Netze

Modul 4: IPv4, Subnetze, ARP, ICMP

Prof. Dr. Hannes Tschofenig

23. Oktober 2025

Semesterplanung — Vorlesungen

Modul	Dozent	Datum	Thema
1	Rademacher	2. Oktober 2025	Einführung, OSI-Referenzmodell und Topologien
2	Rademacher	9. Oktober 2025	Übertragungsmedien und Verkabelung
3	Rademacher	16. Oktober 2025	Ethernet und WLAN
4	Tschofenig	23. Oktober 2025	IPv4, Subnetze, ARP, ICMP
5	Tschofenig	30. Oktober 2025	IPv6 und Autokonfiguration
6	Tschofenig	6. November 2025	Netzwerksegmentierung
7	Tschofenig	13. November 2025	Routing
8	Rademacher	20. November 2025	Transportschicht und UDP
9	Rademacher	27. November 2025	TCP
10	Rademacher	4. Dezember 2025	DNS und HTTP 1
11	Tschofenig	11. Dezember 2025	HTTP 2 und QUIC
12	Tschofenig	18. Dezember 2025	TLS und VPN
/	/	8. Januar 2026	`Bei Bedarf / TBA`
13	Tschofenig	15. Januar 2026	Messaging
14	Rademacher	22. Januar 2026	Moderne Netzstrukturen

Semesterplanung — Übungen und Praktika

ID	KW	Art	Thema
	40	/	/
UE-1	41	Übung	Topologien und OSI
UE-2	42	Übung	Übertragungen bspw. Kabel
P-1	43	Praktikum	Laboreinführung, Netzwerktools und Adressierung
S-1	44	Video	IPv4
P-2a	45	Praktikum	Praktikum IPv4 und Autokonfiguration
P-2b	46	Praktikum	Praktikum IPv6 und Autokonfiguration
P-2c	47	Praktikum	IPv4 und IPv6 Diskussion
P-3	48	Praktikum	Routing
P-4	49	Praktikum	VLANs
P-5	50	Praktikum	Transportprotokolle
S-2	51	Experiment	VPN
S-2	52	Experiment	VPN
	2	/	/
P-6	3	Praktikum	DNS
P-7	4	Praktikum	Webkommunikation

UE - Übung laut Stundenplan in den Seminarräumen

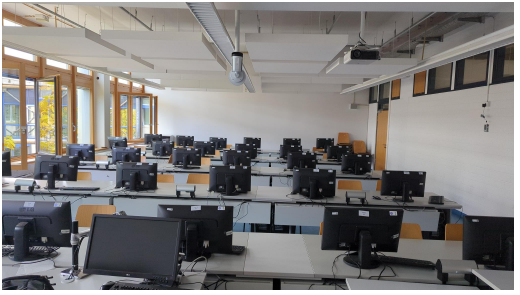
P - Praktikum in C055

S - Selbststudium KEINE Präsenz

Praktikum

Die Auslastung in den Gruppen ist leider sehr unterschiedlich.

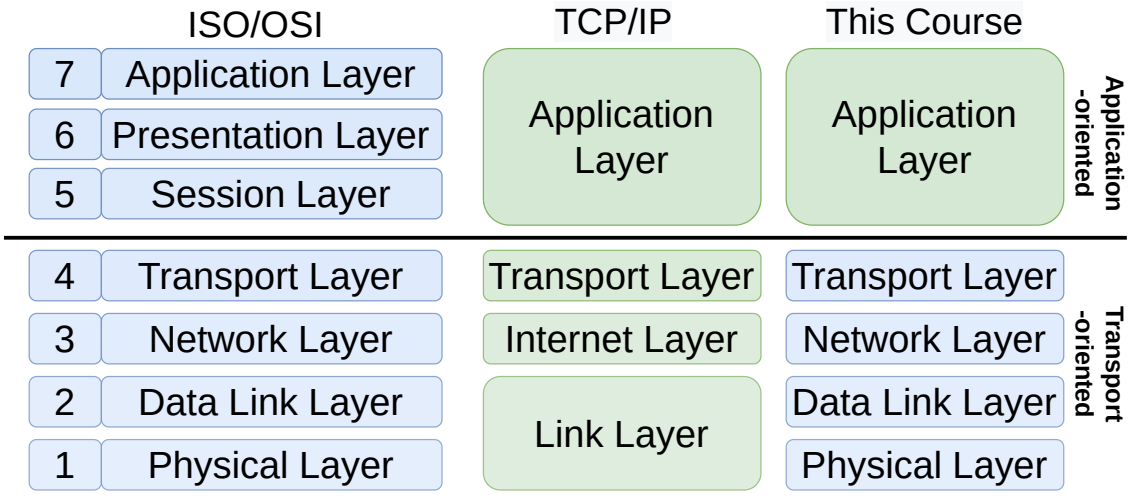
Tag	Zeit
Montag	10:45 – 12:15
Donnerstag	15:15 – 16:45



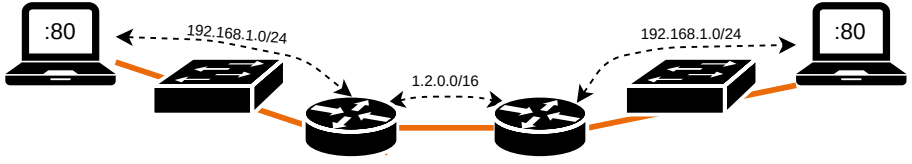
Hinweis

Bitte über LEA in eine der beiden Gruppen ummelden!
Die Donnerstagsgruppe ist nur für Studierende des Bachelorstudiengangs Informatik geeignet!

Revisit - Die sieben Schichten des OSI-Referenzmodells



- Verantwortlich für die **Übertragung und Weiterleitung** von Paketen zwischen Knoten in **unterschiedlichen Netzwerken**.
- **Router** leiten Datenpakete zwischen Netzwerken weiter.
- Das Netzwerk bestimmt den optimalen Übertragungsweg automatisch (**Routing**).
- Der Absender kennt nur die **Adresse des Empfängers**.



IPv4 - Agenda

- IPv4-Adressierung
- IPv4-Paket
- Subnetzbildung
- Address Resolution Protocol (ARP)
- Internet Control Message Protocol (ICMP)

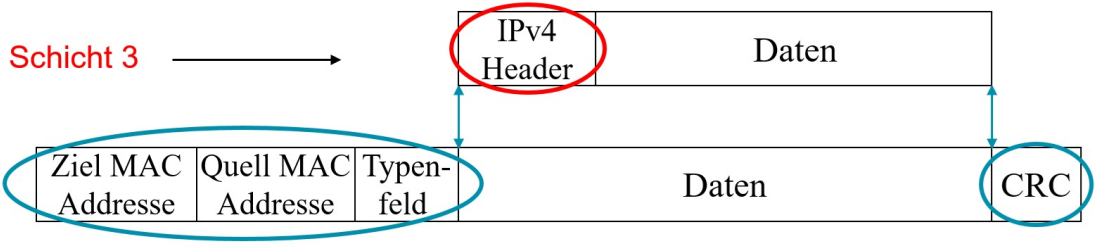
Allgemeines

- Das wichtigste Protokoll der Schicht 3 (Vermittlungsschicht) ist das **Internet Protocol (IP)**.
- Aktuell existieren zwei Versionen: **IPv4** und **IPv6**.

Eigenschaften von IP:

- Verbindungslos (kein Aufbau oder Abbau einer Verbindung)
- Keine Fehlerkorrektur
- Keine Empfangsbestätigung
- Keine Sicherung der Reihenfolge von Datenpaketen
- Keine feste („spezielle“) Route

IP definiert sowohl die **Adressierung** als auch den **Aufbau von IP-Datenpaketen**.

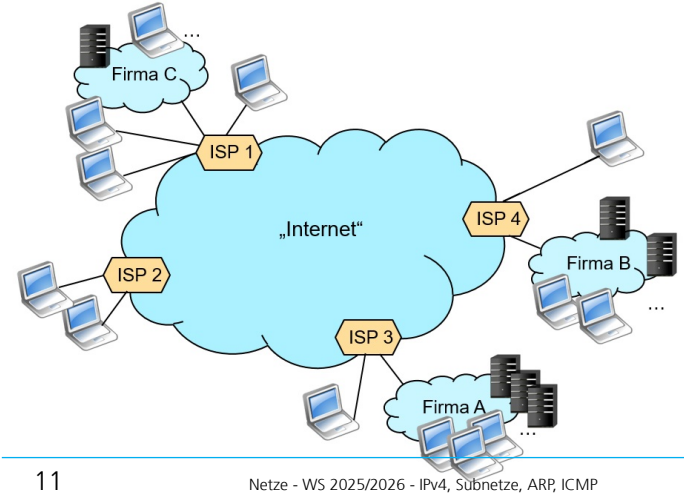


IPv4 -Adressierung

- Adresslänge bei IPv4: 4 Byte bzw. 32 Bit
- Darstellung der 32 Bit IP-Adresse durch 4 durch Punkte getrennte Dezimalzahlen (eine für jedes Byte)
- Beispiel:
  - 01100101.00011110.00000110.00010100
  - = 101.30.6.20

Gemeinsame Adressbereiche für Netze

- Die Vermittlungsschicht verwendet eine Ende-zu-Ende Adressierung
- ISP = Internet Service Provider
- Aber Netze könnten unterschiedlich viele Rechner enthalten
- Frage: Wie könnte IPv4-Adressbereiche an verschiedene Netzgrößen angepasst werden?



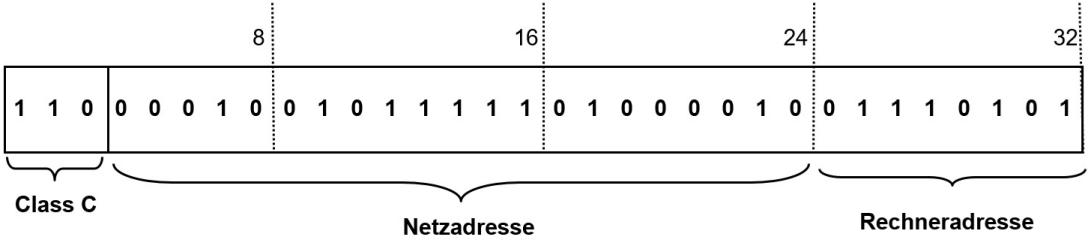
Aufbau der IPv4-Internetadressen

- Anzahl der Adressen bei IPv4
  - Es gibt 2^{32} = 4.294.967.296 IPv4-Adressen; mittlerweile ist das viel zu wenig!
  - Es gibt 2^{128} = 3,4*10^{38} IPv6-Adressen
- Ursprüngliche Aufteilung der IP-Adresse in drei Teile
  - Klasse (A, B, C, D oder E),
  - Netzadresse, und
  - Rechneradresse oder Hostadresse
- Klassen D (Multicast) und E (Experimentelle Verwendung) wird nicht weiter betrachtet.
- Beachte: Oft wird auch die Kombination aus Klasse + Netzadresse + Hostadresse 0 als Netzwerkadresse oder Netzwerk-Id bezeichnet.

Klassen von IPv4-Netzen

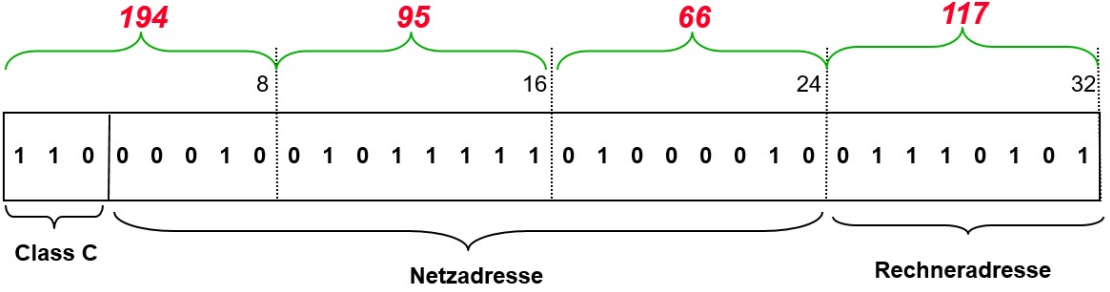
- Sehr große Netze, Adressen der Klasse A
  - 1.Bit: "0| <Netzadresse (7 Bit)> || <Rechneradresse (24 Bit)>. D.h. 2^{24} Rechner
- Mittelgroße Netze, Adressen der Klasse B:
  - 1. und 2. Bit: "10| <Netzadresse (14 Bit)> || <Rechneradresse (16 Bit)>. D.h. 2^{16} Rechner
- Kleinere Netze, Adressen der Klasse C
  - 1. - 3. Bit: "110| <Netzadresse (21 Bit)> || < Rechneradresse (8 Bit)>. D.h. 2^{8} Rechner
- Die Anzahl der benutzbaren Netz- und Rechneradressen ist in der Praxis durch spezielle (reservierte) Bereiche geringer. RFC 6890 [7] spezifiziert diese speziellen Adressen und Netze.
- Diese Einteilung erlaubt im Grunde nur 3 verschiedene Netzgrößen und ist sehr unflexibel.

Beispiel einer IP-Adresse der Klasse C



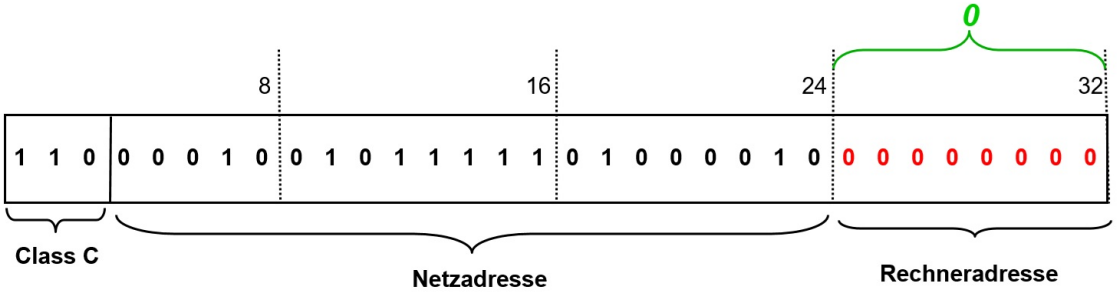
- Schreibweisen:
  - in Bit-Schreibweise: 1100 0010 / 0101 1111 / 0100 0010 / 0111 0101 (unüblich!)
  - in Hex-Schreibweise: C2 5F 42 75 (unüblich!)
  - Übliche Schreibweise: 194.95.66.117
- Schreibweise in Dezimal: 194.95.66.117
- Die Netzmaske gibt den Bitbereich der Netzadresse plus Klassenangabe durch gesetze Bits an. Für das obige Beispiel ergibt sich dann: 255.255.255.0.

Frage: Zu welchem Netz gehört die folgende IP-Adresse?



Frage: Zu welchem Netz gehört die folgende IP-Adresse?

- Setze Rechneradresse auf 0!



- Lösung: Diese genannte IP-Adresse gehört zum Netz 194.95.66.0

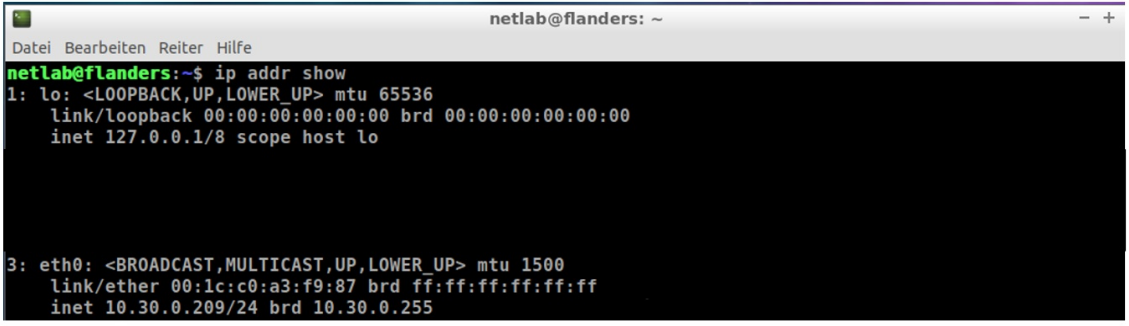
Spezielle IPv4-Adressen

- **Netzwerkadresse**: Hostteil = 0 (z.B. 194.95.66.0 / ‘Das Netz 194.95.66.0‘)
- **Broadcast-Adresse**: Hostteil = „1“ Bits (z.B. 194.95.66.255 / ‘Broadcast an alle Rechner im Netz 194.95.66.0‘)
- Begrenzte Broadcast-Adresse: nur „1“ Bits (z.B. 255.255.255.255 / ‘Broadcast an alle Rechner im lokalen Netz‘)
- **Loopback-Adresse**: Das gesamte Netz 127.0.0.0/8 ist für Loopback reserviert. 127.0.0.1 ist das typische Beispiel.
- Übliche Konvention: Router erhalten als Hostteil die Adresse .1 (z.B. 194.95.66.1 / ‘Router im Netz 194.95.66.0‘)
- Private IPv4-Adressbereiche gemäß RFC 1918 (historisch den Klassen A, B und C zugeordnet) [11]:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

CIDR: Classless Inter-Domain Routing

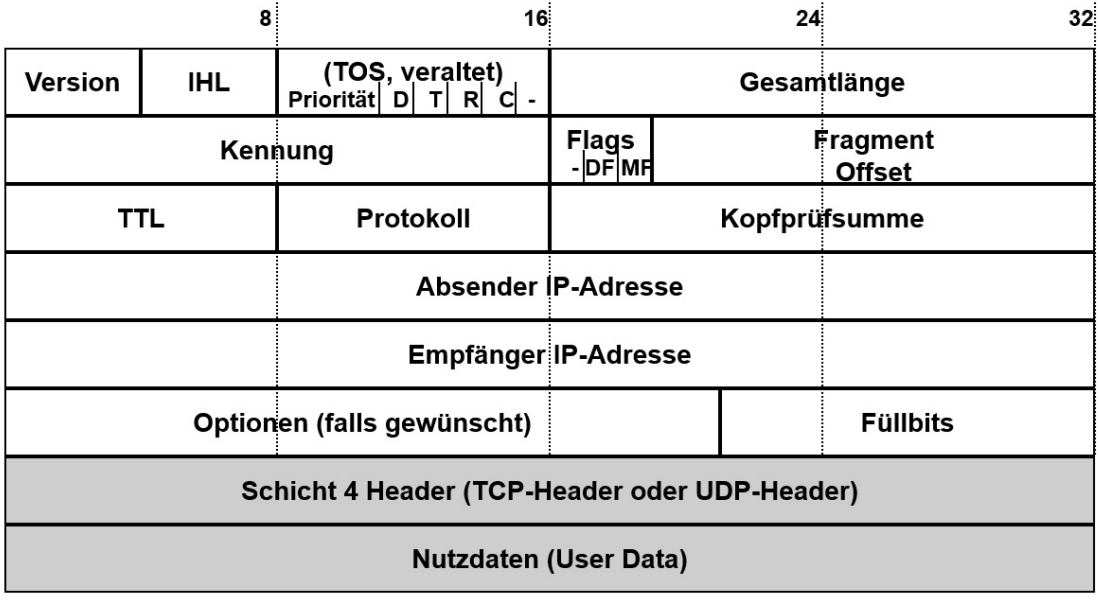
- Durch die zunehmende Knappheit von IPv4-Adressen musste der verbleibende Adressraum effizienter genutzt werden.
- Mit CIDR wurde das starre Klassensystem (A, B, C) aufgehoben, um Netze flexibler und bedarfsgerechter aufteilen zu können.
- CIDR wurde ursprünglich in RFC 1519 [10] definiert und später in RFC 4632 [9] präzisiert.
- Jede IP-Adresse besteht weiterhin aus einem Netz- und einem Hostanteil, jedoch ist die Länge des Netzpräfixes nun variabel.
- Eine CIDR-Adresse besteht aus der IP-Adresse und der Präfixlänge (Anzahl der Netzbits), z. B. `12.10.1.64/26`.
- Früher hätte die Adresse `12.x.x.x` zu einem Klasse-A-Netz gehört, unter CIDR weist jedoch allein die Präfixlänge auf die Netzgröße hin.
- Im Beispiel bedeutet `/26`: 26 Bit für den Netzanteil, 6 Bit für den Hostanteil.

Beispiel für Adressen eines Rechners unter Ubuntu



- Unter Windows
  - Öffnen eines Console-Fenster: Rechtsklick auf Windows Icon am linken, unteren Bildschirmrand
  - „Ausführen“ anklicken
  - Hinter Öffnen: „cmd“ eingeben = > es öffnet sich ein Console-Fenster
  - Im Console Fenster den Befehl „ipconfig /all“ eingeben

Aufbau des IP-Headers [2]



Aufbau des IP-Headers (Fortsetzung)

- **Version**: Zurzeit Version 4; Version 6 ist parallel im Einsatz (4 Bit)
- **IHL (Internet Header Length)**: Länge des IPv4-Headers in 32-Bit-Worten. Normalerweise IHL = 5 $\rightarrow$ 5 × 4 Byte = 20 Byte (keine Optionen) (4 Bit)
- **ToS (Type of Service)**: 8 Bit
  - Heute als **Differentiated Services Code Point (DSCP)** und **Explicit Congestion Notification (ECN)** genutzt.
  - DSCP in RFC 2474 [6], ECN in RFC 3168 [8] definiert.
  - Die ursprüngliche Semantik aus RFC 791 [2] ist veraltet.
  - Verwendung: Dienstgüte-Klassifikation (QoS) und Staukontrolle.
- **Gesamtlänge**: Maximale Paketgröße (Header + Nutzdaten) $2^{16} - 1 = 65{,}535$ Byte (16 Bit)

Aufbau des IP-Headers (Fortsetzung)

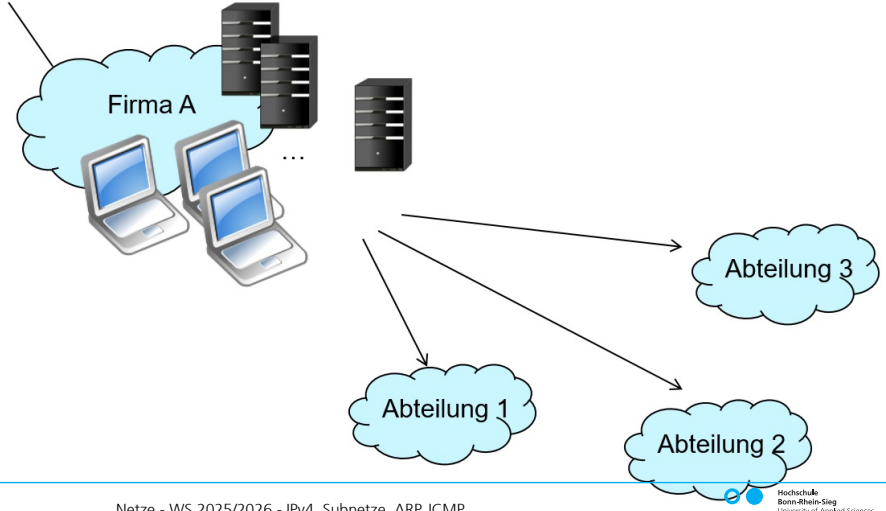
- **Kennung (Identification)**: 16-Bit-Wert zur Identifizierung (Nummerierung) der Fragmente eines Datagramms. Alle Fragmente desselben Pakets tragen die gleiche Kennung.
- **Flags**: 3 Steuerbits zur Kontrolle der Fragmentierung.
  - `DF = 1`: *Do not fragment* – Paket darf nicht fragmentiert werden.
  - `MF = 1`: *More fragments* – weitere Fragmente folgen.
- **Fragment Offset**: Position des Fragments im ursprünglichen Datagramm, angegeben in 8-Byte-Einheiten (13 Bit).
- **TTL (Time To Live)**: Wird bei jedem Router um 1 dekrementiert. Erreicht der Wert 0, wird das Paket verworfen (8 Bit).
  Anmerkung: Ursprünglich als Zeitangabe in Sekunden gedacht, heute als Hop-Zähler interpretiert. Typischer Startwert: z. B. 64.

Aufbau des IP-Headers (Fortsetzung)

- **Protokoll**: 8-Bit-Wert zur Identifizierung des darüberliegenden Transportprotokolls (entspricht der *IANA Protocol Number*).
  - 6 = TCP
  - 17 = UDP
- **Kopfprüfsumme (Header Checksum)**: 16-Bit-Prüfsumme über den IPv4-Header (ohne Nutzdaten). Sie wird bei jedem Hop von Routern überprüft und neu berechnet.
- **Optionen**: Variabel langes Feld (optional), heute kaum noch genutzt. Beispiel: Option 7 (*Record Route*) – Router tragen ihre IP-Adressen in das Optionsfeld ein. (Falls nötig, wird das Feld auf ein Vielfaches von 32 Bit aufgefüllt.)

Subnetzbildung

- Wie können größere Netzwerke sinnvoll strukturiert und verwaltet werden?
- Dazu werden gemeinsame Adressbereiche für Teilnetze, z. B. Abteilungen, eingeführt.
- Ziel: bessere Übersicht, geringerer Broadcast-Verkehr und einfachere Administration.



Aufgabenstellung

- Eine Firma besitzt ein IPv4-Netz und kann – abhängig von der Netzmaske – eine bestimmte Anzahl von Hosts adressieren.
- Broadcasts werden an alle Geräte im selben Netz gesendet (Broadcast-Domäne).
- Eine Aufteilung des Netzes ist nur durch den Einsatz von Routern möglich.
- Jeder Router benötigt pro Interface eine eigene IP-Adresse aus einem separaten Netzbereich.
- Daher ist die Definition von **Subnetzen** innerhalb des vorhandenen Adressraums erforderlich.

Lösung: IP-Subnetzbildung

- **Grundprinzip:** Lokale Erweiterung der Netzadresse
  - Einige Bits des Hostanteils werden „ausgeliehen“, um damit Teilnetze (Subnetze) zu adressieren.
  - Innerhalb eines großen Netzes entstehen mehrere kleinere, logisch getrennte Subnetze.
  - Nachteil: Die Anzahl der pro Subnetz adressierbaren Hosts verringert sich entsprechend.
- Diese Teilnetze heißen **Subnetze** und besitzen jeweils eigene Subnetzadressen.
- Router-Interfaces erhalten IP-Adressen aus unterschiedlichen Subnetzen, wodurch Pakete eindeutig weitergeleitet werden können.
- Organisatorische Einheiten (z. B. Abteilungen) lassen sich auf einzelne Subnetze abbilden – jedes Subnetz mit eigener Adresse.

Lösung: IP-Subnetzbildung (Vorgehensweise)

- Bestimme die Anzahl der benötigten Subnetze und die dafür erforderliche Anzahl an zusätzlichen Bits.
- Überprüfe, ob die verbleibenden Bits für den Hostanteil ausreichen, um die gewünschte Anzahl von Rechnern pro Subnetz zu adressieren.
- Definiere eine **Subnetzmaske**, die alle Bits der Netzadresse einschließlich der lokal erweiterten Subnetzbits umfasst.
- Die Subnetzmaske funktioniert wie die ursprüngliche Netzmaske und dient zur Trennung von Netz- und Hostanteil.
- **Hinweis:** Geräte, die direkt miteinander über Schicht-1- bis Schicht-3-Komponenten verbunden sind, müssen sich im selben Subnetz befinden.

Beispiel: Subnetzbildung

- Ausgangspunkt ist ein IPv4-Netz mit der Adresse `140.64.0.0/16`.

Beispiel: Subnetzbildung

- Ausgangspunkt ist ein IPv4-Netz mit der Adresse `140.64.0.0/16`.
- Die Netzmaske lautet `255.255.0.0`.

Beispiel: Subnetzbildung

- Ausgangspunkt ist ein IPv4-Netz mit der Adresse `140.64.0.0/16`.
- Die Netzmaske lautet `255.255.0.0`.
- Wir möchten weitere 16 Subnetze verwenden.

Beispiel: Subnetzbildung

- Ausgangspunkt ist ein IPv4-Netz mit der Adresse `140.64.0.0/16`.
- Die Netzmaske lautet `255.255.0.0`.
- Wir möchten weitere 16 Subnetze verwenden.
- Zur weiteren Unterteilung werden **4 Bits** des Hostanteils verwendet → es entstehen **16 Subnetze**:

$$2^4 = 16$$

- Die neue Subnetzmaske lautet `255.255.240.0 (/20)` (= 11111111.11111111.11110000.00000000)
- Beispieladresse: `140.64.32.4` (= 10001100.01000000.00100000.00000100)
- Interpretation: Diese Adresse gehört zum **Subnetz** `140.64.32.0/20` und bezeichnet den **Host Nr. 4** in diesem Subnetz.

Weiteres Anwendungsbeispiel (1)

- Die Firma XYZ verfügt über ein **IPv4-Netz** mit der Adresse `196.32.8.0/24`.

Weiteres Anwendungsbeispiel (1)

- Die Firma XYZ verfügt über ein **IPv4-Netz** mit der Adresse `196.32.8.0/24`.
- Netzmaske: `255.255.255.0` (= 11111111.11111111.11111111.00000000)

Weiteres Anwendungsbeispiel (1)

- Die Firma XYZ verfügt über ein **IPv4-Netz** mit der Adresse `196.32.8.0/24`.
- Netzmaske: `255.255.255.0` (= 11111111.11111111.11111111.00000000)
- Broadcast-Adresse: `196.32.8.255` (= 11000100.00100000.00001000.11111111)

Weiteres Anwendungsbeispiel (1)

- Die Firma XYZ verfügt über ein **IPv4-Netz** mit der Adresse `196.32.8.0/24`.
- Netzmaske: `255.255.255.0` (= 11111111.11111111.11111111.00000000)
- Broadcast-Adresse: `196.32.8.255` (= 11000100.00100000.00001000.11111111)
- Gültiger Adressbereich für Hosts: `196.32.8.1` – `196.32.8.254`

Weiteres Anwendungsbeispiel (1)

- Die Firma XYZ verfügt über ein **IPv4-Netz** mit der Adresse `196.32.8.0/24`.
- Netzmaske: `255.255.255.0` (= 11111111.11111111.11111111.00000000)
- Broadcast-Adresse: `196.32.8.255` (= 11000100.00100000.00001000.11111111)
- Gültiger Adressbereich für Hosts: `196.32.8.1` – `196.32.8.254`
- Anzahl möglicher Hosts: $2^8 - 2 = 254$

Weiteres Anwendungsbeispiel (2)

- Die Firma XYZ besteht aus fünf Abteilungen: **Entwicklung, Marketing, Produktion, Vertrieb** und **Verwaltung**.
- Die IP-Adressen der einzelnen Abteilungen sollen eindeutig voneinander unterscheidbar sein.

Weiteres Anwendungsbeispiel (2)

- Die Firma XYZ besteht aus fünf Abteilungen: **Entwicklung, Marketing, Produktion, Vertrieb** und **Verwaltung**.
- Die IP-Adressen der einzelnen Abteilungen sollen eindeutig voneinander unterscheidbar sein.
- **Lösung:** Jede Abteilung erhält ein eigenes Subnetz. Für die Adressierung von 5 Subnetzen werden **3 Bit** benötigt:

$$2^3 = 8 \text{ mögliche Subnetze (3 Bit für die Subnetz-ID).}$$

Weiteres Anwendungsbeispiel (2)

- Die Firma XYZ besteht aus fünf Abteilungen: **Entwicklung, Marketing, Produktion, Vertrieb** und **Verwaltung**.
- Die IP-Adressen der einzelnen Abteilungen sollen eindeutig voneinander unterscheidbar sein.
- **Lösung:** Jede Abteilung erhält ein eigenes Subnetz. Für die Adressierung von 5 Subnetzen werden **3 Bit** benötigt:

$$2^3 = 8 \text{ mögliche Subnetze (3 Bit für die Subnetz-ID).}$$

- Daraus ergibt sich eine Subnetzmaske von `255.255.255.224` (/27) (= 11111111.11111111.11111111.11100000)

Weiteres Anwendungsbeispiel (3)

- Subnetzmaske: `255.255.255.224` (/27) (= 11111111.11111111.11111111.11100000)
- Es werden **3 Bits des Hostanteils** zur Erweiterung der Netzadresse genutzt.
- Gesucht: Die Subnetzadressen für die fünf Abteilungen.
- Ergebnis:
  - `196.32.8.32` (= 11000100.00100000.00001000.00100000)
  - `196.32.8.64` (= 11000100.00100000.00001000.01000000)
  - `196.32.8.96` (= 11000100.00100000.00001000.01100000)
  - `196.32.8.128` (= 11000100.00100000.00001000.10000000)
  - `196.32.8.160` (= 11000100.00100000.00001000.10100000)

Weiteres Anwendungsbeispiel (4)

- IP-Adresse der 1. Abteilung:
  - 11000100.00100000.00001000. 001 00000 = 196.32.8.32
  - ... mit der Broadcast-Adresse:
  - 11000100.00100000.00001000. 001 11111 = 196.32.8.63
- IP-Adresse der 2. Abteilung:
  - 11000100.00100000.00001000. 010 00000 = 196.32.8.64
  - ... mit der Broadcast-Adresse:
  - 11000100.00100000.00001000. 010 11111 = 196.32.8.95
- IP-Adresse der 3. Abteilung:
  - 11000100.00100000.00001000. 011 00000 = 196.32.8.96
  - ... mit der Broadcast-Adresse:
  - 11000100.00100000.00001000. 011 11111 = 196.32.8.127
- IP-Adresse der 4. Abteilung:
  - 11000100.00100000.00001000. 100 00000 = 196.32.8.128
  - ... mit der Broadcast-Adresse:
  - 11000100.00100000.00001000. 100 11111 = 196.32.8.159
- IP-Adresse der 5. Abteilung:
  - 11000100.00100000.00001000. 101 00000 = 196.32.8.160
  - ... mit der Broadcast-Adresse:
  - 11000100.00100000.00001000. 101 11111 = 196.32.8.191

Weiteres Anwendungsbeispiel (5)

- Alle Subnetze haben die gleiche Subnetzmaske.
- Jedes Subnetz hat eine eigene Subnetzadresse und eine eigene Broadcastadresse.
- Pro Abteilung bleiben 5 Bits für die Adressierung der Rechner □2^5 = 32 Zustände
- Aber: Der Zustand 00000 ist bereits für die Netzadresse verwendet und der Zustand 11111 für die Broadcast-Adresse. □Es können 2^5 - 2 = 30 Rechner pro Abteilung adressiert werden.
- Welche beiden Randbedingungen sind grundsätzlich bei einer Subnetzbildung zu beachten?
- Die Anzahl der benötigen Subnetze und die Anzahl der Rechner im größten Subnetz!

Address Resolution Protocol (ARP)

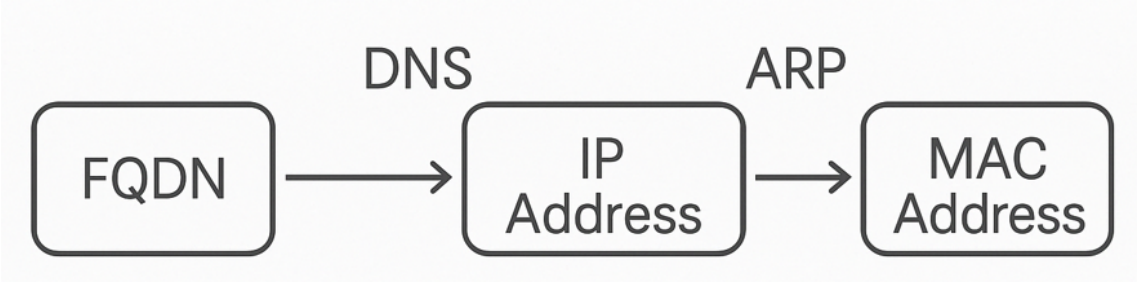
Adressauflösung (1)

- In den verschiedenen Schichten des Netzwerkmodells werden unterschiedliche Arten von Adressen bzw. Namen verwendet:
  - **MAC-Adressen** – 48-Bit-Adressen der Sicherungsschicht (Data Link Layer)
  - **IP-Adressen** – 32-Bit-IPv4-Adressen der Vermittlungsschicht (Network Layer)
  - **Fully Qualified Domain Names (FQDN)** – z. B. `www.h-brs.de`, verwendet auf der Anwendungsschicht

Adressauflösung (2)

- Ein Anwender ruft im Browser eine Webseite auf und gibt eine URL (Uniform Resource Locator) ein.
- Das **Domain Name System (DNS)** wird genutzt, um den **Fully Qualified Domain Name (FQDN)** in eine **IP-Adresse** aufzulösen.
- Diese IP-Adresse dient anschließend als Zieladresse auf **Schicht 3 (IP)**.
- Befindet sich das Ziel nicht im lokalen Netz, wird das Paket an den **Standard-Gateway (Default Router)** gesendet.
- Der **Data Link Layer (Schicht 2)** benötigt dazu die **MAC-Adresse** des nächsten Empfängers.

Adressauflösung (3)



- Die Zuordnung von IP-Adresse zu MAC-Adresse erfolgt über das **Address Resolution Protocol (ARP)** [3].
- Zur Ermittlung einer **IP-Adresse** aus einer bekannten **MAC-Adresse** wurde früher das **Reverse Address Resolution Protocol (RARP)** verwendet [4]. Heute übernehmen diese Funktion Protokolle wie **BOOTP** oder **DHCP**.
- Auch das **Domain Name System (DNS)** unterstützt eine Auflösung in umgekehrter Richtung: Mit **Reverse DNS** kann zu einer IP-Adresse der zugehörige Domainname ermittelt werden [5].

ARP – Address Resolution Protocol

1. Überprüfen, ob die benötigte Zuordnung (IP zu MAC) bereits im lokalen **ARP-Cache** vorhanden ist.
   - Falls ja: Die gespeicherte MAC-Adresse wird verwendet.
   - Falls nein: Weiter mit Schritt 2.
2. Senden einer **ARP-Request**-Nachricht als **Broadcast** (Ziel-MAC-Adresse: `FF:FF:FF:FF:FF:FF`) an alle Geräte im lokalen Netz. Enthalten ist die gesuchte Ziel-IP-Adresse.
3. Alle Geräte im Netzwerk prüfen, ob die angefragte IP-Adresse zu ihrer eigenen gehört.
4. Nur das Gerät mit der passenden IP-Adresse antwortet mit einer **ARP-Reply**-Nachricht und teilt seine MAC-Adresse mit.
5. Der Absender speichert die Zuordnung von IP- und MAC-Adresse im **ARP-Cache**.
6. Das Datenpaket kann nun an die ermittelte MAC-Adresse gesendet werden.

Hinweis: ARP funktioniert nur innerhalb desselben **lokalen Netzes** (Broadcast-Domäne).

Pseudocode: ARP-Algorithmus

```
if target_IP in ARP_Cache then
   use corresponding MAC address
else
   send ARP_Request (broadcast)
   wait for ARP_Reply
   if ARP_Reply received then
      update ARP_Cache
      use received MAC address
   else
      fail (host unreachable)
```

Internet Control Message Protocol (ICMP)

- Definiert in **RFC 792** [1].
- **ICMP** wird der **IP-Schicht (Layer 3)** zugeordnet und dient der Übermittlung von **Fehler- und Statusmeldungen** zwischen Netzwerkteilnehmern.
- Die **ICMP-Nachricht** folgt direkt auf den **IP-Header** und ist durch eine **Prüfsumme** geschützt.
- Jede ICMP-Nachricht ist durch ein **Typfeld** und ggf. ein **Codefeld** eindeutig gekennzeichnet.

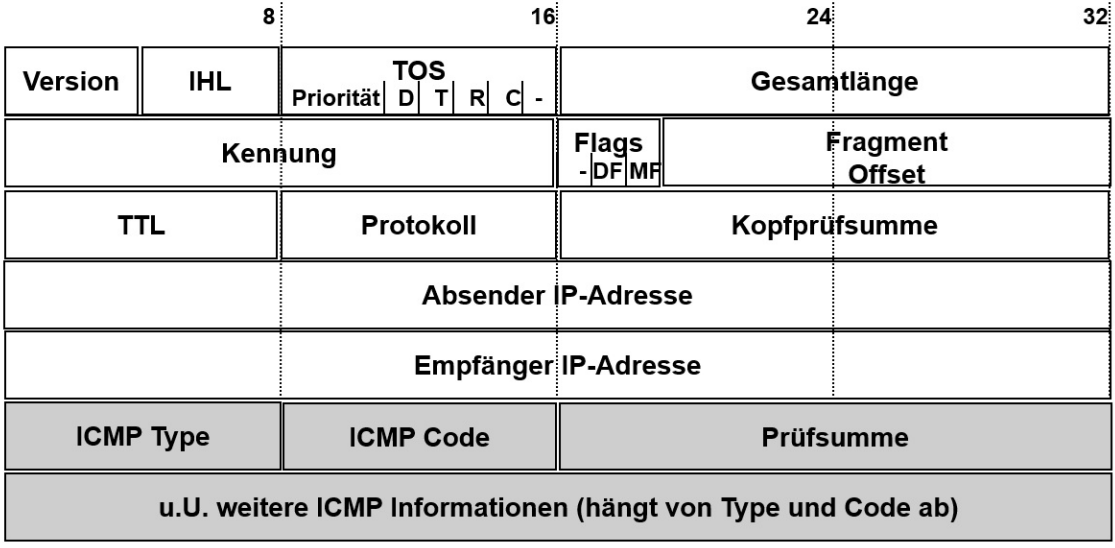
Übersicht über häufig verwendete ICMPv4-Typen

Typ	Name	Beschreibung / Verwendung
0	Echo Reply	Antwort auf einen Echo Request (z. B. `ping`)
3	Destination Unreachable	Ziel nicht erreichbar. Verschiedene Codes geben genauere Ursache an (Netz, Host, Port, etc.)
5	Redirect	Hinweis auf besseren Router – selten verwendet, oft durch Firewalls geblockt
8	Echo Request	Erreichbarkeitsprüfung über `ping`; sendet ICMP-Anfrage
11	Time Exceeded	TTL überschritten – genutzt z. B. von `traceroute` zur Pfadverfolgung
12	Parameter Problem	Fehlerhafte oder ungültige Header-Felder erkannt

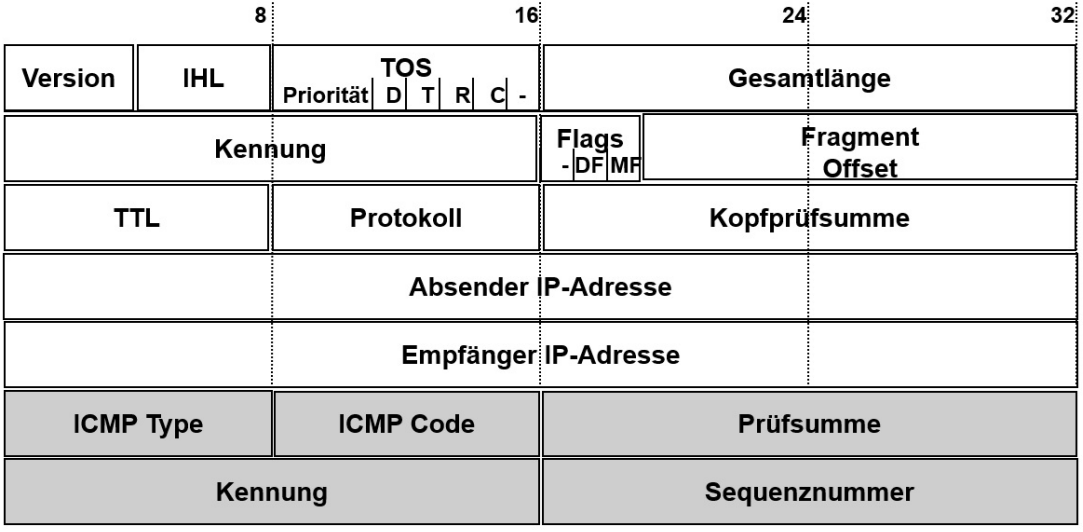
ICMP Type 3 - Destination Unreachable

ICMP Type	Bedeutung
0	Netzwerk unerreichbar
1	Rechner unerreichbar
2	Protokoll unerreichbar
3	Port unerreichbar
4	Fragmentierung erforderlich
6	Zielnetzwerk unbekannt

Aufbau einer ICMP Nachricht



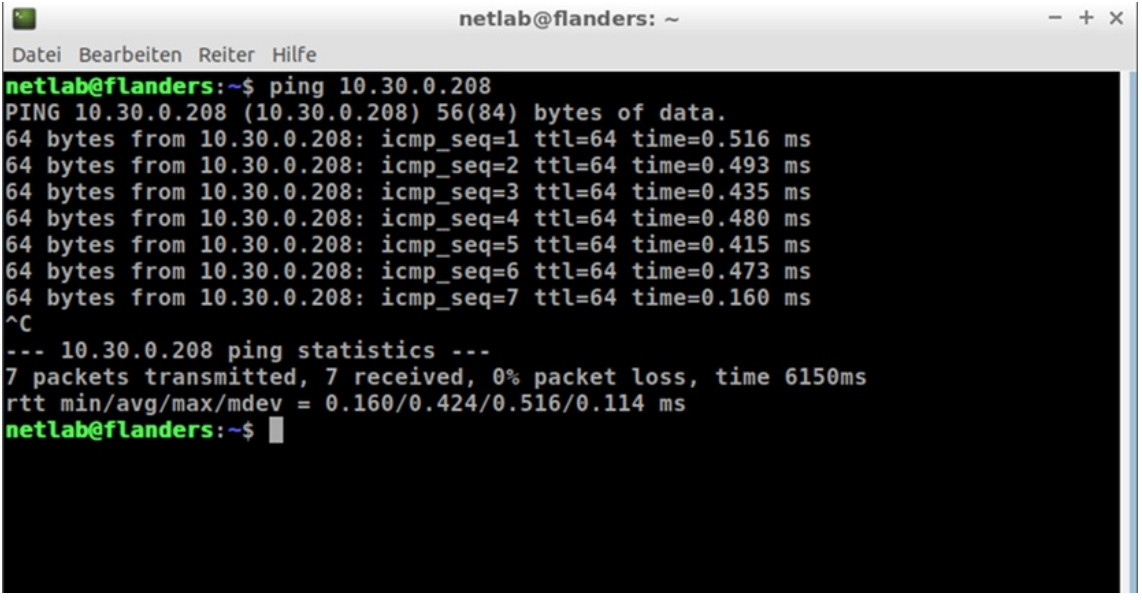
Aufbau einer ICMP Nachricht für Echo Request (Type = 8)



Ping

- Basiert auf den ICMP Typen Echo Request (8) und Echo Response (0) jeweils mit Code = 0
- Kann für grundlegende Verbindungstests verwendet werden
- Kennung und Sequenznummer können von Absender des Echo Request auf einen beliebigen Wert gesetzt werden. Diese Werte werden im Echo Reply zurückgesendet.
- Bei mehrfachen Ping wird die Sequenznummer hochgezählt.

Beispiel aus dem Labor: ping an einen Rechner im gleichen Netz (unter Ubuntu)



Quellen I

[1] Internet Control Message Protocol.
RFC 792, September 1981.

[2] Internet Protocol.
RFC 791, September 1981.

[3] An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware.
RFC 826, November 1982.

[4] A Reverse Address Resolution Protocol.
RFC 903, June 1984.

[5] Domain names - implementation and specification.
RFC 1035, November 1987.

[6] Baker, F., Black, D. L., Nichols, K., and Blake, S. L.
Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.
RFC 2474, December 1998.

[7] Cotton, M., Vegoda, L., Bonica, R., and Haberman, B.
Special-Purpose IP Address Registries.
RFC 6890, April 2013.

[8] Floyd, S., Ramakrishnan, D. K. K., and Black, D. L.
The Addition of Explicit Congestion Notification (ECN) to IP.
RFC 3168, September 2001.

[9] Fuller, V., and Li, T.
Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan.
RFC 4632, August 2006.

[10] Fuller, V., Li, T., Varadhan, K., and Yu, J.
Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy.
RFC 1519, September 1993.

Quellen II

[11] Moskowitz, R., Karrenberg, D., Rekhter, Y., Lear, E., and de Groot, G. J.
Address Allocation for Private Internets.
RFC 1918, February 1996.