





# Lernziele dieses Foliensatzes

- Vermittlungsschicht (Teil 1)
  - Geräte der Vermittlungsschicht
    - Router
    - Auswirkungen auf die Kollisionsdomäne
    - Broadcast-Domäne (Rundsendedomäne)
  - Adressierung in der Vermittlungsschicht
    - Aufbau von IP-Adressen
    - Netzklassen, Netzwerkteil und Geräteteil, Subnetze und Netzmaske
    - Private IP-Adressen
    - Aufbau von IP-Paketen
    - Fragmentieren von IP-Paketen
  - Diagnose und Fehlermeldungen mit ICMP







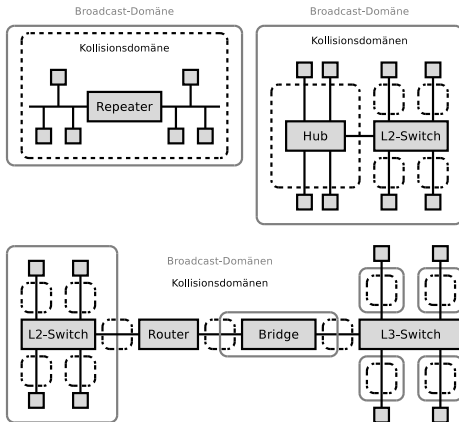






# Broadcast-Domäne – Rundsendedomäne (2/2)

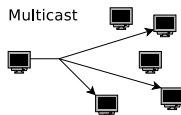
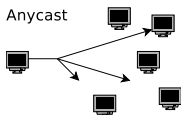
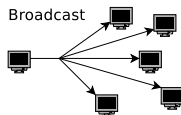
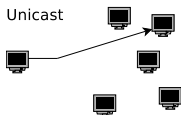
- Broadcast-Domänen bestehen aus einer oder mehreren Kollisionsdomänen
- Router arbeiten auf der Vermittlungsschicht (Schicht 3)
  - Das heißt, an jedem Port eines Routers hängt ein anderes IP-Netz
    - Das ist wichtig, wenn man die Anzahl der nötigen Subnetze berechnen will
  - Man kann mehrere Hubs, Switche, Repeater oder Bridges in einem IP-Subnetz betreiben
    - Man kann aber nicht ein IP-Subnetz an mehreren Ports eines Routers betreiben





## Adressierung in der Vermittlungsschicht (2/2)

- Jedes IP-Paket enthält eine Empfängeradresse
  - Den Aufbau von IP-Adressen definiert das Internet Protocol (IP)



- Eine IP-Adresse kann einen einzelnen Empfänger (**Unicast**) oder eine Gruppe von Empfängern bezeichnen (**Multicast** oder **Broadcast**)
- Einem Netzwerkgerät können auch mehrere IP-Adressen zugeordnet sein
- Bei **Anycast** erreicht man über eine Adresse einen einzelnen Empfänger aus einer Gruppe
  - Es antwortet der Empfänger, der über die kürzeste Route erreichbar ist

Multicast verwenden zum Beispiel die Routing-Protokolle RIPv2 und OSPF (siehe Foliensatz 8) und das Network Time Protocol – NTP (siehe Foliensatz 10) zur Synchronisierung von Uhren  
 Anycast verwenden zum Beispiel einigen Root-Nameserver im Domain Name System (siehe Foliensatz 10)





# Netzklassen (1/2)

- Die Präfixe legen die Netzklassen und ihre Adressbereiche fest

Klasse	Präfix	Adressbereich	Netzteil	Hostteil
A	0	0.0.0.0 - 127.255.255.255	7 Bits	24 Bits
B	10	128.0.0.0 - 191.255.255.255	14 Bits	16 Bits
C	110	192.0.0.0 - 223.255.255.255	21 Bits	8 Bits
D	1110	224.0.0.0 - 239.255.255.255	—	—
E	1111	240.0.0.0 - 255.255.255.255	—	—

- $2^7 = 128$  Klasse A-Netze mit jeweils maximal  $2^{24} = 16.777.216$  Hostadressen
- $2^{14} = 16.384$  Klasse B-Netze mit jeweils maximal  $2^{16} = 65.536$  Hostadressen
- $2^{21} = 2.097.152$  Klasse C-Netze mit jeweils maximal  $2^8 = 256$  Hostadressen
- Klasse D enthält Multicast-Adressen (zum Beispiel für IPTV)
- Klasse E ist für zukünftige (?) Verwendungen und Experimente reserviert

Warum wird der Klasse E-Adressraum von IPv4 nicht verwendet?

*„The class E space has 268 million addresses and would give us in the order of 18 months worth of IPv4 address use. However, many TCP/IP stacks, such as the one in Windows, do not accept addresses from class E space and will not even communicate with correspondents holding those addresses. It is probably too late now to change this behavior on the installed base before the address space would be needed.“*

Quelle: [http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_10-3/103\\_addr-cons.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-3/103_addr-cons.html)

## Netzklassen (2/2)

- Praktisch relevant sind nur die Klassen A, B und C
- Ursprünglich war beabsichtigt, durch die Netzadresse physische Netze eindeutig zu identifizieren
  - Dieses Vorgehen bringt aber Nachteile mit sich
- **Nachteile der Netzklassen:**
  - Sie können nicht dynamisch an Veränderungen angepasst werden
  - Sie verschwenden viele Adressen
    - Ein Klasse C-Netz mit 2 Geräten verschwendet 253 Adressen
    - Bei Klasse C-Netzen kann der Adressraum rasch knapp werden
    - Ein Klasse B-Netz mit 256 Geräten verschwendet > 64.000 Adressen
    - Es gibt es nur 128 Klasse A-Netze
    - Migration vieler Geräte in eine andere Netzklasse ist aufwändig
- Lösung: Unterteilung logischer Netze in **Teilnetze (Subnetze)**
  - 1993: Einführung des klassenlosen Routings – **Classless Interdomain Routing (CIDR)**



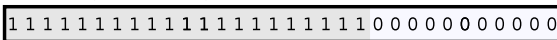


# Netzmaske (2/2)

IP-Adresse der Klasse B



Netzmaske (255.255.248.0)



Ein Teil der Hostadresse in der IP-Adresse definiert die Subnetznummer



- Aufbau der Netzmaske:
  - Einsen kennzeichnen den (Sub-)Netz-Nummernteil eines Adressraumes
  - Nullen kennzeichnen den Teil des Adressraumes, der für die Hostadressen zur Verfügung steht
- Um z.B. ein Klasse B-Netz in 20 Subnetze aufzuteilen, sind 5 Bits nötig
  - Jedes Subnetz braucht nämlich seine eigene Subnetznummer und diese muss binär dargestellt werden
  - Werden 5 Bits für die Darstellung der Subnetznummern verwendet, bleiben noch 11 Bits für den Hostteil

# Schreibweise des Classless Interdomain Routing (CIDR)

- Seit Einführung des **CIDR** 1993 werden IP-Adressbereiche in der Notation Anfangsadresse/Netzbits vergeben
  - Die Netzbits sind die Anzahl der Einsen in der Netzmaske
- Die Tabelle zeigt die möglichen Aufteilungen eines Klasse C-Netzes in Subnetze

Netzbits	/24	/25	/26	/27	/28	/29	/30	/31	/32
Netzmaske	0	128	192	224	240	248	252	254	255
Subnetzbits	0	1	2	3	4	5	6	7	8
Subnetze	1	2	4	8	16	32	64	128	256
Hostbits	8	7	6	5	4	3	2	1	0
Hostadressen	256	128	64	32	16	8	4	2	—
Hosts	254	126	62	30	14	6	2	0	—

# Nicht alle Adressen können/sollen verwendet werden

Netzbits	/24	/25	/26	/27	/28	/29	/30	/31	/32
Netzmaske	0	128	192	224	240	248	252	254	255
Subnetzbits	0	1	2	3	4	5	6	7	8
Subnetze	1	2	4	8	16	32	64	128	256
Hostbits	8	7	6	5	4	3	2	1	0
Hostadressen	256	128	64	32	16	8	4	2	—
Hosts	254	126	62	30	14	6	2	0	—

## 2 Hostadressen können nicht an Knoten vergeben werden, weil jedes (Sub-)Netzwerk benötigt. . .

- eine Adresse (**Netzdeskriptor**) für das Netz selbst (alle Bits im Hostteil = 0)
- eine Broadcast-Adresse, um alle Knoten im Netz zu adressieren (alle Bits im Hostteil = 1)

## 2 Subnetznummern sollen nicht verwendet werden

- Die Subnetznummern, die ausschließlich aus Nullen und ausschließlich aus Einsen bestehen, sollen nicht verwendet werden  $\implies$  diese Regel ist veraltet, wird aber häufig angewendet
- Moderne Router und Netzwerksoftware haben kein Problem damit, wenn alle möglichen Subnetznummern für existierende Subnetze vergeben werden

# Bestimmung der nötigen Bits für Subnetze

Netzbits	/24	/25	/26	/27	/28	/29	/30	/31	/32
Netzmaske	0	128	192	224	240	248	252	254	255
Subnetzbits	0	1	2	3	4	5	6	7	8
Subnetze	1	2	4	8	16	32	64	128	256
Hostbits	8	7	6	5	4	3	2	1	0
Hostadressen	256	128	64	32	16	8	4	2	—
Hosts	254	126	62	30	14	6	2	0	—

- Anhand der Tabelle ist es einfach, die nötigen Bits für Subnetze zu bestimmen
- Beispiel: Ein Klasse C-Netz soll in 5 Subnetze mit jeweils maximal 25 Hosts aufgeteilt werden
  - Jedes Subnetz benötigt eine Subnetznummer
  - Für 5 Subnetze sind 3 Subnetzbits nötig
  - Mit Hilfe der restlichen 5 Bits im Hostteil können in jedem Subnetz bis zu  $32 - 2 = 30$  Hosts adressiert werden
  - Somit ist die Schrägstrichdarstellung /27 geeignet

# Rechenbeispiel zu Subnetzen

- Beispiel: 172.21.240.90/27 ist eine Klasse B-Adresse ( $\implies$  siehe Präfix)
  - /27 = Anzahl der Einsen in der Netzmaske
- **IP-Adresse AND Netzmaske = Subnetzadresse**

1 AND 1 = 1, 1 AND 0 = 0, 0 AND 1 = 0, 0 AND 0 = 0

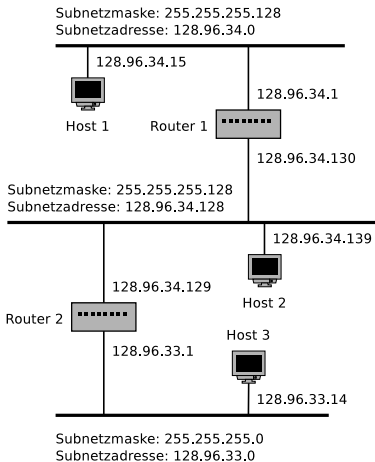
IP-Adresse	172.21.240.90	10101100	00010101	11110000	01011010
Netzmaske	255.255.255.224	11111111	11111111	11111111	11100000
Subnetzadresse	172.21.240.64	<del>10101100</del>	<del>00010101</del>	11110000	01000000
Subnetznummer	1922	<del>10101100</del>	<del>00010101</del>	11110000	01000000

- **IP-Adresse AND (NOT Netzmaske) = Hostadresse**

IP-Adresse	172.21.240.90	10101100	00010101	11110000	01011010
Netzmaske	255.255.255.224	11111111	11111111	11111111	11100000
negierte Netzmaske	000.000.000.31	<del>00000000</del>	<del>00000000</del>	<del>00000000</del>	00011111
Hostadresse	26	<del>00000000</del>	<del>00000000</del>	<del>00000000</del>	00011010

- /27 und Klasse B-Präfix  $\implies$  11 Bits für die Subnetznummer
  - Es verbleiben 5 Bits und damit  $2^5 = 32$  Adressen für den Hostteil
    - Davon sind 30 Hostadressen für Netzwerkgeräte verfügbar

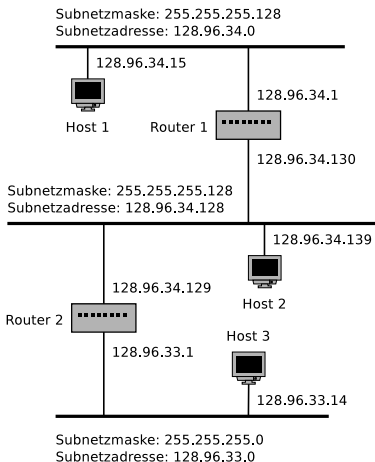
# Beispiel (1/4)



- Alle Hosts im gleichen Subnetz haben die gleiche Subnetzmaske
- $IP \text{ AND Subnetzmaske} = \text{Subnetzadresse}$
- Will ein Host ein Paket versenden, führt er ein AND zwischen der eigenen Subnetzmaske und der IP des Ziels durch
  - Stimmt das Ergebnis mit der Subnetzadresse des Senders überein, weiß er, dass das Ziel im gleichen Subnetz liegt
  - Ist das Ergebnis nicht gleich, muss das Paket an einen Router gesendet werden, der es an ein anderes Subnetz weiterleitet

Quelle: Computernetzwerke. Peterson und Davie. dpunkt (2000)

# Beispiel (2/4)



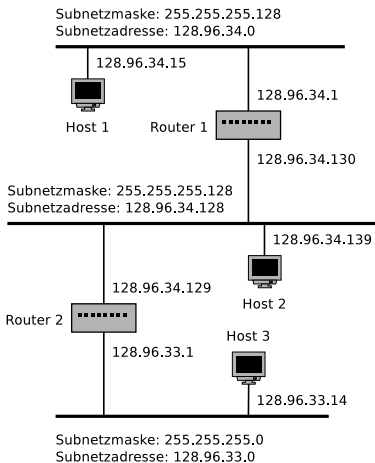
- Beispiel: Host 1 sendet ein Paket an Host 2 (128.96.34.139)
- Host 1 berechnet Subnetzmaske (255.255.255.128) AND Zieladresse (128.96.34.139) und erhält 128.96.34.128
- Das ist nicht die Subnetzadresse von Host 1  $\implies$  Host 2 ist in einem anderem Subnetz
- Host 1 übermittelt das Paket an seinen Standard-Router (128.96.34.1)
- Einträge in der Routing-Tabelle von Router 1

Subnetzadresse	Subnetzmaske	Nächster Hop
128.96.34.0	255.255.255.128	Port 0
128.96.34.128	255.255.255.128	Port 1
128.96.33.0	255.255.255.0	Router 2

- Routing-Protokolle/Algorithmen ( $\implies$  siehe Foliensatz 8) erstellen und pflegen die Einträge in den Routing-Tabellen der Router

Quelle: Computernetzwerke. Peterson und Davie. dpunkt (2000)

# Beispiel (3/4)



- Einträge in der Routing-Tabelle von Router 1

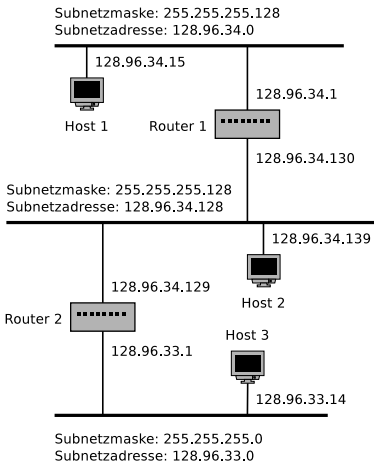
Subnetzadresse	Subnetzmaske	Nächster Hop
128.96.34.0	255.255.255.128	Port 0
128.96.34.128	255.255.255.128	Port 1
128.96.33.0	255.255.255.0	Router 2

- Der Router führt ein AND zwischen der Zieladresse und der Subnetzmaske jedes Eintrags durch
- Stimmt das Ergebnis mit der Subnetzadresse des Eintrags überein, leitet der Router das Paket an den Router oder Port weiter
- Router 1 berechnet für die 1. Zeile: Host 2 (128.96.34.139) AND Subnetzmaske (255.255.255.128) ist 128.96.34.128
- Das stimmt nicht mit der Subnetzadresse (128.96.34.0) überein

Quelle: Computernetzwerke. Peterson und Davie. dpunkt (2000)



# Beispiel (4/4)



- Einträge in der Routing-Tabelle von Router 1

Subnetzadresse	Subnetzmaske	Nächster Hop
128.96.34.0	255.255.255.128	Port 0
128.96.34.128	255.255.255.128	Port 1
128.96.33.0	255.255.255.0	Router 2

- Router 1 berechnet für die 2. Zeile: Host 2 (128.96.34.139) AND Subnetzmaske (255.255.255.128) ist 128.96.34.128
- Das stimmt mit der Subnetzadresse in der Routing-Tabelle überein  
⇒ Der 2. Tabelleneintrag ist ein Treffer
- Router 1 sendet das Paket über Port 1 an Host 2, weil der Port mit dem gleichen Netzwerk wie Host 2 verbunden ist

Wo kommen die Einträge in den Weiterleitungstabellen her?

Durch **Wegbestimmung (Routing)** werden die Weiterleitungstabellen mit **Routing-Protokollen** erstellt  
⇒ siehe Foliensatz 8

Quelle: Computernetzwerke. Peterson und Davie. dpunkt (2000)



# Aufbau von IPv4-Paketen (1/6)

- **Version** (4 Bits)

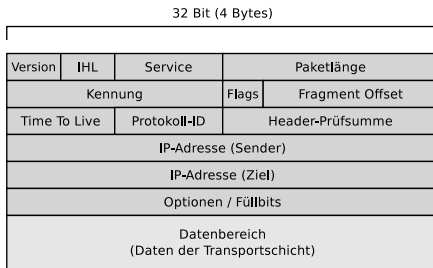
- Version des Protokolls
  - Version = 4  $\implies$  IPv4
  - Version = 6  $\implies$  IPv6

- **IHL** = IP Header Length (4 Bits)

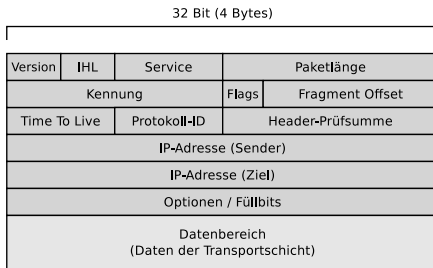
- Länge des IP-Headers in Vielfachen von 4 Bytes
  - Beispiel: IHL = 5  $\implies$  5 \* 4 Bytes = 20 Bytes
- Zeigt an, wo die Nutzdaten beginnen

- **Service** (8 Bits)

- Hiermit ist eine Priorisierung von IP-Paketen möglich (Quality of Service)
- Das Feld wurde mehrfach verändert (RFC 791, RFC 2474, RFC 3168)



# Aufbau von IPv4-Paketen (2/6)

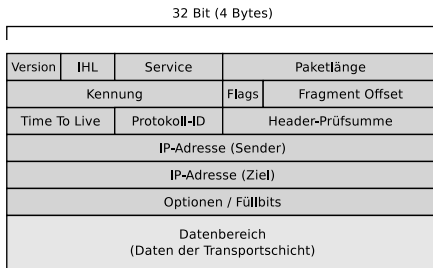


- **Paketlänge (16 Bits)**

- Länge des IP-Pakets (inkl. Header) in Bytes
- Das Feld ist 16 Bits groß  $\implies$  max. Paketlänge in IPv4: 65.535 Bytes

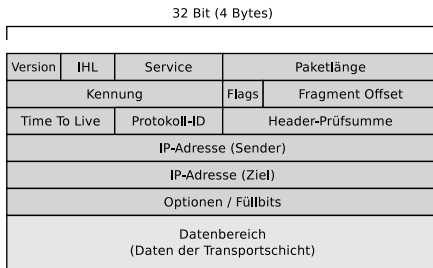
# Aufbau von IPv4-Paketen (3/6)

- Die Datenfelder **Kennung**, **Flags** und **Fragment Offset** steuern das Zusammensetzen fragmentierter IP-Pakete
- **Kennung** (16 Bits)
  - Eindeutige Kennung des IP-Pakets
- **Flags** (3 Bits)
  - Hier gibt der Sender an, ob das Paket fragmentiert werden darf und der Empfänger erfährt, ob noch weitere Fragmente folgen
- **Fragment Offset** (13 Bits)
  - Enthält eine Nummer, die bei fragmentierten Paketen besagt, ab welcher Position innerhalb des unfragmentierten Paketes das Fragment anfängt



Mehr Informationen über das Fragmentieren von IP-Paketen befinden sich auf den Folien 33 + 34

# Aufbau von IPv4-Paketen (4/6)



- **Time To Live (8 Bits)**

- Enthält die maximalen Hops
  - Jeder Router auf dem Weg zum Ziel verringert den Wert um eins
- Das verhindert, dass unzustellbare IP-Pakete endlos im Netz umherirren (kreisen)







# Fragmentieren (1/2)

- Das Zerlegen (und Zusammensetzen) von IP-Paketen in kleinere Pakete (**Fragmente**) heißt **Fragmentieren**
  - Wird in der Regel von Routern durchgeführt
    - Fragmentieren kann aber auch der Sender durchführen
- Grund für Fragmentieren:
  - Die maximale Paketlänge hängt von der Vernetzungstechnologie ab
- Die **Maximum Transmission Unit (MTU)** gibt an, wie viele Nutzdaten ein Rahmen haben darf, also wie groß ein Paket sein darf
  - MTU von Ethernet: meist 1.500 Bytes
    - Bei Gigabit Ethernet gibt es auch *Jumboframes* mit bis zu 9.000 Bytes
  - MTU von WLAN (IEEE 802.11): 2.312 Bytes
  - MTU von Token Ring mit 4 Mbit/s (IEEE 802.5): 4.464 Bytes
  - MTU von Token Ring mit 16 Mbit/s: 17.914 Bytes
  - MTU von PPPoE (z.B. DSL):  $\leq$  1.492 Bytes
  - MTU von ISDN: 576 Bytes
  - MTU von FDDI: 4.352 Bytes

## Fragmentieren (2/2)

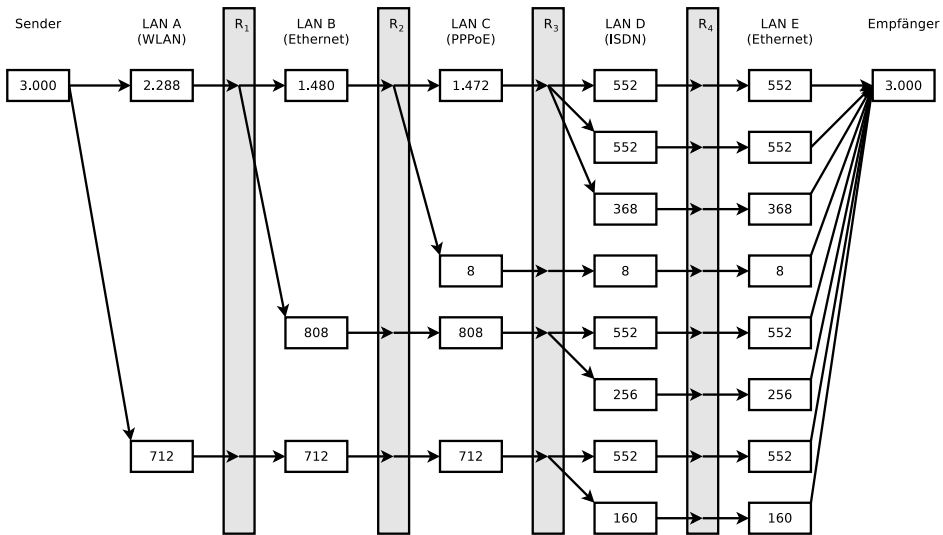
- In IP-Paketen gibt es ein Flag, mit dem das Fragmentieren untersagt werden kann
  - Müsste ein Router ein Paket fragmentieren, weil es für die Weiterleitung zu groß ist, aber die Fragmentierung ist im Paket untersagt, verwirft der Router das Paket, da er es nicht weiterleiten kann
- Netzwerkgeräte, die nicht alle Fragmente eines IP-Pakets innerhalb einer bestimmten Zeitspanne (wenige Sekunden) erhalten, verwerfen alle empfangenen Fragmente
- Router können IP-Pakete in kleinere Fragmente unterteilen, wenn die MTU es nötig macht und es in den Paketen nicht untersagt ist
  - **Kein Router kann aber Fragmente eines Pakets zu einem größeren Fragment zusammenfügen**
    - Nur der Empfänger kann Fragmente zusammenfügen







# Weiteres Beispiel zur Fragmentierung (2/2)





# Diagnose und Fehlermeldungen mit ICMP

- Das **Internet Control Message Protocol (ICMP)** ermöglicht den Austausch von...
  - Diagnosemeldungen
  - Steuernachrichten
  - Fehlermeldungen
- ICMP ist ein Bestandteil (*Partnerprotokoll*) von IPv4
  - Es wird aber wie ein eigenständiges Protokoll behandelt

Für IPv6 existiert mit ICMPv6 ein ähnliches Protokoll

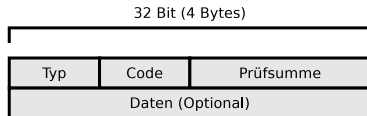
- Alle Router und Endgeräte können mit ICMP umgehen
- Typische Situationen, wo ICMP zum Einsatz kommt:
  - Ein Router verwirft ein IP-Paket, weil er nicht weiß, wie er es weiterleiten kann
  - Nur ein Fragment eines IP-Pakets kommt am Ziel an
  - Das Ziel eines IP-Pakets ist unerreichbar, weil die Time To Live (TTL) abgelaufen ist





# ICMP-Nachrichten

- Das Datenfeld **Typ** im ICMP-Header gibt den Nachrichtentyp an
- **Code** spezifiziert die Art der Nachricht innerhalb eines Nachrichtentyps
- Die Tabelle enthält einige Nachrichtentyp-Code-Kombinationen

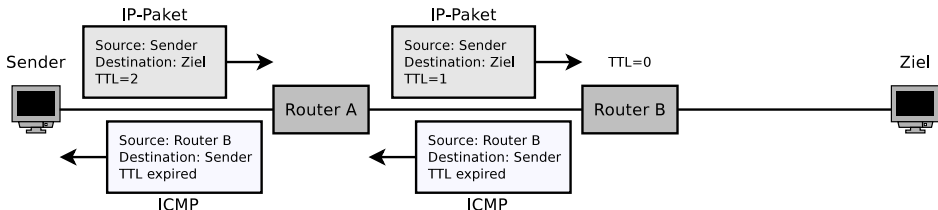


Typ	Typname	Code	Bedeutung
0	Echo-Antwort	0	Echo-Antwort (Antwort auf ping)
3	Ziel nicht erreichbar	0	Netz unerreichbar
		1	Ziel unerreichbar
		2	Protokoll nicht verfügbar
		3	Port nicht verfügbar
		4	Fragmentierung nötig, aber im IP-Paket untersagt
5	Umleitung (Redirect)	13	Firewall des Ziels blockt IP-Paket
		0	Router informiert über eine bessere Route (IP des ersten Hops) zum Zielnetz
8	Echo-Anfrage	1	Router informiert über eine bessere Route (IP des ersten Hops) zum Ziel-Host
		0	Echo-Anfrage (ping)
11	Zeitlimit überschritten	0	TTL (Time To Live) abgelaufen
		1	Zeitlimit während der Defragmentierung überschritten

Das ICMP-Protokoll enthält noch viele weitere Nachrichtentyp-Code-Kombinationen (siehe RFC 792), aber die meisten wurden in der Praxis selten oder nie verwendet und gelten als veraltet (siehe RFC 6633 und RFC 6918)



# Anwendungsbeispiel für ICMP: traceroute (2/3)

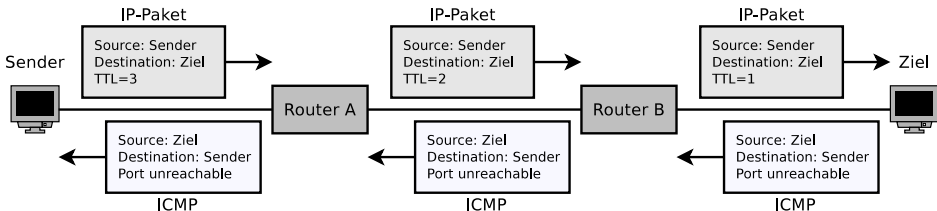


- Daraufhin schickt der Sender ein IP-Paket an den Empfänger mit TTL=2
- Das IP-Paket wird von Router A weitergeleitet
  - Dabei wird auch der Wert von TTL dekrementiert
- Router B empfängt das IP-Paket, setzt TTL=0, verwirft das IP-Paket und sendet eine ICMP-Nachricht vom Nachrichtentyp 11 und Code 0 an den Sender

**Achtung!** Es gibt verschiedene Implementierungen von traceroute

tracert unter Windows verwendet standardmäßig ICMP aber traceroute unter Linux und MacOSX verwendet standardmäßig UDP. Der Einsatz von ICMP kann aber via Kommandozeilenparameter `-I` erzwungen werden. Alternativ ist auch TCP möglich.

# Anwendungsbeispiel für ICMP: traceroute (3/3)



- Sobald der Wert von TTL groß genug ist, dass der Empfänger erreicht wird, sendet dieser eine ICMP-Nachricht vom Nachrichtentyp 3 und Code 3 an den Sender
- So kann der Sender via ICMP den Weg zum Empfänger nachvollziehen

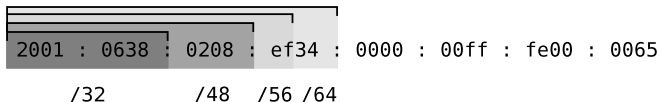
```
$ traceroute -q 1 wikipedia.de
traceroute to wikipedia.de (134.119.24.29), 30 hops max, 60 byte packets
 1 fritz.box (10.0.0.1) 1.834 ms
 2 p3e9bf6a1.dip0.t-ipconnect.de (62.155.246.161) 8.975 ms
 3 217.5.109.50 (217.5.109.50) 9.804 ms
 4 ae0.cr-polaris.fra1.bb.godaddy.com (80.157.204.146) 9.095 ms
 5 ae0.fra10-cr-antares.bb.gdinf.net (87.230.115.1) 11.711 ms
 6 ae2.cgn1-cr-nashira.bb.gdinf.net (87.230.114.4) 13.878 ms
 7 ae0.100.sr-jake.cgn1.dcnnet-emea.godaddy.com (87.230.114.222) 13.551 ms
 8 wikipedia.de (134.119.24.29) 15.150 ms
```

Das Beispiel sendet pro Hop nur ein Paket (-q 1). Darum enthält die Ausgabe pro Hop nur einen Round-Trip-Time-Wert (RTT). Standardmäßig sendet traceroute drei Pakete pro Hop, was die Anzeige von drei RTT-Werten zur Folge hat. Starke Schwankungen der RTT sind ein Indikator für Netzwerkprobleme oder Ressourcenengpässe



# Subnetze in IPv6

- (Sub-)Netzmasken gibt es bei IPv6 nicht
  - Die Unterteilung von Adressbereichen in Subnetze geschieht durch die Angabe der Präfixlänge
- IPv6-Netze werden in CIDR-Notation angegeben
  - Die Adresse eines einzelnen Geräts hat manchmal ein angehängtes /128
  - Ein Beispiel ist die Loopback-Adresse von IPv6: ::1/128
    - Alle Bits – außer das letzte Bit – haben den Wert 0  
(Bei IPv4 ist die Loopback-Adresse: 127.0.0.1)
  - Internetprovider (ISP) oder Betreiber großer Netze bekommen die ersten 32 oder 48 Bits von einer Regional Internet Registry (RIR) zugewiesen
    - Diesen Adressraum teilt der Provider oder Netzbetreiber in Subnetze auf
    - **Endkunden bekommen meist ein /64- oder sogar /56-Netz zugeteilt**



- Bekommt ein Endkunde ein /56-Netz zugeteilt, sind die 8 Bits zwischen dem Präfix und der Interface Identifier das **Subnet Präfix**

# IPv6-Adressen vereinfachen

- Regeln zur Vereinfachung (RFC 5952):
  - Führende Nullen innerhalb eines Blocks dürfen ausgelassen werden
  - Aufeinanderfolgende Blöcke, deren Wert 0 (bzw. 0000) ist, dürfen **innerhalb einer IPv6-Adresse genau 1x** ausgelassen werden
    - Das Auslassen wird durch 2 aufeinander folgende Doppelpunkte angezeigt
  - Gibt es mehrere Gruppen aus Null-Blöcken, ist es empfehlenswert die Gruppe mit den meisten Null-Blöcken zu kürzen
- Beispiele:
  - Die IPv6-Adresse von `j.root-servers.net` ist:  
2001:0503:0c27:0000:0000:0000:0002:0030  
⇒ 2001:503:c27::2:30

## Schreibweise von IPv6-Adressen (URLs)

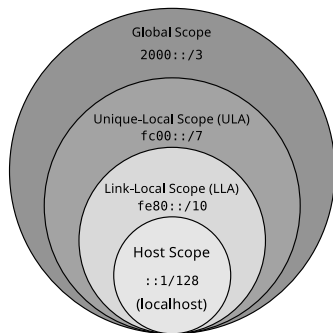
- IPv6-Adressen werden in eckigen Klammern eingeschlossen
- Portnummern werden außerhalb der Klammern angehängt  
`http://[2001:500:1::803f:235]:8080/`
- Das verhindert, dass die Portnummer als Teil der IPv6-Adresse interpretiert wird





# Gültigkeitsbereiche – Scopes (1/4)

- IPv6 unterscheidet nicht nur zwischen privaten und öffentlichen Adressen (wie IPv4), sondern auch mehrere Gültigkeitsbereiche (sog. *Scopes*)
- Jede IPv6-Adresse hat einen sogenannten Scope
- Der Gültigkeitsbereich ist der Teil eines Netzes, in dem die zugehörige Adresse als gültig betrachtet und weitergeleitet wird
- **Host Scope**: Loopback-Adresse
  - Die Loopback-Adresse ist  $::1/128 \implies 0:0:0:0:0:0:0:0:1/128$





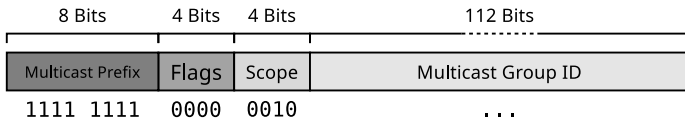






# IPv6-Multicast-Adressen (1/2)

- **IPv6 definiert keine Broadcast-Adressen**
  - **Multicast-Adressen emulieren die Broadcast-Funktionalität**



- Bei Multicast-Adressen haben die ersten 8 Bits den Wert 11111111
  - Somit haben sie das **Multicast-Präfix** ff::/8
- Auf das Präfix folgen 4 Bits **Flags** (1-Bit-Felder) und 4 Bits für den **Scope** (Gültigkeitsbereich)

Die Flags sind im Rahmen dieser Vorlesung irrelevant und haben in der Praxis meist den Wert 0

- In der Praxis relevante Gültigkeitsbereiche (Scopes) sind die Werte 1 und 2 (⇒ siehe nächste Folie)









# Stateless Address Autoconfiguration – Schritte

- In der Praxis arbeitet SLAAC in 5 Schritten
  - 1 Client erzeugt eine modifizierte EUI-64 Host-ID aus seiner eigenen MAC-Adresse
  - 2 Client erzeugt eine Link-Local-Adresse (LLA) mit Präfix  $fe80::/64$

Durch die Verkettung beider Teile entsteht eine 128-Bit-IPv6-Adresse', die im lokalen Netz gültig ist

- 3 Client sendet zur Erkennung doppelter Adressen **Duplicate Address Detection** (DAD) eine **Neighbor Solicitation** (NS)-Nachricht mit seiner eigenen Link-Local-Adresse über Multicast im Link-Local-Netz und wartet, ob eine **Neighbor Advertisement** (NA)-als Antwort eintrifft (siehe Folie 63)
- 4 Client sendet eine **Router Solicitation** (RS) Nachricht an den Router (siehe Folie 66)
- 5 Router sendet eine **Router Advertisement** (RA)-Nachricht an den Client und übergibt das global gültige Netzwerk-Präfix (siehe Folie 65)

# IPv6 Neighbor Discovery Protocol

- IPv6 implementiert keine Broadcast-Adressen und es gibt keine zum Address Resolution Protocol (ARP) vergleichbare Lösung
  - Allerdings ist die Auflösung von IPs in MAC-Adressen auch hier nötig
- **Das Neighbor Discovery Protocol (NDP) löst MACs aus IPv6-Adressen auf und nutzt dafür Multicast-Adressen**

In IPv6 beschreibt der Begriff **Neighbor** Knoten, die sich im gleichen Netzwerk der Sicherungsschicht (Data Link Layer) befinden

- NDP-Nachrichten werden als Nutzdaten in ICMPv6-Nachrichten ausgetauscht
- NDP implementiert 5 Arten von Nachrichten
  - **Router Solicitation** (ICMPv6 Typ 133)
  - **Router Advertisement** (ICMPv6 Typ 134)
  - **Neighbor Solicitation** (ICMPv6 Typ 135)
  - **Neighbor Advertisement** (ICMPv6 Typ 136)
  - **Redirect Message** (ICMPv6 Typ 137)

Mit der **Redirect Message** informiert ein Router über eine bessere Route (anderer First Hop  $\implies$  anderer lokaler Router) für ein Ziel. Dieser Nachrichtentyp wird in dieser Vorlesung nicht weiter behandelt















# SLAAC-Erweiterung: Stable Privacy (RFC 7217) – (1/3)

- **Optionale Erweiterung von SLAAC** (Stateless Address Autoconfiguration)
- Definiert die Adresserzeugung ohne Verwendung einer MAC-Adresse
  - Ein zufälliger geheimer Schlüssel wird erstellt und für die Generierung der Interface-ID verwendet
    - Der geheime Schlüssel ist eine 128-Bit lange hexadezimale Zeichenfolge, die aussieht wie eine IPv6-Adresse

## Speicherort des geheimen Schlüssels in Linux und erforderlicher Kernel-Parameter

Der stabile geheime Schlüssel ist in der Datei `/proc/sys/net/ipv6/conf/eth0/stable_secret` gespeichert und wird durch Setzen des Kernel-Parameters `addr_gen_mode=3` erzeugt

## Beispiel für einen geheimen Schlüssel

```
$ cat /proc/sys/net/ipv6/conf/eth0/stable_secret  
c8c8:036d:9312:71e2:eadc:7c9f:0535:649a
```

- Vorteile:
  - Verbesserte Sicherheit, da keine MAC-Adresse für die Erzeugung verwendet wird
    - Die MAC-Adresse des Knoten wird nicht preisgegeben  $\implies$  Anonymität
  - Stabile Adresse für den Knoten
    - Einmal generiert, ändert sich die Interface-ID nicht (bis zum Neustart)



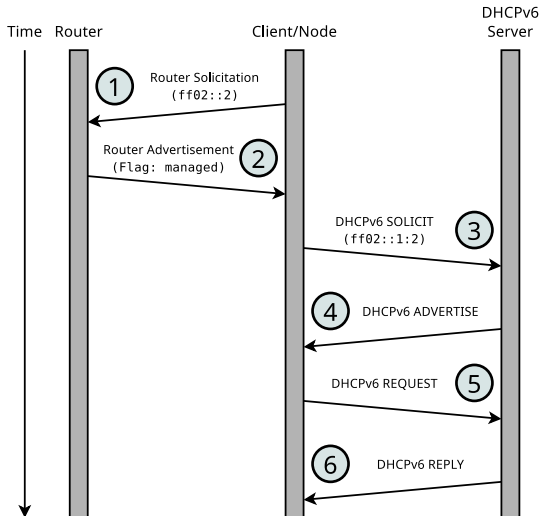






# DHCPv6 (RFC 8415) – (1/2)

- 1 Der Knoten fordert mit einer RS-Nachricht an die Multicast-Adresse `ff02::2` (alle Router) ein Präfix für eine global gültige Adresse an
- 2 Der Router antwortet mit einer RA-Nachricht, in der das Flag `managed` gesetzt ist
- 3 Der Knoten sendet eine Nachricht DHCPv6 SOLICIT an die Multicast-Adresse `ff02::1:2` (alle DHCPv6-Server)
- 4 Alle DHCPv6-Server in Reichweite antworten mit einer Nachricht DHCPv6 ADVERTISE, die eine Netzwerkkonfiguration enthält (DNS-Server, NTP-Server, ein Präfix für die global gültige Adresse, ...)

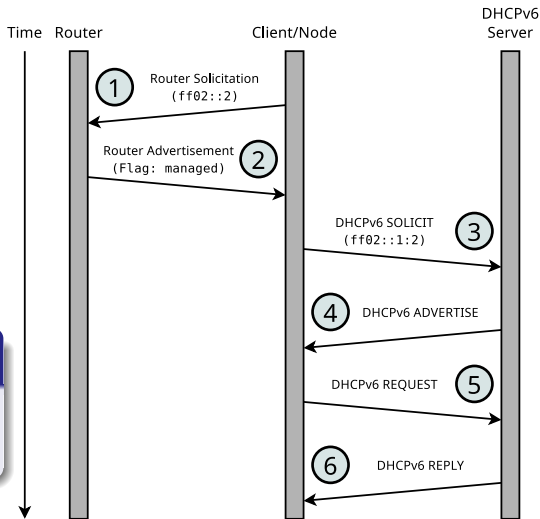


DHCPv6 ist die einzige **zustandsbehaftete** Möglichkeit der IPv6-Adresskonfiguration



# DHCPv6 (RFC 8415) – (2/2)

- 5 Der Knoten wählt ein Konfigurationsangebot aus und fordert es mit einer Nachricht DHCPv6 REQUEST an
- 6 Der DHCPv6-Server markiert die IP in seinem Adresspool mit der Client-ID als zugewiesen und quittiert die Anfrage mit einer Nachricht DHCPv6 REPLY



DHCPv6 und DHCP (für IPv4) sind beides Protokolle der Anwendungsschicht

DHCPv6 verwendet UDP über die Ports 547 (Server oder Relay Agent) und 546 (Client)



