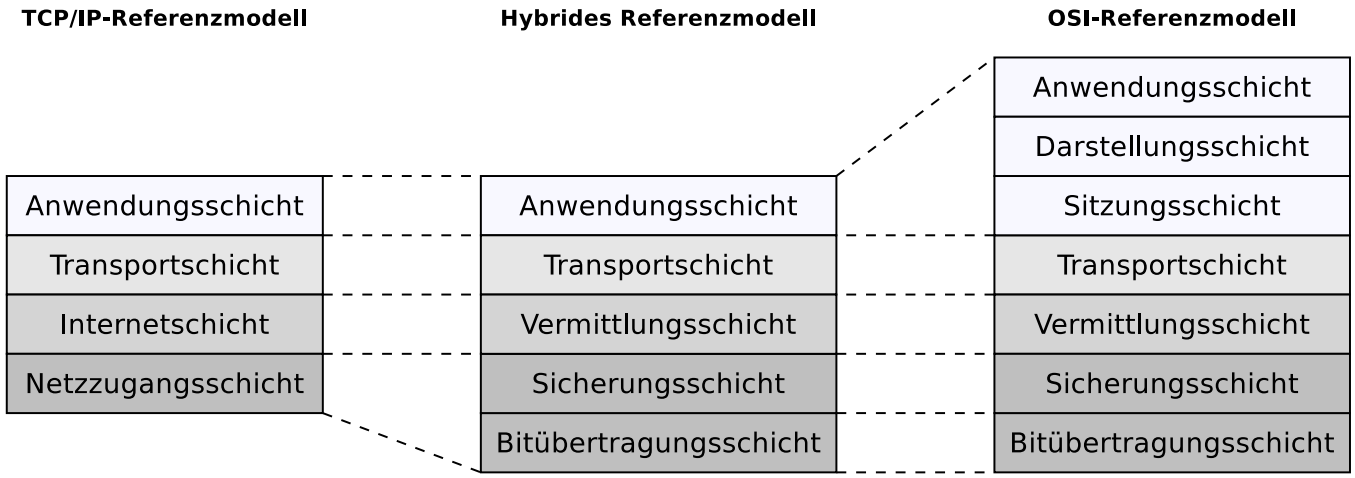
10. Foliensatz
Betriebssysteme und Rechnernetze

Prof. Dr. Christian Baun

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

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



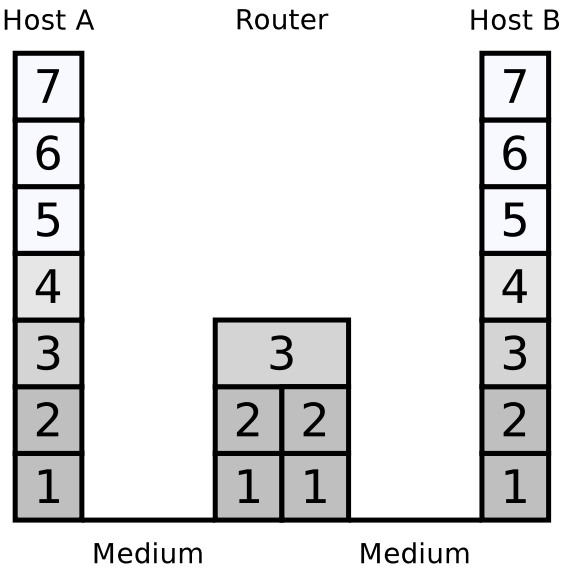
- Geräte: Router, Layer-3-Switch (Router ohne WAN-Schnittstelle)
- Protokolle: IPv4, IPv6, ICMP, IPX/SPX, DECnet

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
    - Aufbau von IP-Adressen
    - Netzklassen, Netzwerkteil und Geräteteil, Subnetze und Netzmaske
    - Private IP-Adressen
    - Aufbau von IP-Paketen
    - ~~Fragmentieren von IP-Paketen~~
  - ~~Weiterleitung und Wegbestimmung~~
    - ~~Distanzvektor-Routing-Protokolle~~
    - ~~Link-State-Routing-Protokolle~~
  - ~~Diagnose und Fehlermeldungen mit ICMP~~
  - Netzübergreifende Kommunikation $\Longrightarrow$ Internetworking (Zusammenfassung)
  - Network Address Translation (NAT)

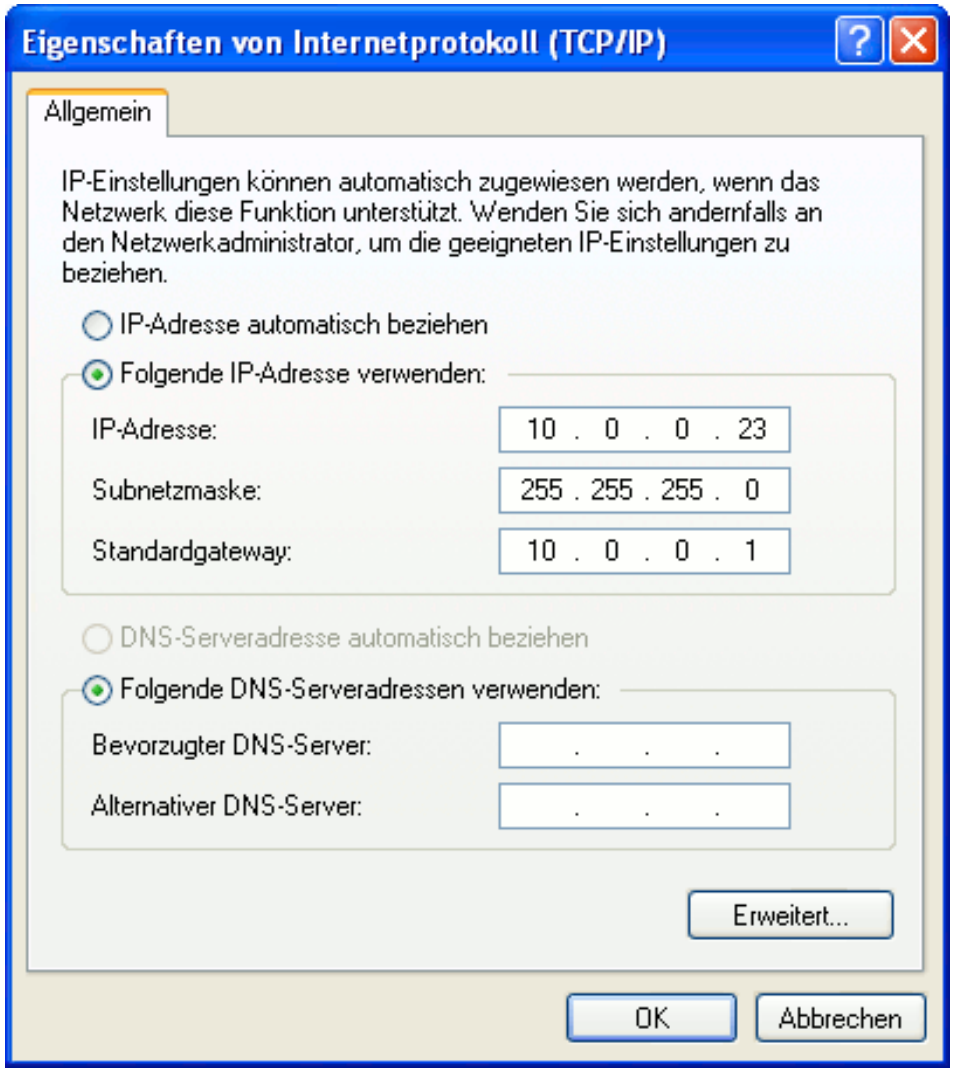
Router, Layer-3-Switch und Gateway

- **Router** leiten Datenpakete zwischen Netzen mit eigenen logischen Adressbereichen weiter
  - Besitzen genau wie Hubs und Switche mehrere Schnittstellen
  - Ermöglichen die Verbindung des lokalen Netzes (LAN) mit einem WAN (z.B. via DSL oder 3G/4G Mobilfunk)
- **Layer-3-Switche** sind Router ohne WAN-Schnittstelle
- **Gateways** sind Protokollumsetzer
  - Ermöglichen Kommunikation zwischen Netzen, die auf unterschiedlichen Protokollen basieren
  - Gateways, die auf der Vermittlungsschicht arbeiten, heißen auch **Mehrprotokoll-Router** oder **Multiprotokoll-Router**



Gateways (1/2)

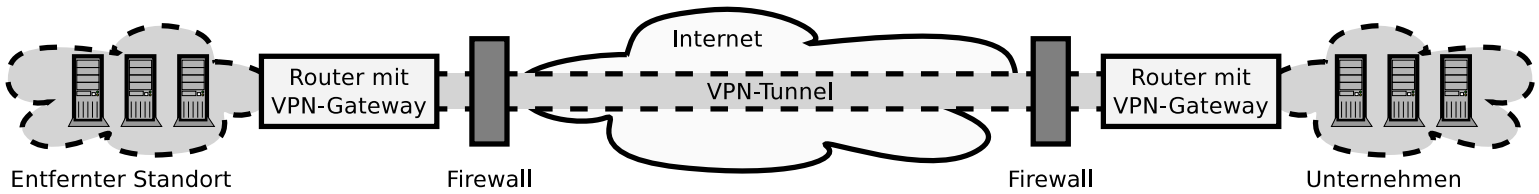
- Moderne Computernetze arbeiten fast ausschließlich mit dem Internet Protocol (IP)
  - Darum ist eine Protokollumsetzung auf der Vermittlungsschicht heute meist nicht nötig
- In früheren Zeiten wurde bei der Konfiguration eines Endgeräts der Gateway als **Default Gateway** eintragen
  - Heute trägt man in diesem Feld den Router ein, weil man keinen Gateway mehr braucht
    - Der Begriff **Default Router** wäre heute also eigentlich passender



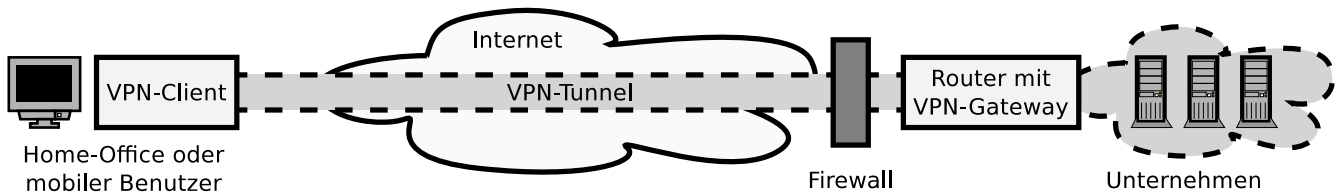
Gateways (2/2)

- 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

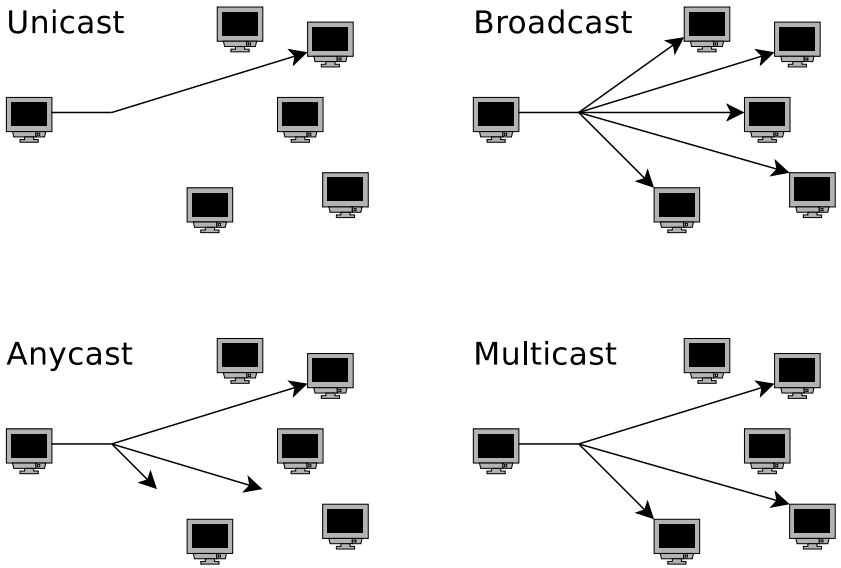


Adressierung in der Vermittlungsschicht (1/2)

- Ausschließlich physische Adressierung via MAC-Adressen ist in Computernetzen mit eventuell globalen Ausmaßen nicht sinnvoll
  - Grund: Wartbarkeit
- Es sind **logische Adressen** nötig, die von der konkreten Hardware unabhängig sind
  - Mit logischer Adressierung wird die Teilnehmersicht für Menschen (logische Adressen) von der internen Sicht für Rechner und Software (physische Adressen) getrennt

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 und das Network Time Protocol (NTP) zur Synchronisierung von Uhren
Anycast verwenden zum Beispiel einigen Root-Nameserver im Domain Name System

Aufbau von IP-Adressen

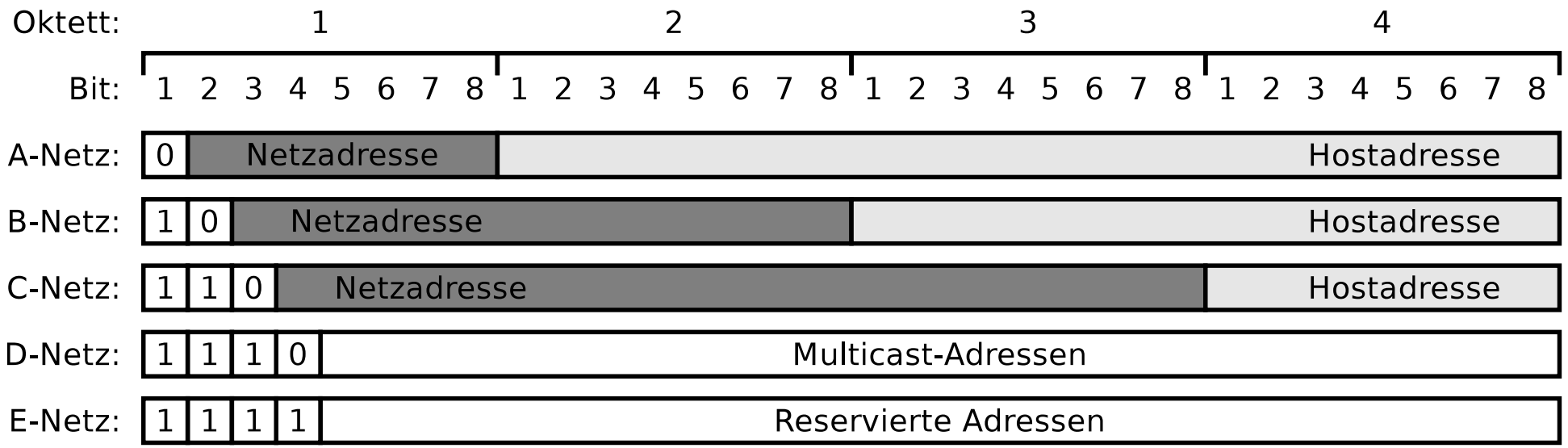
- IPv4-Adressen sind 32 Bits (4 Bytes) lang
  - Daher können $2^{32} = 4.294.967.296$ Adressen dargestellt werden

Adressraum = Menge aller gültigen Netzadressen

- Üblich ist die Darstellung in der sogenannten Dotted decimal notation
  - Die 4 Oktette werden als vier durch Punkte voneinander getrennte ganze Zahlen in Dezimaldarstellung im Bereich von 0 bis 255 geschrieben
    Beispiel: 141.52.166.25

Netzklassen, Netzwerkteil und Geräteteil

- Ursprünglich wurden IPv4-Adressen in Klassen von A bis C eingeteilt
  - Es existierten auch die Klassen D und E für spezielle Aufgaben
- Die 32 Bits einer IPv4-Adresse bestehen aus den beiden Feldern:
  - **Netzadresse** (Network Identifier bzw. Netzwerk-ID)
  - **Hostadresse** (Host Identifier bzw. Host-ID)
  - Klasse A: 7 Bits für Netzadresse und 24 Bits für Hostadresse
  - Klasse B: 14 Bits für Netzadresse und 16 Bits für Hostadresse
  - Klasse C: 21 Bits für Netzadresse und 8 Bits für Hostadresse



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 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?

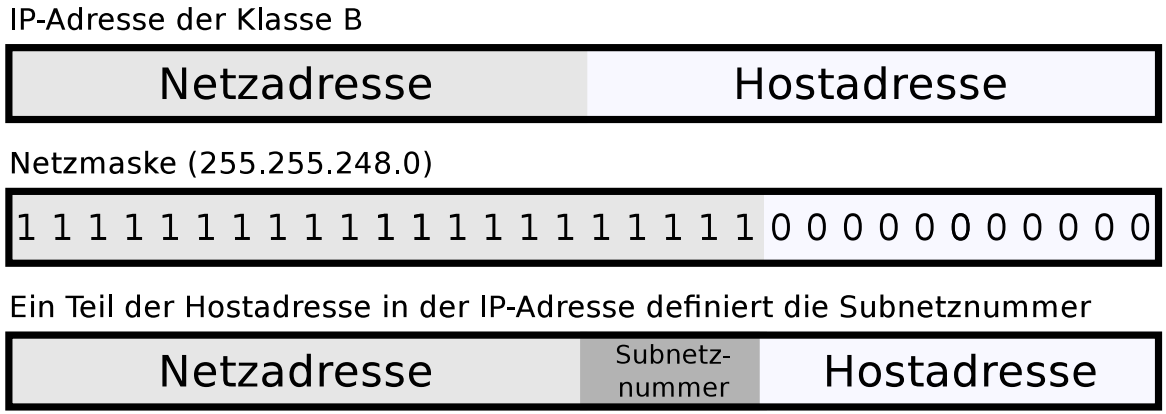
„The class E space has 268 million addresses and would give us in the order of 18 months worth of IPv4 address use. However, many TCP/IP stacks, such as the one in Windows, do not accept addresses from class E space and will not even communicate with correspondents holding those addresses. It is probably 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`

Netzklassen (2/2)

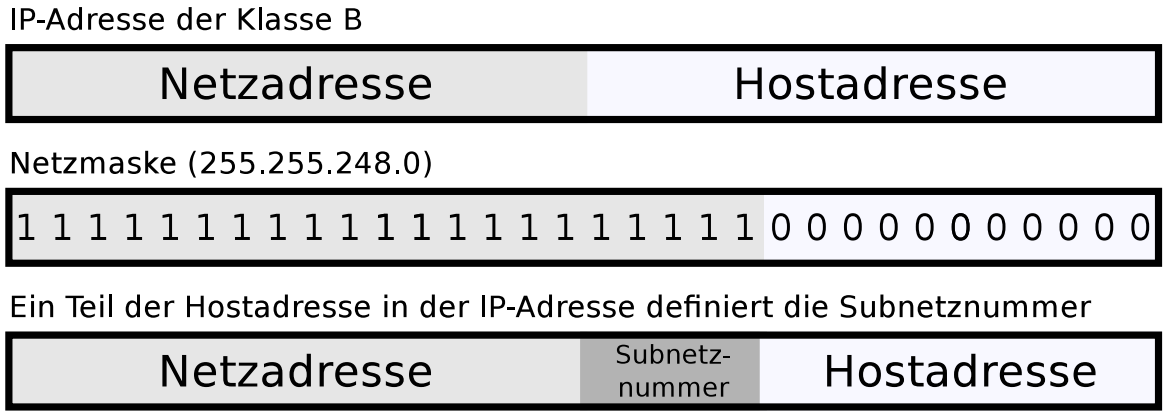
- Praktisch relevant sind nur die Klassen A, B und C
- Ursprünglich war beabsichtigt, durch die Netzadresse physische Netze eindeutig zu identifizieren
  - Dieses Vorgehen bringt aber Nachteile mit sich
- **Nachteile der Netzklassen:**
  - Sie können nicht dynamisch an Veränderungen angepasst werden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



- 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)



- 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 4 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 im Netzwerkteil 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 $\Longrightarrow$ 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

- Anhand der Tabelle ist es einfach, die nötigen Bits für Subnetze zu bestimmen

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	—

- 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 ($\Longrightarrow$ 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
```

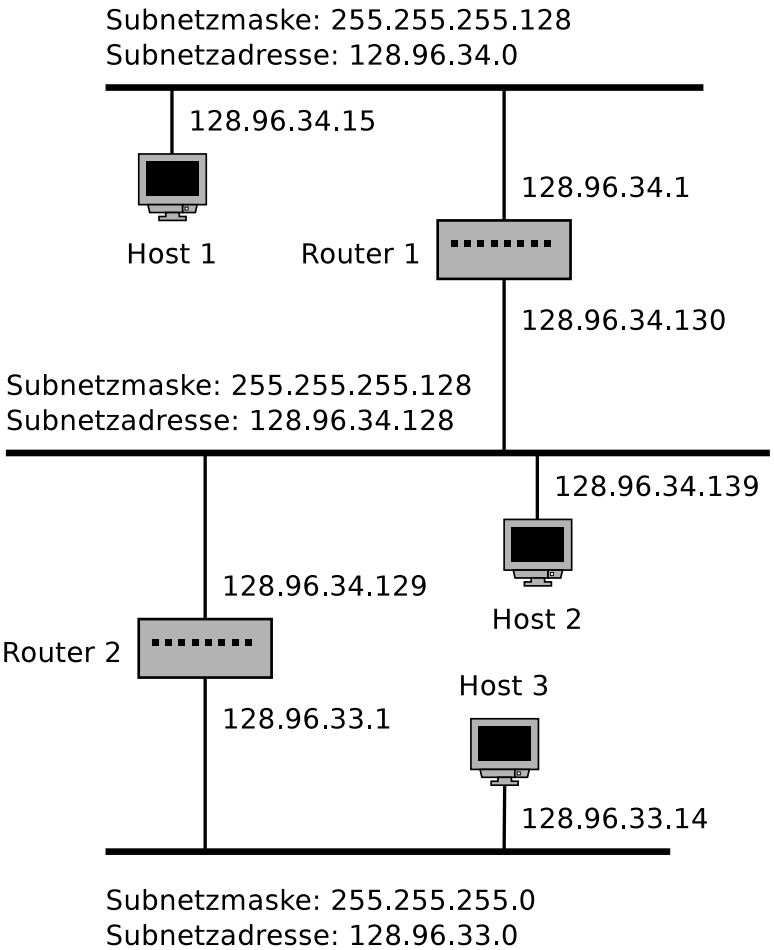
IP-Adresse	172.21.240.90	10101100 00010101 11110000 01011010
Netzmaske	255.255.255.224	11111111 11111111 11111111 11100000
Subnetzadresse	172.21.240.64	~~10101100 00010101~~ 11110000 010~~00000~~
Subnetznummer	1922	~~10101100 00010101~~ 11110000 010~~00000~~

- **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	~~00000000 00000000 00000000 000~~11111
Hostadresse	26	~~00000000 00000000 00000000 000~~11010

- /27 und Klasse B-Präfix $\Longrightarrow$ 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

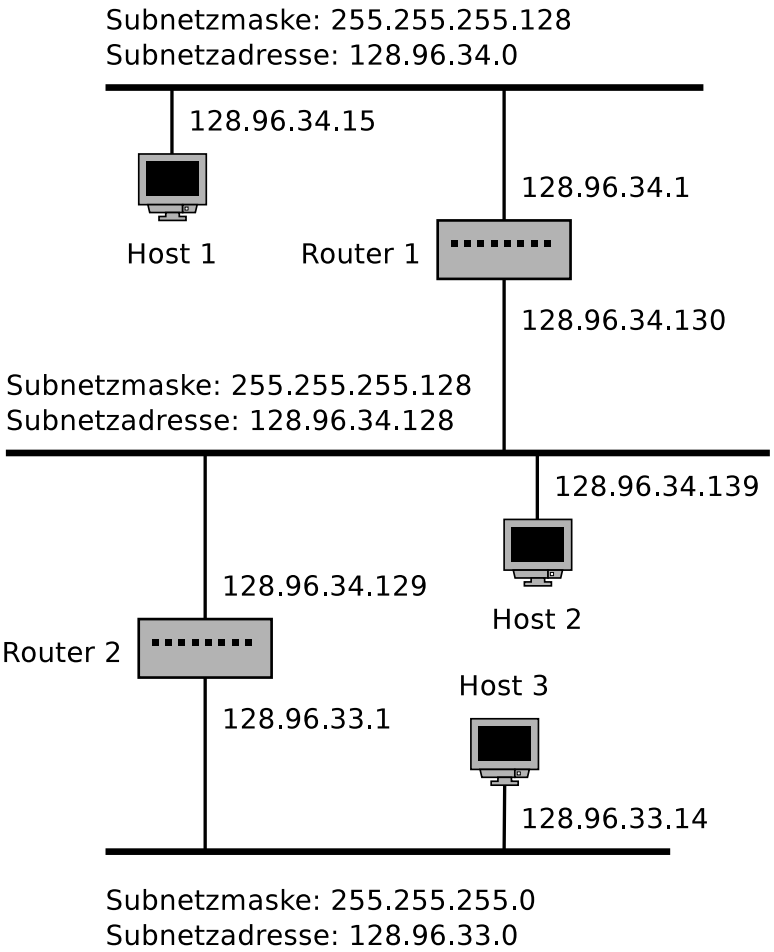
Beispiel (1/4)



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

- Alle Hosts im gleichen Subnetz haben die gleiche Subnetzmaske
- IP `AND` Subnetzmaske = 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

Beispiel (2/4)



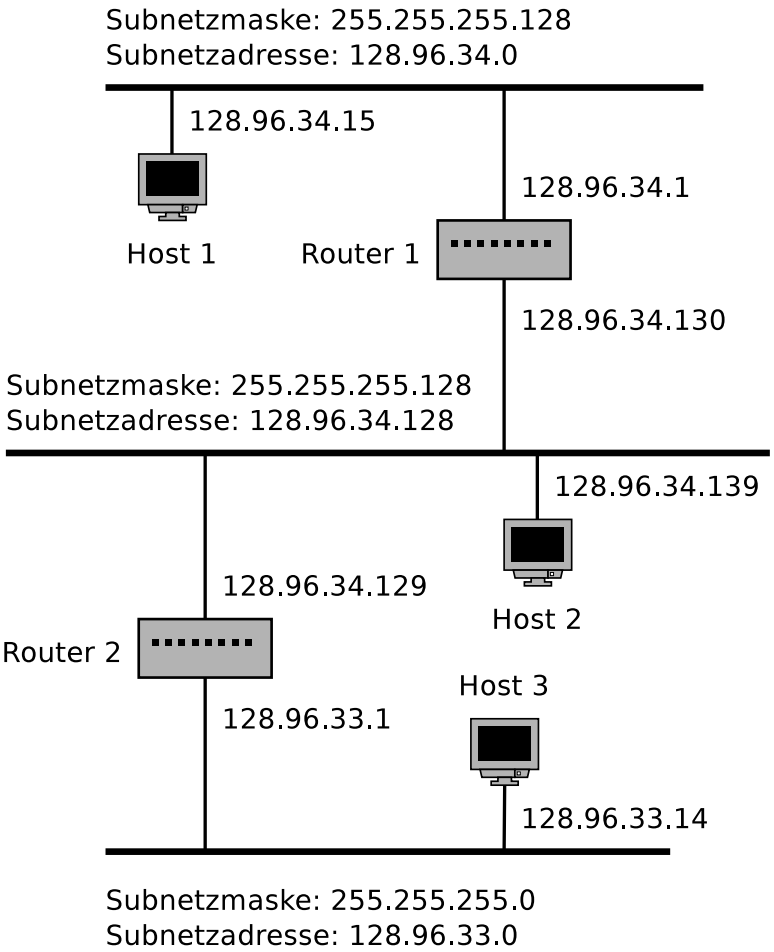
Quelle: Computernetzwerke. Peterson und Davie. dpunkt (2000)

- 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 $\Longrightarrow$ 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 erstellen und pflegen die Einträge in den Routern

Beispiel (3/4)



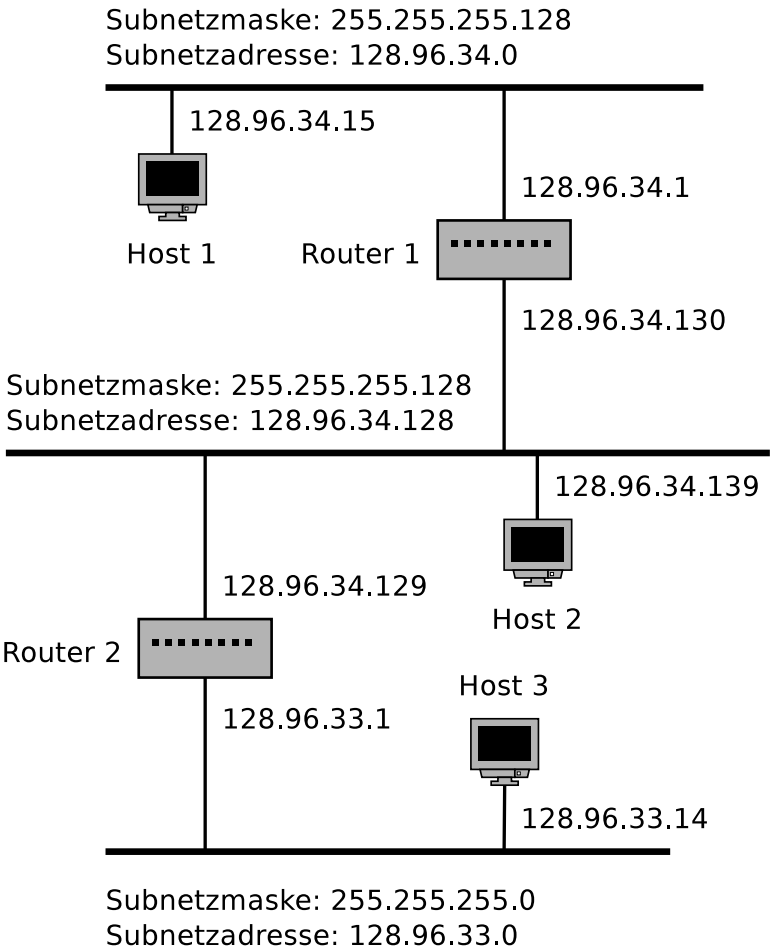
Quelle: Computernetzwerke. Peterson und Davie. dpunkt (2000)

- 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 Eintrage 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 128.92.34.128
- Das stimmt nicht mit der Subnetzadresse (128.96.34.0) überein

Beispiel (4/4)



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

- 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
  $\Longrightarrow$ 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
- Das Thema wird in der Vorlesung BSRN nicht behandelt

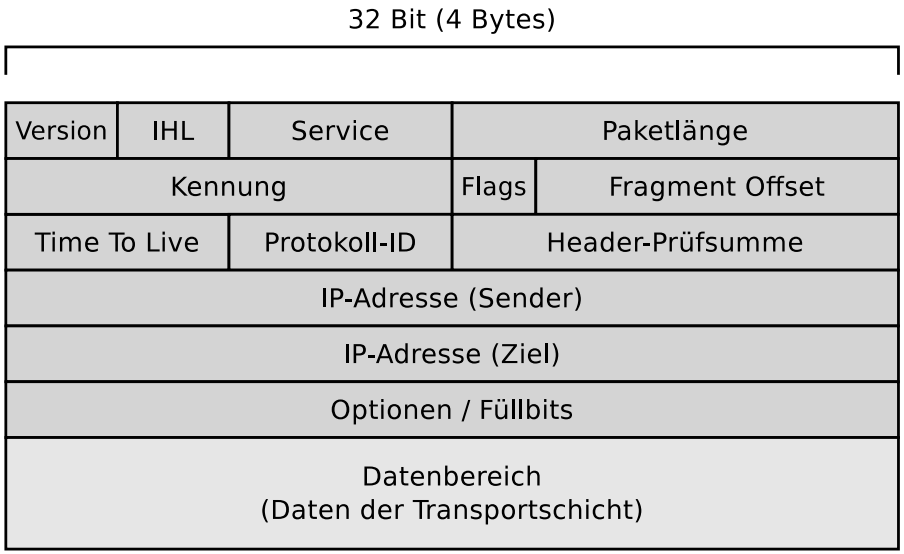
Private Netze – Private IP-Adressen

- Auch im privaten LAN müssen IP-Adressen vergeben werden
  - Diese sollten nicht mit real existierenden Internetangeboten kollidieren
- Dafür existieren Adressbereiche mit privaten IP-Adressen
  - Diese Adressbereiche werden im Internet **nicht geroutet**

Adressbereich:	10.0.0.0 bis 10.255.255.255
CIDR-Notation:	10.0.0.0/8
Anzahl Adressen:	$2^{24} = 16.777.216$
Netzklasse:	Klasse A. 1 privates Netz mit 16.777.216 Adressen
Adressbereich:	172.16.0.0 bis 172.31.255.255
CIDR-Notation:	172.16.0.0/12
Anzahl Adressen:	$2^{20} = 1.048.576$
Netzklasse:	Klasse B. 16 private Netze mit jeweils 65.536 Adressen
Adressbereich:	192.168.0.0 bis 192.168.255.255
CIDR-Notation:	192.168.0.0/16
Anzahl Adressen:	$2^{16} = 65.536$
Netzklasse:	Klasse C. 256 private Netze mit jeweils 256 Adressen

Aufbau von IPv4-Paketen (1/4)

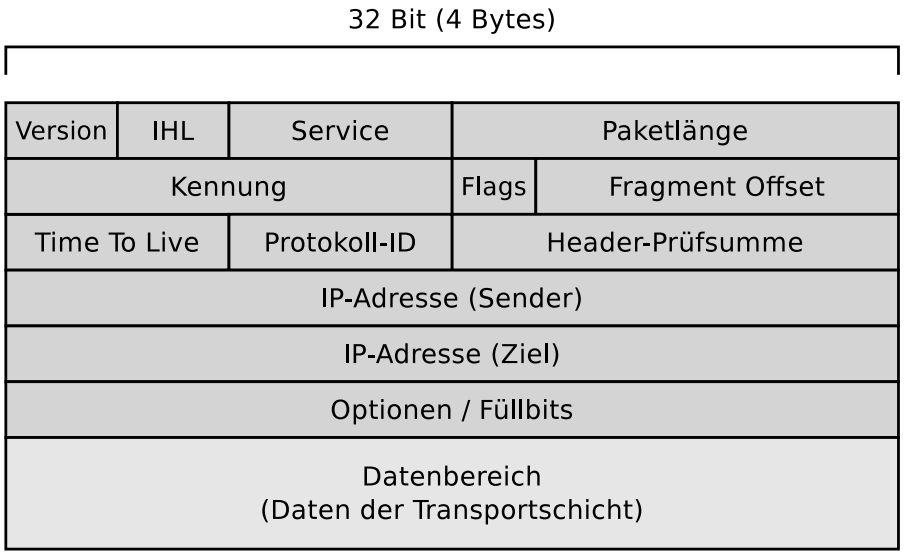
- **Version** (4 Bits)
  - Version des Protokolls
    - Version $= 4 \Longrightarrow$ IPv4
    - Version $= 6 \Longrightarrow$ IPv6



- **IHL** = IP Header Length (4 Bits)
  - Länge des IP-Headers in Vielfachen von 4 Bytes
    - Beispiel: IHL $= 5 \Longrightarrow 5 * 4$ 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 (2/4)

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



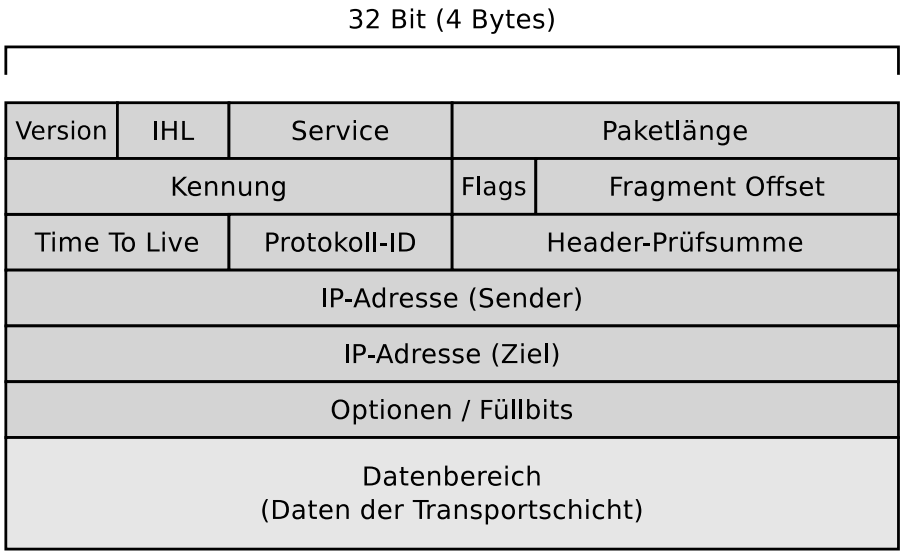
- Die Datenfelder **Kennung**, **Flags** und **Fragment Offset** steuern das Zusammensetzen fragmentierter IP-Pakete

Das Thema Fragmentierung von IP-Paketen wird in der Vorlesung BSRN nicht behandelt

- **Time To Live** (8 Bits)
  - Enthält die maximalen Hops
    - Jeder Router auf dem Weg zum Ziel verringert den Wert um eins
  - Das verhindert, das unzustellbare IP-Pakete endlos im Netz umherirren (kreisen)

Aufbau von IPv4-Paketen (3/4)

- **Protokoll-ID** (8 Bits)
  - Nummer des übergeordneten Protokolls in der Transportschicht
  - TCP-Segment $\Longrightarrow$ 6
  - UDP-Segment $\Longrightarrow$ 17
  - ICMP-Nachricht $\Longrightarrow$ 1
  - OSPF-Nachricht $\Longrightarrow$ 89

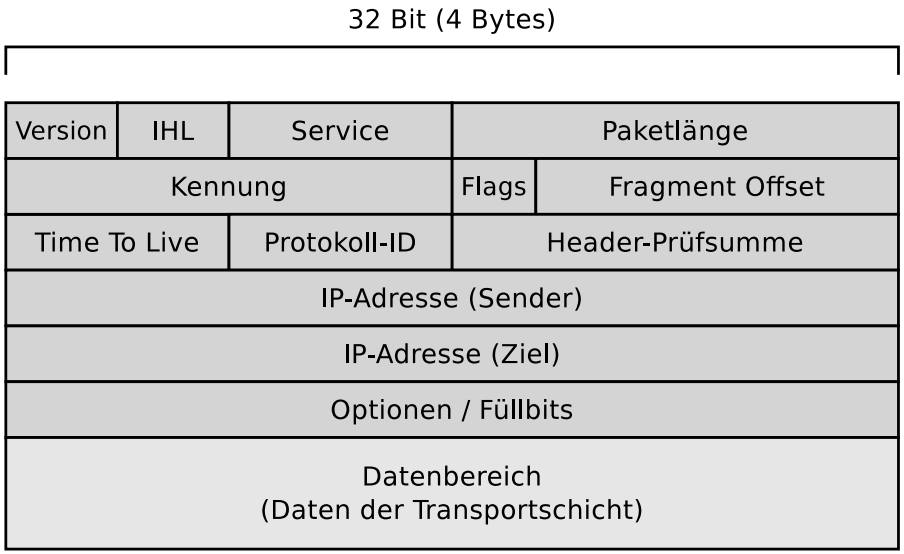


- Jedes IPv4-Paket enthält ein Feld für eine 16 Bits große Prüfsumme über die Daten des Headers
  - Weil sich bei jedem Router auf dem Weg zum Ziel der Inhalt des Datenfelds **Time To Live** ändert, müsste jeder Router die Prüfsumme überprüfen, neu berechnen und in den Header einsetzen

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)



- **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

Struktur von IPv6-Adressen und Netzen (1/5)

- IPv6-Adressen bestehen aus 128 Bits (16 Bytes)
  - Daher können 2^{128}, also $\approx 3{,}4 * 10^{38}$ Adressen dargestellt werden
- Einführung ist wegen des begrenzten Adressraums von IPv4 sinnvoll
- Problem: Dezimaldarstellung ist unübersichtlich
  - Aus diesem Grund stellt man IPv6-Adressen hexadezimal dar
    - Je 4 Bits werden als eine hexadezimale Zahl dargestellt
  - Je 4 Hexadezimalzahlen werden zu Blöcken gruppiert
  - Die Blöcke werden durch Doppelpunkte getrennt
    Beispiel: `2001:0db8:85a3:08d3:1319:8a2e:0370:7344`

- Die letzten 4 Bytes (32 Bits) einer IPv6-Adresse dürfen auch in dezimaler Notation geschrieben werden
- Das ist sinnvoll, um den IPv4-Adressraum in den IPv6-Adressraum einzubetten
  $\Longrightarrow$ siehe Folie 33

RFC 4291 (2006) „IP Version 6 Addressing Architecture“

Struktur von IPv6-Adressen und Netzen (2/5)

- 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`
    $\Longrightarrow$ `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, das die Portnummer als Teil der IPv6-Adresse interpretiert wird

Struktur von IPv6-Adressen und Netzen (3/5)

- IPv6-Adressen bestehen aus 2 Teilen

64 Bits	64 Bits
Network Prefix	Interface Identifier
2001:638:208:ef34	:0:ff:fe00:65

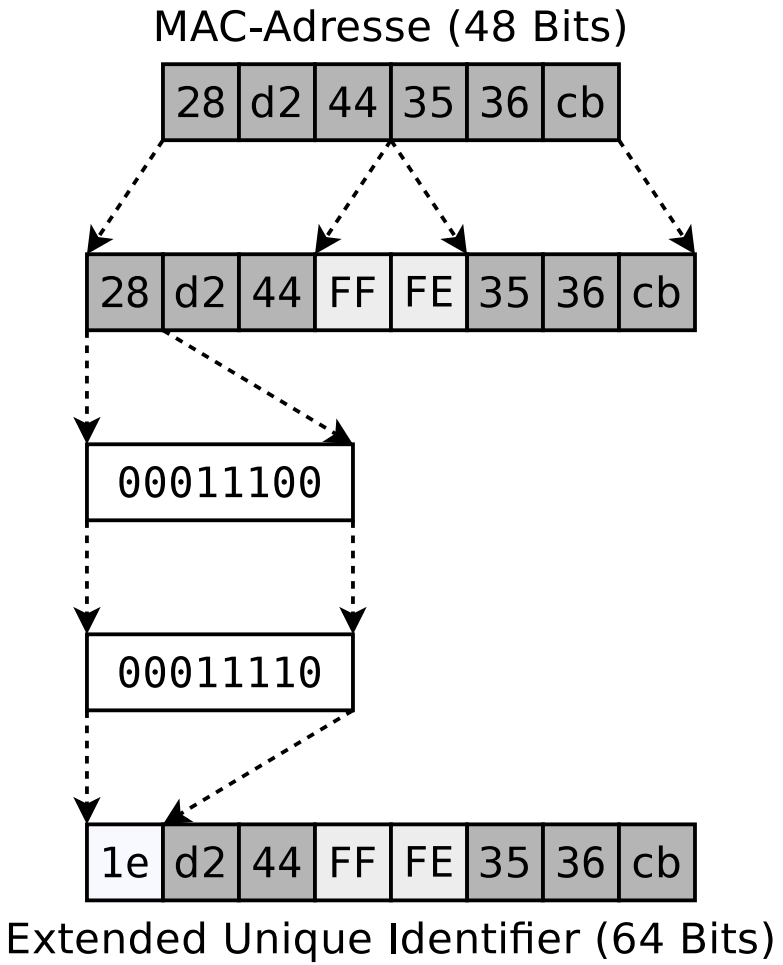
1. **Präfix** (Network Prefix)
   - Kennzeichnet das Netz
2. **Interface Identifier** (Interface-ID)
   - Kennzeichnet eine Netzwerkgerät in einem Netz
   - Kann manuell festgelegt, via DHCPv6 zugewiesen oder aus der MAC-Adresse der Netzwerkschnittstelle gebildet werden
   - Wird der Interface Identifier aus der MAC-Adresse gebildet, heißt er **Extended Unique Identifier** (EUI)
     - Dabei wird die MAC-Adresse (48 Bits) in eine 64-Bit-Adresse umgewandelt $\Longrightarrow$ **modifiziertes EUI-64 Adressformat** (siehe Folie 31)

Einige Adressbereiche

`fe80::/10` $\Longrightarrow$ Link-Local-Adressen. Diese sind nur im lokalen Netz gültig. Sie werden nicht von Routern weitergeleitet
`2000::/3` $\Longrightarrow$ (`2000...` bis `3fff...`) Globale Unicast-Adressen. Diese werden weltweit von Routern weitergeleitet
`ff00::/8` $\Longrightarrow$ (`ff...`) Multicast-Adressen
`2001:db8::/32` $\Longrightarrow$ Adressen nur zu Dokumentationszwecken

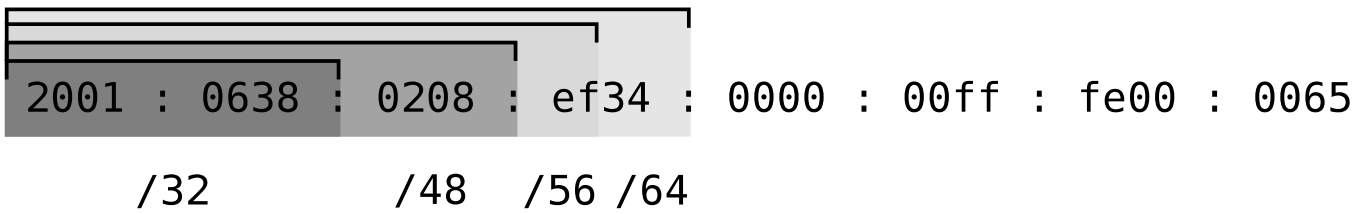
Struktur von IPv6-Adressen und Netzen (4/5)

- Umwandlung einer MAC-Adresse in das Modifizierte EUI-64 Adressformat
  1. Die MAC-Adresse wird in 2 je 24 Bits lange Teile unterteilt
     - Der 1.Teil bildet die ersten 24 Bits
     - Der 2.Teil die letzten 24 Bits der modifizierten EUI-64-Adresse
  2. Die freien 16 Bits in der Mitte der EUI-64-Adresse erhalten folgendes Bitmuster: `1111 1111 1111 1110` (Hex: `FFFE`)
  3. Abschließend wird der Wert des siebten Bits von links invertiert



Struktur von IPv6-Adressen und Netzen (5/5)

- (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-Netz 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**

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 Bytes langen Präfix: `0:0:0:0:0:FFFF::/96`

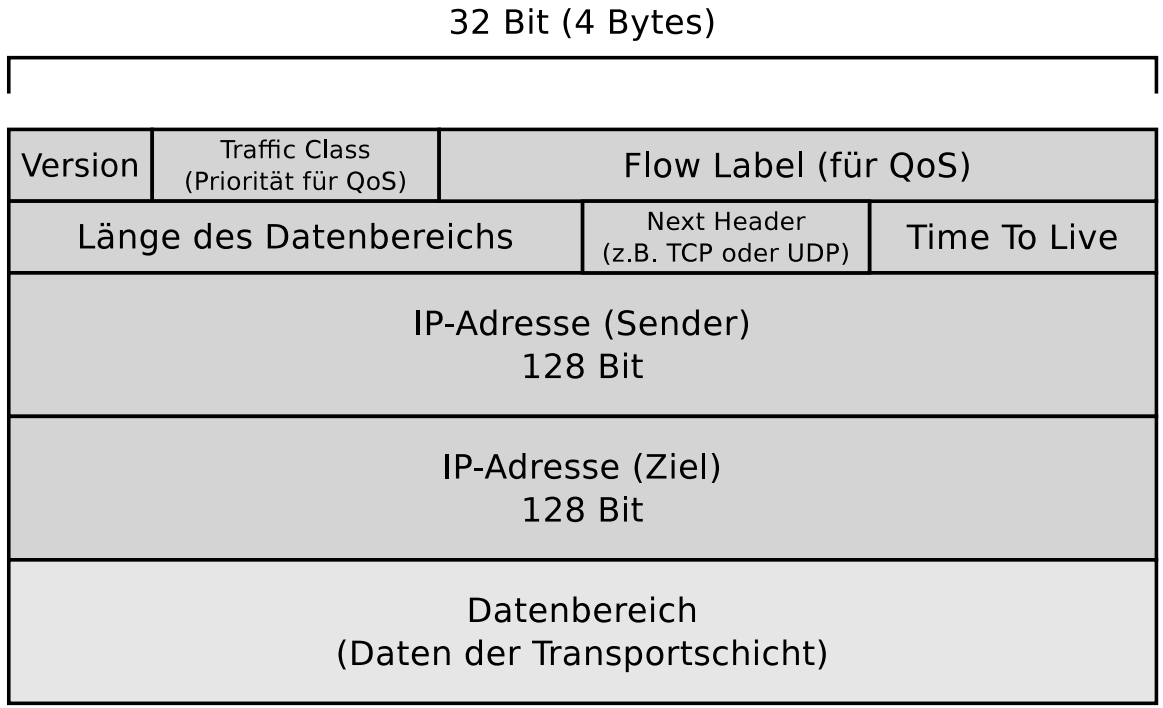
80 Bits					16 Bits	32 Bits
0000	0000	0000	0000	0000	FFFF	IPv4-Adresse

- 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 $\Longrightarrow$ 40 Bytes)



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

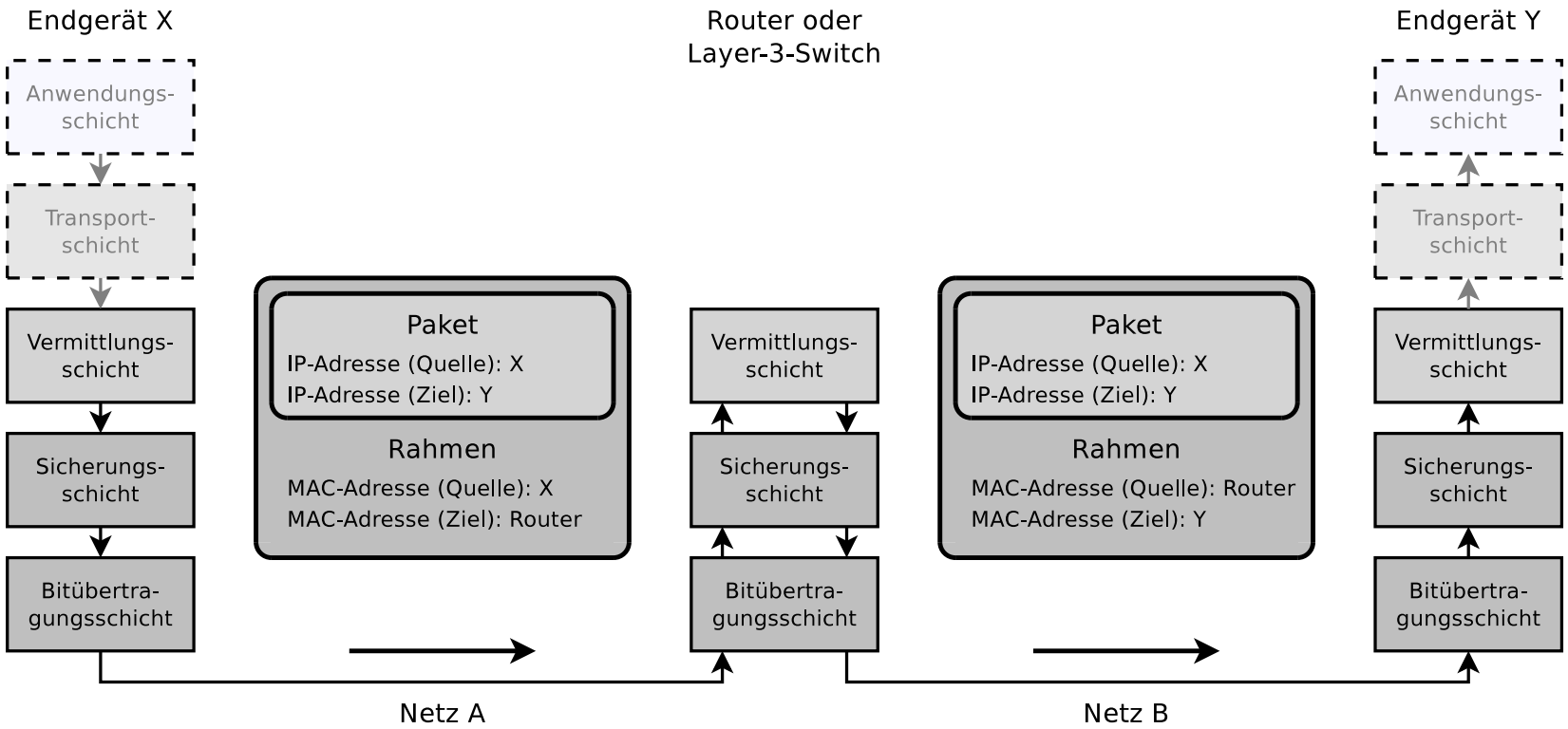
Konzept: Vereinfachte (reduzierte) Paketstruktur und gleichzeitig können zusätzliche (neue) Funktionen durch eine Kette von Erweiterungs-Kopfdaten (*Extension Headers*) hinzugefügt werden

Extension Headers

- **Hop-By-Hop Options** (Typ 0, RFC 2460)
  - Enthält Informationen, die alle IPv6-Geräte auf dem Weg zum Ziel beachten müssen
- **Routing** (Typ 43, RFC 2460)
  - Weg des Paketes durch das Netzwerk kann hier beeinflusst werden
- **Fragment** (Typ 44, RFC 2460)
  - Steuert das Zusammensetzen von zuvor fragmentierten IP-Datenpaketen
- **Encapsulating Security Payload** (Typ 50, RFC 4303)
  - Daten zur Verschlüsselung des Pakets
- **Authentication Header** (Typ 51, RFC 4302)
  - Enthält Informationen, um die Vertraulichkeit des Pakets sicherzustellen
- **No Next Header** (Typ 59, RFC 2460)
  - Platzhalter, um das Ende eines Header-Stapels anzuzeigen
- **Destination Options** (Typ 60, RFC 2460)
  - Optionen, die nur vom Zielrechner des Paketes beachtet werden müssen

Netzübergreifende Kommunikation (1/6)

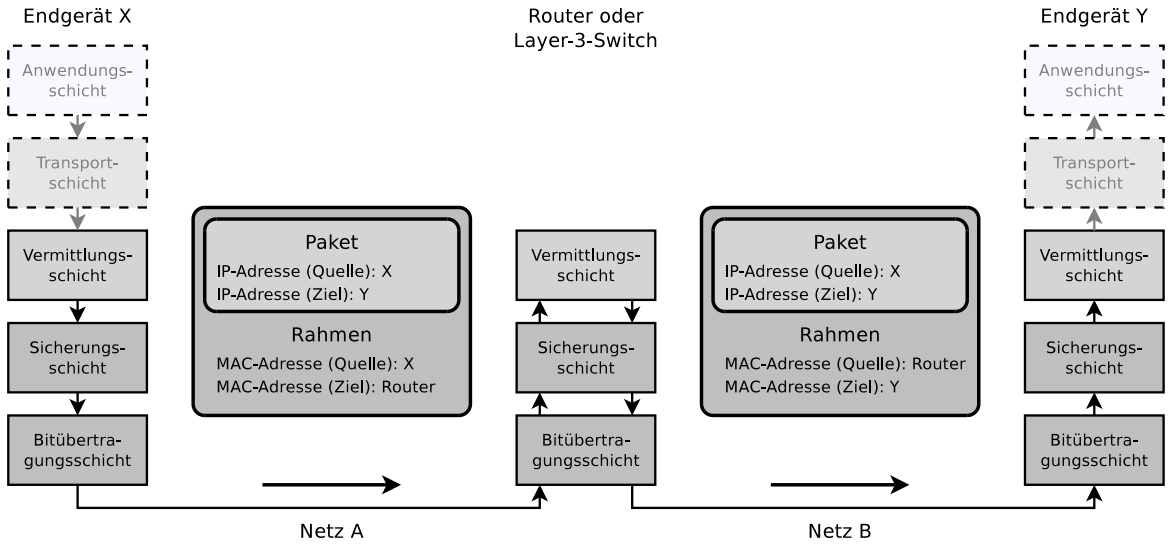
- **Internetworking** = Kommunikation zwischen Netzwerkgeräten mit Protokollen der Sicherungsschicht und Vermittlungsschicht über Netze, die auf unterschiedlichen Vernetzungstechnologien basieren können
- Denkbares Szenario für Internetworking



Netzübergreifende Kommunikation (2/6)

In diesem Szenario haben alle Kommunikationspartner öffentliche IP-Adressen

- X will ein IP-Paket an Y senden
  - Dafür muss X die **logische Adresse** (IP-Adresse) von Y kennen



Sie wissen bereits. . .

Für die Weiterleitung auf der Sicherungsschicht ist zudem die **physische Adresse** (MAC-Adresse) nötig

- X berechnet die Subnetznummern
  - Netzmaske_X `AND` IP-Adresse_X = Subnetznummer des eigenen Netzes
  - Netzmaske_X `AND` IP-Adresse_Y = Subnetznummer des Netzes von Y

Netzübergreifende Kommunikation (3/6)

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

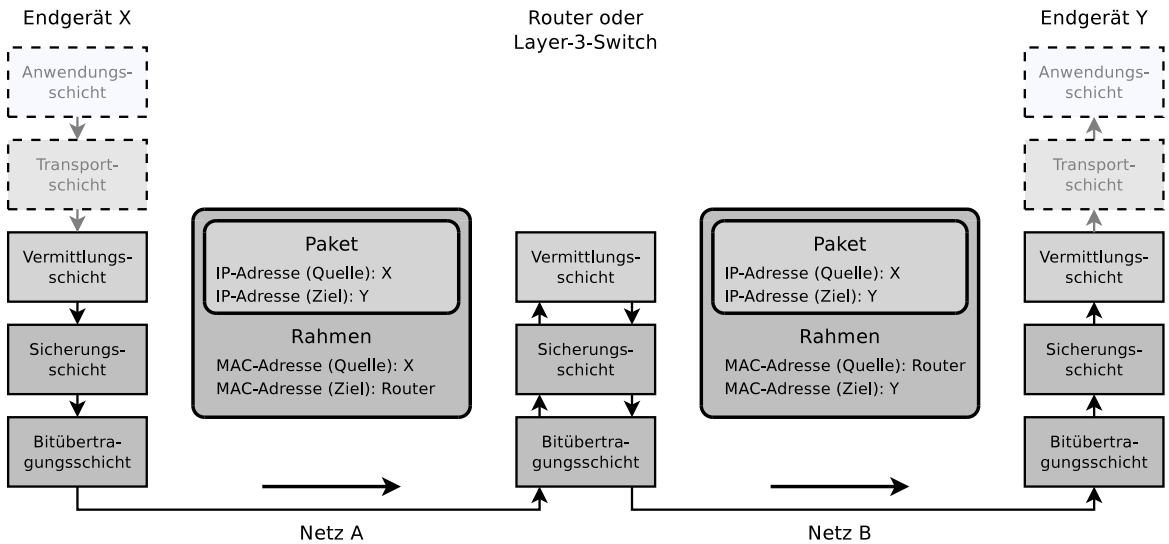
Sie wissen bereits. . .

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

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

Sie wissen bereits. . .

Befinden sich 2 Kommunikationspartner im gleichen logischen und physischen Netz, kann der Sender via Adressauflösung mit ARP die MAC-Adresse von Empfängers erfahren ($\Longrightarrow$ Foliensatz 9)



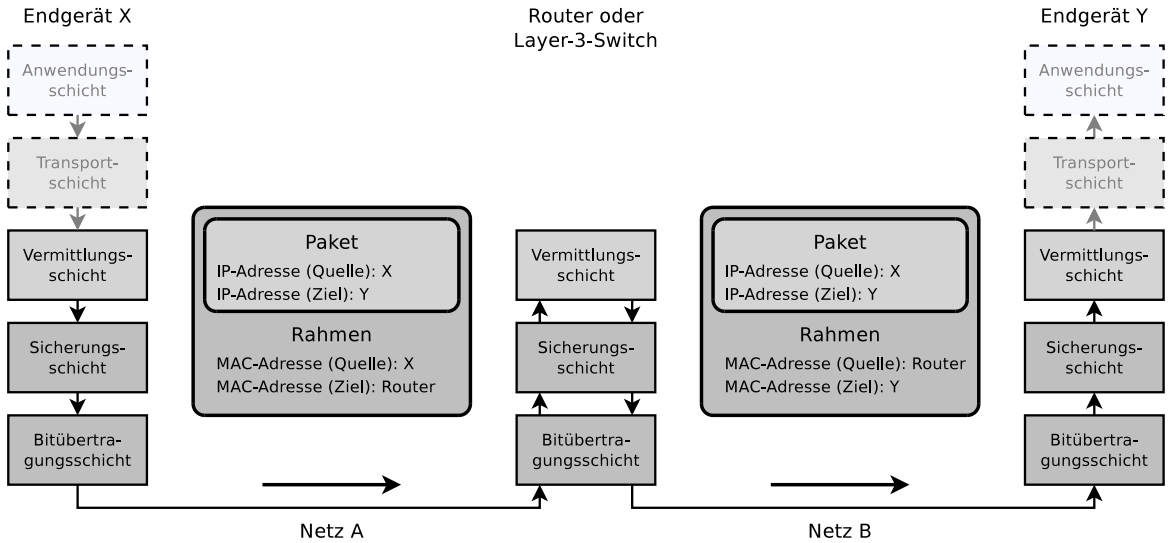
- **Hier handelt es sich um Kommunikation über logische und physische Netzgrenzen hinweg**

Netzübergreifende Kommunikation (4/6)

Sie wissen bereits...

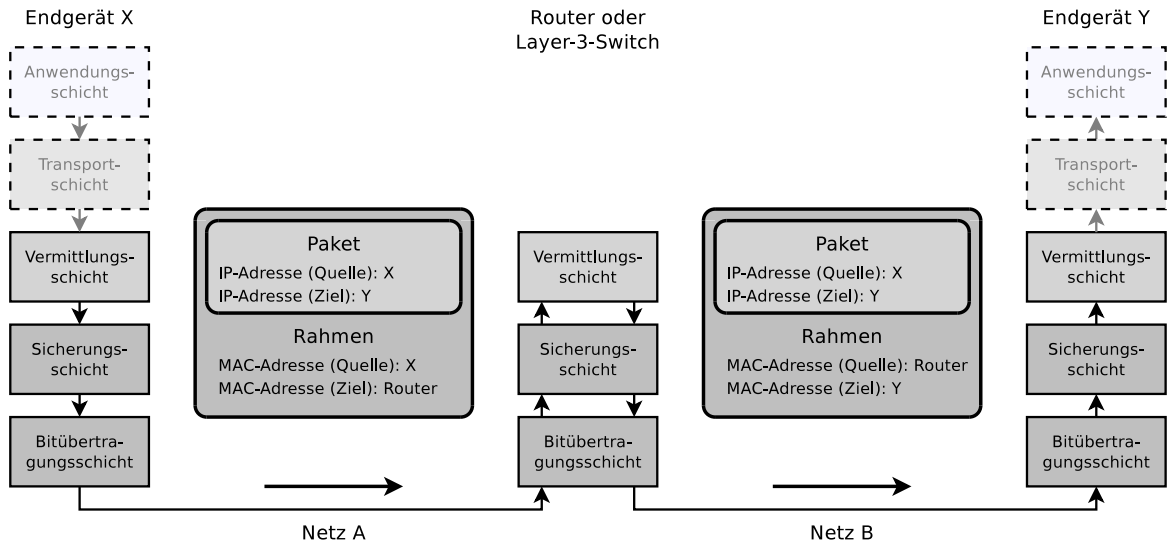
- ARP ist nur für die Auflösung der MAC-Adressen im lokalen physischen Netz zuständig
- Grund: ARP-Anfragen werden in Rahmen der Sicherungsschicht gesendet
- Das Feld mit der Zieladresse enthält die Broadcast-Adresse
- Solche Rahmen werden von Bridges und Switches nicht weitergeleitet
  $\Longrightarrow$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



Netzübergreifende Kommunikation (5/6)

- Der Router empfängt das IP-Paket
  - Er ermittelt mit seiner lokalen Routing-Tabelle, die alle ihm bekannten logischen Netze enthält, die korrekte Schnittstelle für die Weiterleitung des Pakets
- Der Router ist über eine seiner Schnittstellen mit dem physischen Netz verbunden ist, über das auch Y erreichbar ist

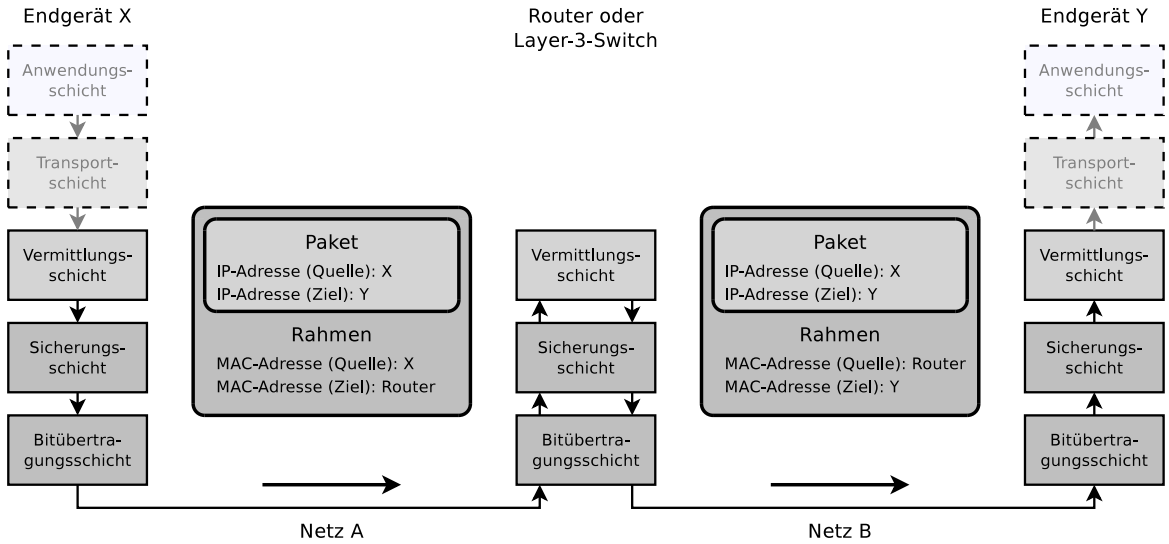


- Der Router ermittelt die MAC-Adresse von Y via Adressauflösung mit ARP
- Der Router verpackt das IP-Paket in einem Rahmen
  - Das Feld mit der Senderadresse enthält die MAC-Adresse des Routers
  - Das Feld mit der Zieladresse enthält die MAC-Adresse von Y

Netzübergreifende Kommunikation (6/6)

- Möglicherweise ist die maximale Paketlänge (*Maximum Transmission Unit*) von Netz B kleiner als die von Netz A
  - Dann kann es abhängig von der Größe des weiterzuleitenden IP-Pakets nötig sein, dass der Router das empfangene Paket in mehrere kleinere Pakete fragmentiert (*wegen Zeitmangel gestrichen*)

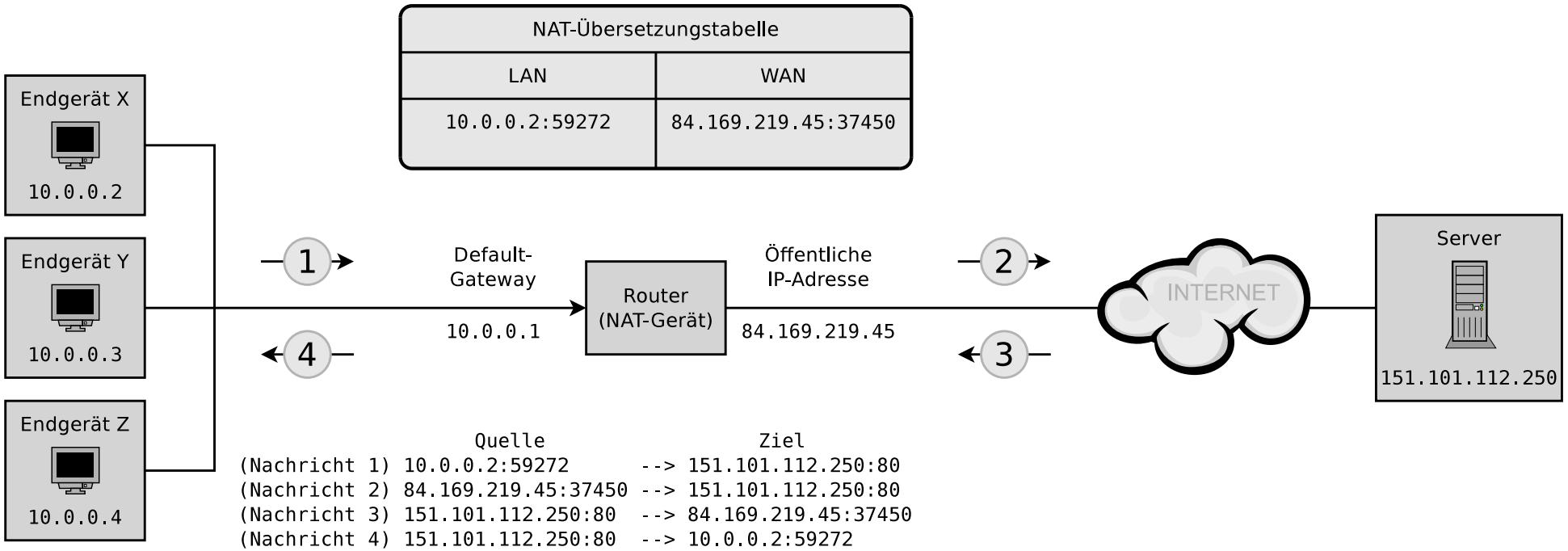
- Die IP-Adressen von Sender (X) und Empfänger (Y) im IP-Paket werden bei der Weiterleitung nicht verändert



Network Address Translation (1/5)

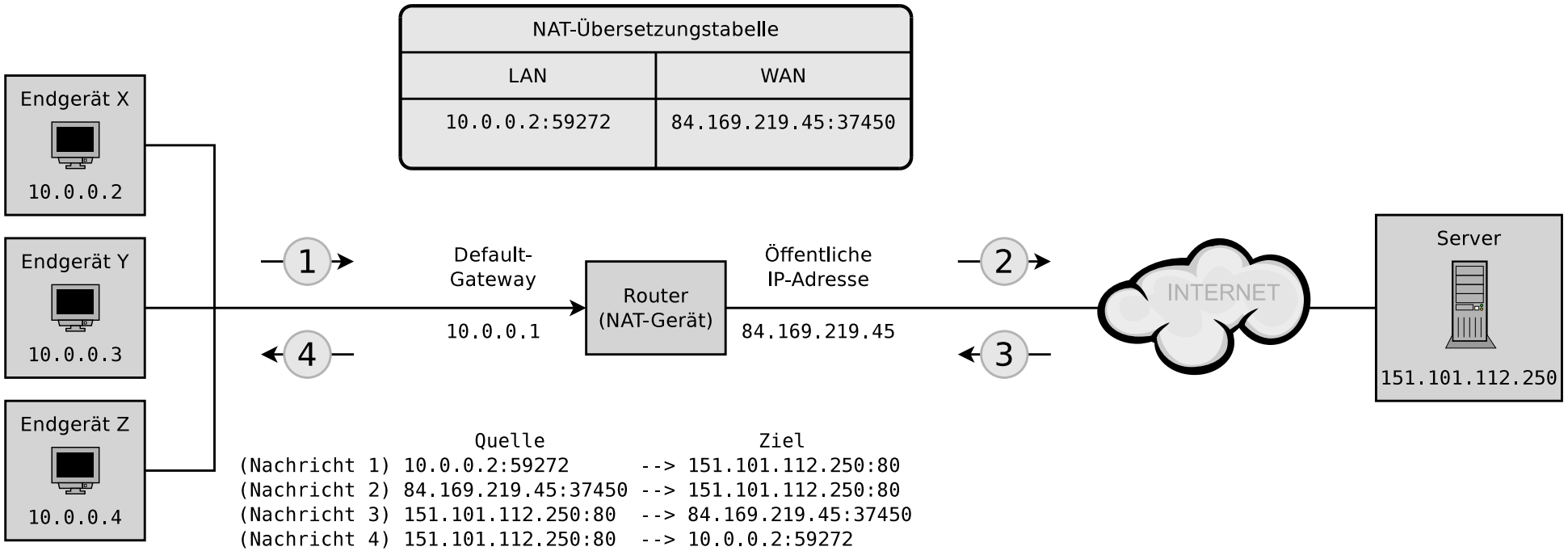
- Problem: Die allerwenigsten Haushalte, Unternehmen und Bildungs-/Forschungseinrichtungen haben genug öffentlich erreichbare IPv4-Adressen, um alle ihre Netzwerkgeräte mit eigenen IPs auszustatten
  - Darum verwenden lokale Netze meist einen privaten IPv4-Adressraum (siehe Folie 23)
  - Problem: Wie 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 (2/5)



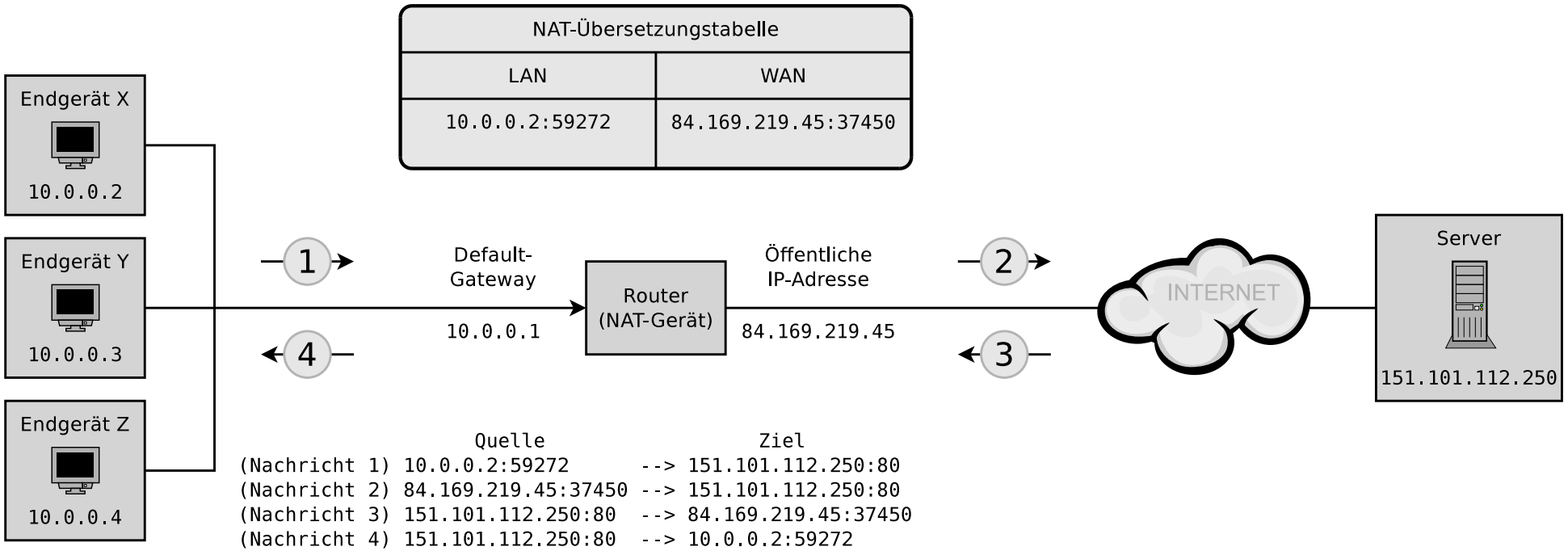
- Die Clients X, Y und Z befinden sich in einem Netz mit einem privaten IP-Adressbereich
- Nur der Router hat eine global erreichbare IP-Adresse
  - Er wirkt für die Außenwelt nicht wie ein Router, sondern wie ein Netzwerkgerät mit einer einzelnen öffentlich registrierten IP-Adresse

Network Address Translation (3/5)



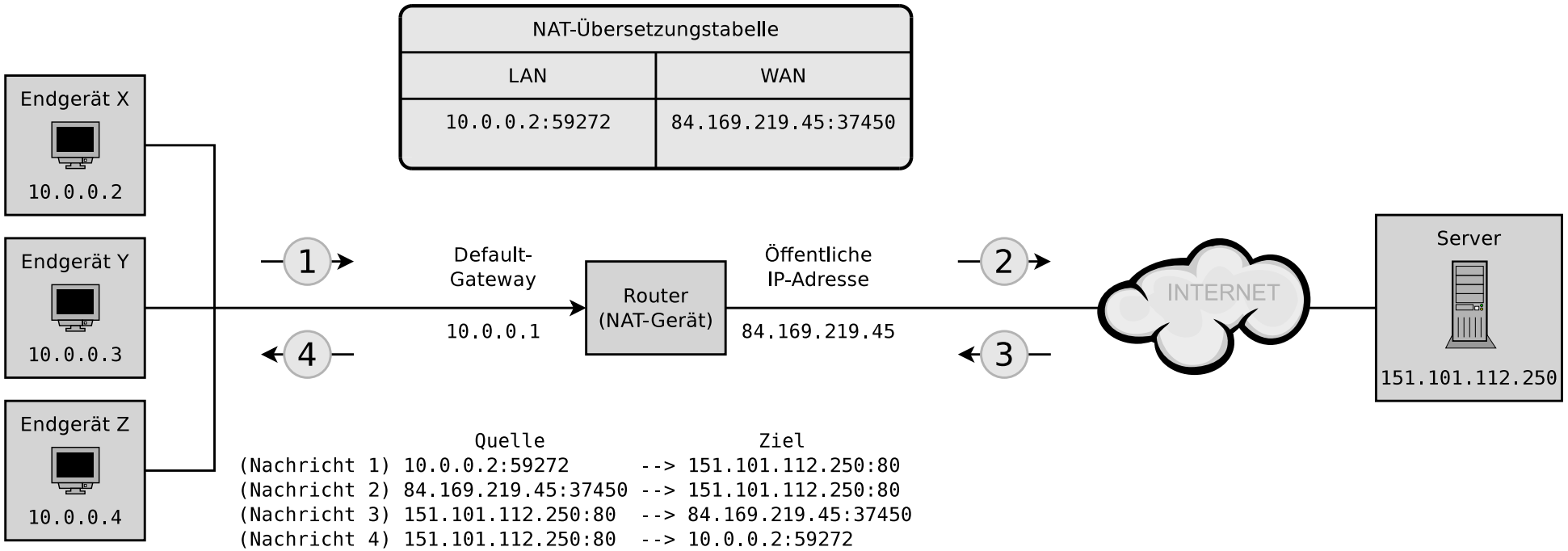
- Client X fordert eine Webseite vom Server an
  - Die Anfrage (Nachricht 1) enthält als Quelladressen die IP-Adresse und Portnummer von X und als Zieladressen die IP-Adresse und Portnummer des Servers
- Der Router ersetzt in der weitergeleiteten Anfrage (Nachricht 2) die IP und Portnummer des Clients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

Network Address Translation (4/5)



- Die Zuordnungen zwischen den Ports des Routers und den zugehörigen Netzwerkgeräten im lokalen Netz speichert der Router in einer **NAT-Übersetzungstabelle** (*NAT Translation Table*)
- Die Antwort des Servers (Nachricht 3) ist an den Router adressiert
  - Dieser ersetzt die Adressinformationen entsprechend der Tabelle und leitet die Antwort an X weiter (Nachricht 4)

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)