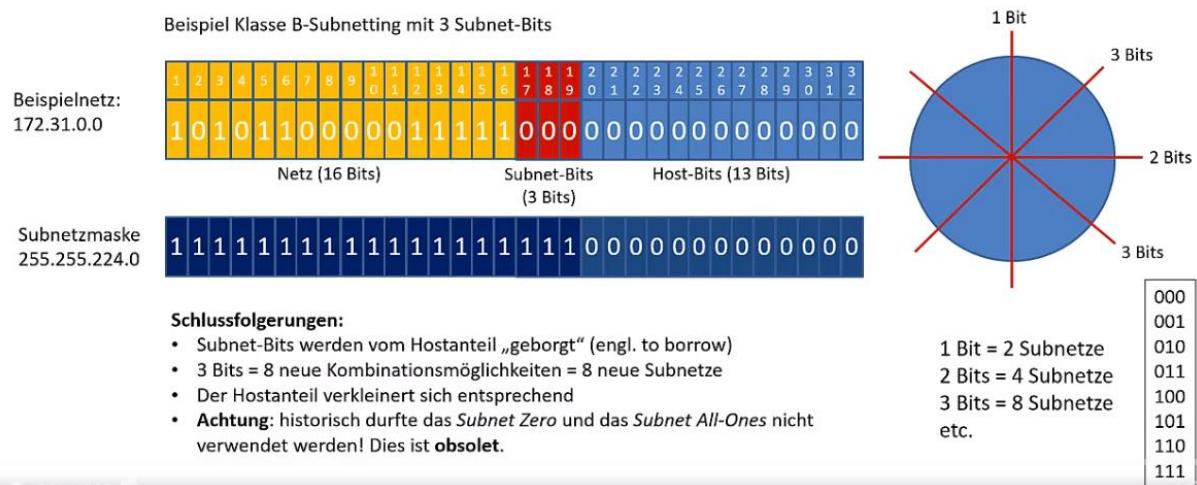
### Spezielle Adressen:

- **Loopback (localhost)**: 127.0.0.0/255.0.0.0 -> meist 127.0.0.1
- **APIPA/ZeroConf**: 169.254.0.0/255.255.0.0

### Klassisches Subnetting

## Das klassische Subnetting

- Subnetting ermöglicht die Aufteilung eines vorhandenen Netzwerks in mehrere gleichgroße Subnetze
- Die Subnetzmaske wird angepasst und bestimmt die Größe jedes Subnetzes



## Das klassische Subnetting

Aufgabe: das Klasse-C-Netz **192.168.125.0** in 4 Subnetze unterteilen

Lösungsfragen:

1. Wie viele Subnetz-Bits werden benötigt?
2. Wie lautet die Subnetzmaske?
3. Wie lauten die Subnetze, 1. und letzter Host, Broadcast-Adresse?
4. Wie viele Hostadressen sind pro Subnetz verfügbar?

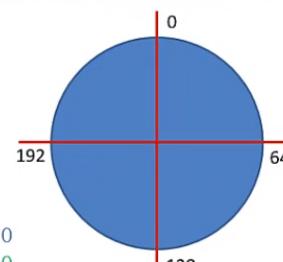
Antworten:

zu 1: es werden **2 Bits** benötigt (4 Kombinationsmöglichkeiten)

Zu 2: die Subnetzmaske lautet **255.255.255.192** (+2 Bits aus dem 4. Oktett)

192.168.125.0 binär:	1100 0000.1010 1000.0111 1101   00 00 0000
Subnetzmaske:	1111 1111.1111 1111.1111 1111   11 00 0000
1. Subnetz:	1100 0000.1010 1000.0111 1111   00 00 0000 = 192.168.125.0
2. Subnetz:	1100 0000.1010 1000.0111 1111   01 00 0000 = 192.168.125.64
3. Subnetz:	1100 0000.1010 1000.0111 1111   10 00 0000 = 192.168.125.128
4. Subnetz:	1100 0000.1010 1000.0111 1111   11 00 0000 = 192.168.125.192

Bit	8	7	6	5	4	3	2	1
Wert	128	64	32	16	8	4	2	1



## Das klassische Subnetting

Subnetz	1. Host	Letzter Host	Broadcast
192.168.125.0	.1	.62	.63
192.168.125.64	.65	.126	.127
192.168.125.128	.129	.190	.191
192.168.125.192	.193	.254	.255

Pro Subnetz sind  $64 - 2 = 62$  Hostadressen verfügbar

Aufgabe: in welchem Subnetz befinden sich die folgenden Hosts:

192.168.125.55 / 255.255.255.192 → 1. Subnetz: 192.168.125.0

192.168.125.82 / 255.255.255.192 → 2. Subnetz: 192.168.125.64

192.168.125.212 / 255.255.255.192 → 4. Subnetz: 192.168.125.192

## Das klassische Subnetting

Aufgabe: das Klasse-C-Netz 192.168.125.0 in 6 Subnetze unterteilen

Bit	8	7	6	5	4	3	2	1
Wert	128	64	32	16	8	4	2	1

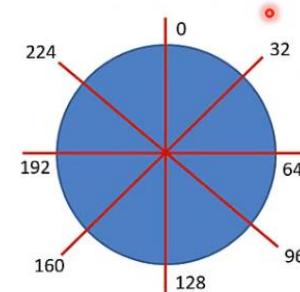
Wie viele Bits werden im Subnetz-Anteil benötigt? Antwort: 3 Bits (8 Kombinationen)

Wie lautet die Subnetzmase? Antwort:  $24 + 3 = 27$  Bits => 255.255.255.224

192.168.125.0 binär:	1100 0000.1010 1000.0111 1101.000 0 0000
Subnetzmase:	1111 1111.1111 1111.1111 1111.111 0 0000
-----	-----
1. Subnetz:	1100 0000.1010 1000.0111 1111.000 0 0000 = 192.168.125.0
2. Subnetz:	1100 0000.1010 1000.0111 1111.001 0 0000 = 192.168.125.32
3. Subnetz:	1100 0000.1010 1000.0111 1111.010 0 0000 = 192.168.125.64
4. Subnetz:	1100 0000.1010 1000.0111 1111.011 0 0000 = 192.168.125.96
5. Subnetz:	1100 0000.1010 1000.0111 1111.100 0 0000 = 192.168.125.128
6. Subnetz:	1100 0000.1010 1000.0111 1111.101 0 0000 = 192.168.125.160
7. Subnetz:	1100 0000.1010 1000.0111 1111.110 0 0000 = 192.168.125.192
8. Subnetz:	1100 0000.1010 1000.0111 1111.111 0 0000 = 192.168.125.224

## Das klassische Subnetting

Subnetz	1. Host	Letzter Host	Broadcast
192.168.125.0	.1	.30	.31
192.168.125.32	.33	.62	.63
192.168.125.64	.65	.94	.95
192.168.125.96	.97	.126	.127
192.168.125.128	.129	.158	.159
192.168.125.160	.161	.190	.191
192.168.125.192	.193	.222	.223
192.168.125.224	.225	.254	.255



Pro Subnetz sind  $32 - 2 = 30$  Hostadressen verfügbar

Aufgabe: Charakterisiere die folgenden Adressen:

192.168.125.191 / 255.255.255.224 → 6. Subnetz: Broadcast-Adresse

192.168.125.96 / 255.255.255.224 → 4. Subnetz: Netzadresse

192.168.125.224 / 255.255.255.192 → 4. Subnetz: normale Adresse! (andere Subnetzmase)

## Das klassische Subnetting

Aufgabe: das Klasse-B-Netz **172.16.0.0** in 10 Subnetze unterteilen

Bit	8	7	6	5	4	3	2	1
Wert	128	64	32	16	8	4	2	1

Wie viele Bits werden im Subnetz-Anteil benötigt? Antwort: 4 Bits (16 Kombinationen)

Wie lautet die Subnetzmase? Antwort:  $16 + 4 = 20$  Bits => 255.255.240.0

Was ist das **Interesting Octet**? Antwort: das 3. Oktett!

Wie groß ist die Schrittweite zwischen den Subnetzen (-> **Magic Number**)?

**Variante A** – ablesen: 4 Bits von links in der Byte-Tabelle -> Schrittweite = 16

**Variante B** – rechnen:  $256 - 240 = 16$

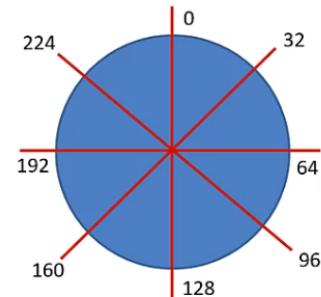
Besonderheit im Klasse-B-Netz: Teile des 3. und das gesamte vierte Oktett stellen den Hostanteil dar

Subnetz	1. Host	Letzter Host	Broadcast
172.16.0.0	0.1	.15.254	.15.255
172.16.16.0	.16.1	.31.254	.31.255
172.16.32.0	.32.1	.47.254	.47.255
...	...	...	...

Host-Adressen verhalten sich im Prinzip wie ein Kilometerzähler

## VLSM und CIDR

- Ausgehend von den Klassen-Netzen (Classful Networks) werden gleichgroße Kuchenstücke erstellt
- Fehlende Flexibilität bei Anforderungen an unterschiedlich großen Subnetzen
- Die Klassen bestimmen die grundsätzliche Netzgröße
- Klassenbasiertes Subnetting löst nicht die grundsätzlichen Probleme beim Routing im Internet



## VLSM und CIDR

### Historisch:

Netze wurden nicht zusammenhängend an die verschiedenen Institutionen und Organisationen vergeben

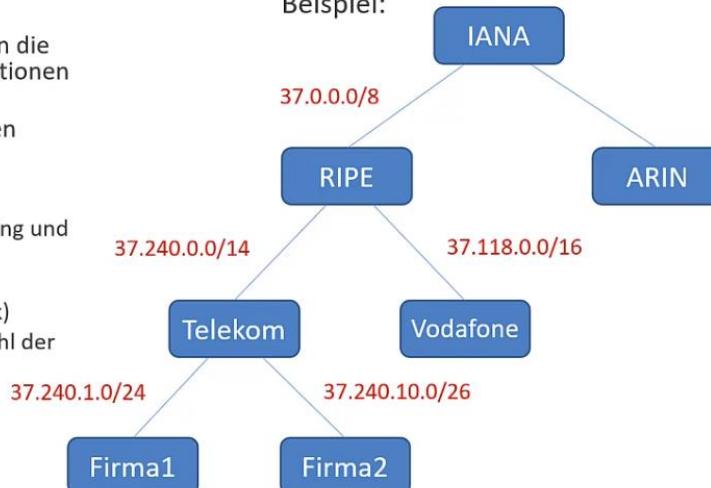
- ⇒ keine einheitliche Vergabe der Adressen
- ⇒ große Routing-Tabellen im Internet

### CIDR (Classless Inter-Domain Routing):

RFCs 1518/1519 führten hierarchisches Routing und VLSM ein

- ⇒ Ziel: Reduzierung der Routing-Tabellen
- ⇒ Weg: VLSM (Variable Length Subnet Mask)
- ⇒ Einführung der Präfix-Schreibweise (Anzahl der gesetzten Bits in der Subnetzmaske)

### Beispiel:



## VLSM und CIDR

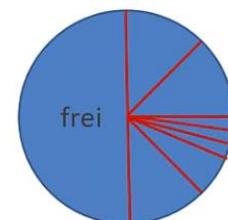
VLSM ermöglicht die beliebige Aufteilung des Ausgangsnetzwerks

Die Subnetze können beliebige Subnetzmasken aufweisen

Ziel: möglichst wenig "Verschnitt" an IP-Adressen

Effizientere Nutzung des IP-Adressenbereichs

Routingtechnisch: Optimierung durch Routen-Zusammenfassung (Supernetting)



## VLSM und CIDR

VLSM dient insbesondere der optimalen Aufteilung der Netze

primäre Fragestellung: Wie viel **Hostadressen** werden benötigt

Ansatz: Wie viele Bits muss ich für den Hostanteil reservieren?

Hostbits	13	12	11	10	9	8	7	6	5	4	3	2	1
Kombinationen	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2
Hostadressen	8190	4094	2046	1022	510	254	126	62	30	14	6	2	0



Beispiel: in einem Subnetz werden 120 Hosts benötigt

-> 7 Bits müssen für den Hostanteil reserviert werden

->  $32 - 7 = 25$  Bits bleiben für den Netzanteil

-> Präfix: /25 bzw. Subnetzmaske: 255.255.255.128

Bit	8	7	6	5	4	3	2	1
Wert	128	64	32	16	8	4	2	1

## VLSM und CIDR

Hostbits	13	12	11	10	9	8	7	6	5	4	3	2	1
Kombinationen	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2
Hostadressen	8190	4094	2046	1022	510	254	126	62	30	14	6	2	0

Eine Filiale bekommt das Subnetz 192.168.1.0/24 zugewiesen

Der Administrator möchte die Abteilungen mit eigenen Subnetzen unterteilen:

Produktion: 120 Hosts -> 7 Bits im Hostanteil -> Präfix:  $32 - 7 = /25 \rightarrow 255.255.255.128$

Vertrieb: 50 Hosts -> 6 Bits im Hostanteil -> Präfix:  $32 - 6 = /26 \rightarrow 255.255.255.192$

Einkauf: 25 Hosts -> 5 Bits im Hostanteil -> Präfix:  $32 - 5 = /27 \rightarrow 255.255.255.224$

Verwaltung: 20 Hosts -> 5 Bits im Hostanteil -> Präfix:  $32 - 5 = /27 \rightarrow 255.255.255.224$

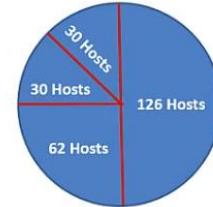
Ansatz: die Subnetze sortiert nach Größe aufteilen, um nahtlos anknüpfen zu können:

192.168.1.0/25 -> Schrittweite 128

192.168.1.128/26 -> Schrittweite 64

192.168.1.192/27 -> Schrittweite 32

192.168.1.224/27

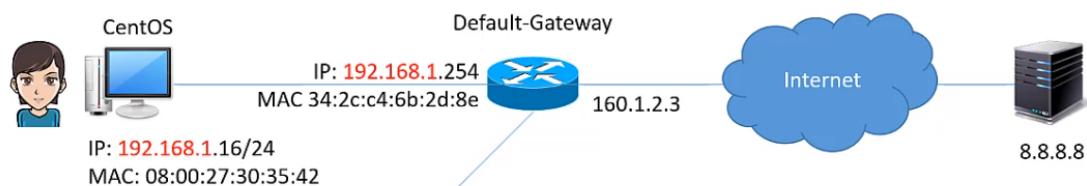


Bit	8	7	6	5	4	3	2	1
Wert	128	64	32	16	8	4	2	1

## ARP und die MAC-Adressen

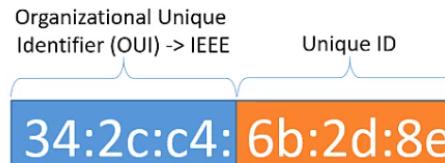
- ARP löst IP-Adressen in MAC-Adressen auf
- Innerhalb eines Subnetzes wird auf Layer 2 mittels MAC-Adressen kommuniziert
- Wird ein IP-Paket an ein Ziel außerhalb des eigenen Subnetzes versendet, so wird es an die MAC-Adresse des Default-Gateways bzw. zuständigen Routers gesendet, der es dann via Routinglogik weiterleitet

## Das Address Resolution Protocol



### Bezeichnungen:

- MAC-Adresse
- Hardware Adresse
- Physische Adresse
- Burned-In-Adresse (BIA)



### Darstellungsform:

- Linux: 08:00:27:30:35:42
- Windows: 08-00-27-30-35-42 oder 080027303542
- Cisco: 0800.2730.3542

## TCP und UDP

- TCP und UDP sind die generischen Transportprotokolle auf Layer 4 (Transport Layer)

- Die meisten Anwendungen nutzen das Transmission Control Protocol (TCP) zur Übertragung der Anwendungsprotokolle (wie HTTP, FTP; SSH, etc.)
- TCP erstellt Sitzungen und überwacht die Datenkommunikation
- Bei jeder TCP-Sitzung wird ein 3-Way-Handshake durchgeführt:
  - 1. Paket: SYN-Flag ist gesetzt (Client zum Server)
  - 2. Paket: SYN/ACK-Flags sind gesetzt (Server zum Client)
  - 3. Paket: ACK-Flag ist gesetzt (Client zum Server)
- Mittels SEQ- und ACK-Numbers prüft TCP, ob der Kommunikationspartner jeweils alle Daten erhalten hat (Fehlerkorrektur/Errorcorrection)
- Window-Size: legt die Größe des Empfangspuffers fest (Flusskontrolle/Flowcontrol)
- TCP und UDP nutzen Ports. Jede Netzwerkanwendung wird an einen Port zwischen 1 und 65535 gebunden (ermöglicht Multiplexing)
- Die Serverports liegen unter 1024 (Well-Known-Ports)
- Client-Ports werden vom Betriebssystem dynamisch festgelegt
- Das User Datagram Protocol (UDP) ist eine leichtgewichtige Alternative zu TCP
- UDP erstellt keine Sitzungen, Übertragungen geschehen nach "Best Effort"

## Die Well-Known-Ports

---

21/tcp – File Transfer Protocol (FTP)

22/tcp – Secure Shell (SSH)

23/tcp – Telnet

25/tcp – Simple Mail Transfer Protocol (SMTP)

53/udp+tcp – Domain Name System (DNS)

67+68/udp – Dynamic Host Configuration Protocol (DHCP)

69/udp – Trivial File Transfer Protocol (TFTP)

80/tcp+udp – HTTP

110/tcp – Post Office Protocol (POP3)

123/udp – Network Time Protocol (NTP)

143/tcp – Internet Message Access Protocol (IMAP4)

161/udp – Simple Network Management Protocol (SNMP)

162/udp – SNMP Trap

389/tcp – Lightweight Directory Access Protocol (LDAP)

443/tcp – Secure Socket Layer (SSL) / Transport Layer Security (TLS) -> HTTPS

514/udp+tcp – Syslog

636/tcp – Secure LDAP (LDAPS)

- In der Datei /etc/services sind fast alle Standard-Ports enthalten

## Wichtige TCP/IP-Anwendungen

# Wichtige TCP/IP-Anwendungen

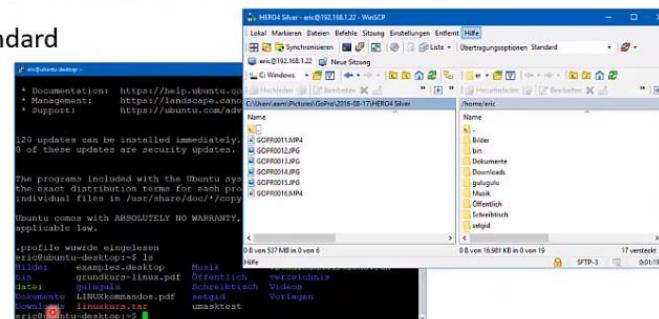
### File Transfer Protocol (FTP) – Ports 21/tcp und 20/tcp (bei aktivem FTP):

- Eines der ältesten Protokolle
- Wird zur Übertragung von Dateien verwendet
- Unterstützt Active und Passive FTP
- Überträgt im Klartext bzw. unverschlüsselt
- Unterstützt Anonymous-FTP, Zugriff ohne Authentisierung möglich
- Auch heute noch an vielen Stellen im Einsatz
- Unter Linux existieren zahlreiche Varianten, wichtige sind:
  - [ProFTPD](#): umfangreicher, einfach zu konfigurierender Server
  - [vsFTPD](#): auf Sicherheit ausgelegter FTP-Server
- Sichere Varianten via SSH-Suite (SCP und SFTP)

## Wichtige TCP/IP-Anwendungen

### Secure Shell (SSH) – Port 22/tcp:

- Standard-Anwendung für Systemadministration von Linux-Systemen und Netzwerk-Geräten
- Unterstützt zahlreiche Algorithmen, die während der Initialisierungsphase festgelegt werden
- Die SSH-Suite in der Version 2 bringt [SCP](#) und [SFTP](#) mit zur sicheren Dateiübertragung
- Unter Windows unter anderem mit [PuTTY](#) und [WinSCP](#) einsetzbar
- Unter Linux ist [OpenSSH-Server](#) der Standard
- SSH ersetzt [rlogin](#) und [rsh](#) sowie [Telnet](#) (Klartextkommunikation)
- SSH unterstützt [X11 Forwarding](#)



## Wichtige TCP/IP-Anwendungen

### Simple Mail Transfer Protocol (SMTP) – Port 25/tcp:

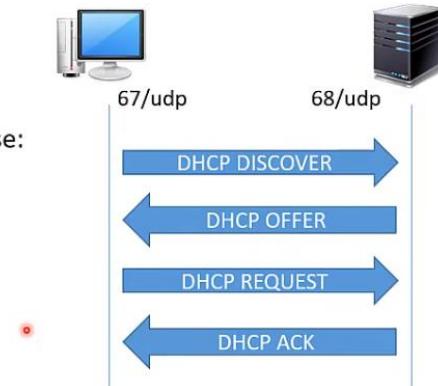
- Eines der ältesten TCP/IP-Protokolle (1982, noch älter als FTP)
- Standard-Protokoll für den Mail-Versand
- Überträgt Daten in Klartext
- Erweiterung durch Extended SMTP (ESMTP)
  - Ermöglicht diverse Zusatzfunktionen (optional vom Server bereitgestellt)
  - Unterstützt auch diverse Authentifizierungsmethoden
- Heutzutage meistens via TLS abgesichert



## Wichtige TCP/IP-Anwendungen

### Dynamic Host Configuration Protocol (DHCP) – Ports 67+68/udp:

- Automatische IP-Konfiguration von DHCP-Clients
- Wird in den meisten Netzwerkumgebungen eingesetzt
- Fast jeder HomeOffice-Router unterstützt DHCP
- Mindestkonfigurationsparameter im Rahmen einer Lease:
  - IP-Adresse
  - Subnetzmaske
  - Default-Gateway
- Viele zusätzliche DHCP-Optionen möglich:
  - DNS-Server
  - DNS-Suffix
  - TFTP-Server (für PXE-Boot)
- Die DHCP-Lease muss nach Ablauf zurückgegeben werden



## Wichtige TCP/IP-Anwendungen

### Domain Name System (DNS) – Port 53/udp + tcp:

- Hierarchisches System zur Namensauflösung im Internet
- Ersetzt bzw. ergänzt Datei /etc/hosts
- Namensraum wird in Zonen (Domains) unterteilt
- Verwaltung der Zonen kann delegiert werden
- Toplevel-Domain (TLD): verwaltet oberste Ebene
- Domain: Wird an Institutionen oder Einzelpersonen vergeben
- Subdomain: Optionale Untereinheit
- Hostname: Werden in der Zone verwaltet, die für eine Domain zuständig ist
- Die Zone enthält verschiedene Eintragstypen (RRs), z.B. A, AAAA, MX, NS, TXT, PTR
- Jede Zone muss im Internet redundant durch mindestens zwei DNS-Server geführt werden

[www.gulugulu.org.](http://www.gulugulu.org)

## Wichtige TCP/IP-Anwendungen

### Hypertext Transfer Protocol (HTTP) – Port 80/tcp:

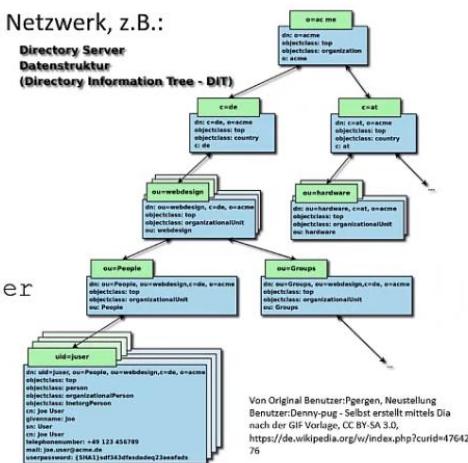
- Kommunikationsprotokoll für Webinhalte (WWW)
- HTTP ist statuslos (viele einzelne Verbindungen)
- Mittlerweile ist HTTP/2 veröffentlicht
- Wichtige HTTP-Methoden:
  - GET (Für normale Anfragen)
  - POST (Für formularbasierten Datenupload)
  - HEAD (Abgleich von Browsecache-Daten)
- Wichtige HTTP-Statuscodes:
  - 2xx – Erfolgreiche Operation (z.B. 200 OK)
  - 3xx – Umleitung des Browsers (z.B. 301 Moved Permanently zum Umleitung auf HTTPS)
  - 4xx – Client-Fehler (z.B. 404 Not Found)
  - 5xx – Server-Fehler (z.B. 502 Bad Gateway für Proxy-Weiterleitungsfehler)
  - Übertragung heute meist via HTTPS (SSL/TLS) – Port 443/tcp

Source	Destination	Protocol	Length	Info
192.168.1.1	192.168.1.254	TCP	66 3449 → 80 [SYN]	Seq=0 Win=64240 Len=0 M
192.168.1.1	192.168.1.254	TCP	66 3441 → 80 [SYN]	Seq=0 Win=64240 Len=0 M
192.168.1.254	192.168.1.1	TCP	66 89 → 3449 [SYN, ACK]	Seq=0 Ack=1 Win=14
192.168.1.254	192.168.1.1	TCP	66 89 → 3441 [SYN, ACK]	Seq=0 Ack=1 Win=14
192.168.1.1	192.168.1.254	TCP	54 3440 → 80 [ACK]	Seq=1 Ack=1 Win=65536 L
192.168.1.1	192.168.1.254	TCP	54 3441 → 80 [ACK]	Seq=1 Ack=1 Win=525568
192.168.1.1	192.168.1.254	HTTP	512 GET / HTTP/1.1	
192.168.1.254	192.168.1.1	TCP	60 89 → 3441 [ACK]	Seq=1 Ack=459 Win=15688
192.168.1.254	192.168.1.1	TCP	1514 89 → 3441 [ACK]	Seq=1 Ack=459 Win=15688
192.168.1.254	192.168.1.1	TCP	1514 89 → 3441 [ACK]	Seq=1461 Ack=459 Win=15
192.168.1.254	192.168.1.1	TCP	1514 89 → 3441 [ACK]	Seq=2923 Ack=459 Win=15
192.168.1.254	192.168.1.1	TCP	1514 89 → 3441 [ACK]	Seq=4381 Ack=459 Win=15
192.168.1.254	192.168.1.1	TCP	1514 89 → 3441 [ACK]	Seq=5841 Ack=459 Win=15
192.168.1.1	192.168.1.254	TCP	54 3441 → 80 [ACK]	Seq=459 Ack=7301 Win=52
192.168.1.254	192.168.1.1	HTTP	470 HTTP/1.1 200 OK (text/html)	
192.168.1.1	192.168.1.254	TCP	54 3441 → 80 [ACK]	Seq=459 Ack=717 Win=52
192.168.1.1	192.168.1.254	HTTP	404 GET /index.html	
192.168.1.254	192.168.1.1	TCP	68 89 → 3441 [ACK]	Seq=7717 Ack=891 Win=16
192.168.1.1	192.168.1.254	HTTP	476 GET /js/avcore.js?lang=en HTTP/1.1	
192.168.1.254	192.168.1.1	TCP	68 89 → 3446 [ACK]	Seq=1 Ack=223 Win=15680
192.168.1.254	192.168.1.1	HTTP	480 HTTP/1.1 304 Not Modified	
192.168.1.1	192.168.1.254	TCP	66 3442 → 80 [SYN]	Seq=0 Win=64240 Len=0 M
192.168.1.254	192.168.1.1	TCP	66 89 → 3442 [SYN, ACK]	Seq=0 Ack=1 Win=14

## Wichtige TCP/IP-Anwendungen

### Lightweight Directory Access Protocol (LDAP) – 389/tcp + 636/tcp (LDAPS)

- Netzwerkprotokoll zur Abfrage von LDAP-basierten Verzeichnisdiensten
- Verzeichnisdienste speichern Informationen zu Ressourcen im Netzwerk, z.B.:
  - Benutzer- und Gruppeninformationen
  - Computer
  - Drucker
  - Standortinformationen
  - Usw.
- Jedes Objekt hat einen eindeutigen Identifier, z.B.  
O=acme, c=de, ou=webdesign, ou=People, uid=juser
- Microsoft Windows Active Directory basiert auf LDAP



## Einführung in IPv6

- Bereits Mitte der 1990er Jahre war absehbar, dass die IPv4-Adressen ca. 2011 ausgehen
- IPv6 wurde als Nextgen Internet Protocol entwickelt (RFC 2460 1998)
- IPv6 bietet einen deutlich erweiterten Adressraum:
  - IPv4: 32 Bit: ca. 4,3 Milliarden Adressen
  - IPv6: 128 Bit: ca. 340 Sextillionen Adressen:  
**340.282.366.920.938.463.463.374.607.431.768.211.456**
  - Das reicht, um jedem Sandkorn auf der Erde mehrere Adressen zuzuweisen
  - Der Adressraum von IPv6 reicht aus, um jeden Quadratmillimeter der Erdoberfläche inklusive der Ozeane mit rund 600 Billiarden Adressen zu versorgen

## Einführung in IPv6

---

- Die 128 Bits der Adresse werden in acht hexadezimale, 16-Bit lange Blöcke unterteilt, die durch Doppelpunkte getrennt werden, z.B.:

`2001:0DB8:0000:0000:0000:0208:FEC5:5E7A`

- Führende Nullen können ausgelassen werden:

`2001:DB8:0:0:0:208:FEC5:5E7A`

- Aufeinanderfolgende Null-Blöcke können EINMAL pro Adresse durch zwei Doppelpunkte ersetzt werden:

`2001:DB8::208:FEC5:5E7A`

- IPv4-Adressen können in IPv6-Adressen eingebettet (bzw. zum Schluss angehängt) werden, z.B. `X:X:X:X:192.168.0.1` oder `0:0:0:0:0:192.168.0.1` bzw. `::192.168.0.1`, oder aber: `::COA8:1`

## Einführung in IPv6

---

- Präfixe geben den Netzanteil der Adresse wieder (wie bei IPv4)
- Die ersten 64 Bit einer Adresse können als Netzanteil frei aufgeteilt werden, während die letzten 64 Bit immer die Interface-ID darstellen

`2001:0DB8:1234:ABCD: AFFE:56AB:DEAD:BEEF`

Zuweisung	Präfix Binär	Präfix Hex
Global Unicast-Adressen	001	2000::/3 (2000-3FFF)
Link-Local Unicast-Adressen	1111 1110 10	FE80::/10
Unique Local IPv6-Adressen	1111 110(1)	FC00::/7 (FD00::/8)
Multicast-Adressen	1111 1111	FF00::/8

## Einführung in IPv6

---

- Multicast spielt bei IPv6 eine viel größere Rolle als bei IPv4, unter anderem für:
  - Neighbor Discovery
  - Router Discovery
  - U.a.
- ICMPv6 wird für diverse essentielle Funktionen genutzt (u.a. ND und RD)
- Neighbor Discovery ersetzt ARP
- IPv6 unterstützt Autoconfiguration (Netz-Präfix wird vom IPv6-Router geliefert)
- DHCPv6 kann ergänzend zur Autoconfiguration (stateless) oder anstatt Autoconfiguration (stateful) eingesetzt werden
- DHCPv6 ist ein eigener Dienst und nicht mit IPv4 kompatibel (Client: 546/udp, Server: 547/udp)
- DNS wurde nur um AAAA-Einträge ergänzt und kann normal weiterverwendet werden
- Die meisten Anwendungen funktionieren weitgehend problemlos mit IPv6