

# 10. Foliensatz

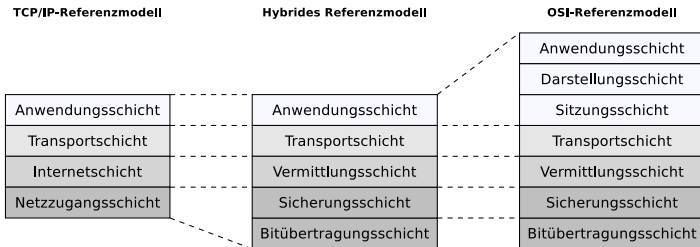
## Betriebssysteme und Rechnernetze

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences  
(1971–2014: Fachhochschule Frankfurt am Main)  
Fachbereich Informatik und Ingenieurwissenschaften  
[christianbaun@fb2.fra-uas.de](mailto:christianbaun@fb2.fra-uas.de)

# Vermittlungsschicht

- Aufgaben der Vermittlungsschicht (Network Layer):
  - Sender: Segmente der Transportschicht in Pakete unterteilen
  - Empfänger: Pakete in den Rahmen der Sicherungsschicht erkennen
  - Logische Adressen (IP-Adressen) bereitstellen
  - Routing: Ermittlung des besten Weges
  - Forwarding: Weiterleitung der Pakete zwischen logischen Netzen, also über physische Übertragungsabschnitte hinweg



Übungsblatt 10  
wiederholt die für  
die Lernziele  
relevanten Inhalte  
dieses Foliensatzes

- Geräte: Router, Layer-3-Switch (Router ohne WAN-Schnittstelle)
- Protokolle: IPv4, IPv6, ICMP, IPX/SPX, DECnet

# Sinnvolle Themen zur Vermittlungsschicht...

- ... und was aus Zeitgründen davon übrig bleibt...
  - Geräte der Vermittlungsschicht
    - Router
    - ~~Auswirkungen auf die Kollisionsdomäne~~
    - ~~Broadcast-Domäne (Rundsendedomäne)~~
  - Adressierung in der Vermittlungsschicht
    - IPv4
    - IPv6
    - ~~Fragmentieren von IP-Paketen~~
  - ~~Weiterleitung und Wegbestimmung~~
    - ~~Distanzvektor-Routing-Protokolle~~
    - ~~Link-State-Routing-Protokolle~~
  - Diagnose und Fehlermeldungen mit ICMP
  - Netzübergreifende Kommunikation  $\implies$  Internetworking (Zusammenfassung)
  - Network Address Translation (NAT)

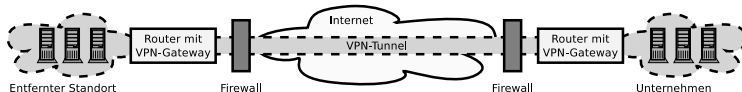




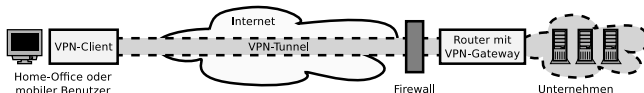
## Gateways (2/2)

- Auch VPN-Gateways (Virtual Private Network) können auf der Vermittlungsschicht arbeiten (z.B. via Protokoll IPSec)
  - Sie ermöglichen über unsichere öffentliche Netze den sicheren Zugriff auf entfernte sichere Netze (z.B. Hochschul-/Firmennetze)
  - Dienste (z.B. Email), die nur innerhalb des sicheren Netzes zur Verfügung stehen, werden über eine getunnelte Verbindung genutzt

### Site-to-Site VPN



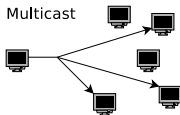
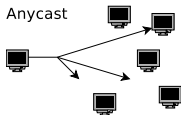
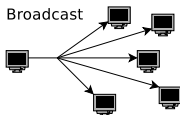
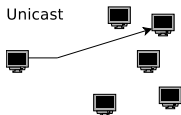
### Remote Access VPN bzw. End-to-Site VPN





## Adressierung in der Vermittlungsschicht mit IPv4 (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 und das Network Time Protocol (NTP) zur Synchronisierung von Uhren

Anycast verwenden zum Beispiel einigen Root-Nameserver im Domain Name System









## Netzklassen (2/2)

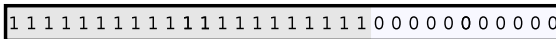
- Praktisch relevant sind nur die Klassen A, B und C
- Ursprünglich war beabsichtigt, durch die Netzadresse physische Netze eindeutig zu identifizieren
  - Dieses Vorgehen bringt aber Nachteile mit sich
- **Nachteile der Netzklassen:**
  - Sie können nicht dynamisch an Veränderungen angepasst werden
  - 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 (1/2)

IP-Adresse der Klasse B



Netzmaske (255.255.248.0)



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



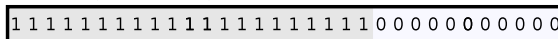
- Um Subnetze zu bilden, ist eine **(Sub-)Netzmaske** nötig
  - Alle Knoten in einem Netzwerk bekommen eine Netzmaske zugewiesen
    - Länge: 32 Bits (4 Bytes)
    - Mit ihr wird die Anzahl der Subnetze und Hosts festgelegt
- Die Netzmaske unterteilt die Hostadresse der IP-Adresse in **Subnetznummer** und **Hostadresse**
  - Die Netznummer bleibt unverändert
  - Die Netzmaske fügt eine weitere Hierarchieebene in die IP-Adresse ein

# 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  $\Rightarrow$  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







## 19/74

## 20/74







# Aufbau von IPv4-Paketen (1/4)

## • Version (4 Bits)

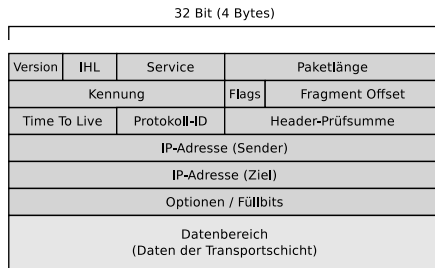
- Version des Protokolls
  - Version = 4  $\Rightarrow$  IPv4
  - Version = 6  $\Rightarrow$  IPv6

## • IHL = IP Header Length (4 Bits)

- Länge des IP-Headers in Vielfachen von 4 Bytes
  - Beispiel: IHL = 5  $\Rightarrow$  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 (3/4)

- **Protokoll-ID** (8 Bits)

- Nummer des übergeordneten Protokolls in der Transportschicht
- TCP-Segment  $\Rightarrow$  6
- UDP-Segment  $\Rightarrow$  17
- ICMP-Nachricht  $\Rightarrow$  1
- OSPF-Nachricht  $\Rightarrow$  89

32 Bit (4 Bytes)			
Version	IHL	Service	Paketlänge
Kennung		Flags	Fragment Offset
Time To Live	Protokoll-ID	Header-Prüfsumme	
IP-Adresse (Sender)			
IP-Adresse (Ziel)			
Optionen / Füllbits			
Datenbereich (Daten der Transportschicht)			

- Jedes IPv4-Paket enthält ein Feld für eine 16 Bits große Prüfsumme über die Daten des Headers
  - Weil sich bei jedem Router auf dem Weg zum Ziel der Inhalt des Datenfelds **Time To Live** ändert, müsste jeder Router die Prüfsumme überprüfen, neu berechnen und in den Header einsetzen

Router ignorieren die Prüfsumme üblicherweise, um die Pakete schneller weiterleiten zu können

Darum enthalten IPv6-Pakete auch kein Datenfeld für die Prüfsumme

# Aufbau von IPv4-Paketen (4/4)

32 Bit (4 Bytes)				
Version	IHL	Service	Paketlänge	
Kennung			Flags	Fragment Offset
Time To Live	Protokoll-ID		Header-Prüfsumme	
IP-Adresse (Sender)				
IP-Adresse (Ziel)				
Optionen / Füllbits				
Datenbereich (Daten der Transportschicht)				

- **IP-Adresse (Sender)** (32 Bits) enthält die Adresse des Senders und das Datenfeld **IP-Adresse (Ziel)** die Adresse des Ziels
- **Optionen / Füllbits** kann Zusatzinformationen wie einen Zeitstempel enthalten
  - Dieses letzte Feld vor dem Datenbereich mit den Nutzdaten wird gegebenenfalls mit Füllbits (Nullen) aufgefüllt, weil es wie der vollständige Header auch ein Vielfaches von 32 Bits groß sein muss
- Der abschließende Datenbereich enthält die Daten der Transportschicht

# 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

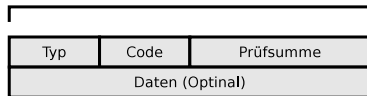
- Alle Router und Endgeräte können mit ICMP umgehen
- Typische Situationen, wo ICMP zum Einsatz kommt:
  - Ein Router verwirft ein IP-Paket, weil er nicht weiß, wie er es weiterleiten kann
  - Nur ein Fragment eines IP-Pakets kommt am Ziel an
  - Das Ziel eines IP-Pakets ist unerreichbar, weil die Time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

32 Bit (4 Bytes)

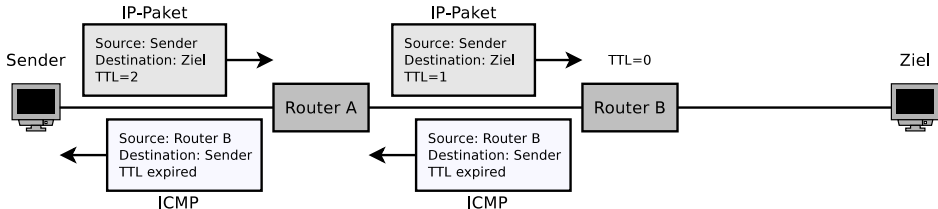


Typ	Typname	Code	Bedeutung
0	Echo-Antwort	0	Echo-Antwort (Antwort auf ping)
3	Ziel nicht erreichbar	0	Netz unerreichbar
		1	Ziel unerreichbar
		2	Protokoll nicht verfügbar
		3	Port nicht verfügbar
		4	Fragmentierung nötig, aber im IP-Paket untersagt
5	Umleitung (Redirect)	13	Firewall des Ziels blockt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



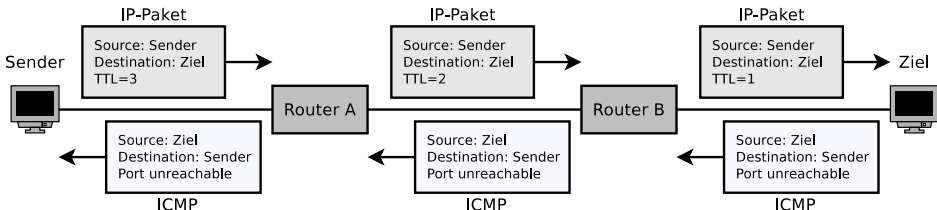
- Daraufhin schickt der Sender ein IP-Paket an den Empfänger mit TTL=2
- Das IP-Paket wird von Router A weitergeleitet
  - Dabei wird auch der Wert von TTL dekrementiert
- Router B empfängt das IP-Paket, setzt TTL=0, verwirft das IP-Paket und sendet eine ICMP-Nachricht vom Nachrichtentyp 11 und Code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

Achtung! Es gibt verschiedene Implementierungen von traceroute

tracert unter Windows verwendet standardmäßig ICMP aber traceroute unter Linux und Mac OSX 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
```



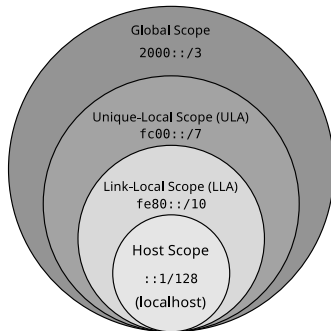






# Gültigkeitsbereiche – Scopes (1/4)

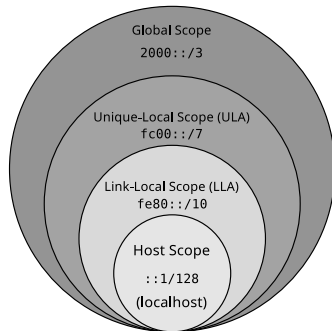
- IPv6 unterscheidet nicht nur zwischen privaten und öffentlichen Adressen (wie IPv4), sondern auch mehrere Gültigkeitsbereiche (sog. *Scopes*)
- Jede IPv6-Adresse hat einen sogenannten Scope
- Der Gültigkeitsbereich ist der Teil eines Netzes, in dem die zugehörige Adresse als gültig betrachtet und weitergeleitet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





## Gültigkeitsbereiche – Scopes (3/4)

- Unique-Local Scope: Unique Local Addresses (ULA)**
    - Router sollen die Adressen `fc00::/7` ( $\implies$  `fc00...` bis `fdff...`) nicht außerhalb des lokalen Verwaltungsbereichs weiterleiten
    - Private Adressen* zur lokalen Kommunikation innerhalb eines Verwaltungsbereichs (Organisation oder Standort)
  - `fc...`  $\implies$  zugewiesene eindeutige ULA
    - Auf das Präfix `fc` folgt eine 40 Bits lange zugewiesene (eindeutige) Site-ID und eine 16 Bits lange Subnetz-ID
    - Weltweit gültige, eindeutige, von einem Provider vergebene Adressen
  - `fd...`  $\implies$  lokal generierte ULA
    - Auf das Präfix `fd` folgt eine 40 Bits lange selbständig generierte (wahrscheinlich eindeutige) Site-ID und eine 16 Bits lange Subnetz-ID
- 







# Unique Local Addresses – ULA (RFC 4193)

- Die Tabelle zeigt das Adressierungsschema für statisches IPv6 RFC 4193

Präfix/Länge	Global-ID	Subnetz-ID	Interface-ID
fd00::/8	40 Bits	16 Bits	64 Bits
fd00::/8	12:3456:789a	0001	0000:0000:0000:0001

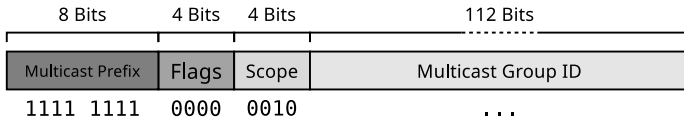
**Resultierende IPv6-Adresse:** fd12:3456:789a:0001:0000:0000:0000:0001

**Vereinfachte IPv6-Adresse:** fd12:3456:789a:1::1

- ULAs werden dort eingesetzt, wo eine Netz-ID (Netzwerk-Präfix) von einem Provider bereitgestellt wird
- Wird nur für lokalen Umgebungen (lokaler Verwaltungsbereich) genutzt
  - Diese Adressen werden nicht in das globale Internet geroutet
    - Bei lokal generierten ULAs sind Adresskonflikte möglich (aber sehr unwahrscheinlich)
  - ULAs sind analog zu privaten Adressen in IPv4
    - Sie werden innerhalb der Verwaltungsdomäne (Standort oder Organisation) verwendet

# 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)

# IPv6-Multicast-Adressen (2/2)

Präfix	Scope	Bedeutung
ff01	interface-local	Pakete an diese Adresse verlassen die Schnittstelle nicht $\Rightarrow$ <b>Loopback</b>
ff02	link-local	Pakete werden von Routern nicht weitergeleitet $\Rightarrow$ bleiben im Subnetz

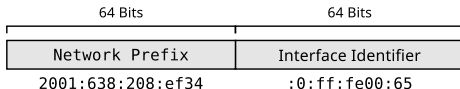
Der veraltete (siehe RFC 3879) Gültigkeitsbereich ff05 (site-local) definiert, dass ein Paket nicht von Border-Routern weitergeleitet wird  $\Rightarrow$  es das Netz einer Organisation nicht verlässt

- Auf den Gültigkeitsbereich folgt die Multicast-Gruppen-ID
- Die Tabelle enthält einige gängige Multicast-Adressen

IPv6-Adressen	Scope	Bedeutung (Adressen...)
ff01::1	interface-local	alle Knoten im lokalen Netz
ff01::2	interface-local	alle lokalen Router
ff02::1	link-local	alle Knoten im lokalen Netz $\Rightarrow$ emuliert <b>Broadcast</b>
ff02::2	link-local	alle lokalen Router
ff02:1:2	link-local	all lokale DHCPv6-Server
ff02::9	link-local	alle lokalen Router, die das Routing-Protokoll RIP einsetzen
ff02::5	link-local	alle lokalen Router, die das Routing-Protokoll OSPF einsetzen
ff02::6	link-local	alle lokalen designierten Router, die OSPF einsetzen
FF02::F	link-local	alle Geräte, die UPnP (Universal Plug and Play) einsetzen

# Konfiguration der Schnittstellenkennung (Interface-ID)

- Die **Interface-ID** kann auf verschiedene Arten konfiguriert werden



## 1 Statische manuelle Adressierung

- Interface ID manuell festlegen – möglich, aber unpraktisch

## 2 Zustandslose automatische Adresskonfiguration (RFC 4862)

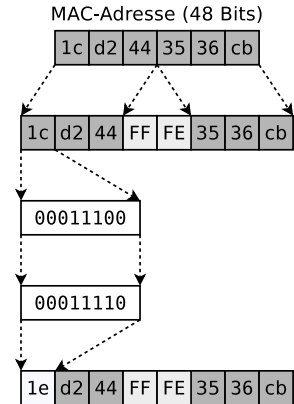
- Interface-ID (64 Bits) aus der MAC-Adresse (48 Bits) berechnen  
⇒ Die Interface-ID heißt dann **Extended Unique Identifier (EUI)**
- Optionale Erweiterung: **Stable Privacy Addresses** (RFC 7217)
  - Dauerhafte Interface-ID mit Hilfe eines zufälligen geheimen Schlüssels berechnen (ohne Verwendung der MAC-Adresse), um Anonymität zu gewährleisten
- Optionale Erweiterung: **Privacy Extension** (RFC 4941)
  - Regelmäßige Berechnung einer neuen Interface-ID unter Verwendung einer Zufallszahl (ohne Verwendung der MAC-Adresse), für noch mehr Anonymität

## 3 Netzwerkkonfiguration über DHCPv6 einstellen (RFC 8415)

- Einzige Möglichkeit zur Adresskonfiguration, die **zustandsbehaftet** (*stateful*) funktioniert

# Stateless Address Autoconfiguration – SLAAC (RFC 4862)

- Automatische zustandslose IPv6-Adressgenerierung durch Verwendung der MAC-Adresse
- MAC in eine Host-ID umwandeln
  - 1 Die MAC-Adresse wird halbiert
    - 1. Teil bildet die ersten 24 Bits
    - 2. Teil bildet die letzten 24 Bits der modifizierten EUI-64-Adresse
  - 2 Bitmuster der 16 Bits in der Mitte der EUI-64-Adresse: 1111 1111 1111 1110 (hex: FFFE)
  - 3 Abschließend das siebte Bits invertieren
- Nachteil: Einfache Rückgewinnung der MAC-Adresse (⇒ Datenschutzbedenken)



Router ⇒ Advertisement Daemon (radvd)

Für die automatische Vergabe von Netzwerk-Präfixen benötigt der Router einen radvd zur Verwaltung von Netzwerk-Präfixen im Netzwerk. Ohne radvd wird das Link-Local-Präfix fe80::/64 zugewiesen



# 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**

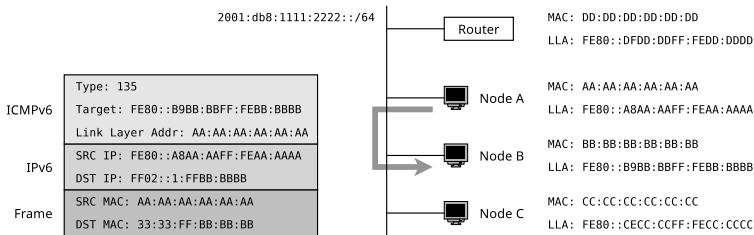
In IPv6 beschreibt der Begriff **Neighbor** Knoten, die sich im gleichen Netzwerk der Sicherungsschicht (Data Link Layer) befinden

- 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



# Neighbor Solicitation – NS (1/2)

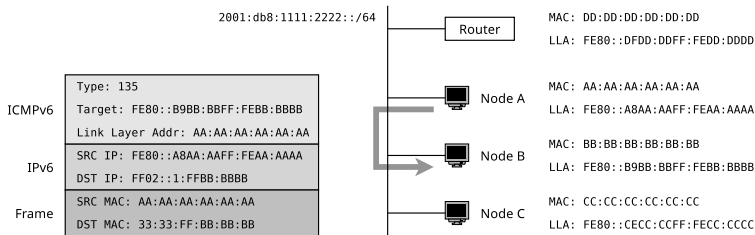


Die Nachricht Neighbor Solicitation (NS) ist die IPv6-Alternative zu einer ARP-Anfrage bei der Nutzung von IPv4

- Anforderung der MAC-Adresse eines Nachbarn
  - In der Abbildung fordert Knoten A die MAC-Adresse von Knoten B an
- Ziel-IP-Adresse im IPv6-Paket = **Solicited-node multicast address**
  - Jeder Knoten tritt einer Multicast-Gruppe für jede konfigurierte IPv6-Adresse bei
    - Die Multicast-Gruppe hat die Adresse FF02::1:FFXX:XXXX

XX:XXXX steht für die letzten 6 hexadezimalen Zeichen der Link-Local (Unicast) Adresse (LLA)

# Neighbor Solicitation – NS (2/2)



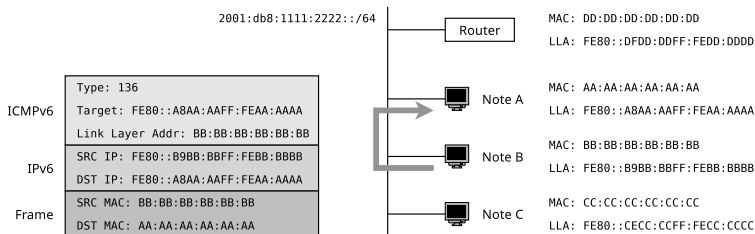
- Ziel-MAC-Adresse im Rahmen = **Multicast-MAC-Adresse**
  - Die Multicast-MAC-Adresse ist 33:33.XX:XX:XX:XX

XX:XX:XX:XX steht für die letzten 8 hexadezimalen Zeichen der Multicast-Adresse des angefragten Knotens

# Erkennung doppelter Adressen mit Neighbor Solicitation

- Die Nachricht Neighbor Solicitation (NS) wird auch zur Erkennung von Adressduplikaten – **Duplicate Address Detection (DAD)** verwendet
  - Wenn ein Knoten eine vorläufige (*tentative*) IPv6-Adresse für sich selbst generiert, muss er prüfen, dass kein anderer Knoten im Netz diese Adresse bereits verwendet
- Der Knoten sendet eine Nachricht Neighbor Solicitation (NS) an die Adresse, die er selbst verwenden möchte
  - Absenderadresse ist die unspezifische Adresse ( $:: \Rightarrow 128$  Nullbits)
  - Wenn ein Knoten im lokalen Netz diese IP-Adresse bereits verwendet, handelt es sich um ein Duplikat
    - Der Knoten antwortet mit einer Nachricht Neighbor Advertisement (NA) an die link-lokale Multicast-Adresse FF02::1 (jeder Knoten im lokalen Netzwerk erhält diese Nachricht)
  - Der Knoten, der die Nachricht Neighbor Solicitation (NS) gesendet hat, muss eine neue Adresse erzeugen und die Duplicate Address Detection erneut durchführen
  - Wenn für einige Zeit keine Nachricht Neighbor Advertisement (NA) empfangen wird, kann die Adresse verwendet werden ( $\Rightarrow$  kein Duplikat)

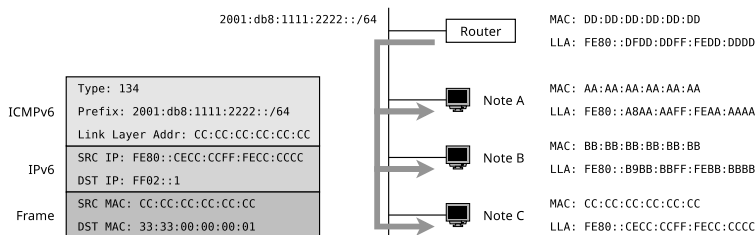
# Neighbor Advertisement – NA



Die Nachricht Neighbor Advertisement (NA) ist die IPv6-Alternative zu einer ARP-Antwort bei der Nutzung von IPv4

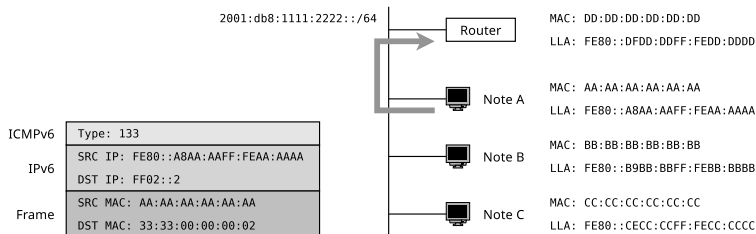
- Antwort auf eine Neighbor Solicitation (NS) Nachricht
- Neighbor Advertisement ist eine Unicast-Nachricht
  - Hier werden keine Multicast-Adressen verwendet

# Router Advertisement – RA



- Router senden in regelmäßigen Abständen (die Zeit kann eingestellt werden) RA-Nachrichten in angeschlossene Netze
  - Damit informieren sie über ihre Anwesenheit, das Netzwerk-Präfix, die Präfix-Länge und u.a. die MTU
  - Zieladresse im IPv6-Paket ist die link-lokale Multicast-Adresse FF02::1 um alle Knoten im lokalen Netz zu erreichen
- Die RA-Nachricht enthält auch das Flag **managed**
  - Ist es gesetzt, soll der Client die Adresse nicht zustandslos selbst festlegen, sondern von einem DHCPv6-Server anfordern (zustandsbehaftet)

# Router Solicitation – RS



- Wenn ein Knoten nicht auf eingehende RA-Nachrichten (Router Advertisement) warten möchte, kann er diese anfordern, indem er RS-Nachrichten sendet
  - Zieladresse im IPv6-Paket ist die link-lokale Multicast-Adresse FF02::2 um alle Router im lokalen Netz zu erreichen
  - In der Abbildung fordert Knoten A von jedem lokalen Router die RS-Nachricht an

# 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

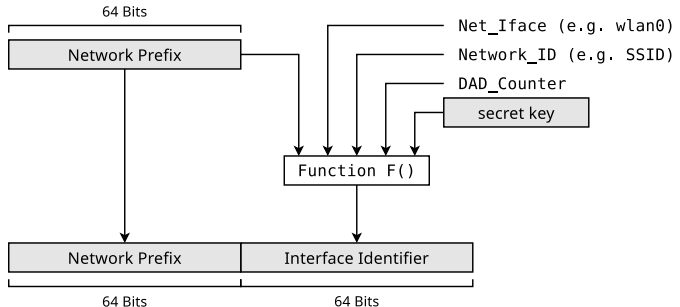
```
$ cat /proc/sys/net/ipv6/conf/eth0/stable_secret  
c8c8:036d:9312:71e2:eadc:7c9f:0535:649a
```

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

# SLAAC-Erweiterung: Stable Privacy (RFC 7217) – (2/3)

**Learned from  
ICMPv6 RA or  
Link-Local  
Unicast Prefix**

**IPv6 Address**



- Pseudozufällige Funktion zur Erzeugung der Schnittstellen-ID:  
 $F(\text{Prefix}, \text{Net\_Iface}, \text{Network\_ID}, \text{DAD\_Counter}, \text{key})$

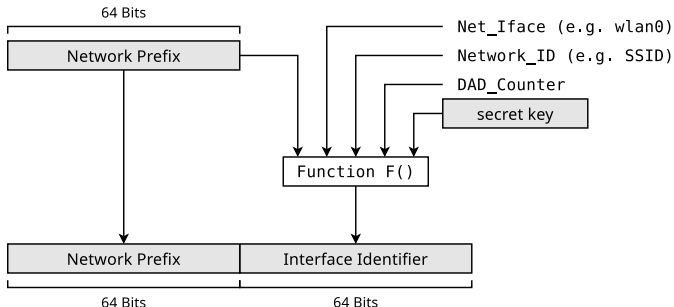
- Prefix: Aus einer ICMPv6 Router Advertisement (RA) Nachricht oder dem Link-Local Unicast Präfix entnommen
- Net\_Iface: Der Netzwerkschnittstelle zugeordnete ID (z. B. wlan0)
- Network\_ID: Dem Netzwerk zugeordnete ID. z. B. der WLAN Service Set Identifier (SSID)
- DAD\_Counter: Zum Auflösen von DAD-Konflikten (Duplicate Address Detection). Der Anfangswert ist 0. Er wird für jede neue Adresse, die konfiguriert wird, um 1 erhöht.
- key: 128 Bits langer geheimer Schlüssel



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

**Learned from  
ICMPv6 RA or  
Link-Local  
Unicast Prefix**

**IPv6 Address**



- SHA-1 und SHA-256 sind zwei mögliche Optionen für F()
- Aber nicht MD5 (siehe RFC 6151)

Beispiel für eine erzeugte Adresse mit Stable Privacy

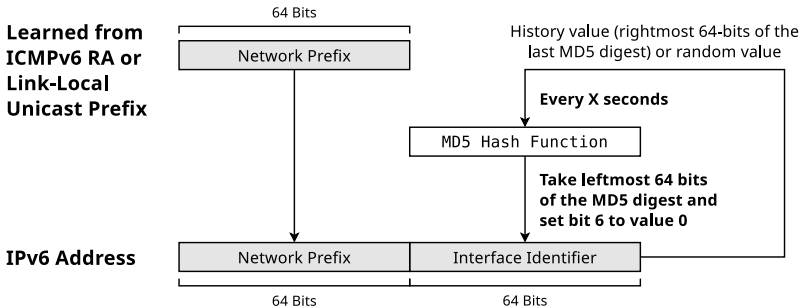
MAC: 86:3a:ea:8a:a7:d9

stable-privacy -> inet6 fe80::6f6d:80e:ab6c:65a0/64

link local EUI-64 -> inet6 fe80::843a:eaff:fe8a:a7d9/64



# SLAAC-Erweiterung: Privacy Extension (RFC 4941) – (2/2)



## Beispiel für eine zufällig erzeugte Adresse mit Privacy Extension

MAC: 86:3a:ea:8a:a7:d9

privacy-extension -> inet6 fd12::8992:3c03:d6e2:ed72/64

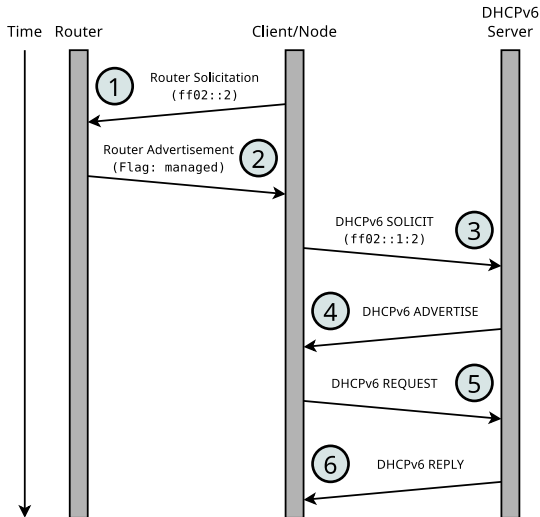
link local -> inet6 fe80::843a:eaff:fe8a:a7d9/64

## Zufällige erzeugte Interface-ID

Die oben gezeigte Adresse wird zufällig und temporär generiert und kann nicht zu irgendwelchen Merkmalen des Knotens zurückverfolgt werden

# 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



# IPv4-Adressen in IPv6-Netze einbetten (*IPv4 mapped*)

- Eine global geroutete (Unicast) IPv4-Adresse kann als IPv6-Adresse dargestellt und somit in den IPv6-Adressraum integriert werden
  - Diese Vorgehensweise heißt in der Literatur *IPv4 mapped*
- Dafür erhält die IPv4-Adresse einen 96 Bits langen Präfix:  
0:0:0:0:0:FFF::/96

80 Bits					16 Bits	32 Bits
0000	0000	0000	0000	0000	FFFF	IPv4-Adresse

- Die IPv4-Adresse darf in hexadezimaler oder in dezimaler Schreibweise dargestellt sein

- Beispiel

IPv4-Adresse: 131.246.107.35  
 IPv6-Adresse: 0:0:0:0:0:FFFF:83F6:6B23  
 Kurzschreibweisen: ::FFFF:83F6:6B23  
 ::FFFF:131.246.107.35

# Aufbau von IPv6-Paketen

- Der Header von IPv6-Paketen hat eine feste Länge (320 Bits  $\Rightarrow$  40 Bytes)

32 Bit (4 Bytes)

Version	Traffic Class (Priorität für QoS)	Flow Label (für QoS)	
Länge des Datenbereichs		Next Header (z.B. TCP oder UDP)	Time To Live
IP-Adresse (Sender) 128 Bit			
IP-Adresse (Ziel) 128 Bit			
Datenbereich (Daten der Transportschicht)			

- Im Feld **Next Header** kann auf einen Erweiterungs-Kopfdatenbereich (Extension Header) oder das Protokoll der Transportschicht (z.B. TCP = Typ 6 oder UDP = Typ 17) verwiesen werden

Konzept: Vereinfachte (reduzierte) Paketstruktur und gleichzeitig können zusätzliche (neue) Funktionen durch eine Kette von Erweiterungs-Kopfdaten (*Extension Headers*) hinzugefügt werden

Das Thema Extension Header (siehe RFC 2460 und RFC 4303) wird in dieser Vorlesung nicht behandelt







## Netzübergreifende Kommunikation (3/6)

- Identische Subnetznummern  $\Rightarrow$  X und Y sind im gleichen logischen Subnetz

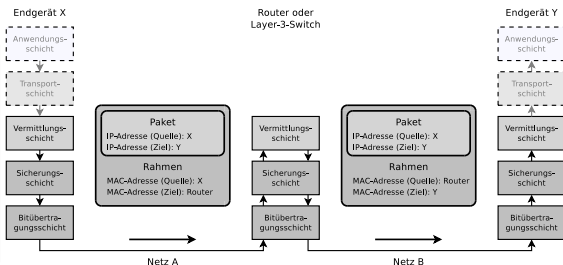
Sie wissen bereits (aus diesem Foliensatz)...

Ein logisches Subnetz deckt mindestens ein physisches Netz ab und kann immer nur mit einer Schnittstelle eines Routers verbunden sein

- Unterschiedliche Subnetznummern  $\Rightarrow$  X und Y sind in verschiedenen logischen Subnetzen  $\Rightarrow$  Der lokale Router muss sich um die Weiterleitung und Wegbestimmung kümmern

Sie wissen bereits (aus  
Foliensatz 9)...

Befinden sich 2 Kommunikationspartner im gleichen logischen und physischen Netz, kann der Sender via Adressauflösung mit ARP die MAC-Adresse von Empfängers erfahren



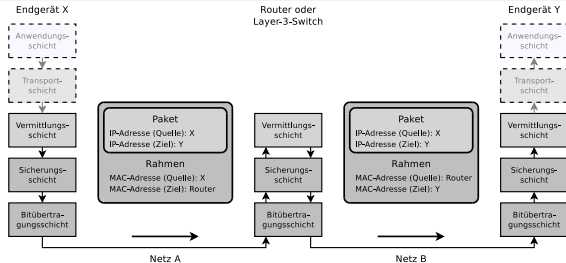
- Hier handelt es sich um Kommunikation über logische und physische Netzgrenzen hinweg

## Netzübergreifende Kommunikation (4/6)

Sie wissen bereits (aus Foliensatz 9)...

- ARP ist nur für die Auflösung der MAC-Adressen im lokalen physischen Netz zuständig
- Grund: ARP-Anfragen werden in Rahmen der Sicherungsschicht gesendet
- Das Feld mit der Zieladresse enthält die Broadcast-Adresse
- Solche Rahmen werden von Bridges und Switches nicht weitergeleitet  
 ⇒ Darum ist mit ARP keine netzübergreifende Adressauflösung möglich

- Im Nutzdatenteil des Rahmens befindet sich das IP-Paket für Y mit der IP-Adresse von X als Quelle und der IP-Adresse von Y als Ziel





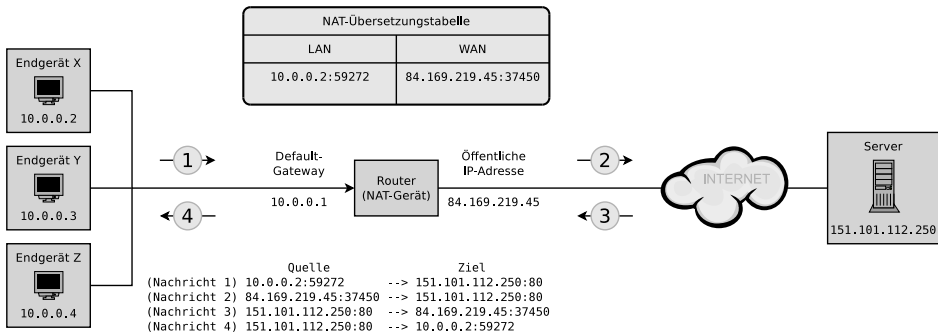


# Network Address Translation (1/5)

- Problem: Die allerwenigsten Haushalte, Unternehmen und Bildungs-/Forschungseinrichtungen haben genug öffentlich erreichbare IPv4-Adressen, um alle ihre Netzwerkgeräte mit eigenen IPs auszustatten
  - Darum verwenden lokale Netze meist einen privaten IPv4-Adressraum (siehe Folie 23)
  - Problem: Wie können Netzwerkgeräte in privaten Netzen mit Netzwerkgeräten mit global erreichbaren Adressen kommunizieren?
  - Lösung: **Network Address Translation (NAT)**
    - Der lokale Router gibt sich selbst als Quelle derjenigen IP-Pakete aus, die er aus dem direkt verbundenen privaten Netz ins Internet weiterleitet
    - Zudem leitet er eintreffende Antworten zu den Teilnehmern im direkt verbundenen privaten Netz zu



# Network Address Translation (3/5)

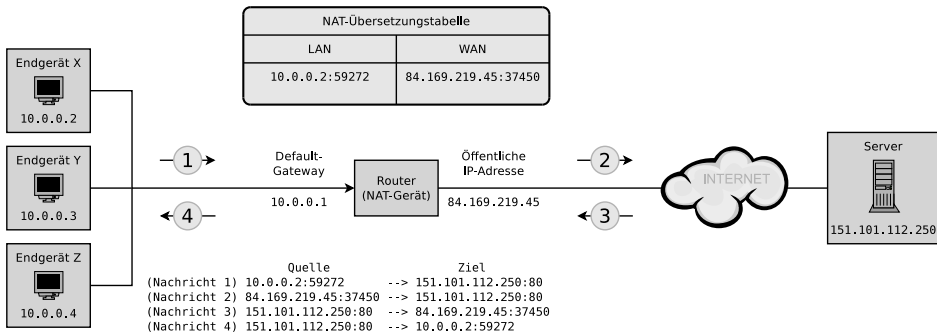


- Client X fordert eine Webseite vom Server an
  - Die Anfrage (Nachricht 1) enthält als Quelladressen die IP-Adresse und Portnummer von X und als Zieladressen die IP-Adresse und Portnummer des Servers
- Der Router ersetzt in der weitergeleiteten Anfrage (Nachricht 2) die IP und Portnummer des Clients durch seine eigenen Adressen





# Network Address Translation (5/5)



- Bei IPv6 ist NAT unnötig, weil der Adressraum groß genug ist, um allen Netzwerkgeräten global erreichbare Adressen zuzuweisen
  - Ob das aus Gründen der Sicherheit allerdings ratsam ist, ist umstritten
    - NAT verbessert die Netzwerksicherheit, weil es die Topologie des lokalen Netzes vor der Außenwelt verbirgt
- NAT bei IPv6: **IPv6-to-IPv6 Network Address Translation (NAT66)**