Netzwerke, Teil 2: IPv4

Handout zum Vortrag im CCC Freiburg

Balrog

2026-11-11

IPv4-Adressen

Eine IPv4-Adresse ist eine 32-Bit-Zahl. Geschrieben wird sie als vier Byte, jedes als Dezimalzahl von 0 bis 255, durch Punkte getrennt.

```
172.16.0.1  =  10101100.00010000.00000000.00000001  =  2886729729
```

Umrechnen in Binär: Stellenwerte aufschreiben und von links abziehen.

128	64	32	16	8	4	2	1	
1	0	1	0	1	1	0	0	= 128 + 32 + 8 + 4 = 172
0	0	0	1	0	0	0	0	= 16

Jedes Byte hat immer acht Stellen, auch mit führenden Nullen.

Logisches UND

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Eine 0 auf einer Seite macht das Ergebnis immer 0. Eine 1 lässt die andere Seite durch. Mit einer Maske verknüpft wirkt das UND deshalb wie eine Schablone.

CIDR und Netzmasken

Classless Inter-Domain Routing (CIDR, RFC 4632): Die Zahl hinter dem Schrägstrich sagt, wie viele Bits von links das Netz bezeichnen. Der Rest ist der Host-Teil.

```
172.16.1.1/24
Adresse   10101100.00010000.00000001.00000001
Maske   & 11111111.11111111.11111111.00000000   (= 255.255.255.0)
        = 10101100.00010000.00000001.00000000   = 172.16.1.0  (Netzadresse)
```

Zwei Rechner liegen im selben Netz, wenn bei beiden dieselbe Netzadresse herauskommt. Dann reden sie direkt miteinander (per ARP). Sonst muss ein Router dazwischen.

Präfix	Maske	Adressen	nutzbar für Geräte
/8	255.0.0.0	16.777.216	16.777.214
/16	255.255.0.0	65.536	65.534
/22	255.255.252.0	1.024	1.022
/24	255.255.255.0	256	254
/25	255.255.255.128	128	126
/26	255.255.255.192	64	62
/27	255.255.255.224	32	30
/28	255.255.255.240	16	14
/30	255.255.255.252	4	2
/31	255.255.255.254	2	2 (nur Router-Verbindungen, RFC 3021)
/32	255.255.255.255	1	eine einzelne Adresse

In jedem Netz sind zwei Adressen vergeben: Die mit lauter Nullen im Host-Teil ist die **Netzadresse**, die mit lauter Einsen der **Broadcast**. Welche das sind, hängt allein von der Maske ab: 172.16.0.128 ist im /24 eine normale Adresse, im /25 eine Netzadresse.

Vor CIDR gab es feste Klassen (A = /8, B = /16, C = /24). Das Wort „Klasse-C-Netz” für ein /24 hört man noch.

Besondere Adressen

Die vollständige Liste führt die IANA als *IPv4 Special-Purpose Address Registry* (RFC 6890).

Bereich	Zweck
`10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`	privat (RFC 1918), im Internet nicht geroutet
`100.64.0.0/10`	Carrier-Grade NAT beim Provider (RFC 6598)
`127.0.0.0/8`	Loopback, immer der eigene Rechner
`169.254.0.0/16`	Link-Local: selbst vergeben, wenn DHCP nicht antwortet
`192.0.2.0/24`, `198.51.100.0/24`, `203.0.113.0/24`	für Dokumentation und Beispiele (RFC 5737)
`198.18.0.0/15`	Benchmark-Tests (RFC 2544)
`224.0.0.0/4`	Multicast
`240.0.0.0/4`	reserviert
`255.255.255.255`	Broadcast im eigenen Netz, wird nie geroutet

Eine Adresse aus `100.64.0.0/10` am eigenen Router oder Handy-Hotspot heißt: Der Provider übersetzt ein zweites Mal, von außen ist man nicht direkt erreichbar.

Der IPv4-Header

20 Byte, mit Optionen bis 60. Die Felder für den Anfang:

Feld	Bit	Inhalt
Version	4	immer 4
Time to Live (TTL)	8	Startwert beim Absender (macOS, Linux 64, Windows 128). Jeder Router zieht 1 ab, bei 0 wird das Paket verworfen
Protocol	8	was in der Nutzlast steckt: 1 ICMP, 6 TCP, 17 UDP
Source Address	32	Absender
Destination Address	32	Ziel

Der Rest: IHL (Headerlänge), DSCP/ECN (Priorität, Stau), Total Length, Identification, Flags (darunter *Don't Fragment*), Fragment Offset und die Header Checksum, die jeder Router neu rechnet, weil sich die TTL ändert.

ARP

Das *Address Resolution Protocol* (ARP, RFC 826) findet zur IP-Adresse die MAC-Adresse.

1. Der Rechner fragt per Broadcast: „Who has 172.16.1.1? Tell 172.16.1.10".
2. Nur der Gemeinte antwortet per Unicast: „172.16.1.1 is at 02:42:00:00:00:01".
3. Das Ergebnis bleibt einige Minuten im ARP-Cache.

So sieht das Paar in Wireshark aus (Filter `arp`, Datei `Sources/mitschnitte/arp.pcap`):

```
1  02:42:00:00:00:10 → Broadcast          ARP  Who has 172.16.1.1? Tell 172.16.1.10
2  02:42:00:00:00:01 → 02:42:00:00:00:10  ARP  172.16.1.1 is at 02:42:00:00:00:01
```

Feld in der Anfrage	Wert
Ethernet Dst / Src	`ff:ff:ff:ff:ff:ff` / `02:42:00:00:00:10`
EtherType	`0x0806`, ARP sitzt direkt auf Ethernet, ohne IP-Header
Opcode	1 = request, in der Antwort 2 = reply
Sender MAC / IP	`02:42:00:00:00:10` / `172.16.1.10`
Target MAC / IP	`00:00:00:00:00:00` (gesucht) / `172.16.1.1`

Der Cache auf den drei Systemen, jeweils mit dem Eintrag für den Router:

```
$ ip neigh                        (Linux)
172.16.1.1 dev eth0 lladdr 02:42:00:00:00:01 REACHABLE
$ arp -an                         (macOS)
? (172.16.1.1) at 2:42:0:0:0:1 on en0 ifscope [ethernet]
> arp -a                          (Windows)
  172.16.1.1            02-42-00-00-00-01     dynamisch
```

Liegt das Ziel in einem anderen Netz, fragt der Rechner nach der MAC seines Routers (Default Gateway). Die IP-Zieladresse im Paket bleibt die des eigentlichen Ziels.

ARP kennt keine Authentifizierung. Wer sich als Router ausgibt, bekommt den Verkehr (*ARP-Spoofing*). Gegenmittel am Switch: *Dynamic ARP Inspection*.

ICMP

Das *Internet Control Message Protocol* (ICMP, RFC 792) ist der Meldedienst von IP. Es steckt direkt im IP-Paket (Protocol 1) und hat keine Ports.

Type	Code	Name	Wer schickt es
8	0	Echo Request	ping
0	0	Echo Reply	das Ziel
3	0	Net Unreachable	Router ohne passende Route
3	1	Host Unreachable	letzter Router, ARP blieb ohne Antwort
3	3	Port Unreachable	Ziel, niemand lauscht auf dem UDP-Port
3	4	Fragmentation Needed	Router, Paket zu groß und Don’t Fragment gesetzt
3	13	Administratively Prohibited	Firewall
11	0	Time Exceeded	Router, bei dem die TTL auf 0 fiel

Echo Request und Echo Reply tragen nach Type, Code und Checksum einen Identifier und eine Sequence Number. Daran ordnet ping die Antworten zu. Es folgen beliebige Daten (macOS und Linux: 56 Byte), die unverändert zurückkommen. Bei Fragmentation Needed steht an dieser Stelle die *Next-Hop MTU*.

Fehlermeldungen enthalten den IP-Header und die ersten 8 Byte des auslösenden Pakets. Daran erkennt der Absender, welches Paket gemeint war.

ICMP nicht pauschal filtern. Jede Leitung hat eine Höchstgröße für Pakete, die *Maximum Transmission Unit* (MTU): Ethernet 1500 Byte, DSL mit PPPoE 1492. Ein Absender schickt 1500 Byte mit gesetztem „Don’t Fragment”. Der Router vor der DSL-Leitung kann das Paket nicht weiterreichen und meldet „Fragmentation Needed” samt der passenden MTU. Der Absender verkleinert daraufhin seine Pakete. Das heißt *Path MTU Discovery* (PMTUD).

Filtert eine Firewall diese Meldung weg, verschwinden die großen Pakete still (*PMTUD Black Hole*). Die Verbindung kommt zustande, hängt aber, sobald große Pakete fließen: Webseite lädt halb, SSH hängt nach dem Login. Destination Unreachable und Time Exceeded deshalb immer durchlassen.

Routing

Jedes Gerät mit IP hat eine **Routing Table**: Zielnetz, nächster Router (*Next Hop*), Interface. Beispiel Router 1 aus dem Vortrag:

Destination	Next Hop	Interface
172.16.1.0/24	connected	eth0
10.0.12.0/30	connected	eth1
10.0.13.0/30	connected	eth2
172.16.2.0/24	10.0.12.2	eth1
0.0.0.0/0	10.0.13.1	eth2

Für jedes Paket: Zieladresse UND Maske jeder Zeile. Kommt die Netzadresse der Zeile heraus, passt die Zeile. Von allen passenden gewinnt die mit der längsten Maske (*Longest Prefix Match*). `0.0.0.0/0` passt immer und heißt **Default Route**.

Für ein Paket an 172.16.2.5 passen `172.16.2.0/24` und `0.0.0.0/0`. Die /24 gewinnt, das Paket geht an 10.0.12.2.

Ein normaler Rechner hat nur zwei wichtige Zeilen: sein eigenes Netz (direkt per ARP) und die Default Route zum Router.

Bei jedem Router wird der Ethernet-Header neu geschrieben. Der IP-Header bleibt gleich, nur die TTL sinkt um 1.

Zwischen Routern liegen meist kleine **Transfernetze** (/30 oder /31). Routen werden von Hand eingetragen (statisch) oder von Routing-Protokollen gelernt (dynamisch: OSPF im eigenen Netz, BGP zwischen den Netzen im Internet).

traceroute

traceroute schickt Pakete mit TTL 1, 2, 3 und so weiter. Jedes stirbt einen Router weiter, und der Router meldet sich mit ICMP Time Exceeded. So erscheint jeder Router als eigene Zeile. Am Ziel antwortet Linux/macOS-traceroute ein Port Unreachable (es schickt UDP), Windows-`tracert` ein Echo Reply. `* * *` heißt: Dieser Router schickt keine Meldungen.

Befehle

Aufgabe	Linux	macOS	Windows
Adresse und Maske	`ip -4 addr`	`ifconfig`	`ipconfig`
Routing Table	`ip route`	`netstat -rn -f inet`	`route print -4`
Welche Route für ein Ziel?	`ip route get 1.1.1.1`	`route -n get 1.1.1.1`	`Find-NetRoute -RemoteIPAddress 1.1.1.1`
ARP-Cache	`ip -4 neigh`	`arp -an`	`arp -a`
ARP-Eintrag löschen	`sudo ip neigh flush dev eth0`	`sudo arp -d 172.16.1.1`	`arp -d 172.16.1.1` (als Admin)
ping	`ping -c 3 ccc.de`	`ping -c 3 ccc.de`	`ping ccc.de`
Weg zum Ziel	`traceroute ccc.de`, `mtr ccc.de`	`traceroute ccc.de`	`tracert ccc.de`
ping, 1500 Byte, nicht zerlegen	`ping -M do -s 1472 ccc.de`	`ping -D -s 1472 ccc.de`	`ping -f -l 1472 ccc.de`

1472 Byte Daten + 8 Byte ICMP + 20 Byte IP = 1500 Byte, genau die Ethernet-MTU.

Wireshark-Anzeigefilter: `ip.addr == 172.16.1.10`, `arp`, `icmp`, `icmp.type == 11` (Time Exceeded), `ip.ttl < 10`.

Übungen

1. **Eigenes Netz.** Finde Adresse und Maske deines Laptops. Rechne die Netzadresse aus. Wie viele Geräte passen in dein Netz?
2. **Rechnen.** Für `10.23.42.77/27` und `172.16.1.200/26`: Netzadresse, Broadcast, Zahl der nutzbaren Adressen?
3. **Gleiches Netz?** Liegen `192.168.4.111/22` und `192.168.7.3/22` im selben Netz? Und mit /24?
4. **Routing Table.** Zeige deine Routing Table an. Welche Zeile ist die Default Route, wer ist dein Router? Was sagt `ip route get` bzw. `route -n get` für `1.1.1.1` und für deinen Router?
5. **ARP.** Leere den ARP-Cache (Befehle oben), pinge deinen Router und schau den Cache wieder an. Schneide dabei mit Filter `arp` mit.
6. **traceroute.** `traceroute ccc.de`: Wie viele Hops? Ab welchem Hop bist du nicht mehr bei deinem Provider? Vergleiche mit der TTL in der Antwort von `ping ccc.de`.

7. **MTU.** Pinge mit 1472 Byte und Don’t Fragment (Befehle oben). Geht es? Taste dich nach oben und unten heran. Hinter einem VPN oder DSL ist die Grenze meist kleiner.

Lösungen

2. `10.23.42.77/27`: Netz `10.23.42.64`, Broadcast `10.23.42.95`, 30 Adressen. `172.16.1.200/26`: Netz `172.16.1.192`, Broadcast `172.16.1.255`, 62 Adressen.
3. Mit /22 ja: Beide ergeben `192.168.4.0`. Mit /24 nein: `192.168.4.0` gegen `192.168.7.0`.

Begriffe

Begriff	Erklärung
Broadcast	Adresse für alle Geräte im Netz, Host-Teil lauter Einsen
CGNAT	Carrier-Grade NAT, zweite Adressübersetzung beim Provider
CIDR	Schreibweise Adresse/Präfixlänge, erlaubt Netze jeder Größe
Default Gateway	Router, an den ein Rechner alles schickt, was nicht im eigenen Netz liegt
Default Route	`0.0.0.0/0`, passt auf jede Adresse, gewinnt nur, wenn nichts Genaueres passt
Hop	ein Router auf dem Weg
Longest Prefix Match	von mehreren passenden Routen gewinnt die mit der längsten Maske
MTU	größtes Paket, das über eine Leitung passt, bei Ethernet 1500 Byte
NAT	Übersetzung privater in öffentliche Adressen am Router
Netzadresse	Adresse mit Host-Teil lauter Nullen, bezeichnet das Netz
Netzmaske	32 Bit, links Einsen für den Netzteil, rechts Nullen für den Host-Teil
Next Hop	nächster Router auf dem Weg zu einem Zielnetz
PMTUD	Path MTU Discovery, Absender lernt per ICMP die kleinste MTU auf dem Weg
PMTUD Black Hole	ICMP-Meldung weggefiltert, große Pakete verschwinden ohne Rückmeldung
Routing Table	Tabelle Zielnetz → Next Hop → Interface
Transfernetz	kleines Netz zwischen zwei Routern
TTL	Time to Live, Zähler im IP-Header, sinkt pro Router um 1

Grafiken im Vortrag: eigene Darstellungen, Quelltexte im Ordner `Sources/bilder/`.

Mitschnitte zum Öffnen in Wireshark, gebaut mit den Adressen aus den Grafiken, im Ordner `Sources/mitschnitte/`:

- `arp.pcap`: ARP-Anfrage und -Antwort wie oben.
- `ipv4-icmp.pcap`: ARP, ping mit TTL 64 und 58, traceroute-Probe mit TTL 1 (Time Exceeded vom Router), UDP an Port 33435 (Port Unreachable vom Ziel), ping an 203.0.113.9 (Net Unreachable vom Router).

Zum Weiterlesen

- W. Richard Stevens, Kevin R. Fall, *TCP/IP Illustrated, Volume 1: The Protocols*, 2. Auflage 2011. Das Standardwerk, jedes Protokoll an echten Mitschnitten erklärt. Zu diesem Teil: Kapitel 2 (Adressen), 4 (ARP), 5 (IP und Forwarding), 8 (ICMP) und 10 (Fragmentierung).
- RFC 791 (IPv4), RFC 826 (ARP), RFC 792 (ICMP), RFC 4632 (CIDR), RFC 1918 (private Adressen), RFC 6890 (besondere Adressen), RFC 1191 (Path MTU Discovery). Alle unter rfc-editor.org.