# Netzwerke

# Contents

# 1 Kapitel 1

## 1.1 DNS - Domain Name System

Ein Domain Name System nimmt Internetadressen wie 'facebook.com' und liefert dessen IP-Adresse, damit Rechenr sich damit verbinden können. Wobei 'com'[1] eine TLD (Top Level Domain) ist. Man unterscheidet TLDs wie folgt:

**gTLD** (auch: genericTLDs, allgemeine TLD) Diese werden wieder in 2 Untergruppen aufgeteilt:

- **sTLD** (auch: sponsored TLD) Diese TLD werden nur an Websiten vergeben, welche bestimmte Forderungen erfüllen. '.gov'
- **uTLD** (auch: unsponsered TLD) TLD werden ohne Vorgaben vergeben. '.com, .xyz'

**ccTLD** (auch: country-codeTLD) TLD die zeigen aus welchem Land die Website kommt. '.de oder .us'

Beim Beispiel von facebook.com nennt man das .facebook eine Second-Level Domain, würde da noch www. stehen wäre das die Third-Level Doamin, Es können (quasi) beliebig viele Subdomains eingeführt werden. Die niedrigste Subdomain heißt hierbei Lowest-Level Domain.
Eine FQDN (Fully Qualieified Domain Name) setzt sich aus TopLevelDomain, LowestLevelDomain und mindestens einer Domain dazwischen zusammem.

## 1.2 LAN - Local Area Network

Das LAN vernetzt Geräte auf einen bestimmten (o.a. begrenzten Bereich) Normalerweise ein Haus im privaten Gebrauch oder ein Firmen-Campus etc. Dabei sind die Geräte ständig miteinander verbunden.

# 2 Kapitel 2

## 2.1 Ethernet

Jede Netzwerkkarte hat eine eigene MAC-Adresse, die benutzt wird damit Rechner sich gegenseitig Nachrichten schicken können. Die Nachrichten bei Etherent werden auch Frames, Package und Header genannt

[1] Eigentlich ist der Punkt rechts von der Domain. Bei der TTL wird der Punkt meistens weggelassen. Richtig heißt es beispielsweise 'com.'

# 3 Kapitel 3

## 3.1 ARP - Address Resolution Protocol

ARP wird verwendet, wenn ein Computer oder Router ein Paket an ein Gerät im eigenen Netz senden will, aber nur die Ziel-IP-Adresse kennt. Er schickt dann ein Ethernet II Frame mit dem Typ-Feld 0x608 an die Broadcast-Adresse FF-FF-FF-FF-FF-FF. Wenn der gesuchte Rechner das Paket empfängt, antwortet er mit seiner MAC-Adresse.

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| Hardware Type | | Protocol Type | |
| HLEN | PLEN | Operation | |
| Sender Hardware Address Byte 0-3 | | | |
| Sender Hardware Address Byte 4-5 | | Sender Internet Address Byte 0-1 | |
| Sender Internet Address Byte 2-3 | | Target Hardware Address Byte 0-1 | |
| Target Hardware Address Byte 2-5 | | | |
| Target Internet Adress Byte 0-3 | | | |

Die einzelnen Felder bedeuten dabei:

**Hardware Type** beschreibt, über welches Mittel kommuniziert wird. Ethernet bedeutet dabei 1.

**Protocol Type** Mit welchem Protokoll soll später kommuniziert werden? Das gleiche wie bei Ethernet II, also 0x800 für IP.

**HLEN** beschreibt, wie lange eine Hardware-Adresse ist. Ist bei Ethernet immer 6.

**PLEN** beschreibt, wie lange eine Protokoll-Adresse ist. Ist bei IPv4 immer 4.

**Operation** Was wird gerade ausgeführt? 1 für Request, 2 für Response

**Sender Hardware Address** ist die MAC-Adresse des Senders

**Sender Internet Address** ist die IP-Adresse des Senders

**Target Hardware Address** ist die MAC-Adresse des Empfängers

**Target Internet Address** ist die IP-Adresse des Enpfängers.

Die gesuchten Felder werden mit Nullen gefüllt. Mit “reverse ARP” kann ein Computer, der über das Netzwerk gebootet wurde, die IP-Adresse zu seiner eigenen MAC-Adresse erfragen. Dazu ist allerdings ein Server nötig.

## 3.2 ICMP - Internet Control Message Protocol

ICMP-Pakete sind in IP-Pakete eingepackt (Protocol-Feld im IP-Header wird auf 1 gesetzt). Es wird unter anderem verwendet, um die Erreichbarkeit von Systemen im Internet zu testen (ping), Netzwerkfehler zu erkennen und um bei Zeitüberschreitungen benachrichtigt zu werden.

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| Type | Code | Checksum | |

Die gültigen Werte für das Type-Feld sind:

**0:** Echo Reply (bei ping)

**3:** Destination unreachable (der Sender wird benachrichtigt, wenn das Ziel nicht erreichbar ist). Die Gründe dafür können im Code-Feld stehen

- **0:** net unreachable
- **1:** host unreachable
- **2:** protocol unreachable
- **3:** port unreachable
- **4:** Fragmentation needed and DF set
- **5:** source route failed (der Sender hat eine Route im IP-Header angegeben, die nicht funktioniert hat)

**4:** Source Quench (der Empfänger bittet den Sender, weniger Pakete zu senden)

**5:** Redirect (Wird von Routern verwendet, um die Netzwerkroute zu beeinflussen)

**8:** Echo (bei ping)

**11:** Time exceeded (TTL wurde unterschritten)

**12:** Parameter Problem, ungültiger IP Header

**13:** Timestamp (für Zeit-Synchronisierung)

**14:** Timestamp Reply (für Zeit-Synchronisierung)

Durch das Code-Feld können zusätzliche Informationen mitgegeben werden, z.B. warum die Verbindung gescheitert ist.
tracert verwendet das IP TTL Feld und die Time exceeded ICMP Nachricht, um die Route zu einem anderen Computer nachzuverfolgen. Das TTL wird dabei hochgezählt, bis das Paket den Zielrechner verwendet.

## 3.3 TCP

TCP stellt eine virtuelle Verbindung zwischen den Kommunikationspartnern her. Durch die Verbindung können Daten zuverlässig übermittelt werden. Bei TCP und UDP gibt es Ports, an die die Daten gesendet werden. *Well known Ports* gehen von 0 bis 1023 sind u. a.:

| Dienst | Portnummer |
|---|---|
| http | 80 |
| https | 443 |
| SMTP (E-Mail versenden) | 25 |
| POP3 (E-Mail abholen) | 110 |
| FTP | 20, 21 |
| ssh | 22 |

Damit der Server antworten kann, sendet der Client auch eine Portnummer als *source port* mit, an die die Antwort geschickt werden soll. Weil sich diese Ports immer ändern können, heißen sie auch dynamische Ports. Sie gehen von Port 49152 bis 65535. Zwischen den Well known ports und den dynamic ports liegen die registered Ports von 1024 bis 49151.

Eine TCP-Verbindung entsteht über einen **3way Handshake**.

1. **(SYN)** Der Client sendet ein Paket, bei dem das SYN-Feld gesetzt ist und mit welcher Sequence Number er beginnen möchte.
2. **(ACK+SYN)** Der Server antwortet dem Client mit einem Paket, bei dem die SYN + ACK Felder gesetzt sind. Außerdem teilt er dem Client die eigene Sequence Number mit.
3. **(ACK)** Der Client bestätigt die Antwort und beide bauen eine sichere Verbindung zur Kommunikation auf.

Nun kann kommuniziert werden. Der Client sendet die Daten und zählt dabei die gesendeten Bytes mit der Sequence Number hoch. Die Sequence Number ist immer ein Pointer auf das erste Byte im aktuellen Paket. Er kann nur so viele Daten wie die Window size senden, ohne auf ein acknowledgen des Servers zu warten. Der Server acknowledged ein einzelnes oder mehrere Pakete mit einem ACK-Paket. Acknowledgement Number ist dabei die als nächstes erwartete Sequence-Number des Gegenübers. Die TCP-Verbindung wird über einen **4way Handshake** geschlossen.

1. **(FIN)** Der Client will die Verbindung trennen und sendet ein FIN-Paket.
2. **(ACK)** Der Server bestätigt den Erhalt des FIN-Pakets.
3. **(FIN)** Sobald das Programm auf dem Server bereit ist, die Verbindung zu schließen, wird ein FIN-Paket an den Client gesendet.
4. **(ACK)** Der Client bestätigt den Erhalt des Pakets. Die Verbindung wird beendet.

<table>
<tr><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">4</td></tr>
<tr><td colspan="4">Source Port</td><td colspan="4">Destination Port</td></tr>
<tr><td colspan="8">Sequence Number</td></tr>
<tr><td colspan="8">Acknowledgement Number</td></tr>
<tr><td colspan="2">data<br>offset</td><td colspan="2">reserved</td><td colspan="2">flags</td><td colspan="2">window</td></tr>
<tr><td colspan="4">checksum</td><td colspan="4">urgent pointer</td></tr>
<tr><td colspan="8">options<br>0 oder mehr 32bit Wörter</td></tr>
<tr><td colspan="8">Daten...</td></tr>
</table>

Die Felder bedeuten dabei folgendes:

**Source Port** Port, von dem aus das Paket geschickt wird.

**Destination Port** Port, an den das Paket gesendet wird.

**Sequence Number** Alle übertragenen Bytes sind durchnummeriert, beginnend bei einem zufälligen Wert.

**Acknowledgement Number** Nur gültig, wenn die ACK-Flag gesetzt ist. Quittiert alle Pakete vom Gegenüber mit einer Acknowledgement Number = Sequence Number+1.

**Data Offset** gibt die Zahl der 32bit-Wörter im TCP-Header an. Min 5, wenn keine Options verwendet werden.

**reserved** Bits werden nicht verwendet.

**Flags** Es gibt folgende Flags:

- **0: CWR** Für uns irrelevant
- **1: ECE** Für uns irrelevant
- **2: URG** Das Urgent Pointer Feld enthält gültige Daten
- **3: ACK** Die Acknowledgement Nummer ist gültig
- **4: PSH** Die Daten des TCP Nutzlasten-Feldes sofort an die nächsthöhere Schicht liefern. (z.B. Telnet)
- **5: RST** Reset: Die Verbindung soll vom Rechner, der RST sendet, zurückgesetzt werden.
- **6: SYN** Wird beim Verbindungsaufbau verwendet.
- **7: FIN** Wird beim Schließen der Verbindung verwendet.

**window** Teilt dem Kommunikationspartner mit, wie viele Datenbytes er noch senden darf, bevor er auf eine Quittierung warten muss.

**checksum** Prüfsumme auf Fehler im Header

**urgent pointer** Pointer auf das letzte Byte, was zu den Vorrangsdaten (urgent data) gehört.

**options** Optionale Daten

## 3.4 UDP

UDP bietet einen Datagramm-Dienst für die darüberliegende Schicht an. UDP ist unzuverlässig und verbindungslos. Wird oft verwendet, wenn die Ziel-Adresse Broadcast oder Multicast ist. Der UDP-Header sieht wie folgt aus:

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| Source Port | | Destination Port | |
| Length | | Checksum | |
| Daten... | | | |

Das **Length**-Feld bezieht sich dabei auf die Länge von UDP-Header + Daten.
UDP wird u. a. verwendet für:

| Dienst | Portnummer |
|---|---|
| Trivial File Transfer Protocol (TFTP) | 69 |
| Domain Name Services (DNS) | 53 |
| Simple Network Management Protocol (SNMP) | 161/162 |
| Routing Information Protocol (RIP) | 520 |