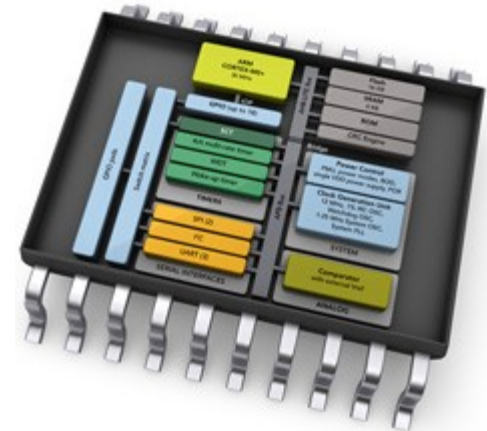


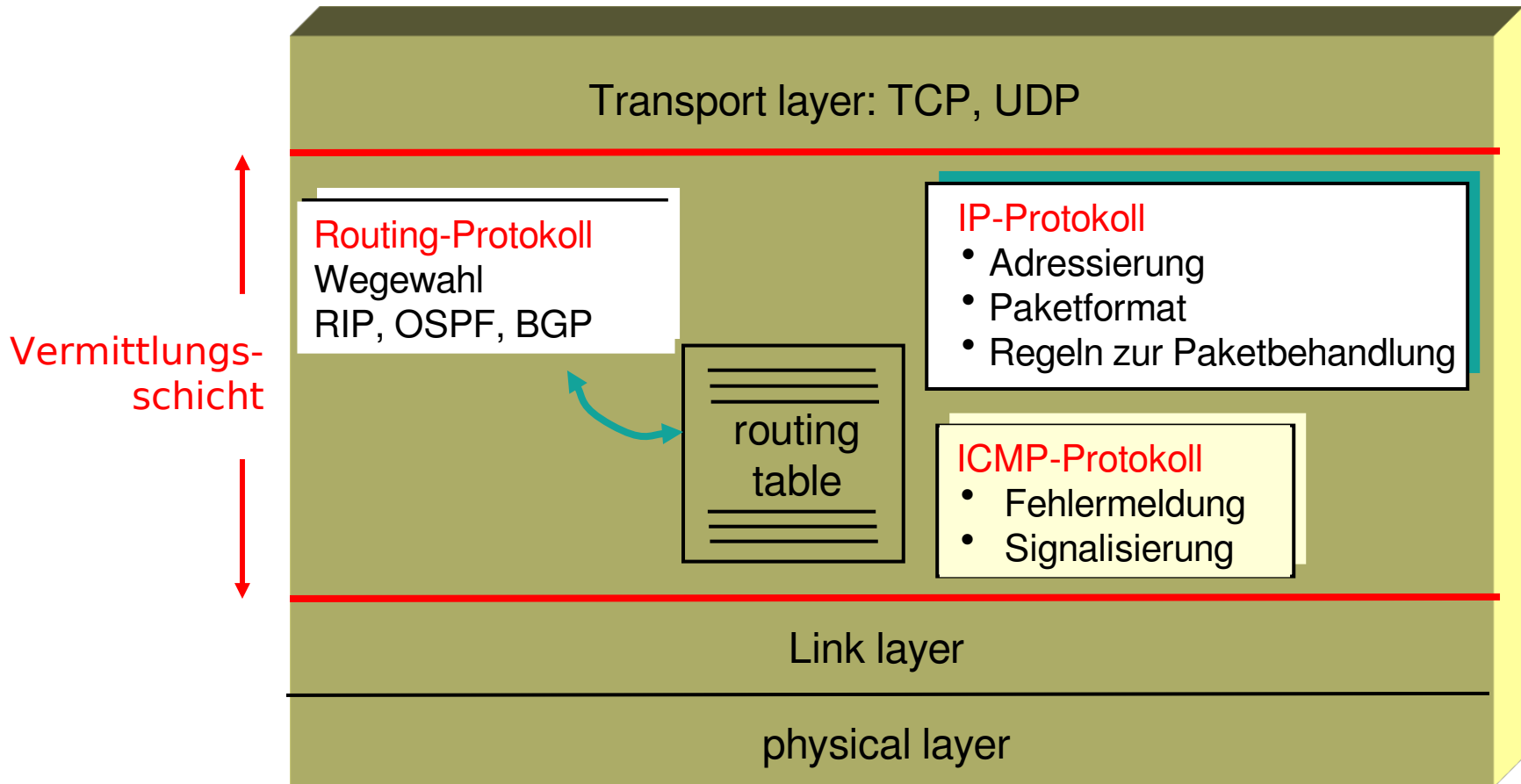
Kommunikationssysteme

(Modulcode 941306)

Prof. Dr. Andreas Terstegge



Die Vermittlungsschicht im Internet



Inhalt

- IP- und MAC Adressen, ARP Protokoll
- RARP, BOOTP, DHCP
- IP Header
- Fragmentierung
- Path MTU Discovery
- ICMP
- IPv6

Adressierung im Internet: IP Adresse

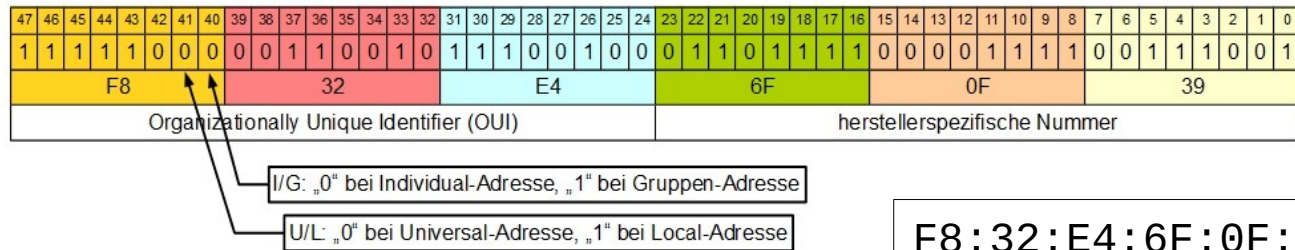
- 32-Bit
- Ist eine **logische Adresse**, die Topologische Informationen enthält (Netzwerk/Host)
→ Wird zur Wege-Findung über Netze hinweg verwendet
- ggf. nicht eindeutig (siehe private Netze)
- Soll ggf. mit anderer Hardware **wiederverwendet** werden können
- Wäre schon lange zu klein um jeden einzelnen Rechner zu adressieren

Adressierung im Internet: IP und MAC Adresse

- Schon kennengelernt: **IP-(Ziel)Adresse** als zentrale Information zur Weiterleitung von IP Datagrammen
- Auf Layer-2 Ebene existiert aber die sog. **MAC-Adresse**
- Warum?
- IP-Adresse: ‚Logische‘ Adressierung (Layer 3)
- MAC-Adresse: ‚Physikalische Adressierung‘ (Layer 2)
- Sinnvoll z.B. beim Szenario: ‚Austausch eines Servers‘
Neue MAC-Adresse, alte IP !

Adressierung im Internet: MAC Adresse

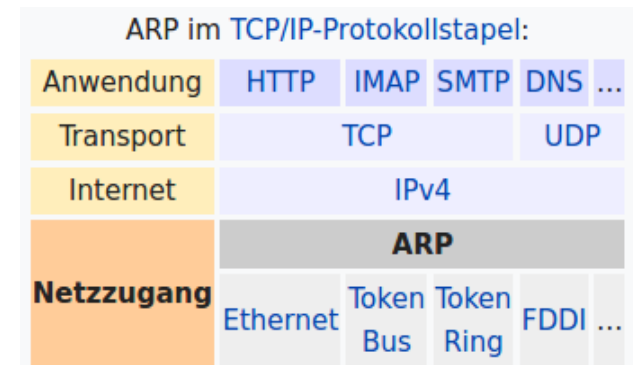
- 48-Bit



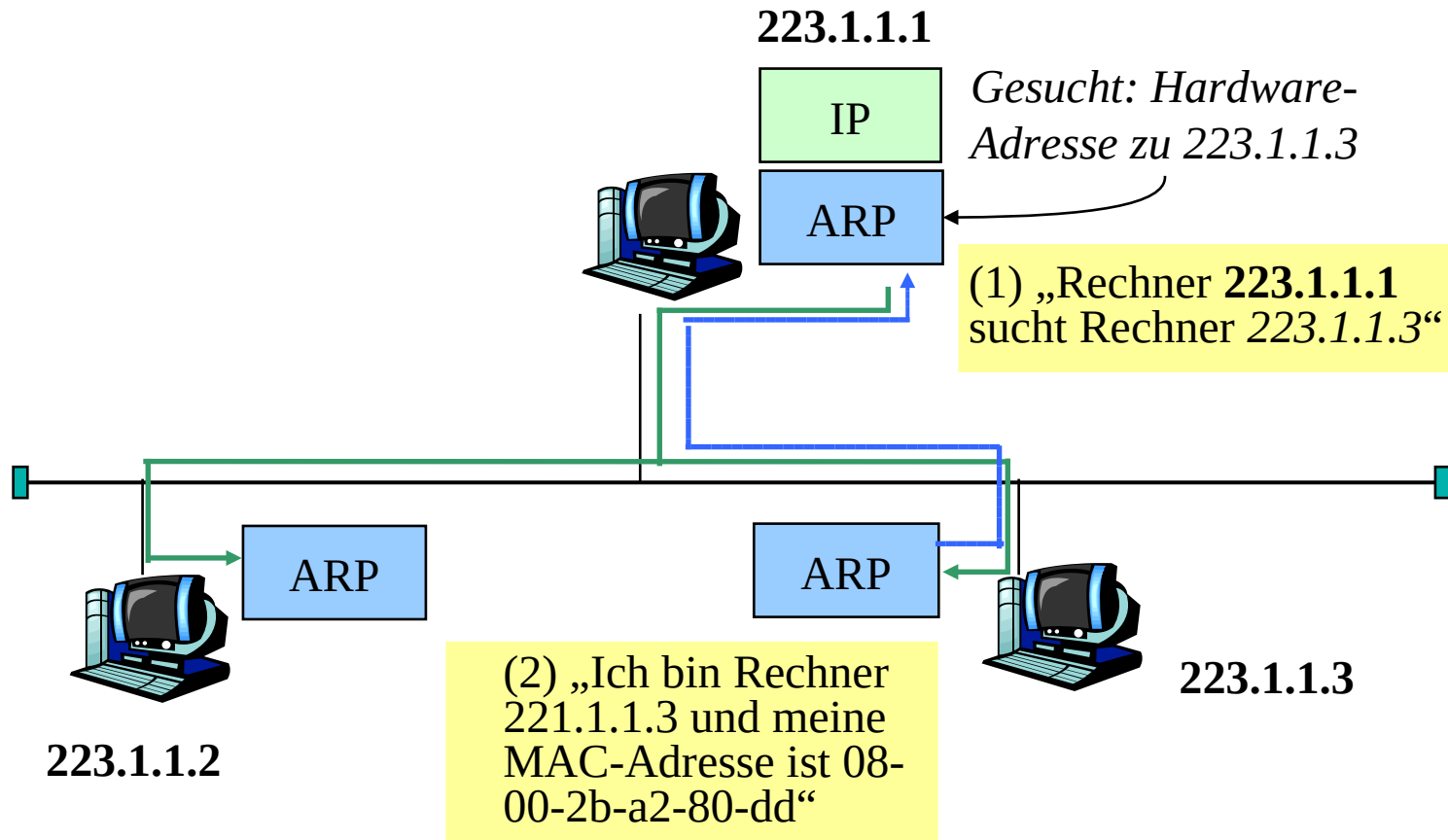
- Ist eine **pysikalische Adresse**, die neben einer ID lediglich Informationen zum Hersteller enthält → kein Routing möglich
- 22 Bit Herstellernummer, 24 ,Bit Seriennummer‘
- **Broadcast**-Adresse: FF:FF:FF:FF:FF:FF
- Ist für jede Netzwerk-Schnittstelle **eindeutig**
- Dient der Adressierung von Paketen **innerhalb des gleichen Subnetzes**
- Ermöglicht die Identifizierung eines neuen Knotens auch **ohne vorherige Konfiguration**
- Ist auf Layer-2 ebene einfach zu vergleichen

Adressierung im Internet: IP & MAC Adresse

- Wenn ein IP Datagram über Schicht 2 versendet werden soll, muss die MAC-Adresse bekannt sein!
- Layer-2 Protokoll:
ARP (**A**ddress **R**esolution **P**rotocol)
- Ermöglicht das Herausfinden einer MAC-Adresse auf Basis einer versendeten IP-Adresse
- Funktioniert über Broadcast-Paket an MAC FF:FF:FF:FF:FF:FF
- Der Host mit der angefragten IP-Adresse antwortet mit seiner MAC-Adresse
- Der empfangende Knoten ‚cached‘ i.d.R. die MAC-Adressen



Address Resolution Protocol (ARP)

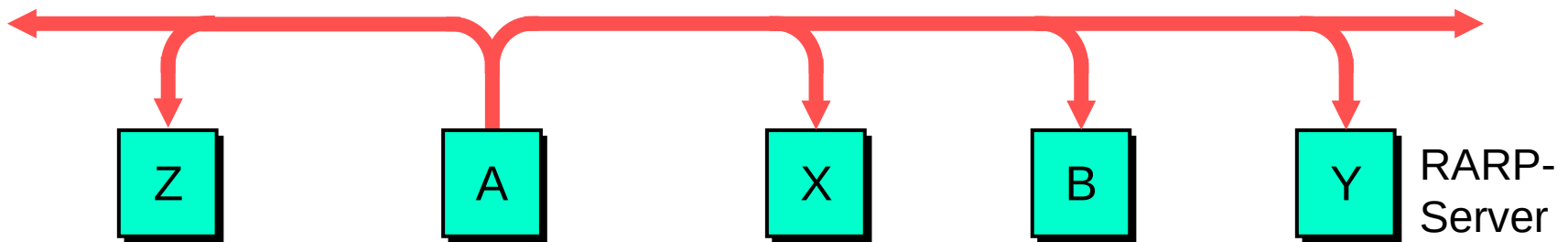


RARP

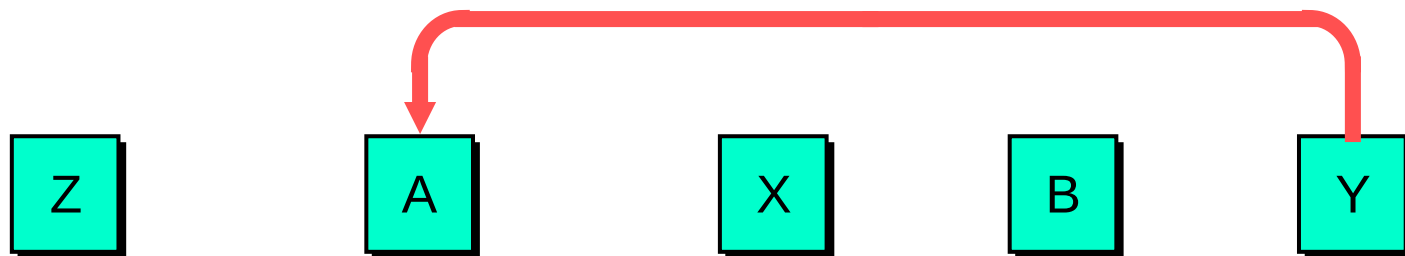
- Beim Neuanschluss eines Knotens an das Netz entsteht das umgekehrte Problem:
- Ich bin ein Knoten mit MAC-Adresse X, und brauche zur Kommunikation im Internet eine **IP**
- Einfache Lösung:
Manuelle Konfiguration
- Benutzerfreundliche Lösung:
Automatisch per Protokoll
- Veraltetes Protokoll:
RARP (**R**everse **A**ddress **R**esolution **P**rotokoll)

RARP (Reverse Address Resolution Protocol)

A sendet RARP-Request mit PHY(A) als Broadcast



RARP-Server Y antwortet mit RARP-Reply mit IP(A)



Probleme von RARP

- Ethernet-Broadcasts sind auf Subnetze beschränkt. In einem LAN mit Subnetze braucht man mehrere RARP-Server
- Durch RARP erfährt ein Rechner nur seine IP-Adresse! Zu einer vollständigen Konfiguration einer Netzwerkschnittstelle gehören noch mindestens Netzmaske und Default-Gateway.
- DHCP hat RARP heute komplett abgelöst!

DHCP

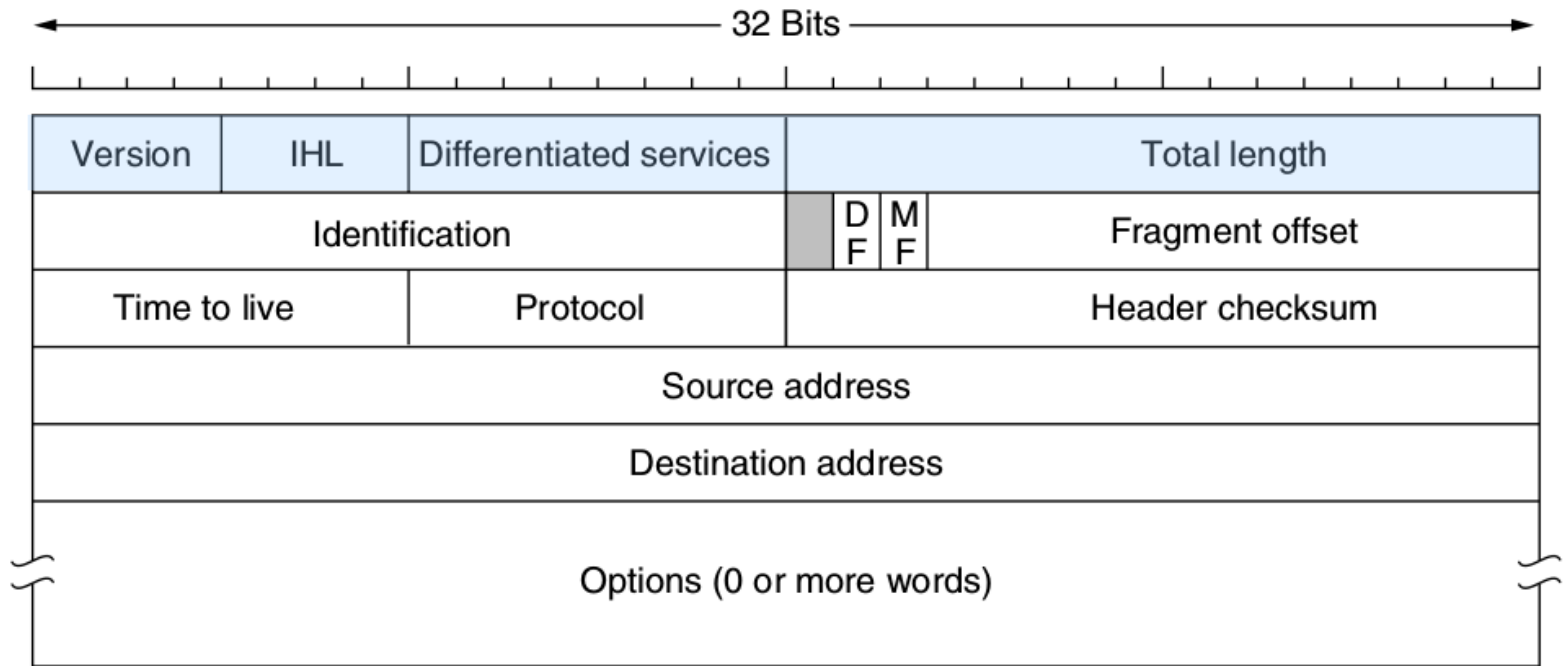
- **DHCP** (Dynamic Host Configuration Protokoll)
- Basiert auf dem älteren BOOTP-Protokoll und ist zu diesem (eingeschränkt) kompatibel
- Realisiert als Application Protokoll über UDP Port 67
- Ermöglicht die Konfiguration über Subsystemgrenzen (DHCP Relay Agents) und mit Szenarien mit mehreren DHCP-Servern
- Erlaubt das Konfigurieren aller wichtigen Parameter (IP/Mask/default Gateway/DNS Server)
- Erlaubt das zeitlich u.A. beschränkte Vergeben von IP-Konfigurationen (leases) und das Steuern der Vergabe
- Sicherheitsprobleme: MAC-Spoofing, DHCP Starvation etc.

Anwendung	DHCP				
Transport	UDP				
Internet	IP (IPv4, IPv6)				
Netzzugang	Ethernet	Token Bus	Token Ring	FDDI	...

Wie wird IP datentechnisch repräsentiert?

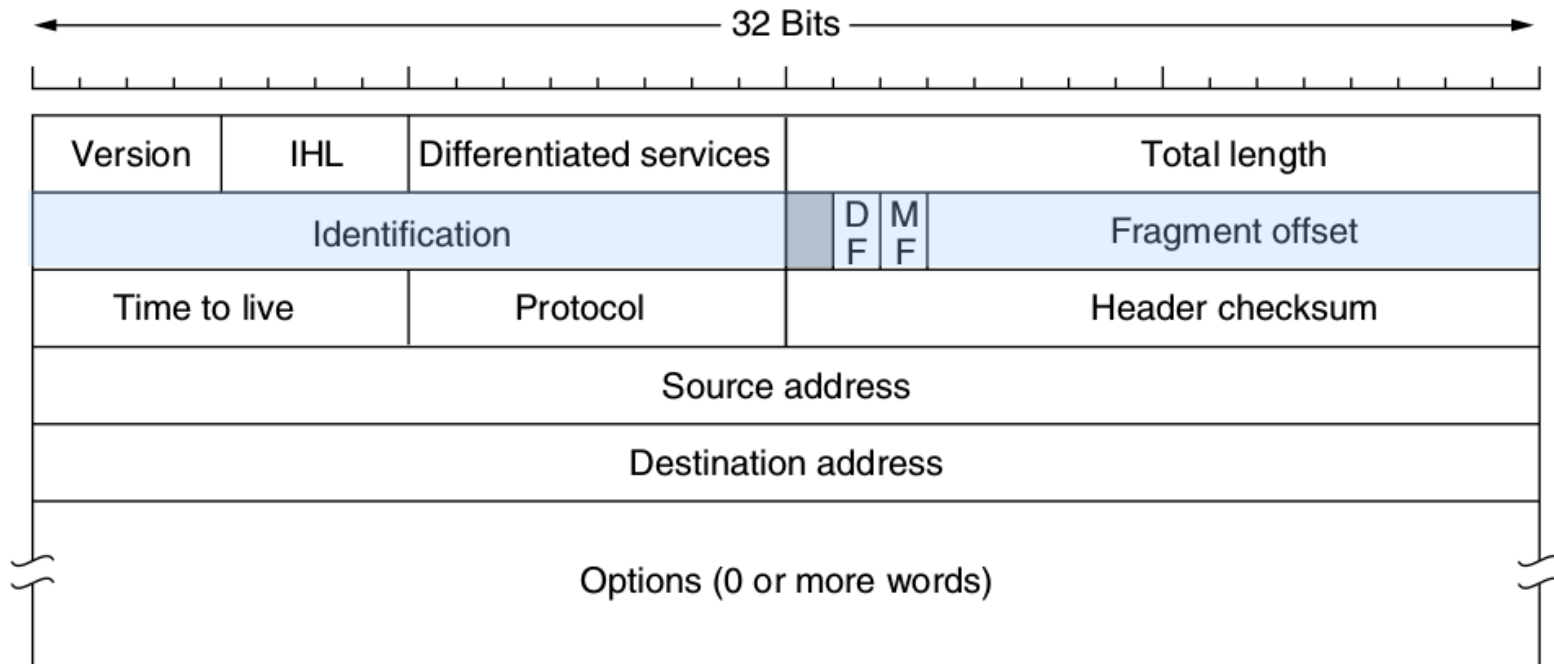
→ **IP Header**

Der IP-Header



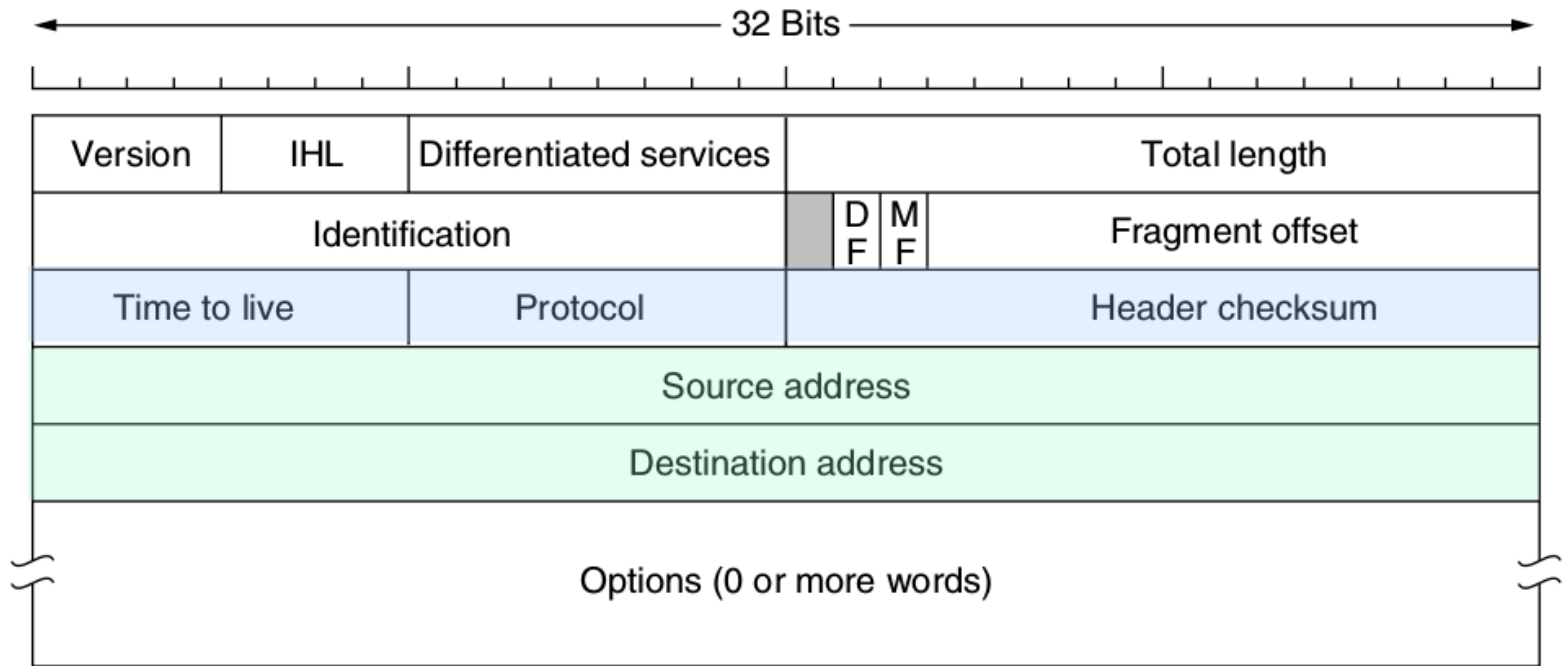
- Version: 4 / 6
- IHL: Länge des Headers mit $N * 32\text{Bit}$
- Diff. Services: Delay / Vorrang etc.
- Total length: Länge des **gesamten** Paketes mit Kopf

Der IP-Header



- Identification: Eindeutige **Kennung** des Datagramms
- DF: **D**on't **F**ragment
- MF: **M**ore **F**ragments
- Fragment offset: Offset im Payload **N** * 8 Byte

Der IP-Header



- Time to live (TTL): max. 255, Dekrementiert, 0→delete
- Protocol: Transportiertes Protokoll (TCP/UDP/ICMP/BGP ...)
- Header checksum: Einfache Prüfsumme
- Source address: Quell-**IP**
- Destination address: Ziel-**IP**

Probleme mit der Paketgröße

- Ein IP-Datagramm ist offensichtlich bis zu 64kB groß
- Auf Layer 2 gibt es in der Regel eine

MTU (**M**aximum **T**ransfer **U**nit)

z.B.

Ethernet: 1500 Bytes

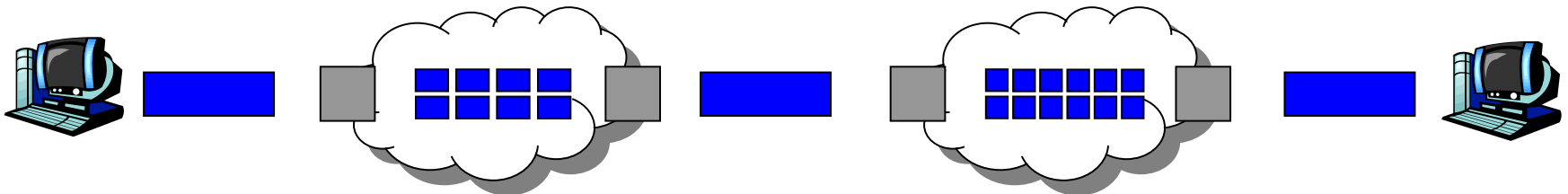
IEEE 802.11 (WLAN): 2272 Bytes

Mögliche Lösungen:

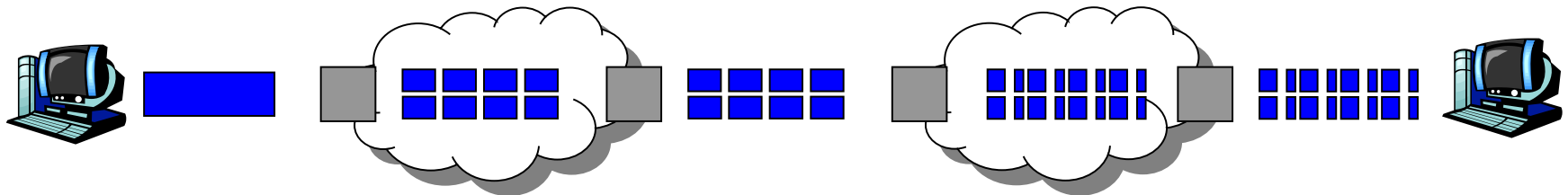
- **Fragmentierung**
- **Path MTU discovery**
- **Jumbo frames**

Fragmentierung

- ‚Zerhacken‘ eines großen IP-Datagramms in mehrere Teile
- Im Internet wird nur eine Vorgabe über die Mindeststrahmenlänge von 576 Bytes gemacht.
- Zerhacken ist einfach, aber wo wird das Datagramm wieder zusammengesetzt?



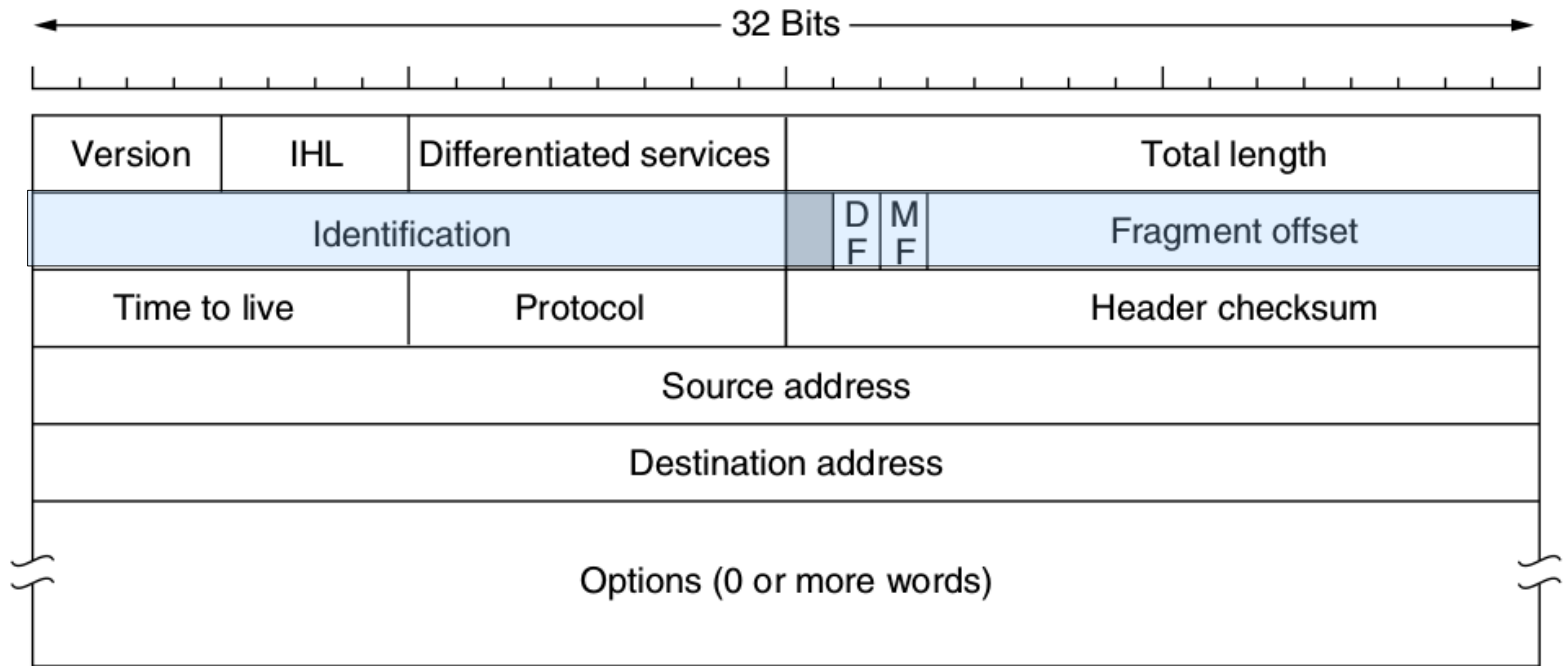
Transparente Fragmentierung



Nicht transparente Fragmentierung (Internet)

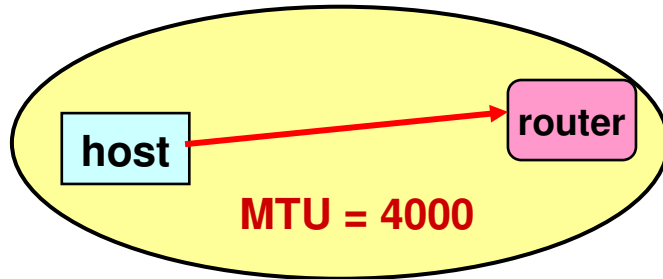
- Im Zielsystem (oder bei den Zwischenstationen) muss aus den Fragmenten wieder die ursprüngliche Dateneinheit hergestellt werden (**reassembly**).
 - Wenn nicht alle Fragmente eines Datagramms das Zielsystem erreichen, muss das gesamte Datagramm von der Quellstation aus wiederholt werden.
 - Fragmente können in unterschiedlicher Reihenfolge beim Zielsystem ankommen
- Empfänger braucht einen 64kB Buffer, in den er die ankommenden Fragmente einsortiert ...

Der IP-Header



- Identification: Eindeutige **Kennung** des Datagramms
- DF: **D**on't **F**ragment
- MF: **M**ore **F**ragments
- Fragment offset: Offset im Payload **N** * 8 Byte

IP-Fragmentierung Beispiel #1



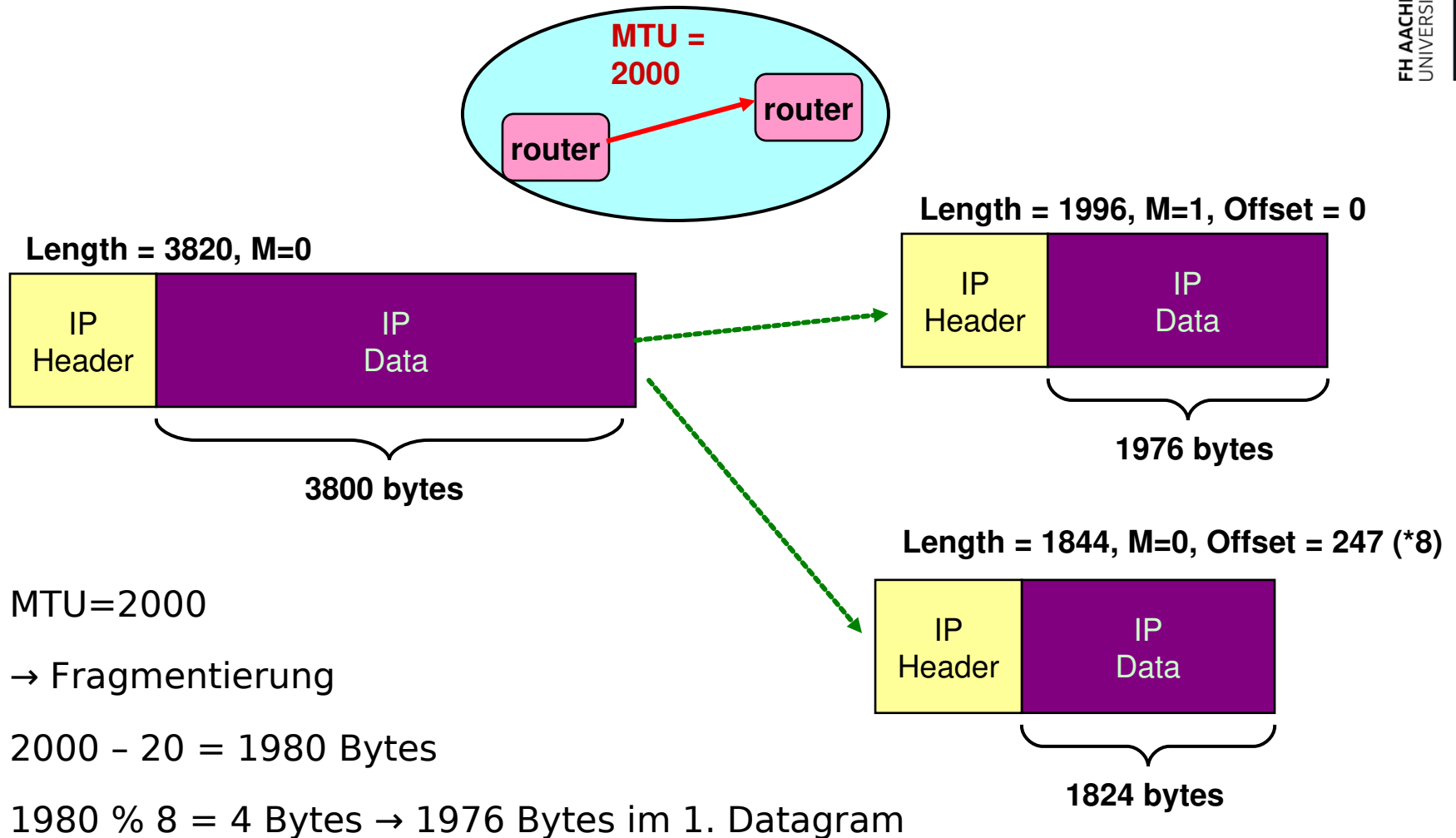
- Es sollen 3800 Bytes übertragen werden
- MTU=4000
- → keine Fragmentierung

Length = 3820, M=0

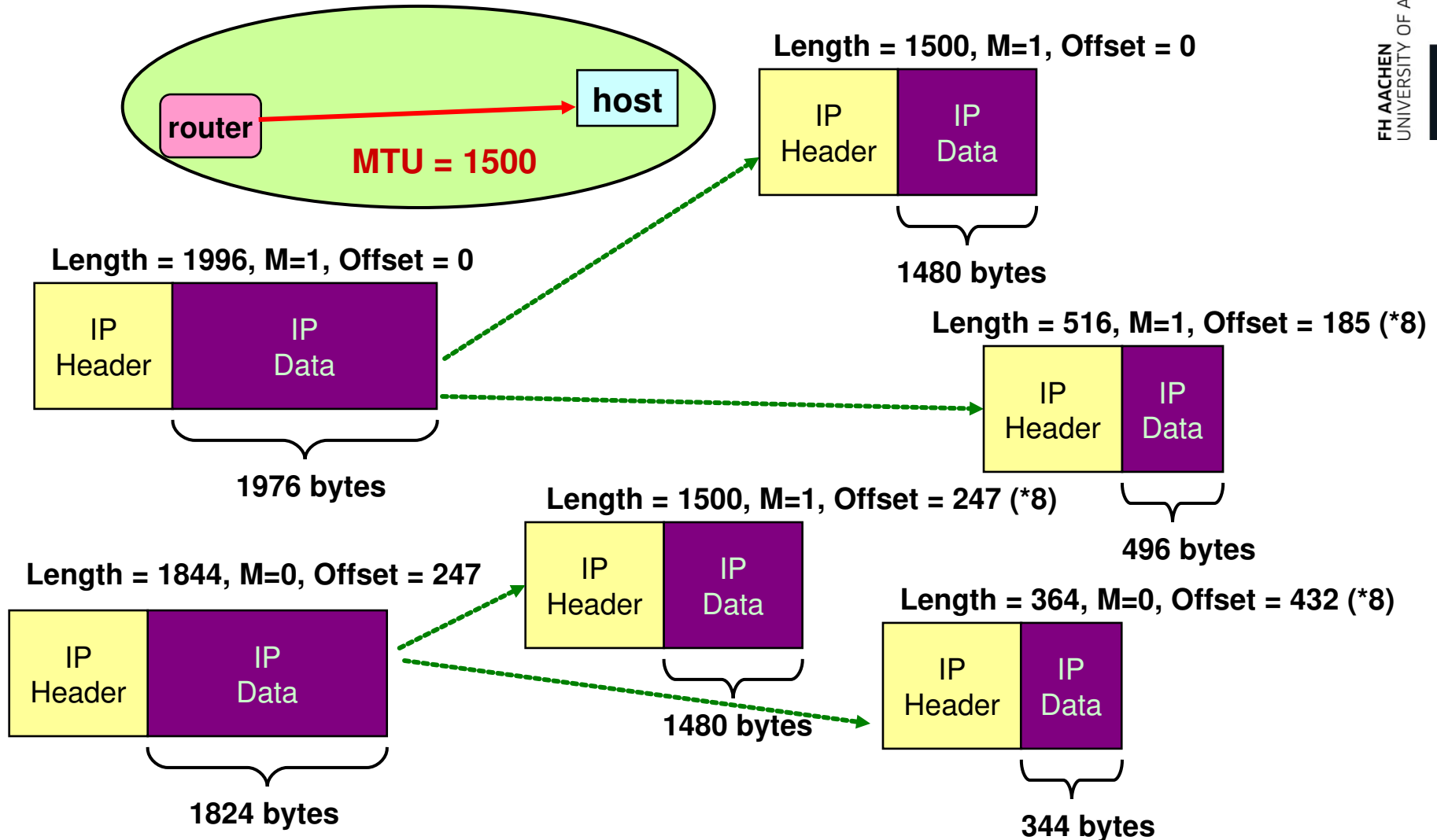


M : ‚More Fragments‘ - Flagge

IP-Fragmentierung Beispiel #2



IP-Fragmentierung Beispiel #3

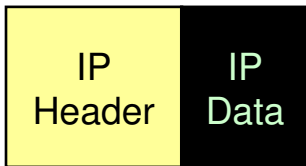


Zusammensetzung der Fragmente

Length = 1500, M=1, Offset = 0



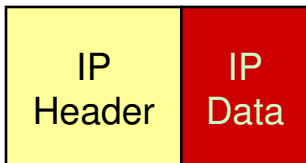
Length = 520, M=1, Offset = 185



Length = 1500, M=1, Offset = 247



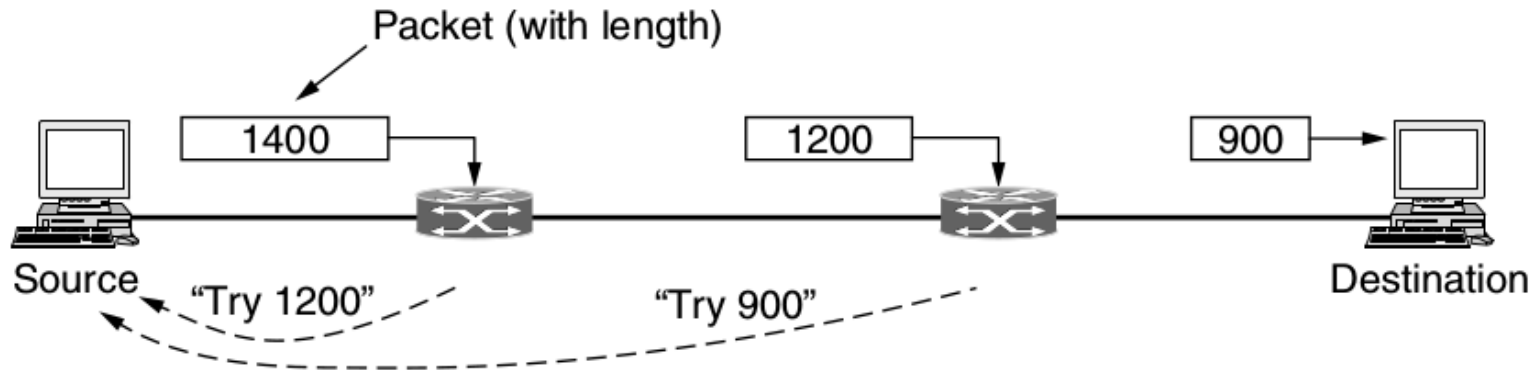
Length = 360, M=0, Offset = 432



- Zusammensetzen möglich durch Fragment Offset
- Fragmente können out-of-order ankommen
- Es ist unklar wieviel Speicher verwendet werden muss (max 64 kB)
- MF = 0 kann auch durch Misordered Packets vorab kommen
- Fragmente können dupliziert werden
- Identifikation und Filtern
- Einige Fragmente können nie ankommen
→ irgendwann muss aufgegeben werden



Path MTU Discovery



- Das Quell-System schickt die Pakete mit der Flagge DF (don't fragment)
- Wenn ein Router eine zu kleine MTU erkennt, wird das Paket verworfen und eine **ICMP**-Nachricht zurück gesendet
- Das Quell-System kann jetzt kleinere Pakete erzeugen, die weiter geleitet werden können
- Der Prozess muss ggf. mehrmals wiederholt werden!

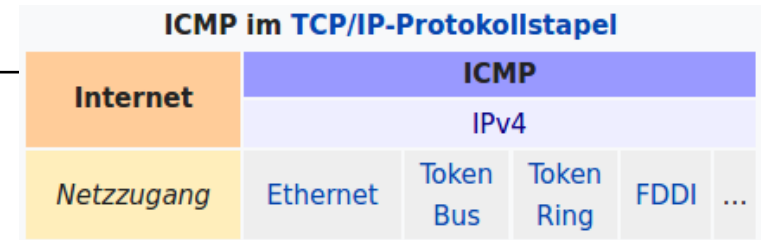
Jumbo Frames

- Auf Layer 2 können größere MTUs verwendet werden
- Für standard Ethernet existiert ein (eher selten unterstützter) Standard für eine MTU von 9000 Byte
- Problem für IP Datagramme > 9000 Byte besteht weiter...

Frame-level bandwidth efficiency										
Frame type	MTU	Layer 1 overhead		Layer 2 overhead		Layer 3 overhead	Layer 4 overhead	Payload size	Total transmitted ^[A]	Efficiency ^[B]
Standard	1500	preamble 8 byte	IPG 12 byte	frame header 14 byte	FCS 4 byte	IPv4 header 20 byte	TCP header 20 byte	1460 byte	1538 byte	94.93%
Jumbo	9000	preamble 8 byte	IPG 12 byte	frame header 14 byte	FCS 4 byte	IPv4 header 20 byte	TCP header 20 byte	8960 byte	9038 byte	99.14%
Other frame sizes for reference										
IEEE 802.11 ^{[14][15]}	7935	PLCP preamble & header 24 byte	IPG <i>varies</i>	frame header & security ovhd 52 byte	FCS 4 byte	IPv4 header 20 byte	TCP header 20 byte	7895 byte	8015 + IPG size byte	< 98.5%
IEEE 802.11 bridged to Ethernet	1500	PLCP preamble & header 24 byte	IPG <i>varies</i>	frame header & security ovhd 52 byte	FCS 4 byte	IPv4 header 20 byte	TCP header 20 byte	1460 byte	1580 + IPG size byte	< 92.4%

1500 Bytes ohne Layer1/2 header

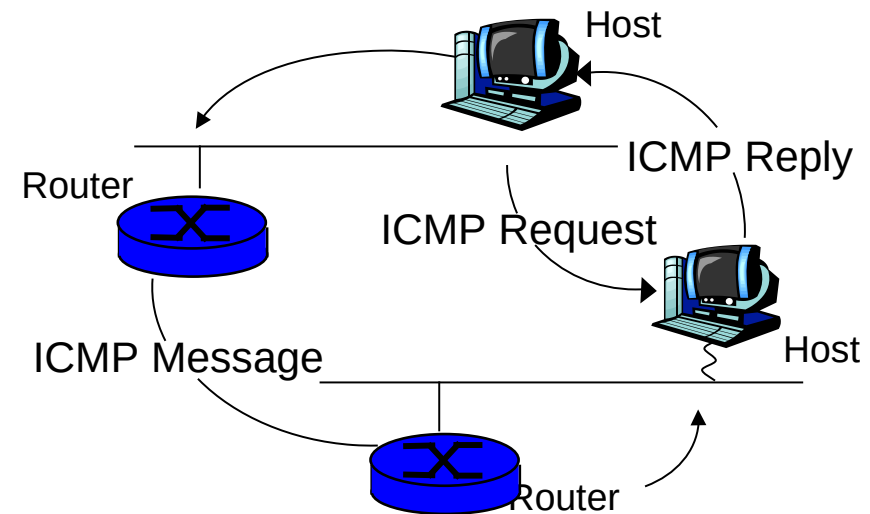
ICMP



- **I**nternet **C**ontrol **M**essage **P**rotocol
- ICMP ist ein Steuerprotokoll der Schicht 3, **welches auf IP aufbaut!** Dieses Protokoll wird z.B. von Routern verwendet, wenn etwas Unerwartetes passiert.

- Aufgaben:

- Mitteilung von Problemen beim Paketversand
- Echo-Anfragen (existiert der Zielknoten?)
- Unterstützung von höheren Protokollen und Anwendungen (z.B. Path MTU discovery, traceroute, ...)



ICMP - Header

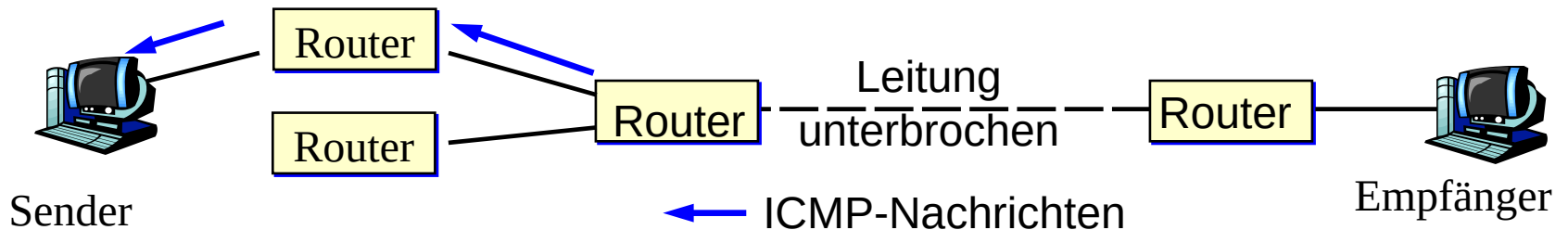
- ICMP versendet Fehler- und Kontrollnachrichten auf Netzebene. Diese Nachrichten werden in ein IP-Paket verpackt
- ICMP-Nachrichtenformat (Auszug!):

IP Header (Protocol = 1)		
Type	Code	Checksum
Identifier		Sequence Number
Optional Data		

Typ	Typname	Code	Bedeutung
0	Echo-Antwort	0	Echo-Antwort
3	Ziel nicht erreichbar	0	Netzwerk nicht erreichbar
		1	Host (Zielstation) nicht erreichbar
		2	Protokoll nicht erreichbar
		3	Port nicht erreichbar
		4	Fragmentierung nötig, Don't Fragment aber gesetzt
		5	Route nicht möglich (die Richtung in IP-Header-Feld Option falsch angegeben)
		13	Communication administratively prohibited (Paket wird von der Firewall des Empfängers geblockt)
4	Entlasten der Quelle	0	Datagramm verworfen, da Warteschlange voll
8	Echo-Anfrage	0	Echo-Anfrage (besser bekannt als „Ping“)
11	Zeitlimit überschritten	0	TTL (Time To Live, Lebensdauer) abgelaufen
		1	Zeitlimit während der Defragmentierung überschritten

Steuerung von IP: ICMP

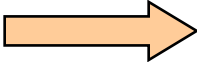
- IP ist nur für den (unzuverlässigen) Datenaustausch zuständig.



- **Nachrichtentypen, Beispiele:**

- **Destination Unreachable:** Ziel nicht erreichbar.
- **Time Exceeded:** Time-to-Live-Feld eines Pakets ist abgelaufen.
- **Echo Request / Reply:** Echo Reply wird angefordert ("ping").
- **Timestamp Request / Reply:** Ähnlich Echo Request. Zusätzlich Zeitstempel mit Ankunftszeit der Anfrage/Sendezeit der Antwort.

IPv6

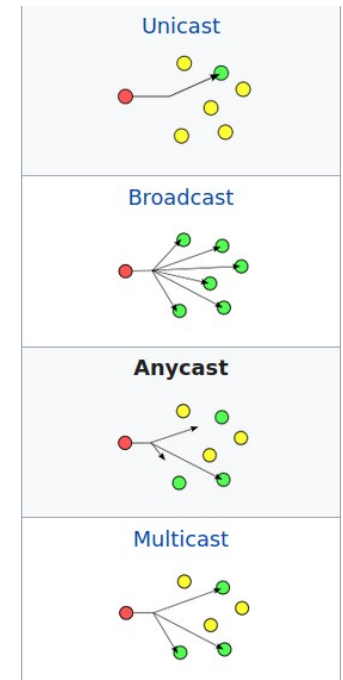
IPv4 (September 1981)  IPv6 (seit Dezember 1995)

Warum ein Wechsel, wenn IPv4 gut funktioniert?

- Dramatisch anwachsender Bedarf für neue IP-Adressen
- Vereinfachung des Protokolls, um eine schnellere Abarbeitung zu gewährleisten
- Sicherheitsmechanismen (Authentifikation und Datenschutz)
- Mehr Gewicht für Dienstarten, insbesondere für Echtzeitanwendungen
- Unterstützung von Mobilität (Hosts können ohne Adressänderung auf Reisen gehen)
- Möglichkeiten zur Fortentwicklung des Protokolls

IPv6 – Auswahl der Eigenschaften

- Adressgröße
 - 128-Bit-Adressen (8 Gruppen zu je 4 Hexadezimal-Zahlen)
- Verbesserter Optionsmechanismus / Einfacher Header
 - Vereinfacht und beschleunigt die Verarbeitung von IPv6-Paketen für Router
 - IHL: überflüssig, keine Optionen mehr
 - Protocol, Fragmentierung: überflüssig, wird durch Optionen mit abgedeckt
 - Checksum: Handhabung durch Schicht 2 und 4
- Verbesserung der Adressflexibilität
 - Anycast Address: Erreiche irgendeinen von mehreren (alle haben die gleiche Adresse)
- Unterstützung der Reservierung von Ressourcen
 - Erkennen von Datenströmen in IP (FlowLabel)
- Sicherheitsmaßnahmen
 - Authentifizierung und Privacy



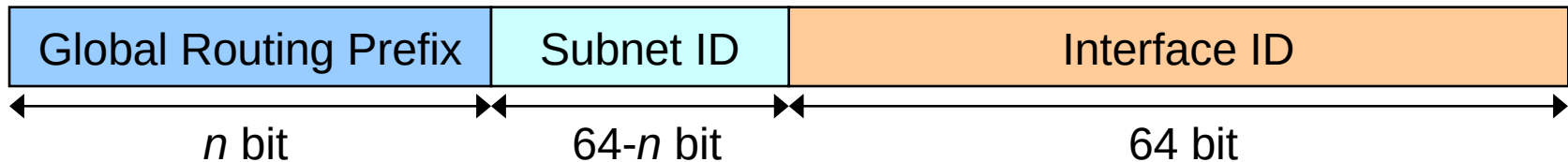
IPv6: Adressierung

IPv6-Adressen umfassen 128 Bit

- IPv6-Adressen werden in **hexadezimaler** Notation mit Doppelpunkten geschrieben: Format x:x:x:x:x:x:x:x, wobei jede Stelle 16 Bit hexadezimal kodiert
- Beispiele: 3FFE:400:20::A00:2BFF:FEA3:ADCB
 FF01:0:0:0:0:0:0:101 oder FF01::101
 FEDC:BA98:7600::/40 40 Bit langes Präfix für das Routing
- Unterscheidung von **Adressklassen**:
 - > Unicast-, Anycast- (one-to-nearest), Multicast-Adressen
- Typ einer Adresse wird durch *Präfix* (führende Bits) festgelegt:
 - > Loopback-Adresse: 00...01 (128 Bit) = ::1/128
 - > Multicast-Adresse: 11111111 = ff00::/8
 - > Link-local Unicast-Adresse (DHCP-Ersatz) 1111111010 = fe80::/10
 - > Site local Unicast-Adresse (private) 1111110 = fc00::/7
 - > IPv4-Adresse: 0....01...1 = 0:0:0:0:0:ffff:0:0/96
 - > Global Unicast-Adressen alles andere

Adressbeispiel

Global Unicast: dreigeteilte Hierarchie



- Globales Routing-Präfix zur Reduktion des Umfangs von Routing-Tabellen (z.B. geographischer Identifier/Identifier pro Provider)
- Subnet-ID als Adresse eines bestimmten Netzes
- Interface-ID als eindeutige Adresse eines Rechners, automatisch generiert, z.B. aus der MAC-Adresse:

> MAC-Adresse 00:1D:72:8E:74:6C (48 bit)

→ Interface ID: 00:1D:72:FF:FF:8E:74:6C (64 bit)

IPv4-Mapped IPv6 Adressen

- IPv4-Mapped-Adressen ermöglicht die Kommunikation mit End-Systemen, die nur IPv4 können
- Die IPv6 basiert vollständig auf der IPv4 Adresse

0000 . . . 0000

80 bits

FFFF

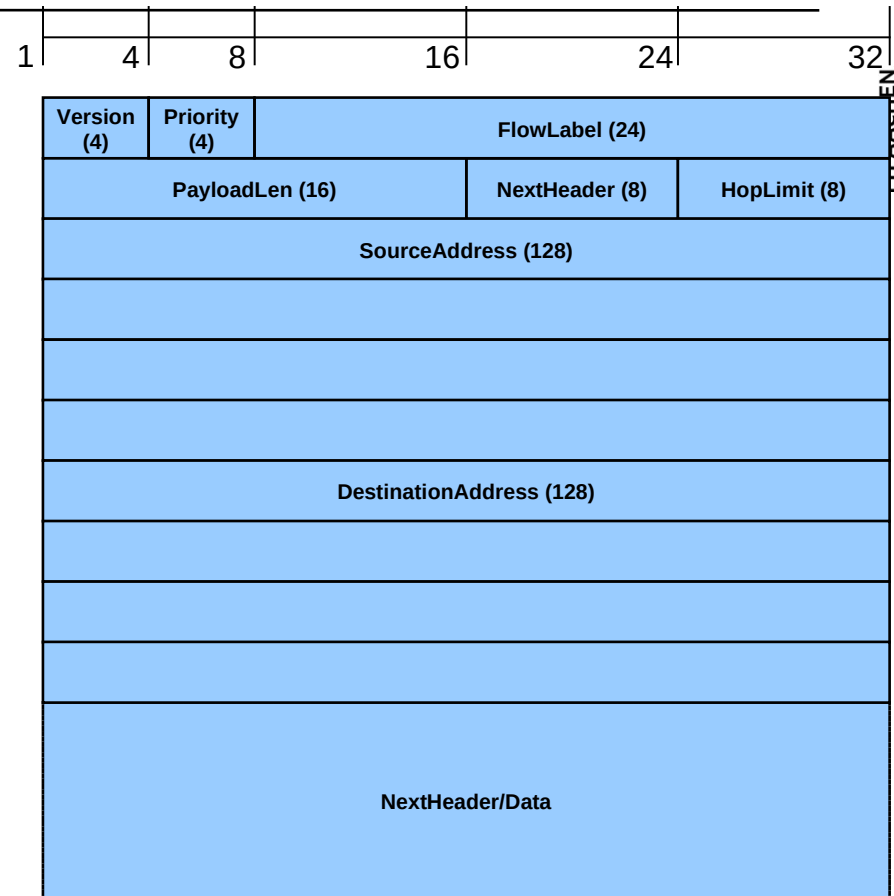
16 bits

IPv4 Address

32 bits

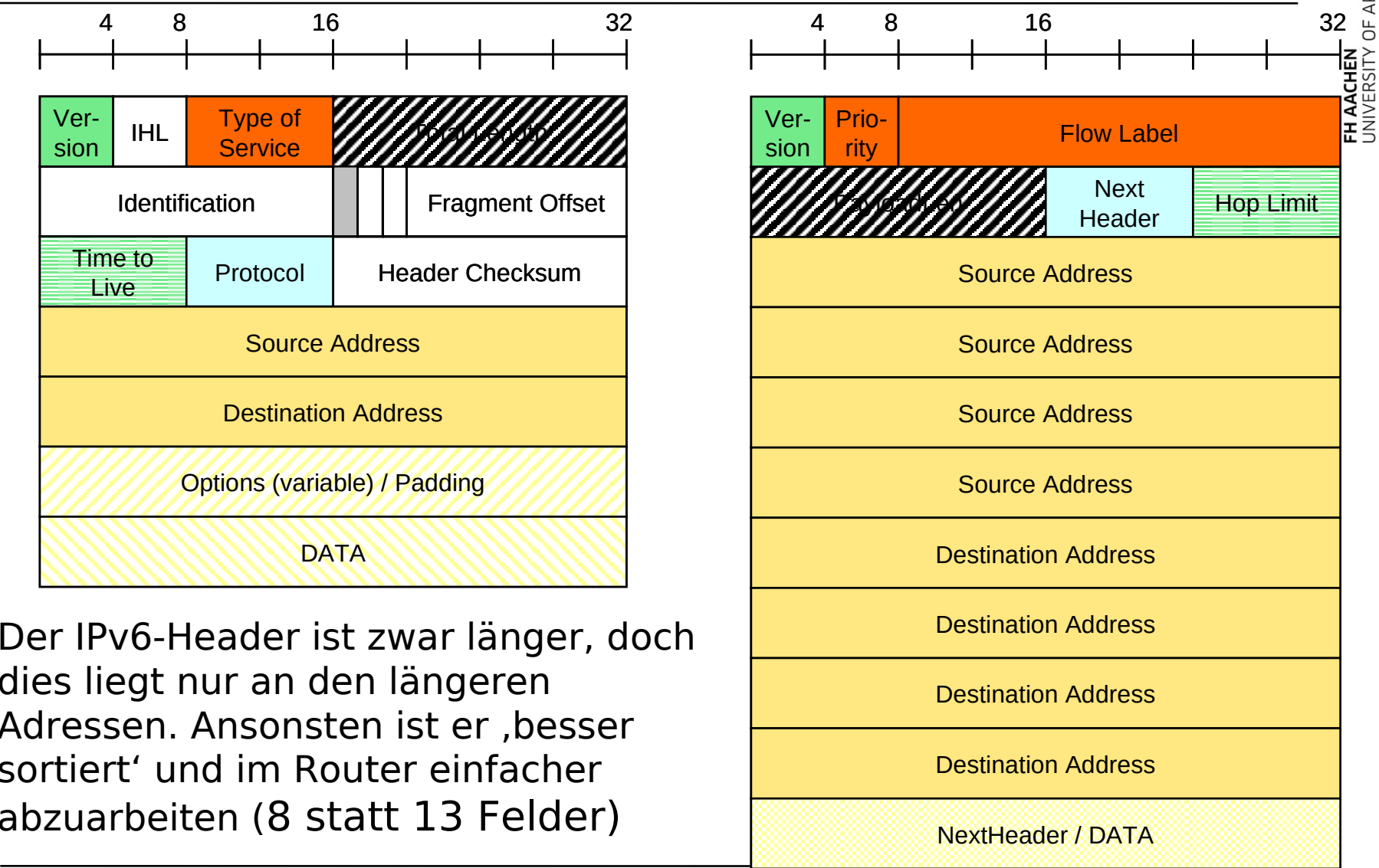
IPv6 Haupt-Header

- **Version:** IP Version Nummer.
- **Priority:** 4 Bit für Priorität. 1 - News, 4 - FTP, 6 - Telnet, 8 bis 15 - Echtzeitverkehr.
- **FlowLabel:** virtuelle Verbindung mit bestimmten Merkmalen/Anforderungen
- **PayloadLen:** Paketlänge nach dem 40-Byte-Header (also ohne Header)
- **NextHeader:** 8-Bit-Selektor. Gibt den Typ des folgenden Erweiterungs-Headers an (oder den Transport-Header)
- **HopLimit:** Wird bei jedem Knoten dekrementiert. Bei Null wird das Paket verworfen
- **SourceAddress:** Die Adresse des ursprünglichen Senders des Pakets
- **DestinationAddress:** Die Adresse des Empfängers (nicht unbedingt das endgültige Ziel, wenn es einen Optional Routing Header gibt)



Das Präfix einer Adresse charakterisiert geographische Bereiche, Provider, lokale interne Bereiche, ...

IPv4 vs. IPv6: Header



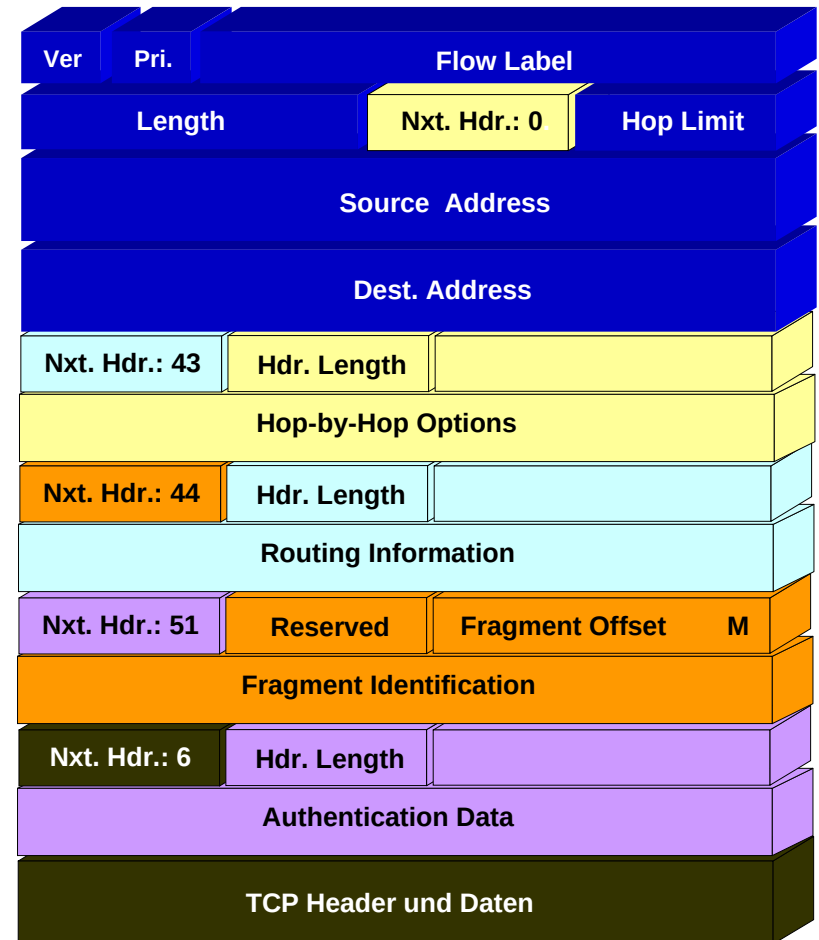
Optionale Angaben folgen in *Erweiterungs-Headern*. Davon sind 6 definiert:

- **Hop-by-Hop** (Informationen für Teilstrecken)
Alle Router müssen dieses Feld prüfen. Momentan definiert ist nur die Unterstützung von Jumbogrammen, d.h. Paketen mit Überlänge (Hierbei wird eine Längenangabe eingetragen).
- **Routing** (Definition einer vollen oder teilweise festgelegten Route)
- **Fragmentierung** (Verwaltung von Fragmenten)
Unterschied zu IPv4: Nur die Quelle kann eine Fragmentierung vornehmen. Router, für die ein Paket zu groß ist, schicken eine Fehlermeldung an die Quelle.
- **Authentifikation** (des Senders)
- **Verschlüsselte Sicherheitsdaten** (Informationen zur Verschlüsselung der Daten)
- **Zieloptionen** (Zusatzinformationen für das Ziel)

Der IPv6-Header ist erweiterbar

Nutzung der Erweiterungs-Header

- Per Hop ausgewertete Header
 - Hop-by-Hop Options (z.B. Jumbogramm Notifier)
 - Routing Information Header
- Nur im Endsystem ausgewertete Header
 - Fragmentation Header
 - Authentication Header
- Header-Extensions u.U. auf Applikationsniveau direkt nutzbar



FH Aachen
Fachbereich 9 Medizintechnik und Technomathematik
Prof. Dr.-Ing. Andreas Terstegge
Straße Nr.
PLZ Ort
T +49. 241. 6009 53813
F +49. 241. 6009 53119
Terstegge@fh-aachen.de
www.fh-aachen.de