Die Vermittlungsebene

Prof. Dr. Norbert Pohlmann
Fachbereich Informatik
Verteilte Systeme und Informationssicherheit

Fachhochschule Gelsenkirchen

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- IPv6
- Zusammenfassung

Inhalt

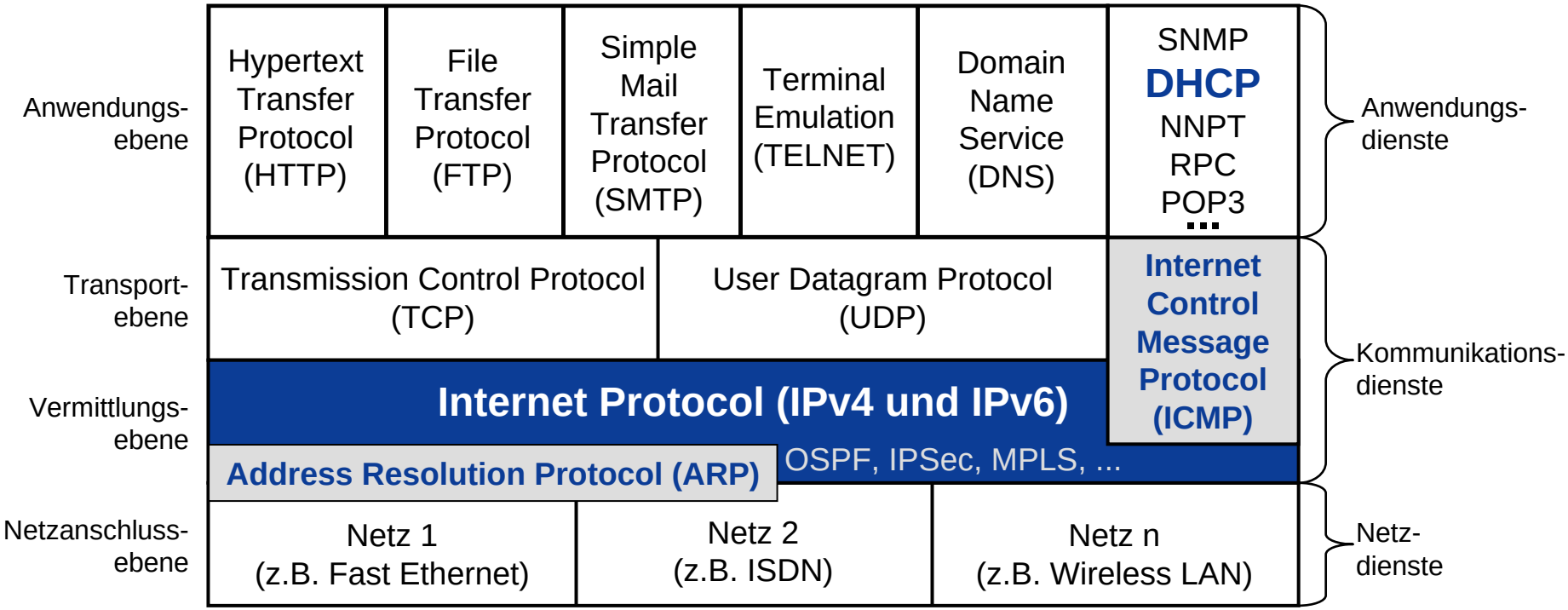
- **Ziele und Einordnung**
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- IPv6
- Zusammenfassung

Die Vermittlungsebene
→ Ziele

- Gutes Verständnis für die Vermittlungsebene in der TCP/IP-Kommunikationsarchitektur
- Erlangen der Kenntnisse über die Aufgaben, Prinzipien und Mechanismen auf der Vermittlungsebene
- Gewinnen von praktischen Erfahrungen über die Vermittlungsebene mit Hilfe von Protokollanalysen und Statistiken (IAS)

Die Vermittlungsebene
→Einordnung

Internet-Protokollstack



Inhalt

- Ziele und Einordnung
- **IP - Internet Protocol (IPv4)**
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- IPv6
- Zusammenfassung

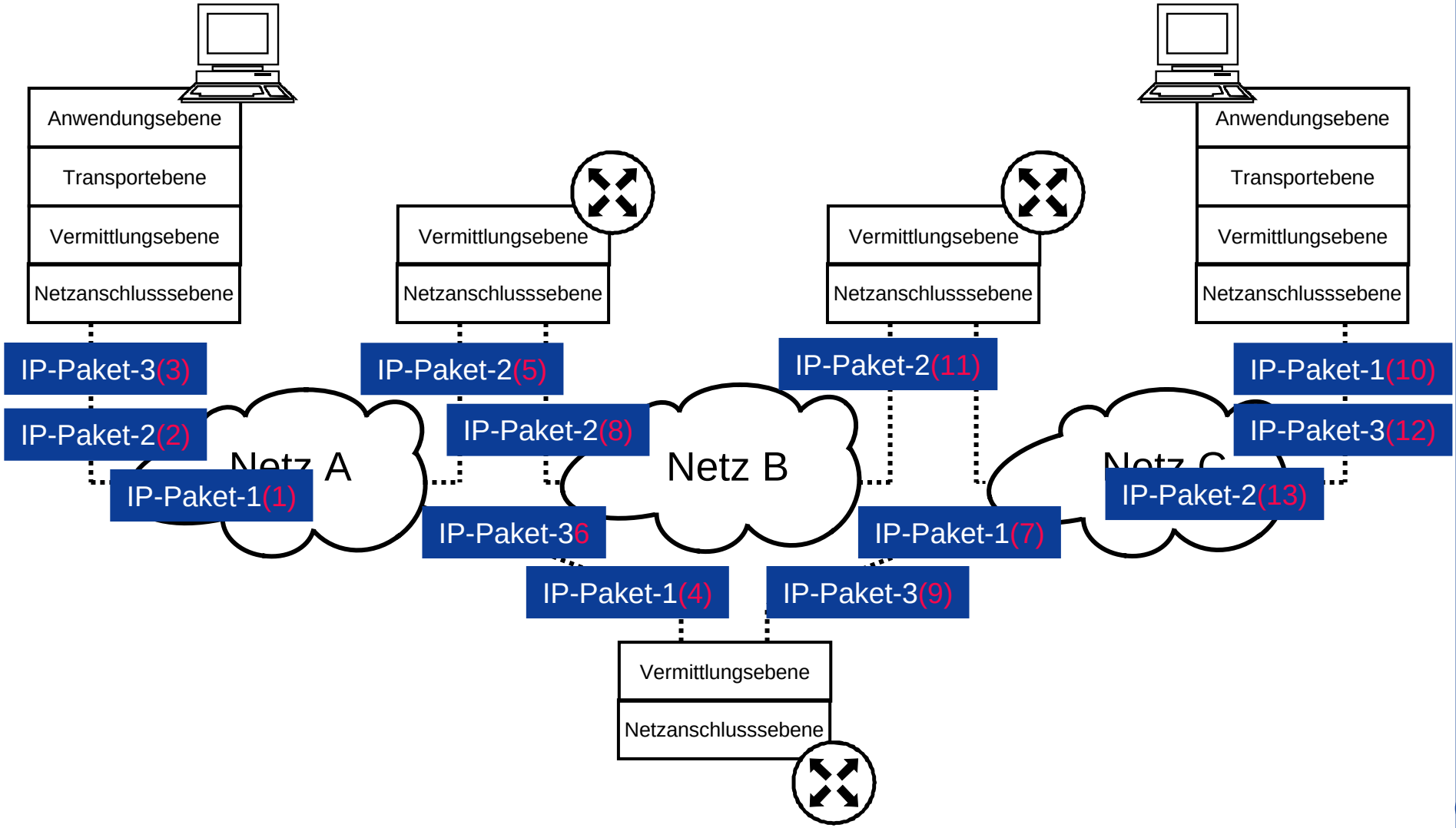
IP - Internet Protocol
→ Standards

RFC 791	IP-Protokoll
RFC 815	IP over X.25 Networks
RFC 894	IP over Ethernet-Networks
RFC 948	IP over 802.3 Networks
RFC 1051	IP over Arcnet-Networks
RFC 1055	IP over Serial Lines („SLIP“)
RFC 1088	IP over Netbios Networks
RFC 1577	IP over ATM Networks („Classical IP“)

IP - Internet Protocol (1/3)

- Das Internet-Protokoll bietet den Protokollen der Transportschicht einen verbindungslosen, **unzuverlässigen Paketübermittlungsdienst**.
- Hauptaufgabe von IP sind die Adressierung von Rechnersystemen, das Fragmentieren von Paketen und das Routing der Pakete.
- Es enthält keine Funktion für die Ende-zu-Ende-Sicherung von Nachrichten oder für die Flußkontrolle.
- Pakete werden so gut wie möglich übertragen (**Best Effort Prinzip**) - **garaniert ist die Zustellung allerdings nicht**.
- Jedes IP-Datagramm wird als einzelnes Paket, völlig unabhängig von anderen Datagrammen, durch das Netz zum Empfänger übertragen.
- Für jedes Datagramm wird innerhalb des Netzes der optimale Weg ermittelt.
- Dabei können sich **Datagramme** auf dem Weg zum Empfänger **überholen** und dadurch in **geänderter Reihenfolge** beim Empfänger eintreffen.

IP - Internet Protocol (2/3)



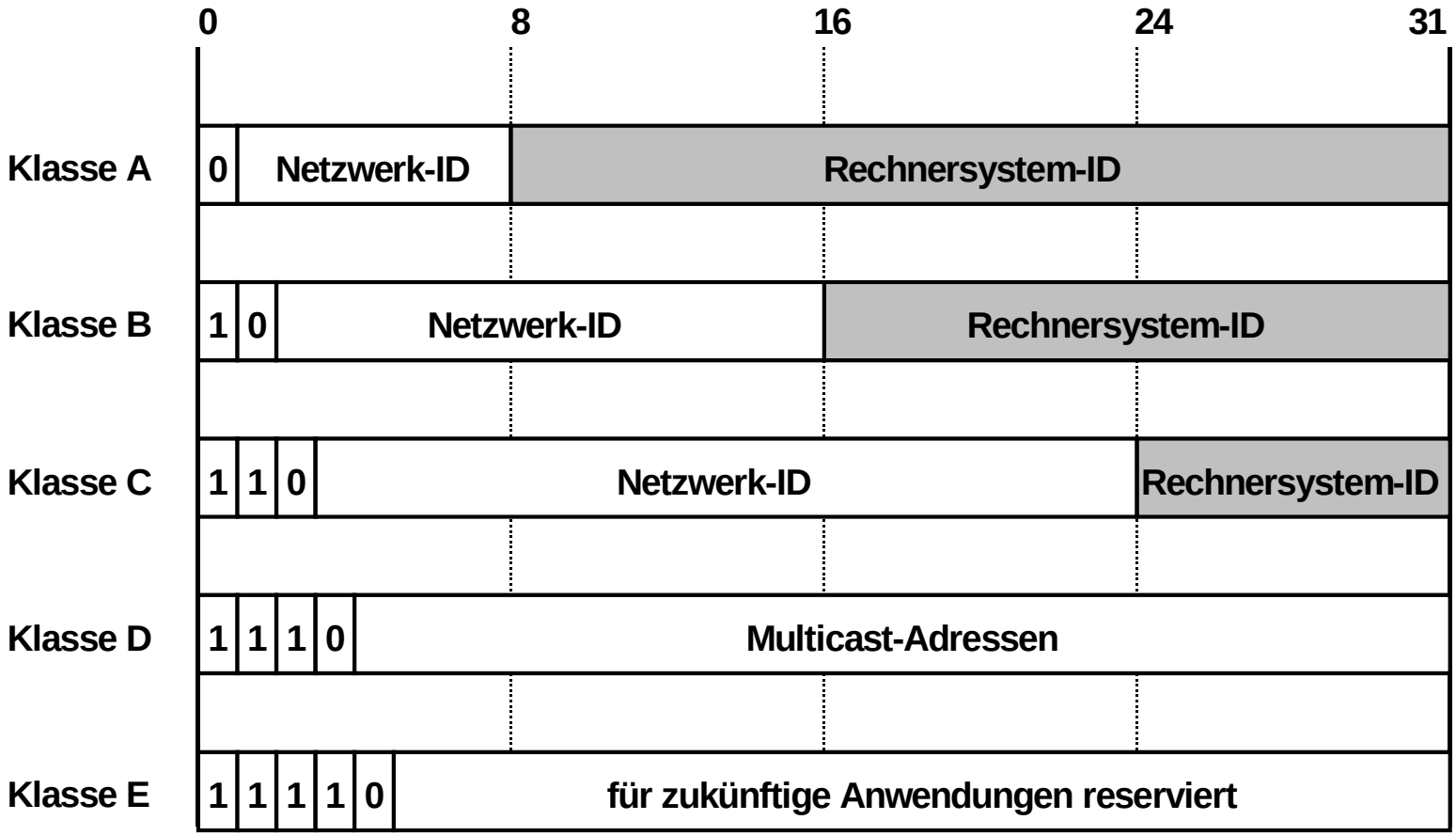
IP - Internet Protocol (3/3)

- Die Aufgabe, die Pakete in die **richtige Reihenfolge** zu bringen, muss die **Transportschicht** übernehmen.
- **IP stellt keine gesicherte Verbindung zur Verfügung**, sondern verlässt sich darauf, dass die Protokolle der höheren Ebenen die Ende-zu-Ende-Kontrolle gewährleisten.
- **IP ist nicht in der Lage,** verlorene oder von der Netzanschlussebene abgelehnte **Datagramme neu zu generieren** und erneut zu übertragen.
- Die Definition des IP-Protokolls beschreibt das Format des IP-Datagramms, die einzelnen Felder im Header und den Ablauf der Datenübermittlung.

IP-Adressen (1/12)

- Zur Adressierung eines Kommunikationspartners kommen auf IP-Ebene **32-Bit lange Adressen** (IP-Adressen) zum Einsatz (IPv4).
- In diesem Zusammenhang ist wichtig, dass eine IP-Adresse die **Verbindung eines Rechners zum Netz identifiziert**, nicht den Rechner selbst.
- Dies hat zur Folge, dass ein Rechner, der gleichzeitig an **zwei oder mehr Netzen** angeschlossen ist (ein sogenannter multihomed host), auch **mehrere IP-Adressen** benötigt.
- Eine weitere Folge ist, dass ein Rechner, der in ein **anderes Netz** verlegt wird, auch **eine neue IP-Adresse** benötigt.
- **Jede Adresse besteht aus zwei Teilen:**
  - einem Teil, der das Netzwerk bezeichnet, in dem sich der Rechner befindet
    - **Netzwerk- ID (netid)**
  - und einem weiteren Teil, der den Rechner im Netz adressiert
    - **Rechnersystem- ID (hostid)**

IP-Adressen (2/12)



IP-Adressen (3/12)

- In den Klassen A, B und C ist die Netzwerk-ID (netid) entweder 8, 16 oder 24 Bit lang, woraus sich die Anzahl der Netze und der Rechner in einem Netz ergibt:

- **Klasse A:**
  → 8 Bit Netzadresse → 2^7 = 128 A-Netze (127 für Loopback)
  → 24 Bit Rechneradresse → 2^{24} = 16.777.261 Rechner

- **Klasse B:**
  → 16 Bit Netzadresse → 2^{14} = 16.384 B-Netze („10“)
  → 16 Bit Rechneradresse → 2^{16} = 65.536 Rechner

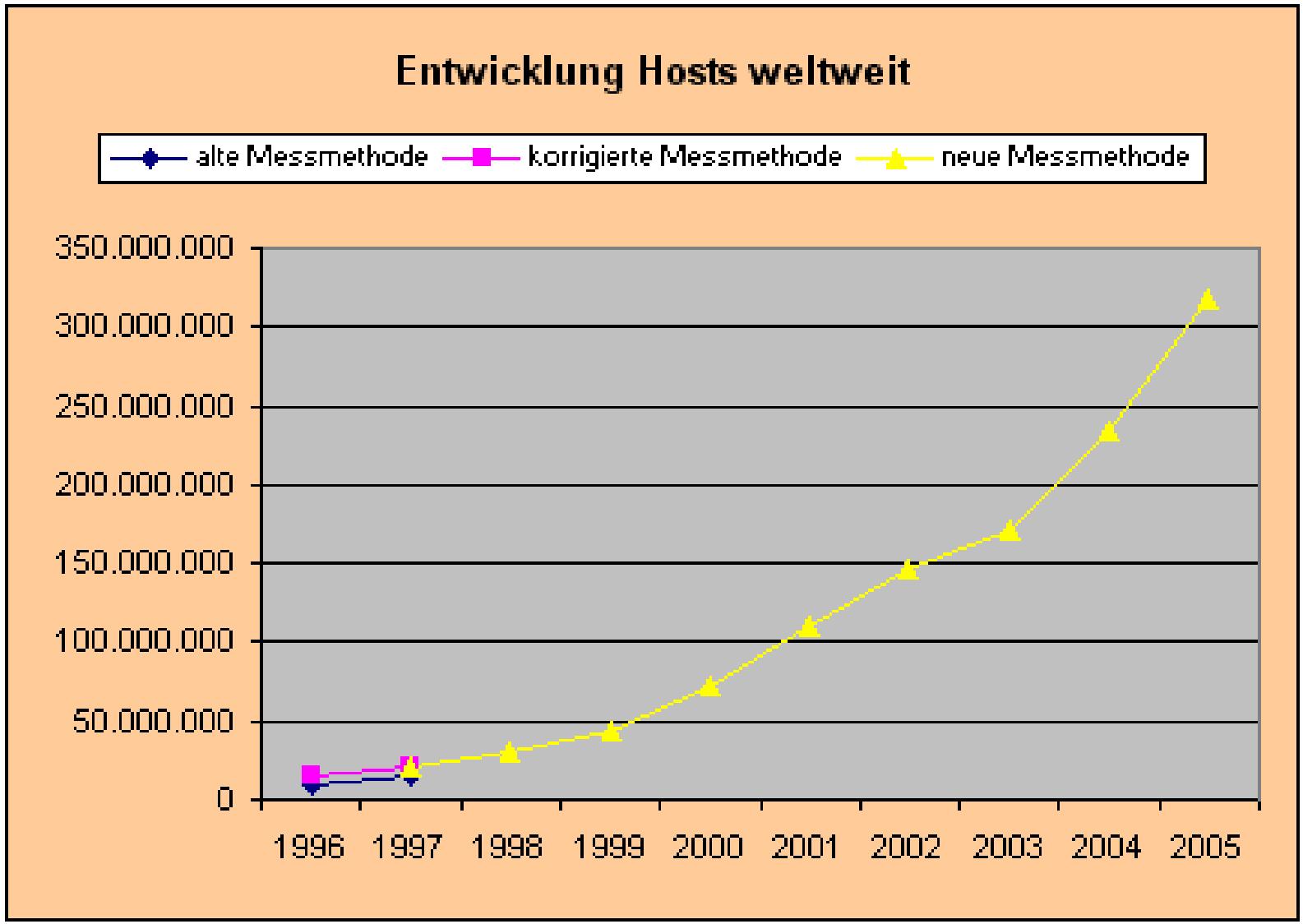
- **Klasse C:**
  → 24 Bit Netzadresse → 2^{21} = 2.097.152 C-Netze („110“)
  → 8 Bit Rechneradresse → 2^8 = 256 Rechner

- **Summe alle Adressen:** 3.758.096.384 (theoretisch: $2^{31}+2^{30}+2^{29}$)

- **Rechner im Internet (2005):**

Weltweit:	320.000.000
Europa:	26.000.000
Deutschland:	3.000.000

IP-Adressen (4/12)



IP-Adressen (5/12)

- **Besonderheiten:**
  - Eine IP-Adresse, bei der alle Bits der Rechnersystem-ID (hostid) „0“ sind, bezeichnet nicht einen einzelnen Rechner, sondern ist reserviert, um das gesamte IP-Netz zu bezeichnen.
  - Sind alle Bits der Rechnersystem-ID (hostid) „1“, ist diese Adresse die Broadcastadresse des Netzes und adressiert alle Rechner in diesem Netz.
- Notiert werden IP-Adressen meist als 4, durch einen Punkt getrennte Dezimalzahlen, diese Darstellung wird auch als **„dotted decimal notation“** bezeichnet.
- Jede Dezimalzahl repräsentiert dabei 8 Bit der IP-Adresse, kann also einen Wert zwischen 0 und 255 annehmen.
- **Beispiel:**

  10000110 01101100 00111000 00111100 = **134.108.56.60**

IP-Adressen (6/12)
→ Verfügbarer Adressbereich

Klasse	niedrigste Adresse	höchste Adresse
A	1.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

FH FB5 (Informatik) hat ein: Klasse C-Netz: 194.94.127.0

IP-Adressen (7/12)

- Um sicherzustellen, dass die Netzwerk-ID (netid) eines an das INTERNET angeschlossenen Netzes auf der ganzen Welt eindeutig ist, werden die IP-Adressen von einer zentralen Organisation, der Internet Assigned Number Authority (IANA), vergeben.
- Wenn eine Organisation ihr Netz an das INTERNET anschließen möchte, muss sie diese Netz-IP-Adresse vom **Internet Network Information Center (INTERNIC)** erwerben.
- Die Adressen innerhalb des erworbenen Adressbereiches kann jede Organisation selbst vergeben, allein **der Teil der Adresse, der das Netz bezeichnet**, wird von der **IANA** vergeben.
- Aufgrund der begrenzten Anzahl der zur Verfügung stehenden Klasse A- und Klasse B-Adressen sind derzeit fast nur noch Klasse C-Adressen erhältlich, so dass auch Organisationen mit mehr als 254 Rechnern für ihr Netz Klasse C-Adressen zugeteilt bekommt.

IP-Adressen (8/12)
→ Address Allocation For Private Internets

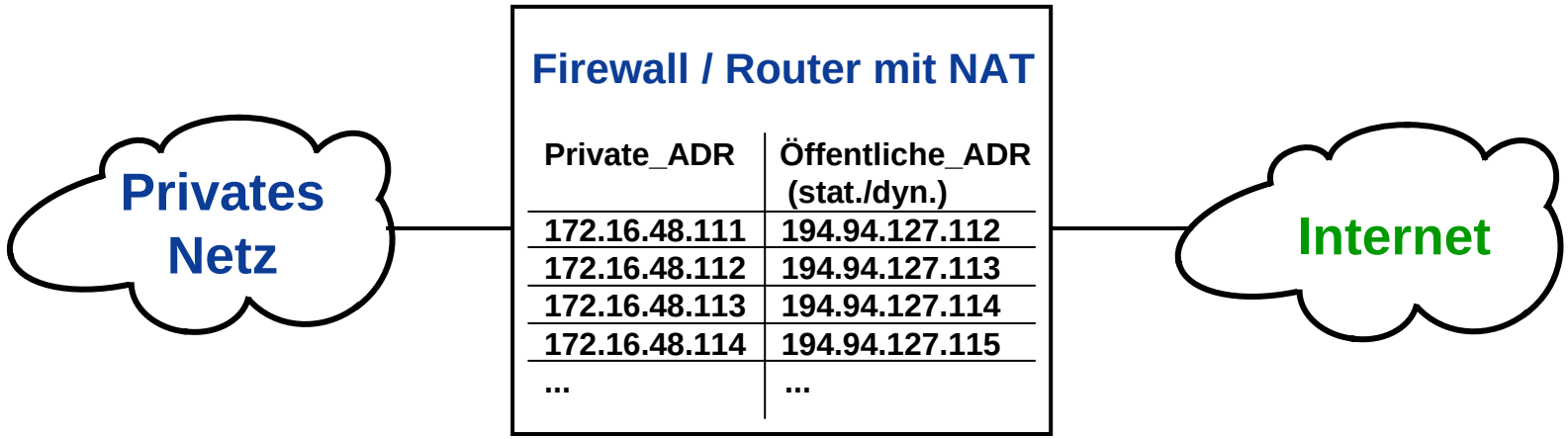
- Neben den offiziellen Adressen sind einige Adressbereiche für den **privaten Gebrauch** reserviert und werden nicht offiziell vergeben.
- Diese Adressen sollten immer verwendet werden, wenn für ein nicht am Internet partizipierendes Netz IP-Adressen benötigt werden.
- In RFC 1918 - **Address Allocation For Private Intranets** - sind die folgenden Adressbereiche für den privaten Gebrauch spezifiziert worden.

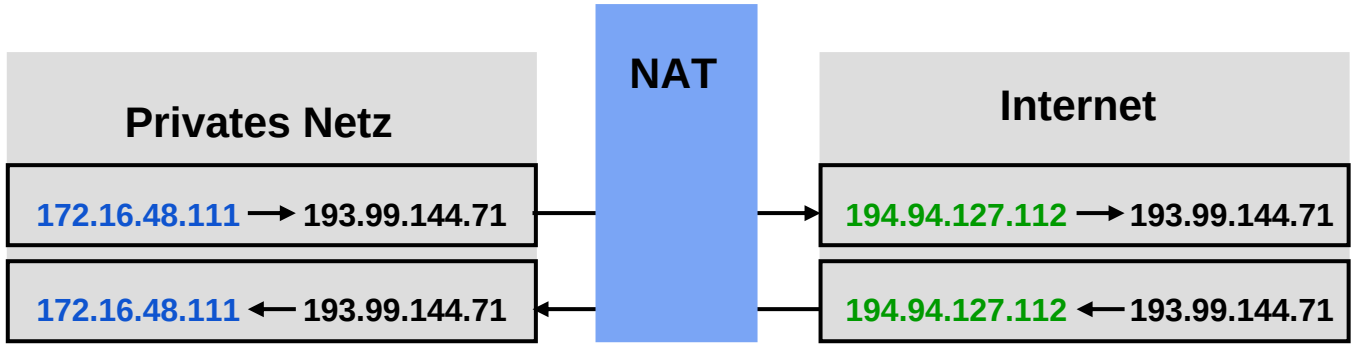
  10.0.0.0 - 10.255.255.255 (ein Klasse A-Netz)
  172.16.0.0 - 172.31.255.255 (16 Klasse B-Netz)
  192.168.0.0 - 192.168.255.255 (255 Klasse C-Netz)

  FH FB5 (Informatik) arbeitet mit: Klasse B-Netz: 172.16.0.0

- Soll das Netz zu einem späteren Zeitpunkt an das Internet angeschlossen werden, so kann dies ohne eine Änderung der bereits vergebenen IP-Adressen geschehen.
- Dies wird durch das **Network Address Translation (NAT)** erreicht.
- Das Gateway (oft eine Firewall), das die Verbindung zum Internet darstellt, übersetzt dabei die internen, privaten Adressen in eine offizielle IP-Adresse.

IP-Adressen (9/12)
→ Network Address Translation (NAT)





IP-Adressen (10/12)
→ NAPT (Network Address Port Translation) – (1/3)

- **NAPT (Network Address Port Translation)** oder **PAT (Port Address Translation)** ist eine spezielle Form von NAT und wird zumeist verwendet, um mehreren Rechnersystemen in einem Local Area Network (LAN) Zugriff auf das Internet zu ermöglichen.
- Dabei werden im Gegensatz zu NAT nicht nur die **IP-Adressen**, sondern auch **Port-Nummern** umgeschrieben.
- Wird jede IP-Adresse zu einer einzigen IP-Adresse übersetzt, spricht man von einer **N:1-Übersetzung**.
- Werden mehrere IP-Adressen zu weniger IP-Adressen abgebildet, handelt es sich um eine **N:M-Übersetzung**.

IP-Adressen (11/12)
→ NAPT (Network Address Port Translation) – (2/3)

lokales Netz (LAN)

Quell IP:Port	Ziel IP:Port
192.168.0.2:5000	170.0.0.1:80
192.168.0.3:5000	170.0.0.1:80
192.168.0.5:5001	170.0.0.1:80

Router
==========>
Masquerading

öffenliches Netz (WAN)

Quell IP:Port	Ziel IP:Port
205.0.0.2:6000	170.0.0.1:80
205.0.0.2:6001	170.0.0.1:80
205.0.0.2:6002	170.0.0.1:80

- Die Quell-IP-Adressen werden durch die (einzige) öffentliche IP-Adresse ersetzt.
- Die internen Port-Nummern werden durch eindeutige öffentliche Port-Nummern ersetzt.
- Mittels einer Tabelle merkt sich der Router jeweils die interne Quell-IP-Adresse samt Port-Nummer und die öffentliche Port-Nummer des ausgehenden Pakets:
  192.168.0.2:5000 <=> 6000
  192.168.0.3:5000 <=> 6001
  192.168.0.5:5001 <=> 6002

IP-Adressen (12/12)
→ NAPT (Network Address Port Translation) – (3/3)

lokales Netz (LAN)

Quell IP:Port	Ziel IP:Port
170.0.0.1:80	**192.168.0.2:5000**
170.0.0.1:80	**192.168.0.3:5000**
170.0.0.1:80	**192.168.0.5:5001**

Router
<==========
Masquerading

öffenliches Netz (WAN)

Quell IP:Port	Ziel IP:Port
170.0.0.1:80	**205.0.0.2:6000**
170.0.0.1:80	**205.0.0.2:6001**
170.0.0.1:80	**205.0.0.2:6002**

- Bei eingehenden Paketen kann anhand der Port-Nummer der Ziel-IP und des Tabelleneintrags (connection tracking) festgestellt werden, welches Rechnersystem die Pakete angefordert hatte (hier: 192.168.0.2, 192.168.0.3 und 192.168.0.5).
- Der Router kann dadurch die Ziel-IP durch die ursprüngliche Quell-IP 192.168.0.2, 192.168.0.3 bzw. 192.168.0.5 und die öffentliche Port-Nummer die ursprüngliche interne Port-Nummer austauschen.

Subnetze und Netzmasken (1/6)

- An ein Klasse A Netz können 2^{24}= 16.777.261 Rechner angeschlossen werden.
- Es ist leicht einzusehen, dass dies **nicht sinnvoll** ist, da alle diese Rechner **an ein physikalisches Netzwerk angeschlossen** sein müssen.
- Ein Router setzt ja bereits zwei IP-Netze voraus!
- Daher gibt es die Möglichkeit, ein bestehendes Netz weiter in sogenannte **Subnetze** zu unterteilen.

- Falls ein Unternehmen mit Klasse C Netzen arbeitet, besteht das Problem darin, dass jedesmal, wenn ein neues Netz aufgebaut werden soll der Systemverwalter eine neue Netznummer von der „NIC“ einholen und diese weltweit angekündigt werden muss.

Subnetze und Netzmasken (2/6)

- **Subnetze** werden dadurch geschaffen, dass ein **Teil der Rechnersystem-ID (hostid)** für die Subnetz (subnetid) Adressierung verwendet wird.
- Wie viele Bits der Rechnersystem-ID (hostid) als Subnetz-ID (subnetid) verwendet werden kann der Systemverwalter selber festlegen.
- Damit ist der Systemverwalter **unabhängig von der „NIC“.**
- Es ist allerdienst nicht mehr möglich, anhand der Adressklasse zu unterscheiden, welcher Teil der Adresse das Netz bezeichnet.
- Um festzulegen, welcher Teil der IP-Adresse zur Rechnersystem-ID (hostid) mit Subnetz-ID (subnetid) gehört, wird eine **Netzmaske** oder auch **Subnetzmaske** benötigt.
- Möchte man z.B. bei einem Klasse A-Netz das komplette zweite Octet dazu benutzen, Subnetze zu bilden, so muss die Netzmaske den Wert 255.255.0.0 haben.
- Durch eine **logische UND** Verknüpfung der Netzmaske mit einer IP-Adresse kann auf einfachem Weg die IP-Adresse in Rechnersystem-ID (hostid) und Netzwerk-ID (netid) zerlegt werden.

Subnetze und Netzmasken (3/6)

IP-Adresse	00001001	00100110	10011101	00010101 = 9.38.157.21
Netzmaske	11111111	11111111	00000000	00000000 = 255.255.0.0

Netzwerk-ID (netid) **IP-Adresse *UND* Netzmaske**	00001001	00100110	00000000	00000000 = 9.38.0.0
Rechnersystem-ID (hostid)	00000000	00000000	10011101	00010101 = 0.0.157.21

- Durch die **UND-Verknüpfung** der **IP-Adresse mit der Netzmaske** wird die **Netzwerk-ID (**netid) ermittelt.
- Die **Rechnersystem-ID** (hostid) kann durch eine **UND-Verknüpfung** der **IP-Adresse mit der invertierten Netzmaske** ermittelt werden.
- Das Beispiel zeigt das IBM Klasse A Netz (9.0.0.0), das mit Hilfe der Netzmaske 255.255.0.0 in 256 Subnetze unterteilt wird.

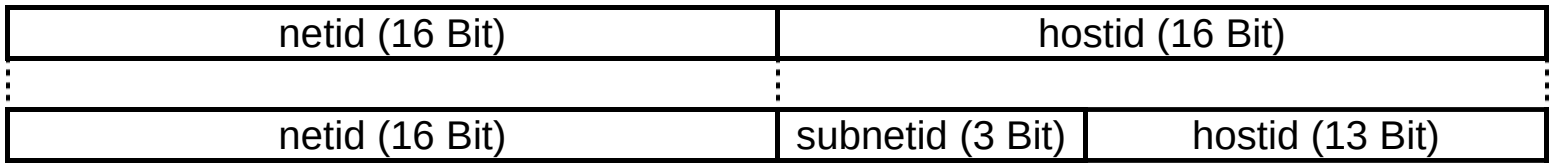
Subnetze und Netzmasken (4/6)
→ Beispiel

- Ein Klasse B Netz mit der IP-Adresse 134.108.0.0 soll mit Hilfe einer Netzmaske in 8 Subnetze unterteilt werden.
- Wie teilt sich die Rechnersystem-ID (hostid) in den Subnetzen auf?

Subnetz Nr.	niedrigste Adresse	höchste Adresse
0		
1		
2		
3		
4		
5		
6		
7		

Subnetze und Netzmasken (5/6)
→ Lösung (1/2)

- Um die Klasse B-Adressen in 8 Subnetze zu unterteilen, müssen 3 Bit der Rechnersystem-ID (hostid) als Subnetz-ID (subnetid) benutzt werden.
- Die Netzmaske muss als die 16 Bit der Klasse B Adresse und zusätzlich 3 Bit der Rechnersystem-ID (hostid) maskieren.
- Das Netz erscheint von außen nach wie vor als das Klasse B-Netz.
- Die benötigte Netzmaske muss die 3 höchstwertigen Bit des 3. Octet und die 16 Bit der netid ausmaskieren.



- Netzmaske 11111111 11111111 11100000 00000000 = 255.255.224.0

- Bei einer „subnetid“ von 4 Bit, ist die Netzmaske:
  Netzmaske 11111111 11111111 11110000 00000000 = 255.255.240.0

Subnetze und Netzmasken (6/6)
→ Lösung (2/2)

- Das Klasse B-Netz ist in 8 gleich große Subnetze unterteilt worden.

Subnetz Nr.	niedrigste Adresse	höchste Adresse
0	134.108.0.0	134.108.31.255
1	134.108.32.0	134.108.63.255
2	134.108.64.0	134.108.95.255
3	134.108.96.0	134.108.127.255
4	134.108.128.0	134.108.159.255
5	134.108.160.0	134.108.191.255
6	134.108.192.0	134.108.223.255
7	134.108.224.0	134.108.255.255

CIDR
→ Classless Inter-Domain Routing (1/3)

- Classless Inter-Domain Routing (CIDR) beschreibt ein Verfahren zur **effizienteren Nutzung** des bestehenden 32-Bit-IP-Adress-Raumes.
- Es wurde 1993 eingeführt, um die Größe von Routing-Tabellen zu reduzieren und um die **verfügbaren Adressbereiche besser auszunutzen**.
- Mit CIDR **entfällt** die feste **Zuordnung einer IP-Adresse zu einer Netzklasse** und die eventuelle Unterteilung (Subnetting) in weitere Netze.
- Es existiert nur noch **eine Netzmaske**, welche die IP-Adresse in den Netzwerk- und Hostteil aufteilt.

CIDR
→ Classless Inter-Domain Routing (2/3)

- Bei CIDR führte man als neue Notation eine so genannte **Suffix** ein.
- Das **Suffix** gibt die **Anzahl der 1-Bits in der Netzmaske** an.
- Diese Schreibform ist viel kürzer als die dotted decimal notation und auch eindeutig.
- Beispiel:

  CIDR: 192.168.0.0/24

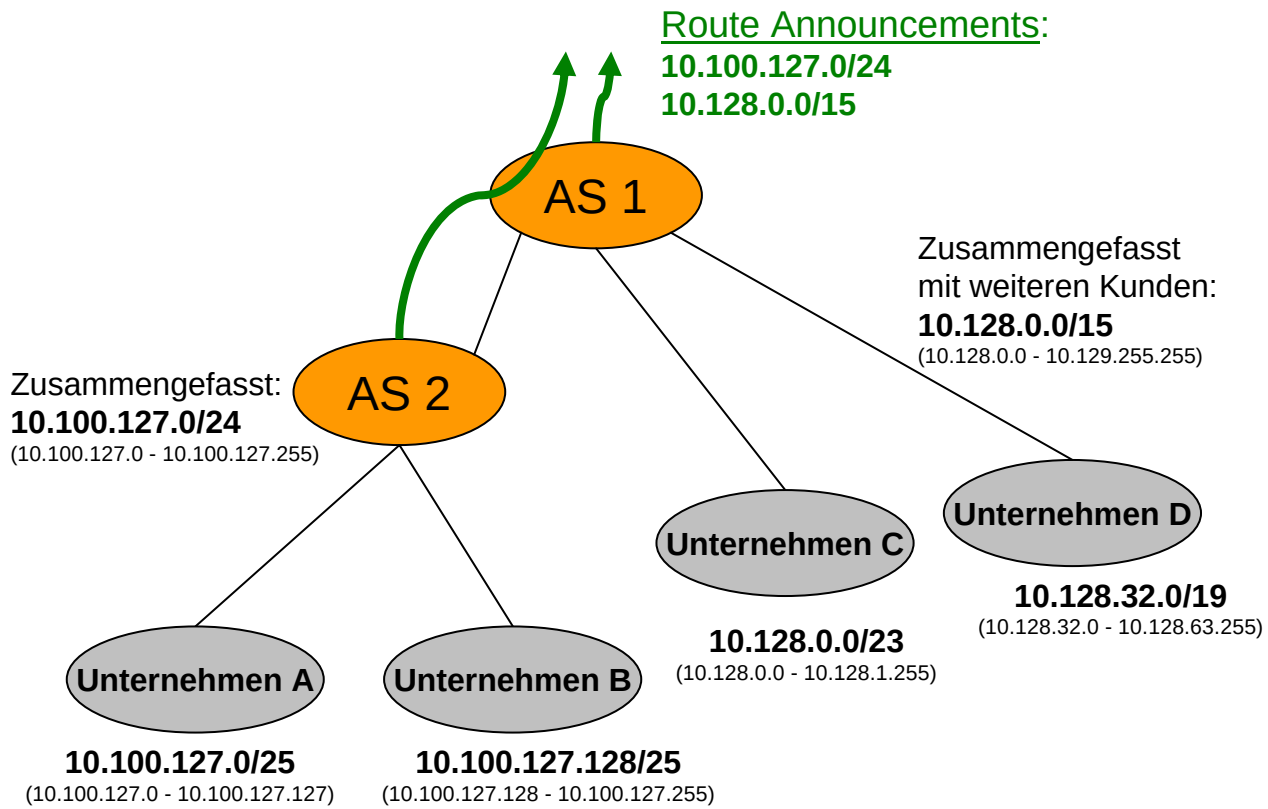
  Altes Verfahren: 192.168.0.0 mit der Netzmaske 255.255.255.0

- Netzmaske (binärer Schreibweise):
  11111111.11111111.11111111.00000000,

  Anzahl der 1-Bits 3*8 = 24
  wird mit Suffix angegeben

CIDR
→ Classless Inter-Domain Routing (3/3)

- Beispiel: Route Announcement



Subnetze und Netzmasken
→ Beispiele

Subnetze im Fachbereich Informatik

Offizielle Internet Adressen: Klasse C-Netz, Netzmaske: 255.255.255.224

194.94.127.0	–	194.94.127.31	FB5-Internet
194.94.127.32	–	194.94.127.63	ISP
194.94.127.64	–	194.94.127.79	*Tecmedic*
194.94.127.80	–	194.94.127.95	*DMZ2*
194.94.127.96	–	194.94.127.111	DMZ3
194.94.127.112	–	194.94.127.127	*Firewall mit NAT*

3 Bit - 8 Netze

Private IP Adressen: Klasse B-Netz, Netzmaske: 255.255.255.240

172.16.0.0	-	172.16.15.255	*FB5-Intranet*
172.16.16.0	-	172.16.16.255	*DMZ4*
172.16.17.0	-	172.16.17.255	*DMZ1*
172.16.32.0	-	172.16.47.255	*L&F 501*
172.16.48.0	-	172.16.63.255	*L&F 502*
172.16.64.0	-	172.16.79.255	*L&F 503*
172.16.80.0	-	172.16.95.255	*L&F 504*
172.16.96.0	-	172.16.111.255	*L&F 505*

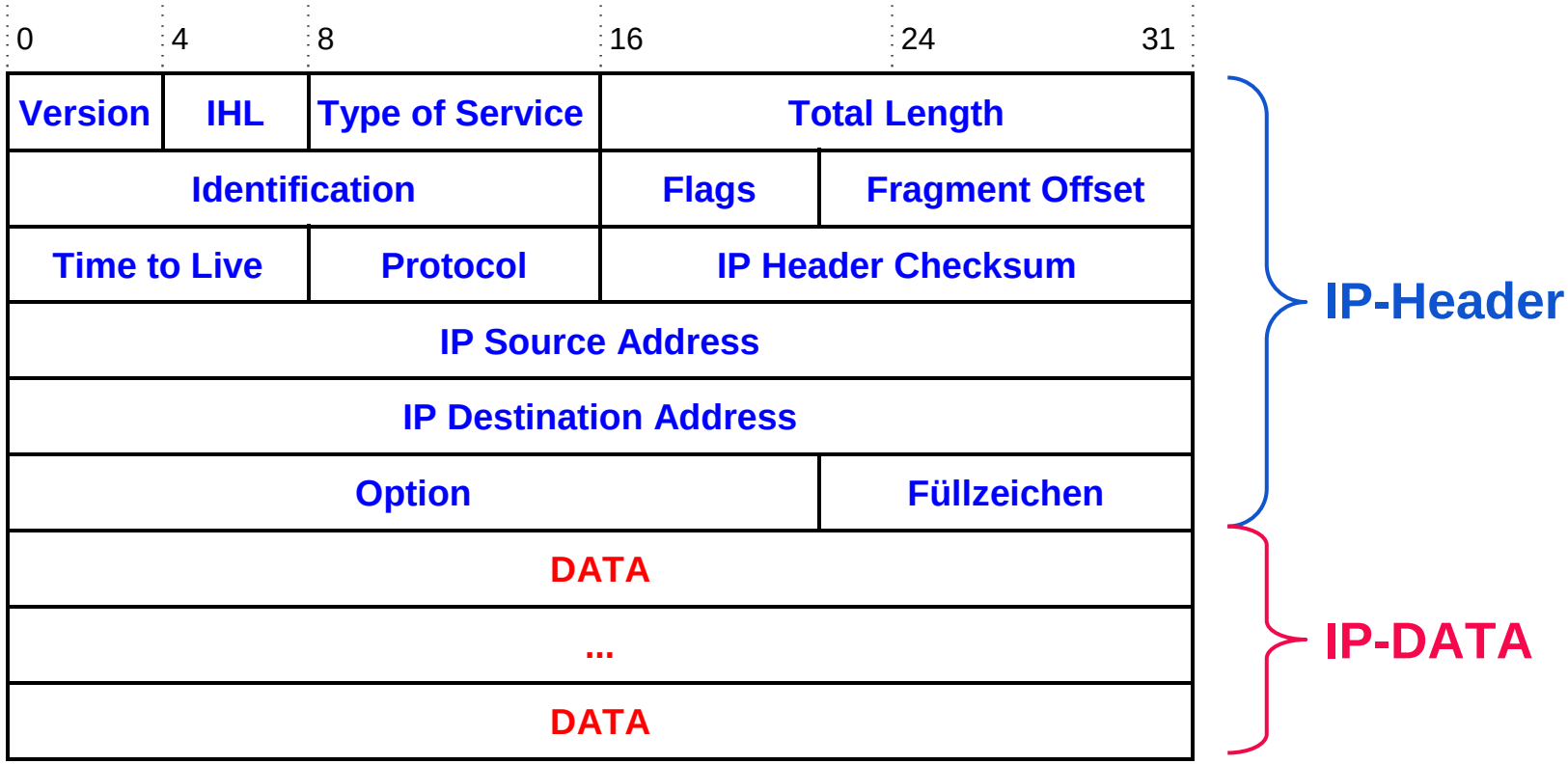
4 Bit - 16 Netze

Subnetze und Netzmasken
→ Beispiele

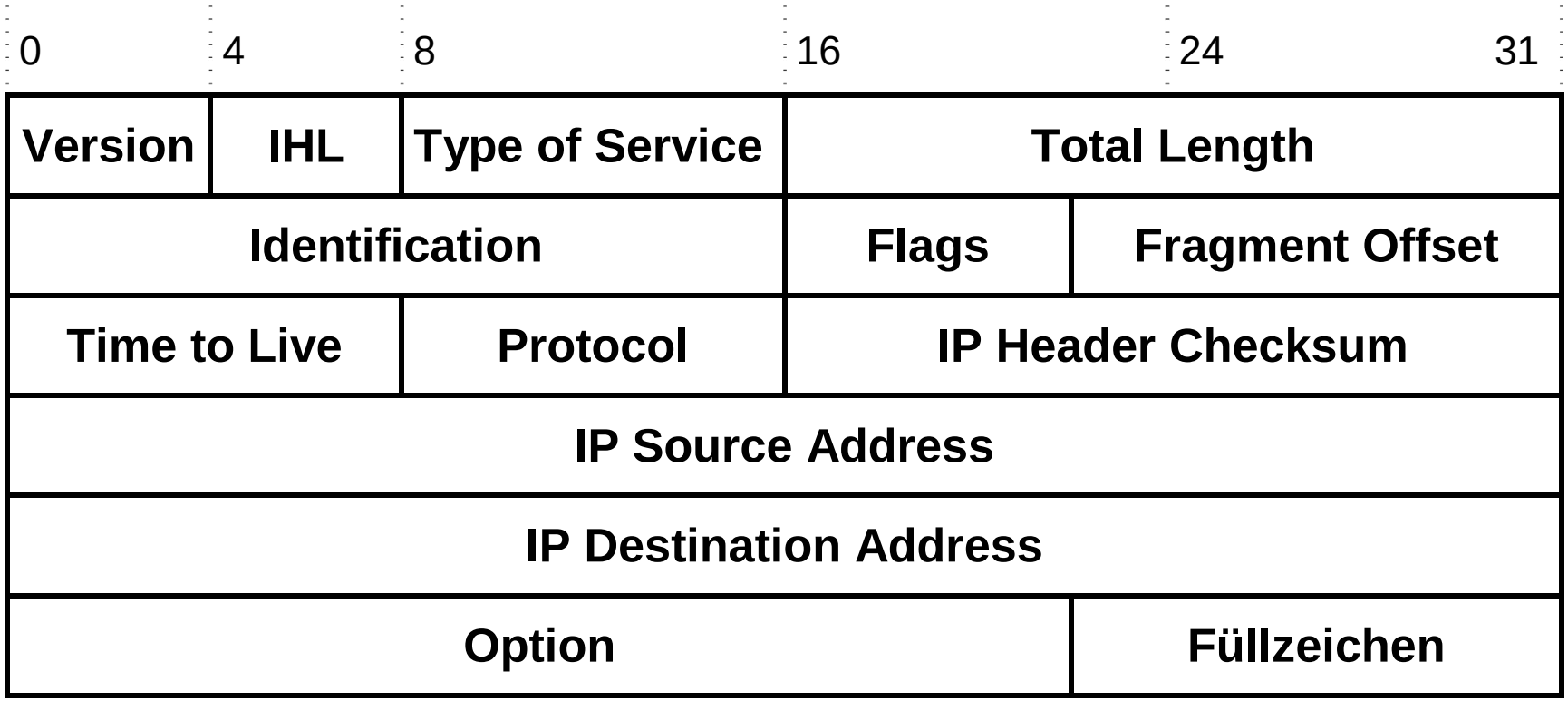
Verteilung der IP- Adressen im Subnetz *FB5-Intranet*

172.16.0.0	-	172.16.0.255	Geräte im Backbone des FB5
172.16.1.0	-	172.16.1.255	Ursprünglich für Dekanat
172.16.2.0	-	172.16.3.255	L&F 501
172.16.4.0	-	172.16.5.255	L&F 502
172.16.6.0	-	172.16.7.255	L&F 503
172.16.8.0	-	172.16.9.255	L&F 504
172.16.10.0	-	172.16.10.255	L&F 505
172.16.11.0	-	172.16.11.255	L&F 506
172.16.12.0	-	172.16.15.255	frei

IP-Paket, IP-Datagramm

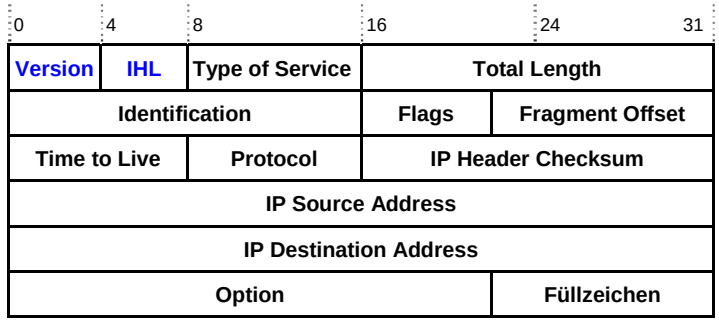


Aufbau des IP-Headers



RFC 791

Feldelemente des IP-Headers (1/8)

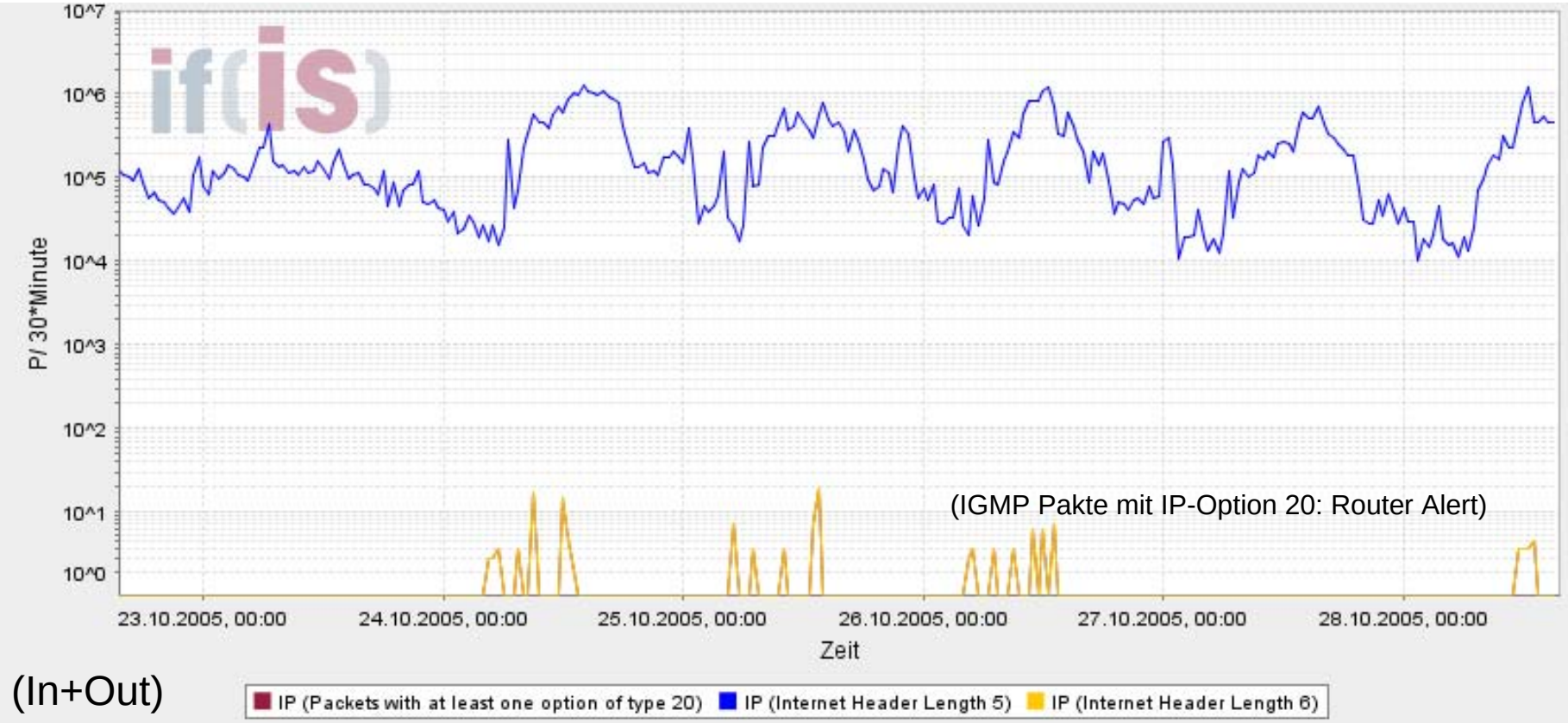


- **Version (Vers)**
  - Feldlänge: 4 Bit
  - **Beschreibung**
    - Das Versions-Feld gibt die verwendete Version des IP-Protokolls an.
    - In der Regel wird heute noch die **Version 4** verwendet, die Standardisierung der Version 6 ist jedoch abgeschlossen. (Codierung Version 4 = 4)

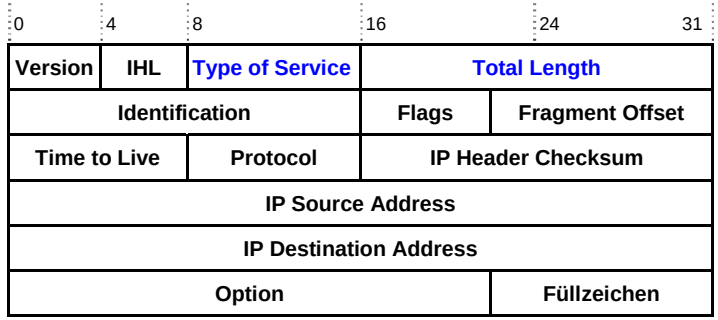
- **Internet Header Length (IHL)**
  - Feldlänge: 4 Bit, Einheiten: 4 Octet-Gruppen, Bereich: 5-15 (Standard: 5)
  - **Beschreibung**
    - Die Internet Header Length steht für die gesamte Länge des IP-Headers, ausgedrückt in 32-Bit-Einheiten (4 Octets).
    - Das IHL Length-Feld ist durch die variable Länge des Optionen-Feldes im IP-Header erforderlich, der am häufigsten verwendete Header hat die Länge von 20 Byte, das IHL-Feld enthält dann also den **Wert 5** (Mindestwert).

Internet-Analyse-System: FB Informatik
→ Internet Header Length (IHL)

- Nur Länge 5 und 6 vorhanden

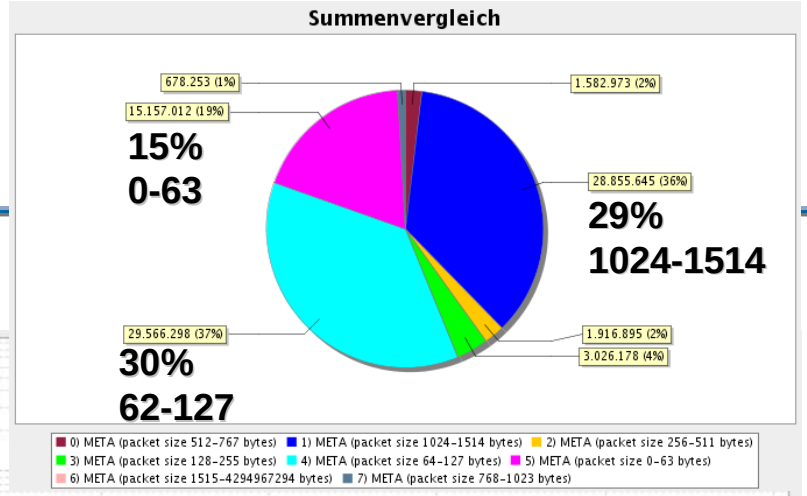


Feldelemente des IP-Headers (2/8)



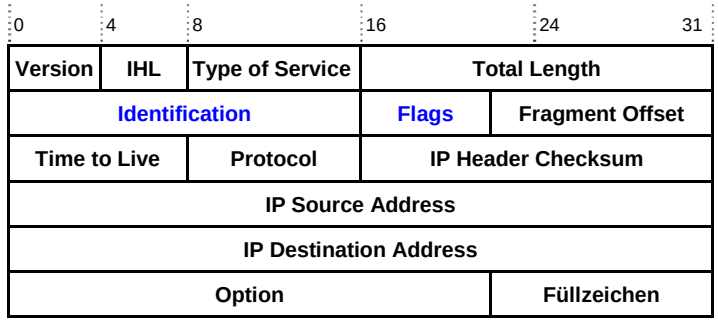
- **Type of Service (TOS)**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Dieses Feld gibt die gewünschte **Qualität des Dienstes** (Vorrang, Verzögerung, Durchsatz und Zuverlässigkeit) für dieses Datagramm an.
    - Wird auch als Diffenrentiated Service Code Point (DSCP) bezeichnet, siehe z.B. Differentiated Services (DiffServ) – REN2-Vorlesung: Quality of service
    - Die angesprochenen Netzknoten können entweder den gewünschten Dienst erbringen oder leisten diesen Dienst oder Teile davon nicht.
- **Total Length (TL)**
  - Feldlänge: 16 Bit, minimaler Wert: 20
  - **Beschreibung**
    - Dieses Feld gibt die Länge des Datagramms (sowohl **Kopf als auch Benutzerdaten**), gemessen in Octets an.
    - Da dieses Feld 16 Bit lang ist, kann ein IP-Paket inklusive Header maximal 2^{16} oder 65.535 Octets lang sein.

IAS: FB Informatik
→ Total Length (TL)



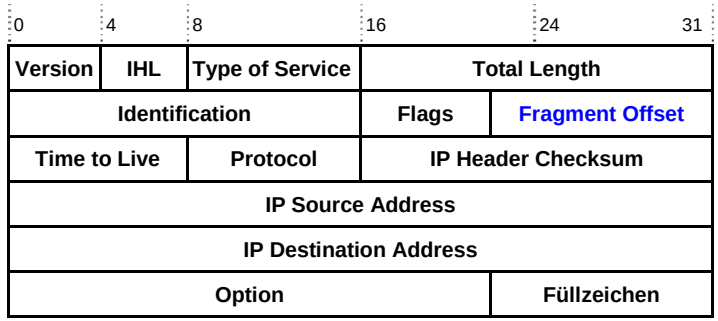
(In+Out)

Feldelemente des IP-Headers (3/8)



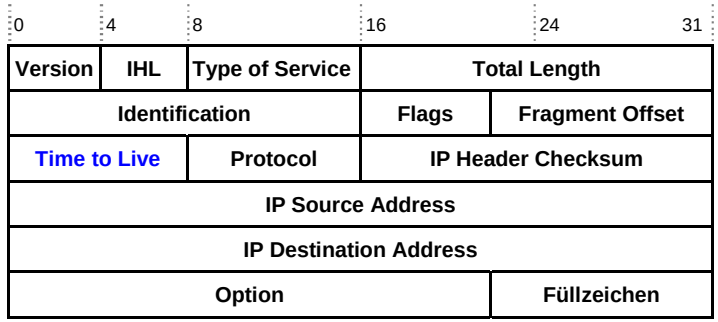
- **Identification (ID)**
  - Feldlänge: 16 Bit
  - **Beschreibung**
    - Dieses Feld enthält eine eindeutige Identifikation des IP-Paketes, z.B. einen **Zähler**, der durch den absendenden Host vergeben wird.
    - Dieses Feld wird bei der Reassemblierung von Fragmenten verwendet, um alle Teile einer Fragmentkette identifizieren zu können.
- **Flags**
  - Feldlänge: 3 Bit
  - **Beschreibung**
    - Zwei Bits namens DF („don´t fragment“ - 2. Bit) und MF („more fragment“ - 3. Bit) steuern die Behandlung des Paketes im Falle der Fragmentierung.
    - Ist das DF-Bit gesetzt, darf das IP-Paket unter keinen Umständen fragmentiert werden, auch wenn es dann nicht mehr weiter transportiert werden kann und verworfen werden muss.
    - Das erste Bit dieses Felds ist ungenutzt.

Feldelemente des IP-Headers (4/8)



- **Fragment Offset (FO)**
  - Feldlänge: 13 Bit, Einheiten 8 Octet
  - **Beschreibung**
    - Dieses Feld gibt die Lage der Fragmentdaten relativ zum Anfang des Datenblockes im ursprünglichen Datagramm an.
    - Bei einem **nicht fragmentierten** Datagramm oder beim ersten Fragment ist der Wert des FO immer **auf Null** gesetzt.
    - Der FO definiert die Lage des jeweiligen Fragments als ein Vielfaches von 8 Byte (Grundeinheit der Fragmentierung).
    - Durch die zur Verfügung stehenden 13 Bits sind maximal 8.192 Fragmente pro Datagramm möglich (65.536 Byte).
    - Das FO-Feld ermöglicht dem Empfänger, mehrere Fragmente in der richitgen Reihenfolge zusammenzusetzen.

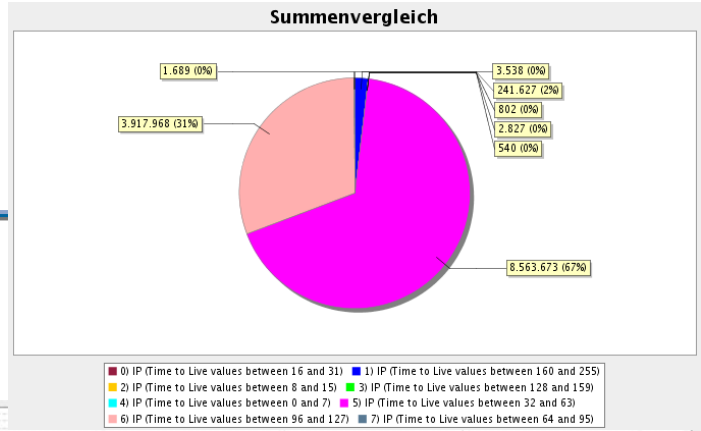
Feldelemente des IP-Headers (5/8)

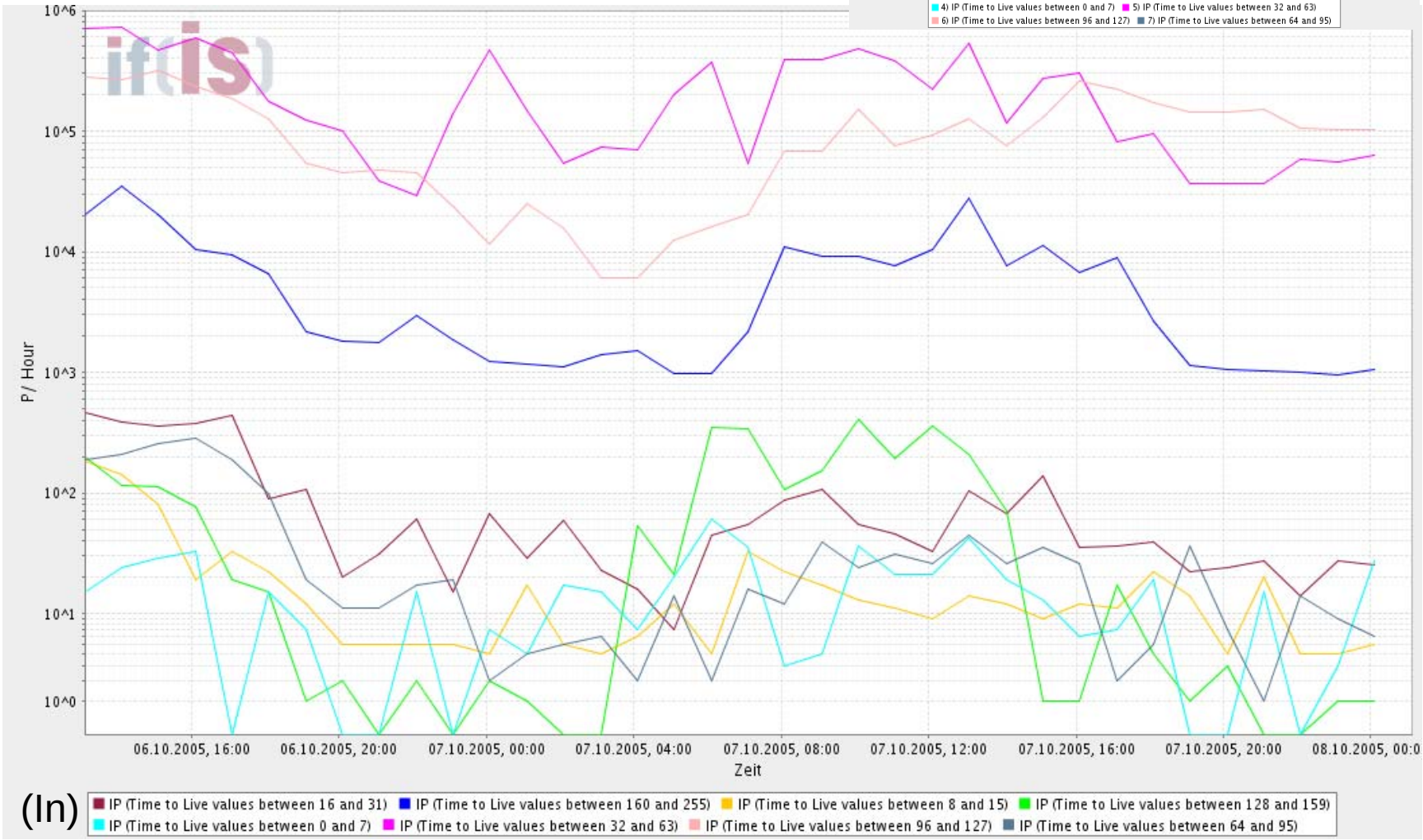


- **Time to Live (TTL)**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Der absendene Host gibt an, wie lange das Paket im Netz verweilen darf, bevor es weggeworfen werden muss.
    - Jedes Mal, wenn das Datagramm in einem Netzelement die Vermittlungsebene durchläuft, muss die IP-Einheit dieses Feldes mindestens um eins vermindern.
    - Somit ist die Lebenszeit meist gleichbedeutend mit der Anzahl der Netzknoten, die von einem Paket maximal durchlaufen werden können (=hops).
    - Wenn dieses Feld den **Wert 0** enthält, muss das Paket **weggeworfen** werden.
    - Somit wird verhindert, dass ein Paket **endlos im Netz zirkuliert!**
    - Der Absender des Paketes erhält in diesem Fall eine ICMP-Nachricht über den Vorgang.

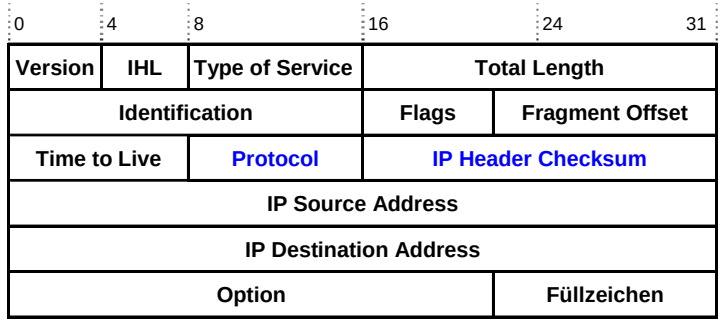
IAS: FB Informatik
→ Time to Live (TTL)

- Linux: TTL Default 64
- Windows XP: TTL Default 128
- Router (Cisco): TTL Default 254





Feldelemente des IP-Headers (6/8)

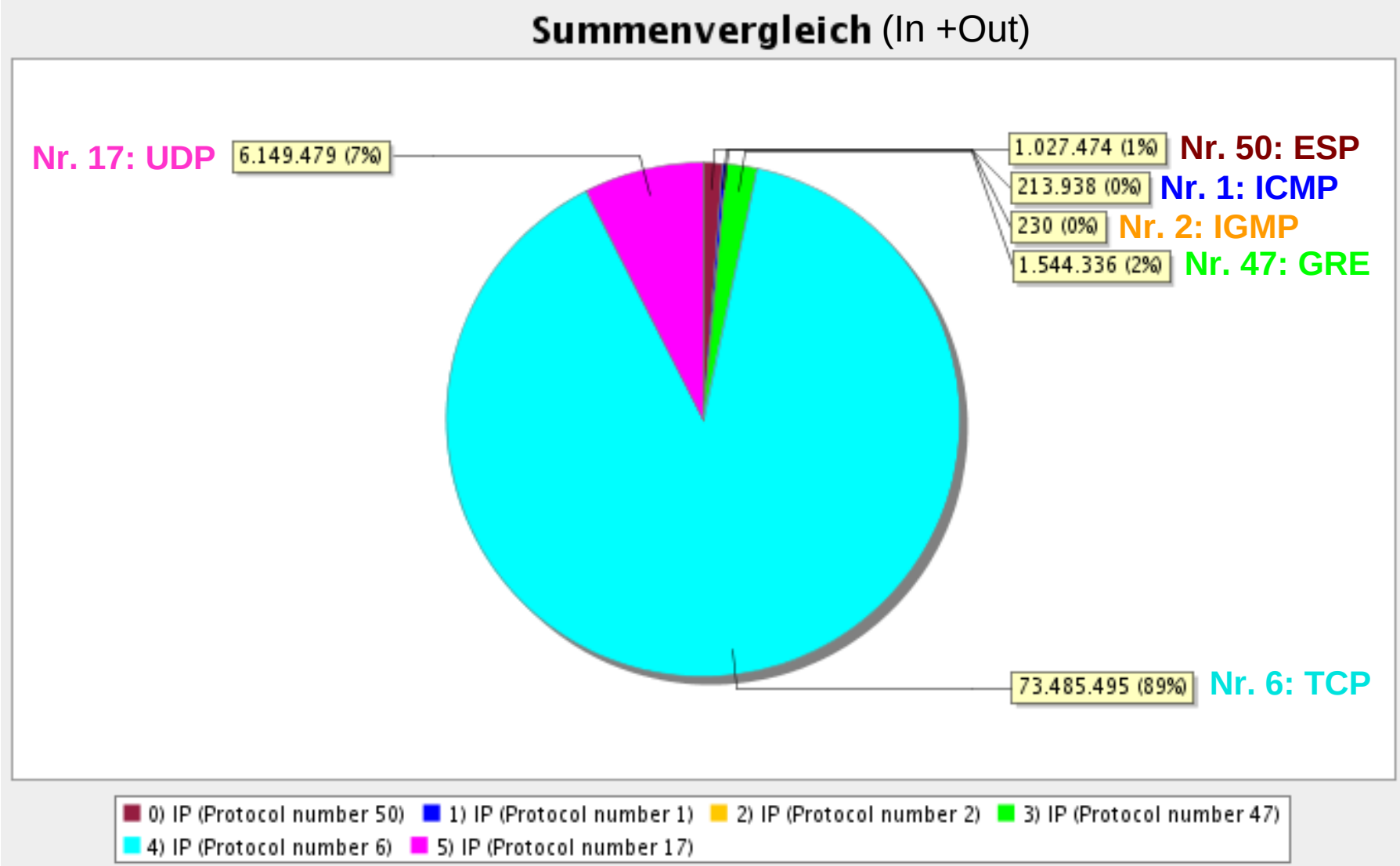


- **Protocol (PROT)**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Dieses Feld enthält die **Identifikation des Transportprotokolls**, dem das Paket zugestellt werden muss.

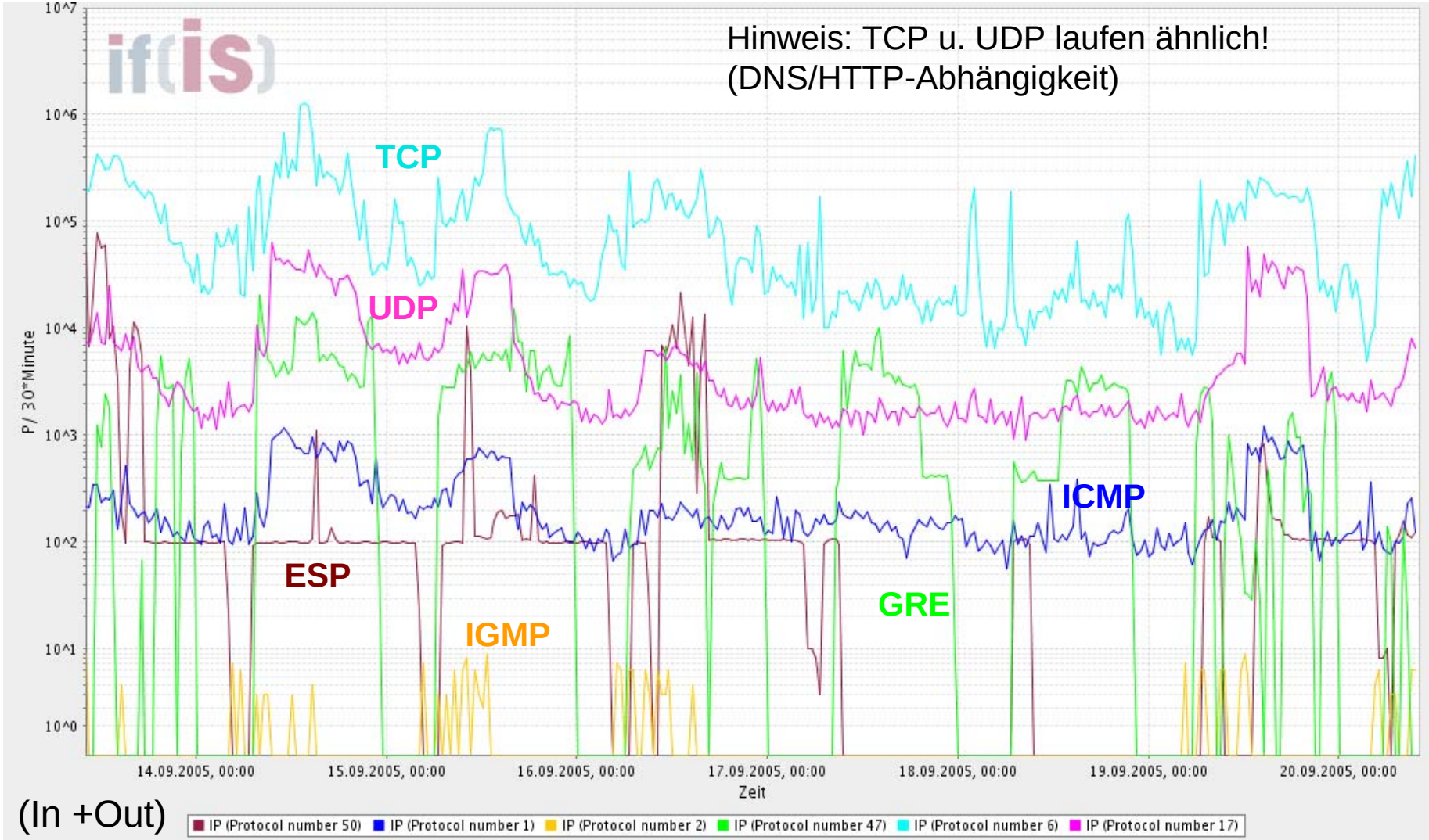
Wert (dez.)	Protokoll	
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
6	TCP	Transmission Control Protocol
8	EGP	Exterior Gateway Protocol
17	UDP	User Datagram Protocol
47	GRE	Generic Routing Encapsulation Protocol
50	ESP	Encapsulated Security Payload
89	OSPF	Open Shortest Path First

- **IP Header Checksum**
  - Feldlänge: 16 Bit
  - **Beschreibung**
    - Enthält eine **Prüfsumme**, die nur den **IP-Header** gegen Fehler sichert
    - Beim Durchgang durch einen Router verändert sich der Header (z.B. Herabsetzen von TTL um Eins) und die Prüfsumme wird neu berechnet.

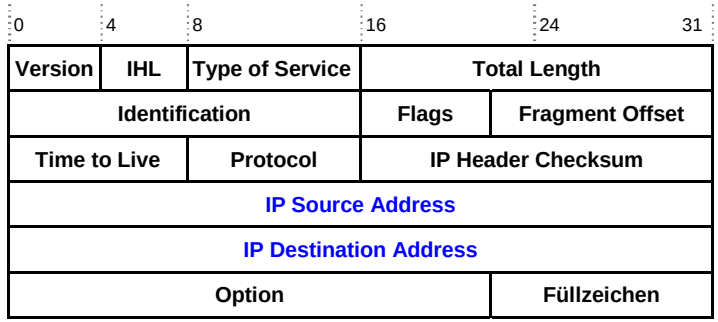
Internet-Analyse-System: FB Informatik
→ Protocol (PROT) – (1/2)



Internet-Analyse-System: FB Informatik
→ Protocol (PROT) – (2/2)



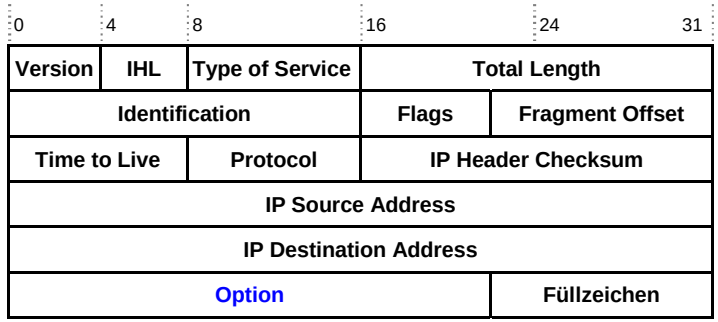
Feldelemente des IP-Headers (7/8)



- **IP Source Address (Source)**
  - Feldlänge: 32 Bit
  - **Beschreibung**
    - Enthält die Internet-Adresse des Netzknotens, der das Datagramm erzeugt hat

- **IP Destination Address (Dest)**
  - Feldlänge: 32 Bit
  - **Beschreibung**
    - Enthält die Internet-Adresse des Netzknotens, für den das Datagramm bestimmt ist

Feldelemente des IP-Headers (8/8)



- **Options (Opt)**
  - Feldlänge: variabel
  - **Beschreibung**
    - Die Dienste (Netzmanagement, Sicherheit), die IP den speziellen Anforderungen eines höheren Protokolls anpasst, werden durch die Option definiert.
    - Dieses Feld ist für die heutige Praxis zu klein (40 Byte - maximal 9 Router).

Option	Beschreibung
Security	bezeichnet, wie geheim das Datagramm ist
Strict Source Routing	bestimmt den kompletten Pfad
Loose Source Routing	gibt eine Liste von Routern aus, die nicht zu verfehlen sind
Record Route	veranlaßt jeden Router, seine IPAdresse anzuhängen
Time Stamp	veranlaßt jeden Router, seine IP-Adresse und einen Zeitstempel anzuhängen

Übertragung von IP-Paketen

- Um Daten mit einem Kommunikationspartner austauschen zu können, benötigt der Protokollstack eines Rechners außer der IP-Adresse des Partners noch einige weitere Informationen:
  - seine eigene MAC-Adresse, die entweder im Netzadapter hardwaremäßig codiert ist oder durch Jumper auf die Netzadapterkarte vergeben wird.
  - Seine eigene IP-Adresse
  - die Subnetzmaske
  - die IP-Adresse eines Netzknoten, an den er alle Pakete schickt, für die er in seiner eigenen Wegetabelle keinen Weg eingetragen hat: das Default Gateway bzw. der Default Router.
- Diese Informationen (außer der MAC-Adresse) stehen üblicherweise in Konfigurationsdateien des Betriebssystems.
- Unter UNIX können einige dieser Parameter mit dem Befehl ifconfig kontrolliert bzw. verändert werden.
- Bei den aktuellen 32-Bit Windows-Varianten (NT, XP) ist dies über die Netzwerkeigenschaften möglich.

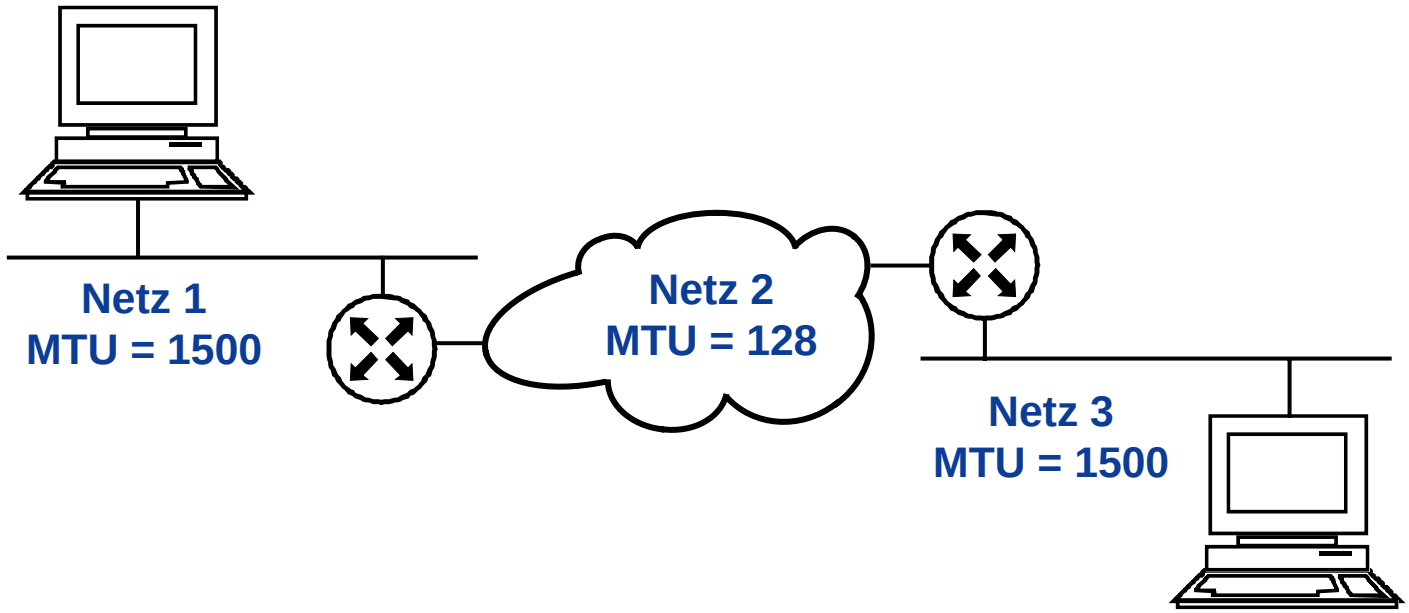
Fragmentierung (1/7)

- Um die IP-Pakete über das darunterliegende Netz zu transportieren, müssen die IP-Pakete in MAC-Rahmen (Media Access Control-Rahmen) verpackt werden.
- Im Idealfall passt ein IP-Paket genau in den Datenteil eines MAC-Rahmens.
- Unglücklicherweise hat aber jede Netztechnologie eine unterschiedliche Obergrenze für die Menge an Daten, die in einem MAC-Rahmen transportiert werden kann.
- Diese Anzahl, **Maximum Transfer Unit (MTU)** genannt, beträgt z.B. bei Ethernet 1500 Octets, bei FDDI 4470 Octets, bei X.25 jedoch nur 128 Octets pro MAC-Rahmen.
- Eine Möglichkeit zur Lösung des Problems wäre, die Größe eines IP-Paketes auf die kleinste mögliche MTU festzulegen, dies wäre jedoch reichlich ineffizient.

Fragmentierung (2/7)

- Die Kommunikationspartner können auch nicht vor der Übertragung die Größe des IP-Paketes feststellen, da IP einen verbindungslosen Transportdienst bietet und jedes Paket einen anderen Weg durch das Netz nehmen kann.
- Es muss also innerhalb von IP eine Möglichkeit geben, die Größe der Datagramme flexibel der MTU des jeweiligen Netzes anzupassen.
- Diese Möglichkeit ist die **Fragmentierung**.
- Der absendende Knoten generiert Pakete, die auf MTU des Netzes, an das er angeschlossen ist, optimiert sind.
- Soll dieses Paket dann in ein Netz übertragen werden, das eine kleinere MTU hat, wird das Paket in mehrere Teile, sogenannte **Fragmente,** zerteilt.
- Jedes dieser Fragmente wird in einen Rahmen des Netzes verpackt und weitertransportiert.
- Beim Empfänger müssen dann diese Fragmente wieder zusammengesetzt werden.
- *Weitere Anwendung: Damit bei VoIP keine großen Pakete stören!*

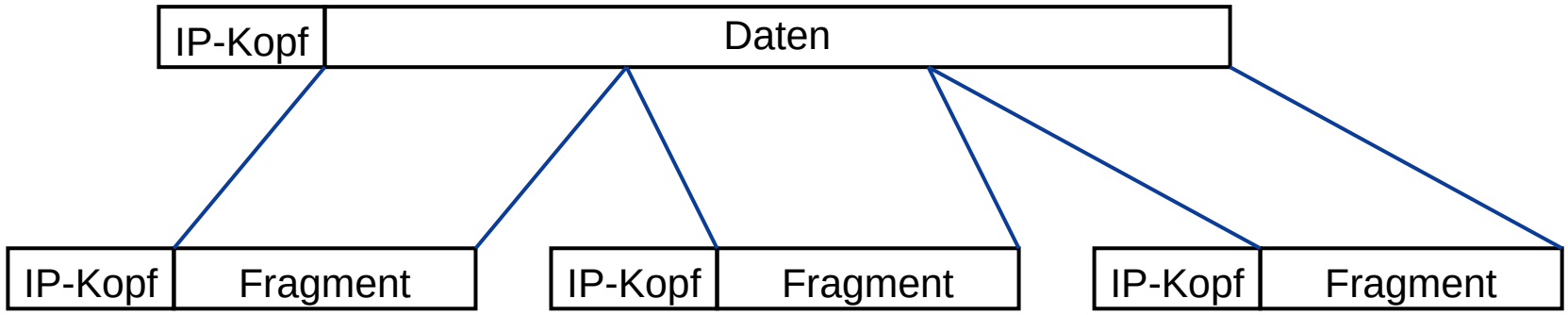
Fragmentierung (3/7)
→ Beispiel (1/3)



- Rechner 1 und Rechner 2 sind an ein **Ethernet** angeschlossen und können Pakete mit einer Maximalgröße von **1500 Octets** erzeugen.
- Der Pfad zwischen den beiden Rechnern enthält ein Netz mit einer MTU von **128 Octets** (z.B. ein **X.25-Netz**).

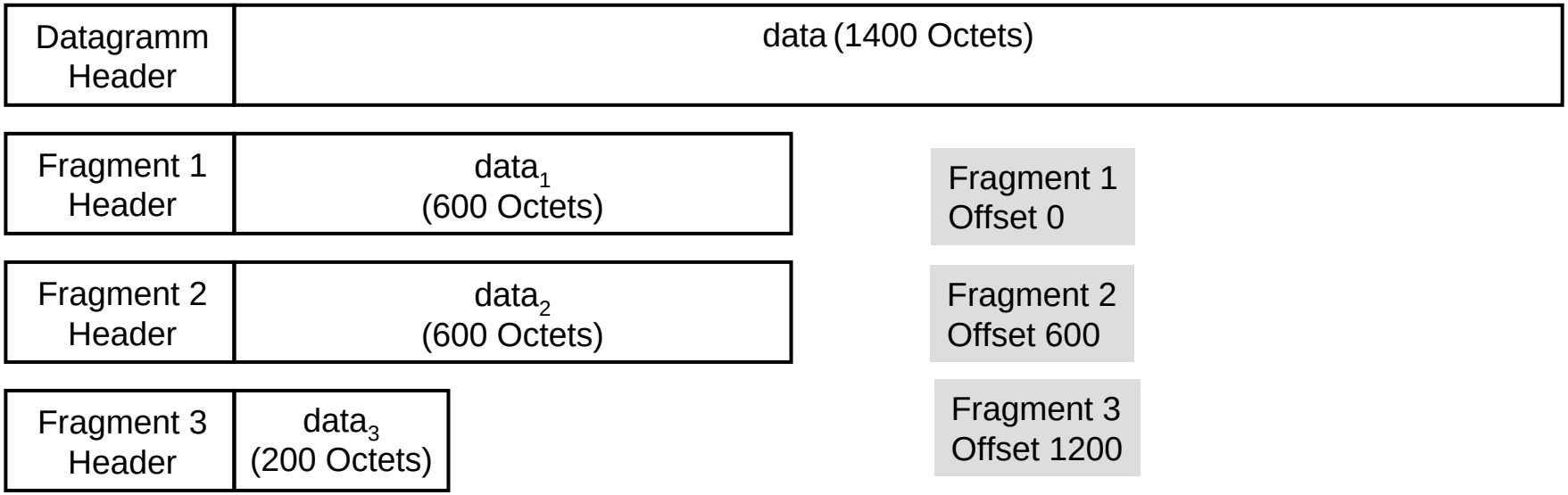
Fragmentierung (4/7)
→ Beispiel (2/3)

- Wenn Rechner 1 ein Paket an Rechner 2 schickt, das größer als 128 Octets ist, so muss der Router 1 das Paket in Fragmente mit einer Größe von 128 Octets zerlegen.



- Jedes Fragment einer Nachricht erhält einen vollständigen IP-Protokollkopf sowie das **Identifikationsfeld** der Ausgangs-Nachricht, mit deren Hilfe alle Fragmente einer Nachricht wiedererkannt werden.
- Die Lage der Daten eines Fragmentes innerhalb der Gesamtnachricht wird mit Hilfe des Fragmentabstandsfelds **(Fragment Offset)** ermittelt.

Fragmentierung (5/7)
→ Beispiel (3/3)



- Zu sehen ist eine Fragmentierung, bei der ein Paket, das ursprünglich 1400 Octets Nutzdaten enthielt, so fragmentiert wird, dass es über ein Netz mit einer MTU von 620 transportiert werden kann.
- Die Fragment Header sind dabei identisch mit dem Datagramm Header.
- Lediglich das **„more fragment“** Flag, das bei den Fragmenten 1 und 2 gesetzt ist, und das Fragment **Offset-Feld**, das die Lage im ursprünglichen Paket angibt, enthalten unterschiedliche Werte.

Fragmentierung (6/7)

- Wenn der absendende Rechner verhindern will, dass das Paket fragmentiert wird, etwa weil der Empfänger nicht zur Reassemblierung in der Lage ist, kann er die Fragmentierung durch das Setzen des **„Don´t Fragment“-**Flags im IP-Header verhindern.
- Die Nachricht wird dann verworfen, wenn sie über ein Netz nicht ohne Fragmentierung übertragen werden kann.
- **Das ursprüngliche Paket wird beim Empfänger der Nachricht zusammengesetzt, selbst wenn die Fragmente weitere Netze mit größerer MTU durchlaufen.**
- Dies ist schon deswegen notwendig, weil nicht alle Fragmente den gleichen Weg zum Empfänger nehmen müssen.
- Sobald das erste Fragment einer Nachricht beim Empfänger eintrifft, startet der Empfänger einen Timer, den **reassambly timer**.
- Läuft dieser Timer ab, bevor alle restlichen Fragmente eingetroffen sind, wird die noch unvollständige Nachricht verworfen.

Fragmentierung (7/7)

- Auf diese Weise wird verhindert, dass unvollständige Nachrichten für immer unnötigen Pufferspeicher belegen, wenn ein Fragment verloren geht.
- Bei der Reassemblierung werden die Felder ***Identification***, ***Fragment Offset*** und das ***MF-Flag*** des IP-Headers benötigt.
- Durch das Feld **Identification** kann der Empfänger eindeutig feststellen, zu **welchem Paket das Fragment gehört**.
- Der Wert des Fragment **Offset** Feldes gibt die Lage des Fragmentes **innerhalb des gesamten Pakets** an und
- mit Hilfe des **MF-Flags** kann festgestellt werden, ob das aktuelle Fragment **das letzte Fragment des Paketes** ist.

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- **ARP - Address Resolution Protocol**
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- IPv6
- Zusammenfassung

Address Resolution Protocol (ARP)

RFC 826

Reverse Address Resolution Protocol (RARP)

RFC 903

Address Resolution Protocol (ARP) und Reverse Address Resolution Protocol (RARP)

- Bisher haben wir uns mit der Feststellung zufriedengegeben, dass mit Hilfe der IP-Adresse der Zielrechner adressiert wird.
- Um einen Rechner tatsächlich zu adressieren, wird jedoch nicht die IP-Adresse benutzt, sondern die Hardware-Adresse (MAC-Adresse) des Rechners.
- Es muss also möglich sein, die IP-Adresse in eine MAC-Adresse umzusetzen und damit das Paket zum Partner zu übertragen.
- Grundsätzlich gibt es drei Möglichkeiten:
  - **Statische Umsetzung**
    → Die Tabelle muss von Hand gepflegt werden.
  - **Umwandlung der IP- in eine MAC-Adresse mit Hilfe einer Formel**
    → nur möglich, wenn die MAC-Adresse frei wählbar ist
    → fehleranfällig und umständlich
  - **Dynamische Umsetzung durch Abfragen im Subnetz**
    → dadurch werden Veränderungen der Ethernet-Adressen transparent
    → **Address Resolution Protocol (ARP)**

ARP und RARP

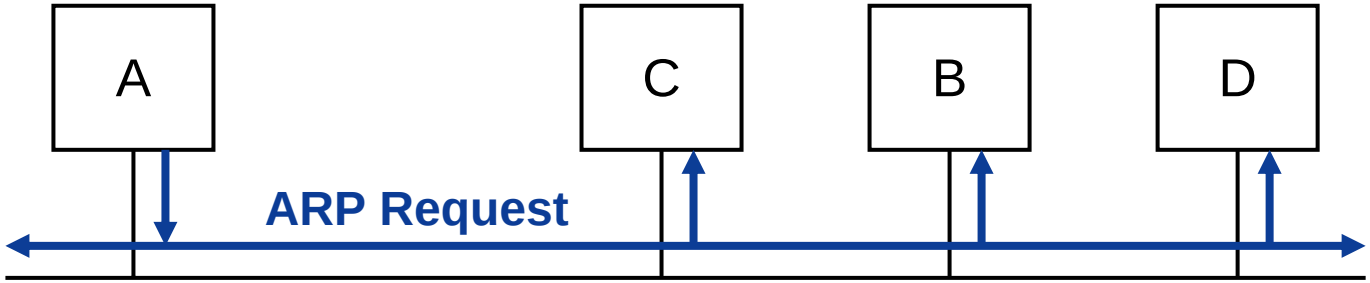
- Die Idee des ARP ist einfach:
- Wenn ein Rechner A eine Nachricht an Rechner B schicken möchte, sendet er vorher ein spezielles Paket als MAC-Broadcast aus, das nach der zur IP-Adresse gehörenden MAC-Adresse fragt (der sogenannte ***ARP-Request***).
- Der Rechner, der die gesuchte IP-Adresse hat (also Rechner B), beantwortet das Paket mit einem ***ARP-Reply***.
- Nun kann der Rechner A die Nachricht an Rechner B senden.
- Damit nicht vor jeder Übertragung ein ARP-Request ausgesendet werden muss, wird die Zuordnung der IP-Adresse zur MAC-Adressse in einem Cache, dem ***ARP-Cache***, zwischengespeichert.
- Wenn eine Nachricht verschickt werden soll, wird zunächst überprüft, ob der ARP-Cache einen Eintrag für die benötigte IP-Adresse enthält.

ARP und RARP

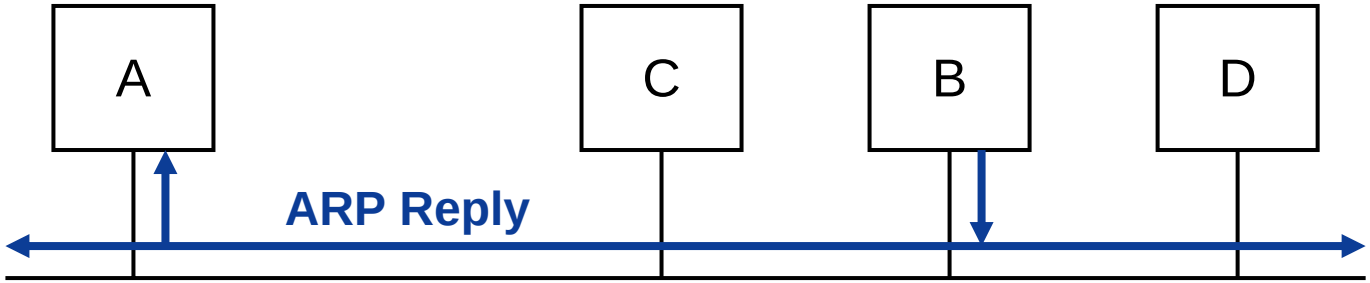
- Damit diese Methode möglichst effizient funktioniert, werden einige Tricks verwendet.
- So kann z.B. Rechner B, wenn er einen ARP-Request von Rechner A empfängt, die Zuordnung von Rechner A´s IP-Adresse zu dessen MAC-Adresse, die er dem ARP-Request entnehmen kann, in seinem ARP-Cache abspeichern.
- Ebenso können alle anderen Rechner im Netz verfahren, die diesen Broadcast empfangen.
- Damit wäre bei einer Kommunikation mit A kein neuerlicher Broadcast mehr nötig.
- Da der Eintrag im ARP-Cache nicht ewig bestehen bleiben kann (z.B. Austausch eines Netzadapters), werden innerhalb einer festgelegten Zeit (z.B. 20 Minuten) diese aus dem Cache gelöscht.

ARP und RARP

Request

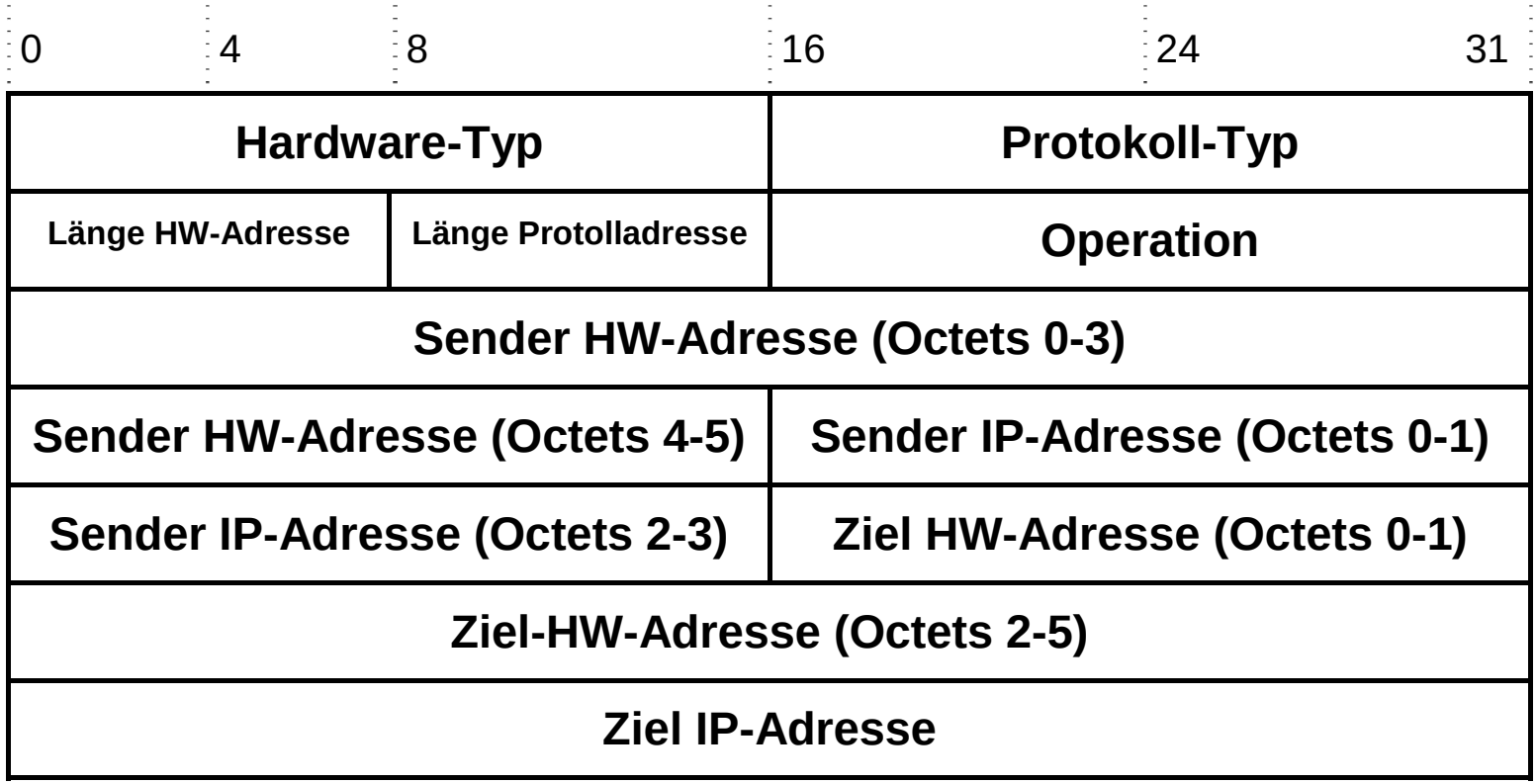


Reply

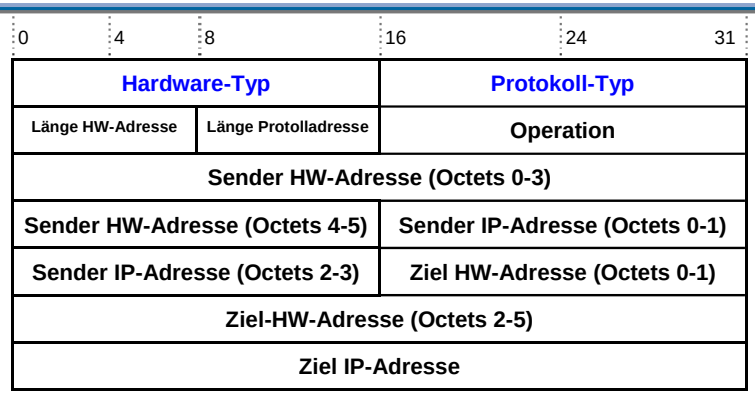


- ARP-Pakete werden als Daten in einen MAC-Rahmen übertragen.
- Um den Rahmen zu kennzeichnen, wird das Type-Feld im MAC-Header auf einen entsprechenden Wert gesetzt (bei Ethernet = 0806h).

Aufbau des ARP-Headers



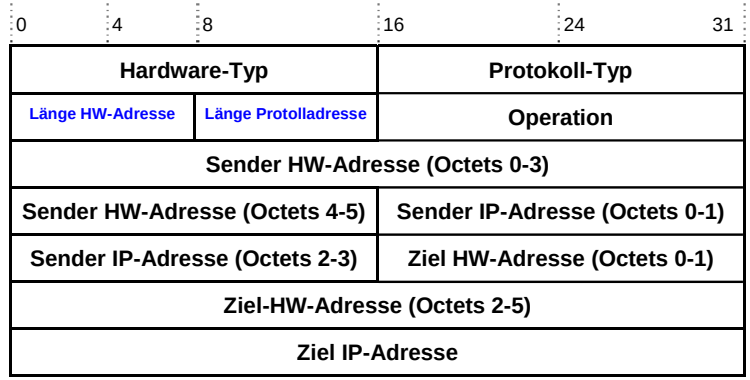
Feldelemente des ARP-Headers (1/5)



- **Hardware-Typ (HRD)**
  - Feldlänge: 16 Bit
  - **Beschreibung**
    - Definiert das verwendete Übertragungsmedium, Kommunikationsgeschwindigkeit und Datenstruktur, z.B.
      1 = Ethernet
      6 = IEEE 802.x Netzwerke

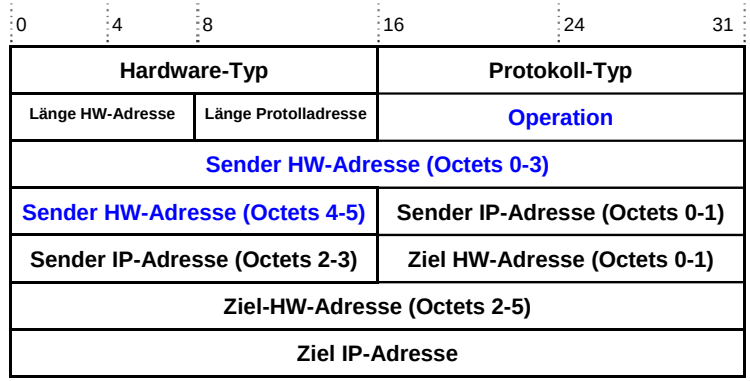
- **Protocol-Typ (PRO)**
  - Feldlänge: 16 Bit
  - **Beschreibung**
    - Typ-Feld-Nummer zur Unterscheidung des verwendeten Netzprotokolls, 0800h für IP.

Feldelemente des ARP-Headers (2/5)



- **Länge HW-Adresse (HRD)**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Anzahl der Hardware-Adress-Octets.
    - Bei Ethernet beträgt der Wert immer „06“, was einer Länge von 48 Bit entspricht.
- **Länge Protokoll**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Anzahl der Adress-Octets, die durch höhere Protokolle (z.B. IP) definiert werden.
    - Bei TCP/IP-Protokollen beträgt der Wert immer „04“.

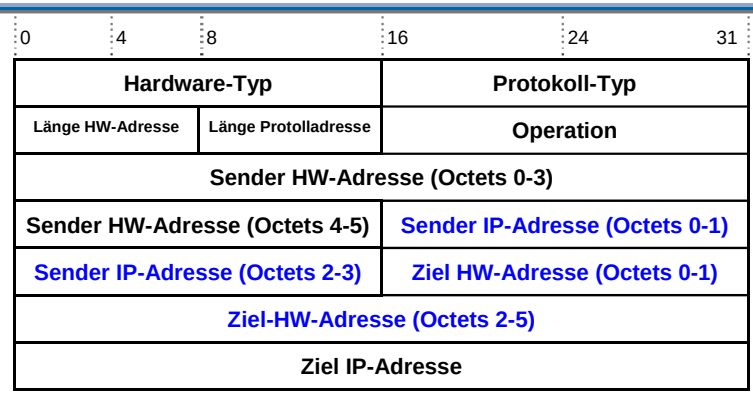
Feldelemente des ARP-Headers (3/5)



- **Operation (OP)**
  - Feldlänge: 16 Bit
  - **Beschreibung**
    - Definiert die Art des ARP-Paketes:
      1 = ARP-Anfrage (Request)
      2 = ARP-Antwort (Reply)
      3 = RARP-Anfrage (Request)
      4 = RARP-Antwort (Reply)

- **Sender HW-Adresse (Source)**
  - Feldlänge: 48 Bit
  - **Beschreibung**
    - Die Sender HW-Adresse gibt die Hardware-Adresse des Senders an.

Feldelemente des ARP-Headers (4/5)

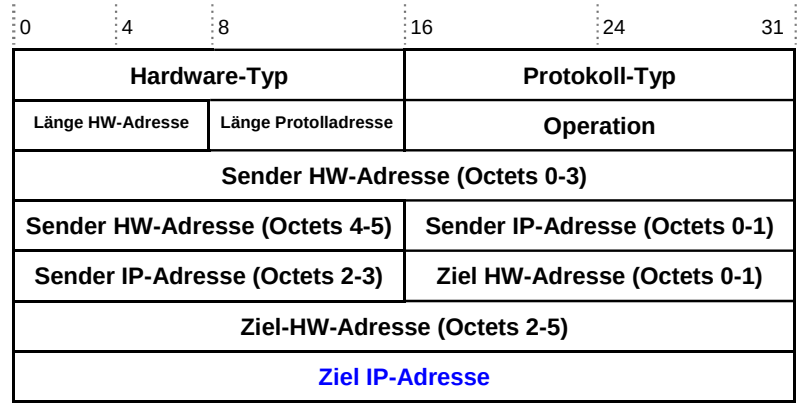


- **Sender IP-Adresse (So-Adr)**
  - Feldlänge: 32 Bit
  - **Beschreibung**
    - IP-Adresse des Senders

- **Ziel HW-Adresse (De-Adr)**
  - Feldlänge: 48 Bit
  - **Beschreibung**
    - Diese Feld enthält bei einem ARP-Reply die Hardware Adresse des Empfängers.
    - Bei einem ARP-Request ist die Hardware-Adresse des Zielknotens nicht bekannt, deshalb wird bei ARP-Anfragen dieses Feld mit der Broadcast-Adresse des Netzes belegt.
    - Diese Adresse wird bei der Anwort durch die richtige Hardwareadresse ersetzt.

Feldelemente des ARP-Headers (5/5)

- **Ziel IP-Adresse (Pro-Des)**
  - Feldlänge: 32 Bit
  - **Beschreibung**
    - IP-Adresse des Empfängers.



Reverse Address Resolution (RARP)

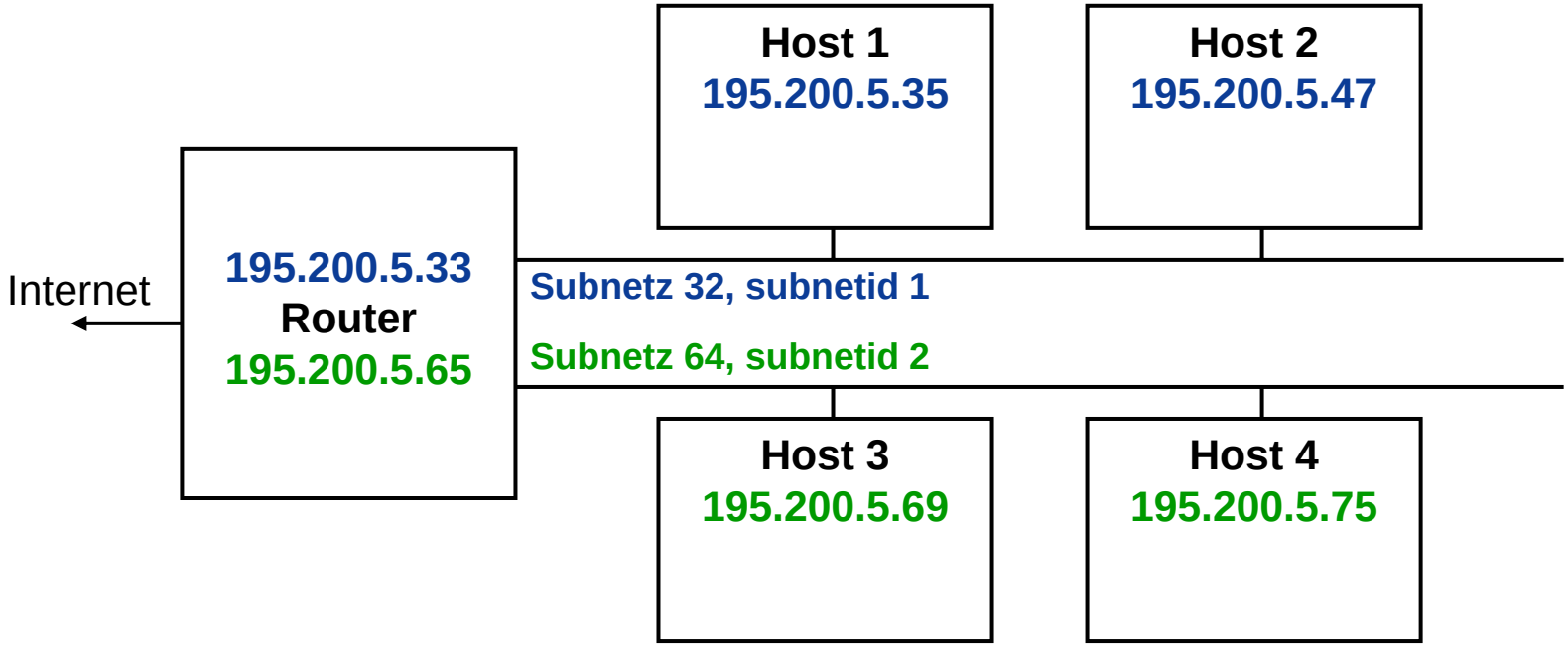
- Der Aufbau eines Paktes des Reverse Adress Resolution Protocol (RARP) ist identisch zum Aufbau eines ARP-Paketes.
- Im Gegensatz zu ARP ermittelt RARP zu einer bekannten MAC-Adresse eine IP-Adresse.
- Dies ist notwendig, wenn z.B. ein Rechner ohne eigene Festplatte gebootet wird.
- Da die Konfigurationsdatei zusammen mit dem Betriebssystem des Rechners auf dem Server abgelegt sind, muss der Rechner eine Möglichkeit haben, seine eigene IP-Adresse zu erfahren, um sein Betriebssystem vom Server laden zu können.
- Dazu versendet er einen RARP-Request als MAC-Broadcast, das im Typ/Längenfeld den Wert 0835h enthält.
- Der RARP-Server empfängt dieses Paket, sucht in seiner Tabelle nach der zur MAC-Adresse gehörenden IP-Adresse und sendet den RARP-Reply direkt an den Rechner.
- Dieser kann nun über TCP/IP sein Betriebssystem laden.

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- **Beispiele für die Übertragung eines IP-Paketes**
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- IPv6
- Zusammenfassung

Beispiele für die Übertragung eines IP-Paketes

- Den Beispielen liegt die folgende Netzstruktur zugrunde



- Es wird davon ausgegangen, dass noch kein Netzverkehr stattgefunden hat und alle Caches leer sind.
- Es handelt sich hierbei um ein Klasse C-Netz, das in 8 Subnetze unterteilt wurde.
- Die dazu benötigte Netzmaske lautet 255.255.255.224.

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **gleichen** Subnetz (1/3)

- Host 1 möchte ein Paket an Host 2 schicken, der sich **im gleichen Subnetz befindet**.
- Um dies herauszufinden, muss jedoch zuerst die Netzadresse des Zielknotens mit der eigenen Netzadresse verglichen werden.
- **Für Host 1:**

IP-Adresse 1:	11000011	11001000	00000101	00100011
Netzmaske:	11111111	11111111	11111111	11100000
Netzadresse 1:	**11000011**	**11001000**	**00000101**	**00100000**

- **Für Host 2:**

IP-Adresse 2:	11000011	11001000	00000101	00101111
Netzmaske:	11111111	11111111	11111111	11100000
Netzadresse 2:	**11000011**	**11001000**	**00000101**	**00100000**

- Aus der Gleichheit der beiden Netzadressen folgt, dass sich der Zielknoten im gleichen Subnetz befindet und das Paket direkt an diesen Knoten geschickt werden kann.

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **gleichen** Subnetz (2/3)

- Da die MAC-Adresse des Zielknotens sich nicht im ARP-Cache befindet, sendet Host 1 eine ARP-Request aus, um die MAC-Adresse von Host 2 herauszufinden.

- MAC-DA = FFFFFFFFh (Broadcast)
  MAC-SA = MAC-Adresse (Host 1)
  Type = 0806h (ARP)
  IP-DA = 195.200.5.47
  IP-SA = 195.200.5.35
  Operation = 1 (ARP-Request)

- Da es sich bei diesem Paket um ein MAC-Broadcast handelt, empfangen alle Knoten im Subnetz dieses Paket.

- Der Router leitet diesen Broadcast jedoch ***nicht*** weiter, so dass Host 3 und 4 dieses Paket nicht empfangen.

- Nur der Host, mit der richtigen IP-Adresse, sendet ein ARP-Reply.

- MAC-DA = MAC-Adresse (Host 1)
  MAC-SA = MAC-Adresse (Host 2)
  Type = 0806h (ARP)
  IP-DA = 195.200.5.35
  IP-SA = 195.200.5.47
  Operation = 2 (ARP-Reply)

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **gleichen** Subnetz (3/3)

- Host 1 kennt nun die MAC-Adresse von Host 2 und kann das Datenpaket an Host 2 senden.
- Beide Hosts tragen die Zuordnung IP-Adresse <> MAC-Adresse in ihren Cach ein und können in Zukunft, solange der Eintrag gültig ist, ohne einen vorhergehenden ARP-Request miteinander verkehren.

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **anderen** Subnetz (1/4)

- Host 1 möchte ein Paket an Host 3 schicken, der sich in einem anderen Subnetz befindet.
- Um dieses herauszufinden, muss jedoch zuerst die Netzadresse des Zielknotens mit der eigenen Netzadresse verglichen werden.
- **Für Host 1:**

IP-Adresse 1:	11000011	11001000	00000101	00100011
Netzmaske:	11111111	11111111	11111111	11100000
Netzadresse 1:	**11000011**	**11001000**	**00000101**	**00100000**

- **Für Host 3:**

IP-Adresse 3:	11000011	11001000	00000101	01001111
Netzmaske:	11111111	11111111	11111111	11100000
Netzadresse 3:	**11000011**	**11001000**	**00000101**	**01000000**

- Nun ist die Netzadresse des Zielknotens nicht mehr mit der des Quellknotens identisch.
- Das Paket kann also nicht direkt an den Zielknoten gesendet werden, da sich dieser nicht im eigenen Subnetz befindet!

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **anderen** Subnetz (2/4)

- In diesem Fall muss das Paket über einen weiteren Rechner, den **Default Router**, an Knoten 3 geschickt werden.
- Der Default Router ist derjenige Rechner, an den ein Host alle Pakete schickt, die nicht an einen Host im eigenen Subnetz adressiert sind und für die er keinen Eintrag in seiner lokalen Routing-Tabelle hat.
- Da Host 1 die IP-Adresse des Default Routers aus seinen lokalen Konfigurationsdaten kennt, aber noch nicht dessen MAC-Adresse, muss er zunächst die MAC-Adresse des Routers durch einen ARP-Request herausfinden.
- Ist dies geschehen, schickt Host 1 ein IP-Paket an den Router, das als MAC-DA die Adresse des Routers enthält, als IP-DA jedoch die IP-Adresse von Host 3.
- **MAC-DA = MAC-Adresse Router (Subnetz 32)**
  MAC-SA = MAC-Adresse Host 1
  Type = 0800h (IP)
  IP-DA = 195.200.5.69 (Host 3)
  IP-SA = 195.200.5.35 (Host 1)

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **anderen** Subnetz (3/4)

- Der Router empfängt das IP-Paket und entscheidet anhand der IP-Adresse, an welchen seiner Anschlüsse er das Paket weiterleitet.
- Durch den Vergleich der IP-Adresse mit den Einträgen in seiner Routing-Tabelle erkennt er, dass an einem anderen Anschluss ein „passendes“ Netz angeschlossen ist.
- Er leitet das Paket in das Subnetz weiter, muss sich aber zunächst über ARP die MAC-Adresse des Hosts mit der im IP-Paket stehenden IP-Adresse beschaffen.
- Erst dann kann er das Paket an Host 3 senden, wobei als MAC-SA seine MAC-Adresse im entsprechenden Rahmenfeld steht:
- MAC-DA = MAC-Adresse Host 3
  MAC-SA = MAC-Adresse Router (Subnetz 64)
  Type = 0800h (IP)
  IP-DA = 195.200.5.69 (Host 3)
  IP-SA = 195.200.5.35 (Host 1)

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host im **anderen** Subnetz (4/4)

- Möchte nun z.B. Host 4 ein Paket an Host 1 senden, sind keinerlei ARP-Pakete mehr nötig, da Host 4 durch Mithören des ARP-Request im Beispiel (Router -> Host 3) die MAC-Adresse des Routers kennt.
- Der Router kennt die MAC-Adresse von Host 1 bereits aus dem ersten Fallbeispiel durch Mithören des ARP-Requests von Host 1 an Host 2.

Beispiele für die Übertragung eines IP-Paketes
→ Übertragung an einen Host **außerhalb** des eigenen Netzes

- Als letzten Fall betrachten wir noch die Situation, dass Knoten 1 ein Paket an einen Knoten schickt, der außerhalb des eigenen Netzes liegt.
- Host 1 vergleicht zunächst die Netzadresse des Zielknotens mit seiner Netzadresse und stellt fest, dass der Host außerhalb seines Subnetzes liegt.
- Daher sendet er das Paket an den eigenen (Default)-Router.
- Dieser stellt durch Überprüfung seiner Routing-Tabelle fest, dass sich der Host nicht in einem der an seinen Anschlüssen befindlichen Netze befindet.
- Da er für diesen Host auch keinen speziellen Eintrag in seiner Routing-Tabelle findet, schickt er das Paket an seinen Default-Router im Internet.
- Die weitere Vermittlung erfolgt durch die Router im Internet.

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- **DHCP – Dynamic Host Configuration Protocol**
- ICMP - Internet Control Message Protocol
- IPv6
- Zusammenfassung

Dynamic Host Configuration Protocol (DHCP)
→ Übersicht

- DHCP ist ein **Anwendungsdienst** und ermöglicht mit Hilfe eines entsprechenden Servers die **dynamische Zuweisung einer IP-Adresse und weiterer Konfigurationsparameter** wie Netzmaske, Gateway, DNS-Server, usw. an Rechnersysteme in einem Netzwerk (z.B. Internet oder LAN).
- Netzwerken bietet DHCP den Vorteil, dass bei Topologieänderungen nicht mehr alle betroffenen Workstations per Hand umkonfiguriert werden müssen, sondern die entsprechenden Vorgaben vom Administrator nur einmal in der Konfigurationsdatei des DHCP-Servers gemacht werden müssen.
- Auch für Rechner mit häufig wechselndem Standort (z.B. Notebooks) entfällt die fehleranfällige Konfiguration - der Rechner wird einfach ans Netzwerk gesteckt und erfragt alle relevanten Parameter vom DHCP-Server (Plug'n'Play).

Aufgaben von DHCP

- Zentrale Verwaltung von IP Adressen
- Zuweisung von IP Adressen auf Leihbasis (lease)
  - Dynamische Zuweisung
    - „freie“ IP Adresse aus einem IP Adressbereich (Scope)
  - Statische Zuweisung
    - „feste“ IP Adresse wird MAC Adresse zugewiesen
- Zentrale Verwaltung weiterer Netzwerk Parameter
    - Default Gateway
    - DNS Server
    - Usw.

Vorteile eines DHCP-Servers

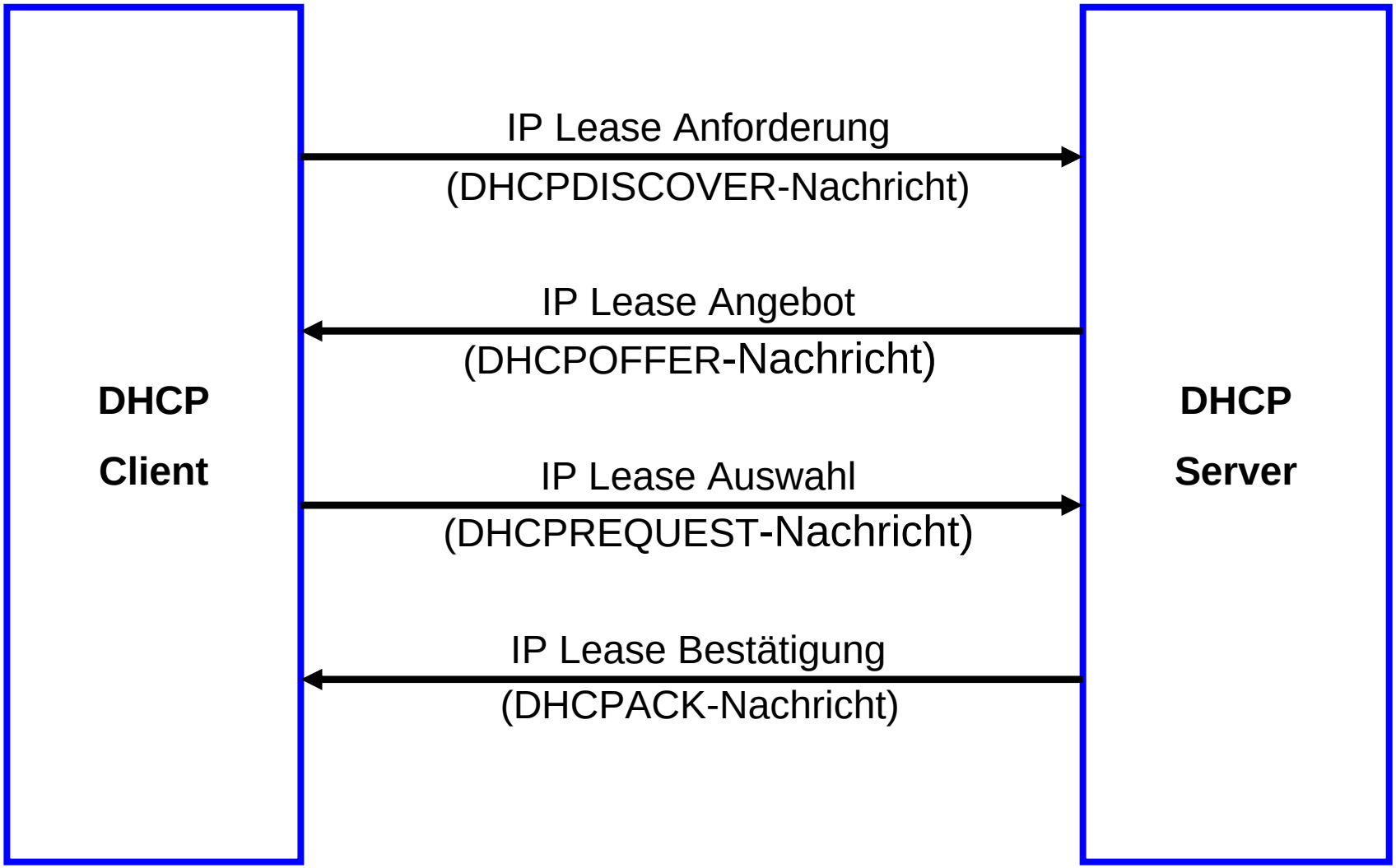
- Automatische IP Konfiguration am Client
  - Keine manuelle IP Konfiguration mehr nötig
    - Zeitersparnis
    - Keine Fehlerquellen durch manuelle Eingabe
- DHCP Client überall im Netz anschließbar
  - Keine Anfrage an Netzwerk Manager nötig
- Flexible und schnelle Konfigurationsänderung
  - Durch zentrale Verwaltung von Netzwerk Parametern
    - Default Gateway, DNS Server, etc.

Nachteile eines DHCP-Servers

- Zusätzliche Belastung des Netzes
  - durch Client <-> Server Requests

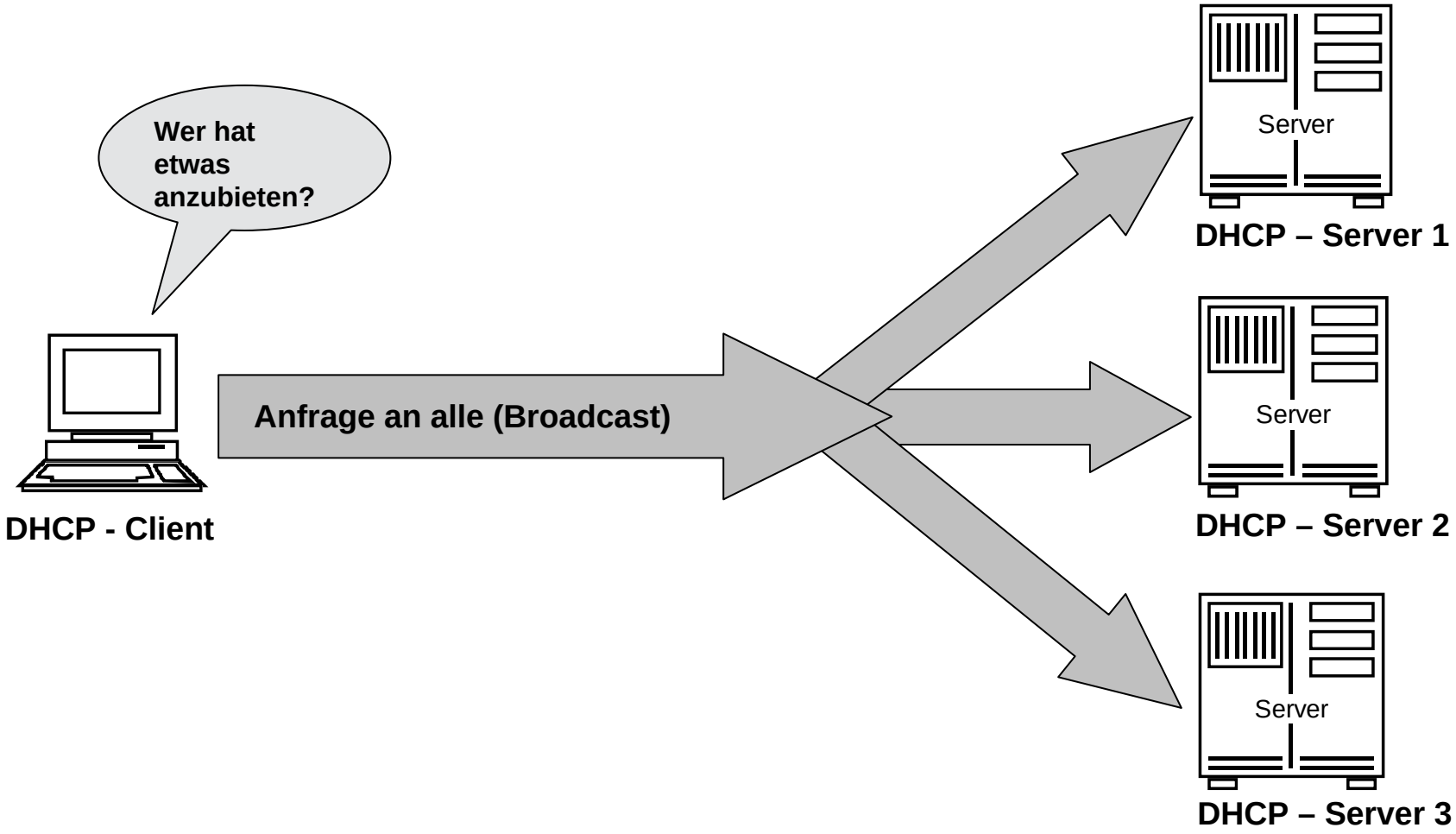
- zusätzlicher Aufwand für Administration/Planung
  - DHCP im WAN :
    - DHCP Requests meist Broadcasts --> nicht routebar
  - Ausfallsicherheit des DHCP Servers

Wie bekommt DHCP Client seine IP Adresse?



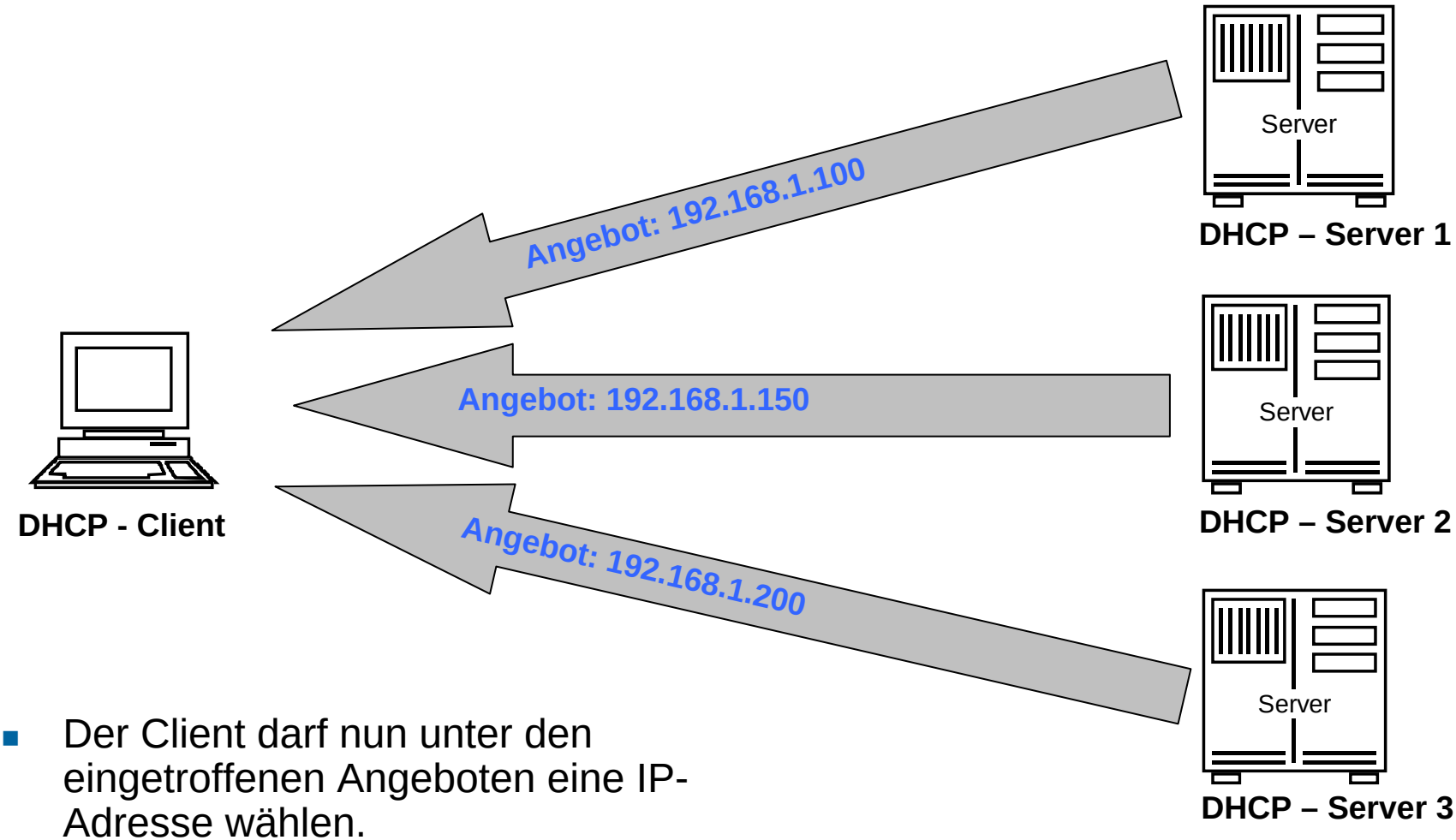
Ablauf der DHCP-Kommunikation
→ DHCPDISCOVER

- Wenn ein Client erstmalig eine IP-Adresse benötigt, schickt er eine DHCPDISCOVER-Nachricht als Netzwerk-Broadcast an die verfügbaren DHCP-Server



Ablauf der DHCP-Kommunikation
→ DHCPOFFER

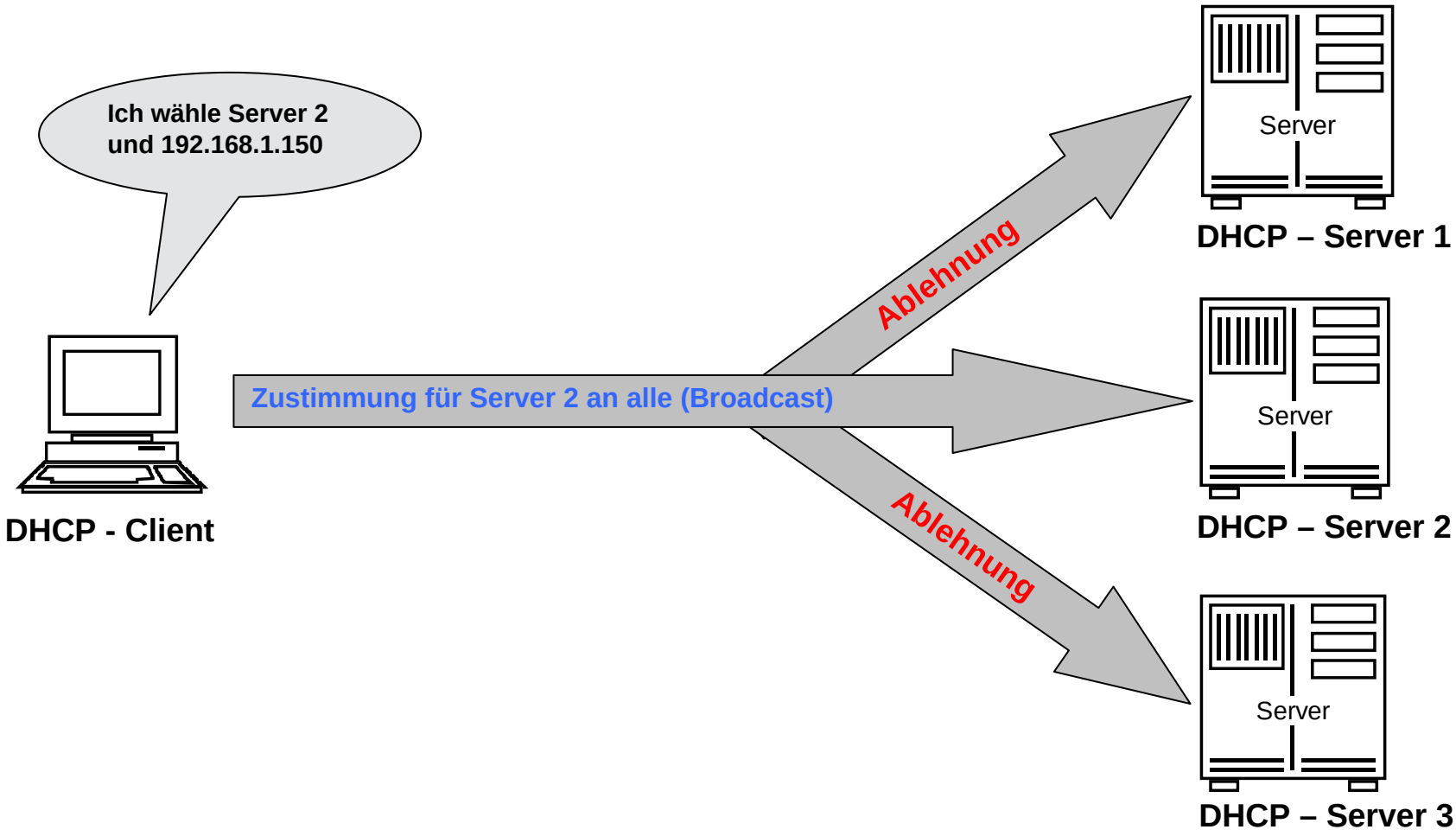
- Die DHCP-Server antworten mit der DHCPOFFER-Nachricht und machen einen Vorschlag für eine IP-Adresse.



- Der Client darf nun unter den eingetroffenen Angeboten eine IP-Adresse wählen.

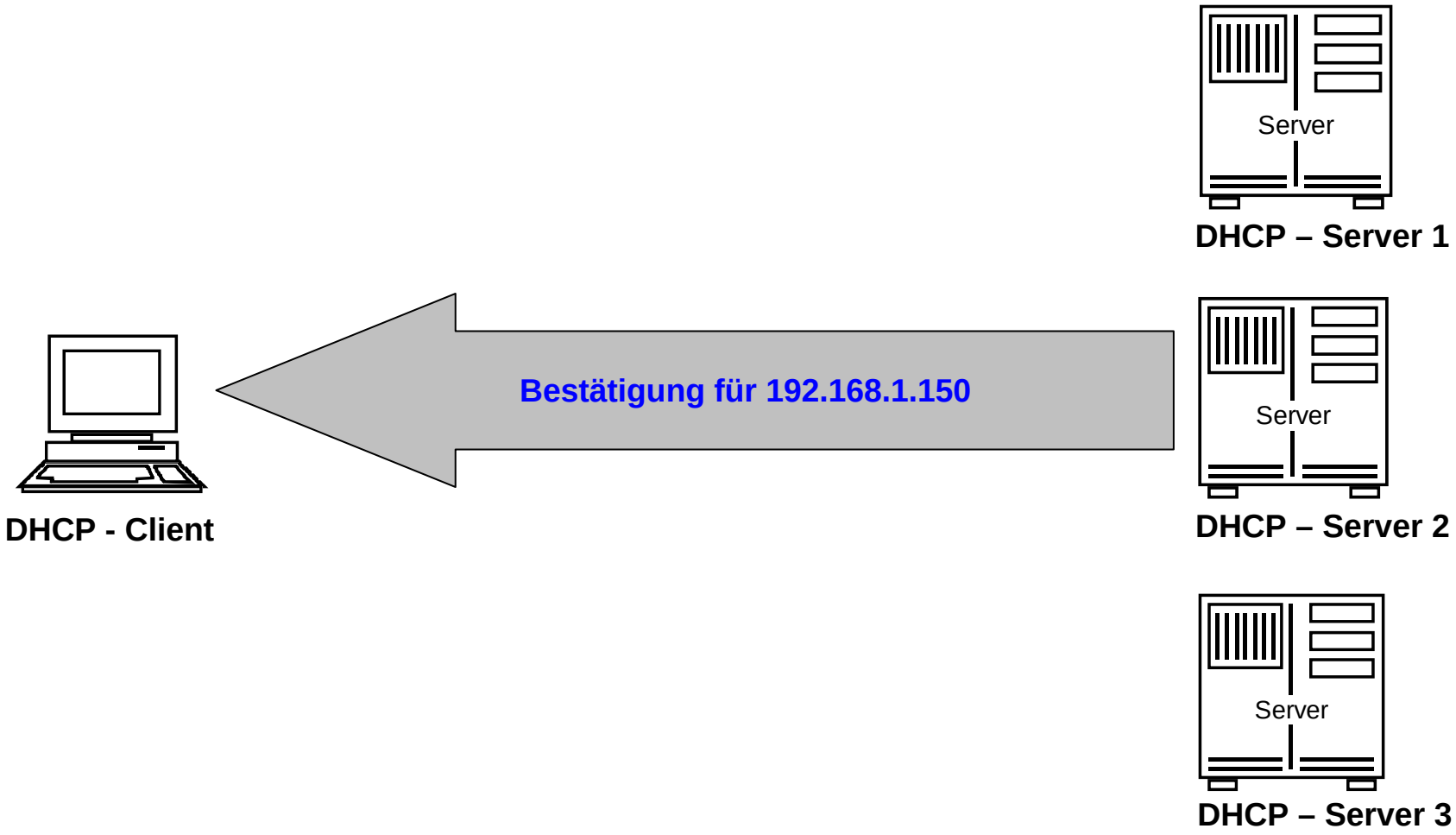
Ablauf der DHCP-Kommunikation
→ DHCPREQUEST

- Wenn sich der Client für eine entschieden hat (z.B. wegen längster Lease-Zeit), sendet er per Broadcast die Nachricht-DHCPREQUEST.



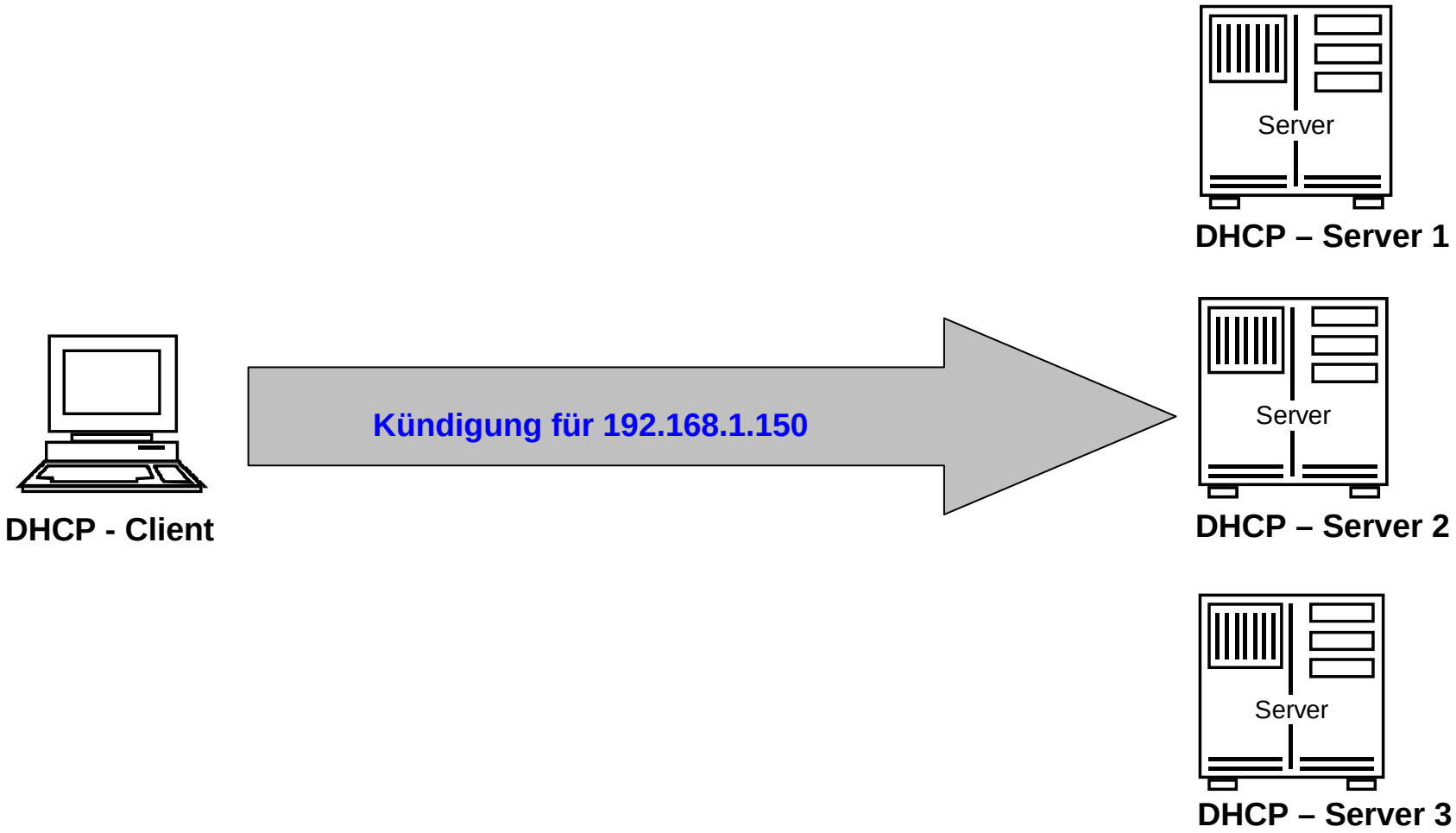
Ablauf der DHCP-Kommunikation
→ DHCPACK

- Der Server übermittelt dann mit der DHCPACK-Nachricht die IP-Adresse mit den weiteren relevanten Daten.



Ablauf der DHCP-Kommunikation
→ DHCPRELEASE

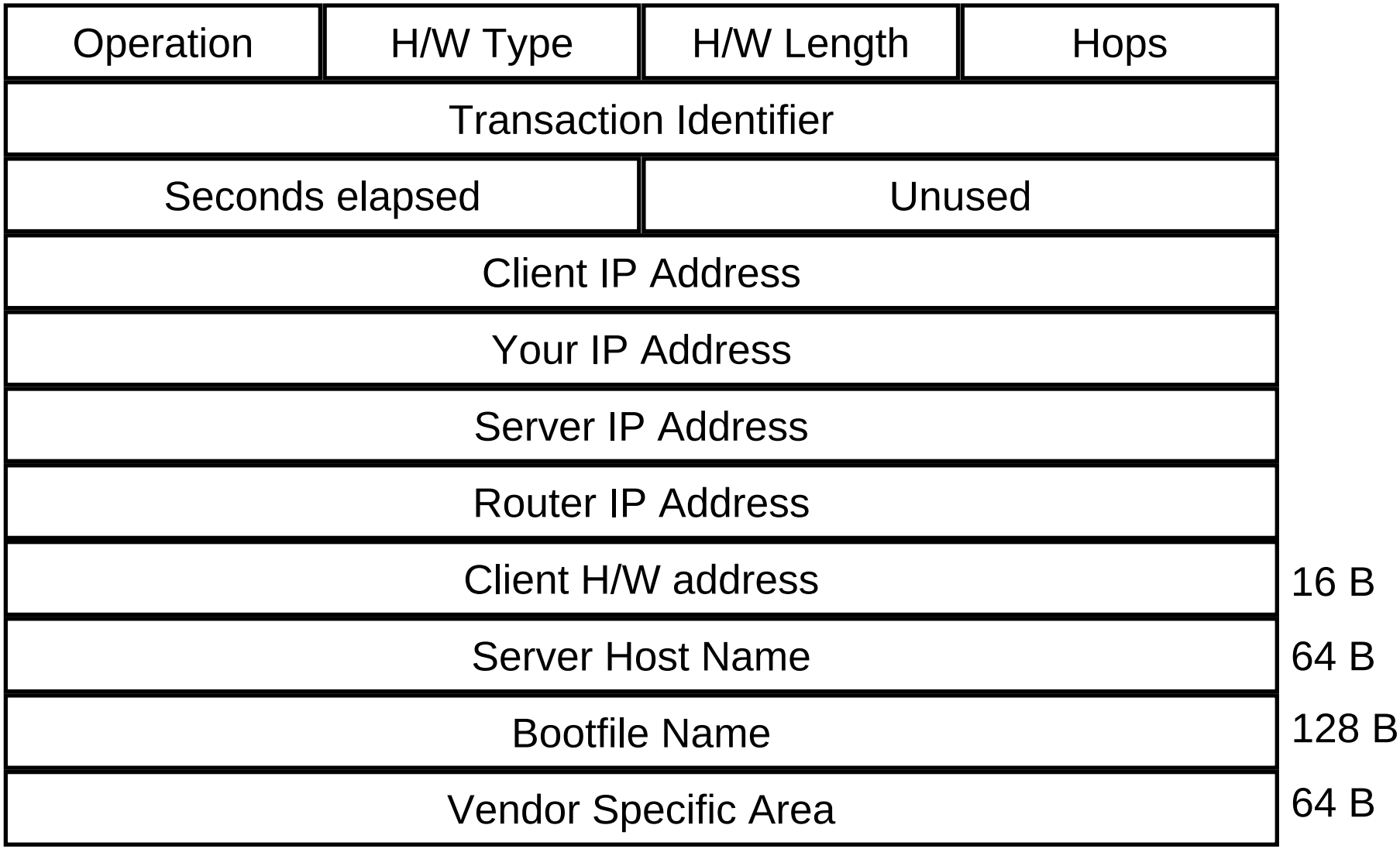
- Ein Client sendet DHCPRELEASE, wenn er eine IP-Adresse vor Ablauf der Lease-Zeit zurückgeben will.



Ablauf der DHCP-Kommunikation
→ DHCP-Refresh

- Zusammen mit der IP-Adresse erhält der Client in der DHCPACK-Nachricht die Lease-Zeit.
- Der Standard sieht vor, dass der Client nach der Hälfte der Lease-Zeit einen erneuten DHCPREQUEST sendet und so bekundet, dass weiter Interesse an der reservierten IP-Nummer besteht.
- Dieser DHCPREQUEST wird per Unicast an den im Datenpaket enthaltenen Server gesendet.
- Der Server sollte dann in der Regel ein DHCPACK mit identischen Daten wie vorher, aber mit einer neuen Lease-Zeit senden.
- Damit gilt die Adresse als verlängert.

BOOTP Message Format



DHCP-Kommunikation
→ Protokollmitschnitt (1/2) - DHCPREQUEST

```
Ethernet II, Src: 00:0d:60:79:ac:42, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: 0.0.0.0 (0.0.0.0), Dst Addr: 255.255.255.255 (
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xa7031f32
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client hardware address: 00:0d:60:79:ac:42
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP Request
    Option 61: Client identifier
        Hardware type: Ethernet
        Client hardware address: 00:0d:60:79:ac:42
    Option 50: Requested IP Address = 192.168.2.106
    Option 12: Host Name = "FH-Pohlmann"
    Option 81: FQDN
        Flags: 0x00
            0000 .... = Reserved flags: 0x00
            .... 0... = No server ddns: Some server updates
            .... .0.. = Binary encoding: ASCII encoding
            .... ..0. = Server overrides: No override
            .... ...0 = Server: Client
        A-RR result: 0
        PTR-RR result: 0
        Client name: FH-Pohlman
    . . .
    . . .
    Option 60: Vendor class identifier = "MSFT 5.0"
    Option 55: Parameter Request List
        1 = Subnet Mask
        15 = Domain Name
        3 = Router
        6 = Domain Name Server
        44 = NetBIOS over TCP/IP Name Server
        46 = NetBIOS over TCP/IP Node Type
        47 = NetBIOS over TCP/IP Scope
        31 = Perform Router Discover
        33 = Static Route
        Unknown Option Code: 249
        43 = Vendor-Specific Information
    End Option
```

DHCP-Kommunikation
→ Protokollmitschnitt (2/2) - DHCPACK

```
Ethernet II, Src: 00:30:f1:8b:4d:8c, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol, Src Addr: 192.168.2.1 (192.168.2.1), Dst Addr:
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xa7031f32
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
        0... .... .... .... = Broadcast flag: Unicast
        .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.2.106 (192.168.2.106)
    Next server IP address: 192.168.2.1 (192.168.2.1)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client hardware address: 00:0d:60:79:ac:42
    Server host name not given
    Boot file name not given
    Magic cookie: (OK)
    Option 53: DHCP Message Type = DHCP ACK
    Option 54: Server Identifier = 192.168.2.1
    Option 51: IP Address Lease Time = infinity
    Option 1: Subnet Mask = 255.255.255.0
    Option 3: Router = 192.168.2.1
    Option 6: Domain Name Server = 192.168.2.1
    End Option
    Padding
```

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- **ICMP - Internet Control Message Protocol**
- IPv6
- Zusammenfassung

ICMP - Internet Control Message Protocol
→ Standards

RFC 792

ICMP - Internet Control Message Protocol

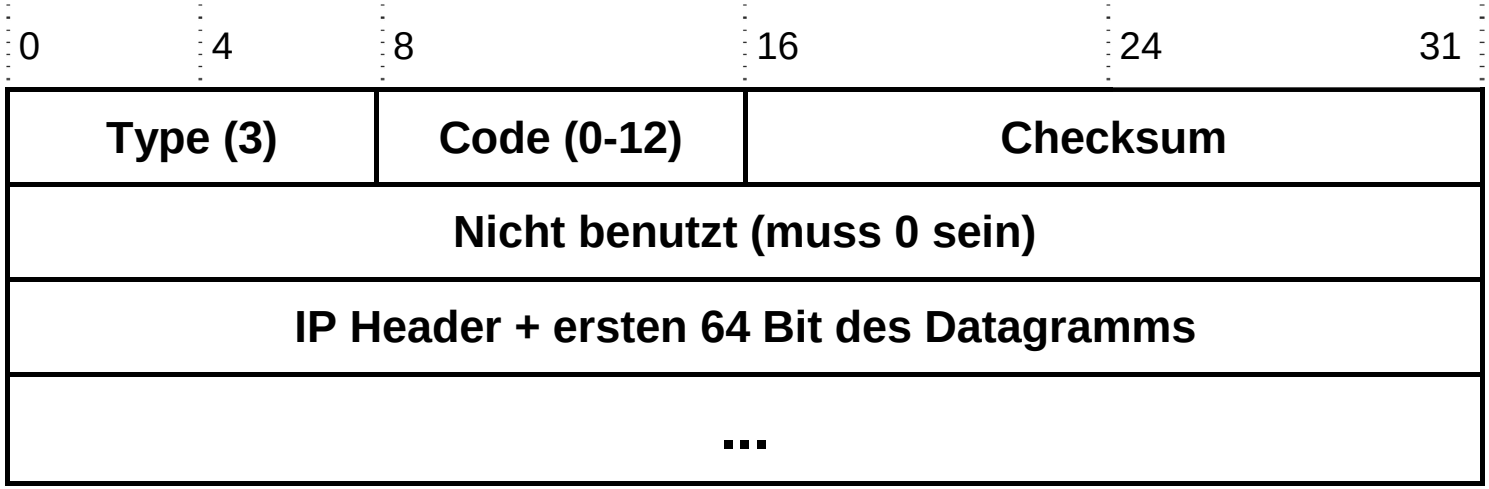
- Das Internet Control Message Protocol (ICMP) ist ein Protokoll der Vermittungsebene und erlaubt es einer IP-Realisierung auf einem Rechner, an die IP-Realisierung eines anderen Rechners **Kontroll- oder Fehlermeldungen** zu schicken.
- Diese Möglichkeit wurde geschaffen, damit Router den Hosts (Rechnern) den Grund eines Fehlers bei der Zustellung eines IP-Paketes zustellen können.
- ICMP ist Bestandteil jeder IP-Implementierung und transportiert Fehler- und Diagnoseinformationen für IP.
- Das ICMP-Paket wird im Datenteil eines IP-Paketes transportiert, seine Transportprotokoll-Adresse im Protokoll-Feld des IP-Headers ist „1“.
- Dennoch wird **ICMP** nicht als ein Protokoll der höheren Schicht, sondern als **Bestandteil der Vermittlungsebene betrachtet**.

ICMP - Internet Control Message Protocol

- Jede ICMP-Nachricht hat ihr eigenes Format, alle beginnen jedoch mit drei identischen Feldern:
  - einem 8 Bit TYPE-Feld, das die Art der ICMP-Nachricht angibt
  - einem 8 Bit CODE-Feld, das weitergehende Informationen enthalten kann
  - und einem 16 Bit CHECKSUM-Feld, das eine Prüfsumme über das ICMP-Paket enthält.

- **Die restlichen Felder des Paketes sind abhängig von der ICMP-Nachricht.**

ICMP - Internet Control Message Protocol
→ Beispiel: Destination Unreachable-Nachricht [22]



- Mit diesem Datagramm wird dem Absender mitgeteilt, dass sein Paket nicht zugestellt werden könnte.
- Das CODE-Feld enthält weitergehende Informationen zum Grund des Fehlers, wie z.B. „Network Unreachable (0)“ oder Fragmentierung „needed and DF set (4)“.
- Im Datenteil sind außerdem der Header des betroffenen Pakets und die ersten 64 Bit des Datagramms enthalten, woraus der Sender zweifelsfrei ermitteln kann, welches Paket gemeint ist.

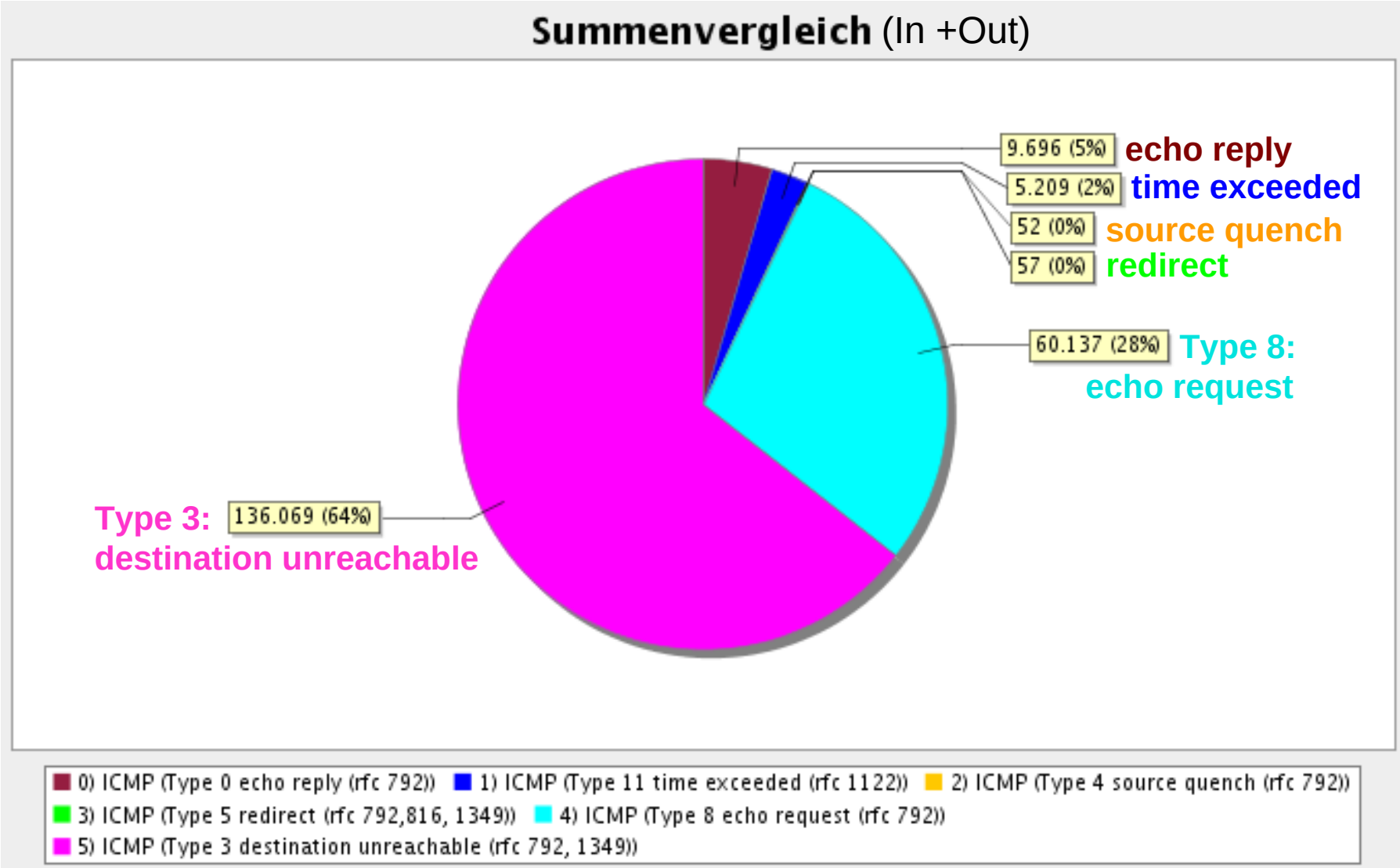
ICMP - Internet Control Message Protocol

- Die von ICMP unterstützten Kontrollnachrichten sind:
  - **destination unreachable (Type 3)**
    Ein Datagramm konnte nicht zugestellt werden, da ein Netzwerk oder Rechner nicht erreichbar war, ein Protokoll nicht betriebsbereit war, oder Fragmentierung notwendig gewesen wäre, aber durch das Flag-Feld (im IP-Header) verboten wurde.
  - **time exceeded (Type 11)**
    Ein Datagramm wurde weggeworfen, da seine Lebensdauer ablief oder ein Fragment wurde weggeworfen, weil es zu lange in der Warteschlange für die Reassemblierung war.
  - **parameter problem (Type 12)**
    Der Absender eines IP-Datagarmms wird verständigt, dass das Paket aufgrund von fehlerhaften Angaben im IP-Protokollkopf weggeworfen werden musste.
  - **source quench (Type 4)**
    Ein Netzwerkgerät wirft Datagramme weg, da es zu wenig Betriebsmittel (z.B. Zwischenspeicher) hat.

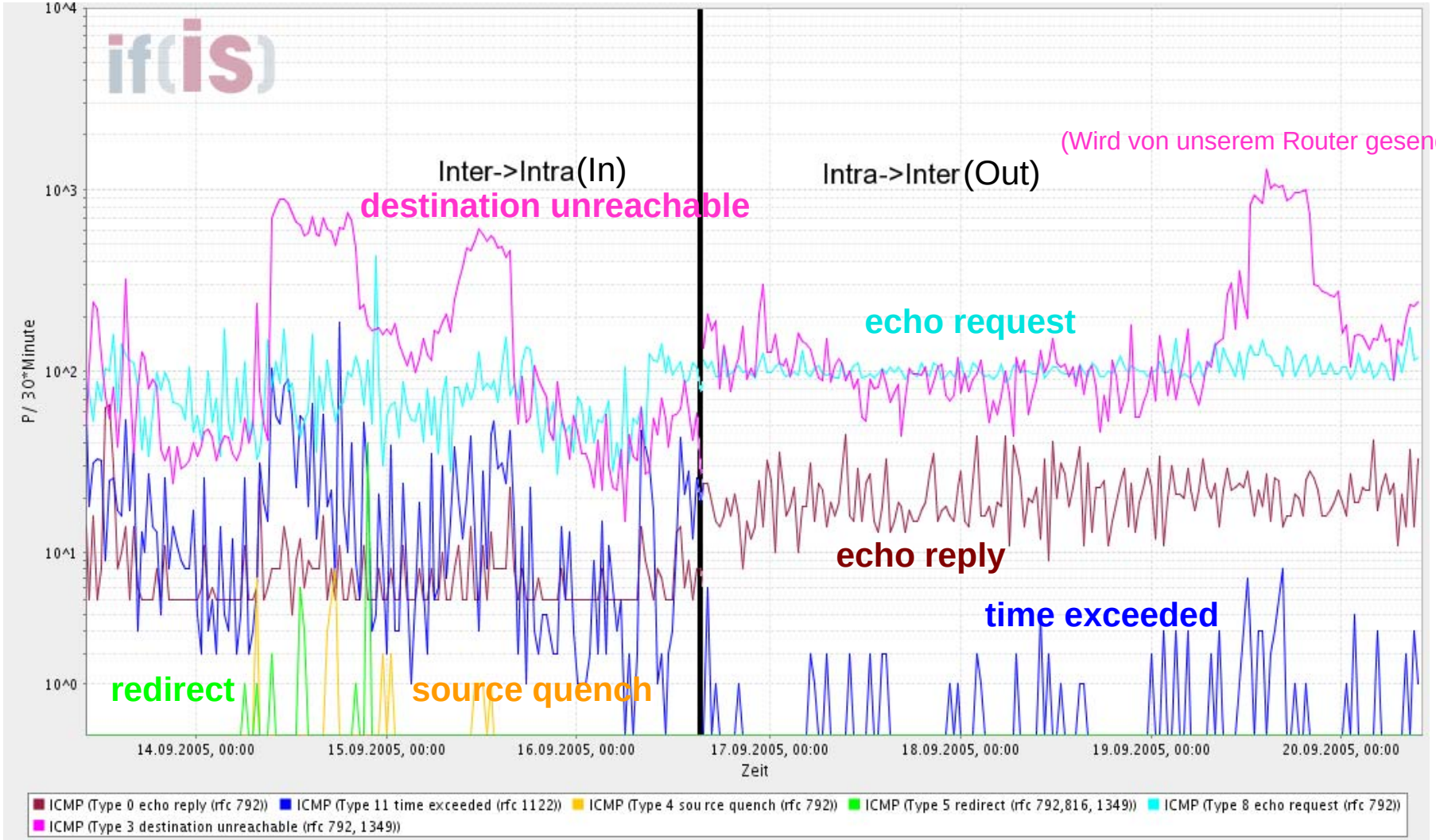
ICMP - Internet Control Message Protocol

- **redirect (Type 5)**
  Wird ausgesendet, wenn ein Gateway erkennt, dass der Absender eines IP-Paketes dieses direkt an den nächsten Gateway senden könnte, d.h. ein unnötiger Umweg gegangen wird. Die ICMP-Nachricht enthält die Internet-Adresse des nächsten direkten Gateway **(Sicherheitsproblem!)**
- **echo request / echo reply (Type 8/0)**
  Zum Test, ob eine IP-Adresse existiert, wird eine *echo request*-Nachricht gesendet. Nach Erhalt einer solchen Nachricht antwortet die empfangende Einheit mit einer *echo reply*-Nachricht.
- **timestamp request / timestamp reply (Type 15/16)**
  Zum Feststellen der Verzögerung im Netzwerk zwischen zwei Netzwerkgeräten.
- **address mask request / address mask reply (Type 17/18)**
  Zur Bestimmung der Subnetz-Adressmaske.

Internet-Analyse-System: FB Informatik
→ ICMP – (1/2)



Internet-Analyse-System: FB Informatik
→ ICMP – (2/2)

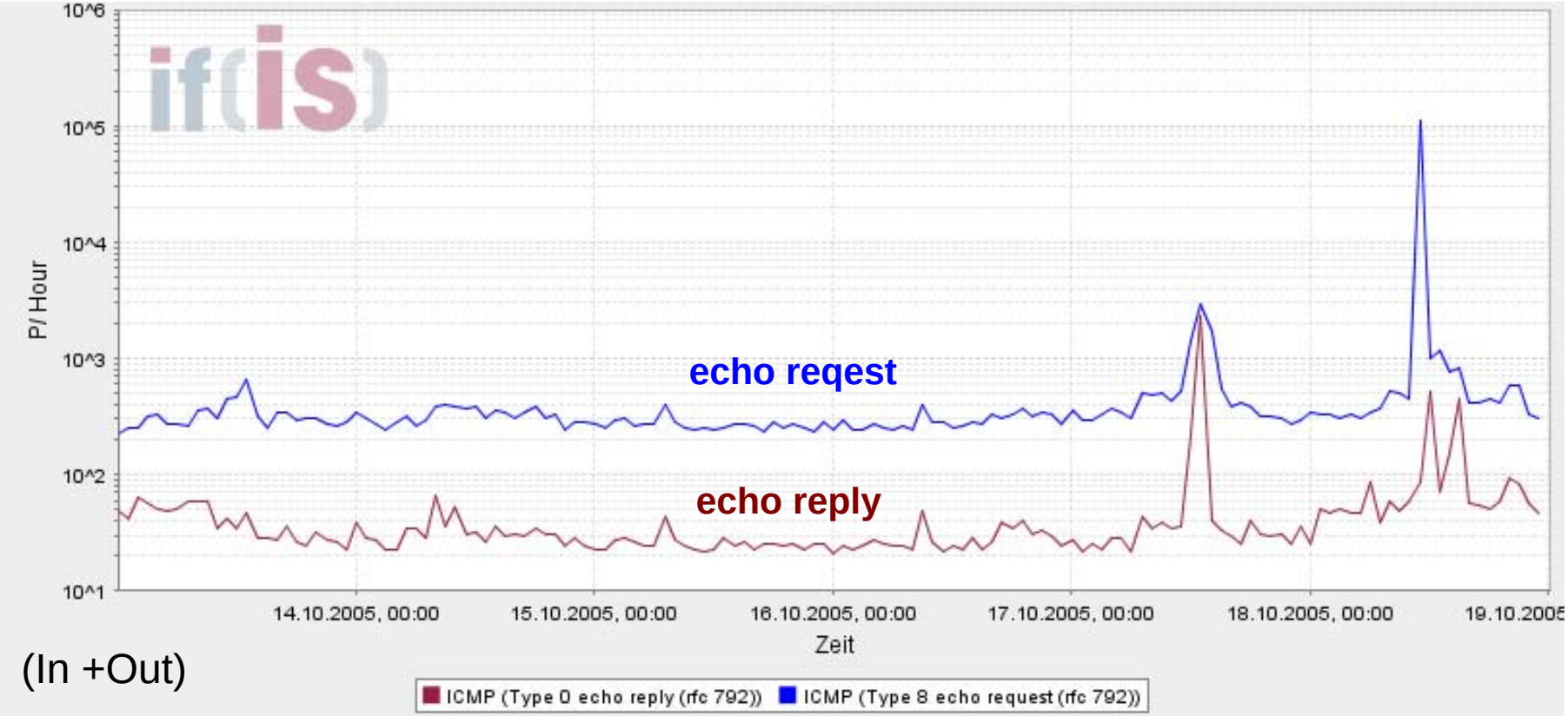


Ping und Traceroute

- Zwei bekannte Programme, die ICMP-Nachrichten benutzen, sind die Programme „ping“ und „traceroute“.
- Mit ping wird eine echo request-Nachricht an einen Host geschickt.
- Wird dieses Paket mit einer echo reply-Nachricht beantwortet, ist sichergestellt, dass der Host erreichbar ist.
- Damit ist gleichzeitig sichergestellt, dass der Protokollstack auf Sender- und Empfängerseite korrekte Einträge in ihren Routing-Tabellen haben.
- Daher stellt dieses Programm ein wichtiges Hilfsmittel beim Test von TCP/IP-Implementierungen dar.
- Je nach Implementierung des ping-Kommandos werden mehrere ICMP echo request Pakete ausgesendet und eine **Statistik über die Anzahl der verlorengegangenen Pakete** und die Zeit, die die Pakete benötigen, angezeigt.

Internet-Analyse-System: FB Informatik
→ Ping

- Normal: 500/h
- Ping of Death/DOS Floodping
  - Etwas rechts: Ping auf den noch geantwortet wird („nur“ 5000/h) – nicht erfolgreich!
  - Ganz rechts: DOS Floodping 10**5 /h! – erfolgreich!



Beispiel eins „Pings“

Ping

```
vmsuse80:/ # ping www.heise.de
PING www.heise.de (193.99.144.71) from 172.16.48.111 : 56(84) bytes of
data.
64 bytes from www.heise.de (193.99.144.71): icmp_seq=1 ttl=244 time=18.2 ms
64 bytes from www.heise.de (193.99.144.71): icmp_seq=2 ttl=244 time=16.8 ms
64 bytes from www.heise.de (193.99.144.71): icmp_seq=3 ttl=244 time=15.9 ms
64 bytes from www.heise.de (193.99.144.71): icmp_seq=4 ttl=244 time=16.8 ms
64 bytes from www.heise.de (193.99.144.71): icmp_seq=5 ttl=244 time=17.0 ms

--- www.heise.de ping statistics ---
6 packets transmitted, 5 received, 16% loss, time 5055ms
rtt min/avg/max/mdev = 15.910/16.955/18.201/0.746 ms
vmsuse80:/ #
```

Welche Gründe kann das haben?

Beispiel eins „Pings“

- Verlustrate im Internet liegt bei 1 bis 5 % (Best Effort Prinzip)

- Gründe der Verlustrate:
  - **Ein Netzwerkgerät (z.B. Router) wirft Datagramme weg, da es zu wenig Betriebsmittel (z.B. Zwischenspeicher) hat.**
  - Ein Datagramm wurde weggeworfen, da seine Lebensdauer ablief (TTL = 0; zu viele „hops“).
  - Fehler im IP-Header (Übertragungsfehler)
  - Da eine Fragmentierung notwendig gewesen wäre, aber durch das Flag-Feld (im IP-Header) verboten wurde.
  - Ein Fragment wurde weggeworfen, weil es zu lange in der Warteschlange für die Reassemblierung war (falls fragmentiert wurde).
  - ...

Traceroute

- Mit Hilfe des Programms „traceroute“ (UNIX) oder „tracert“ (MS) kann der Weg, den ein IP-Datagramm nimmt, verfolgt und angezeigt werden.
- Dazu sendet „traceroute“ eine Reihe von IP-Datagrammen aus, bei denen das TTL-Feld von 1 ansteigende Werte enthält.
- Aus den zurückkommenden TIME EXEEDED-Meldungen kann entnommen werden, welche Stationen das Datagramm durchlaufen hat.

Traceroute
→ Beispiel

Traceroute

```
vmsuse80:/ # traceroute www.heise.de
traceroute to www.heise.de (193.99.144.71), 30 hops max, 40 byte packets
 1  gw502_48.informatik.fh-ge.de (172.16.48.2)  1 ms  1 ms  1 ms
 2  fb5gwint.informatik.fh-ge.de (172.16.0.5)  1 ms  1 ms  1 ms
 3  172.16.16.3 (172.16.16.3)  2 ms  2 ms  2 ms
 4  fb5gw.informatik.fh-gelsenkirchen.de (194.94.127.2)  3 ms  3 ms  3 ms
 5  193.175.172.2 (193.175.172.2)  3 ms  3 ms  3 ms
 6  ar-essen2.g-win.dfn.de (188.1.44.33)  6 ms  5 ms  5 ms
 7  cr-essen1-ge0-0.g-win.dfn.de (188.1.86.1)  5 ms  5 ms  5 ms
 8  cr-frankfurt1-po8-1.g-win.dfn.de (188.1.18.89)  16 ms  16 ms  16 ms
 9  ir-frankfurt2-po3-0.g-win.dfn.de (188.1.80.38)  15 ms  15 ms  15 ms
10  de-cix2.ffm.plusline.net (80.81.193.132)  16 ms  16 ms  15 ms
11  c22.f.de.plusline.net (213.83.57.53)  16 ms  16 ms  16 ms
12  www.heise.de (193.99.144.71)  16 ms  16 ms  17 ms
vmsuse80:/ #
```

- Zeitanalyse:
  - Verweildauer in einem Router: ca. 0.1 bis 1.5 ms ? **(12 * 1ms = 12ms)**
  - Übertragungszeit für ein Paket bei einer Übertragungsrate von 2 Mbits: 1.500 Byte: ca. 6 ms; 100 Byte: ca. 0.4 ms **(ca. 5ms)**

Traceroute

→ Protokollmitschnitt (1/7)

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
2	0.000455	172.16.48.2	172.16.48.216	ICMP	Time-to-live exceeded
3	0.000648	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
4	0.001045	172.16.48.2	172.16.48.216	ICMP	Time-to-live exceeded
5	0.002423	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
6	0.002890	172.16.48.2	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 1

10	5.512689	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
11	5.513506	172.16.0.5	172.16.48.216	ICMP	Time-to-live exceeded
12	5.514953	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
13	5.515733	172.16.0.5	172.16.48.216	ICMP	Time-to-live exceeded
14	5.515854	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
15	5.516615	172.16.0.5	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 2

19	11.023238	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
20	11.025016	172.16.16.3	172.16.48.216	ICMP	Time-to-live exceeded
21	11.025245	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
22	11.027058	172.16.16.3	172.16.48.216	ICMP	Time-to-live exceeded
23	11.027303	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
24	11.029649	172.16.16.3	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 3

25	16.537852	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
26	16.540127	194.94.127.2	172.16.48.216	ICMP	Time-to-live exceeded
27	16.540385	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
28	16.543626	194.94.127.2	172.16.48.216	ICMP	Time-to-live exceeded
29	16.543906	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
30	16.546613	194.94.127.2	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 4

Traceroute

→ Protokollmitschnitt (2/7)

No.	Time	Source	Destination	Protocol	Info
31	17.545072	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
32	17.552839	193.175.172.2	172.16.48.216	ICMP	Time-to-live exceeded
33	17.553142	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
34	17.556848	193.175.172.2	172.16.48.216	ICMP	Time-to-live exceeded
35	17.557119	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
36	17.571955	193.175.172.2	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 5

40	23.061025	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
41	23.112306	188.1.44.33	172.16.48.216	ICMP	Time-to-live exceeded
42	23.112610	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
43	23.122883	188.1.44.33	172.16.48.216	ICMP	Time-to-live exceeded
44	23.123173	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
45	23.375880	188.1.44.33	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 6

46	24.126314	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
47	24.130746	188.1.86.1	172.16.48.216	ICMP	Time-to-live exceeded
48	24.131031	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
49	24.135302	188.1.86.1	172.16.48.216	ICMP	Time-to-live exceeded
50	24.135552	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
51	24.139830	188.1.86.1	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 7

52	25.136544	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
53	25.150861	188.1.18.89	172.16.48.216	ICMP	Time-to-live exceeded
54	25.151153	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
55	25.165482	188.1.18.89	172.16.48.216	ICMP	Time-to-live exceeded
56	25.165770	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
57	25.182477	188.1.18.89	172.16.48.216	ICMP	Time-to-live exceeded

Time to live: 8

Traceroute
→ Protokollmitschnitt (3/7)

No.	Time	Source	Destination	Protocol	Info
58	26.166823	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
59	26.181683	188.1.80.42	172.16.48.216	ICMP	Time-to-live exceeded
60	26.181974	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
61	26.196689	188.1.80.42	172.16.48.216	ICMP	Time-to-live exceeded
62	26.196982	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
63	26.212469	188.1.80.42	172.16.48.216	ICMP	Time-to-live exceeded
Time to live: 9					
67	31.702500	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
68	31.717414	80.81.193.132	172.16.48.216	ICMP	Time-to-live exceeded
69	31.717708	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
70	31.732510	80.81.193.132	172.16.48.216	ICMP	Time-to-live exceeded
71	31.732891	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
72	31.748401	80.81.193.132	172.16.48.216	ICMP	Time-to-live exceeded
Time to live: 10					
73	32.733773	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
74	32.751919	213.83.57.53	172.16.48.216	ICMP	Time-to-live exceeded
75	32.753005	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
76	32.767937	213.83.57.53	172.16.48.216	ICMP	Time-to-live exceeded
77	32.768218	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
78	32.783387	213.83.57.53	172.16.48.216	ICMP	Time-to-live exceeded
Time to live: 11					
79	33.770668	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
80	33.791504	193.99.144.71	172.16.48.216	ICMP	Echo (ping) reply
81	33.791803	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
82	33.807985	193.99.144.71	172.16.48.216	ICMP	Echo (ping) reply
83	33.808285	172.16.48.216	193.99.144.71	ICMP	Echo (ping) request
84	33.824839	193.99.144.71	172.16.48.216	ICMP	Echo (ping) reply
Time to live: 12					

Traceroute
→ Protokollmitschnitt (4/7)

```
Frame 1 (106 bytes on wire, 106 bytes captured)
Internet Protocol, Src Addr: 172.16.48.216 (172.16.48.216), Dst Addr: 193.99.144.71
(193.99.144.71)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 92
    Identification: 0x0b2d (2861)
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 1
    Protocol: ICMP (0x01)
    Header checksum: 0x7fe1 (correct)
    Source: 172.16.48.216 (172.16.48.216)
    Destination: 193.99.144.71 (193.99.144.71)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x7fff (correct)
    Identifier: 0x0200
    Sequence number: 76:00
    Data (64 bytes)
```

Traceroute
→ Protokollmitschnitt (5/7)

```
Frame 2 (134 bytes on wire, 134 bytes captured)
Internet Protocol, Src Addr: 172.16.48.2 (172.16.48.2), Dst Addr: 172.16.48.216
(172.16.48.216)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
    Total Length: 120, Identification: 0x59f5 (23029)
    Flags: 0x00, Fragment offset: 0, Time to live: 255
    Protocol: ICMP (0x01)
    Header checksum: 0xa7d4 (correct)
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (TTL equals 0 during transit)
    Checksum: 0xf4ff (correct)
    Internet Protocol, Src Addr: 172.16.48.216 (172.16.48.216), Dst Addr: 193.99.144.71
       (193.99.144.71)
        Version: 4
        Header length: 20 bytes
        Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
         Total Length: 92
        Identification: 0x0b2d (2861)
        Flags: 0x00
        Fragment offset: 0
        Time to live: 1
        Protocol: ICMP (0x01)
        Header checksum: 0x7fe1 (correct)
     Internet Control Message Protocol
        Type: 8 (Echo (ping) request)
        Code: 0
        Checksum: 0x7fff (correct)
        Identifier: 0x0200
        Sequence number: 76:00
        Data (64 bytes)
```

Traceroute
→ Protokollmitschnitt (6/7)

```
Frame 10 (106 bytes on wire, 106 bytes captured)
Internet Protocol, Src Addr: 172.16.48.216 (172.16.48.216), Dst Addr: 193.99.144.71
(193.99.144.71)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 92
    Identification: 0x0b34 (2868)
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 2
    Protocol: ICMP (0x01)
    Header checksum: 0x7eda (correct)
    Source: 172.16.48.216 (172.16.48.216)
    Destination: 193.99.144.71 (193.99.144.71)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x7cff (correct)
    Identifier: 0x0200
    Sequence number: 79:00
    Data (64 bytes)
```

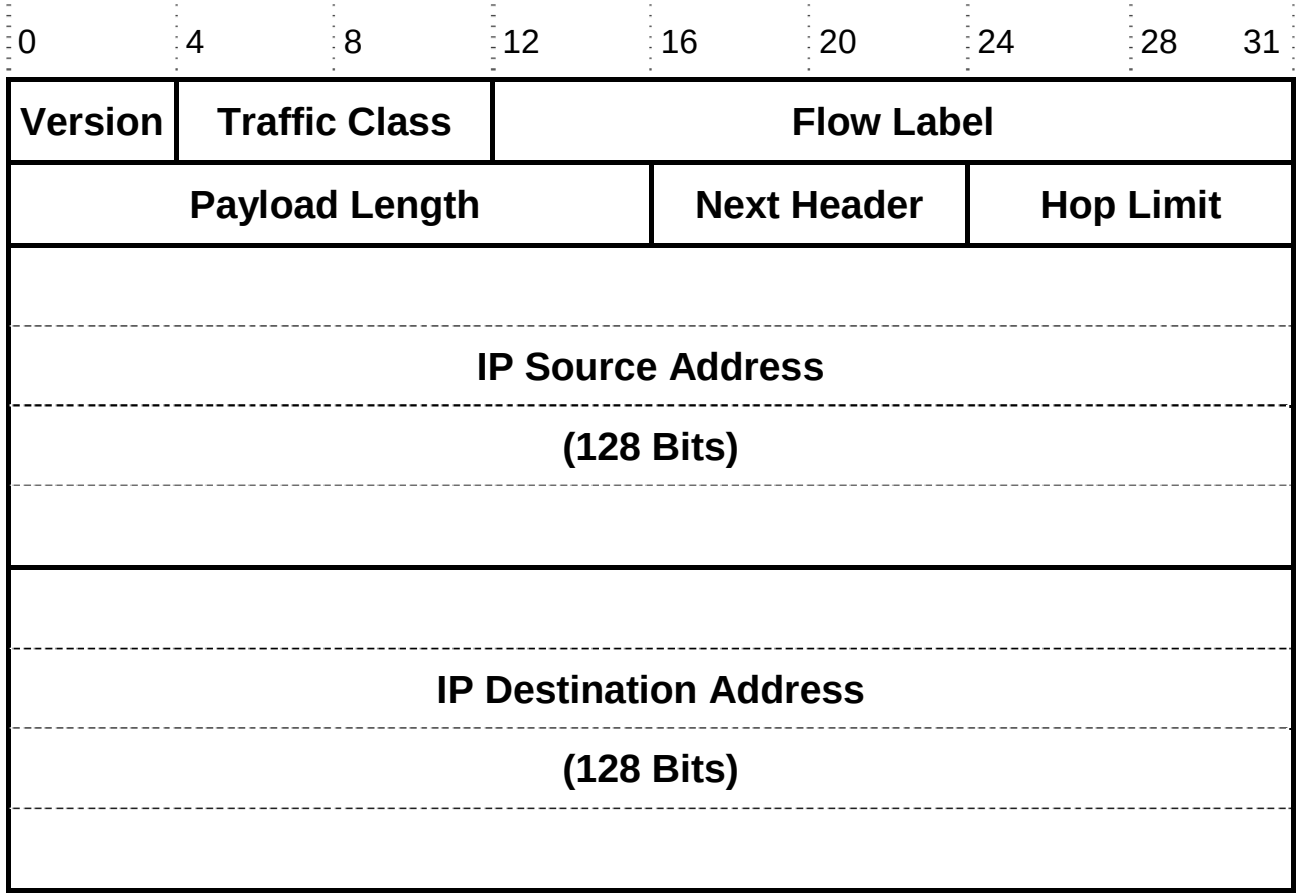
Traceroute
→ Protokollmitschnitt (7/7)

```
Frame 11 (134 bytes on wire, 134 bytes captured)
Internet Protocol, Src Addr: 172.16.0.5 (172.16.0.5), Dst Addr: 172.16.48.216 (172.16.48.216)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
    Total Length: 120, Identification: 0xfe06 (65030)
    Flags: 0x00, Fragment offset: 0, Time to live: 254
    Protocol: ICMP (0x01)
    Header checksum: 0x34c0 (correct)
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (TTL equals 0 during transit)
    Checksum: 0xf4ff (correct)
    Internet Protocol, Src Addr: 172.16.48.216 (172.16.48.216), Dst Addr: 193.99.144.71
       (193.99.144.71)
        Version: 4
        Header length: 20 bytes
        Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        Total Length: 92
        Identification: 0x0b34 (2868)
        Flags: 0x00
        Fragment offset: 0
        Time to live: 1
        Protocol: ICMP (0x01)
        Header checksum: 0x7fda (correct)
     Internet Control Message Protocol
        Type: 8 (Echo (ping) request)
        Code: 0
        Checksum: 0x7cff (correct)
        Identifier: 0x0200
        Sequence number: 79:00
        Data (64 bytes)
```

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- **IPv6**
- Zusammenfassung

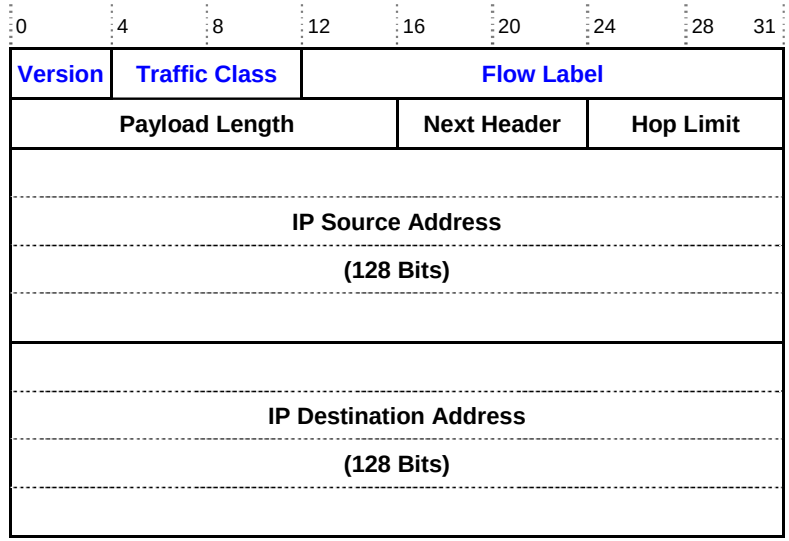
IPv6
→ Header: Übersicht



Basis-Header

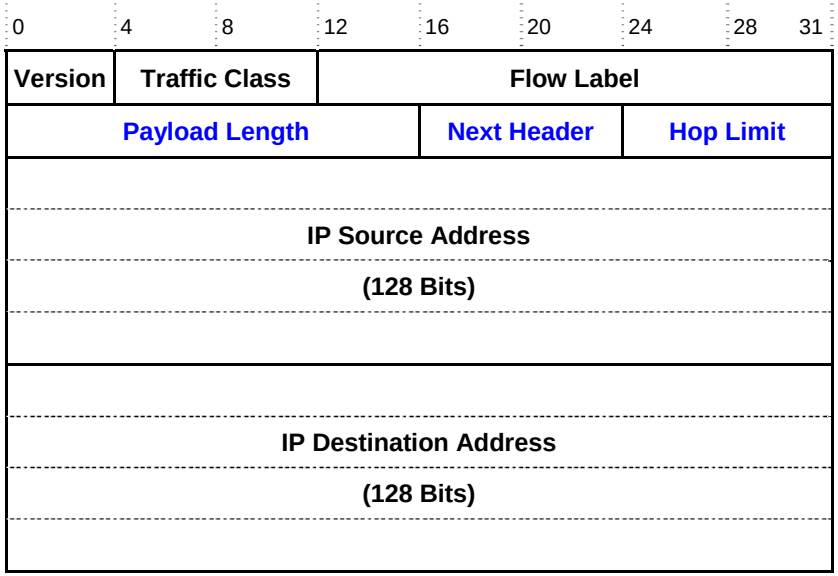
Erweiterungs-Header

IPv6
→ Header: Feldelemente (1/3)



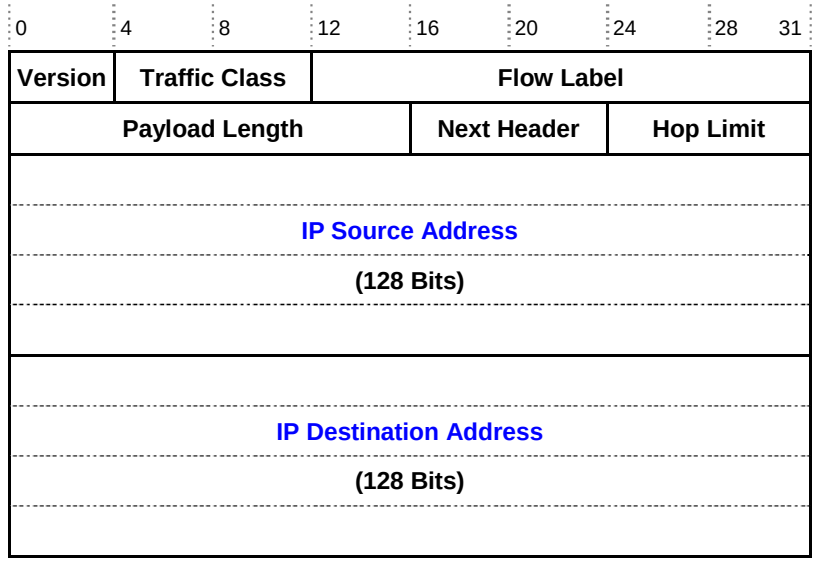
- **Version (Vers)**
  - Feldlänge: 4 Bit
  - **Beschreibung**
    - IP-Versionsnummer (Wert=6)
- **Traffic Class**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Für Quality of Service (QoS) verwendeter Wert
- **Flow Label**
  - Feldlänge: 20 Bit
  - **Beschreibung**
    - Ebenfalls für QoS oder Echtzeitanwendungen verwendeter Wert. Bietet Platz für eine zufällig gewählte Identifikationsnummer für End-to-End-Verbindungen.

IPv6
→ Header: Feldelemente (2/3)



- **Payload Length**
  - Feldlänge: 16 Bit
  - **Beschreibung**
    - Länge des IPv6-Paketinhaltes (ohne Header aber inklusive der Erweiterungs-Header)
- **Next Header**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Identifiziert den Typ des nächsten Extension Headers
- **Hop Limit**
  - Feldlänge: 8 Bit
  - **Beschreibung**
    - Maximale Anzahl an Zwischenschritten über Router, die ein Paket zurücklegen darf; wird beim Durchlaufen eines Routers ("Hops") um Eins verringert. Pakete mit Null als Hop Limit werden verworfen

IPv6
→ Header: Feldelemente (3/3)



- **IP Source Address (Source)**
  - Feldlänge: 128 Bit
  - **Beschreibung**
    - Enthält die Internet-Adresse des Netzknotens, der das Datagramm erzeugt hat

- **IP Destination Address (Dest)**
  - Feldlänge: 128 Bit
  - **Beschreibung**
    - Enthält die Internet-Adresse des Netzknotens, für den das Datagramm bestimmt ist

IPv6
→ Die wesentlichen Merkmale von IPv6 (1/2)

- Adressgröße:
  - Als wichtigstes Merkmal hat IPv6 gegenüber IPv4 größere Adressen.
  - Statt bisher 32 Bit stehen nun 128 Bit für die Adressen bereit.
  - Theoretisch lassen sich damit $2^{128} = 3.4*10^{38}$ (statt $4*10^{9}$) Adressen vergeben.

- Header-Format:
  - Der IPv6 (Basis)Header wurde vollständig geändert.
  - Der Header enthält nur 7 statt bisher 13 Felder.
  - Diese Änderung ermöglicht Routern, Pakete schneller zu verarbeiten.
  - Im Gegensatz zu IPv4 gibt es bei IPv6 nicht mehr nur einen Header, sondern mehrere Header.
  - Ein Datengramm besteht aus einem Basis-Header, sowie einem oder mehreren Zusatz-Headern, gefolgt von den Nutzdaten.

- Erweiterte Unterstützung von Optionen und Erweiterungen:
  - Die Erweiterung der Optionen ist notwendig geworden, da einige, bei IPv4 notwendige Felder nun optional sind.
  - Darüber hinaus unterscheidet sich auch die Art, wie die Optionen dargestellt werden.
  - Für Router wird es damit einfacher, Optionen, die nicht für sie bestimmt sind, zu überspringen.
  - Dies ermöglicht ebenfalls eine schnellere Verarbeitung von Paketen.

IPv6
→ Die wesentlichen Merkmale von IPv6 (2/2)

- Dienstarten:
  - IPv6 legt mehr Gewicht auf die Unterstützung von Dienstarten.
  - Damit kommt IPv6 den Forderungen nach einer verbesserten Unterstützung der Übertragung von Video- und Audiodaten entgegen.
  - IPv6 bietet hierzu eine Option zur Echtzeitübertragung.

- Sicherheit:
  - IPv6 beinhaltet nun im Protokoll selbst Mechanismen zur sicheren Datenübertragung.
  - Wichtige neue Merkmale von IPv6 sind hier Authentifikation (authentication), Datenintegrität (data integrity) und Datenvertraulichkeit (data confidentiality).

- Erweiterbarkeit:
  - IPv6 ist ein erweiterbares Protokoll.
  - Bei der Spezifikation des Protokolls wurde nicht versucht, alle potentiell möglichen Einsatzfelder für das Protokoll in die Spezifikation zu integrieren.
  - Vielmehr bietet IPv6 die Möglichkeit, über Erweiterungs-Header (s.u.) das Protokoll zu erweitern.
  - Damit ist das Protokoll offen für zukünftige Verbesserungen.

IPv6
→ Adresskategorien

- Unicast-Adressen
  - Eine Unicast-Adresse kennzeichnet eine einzelne Netzwerk-Schnittstelle, also einen einzelnen Rechner.
  - Unicast-Adressen werden für die Unterstützung von Punkt-zu-Punkt-Verbindungen verwendet.

- Multicast-Adressen
  - Eine Multicast-Adresse identifiziert eine Gruppe von mehreren, zusammengehörigen Netzwerk-Schnittstellen.
  - Ein IP-Datagramm mit einer Multicast-Adresse im Empfängerfeld wird an alle Mitglieder der betreffenden Gruppe weiterversendet.

- Anycast-Adressen
  - Eine Anycast-Adresse identifiziert eine Gruppe von Rechnern, die lokal in funktionalem Zusammenhang stehen.
  - Ein IP-Datagramm, das mit einer Anycast-Adresse versehen ist, wird zunächst an einen bestimmten Router im Zielnetzwerk gesendet, der die Weiterleitung an bestimmte Rechner dieses Netzwerkes übernimmt.

IPv6
→ Adressaufbau (1/2)

- IPv6-Adressen werden nicht in dezimaler (zum Beispiel 80.130.234.185), sondern in **hexadezimaler Notation mit Doppelpunkten** geschrieben, die die Adresse in acht Blöcke mit einer Länge von jeweils 16 Bit unterteilen.
  Beispiel einer IPv6-Adresse:
  2001:0db8:85a3:08d3:1319:8a2e:0370:7344
- Eine oder mehrere 16-Bit-Gruppen mit dem Wert 0000 können durch zwei aufeinanderfolgende Doppelpunkte ersetzt werden.
- Die resultierende Adresse darf höchstens einmal zwei aufeinander folgende Doppelpunkte enthalten.
- 2001:0db8::1428:57ab ist gleichbedeutend mit 2001:0db8:0000:0000:0000:0000:1428:57ab, aber
  2001::25de::cade ist nicht korrekt, da nicht nachvollzogen werden kann, wie viele 16-Bit-Gruppen durch die zwei Doppelpunkte jeweils ersetzt wurden.
- Führende Nullen einer 16-Bit-Gruppe dürfen ausgelassen werden, 2001:db8::28:b ist gleichbedeutend mit 2001:0db8::0028:000b.
- Die ersten **64 Bit** der IPv6-Adresse dienen üblicherweise der **Netzadressierung**, die letzten **64 Bit** werden zur **Host-Adressierung** verwendet.

IPv6
→ Adressaufbau (2/2)

- Das Konzept der Netzmasken von IPv4 wird durch Angabe der Präfixlänge des adressierten Subnetzes implementiert.
- Die Präfixlänge in Bits wird als Dezimalzahl mit vorangehendem "/" an die IPv6-Adresse angehängt.
  Beispiel: hat ein Netzwerkgerät die IPv6-Adresse
  2001:0db8:85a3:08d3:1319:8a2e:0370:7344/64
  so stammt es aus dem Subnetz
  2001:0db8:85a3:08d3::/64
  das mit den ersten 64 Bit seiner Adresse identifiziert wird.
- Analog gehört das Subnetz 2001:0db8:85a3:08d3::/64 hierarchisch zum Subnetz mit dem kürzeren Präfix 2001:0db8:85a3::/48.
- Die korrekte Form einer IPv6-Adresse in einem URL ist beispielsweise http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]/
- Diese Notation verhindert die falsche Interpretation von Portnummern als Teil der IPv6-Adresse:
  http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]:443/

IPv6
→ Arten von IPv6-Adressen

- Es gibt verschiedene IPv6-Adressen mit Sonderaufgaben und unterschiedlichen Eigenschaften, die durch die ersten Bits der Adresse (das "Präfix") signalisiert werden.
- Das Präfix 00 steht für IPv4 und IPv4-über-IPv6-Kompabilitätsadressen.
  - Ein geeigneter Router kann diese Pakete zwischen IPv4 und IPv6 konvertieren und so die neue mit der alten Welt verbinden.
- Die Präfixe 2 oder 3 stehen für globale Unicast-Adressen, also eine routbare und weltweit einzigartige Adresse.
- fe80 bis febf sind so genannte linklokale Adressen (link local address), die von Routern nicht weitergeleitet werden dürfen und daher nur im gleichen Teilnetz erreichbar sind.
- Das Präfix ff steht für Multicast-Adressen.
  - ffx1: knotenlokal, diese Pakete verlassen den Knoten nie.
  - ffx2: linklokal, werden von Routern grundsätzlich nie weitergeleitet und können deshalb das Teilnetz nicht verlassen.
  - ffx5: sitelokal, dürfen zwar geroutet werden, jedoch nicht von Border-Routern.
  - ffx8: organisationslokal, die Pakete dürfen auch von Border-Routern weitergeleitet werden, bleiben jedoch "in der Firma" (hierzu müssen seitens des Routing-Protokolls entsprechende Vorkehrungen getroffen werden).
  - ffxe: globaler Multicast, der überall hin geroutet werden darf.

Inhalt

- Ziele und Einordnung
- IP - Internet Protocol (IPv4)
- ARP - Address Resolution Protocol
- Beispiele für die Übertragung eines IP-Paketes
- DHCP – Dynamic Host Configuration Protocol
- ICMP - Internet Control Message Protocol
- IPv6
- **Zusammenfassung**

Die Vermittlungsebene
→ Zusammenfassung: IP - Internet Protocol

- Das Internet-Protokoll bietet den Protokollen der Transportschicht einen **verbindungslosen, unzuverlässigen** Paketübermittlungsdienst.
- Hauptaufgabe von IP sind die Adressierung von Rechnersystemen, das Fragmentieren von Paketen und das Routing der Pakete.
- Es enthält keine Funktion für die Ende-zu-Ende-Sicherung von Nachrichten oder für die Flußkontrolle.
- Pakete werden so gut wie möglich übertragen (**Best Effort Prinzip**) - **garantiert ist die Zustellung allerdings nicht !**
- Jedes IP-Datagramm wird als einzelnes Paket, völlig unabhängig von anderen Datagrammen, durch das Netz zum Empfänger übertragen.
- Für jedes Datagramm wird innerhalb des Netzes der optimale Weg ermittelt.
- Dabei können sich **Datagramme** auf dem Weg zum Empfänger überholen und dadurch in **geänderter Reihenfolge** beim Empfänger eintreffen.

Die Vermittlungsebene
→ Zusammenfassung: ARP, RARP und ICMP

ARP und RARP

- **ARP** ermittelt zu einer bekannten IP-Adresse eine MAC-Adresse.
- **RARP** ermittelt zu einer bekannten MAC-Adresse eine IP-Adresse.

Internet Control Message Protocol (ICMP)

- Das Internet Control Message Protocol (ICMP) ist ein Protokoll der **Vermittlungsebene**.
- ICMP erlaubt einer IP-Realisierung auf einem Rechner, an die IP-Realisierung eines anderen Rechners **Kontroll- oder Fehlermeldungen** zu schicken.

Die Vermittlungsebene

Vielen Dank für Ihre Aufmerksamkeit

Fragen ?

norbert.pohlmann@informatik.fh-gelsenkirchen.de

Fachhochschule Gelsenkirchen