

Folien zur Veranstaltung Rechnernetze in der AI Wintersemester 2018 (Teil 5)

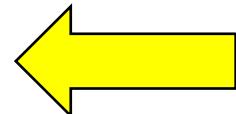
Prof. Dr. Franz Korf
Franz.Korf@haw-hamburg.de

Basierend auf der RN Vorlesung von M. Hübner

Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6
- Paketfilterung (Firewalls)
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung



Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

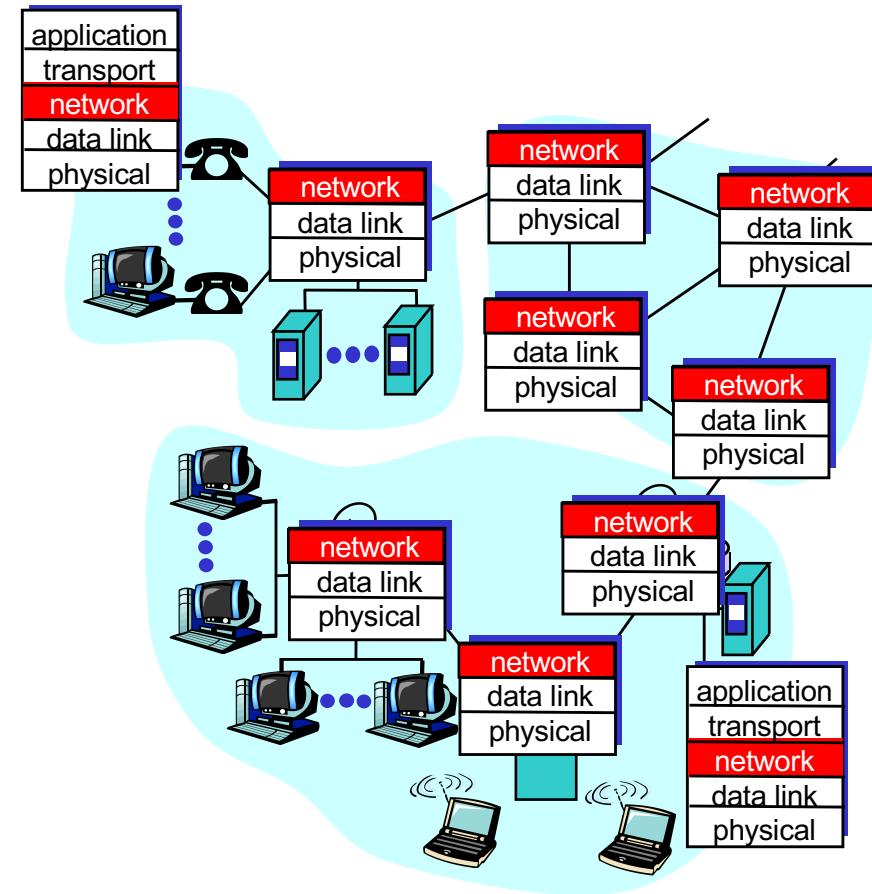
J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

Funktionen der Netzwerkschicht

- Transport der Pakete vom sendenden zum empfangenden Host

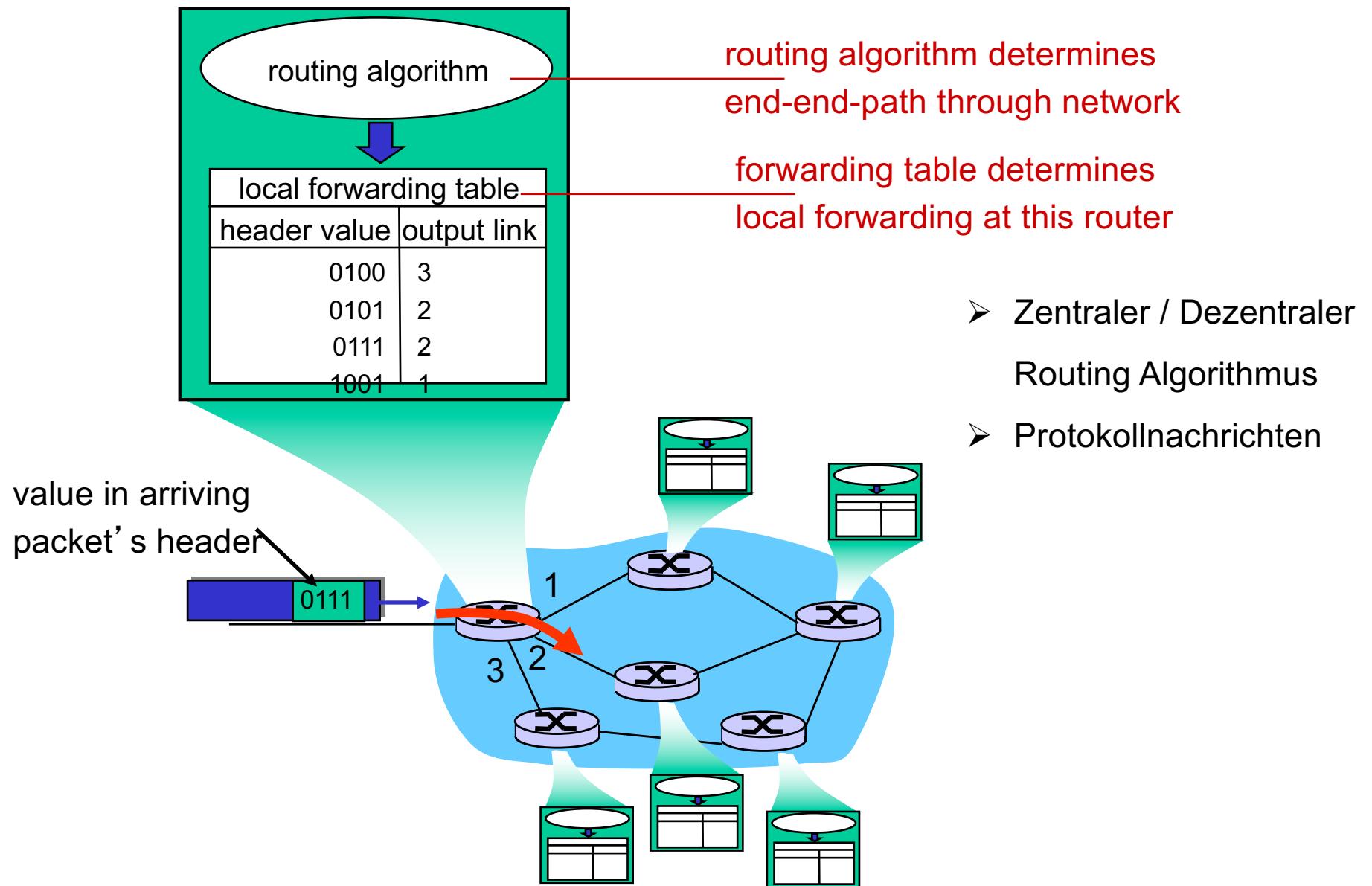
Drei wichtige Aufgaben:

- **Pfadbestimmung:** Bestimme den Weg (Route), den die Pakete von der Quelle zum Ziel laufen → Routing-Algorithmen
- **Switching:** Transportiere Pakete vom Eingang des Routers zum richtigen Ausgang
- **Verbindungsauftbau:** Einige Netzwerkarchitekturen benötigen die "Einrichtung" eines Pfades durch die Router vor dem Datenfluss



Netzwerkschichtprotokolle laufen in jedem Router und Host

Zusammenspiel Forwarding & Routing



Dienstmodell der Netzwerkschicht

Welches **Dienstmodell** gibt es für den “Kanal”, durch den Pakete vom Sender zum Empfänger transportiert werden?

Netzwerk-architektur	Dienst- modell	Band- breiten- garantien	Garantie der Verlust- freiheit	Reihen- folge	Zeit- garantien	Hinweis auf Überlast
Internet	Best Effort	Keine	Nein	Beliebige möglich	Nicht unter- stützt	Keiner
ATM	CBR	Garantiert eine kons- tante Rate	Ja	In korrek- ter Rei- henfolge	Unterstützt	Überlast tritt nicht auf
ATM	ABR	Garantier- tes Mini- mum	Nein	In korrek- ter Rei- henfolge	Nicht unter- stützt	Überlasthin- weise werden verwendet

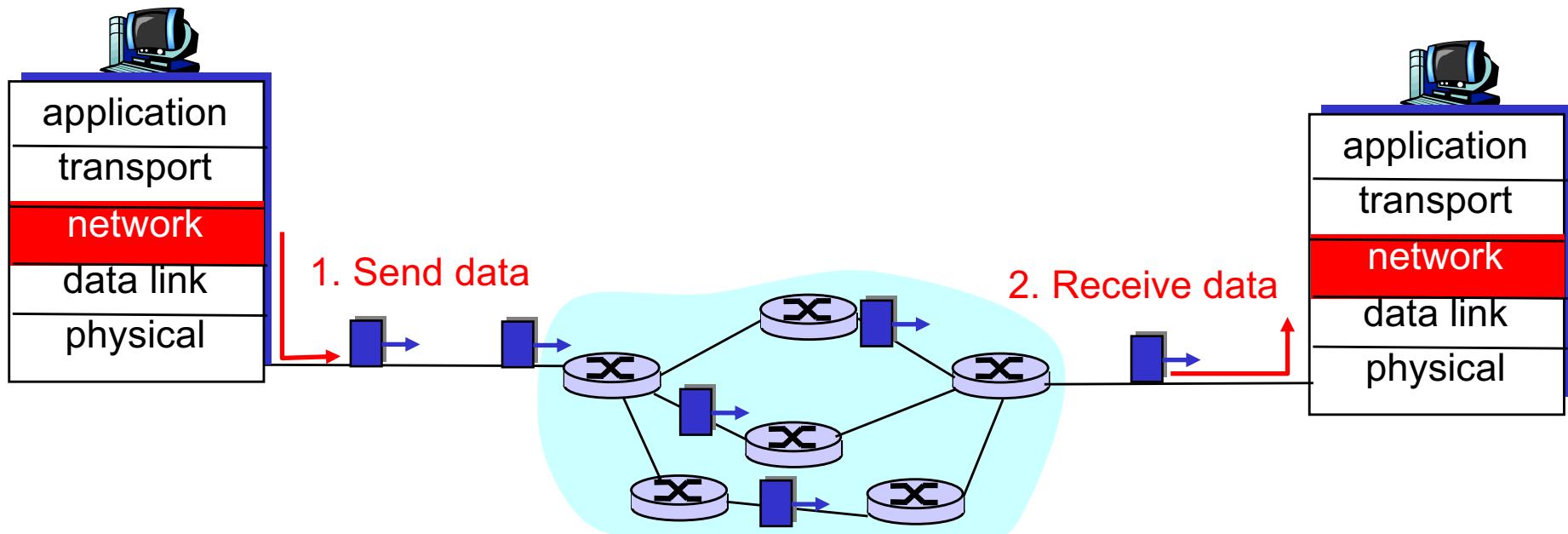
Tabelle 4.1: Dienstmodelle des Internets, von ATM CBR und von ATM ABR.

Verbindungslose und Verbindungsorientierte Dienste

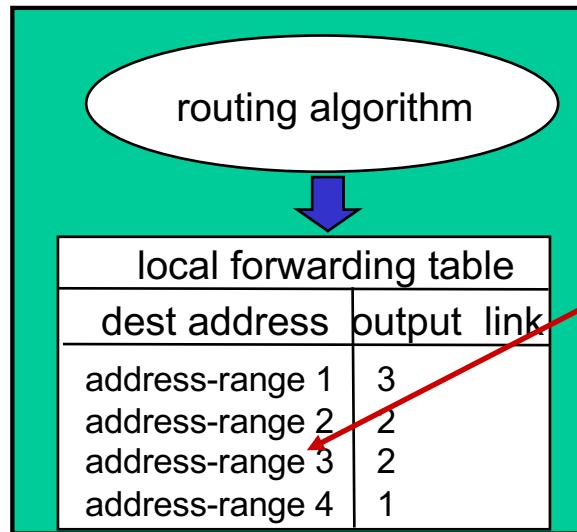
- Verbindungslose – und Verbindungsorientierte Dienste von der Transportschicht bekannt.
- Transportschicht: Prozess-zu-Prozess Dienst
Netzwerkschicht: Host-zu-Host Verbindungen
- Große Computernetzwerkarchitekturen (ATM, Internet, Frame Relay, ...) bieten entweder verbindungslose - oder verbindungsorientierte Dienste an, aber nicht beides
- Transportschicht: Verbindungsaufbau ist nur auf den Hosts implementiert
Netzwerkschicht: Verbindungsaufbau ist auf den Hosts und Routern implementiert

Datagramm-Netzwerke: das klassische Internet-Modell

- **Verbindungslos:** Kein Verbindungsaufbau auf der Netzwerkebene
- Router: kein “Zustand” über die Ende-zu-Ende-Verbindungen
 - kein “Verbindungskonzept” auf der Netzwerkebene
- Pakete werden typischerweise durch Verwendung der Zieladresse geroutet
 - Pakete zwischen dem gleichen Quelle-Ziel-Paar können unterschiedliche Wege durchs Netz laufen

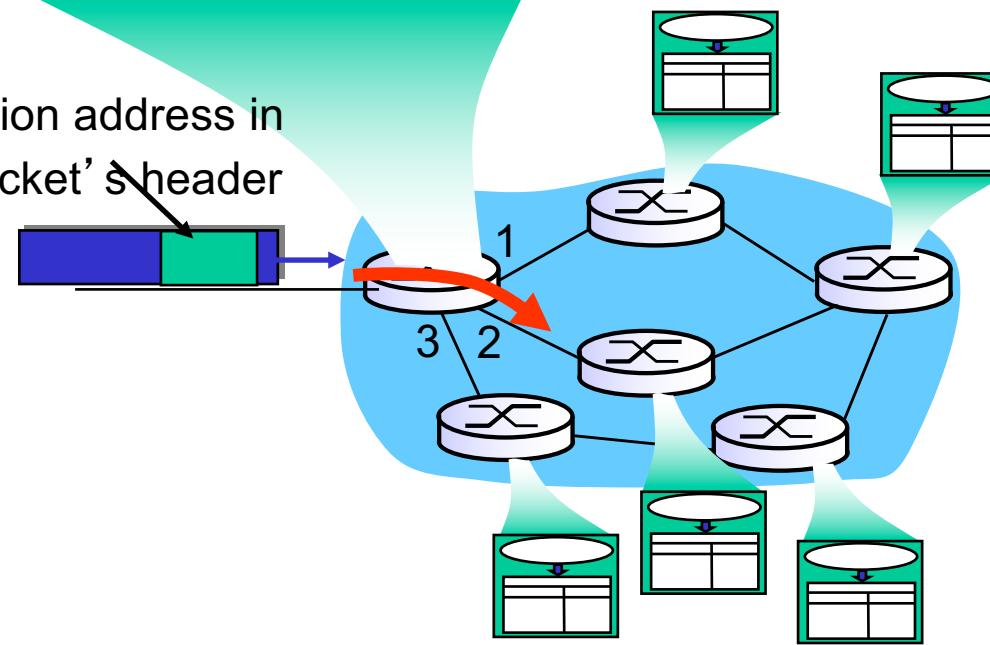


Datagram forwarding Tabelle



4 billion IP addresses,
so rather than list
individual destination
address
list range of addresses
**(aggregate table
entries)**

IP destination address in
arriving packet's header



Weiterleiten anhand
der **Zieladresse** des
Datagramms

Datagram forwarding Tabelle

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Hier Spezialfall: Eindeutige Zuordnung Adresse – Range.

In der Netzwerk-Praxis nicht gegeben (s. nächste Folie)

Longest prefix matching

- Longest Prefix Matching: Beim Zugriff auf die Forwarding Table wird der Eintrag zu der Zieladresse verwendet, die den längsten matching Prefix hat.

Destination Address Range	Link interface
11001000 00010111 00010**** *****	0
11001000 00010111 000110000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

Beispiele

Zieladr: 11001000 00010111 00010110 10100001 which interface?

Zieladr: 11001000 00010111 00011000 10101010

Virtuelle Kanäle ("Virtual Circuits" – VC-Netzwerke)

- “Quelle-Ziel-Pfad verhält sich wie eine klassische Telefonleitung”:
 - In Bezug auf die Performanz
 - und auf die Netzwerkaktionen auf dem Pfad von der Quelle zum Ziel
- **Verbindungsorientiert:** Verbindungsaufbau für jede Verbindung vor dem Transport der Daten (und Verbindungsabbau hinterher)
- Jedes Paket trägt die ID des virtuellen Kanals (VC) (nicht die Adresse des Zielhosts)
- Jeder Router auf dem Quelle-Ziel-Pfad speichert einen “Zustand” für jede durch ihn laufende Verbindung
- Ressourcen der Verbindung (Übertragungskapazität, Puffer) können für den VC *reserviert* werden
 - um ein Verhalten zu erhalten, das dem einer festen Leitung entspricht
- Beispiel: ATM
- VC auf Sicherungsschicht: AVB, AFDX

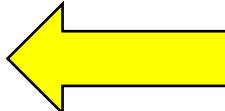
Vergleich Datagramm- / VC-Netzwerke

	Datagramm-Netzwerk	VC-Netzwerk
Verbindungsaufbau	Nicht erforderlich	Erforderlich
Adressierung	Jedes Paket enthält die volle Quell- und Zieladresse	Jedes Paket enthält eine kurze VC-Nummer
Zustandsinformation	Router führen keine Zustandsinformationen	Für jede virtuelle Verbindung ist ein Tabelleneintrag erforderlich
Routing	Jedes Paket wird unabhängig befördert	Die Route wird beim Aufbau der virtuellen Verbindung gewählt; alle Pakete folgen dieser Route
Wirkung von Routerfehlern	Nur Verlust einzelner Pakete	Alle virtuellen Verbindungen über den ausgefallenen Router werden beendet
Dienstgüte-Garantie	Schwierig	Einfach, wenn ausreichende Ressourcen reserviert sind
Überlastkontrolle	Schwierig	Einfach, wenn ausreichende Ressourcen reserviert sind
Flexibilität	sehr hoch	gering (hoher Verwaltungs- und Abstimmungsaufwand)

Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6
- Paketfilterung (Firewalls)
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung



Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

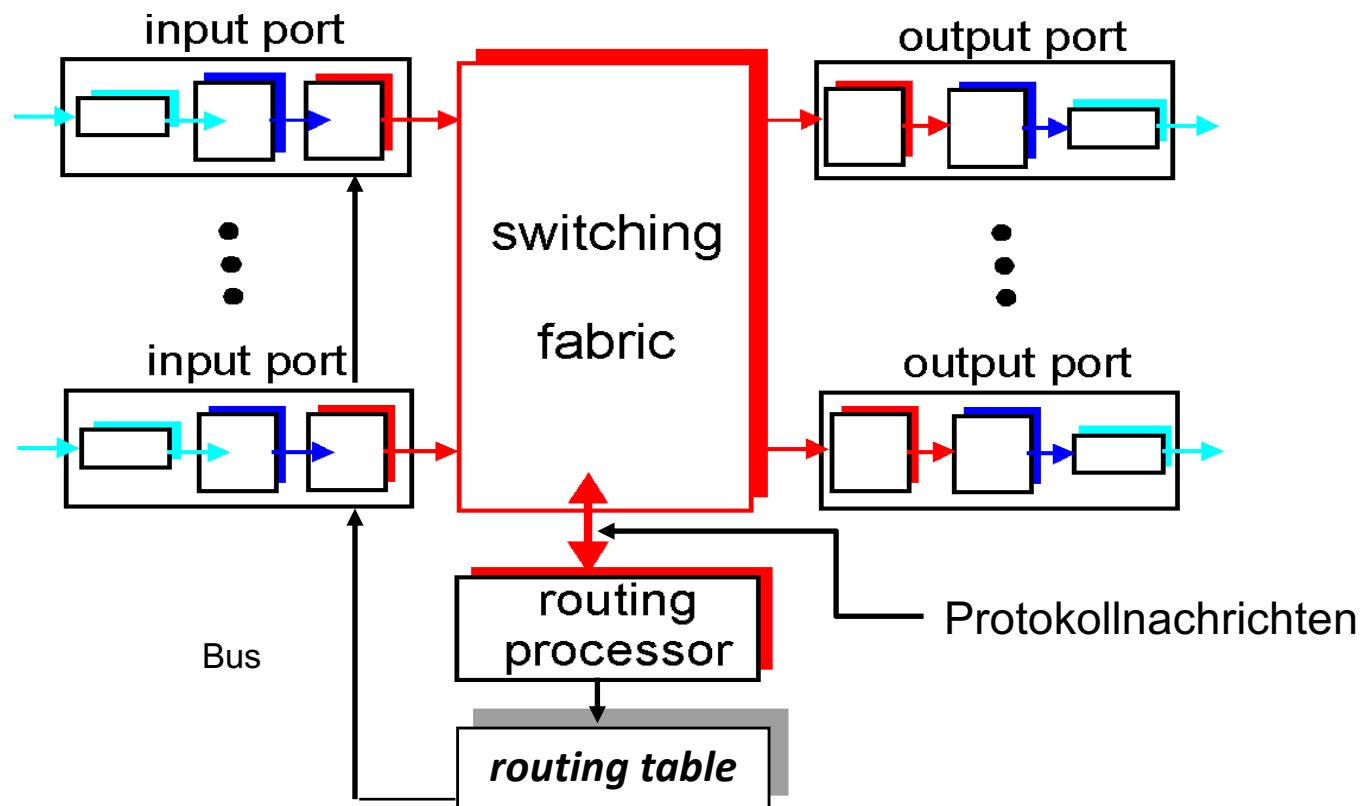
Folien und Abbildung teilweise aus:

J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

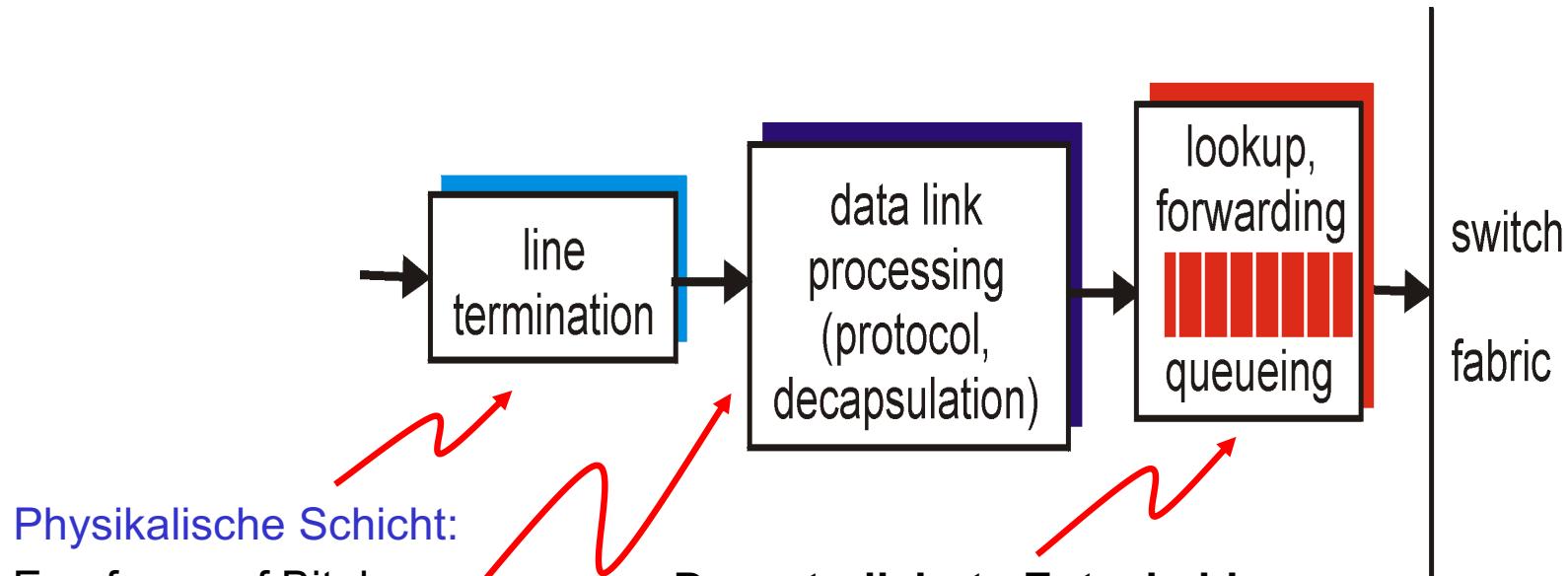
Router-Architektur: Überblick

Hauptfunktionen:

- **Pfadermittlung (“routing”):** Aktualisierung der “Routing-Tabelle”
 - Def. der Abbildung: Zieladresse → Ausgangsleitung
- **Weiterleitung (“forwarding”)** von Paketen (Datagrammen) von einer Eingangs- zu einer Ausgangsleitung

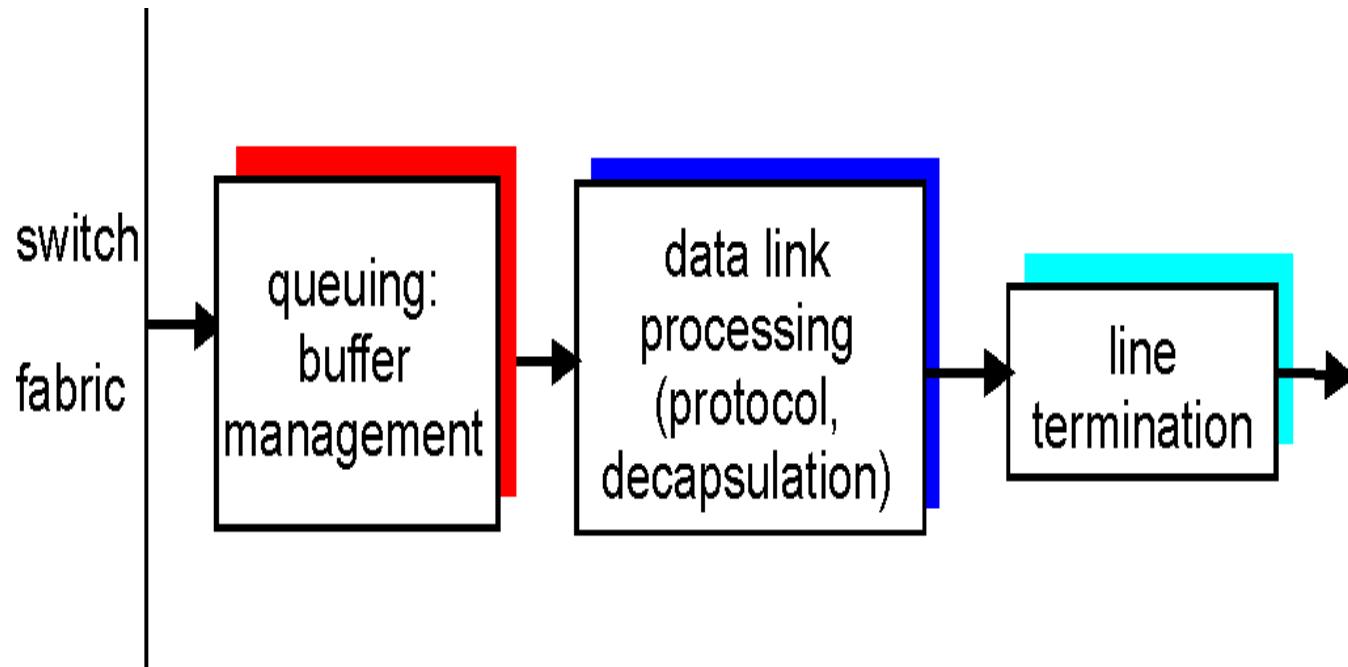


Input Port Funktionen



- Bestimme aufgrund einer „Kopie“ der **Routingtabelle** die Ausgangsleitung
- **Ziel:** Komplette Verarbeitung eines Datagramms innerhalb der Empfangszeit!
- **Warteschlange** (“queuing”): Nötig, wenn Datagramme schneller ankommen als sie in das Schaltnetz (“switch fabric”) eingestellt werden können

Output Port Funktionen



- Pufferung (“queuing”): Nötig, wenn Datagramme schneller aus dem Schaltnetz (“switch fabric”) ankommen als sie übertragen werden können
- Über eine Scheduling-Strategie muss das nächste zu übertragende Datagramm aus dem Puffer gewählt werden (FCFS, Prioritäten, ...)

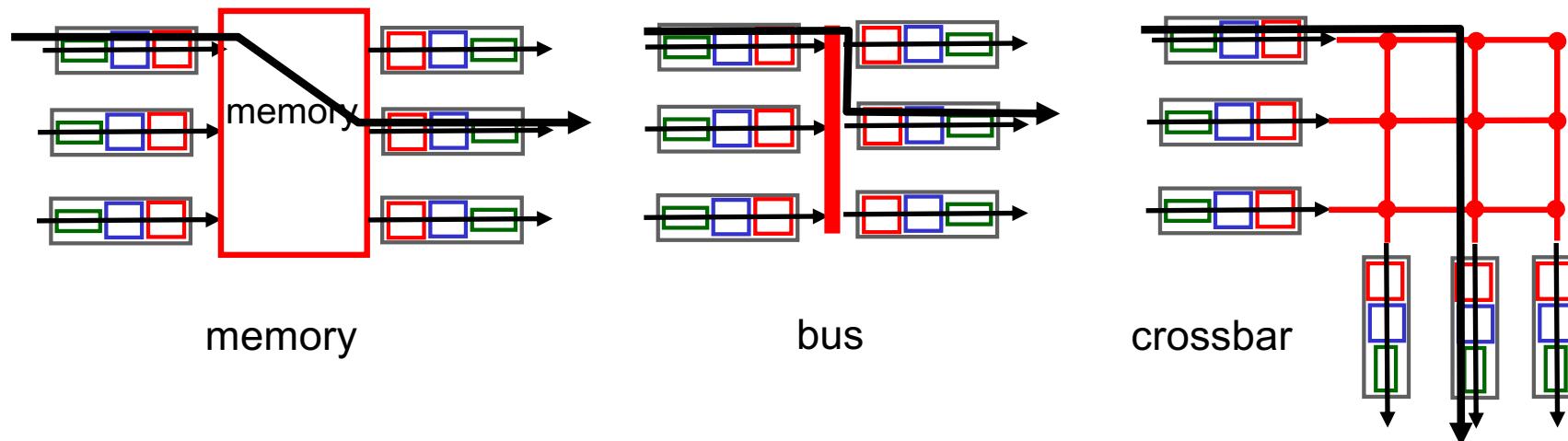
Output Port Funktionen

Situation: Queue ist voll. Welches Paket wird verworfen?

- **Drop-Tail:** Das ankommende Paket wird verworfen
- **AQM:** Active Queue Management
 - Idee: verwerfe oder markiere ein Paket, so dass der Sender von der Überlast erfährt, **bevor die Queue voll ist.**
 - Unter AQM versteht man unterschiedliche Strategien zu dieser Idee
- **RED** (Random Early Detection Algorithmus) (eine AQM Strategie)
 - Zwei Schwellwerte: min_{th} , max_{th} für die Queue Länge
 - Datagramm trifft ein und Aktuelle Queue Länge $< \text{min}_{\text{th}}$: Füge Paket ein
 - Datagramm trifft ein und Aktuelle Queue Länge $> \text{max}_{\text{th}}$: Verwerfe / markiere Paket
 - Datagramm trifft ein und Aktuelle Queue Länge in $[\text{min}_{\text{th}}, \text{max}_{\text{th}}]$: Verwerfe / markiere Paket mit einer Wahrscheinlichkeit, die von der durchschnittlichen Queue Länge und den Schwellwerten abhängt.

Switching Fabric

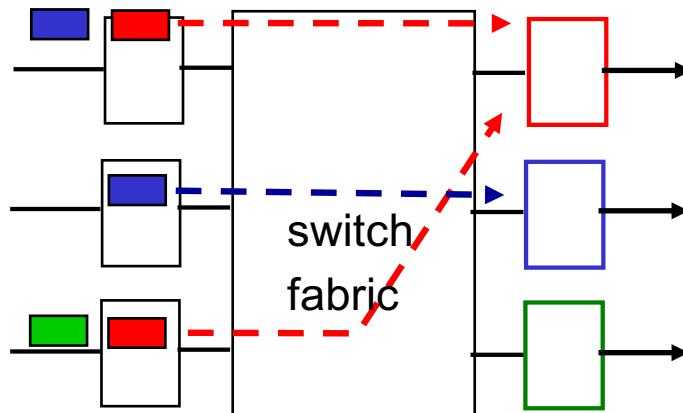
- **Aufgabe:** Übertragung eines Pakets von einem Input Puffer zum passenden Output Puffer
- Ist die Switching Rate (in der Regel eine Vielfache der Input Line Rate) zu klein, müssen Pakete bei Eingabeport gepuffert werden
- Drei typische Aufbauten einer Switching Fabric



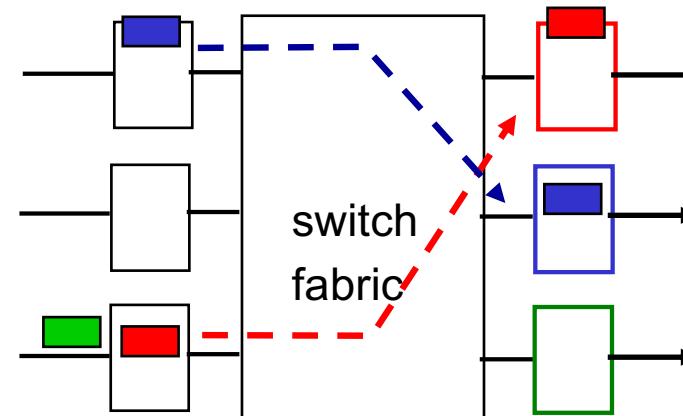
Head of Line (HOL) Blocking

Situation:

- Die Übertragungsrate der Switching Fabric ist langsamer als die (aktuelle) Ankunftsrate einer Line Card.
- Puffer am Input Port notwendig.
- HOL: Das vorderste Paket in der Queue versperrt den nachfolgenden Pakete den Zugang zur Switching Fabric



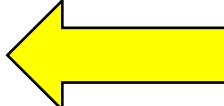
output port contention:
only one red datagram can be
transferred.
lower red packet is blocked



one packet time later:
green packet
experiences HOL
blocking

Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4) 
- NAT vs. IPv6
- Paketfilterung (Firewalls)
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung

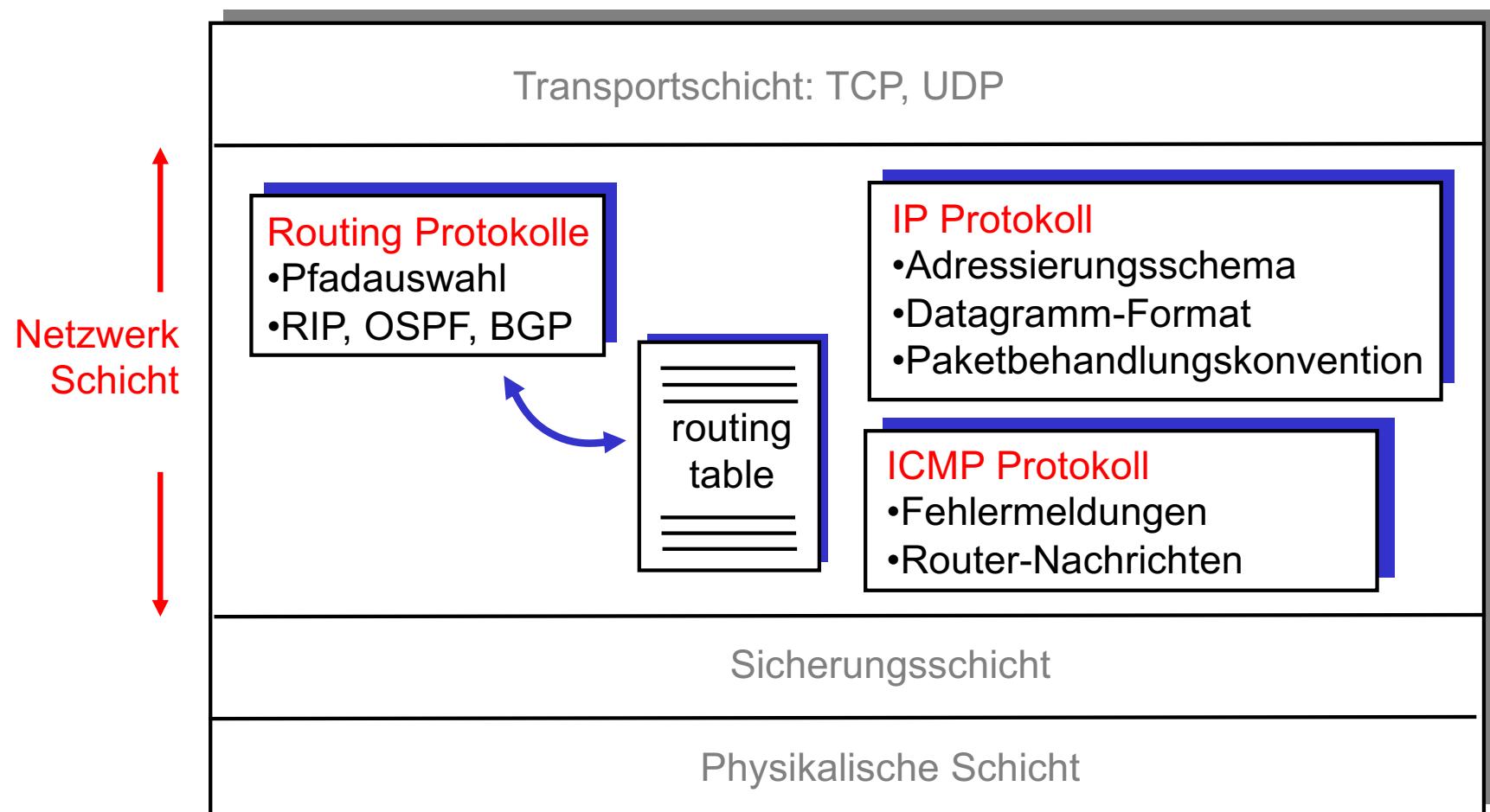
Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

Funktionen der Netzwerkschicht:

- Auf Hosts und Routern implementiert

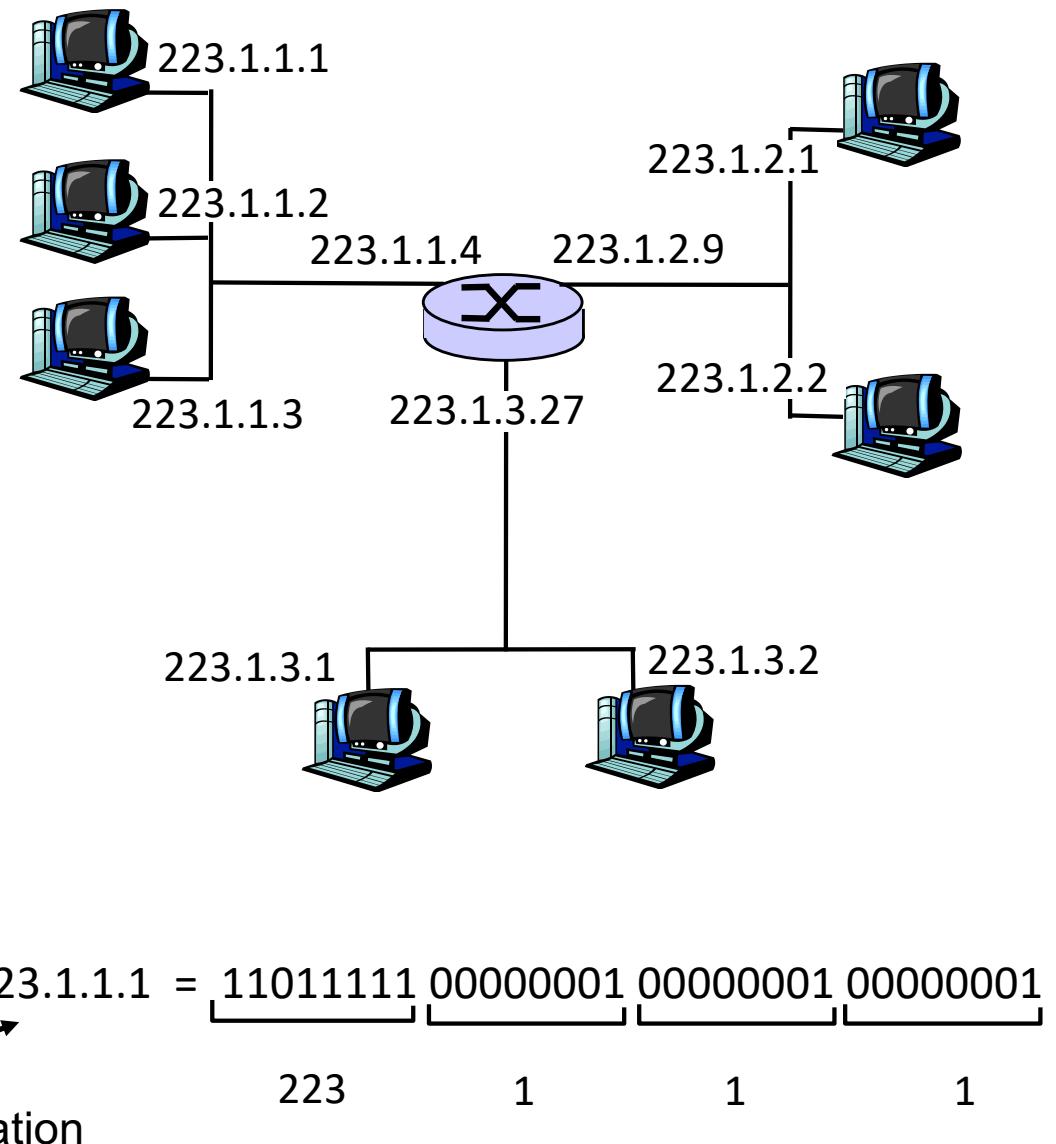


Einführung in die IPv4 Adressierung

IP-Adresse: 32-bit ID für Host- und Router-Interface

Interface: Schnittstelle zwischen Host/Router und physikalischer Verbindungsleitung

- Router haben viele Interfaces
- Hosts können mehrere Interfaces haben
- IP-Adressen werden einem **Interface** (nicht Host oder Router) zugewiesen



IPv4 Adressierung & IP Netzwerke

IP-Adresse:

- Netzwerk-Teil (high order bits)
- Host-Teil (low order bits)

Was ist ein Netzwerk? (aus IP-Perspektive)

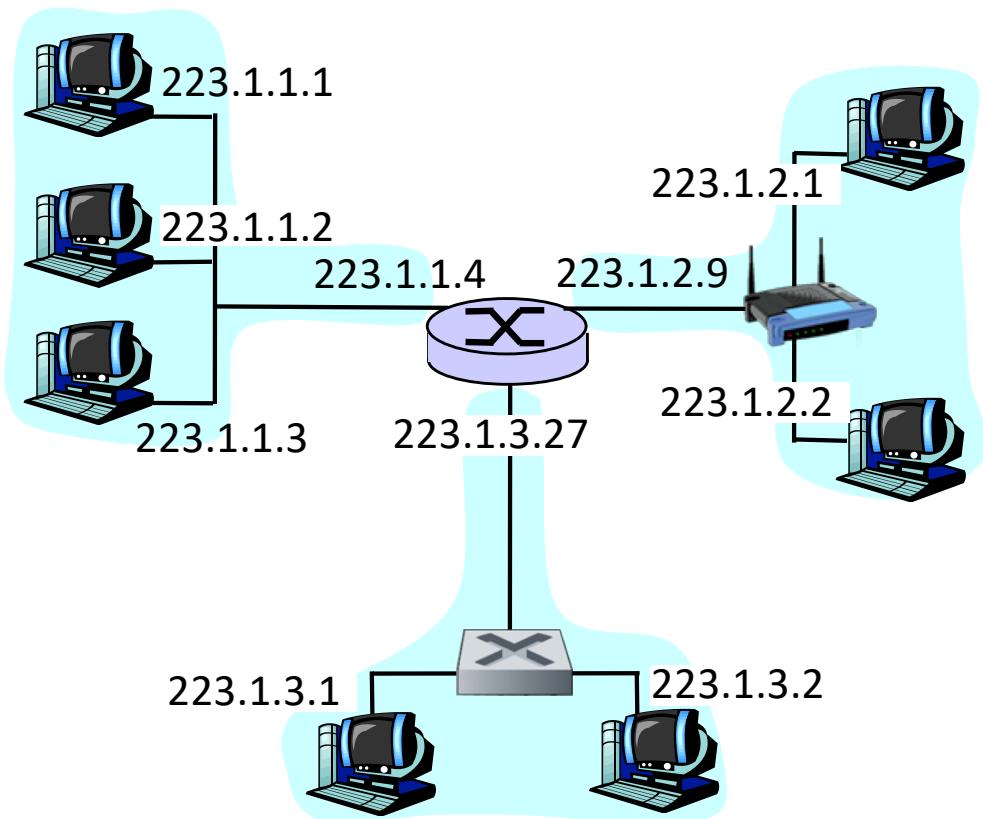
- Interfaces mit identischem Netzwerk-Teil der IP-Adresse,
- die sich physikalisch gegenseitig **ohne Inanspruchnahme eines Routers** erreichen können



Switched Ethernet



WiFi Basis Station

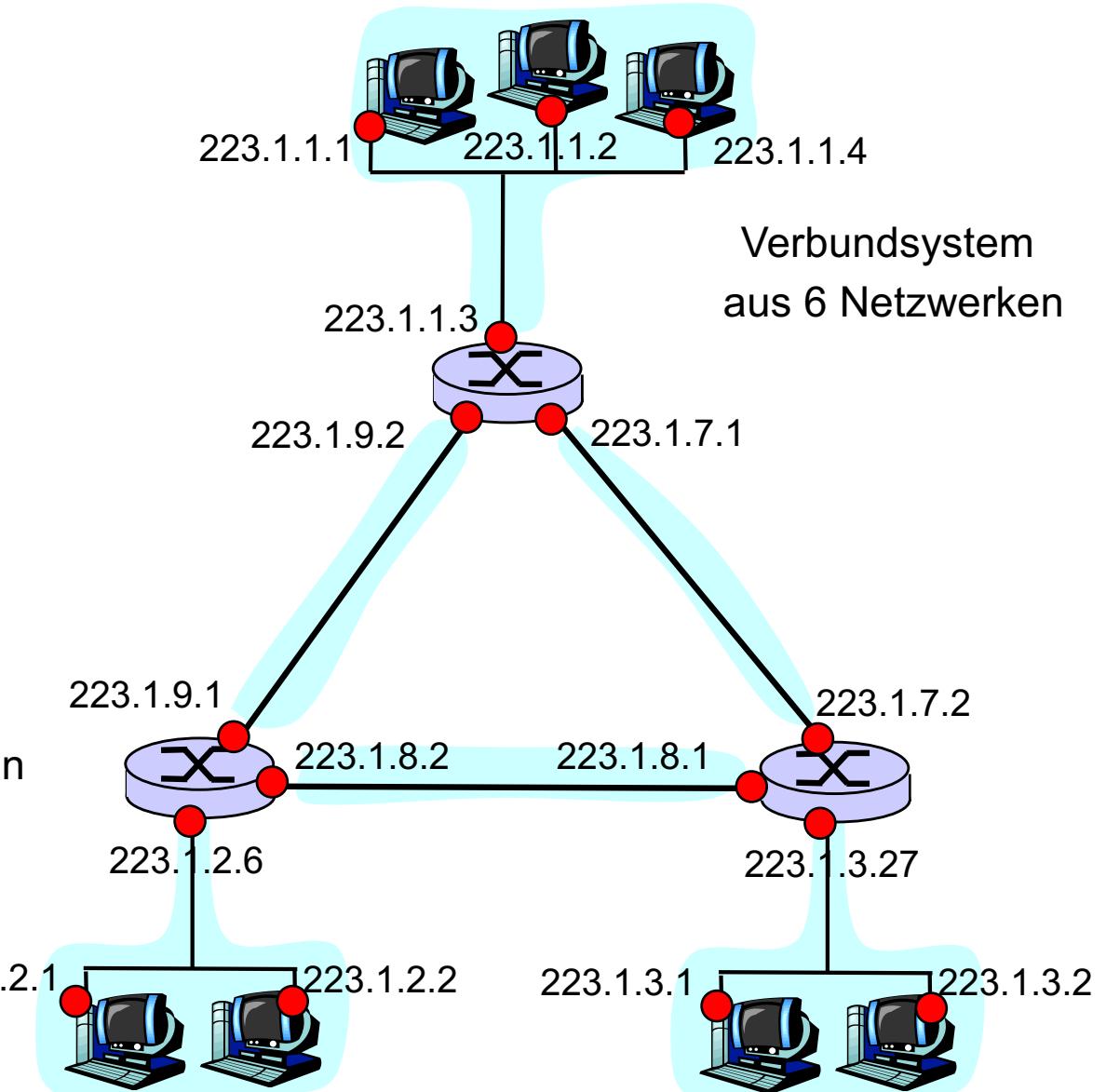


Netzwerk bestehend aus 3 IP-Netzwerken
(die ersten 24 Bit einer IP-Adresse sind **hier** der Netzwerk-Teil)

Definition von IP Netzwerken

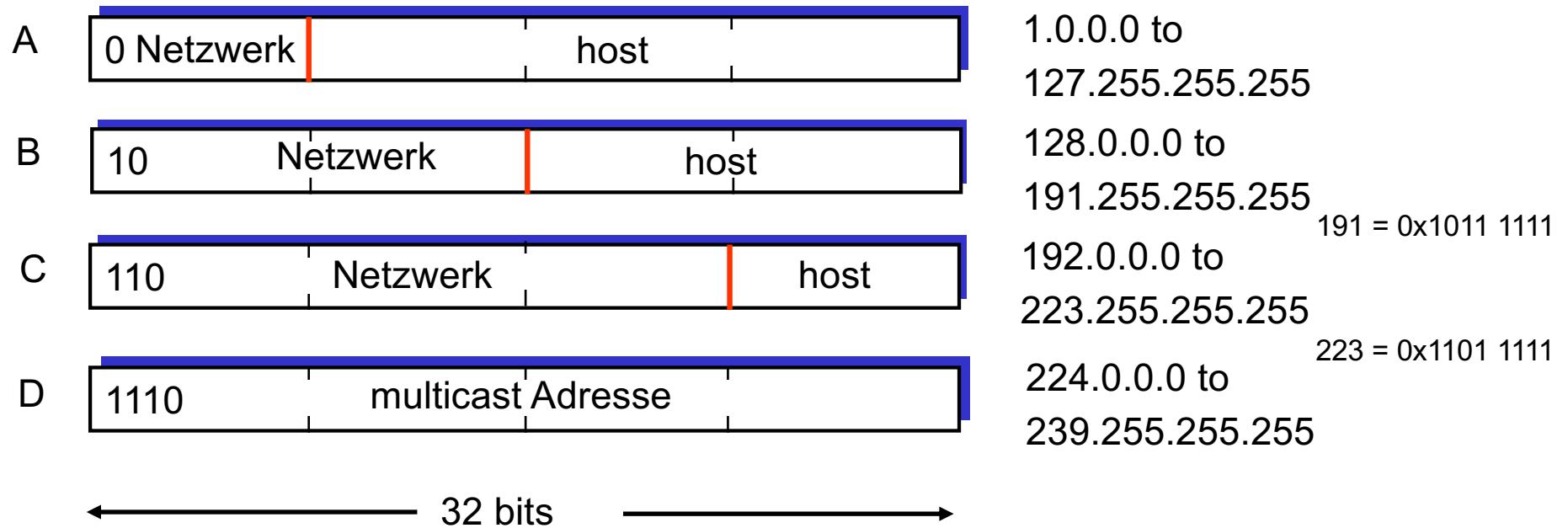
Bestimmung / Definition von IP-Netzwerke

- Jedes Interface wird von seinem **Router/Host** abgekoppelt
- So entstehen “Inseln” mit isolierten IP-Netzwerken
- Die Interfaces der Router sind Endpunkte, über die ein IP-Netzwerk mit anderen verbunden werden kann



Strukturierung des IP-Adressraums: Adressklassen (klassisch)

Klasse



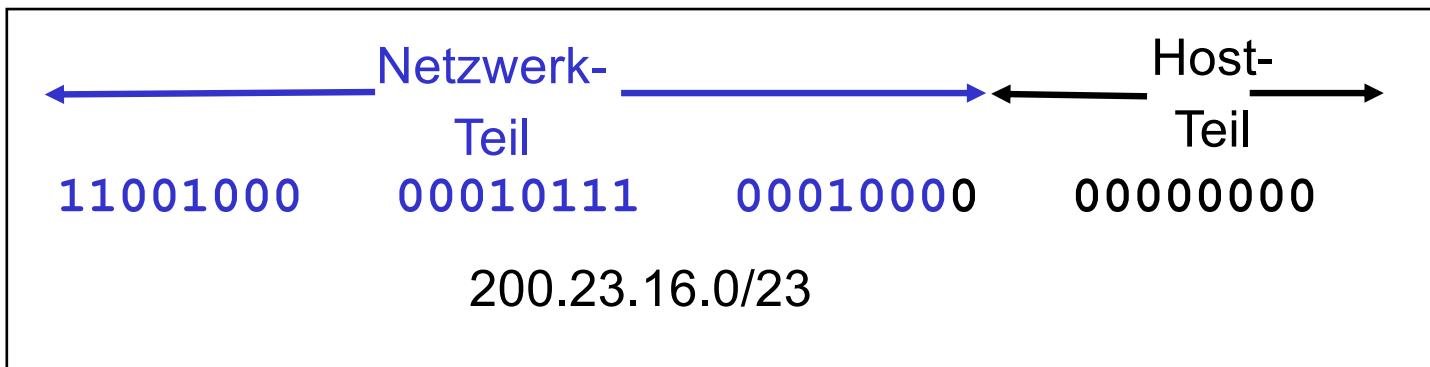
Ineffiziente Nutzung des Adressraums!

- Beispiel: Ein Klasse-B-Netz belegt 65.536 Adressen, auch wenn in einer Firma nur 2.000 genutzt werden
- Oftmals ist ein Class C Adresse zu klein und eine Class B Adresse zu groß
- **Wird nicht mehr eingesetzt.**

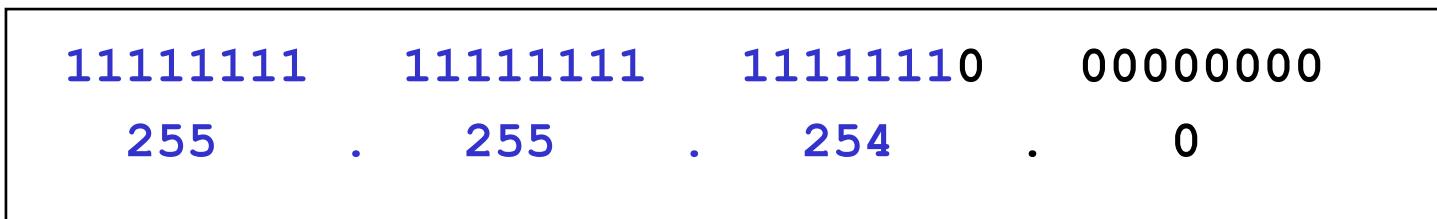
Strukturierung des IP-Adressraums: CIDR (aktuell)

CIDR: Classless InterDomain Routing

- Netzwerk-Teil einer IP-Adresse kann von *beliebiger* Länge sein
 - Adressformat: **a.b.c.d/x**, wobei x die Anzahl der Bits im Netzwerk-Teil der Adresse darstellt



- Aufteilung Netzwerkteil/Hostteil wird auch über eine **Subnetzmaske** angegeben



Vergabe von IP-Adressen

ICANN: Internet Corporation for Assigned Names and Numbers

- “Politische” Oberorganisation
- Vergabe von IP-Adressbereichen (→ Netzwerkteil) an ISPs und große Organisationen
- Verwaltung von DNS-Top-Level-Domains und Betrieb der DNS-Root Server
- Delegation der technischen Durchführung:
- **IANA: Internet Assigned Numbers Authority**
 - „Technische“ Zentralorganisation
 - Delegation der regionalen Zuständigkeit:
 - **ARIN** (American Registry for Internet Numbers)
 - **RIPE** (Réseaux IP Européens)
 - **APNIC** (Asia Pacific Network Information Centre)

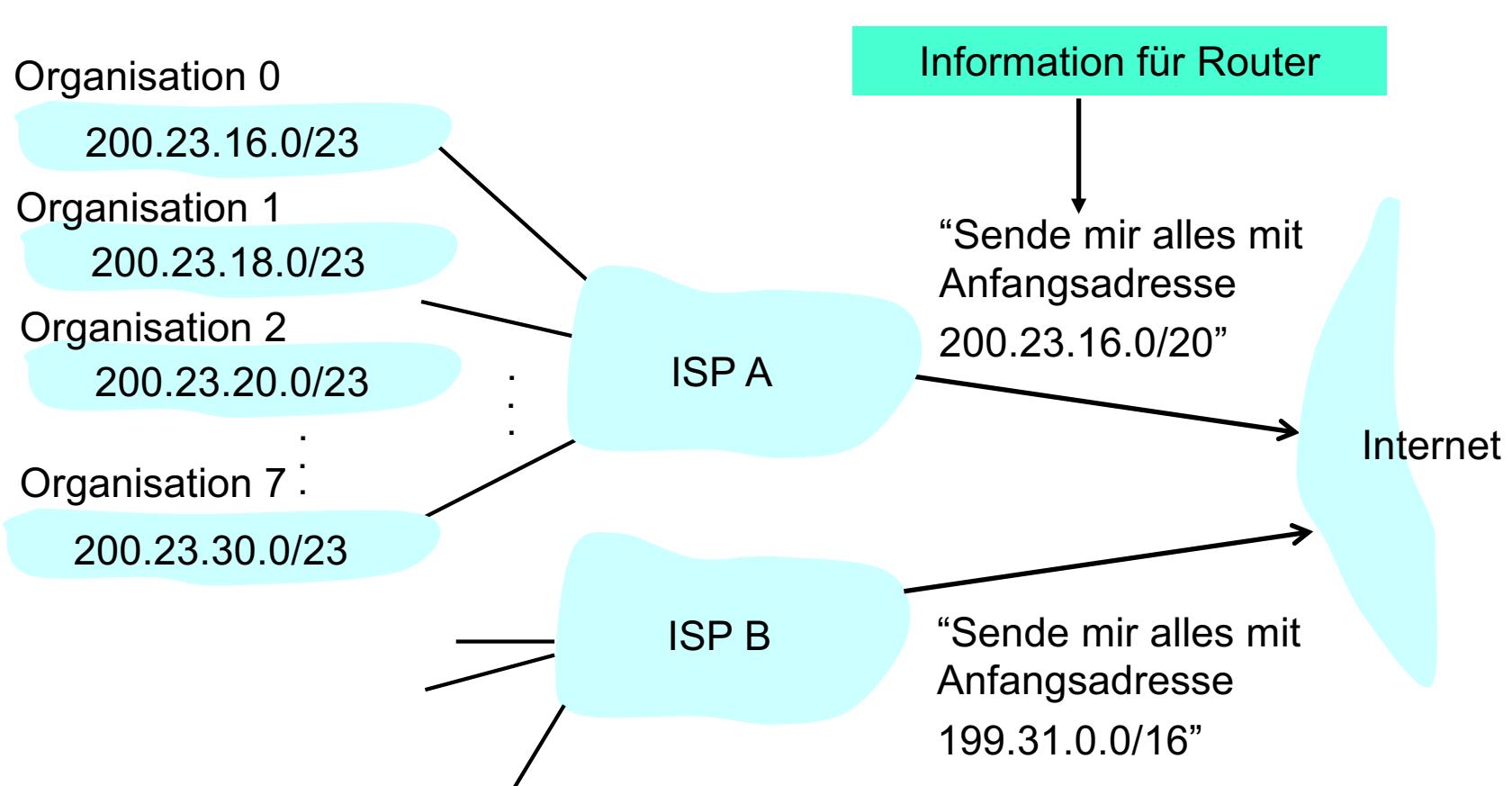
Weitergabe von IP-Adressen durch ISPs

- Ein ISP kann seinen zugewiesenen Adressbereich untergliedern (indem er den Netzwerk-Teil erweitert) und damit Subnetze an Organisationen weitergeben (**“Subnetting”** RFC 950)



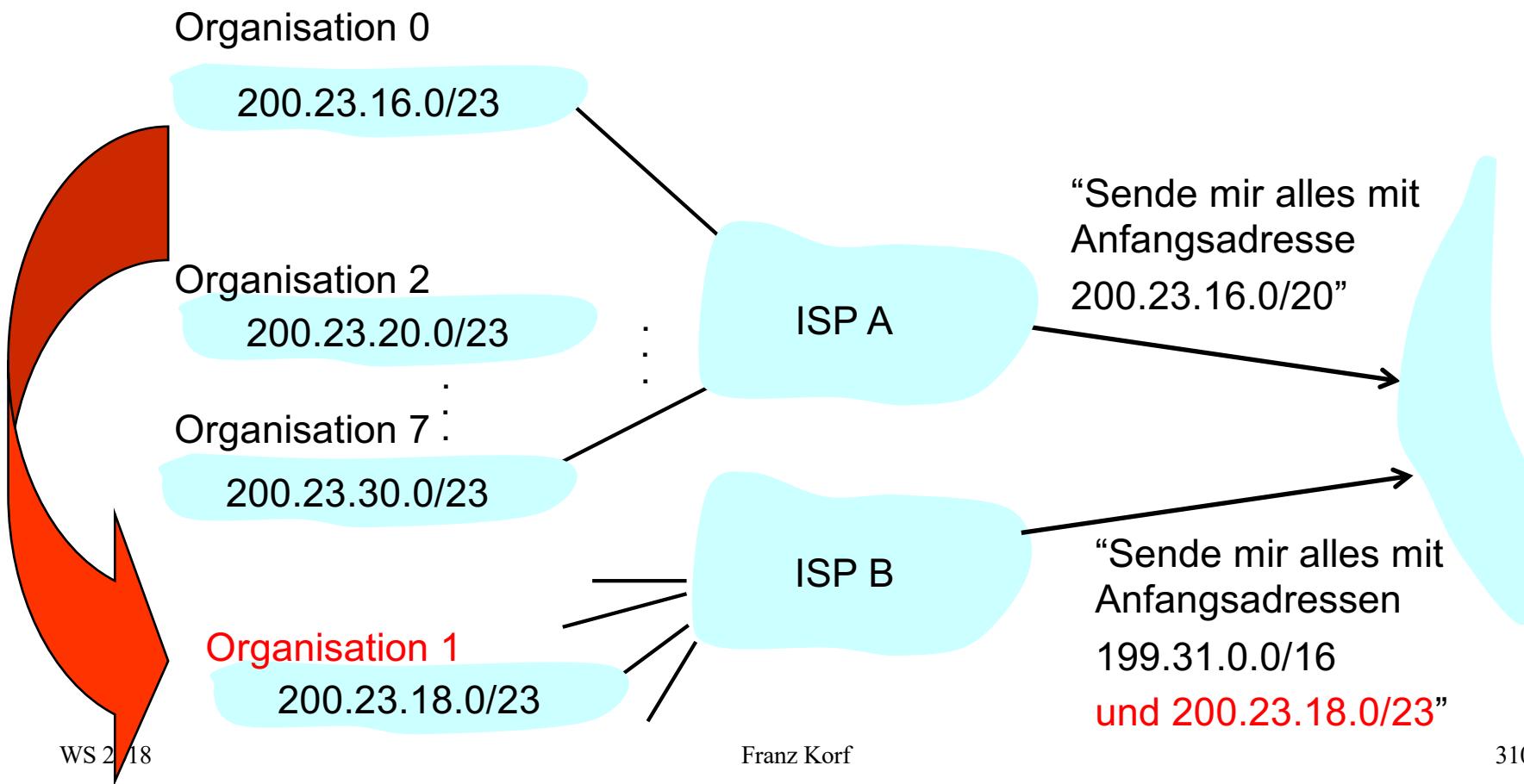
Hierarchische Adressierung: Adress/Routenaggregation

Vereinfachung von **Routingtabellen** durch Zusammenfassung von Subnetzen



Hierarchische Adressierung: “Longest Prefix Matching”

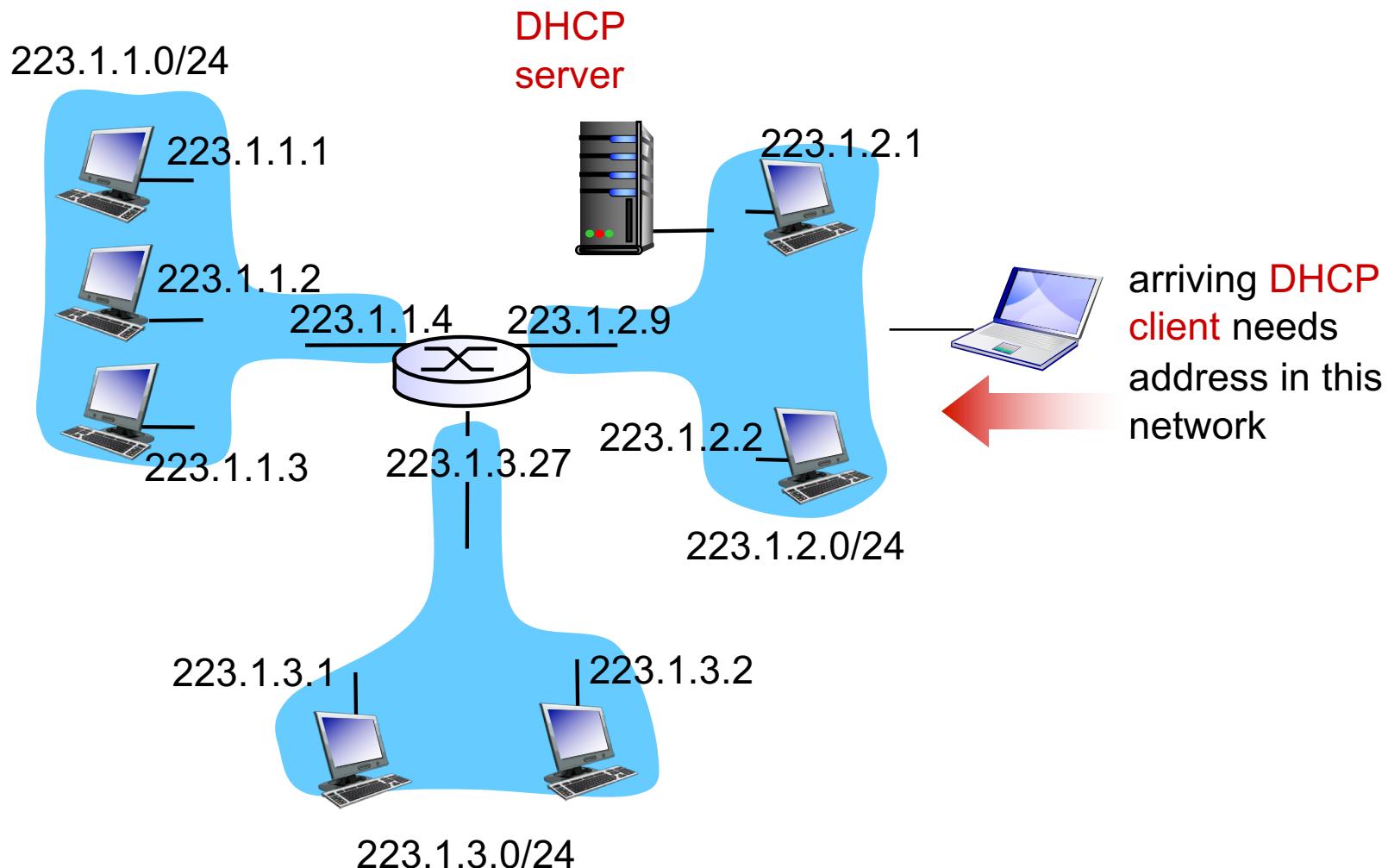
- Longest Prefix Matching => Router müssen Adressen mit längeren Netzwerk-Teilen zuerst auswerten!
- Beispiel: Organisation 1 ist zu ISP B gewechselt und hat seinen IP-Adressbereich mitgenommen. Pakete für Organisation 1 werden aus dem Internet an ISP B gesendet, wenn die ersten 23 Bit der Adresse übereinstimmen!



Zuweisung von IP-Adressen

- **An Interfaces von Routern:** In der Regel manuell
- **An Interfaces von Hosts:**
 - Feste IP-Adresse Manuell – Eintrag in eine Systemdatei
 - Temporäre IP-Adresse: Dynamische IP-Adresszuweisung in einem LAN:
“plug-and-play”
 - **DHCP: Dynamic Host Configuration Protocol [RFC 2131]**
- DHCP
 - Client – Server Protokoll, UPD, Port 67 (für Server) und 68 (für Client)
 - DHCP Server ist einem Subnetz zugeordnet. Der Server läuft im Subnetz
(alternativ: DHCP-Relay-Agent)
 - DHCP Server vergibt IP Adressen an Hosts, die das Subnetz betreten
 - Statisch (z.B. in Firmennetzen): “feste” Zuordnung
 - Dynamisch (z.B. bei ISPs): “zufällige” Zuordnung (IP-Adresse aus Pool
wird “vermietet”)
 - DHCP liefert, IP address, DNS Server, Subnetzmaske, first hop Router

DHCP Client Server Szenario



DHCP Protokoll

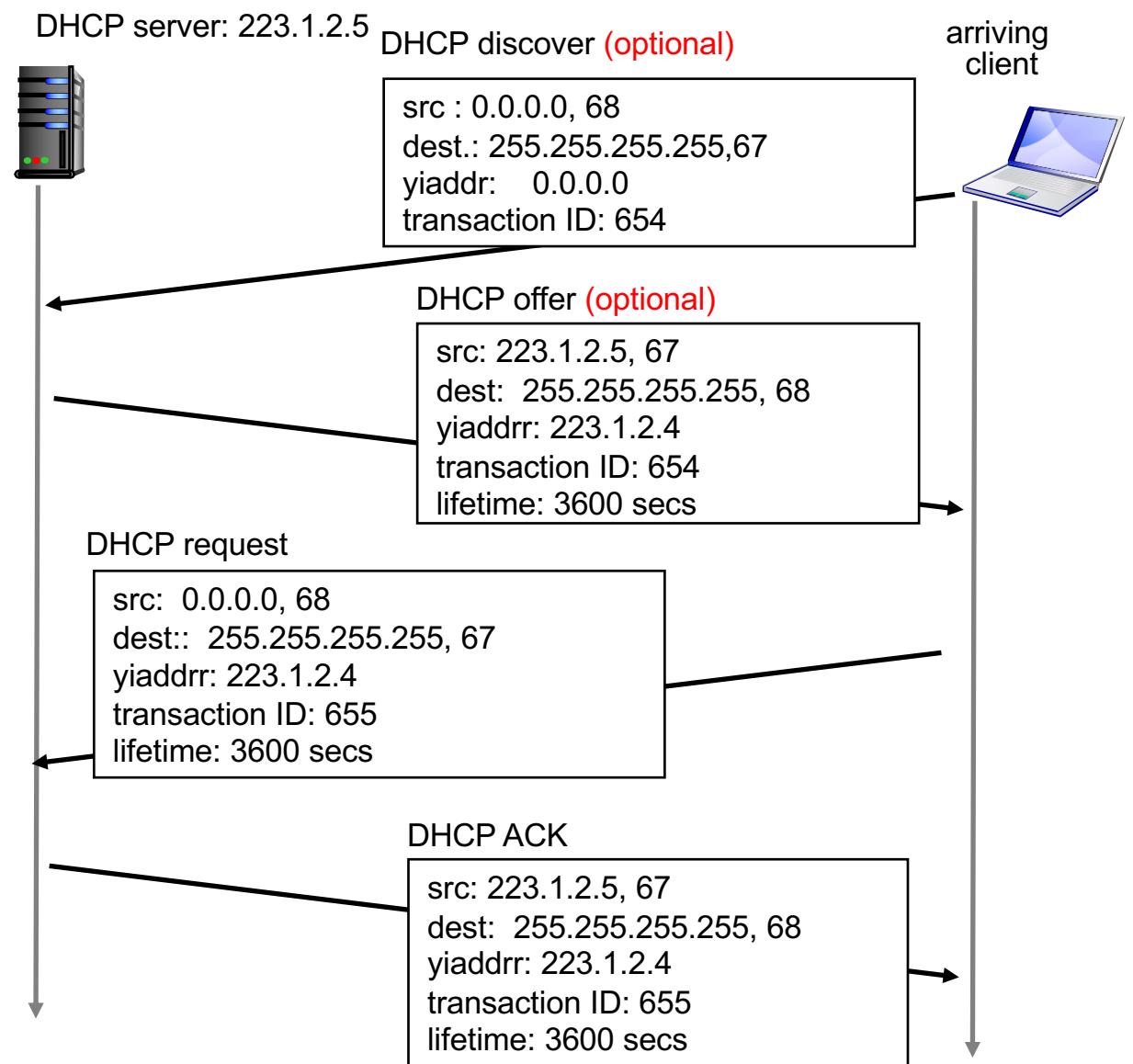
Discover: Client besitzt keine IP Adresse und kennt die IP Adresse des DHCP Servers nicht.

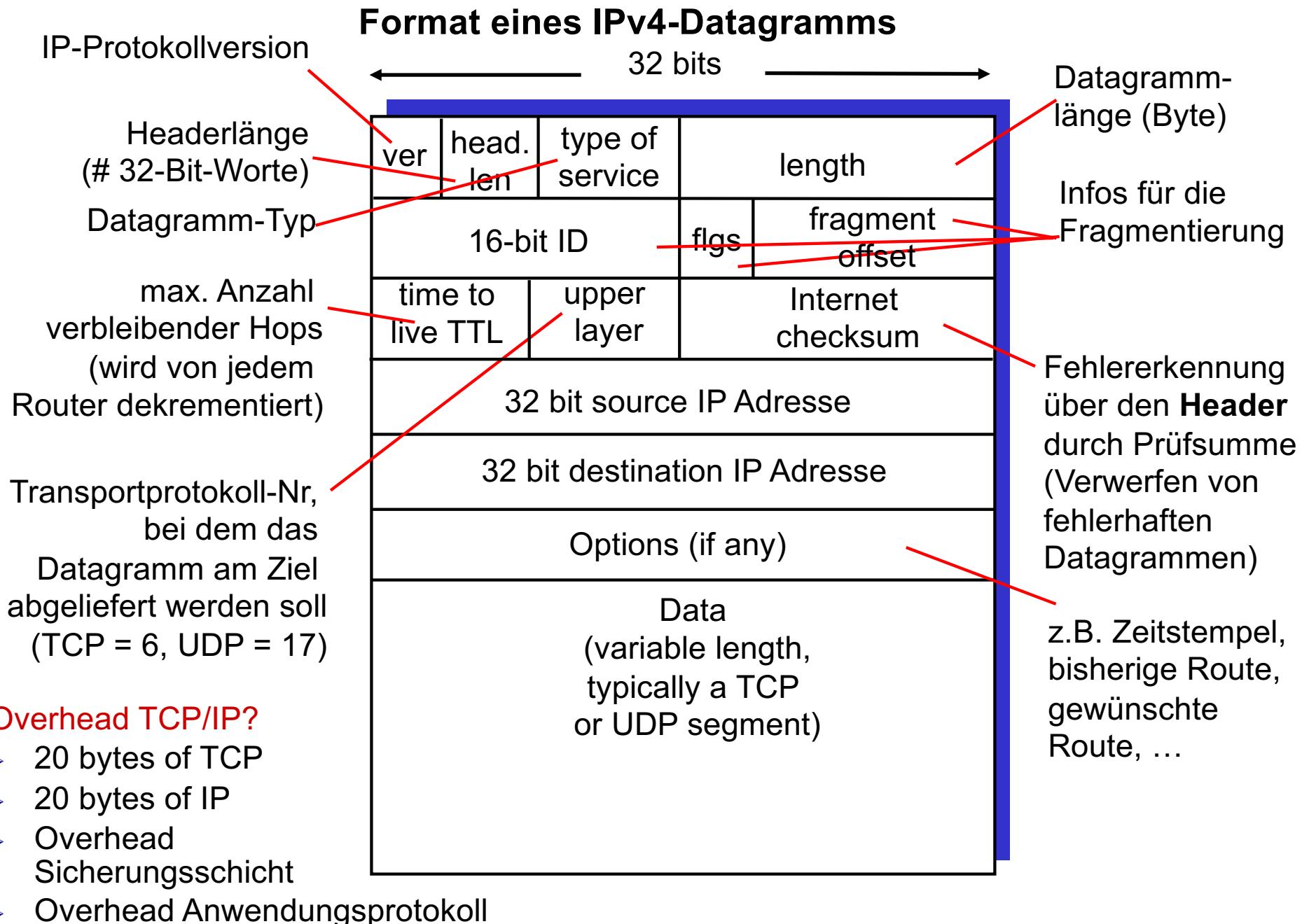
255.255.255.255: reservierte Adresse, Broadcast im Subnetz

0.0.0.0: reservierte Adresse mit der Semantik „dieser Host“

Transaction: Damit der Client trotz Broadcast Adresse im DHCP offer die Antwort seiner Anfrage zuordnen kann.

Lifetime: Ausleihzeit





Der Weg eines Datagramms von der Quelle zum Ziel: Szenario

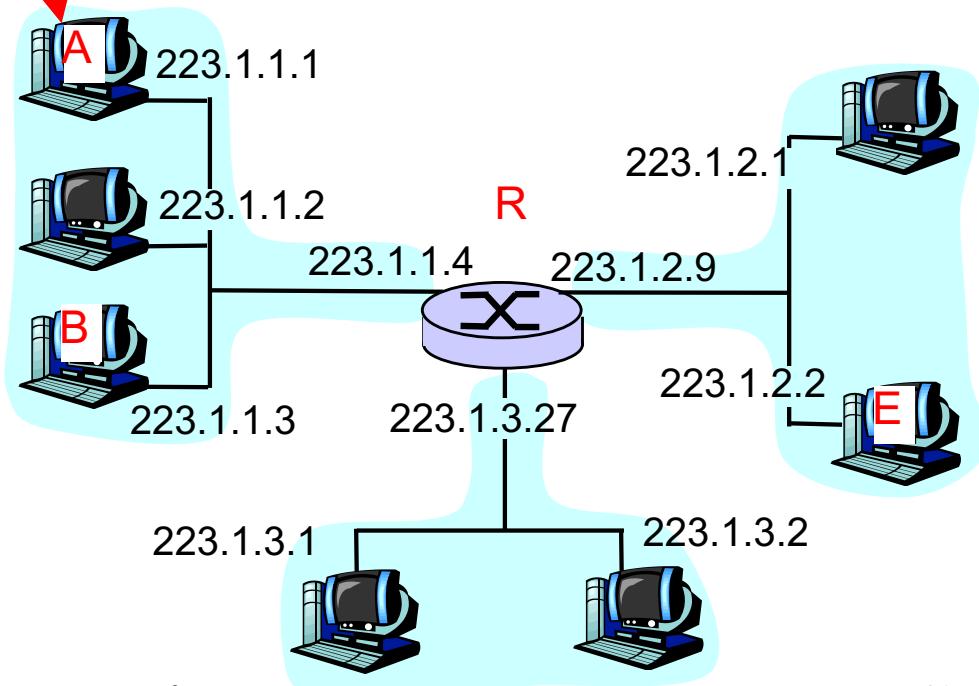
IP Datagramm:

...	source IP addr	dest IP addr	data
-----	----------------	--------------	------

- Ein Datagramm wird auf dem Weg von einer Quelle (A) zu einem Ziel (B oder E) in den IP-Adressfeldern **nicht verändert!**
- Zusätzlich benötigt:
“LAN-Adresse” auf Schicht 2 → kommt später genauer.

Routingtabelle in A

Zielnetz	Nächster Router	Interface
223.1.1.0/24	–	223.1.1.1
223.1.2.0/24	223.1.1.4	223.1.1.1
223.1.3.0/24	223.1.1.4	223.1.1.1



Der Weg eines Datagramms von der Quelle zum Ziel: Beispiel 1

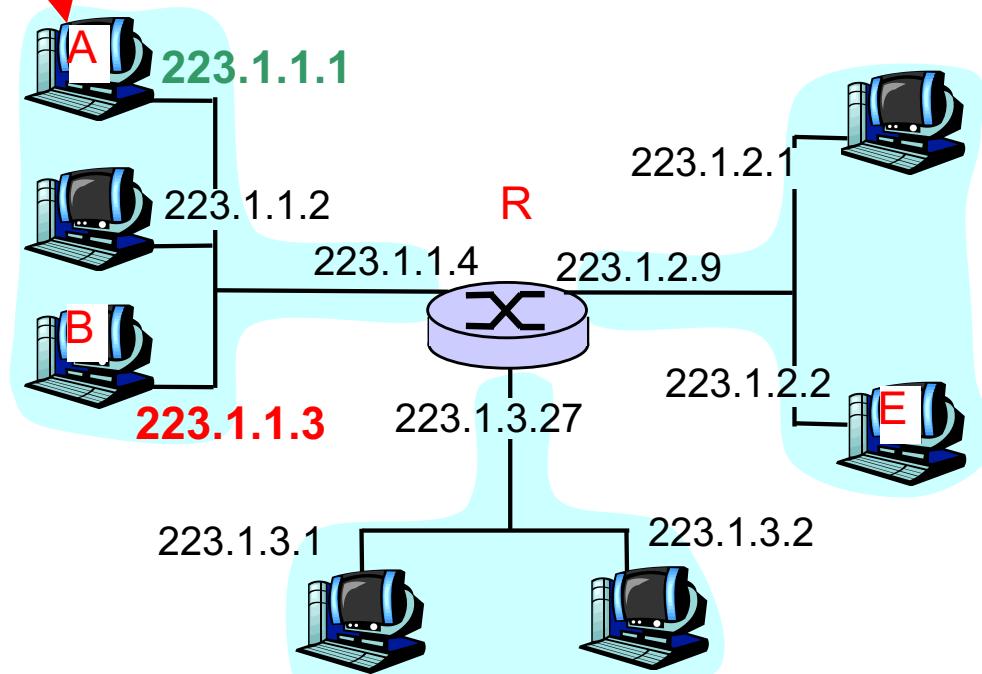
Quelle A sendet ein IP-Datagramm zum Ziel B

...	223.1.1.1	223.1.1.3	data
-----	-----------	-----------	------

- A: Ermittle Netzwerkadresse von B = Zielnetz
- A: Suche den nächsten Router in der Routingtabelle: Eigenes Netz!
- A: Sende das Datagramm über die Sicherungsschicht ins eigene LAN mit LAN-Adresse von B (A und B sind direkt miteinander verbunden!)

Routingtabelle in A

Zielnetz	Nächster Router	Interface
223.1.1.0/24	–	223.1.1.1
223.1.2.0/24	223.1.1.4	223.1.1.1
223.1.3.0/24	223.1.1.4	223.1.1.1



Der Weg eines Datagramms von der Quelle zum Ziel: Beispiel 2-1

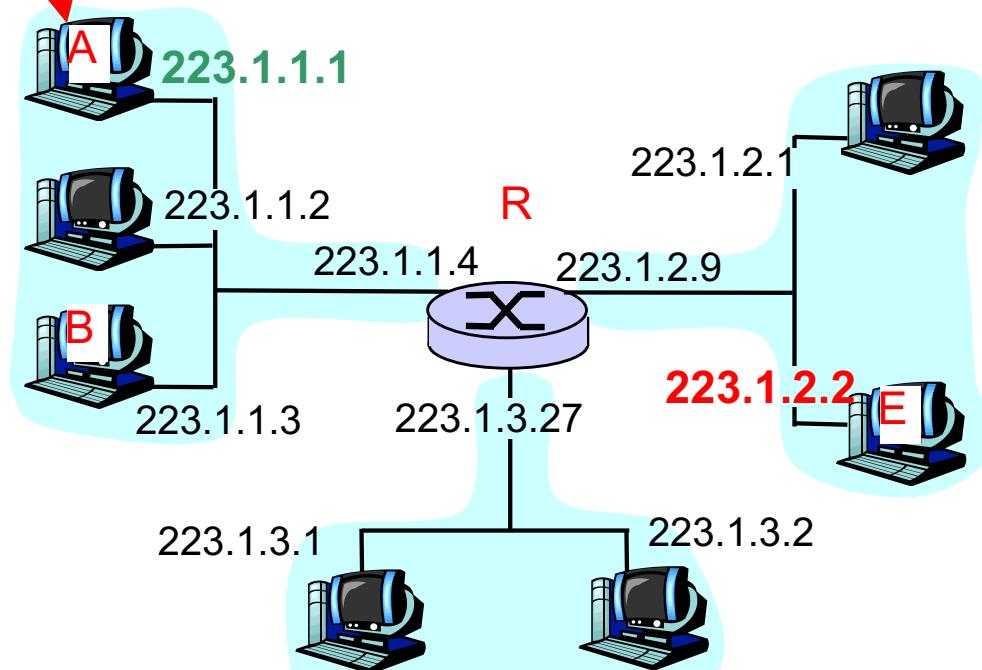
Quelle A sendet ein IP-Datagramm zum Ziel E

...	223.1.1.1	223.1.2.2	data
-----	-----------	-----------	------

- A: Ermittle Netzwerkadresse von E = Zielnetz
- A: Suche den nächsten Router in der Routingtabelle:
→ Nächster Router R hat die Adresse 223.1.1.4
- A: Sende das Datagramm über die Sicherungsschicht ins eigene LAN mit LAN-Adresse des Routers R

Routingtabelle in A

Zielnetz	Nächster Router	Interface
223.1.1.0/24	–	223.1.1.1
223.1.2.0/24	223.1.1.4	223.1.1.1
223.1.3.0/24	223.1.1.4	223.1.1.1



Der Weg eines Datagramms von der Quelle zum Ziel: Beispiel 2-2

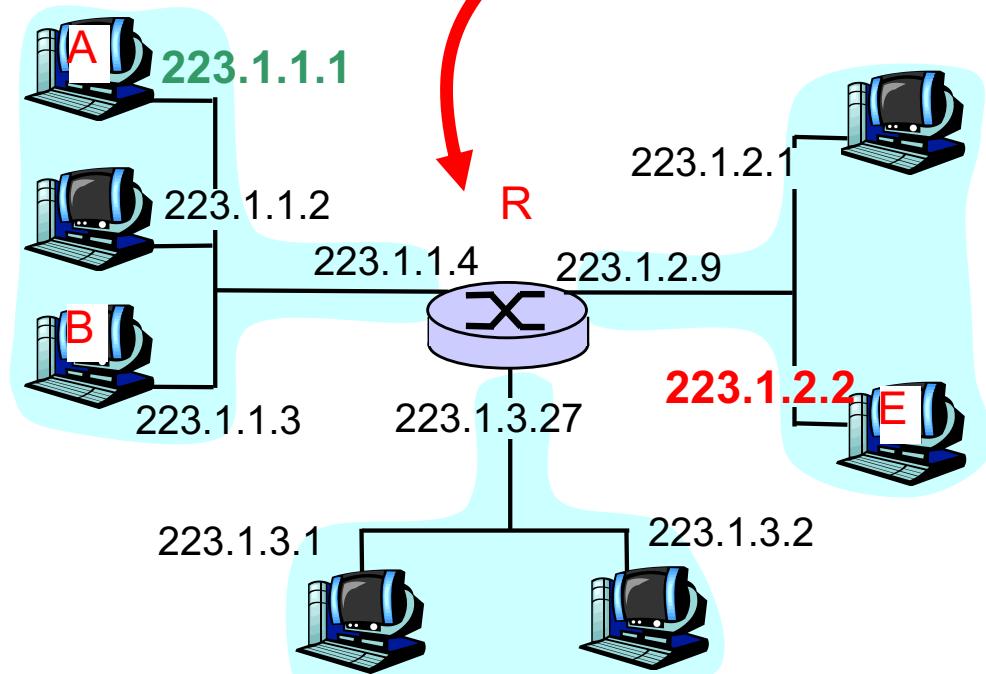
Router R erhält das Datagramm auf Interface 223.1.1.4 und muss es an E weiterleiten

...	223.1.1.1	223.1.2.2	data
-----	-----------	-----------	------

1. R: Ermittle Netzwerkadresse von E = Zielnetz
2. R: Suche das Zielnetz in der Routingtabelle: E ist im selben Netzwerk wie Interface 223.1.2.9
3. R: Sende das Datagramm über Interface 223.1.2.9 und die entspr. Sicherungsschicht ins LAN mit LAN-Adresse von E

Routingtabelle im Router R

Zielnetz	Nächster Router	Interface
223.1.1.0/24	-	223.1.1.4
223.1.2.0/24	-	223.1.2.9
223.1.3.0/24	-	223.1.3.27



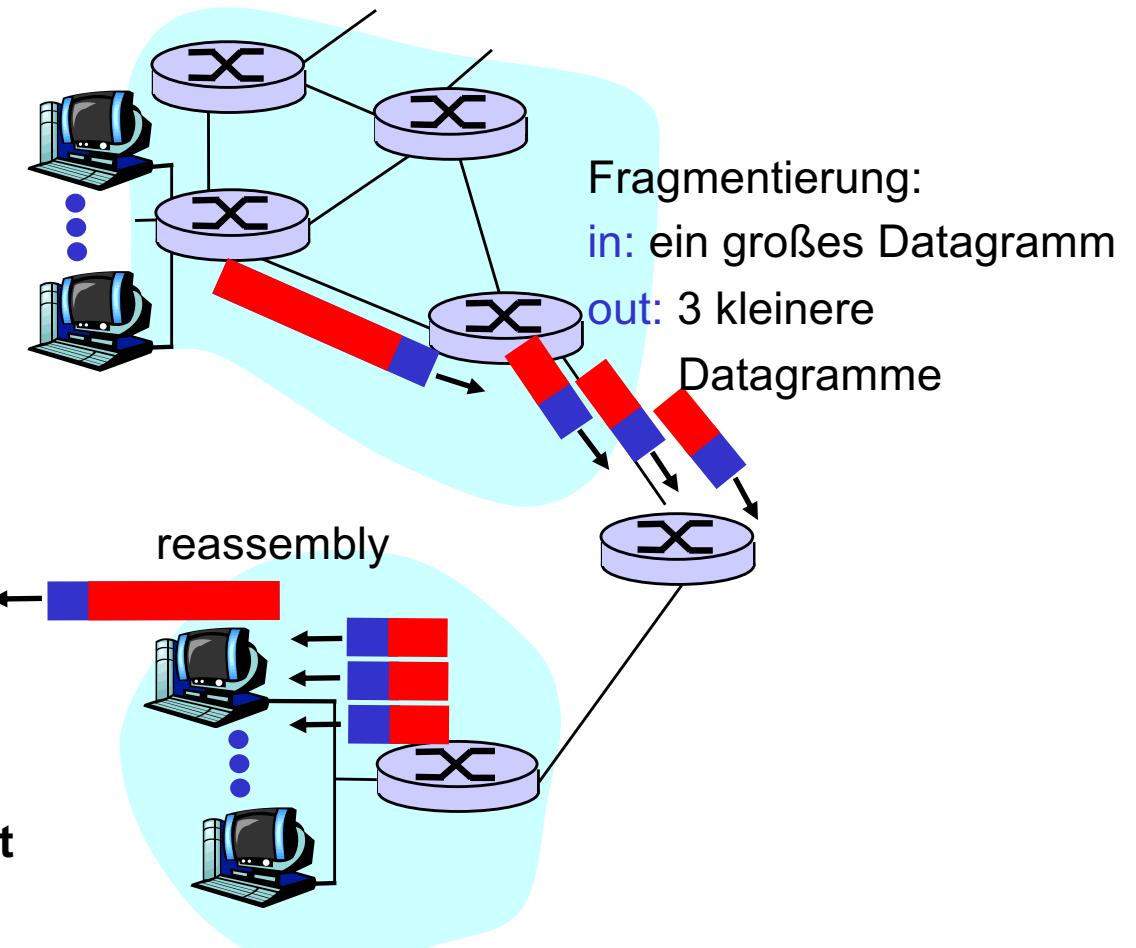
Beispiel: Routingtabelle eines PC mit 3 Netzwerkschnittstellen (Auszug)

Destination	Gateway	Interface
127.0.0.1	127.0.0.1	lo
192.168.2.	192.168.2.5	fa0
193.55.114.	193.55.114.6	le0
192.168.3.	192.168.3.5	qaa0
224.0.0.0	193.55.114.6	le0
default	193.55.114.129	le0

- Loopback-Interface 127.0.0.1: Ein gesendetes Paket wird sofort an das eigene Interface zurückgegeben und wie ein angekommenes Paket behandelt (*Hostname: "localhost"*)
- Drei angeschlossene Klasse-C-Netzwerke 192.168.2, 193.55.114, 192.168.3
- Multicast-Adresse: 224.0.0.0 für Gruppenkommunikation
- Der Default-Router 193.55.114.129 wird benutzt für alle Zielnetze, für die keine Route bekannt ist.

IPv4-Fragmentierung und Reassemblierung

- Sicherungsschicht-Pakete haben eine MTU (Max. Transfer Unit) = größte mögliche Paketlänge
- MTU ist abhängig vom Protokoll, Hardware, Betriebssystem, ...
- Große IP-Datagramme müssen daher evtl. aufgeteilt ("fragmentiert") werden
 - aus *einem* Datagramm werden *mehrere* Datagramme
 - **Zusammensetzen** ("Reassemblierung") findet **nur auf dem Zielhost statt!**
 - ■ IP-Headerinformationen werden zur Identifikation und Reihenfolgeerhaltung der einzelnen Fragmente benötigt



IPv4-Fragmentierung und Reassemblierung: Beispiel

20 Byte IP-Header
→ 3980 Byte Nutzdaten

fragflag = 1 zeigt an,
dass noch mehr
kommt!

offset = 185 heißt, dass
die Daten am Ziel ab
Byte $8 * 185 = 1480$
wieder eingefügt
werden müssen!

Aus einem großen Datagramm werden
mehrere kleine Datagramme

	Länge	ID	fragflag	offset	
	=4000	=x	=0	=0	

	Länge	ID	fragflag	offset	
	=1500	=x	=1	=0	

	Länge	ID	fragflag	offset	
	=1500	=x	=1	=185	

	Länge	ID	fragflag	offset	
	=1040	=x	=0	=370	

ID: Ordnet die Fragmente einander zu.

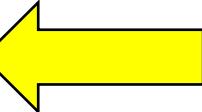
ICMP: Internet Control Message Protocol

- Benutzt von Hosts und Routern, um Steuerungsinformationen auf Netzwerkebene auszutauschen
 - Fehlermeldungen (z.B. "dest host unreachable")
 - Statusmeldungen (z.B. "echo request/reply" → ping)
- ICMP-Nachrichten werden in IP-Datagrammen transportiert (Protokoll Nummer für Upper Layer Feld: 0x01)
- ICMP-Nachrichtenformat:
 - Typ
 - Code
 - Erste 8 Byte des IP-Datagramms, das den Fehler verursacht hat

Type	Code	description
0	0	echo reply (ping)
3	0	dest network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header
...	...	

Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6 
- Paketfilterung (Firewalls)
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung

Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

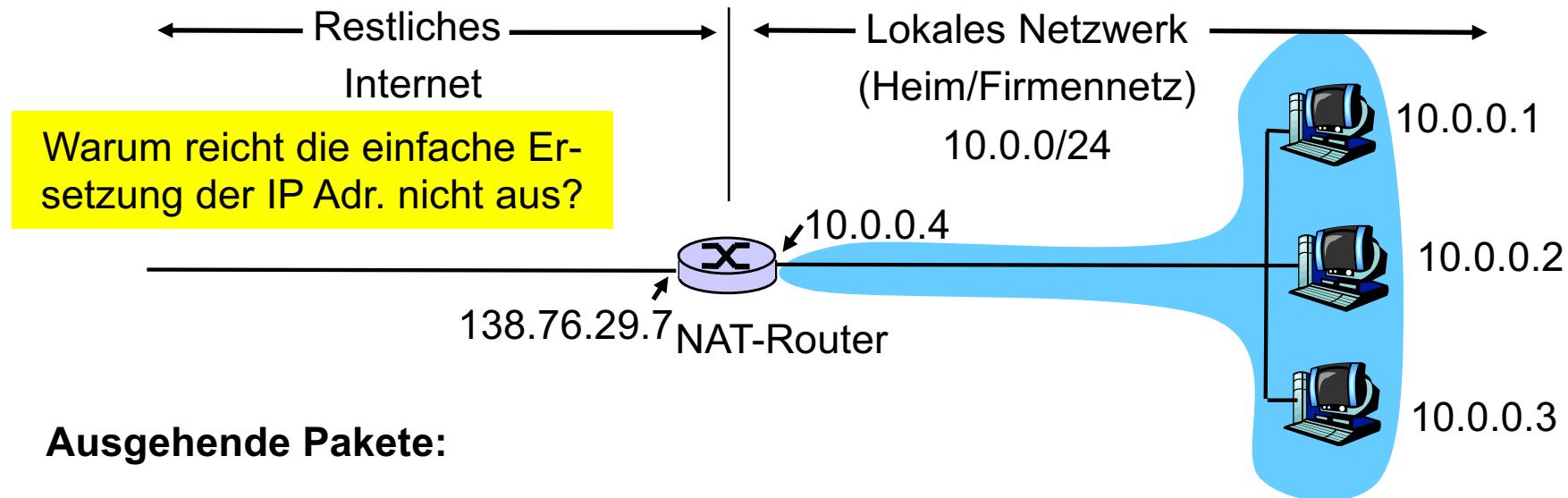
Problemstellung

- Der IPv4-Adressraum (32 Bit) ist zu klein, um allen Internet-Hosts eine weltweit eindeutige IP-Adresse zu geben!
- Lösungswege:
 - CIDR ✓
 - DHCP ✓
 - **NAT** (“Network Address Translation”):
Lösung für private Netze
 - **IPv6**: Einzige langfristige Lösung

NAT (“Network Address Translation”) [RFC 3022]

- Folgende Adressen werden im Internet **nicht geroutet**:
 - 10.0.0.0/8 – 10.255.255.255/8
 - 172.16.0.0/12 – 172.31.255.255/12
 - 192.168.0.0/16 – 192.168.255.255/16
- Benutzung dieser „**privaten**“ IP-Adressen im Intranet (Heim- oder Firmennetz)
- Umsetzung der virtuellen IP-Adresse in eine „**öffentliche**“ IP-Adresse, wenn eine Verbindung zum Internet nötig ist (im NAT-Router)
- Vorteile:
 - Nur eine (öffentliche IP Adresse) für das gesamte Subnetzwerk
 - Im Subnetzwerk kann man Adresse ohne Konsequenzen für die Außenwelt ändern.
 - Host im Netzwerk sind von außen nicht direkt adressierbar (auch in gravierender Nachteil).

NAT - Realisierung



Ausgehende Pakete:

- NAT-Router speichert private IP-Adresse und originalen TCP/UDP-Quellport in einer Ersetzungstabelle unter einem neu erzeugten Index
- NAT-Router ersetzt im Paket Quell-IP-Adresse und TCP/UDP-Quellport
Quell-IP-Adresse → Öffentliche IP-Adresse des NAT-Routers (hier 138.76.29.7)
TCP/UDP Quellport → Index des neuen Ersetzungstabelleneintrags

Eingehende Pakete (Antworten):

- NAT-Router ersetzt Ziel-IP-Adresse (NAT-Routeradresse) und TCP/UDP-Zielport (Index) anhand der Ersetzungstabelle durch die gespeicherten Werte

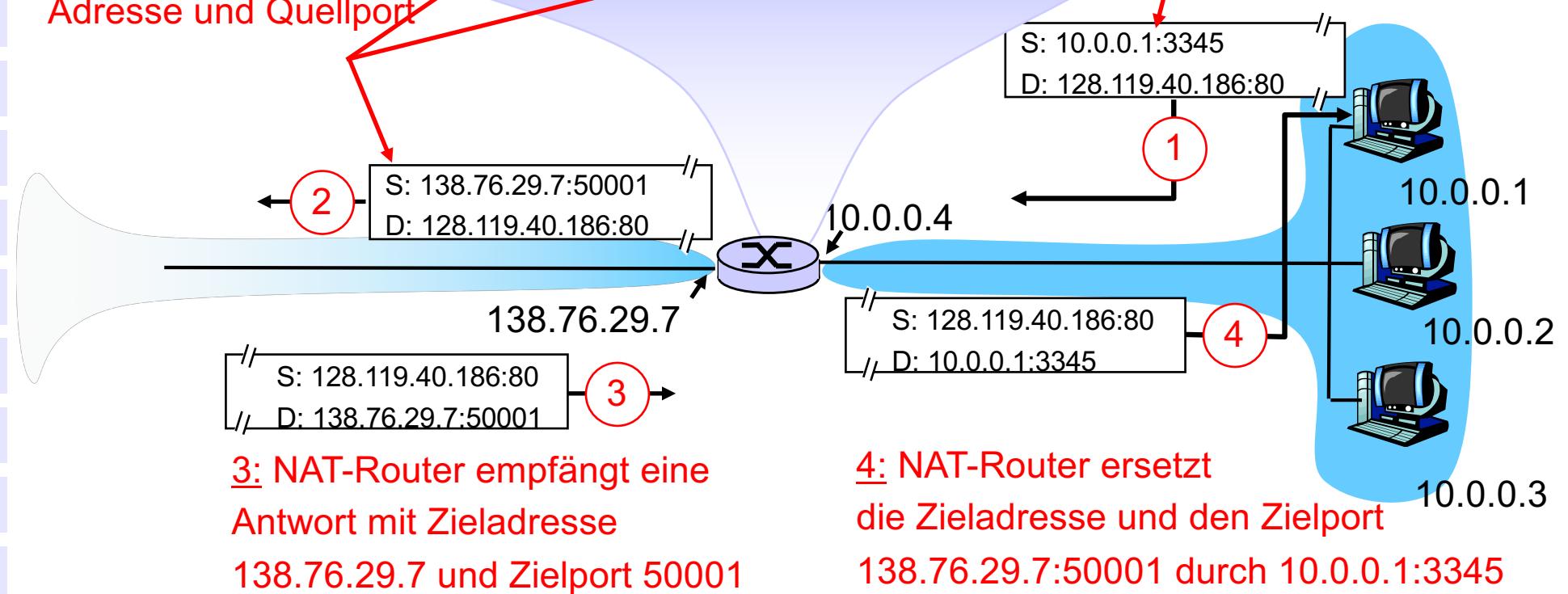
NAT Beispiel

2: NAT-Router

erzeugt neuen Eintrag
in der Ersetzungstabelle
(Index 50001)
und ersetzt Quell-IP-
Adresse und Quellport

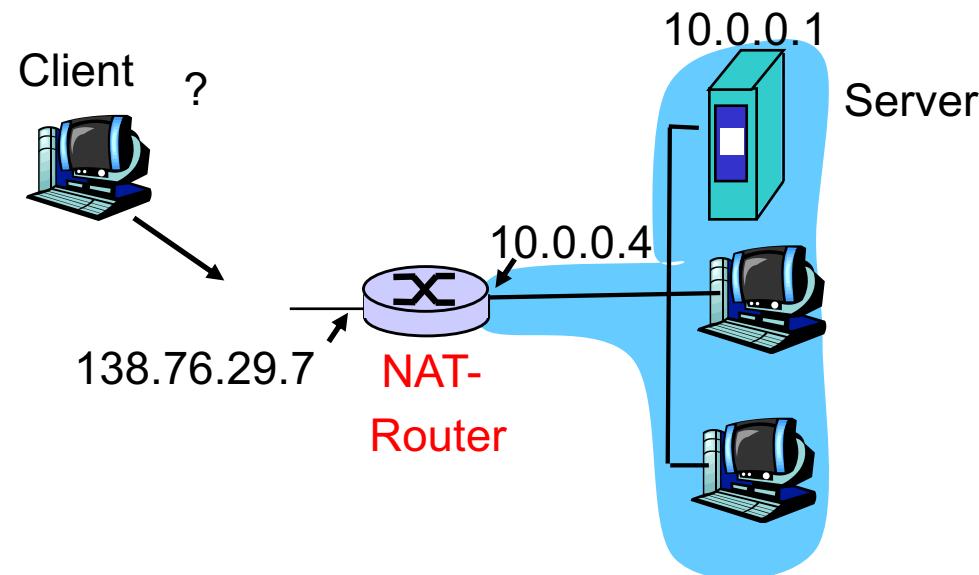
NAT Ersetzungstabelle	
WAN-Seite (Index)	LAN-Seite
138.76.29.7:50001	10.0.0.1:3345
.....

1: Host 10.0.0.1
sendet Paket an
128.119.40.186:80
mit Quellport 3345



Diskussion NAT

- Gute 60000 Verbindungen mit nur einer IP Adresse
- 💣 Unzulässige Vermischung der Protokollsichten!
- 💣 Widerspricht dem end-to-end Gedanken.
 - Wie kann man mit einem P2P Partner oder einem Server kommunizieren, der hinter einem NAT Router liegt?
- NAT ist eine Zwischenlösung auf dem Weg zu IPv6.

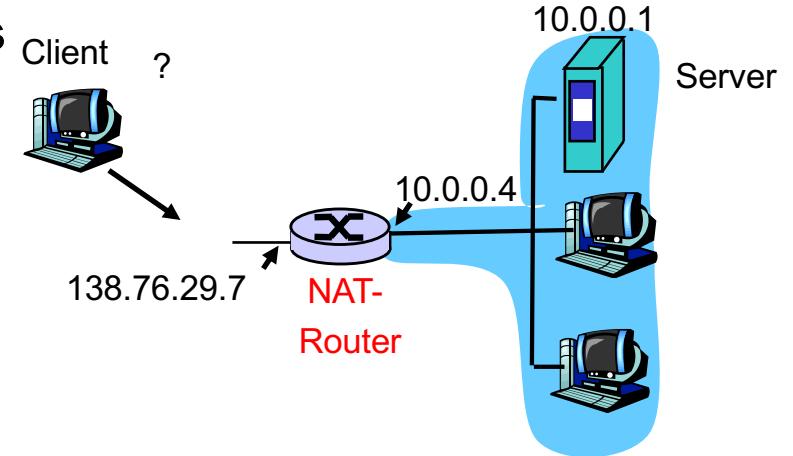


Work-A-Rounds

Work-A-Round 1:

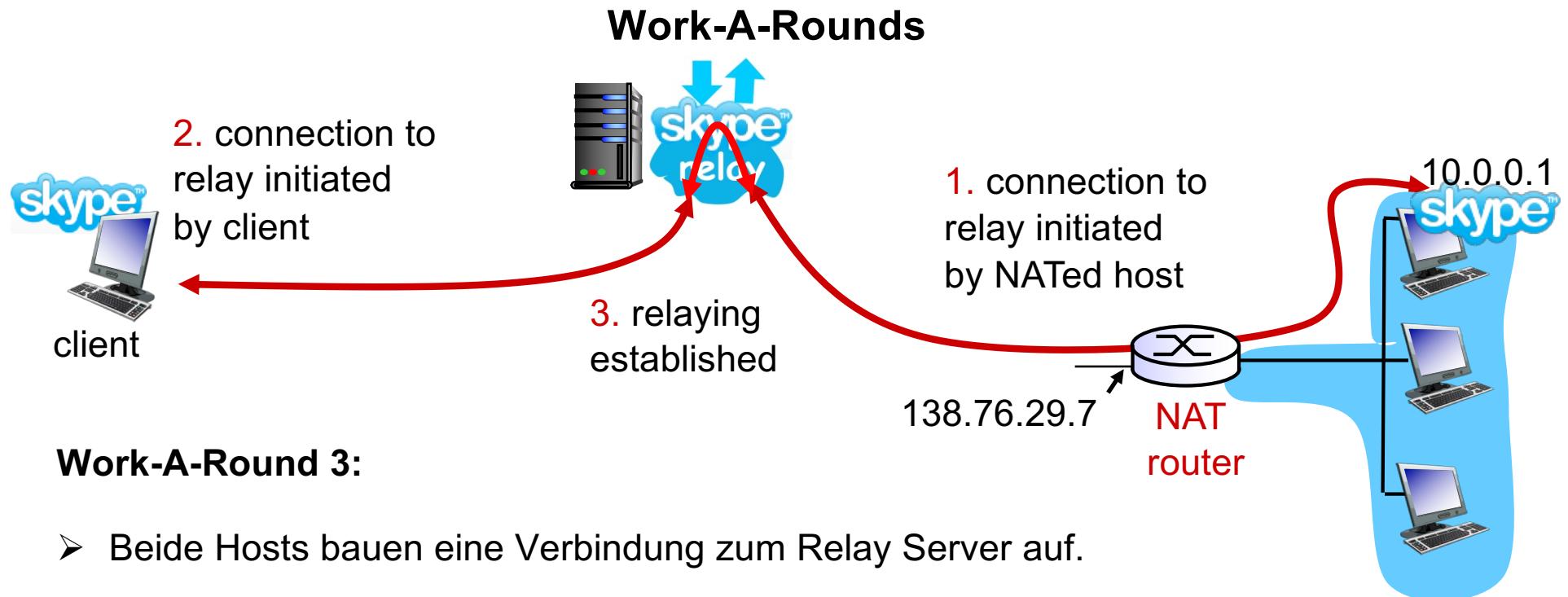
- Statisch konfigurierter NAT Router, der die ein-kommenden Anforderungen auf einen festen Host routet.

z.B.: (123.76.29.7, port 25000) always forwarded to 10.0.0.1 port 25000



Work-A-Round 2:

- Universal Plug and Play (UPnP)
- Dies ist ein Protokoll, über das ein Host eine Zuordnung
(private IP-adr, port) \Leftrightarrow (öffentliche IP-adr, port)
anfordern kann.



IPv6 [RFC 2460 -2466]

Historie

- 1993 Erste Entwürfe, IETF Arbeitsgruppe startet
- 1998 RFC 2460: Spezifikation als Internet-Standard abgeschlossen

Designziele:

- Vergabe einer weltweit eindeutigen Adresse an Milliarden von Rechnern (auch bei "Verschnitt")
- Routingtabellenverkleinerung
- Vereinfachung des IPv4-Protokolls
- Verbindliche Sicherheitsmechanismen
- Verkehrsklassen über Prioritätenangabe ("Quality of Service")
- Intelligentes Multicasting für definierte Gruppen
- Unterstützung mobiler Endgeräte
- Zukünftige Erweiterungsmöglichkeiten
- Koexistenzmöglichkeit von IPv4 und IPv6

Format von IPv6-Adressen

Länge: 128 Bit (16 Byte)

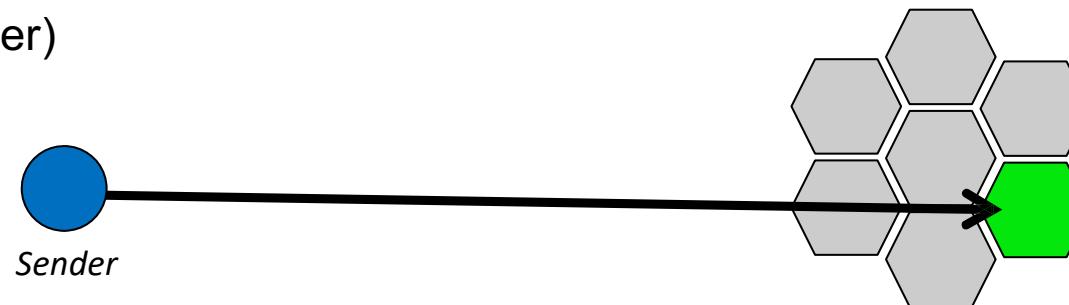
- Ergibt insgesamt ca. $3 \cdot 10^{38}$ Adressen oder $7 \cdot 10^{23}$ Adressen pro Quadratmeter (auf der gesamten Erde)

Format:

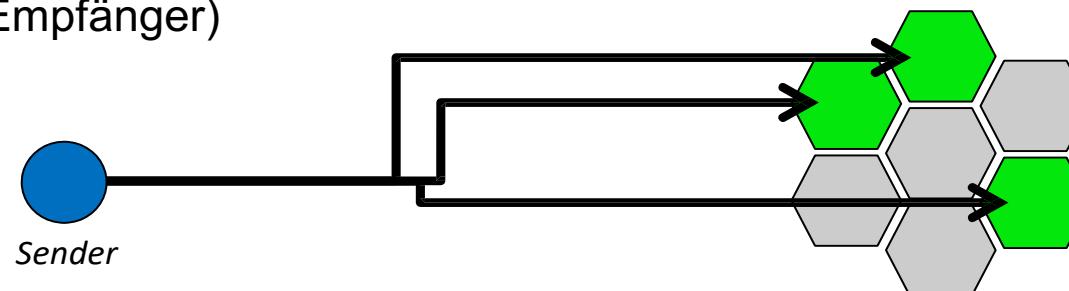
- Acht Gruppen von jeweils 4 Hex-Ziffern (mit je 4 Bit), durch Doppelpunkt getrennt
 - Beispiel: **8000:0000:0000:0000:0123:4567:89AB:CDEF**
- Führende Nullen können weggelassen werden
- Eine oder mehrere aufeinander folgende Gruppen mit 16 Null-Bits können einmal durch :: ersetzt werden.
 - Beispiel: **8000::123:4567:89AB:CDEF**
- Schreibweise in URLs mit eckigen Klammern
 - Beispiel: **http://[2001:638::8:15]:80**

Typen von IPv6-Adressen

Unicast „Normale“ Adresse eines Interfaces
(1 Sender – 1 Empfänger)



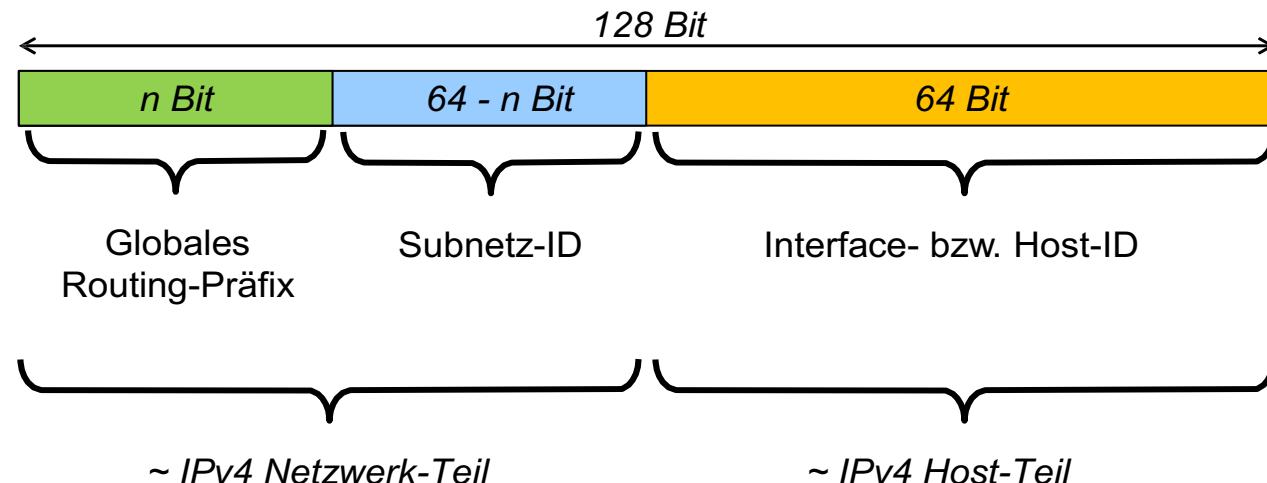
Multicast Gruppenadresse
(1 Sender → mehrere Empfänger)



Anycast Gruppenadresse mit Zustellung an das „nächste“ Mitglied

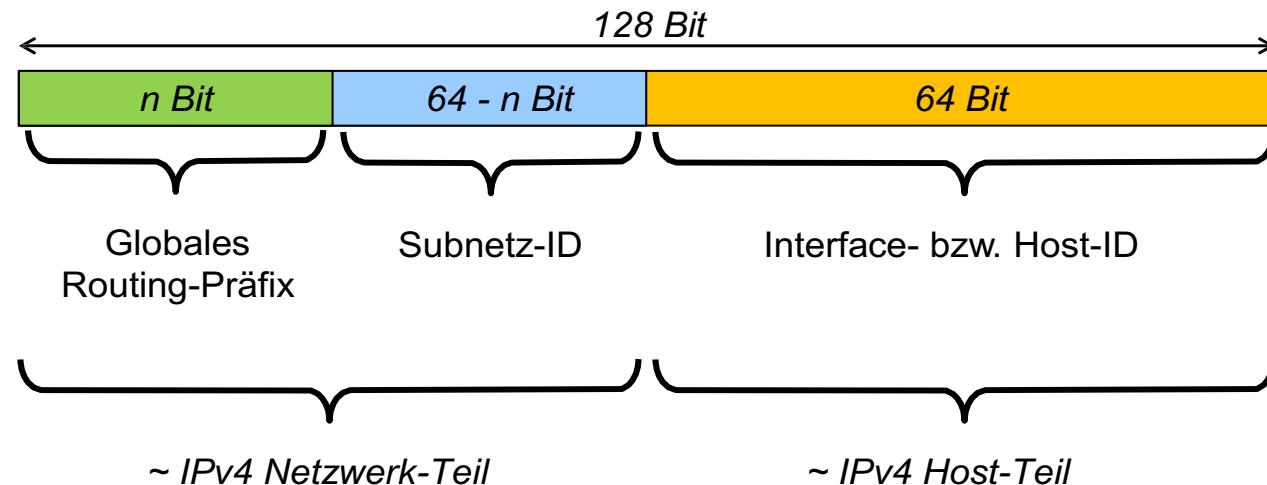
Jedes Interface kann beliebig viele IPv6-Adressen besitzen!

IPv6: Unicast-Adressen



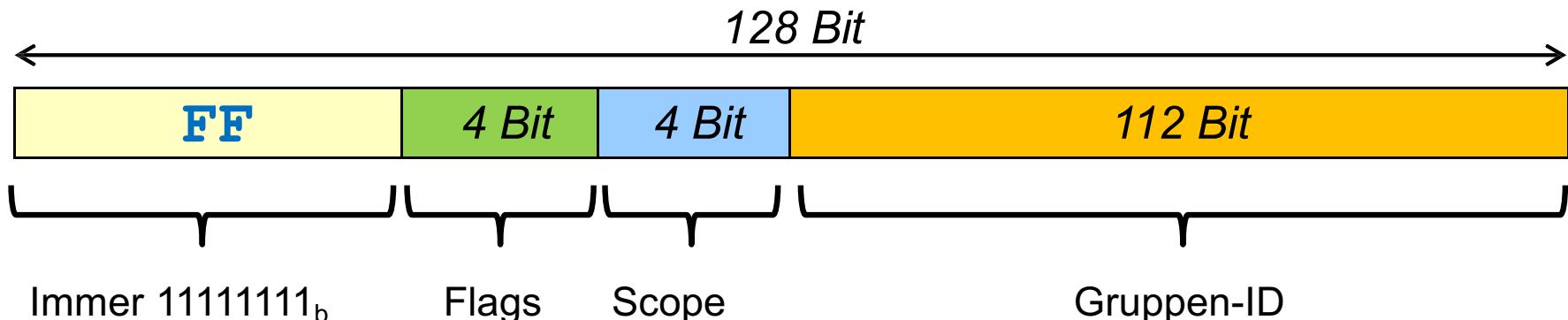
- **Global Unicast-Adr.** Präfix $001_2 \approx 2000::/3$
 - 12,5 % des Adr. Raums
 - Momentan werden Adressen aus diesem Bereich von IANA vergeben
 - Werden auch für Anycast verwendet
- **Link-local Unicast-Adr.** Präfix $1111\ 1110\ 10_2 \approx \text{FE80}::/10$ mit Subnetz-ID 0
 - 0,1 % des Adr. Raums
 - Nur gültig in gleichem LAN, Pakete werden von Routern nicht weitergeleitet

IPv6: Unicast-Adressen



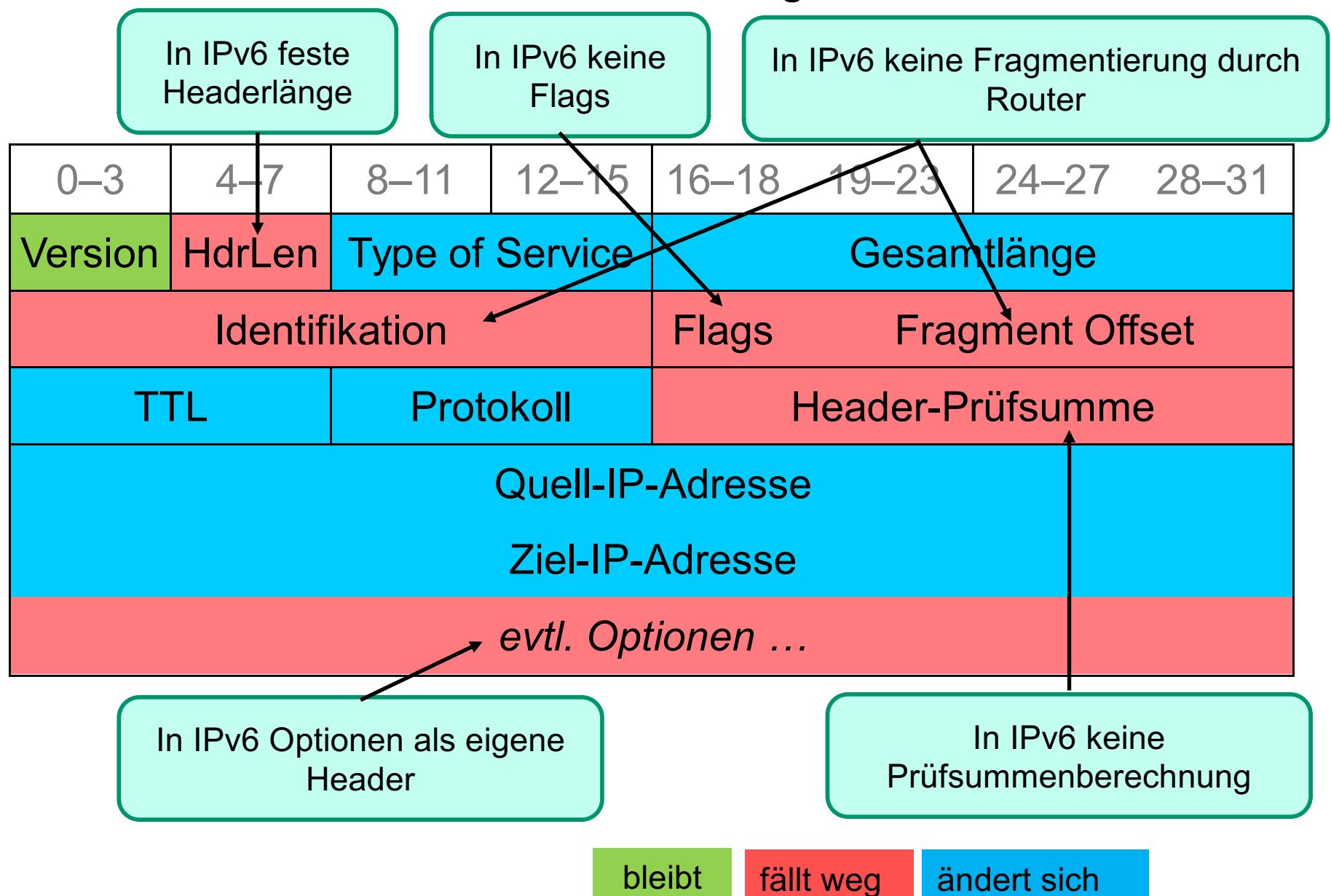
- **Unique-local IPv6-Adr.** Präfix 1111 110 \approx FC00::/7
 - + 40 Bit für eindeutige Site-ID + 16 Bit Subnet ID
 - „Private“ Adresse, nur gültig in eigenen Subnetzen
- 0:0:0:0:0:0:0:
 - Unspezifizierte Adresse
- 0:0:0:0:0:0:1:
 - Loop Back Adr.

IPv6: Multicast-Adressen

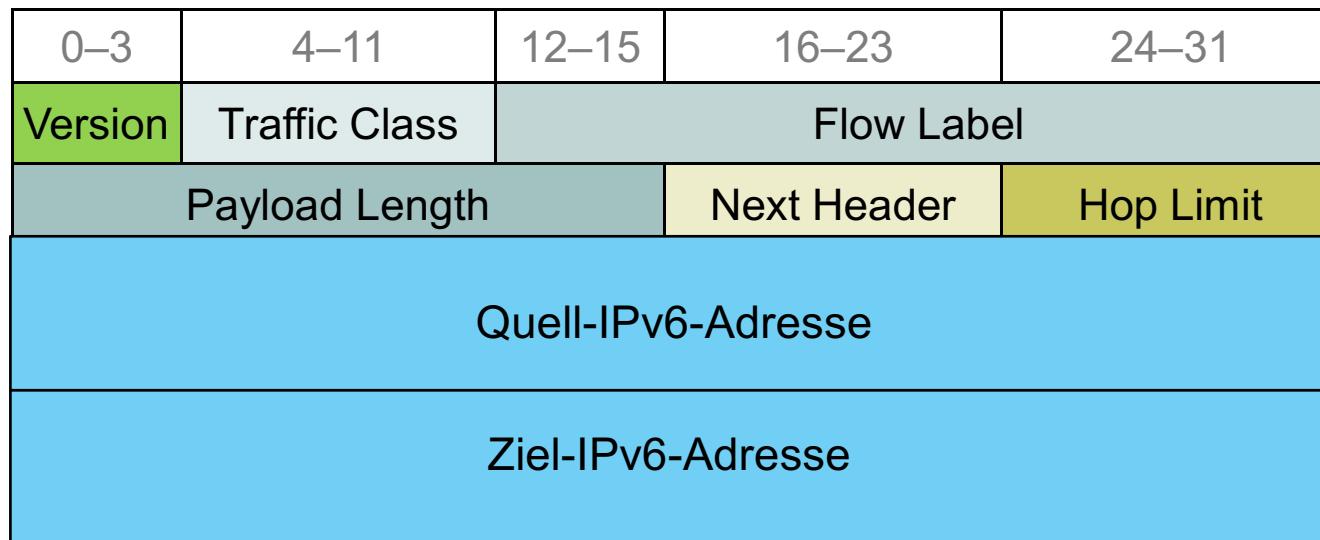


- Reservierter Adressbereich für Multicast: **FF00::/8**
- **Scope:** Definition der Reichweite
 - 1: interface-local 2: link-local 3: subnet-local E: global
- **Beispiel** für vordefinierte link-local Multicast-Gruppen:
 - FF02::1** All Nodes (entspricht dem Broadcast)
 - FF02::2** All Routers

IPv4-Header-Änderungen



IPv6 Header



IPv6 Datagramm-

Format:

- Header hat eine feste Länge von 40 Byte
- Fragmentierung von Datagrammen nur durch den Sender erlaubt!

- **Version:** IP-Versionsnummer
- **Traffic Class:** Priorität innerhalb eines “Flows”
- **Flow Label:** ID eines Flows
- **Payload Length:** Länge des Nutzdatenfelds (ohne IPv6-Header)
- **Next Header:** Code für die ersten Bytes des Nutzdatenfeldes: z.B. TCP/UDP-Header oder weiterer optionaler IPv6-Header
- **Hop Limit:** analog TTL bei IPv4

Verarbeitung von Optionen: IPv4 vs. IPv6

IPv4:

- Erlaubt, IPv4-Header wird entsprechend verlängert (20 – 60 Byte)

IPv6:

- Erlaubt, aber nur außerhalb des festen 40 Byte-IPv6-Headers
 - Verwendung von zusätzlichen Headern fester oder variabler Länge ("Extension Header")!
- Optionale Extension Header stehen vor TCP/UDP-Header, falls benötigt
- Angezeigt durch Zahlencode im "Next Header"-Feld
 - "Header-Chaining"
- Beispiel:

IPv6 header Next=43(Routing)	Routing header Next=44(Fragment)	Fragment header Next=6(TCP)	TCP segment
------------------------------------	--	-----------------------------------	-------------

Auswahl von Pv6 Extension Header für Optionen

Hop-by-Hop Options (0)

Spezielle Optionen, die an jedem Router verarbeitet werden

Destination Options (60)

Informationen für den Empfänger-Host

Routing (43)

Erweiterte Routinginformationen (vorgegebene Route)

Fragmentation (44)

Fragmentierungs-/Defragmentierungsinformationen

Encapsulation (50)

Verschlüsselung, z.B. für ‚Tunneling‘ vertraulicher Daten (IPSec/ESP-Protokoll)

Authentication (51)

Sicherheitsinformationen: Authentizität und Integrität (IPSec/AH-Protokoll)

No Next Header (59)

Verweise, dass kein weiterer Header folgt.

ICMPv6

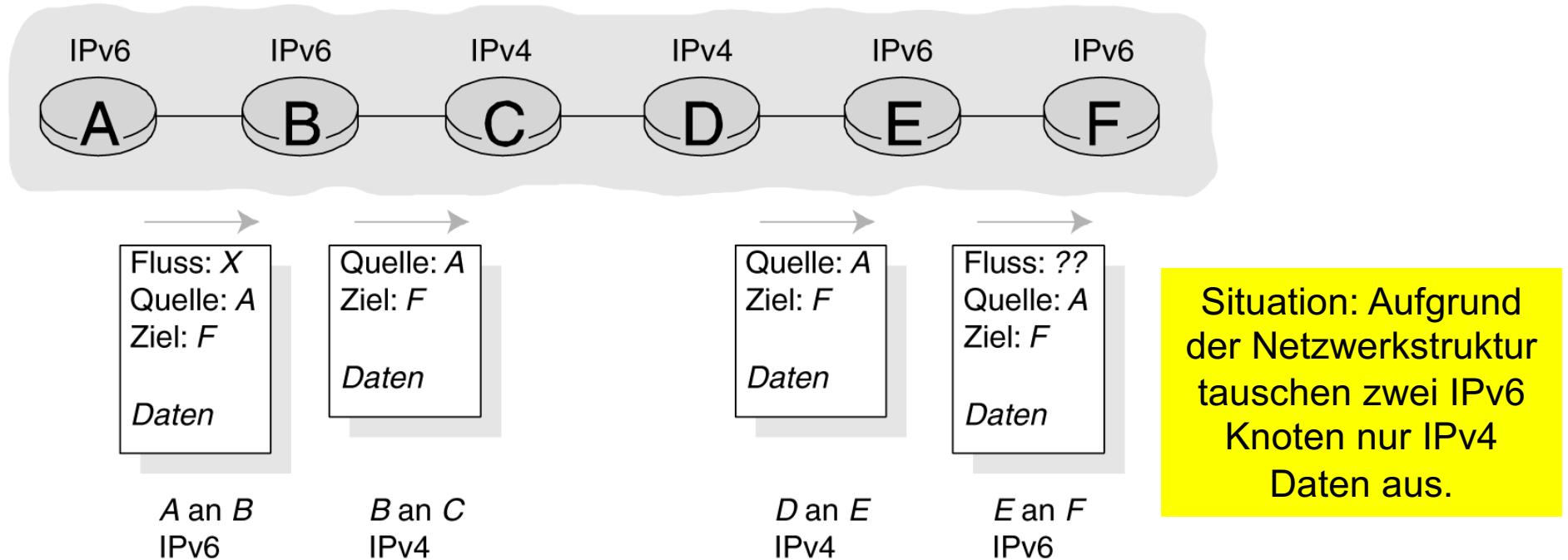
- **ICMPv6 (Internet Control Message Protocol v6):** Teil von IPv6, muss komplett implementiert sein!
- **Neue Version von ICMP mit zusätzlichen Aufgaben:**
 - Zusätzliche Nachrichtentypen (z.B. "Packet Too Big")
 - "Neighbor Discovery Protocol" (NDP)
 - (Auto-)Konfiguration von IP-Adressen
 - Ermittlung von doppelten IP-Adressen im LAN
 - Ermittlung von Routern im LAN
 - Ermittlung von LAN-Adressen (IPv4: ARP-Protokoll → Kap. 5)
 - Ermittlung, ob Nachbarn im LAN noch erreichbar sind
 - Ermittlung der MTU (Maximum Transfer Unit) eines Pfads
 - Multicast-Unterstützung (IPv4: IGMP-Protokoll)
- ...

Übergang von IPv4 auf IPv6

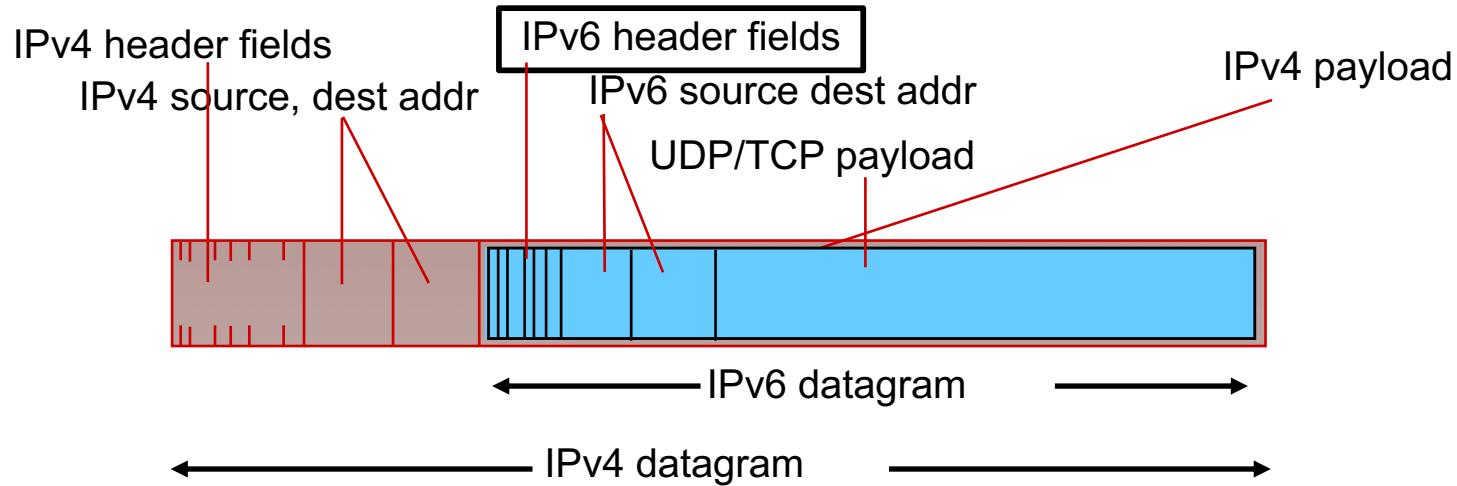
- Nicht alle Router können gleichzeitig aufgerüstet werden
- Wie kann ein Netzwerk mit gemischten IPv4- und IPv6-Routern arbeiten?
- Spezielle IPv6-Router müssen mit IPv4-Routern IPv4-Datagramme austauschen können!
- Zwei Ansätze:
 - **Dual Stack:** Zwischen den IP-Formaten findet eine “Übersetzung” statt
 - **Tunneling:** IPv6-Datagramme werden als Nutzdaten in IPv4-Datagrammen übertragen

Dual Stack

- IPv6 Knoten verfügt über einen IPv4 und einen IPv6 Stack
- Übersetzung von IPv4 in IPv6 Datagramme
 - Informationen gehen immer dann verloren, wenn eine IPv6 Information nicht auf eine IPv4 Information abgebildet werden kann (z.B. Flow)



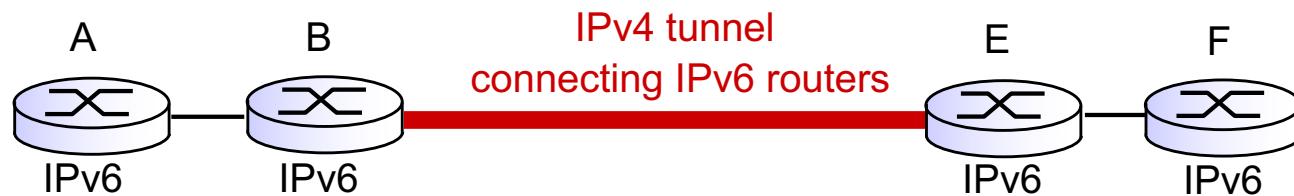
Tunneling



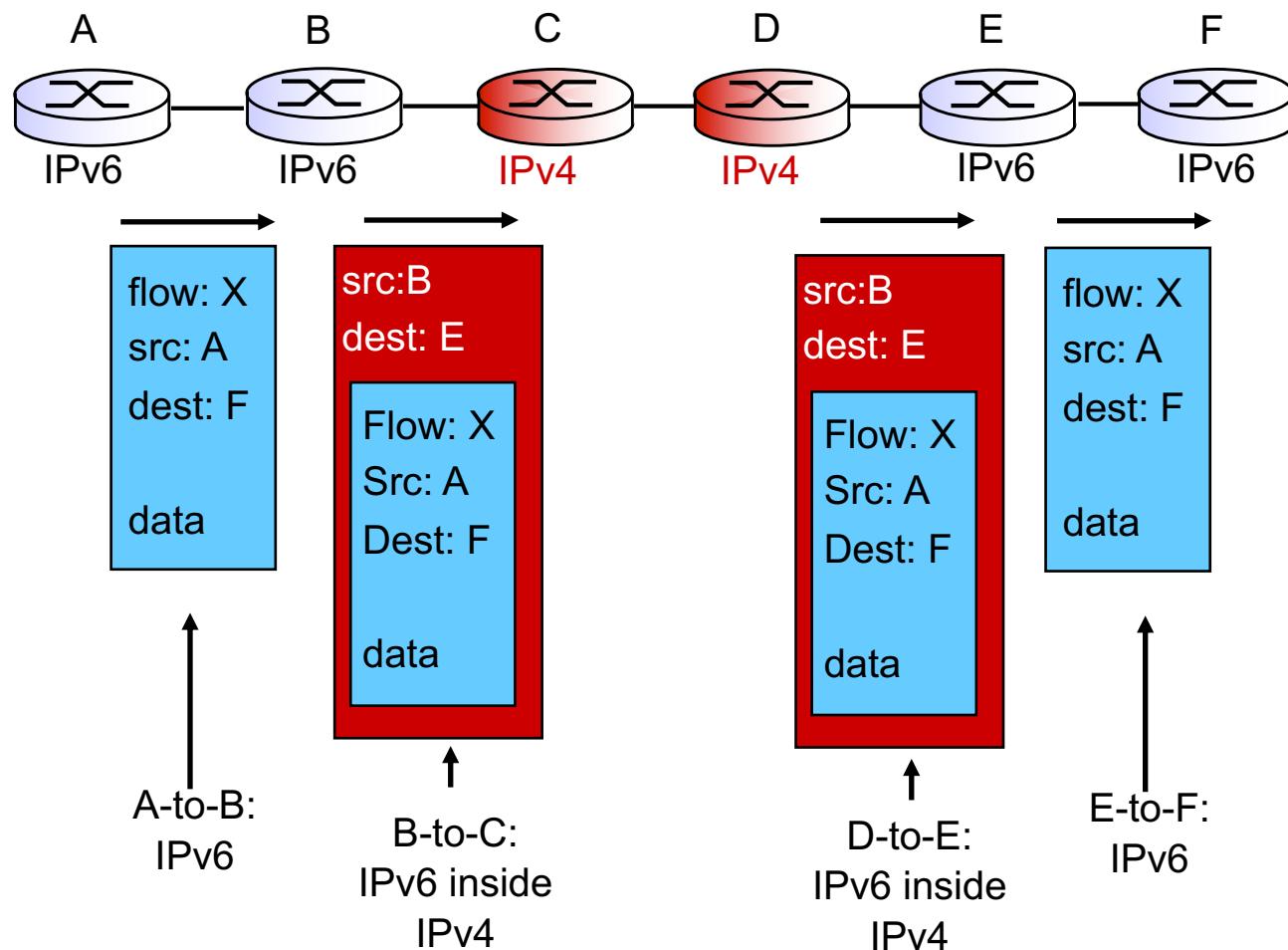
- Das IPv6 Paket wird als Payload in in IPv4 Paket gepackt und zwischen IPv4 Routern verschickt.

Tunneling

logical view:



physical view:



Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6
- Paketfilterung (Firewalls) 
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung

Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

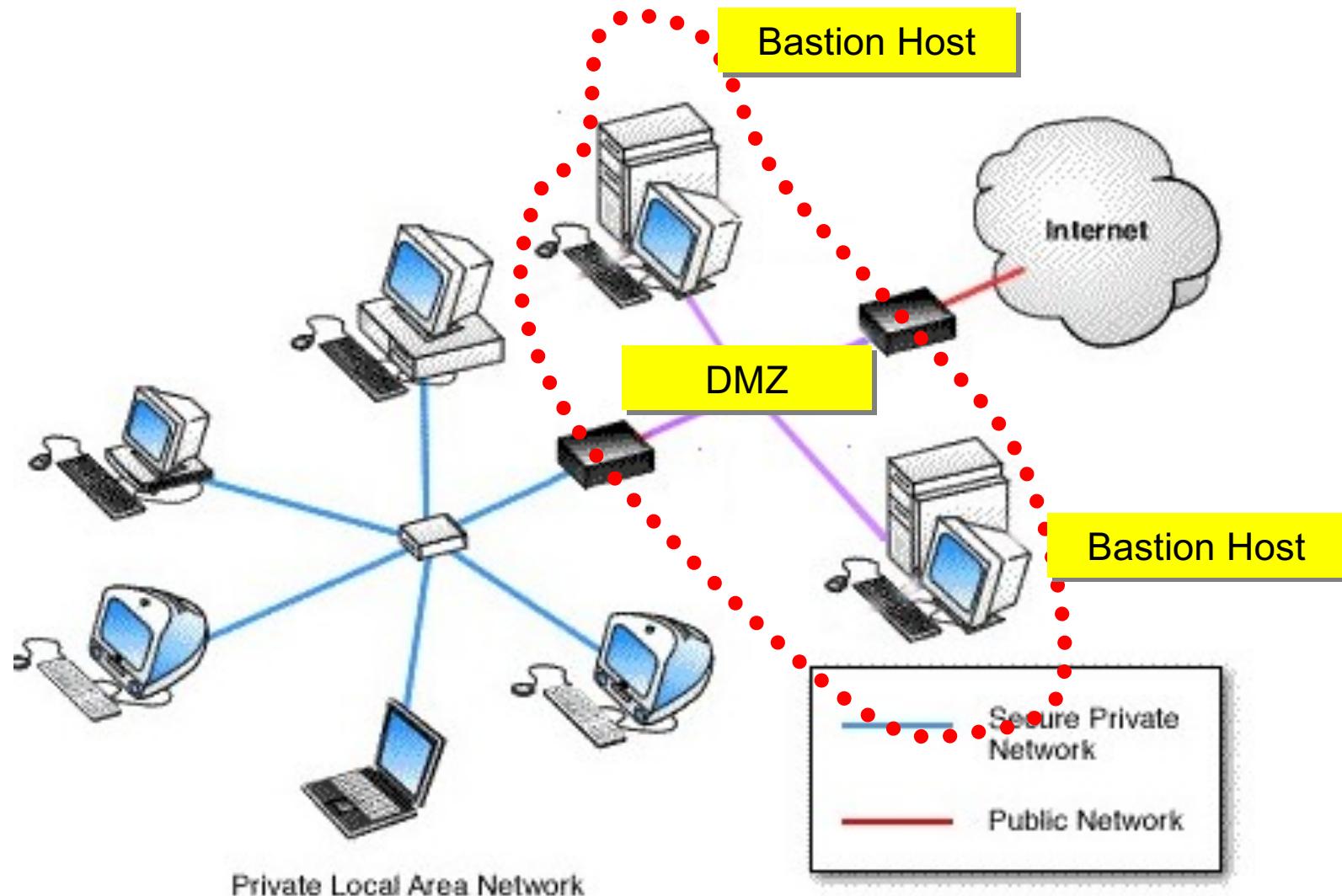
Firewall

- Vergleich mit Burgtor / Burggraben einer mittelalterlichen Burg:
 - Erlaubt Eintritt und Verlassen nur an einem bewachten Punkt
 - Verhindert, dass Angreifer an weitere Verteidigungsanlagen herankommen
- Grenze zwischen unsicherem und vertrauenswürdigem Netz
- Technische Umsetzung einer Sicherheitspolitik:
 - Durchlassen von akzeptablem Netzverkehr
 - Sperren von nicht akzeptablem Netzverkehr (Verwerfen, Protokollieren, ggf. weitergehende Analyse: Entdeckung von Angriffen/Störungen)

Begriffsbildung

- Eine Firewall kann aus den folgenden Komponenten bestehen:
 - **Paketfilter:** Anwendungsunabhängige Filterung der Datenpakete
 - **Applikationsfilter (= Proxy Server):** Anwendungsspezifische Filterung der Datenpakete
- Ein im Internet „sichtbarer“ Rechner heißt **Bastion Host**.
 - Ein Server, der Dienste an außen anbietet
 - Beim Einsatz mehrerer Firewall-Komponenten heißt der Netzbereich zwischen der ersten Firewall-Komponente vor dem Internet und der letzten Firewall-Komponente vor dem internen Netz „**Demilitarisierte Zone**“ (**DMZ**)

Firewall: Standard-Architektur



Paketfilter

- Filterung von Datenpaketen aufgrund von Informationen auf Netzwerk und Transport Layer:
 - Quell-IP-Adresse
 - Ziel-IP-Adresse
 - TCP/UDP Quell- und Ziel-Portnummern
 - ICMP Nachrichtentyp
 - TCP SYN und ACK Flags
- Spezifikation über Filterregeln / -tabellen
- Zwei Typen
 - Zustandslose Paketfilter (stateless): Untersuche Pakete unabhängig von Wissen über andere Pakete
 - Zustandsbasierte Paketfilter (statefull): Untersuche Pakete und nutze Wissen über vorherige Pakete (z.B. aktuelle TCP Verbindungen)

Zustandslose Paketfilter - Beispielkonfigurationen

Beispiel 1

- **Aufgabe:** Filter alle UDP Pakete und alle Telnet Verbindungen raus
- **Wissen:**
 - Erkenne Telnet Verbindung am Zielport 23 des Servers
 - Erkenne UDP Paket am IP-Protokollfeld Eintrag (upper layer) 17
- **Regel:** Blockiere ein- und ausgehende Pakete, bei denen im IP-Protokollfeld (“upper layer”) 17 steht oder deren Quell- oder Zielport 23 ist

Beispiel 2

- **Aufgabe:** Verhindert, dass externe Rechner TCP-Verbindungen mit einem internen Rechner aufbauen, erlaubt aber umgekehrt allen internen Clients Verbindungen nach außen.
- **Wissen:** Bei der Anfrage eines TCP Verbindungsaufbau ist ACK == 0
- **Regel:** Blockiere eingehende TCP-Segmente mit ACK=0

Beispiel für eine ACL einer Paketfilter-Firewall

- **ACL = Access Control List**
- **Beispiel:** ACL für das Subnet 222.22/16

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

- Anwendung der Regeln (übliches Verfahren)
 - Sequentielles zeilenweises Durchlaufen der Tabelle
 - Die Aktion in der ersten Zeile, in der alle Bedingungen erfüllt sind, wird ausgeführt!

Zustandsbasierte Paketfilter

- In der ACL der letzten Folie blockiert folgende sinnlosen Paket **nicht**
 - SRC Port 80, ACK Bit gesetzt aber TCP Verbindung existiert nicht
- z.B. Möglichkeit für eine DOS Attacke
- Lösung: Zustandsbasierte Paketfilter
 - z.B.: Erkenne den Anfang und das Ende einer TCP Verbindung bzw. nutze Timeout: Lasse nur Paket zu einer aktiven TCP Verbindung ins Netz

Zustandsbasierte Paketfilter - Beispielkonfigurationen

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Diskussion conxion

Einige Leitlinien zur Aufstellung von ACLs

- Möglichst frühe Filterung eingehender Pakete:
- Reihenfolge der Regeln beachten (meist sequentielle Abarbeitung)!
- Blockieren von Paketen unbekannter Protokolle!
- Blockieren von Paketen problematischer Dienste bzw. Protokolle,
- **Grundsätzlich: Alles sperren**, außer wohlbekannten und benötigten Protokollen/Diensten (→ Ports)
- CERT bietet Einstellungen für Paketfilter an (www.cert.org)

Diskussion Paketfilter

Vorteile

- Zugriff auf Netzdienste geschieht völlig transparent
- Die meisten Router unterstützen die Angabe von Filterregeln, so dass keine teure Zusatzhardware nötig ist

Nachteile

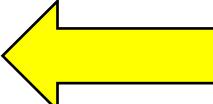
- Konfiguration kann sehr komplex werden
- Zustandsbehaftete Paketfilter sind teilweise noch komplexer
- Maleware kann nicht erkannt werden

Intrusion Detection Systems (IDS)

- Paketfilter erkennen zum Beispiel Malware (Schadsoftware) nicht
- Analyse der Payload durch IDS
 - Signaturbasierte Systeme
 - Angriffe müssen bekannt sein
 - Signaturen müssen kontinuierlich nachgepflegt werden
 - Anomaliebasierte Systeme
 - Erstellt Verkehrsprofil
 - Erkenne statistisch ungewöhnliche Ströme (z.B.: hoher Anteil an ICMP Pakete)
 - Vorteil: Unbekannte Angriffe können erkannt werden.
 - Problem: Unterscheidung zwischen „normalem“ und „ungewöhnlichem“ Verkehr

Kapitel 5: Netzwerkschicht & Routing

Gliederung

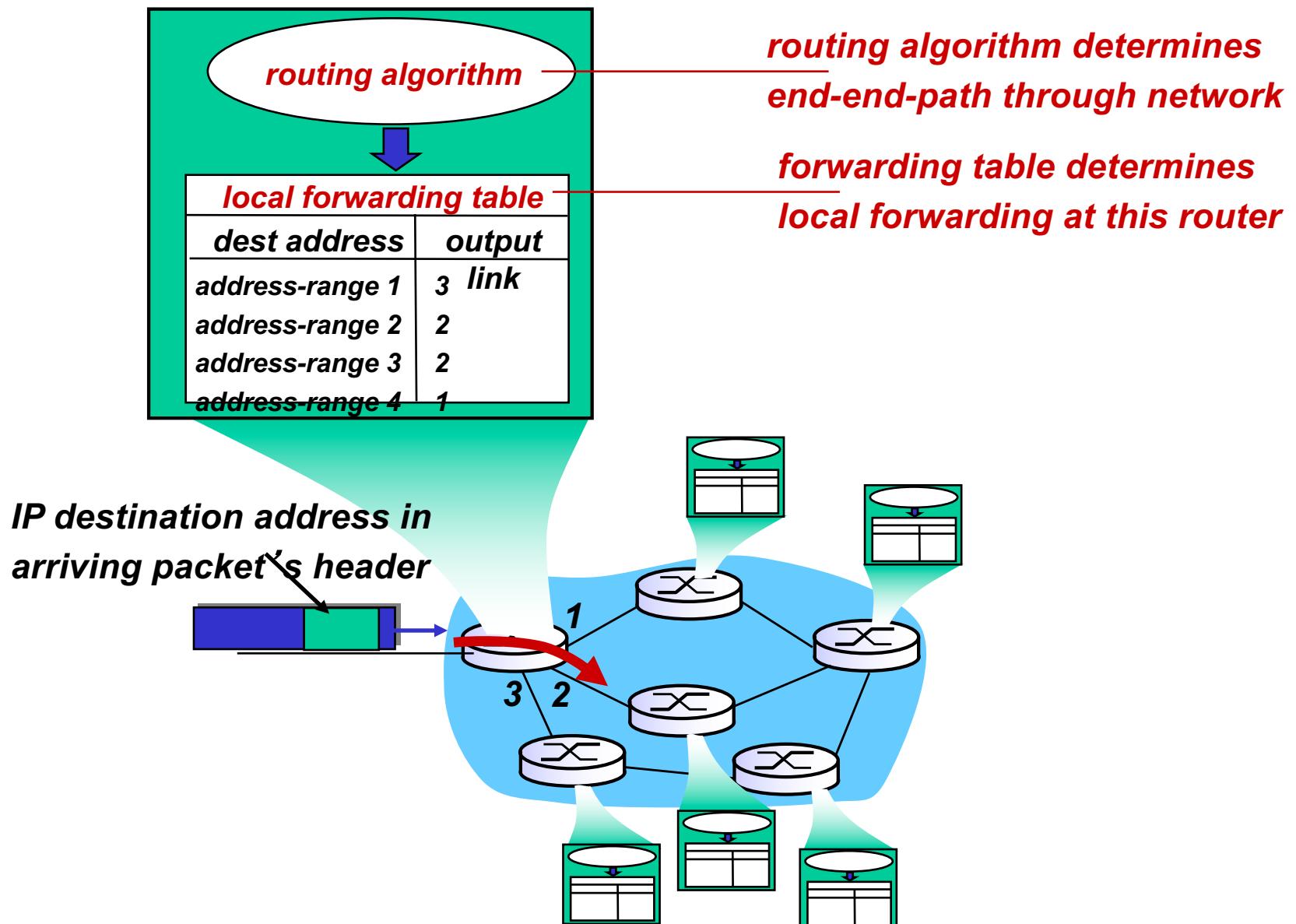
- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6
- Paketfilterung (Firewalls)
- Routing-Algorithmen 
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung

Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

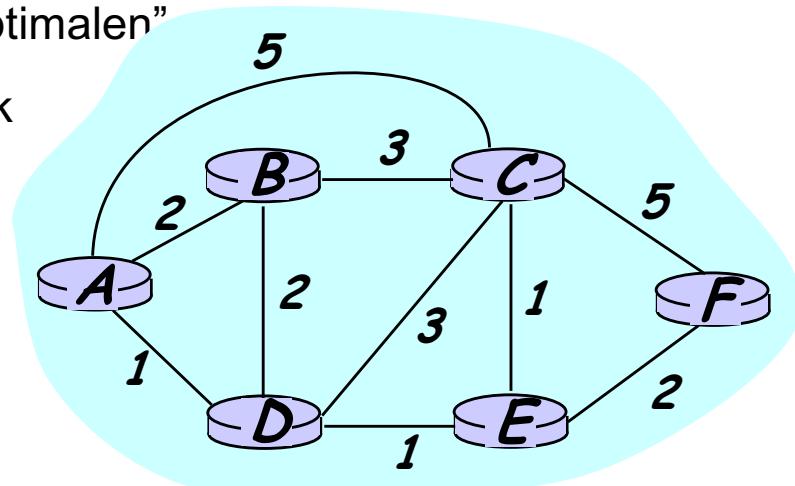
J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

Routing und Forwarding



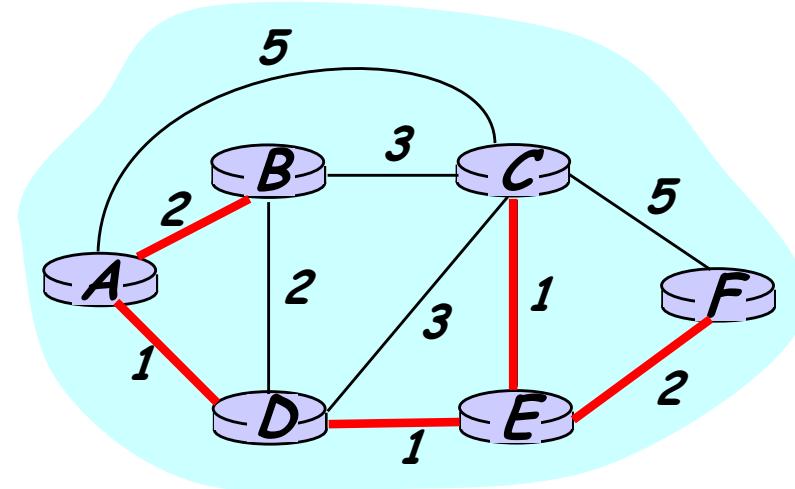
Routing

- **Ziel des Routingalgorithmus:** Ziel: finde “optimalen” Pfad (Folge von Routern) durch das Netzwerk
- **Graph-Abstraktion** für Routingalgorithmen:
 - Knoten im Graph sind Router
 - Graph-Kanten sind die physikalischen Verbindungen (“Links”)
 - Verbindungskosten: Verzögerung, € Kosten, Staugefahr, ...
- **“optimaler” Pfad:**
 - heißt meist minimale Kosten
 - andere Definitionen sind möglich
 - Wichtig: Nur **Kostenwerte ≥ 0 sind erlaubt!**
- **Einstieg ins Netzwerk:**
 - Standard Router / First Hop Router: Zuordnung Host – Router
 - Ergeben Quell – und Ziel-Router für einen Pfad



Abstraktion durch Graphen

- graph: $G = (N, E)$
- N = Menge der Knoten/Router
 $= \{ A, B, C, D, E, F \}$
- E = Menge der Kanten
 $= \{ (A, B), (A, D), (B, C), (B, D), (C, D), (C, E), (C, F), (D, E), (E, F) \}$



- **Eigenschaft optimaler Pfade**
 - Wenn Router D auf dem optimalen Pfad r von Router A zu Router F liegt, dann ist der optimale Pfad von D zu F ein Teil von r
 - Folgerung: **Die optimalen Pfade von A zu allen möglichen Zielen bilden einen Baum mit Wurzel A**

Klassifizierung von Routing-Algorithmen

Globale vs dezentrale Information?

- **Global:** Alle Router kennen die komplette Topologie / Verbindungskosten
 - “Link state”-Algorithmen
- **Dezentral:** Jeder Router kennt die physikalisch direkt verbundenen Nachbarn mit den entsprechenden Verbindungskosten
 - Iterativer Berechnungsprozess, Austausch der Information mit den direkten Nachbarn
 - “Distanz-Vektor”-Algorithmen

Statisch vs. dynamisch?

- **Statisch:** Routen ändern sich langsam mit der Zeit
- **Dynamisch:** Routen ändern sich häufig
 - periodische Updates
 - in Reaktion auf die Änderung der Verbindungskosten

Lastsensitive vs. lastinsensitive Algorithmen

- Leitungskosten ändern sich dynamisch gemäß der aktuellen Last

Ein Link-State (LS) Routing Algorithmus

Dijkstra's Algorithmus

- Netz-Topologie, Verbindungskosten in allen Knoten bekannt
 - Erreicht durch “Link state broadcast”: Rundsenden der Identitäts- und Kosteninformationen
 - Alle Knoten haben die gleiche Information
- Berechnet die kürzesten Pfade von einem Knoten ('Quelle') zu allen anderen
 - Ergibt Routing-Tabelle für diesen Knoten
- Kosten der Kanten sind konstant und größer gleich 0.

Notation:

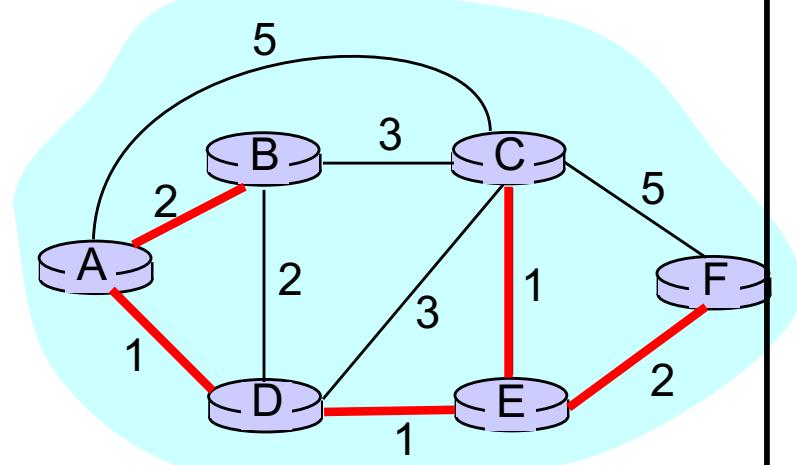
- **A**: Quelle
- **c(i,j)**: Verbindungskosten von Knoten i nach j, Kosten ∞ , wenn nicht direkter Nachbar
- **D(v)**: Aktueller minimaler Wert der Pfadkosten von der Quelle zu Knoten v
- **p(v)**: Vorgängerknoten von v auf dem momentan besten Weg von der Quelle nach v
- **N**: Menge der Knoten, für die die minimalen Pfadkosten bereits feststehen (fertig)

Dijkstra's Algorithmus

```
1  /* Initialisierung: */
2  N = {A}
3  for alle Knoten v ∈ N
4      if v direkter Nachbar von A
5          then D(v) = c(A,v)
6      else D(v) = ∞
7  Loop
8  Finde den Knoten w ∈ N mit: D(w) ist minimal
9  Füge w zu N hinzu /*Kürzester Pfad zu w steht fest*/
10   for alle Knoten v ∈ N ,
11       die direkte Nachbarn von w sind:
12       D(v) = min{ D(v) , D(w) + c(w,v) }
13   /* Die neuen Kosten der direkten Nachbarn v sind
14       entweder die alten Kosten oder die bekannten
15       Kosten des kürzesten Pfades zu w zuzüglich der
16       Kosten von w zu v */
17
18 until alle Knoten sind in N
```

Dijkstra's Algorithmus: Beispiel für Quelle A

Iteration	N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	∞	∞
→ 1	AD	2,A	4,D		2,D	∞
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
→ 5	ADEBCF					

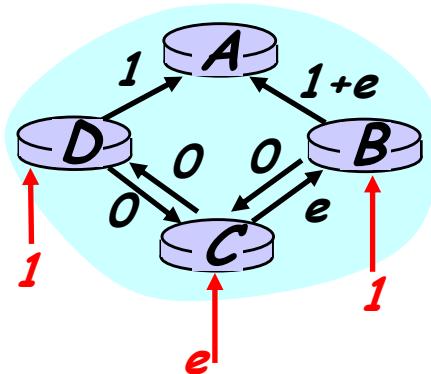


Ermittlung der Routing-Tabelle für A:
Vom Ziel über $p(v)$ Pfad zu A zurückverfolgen

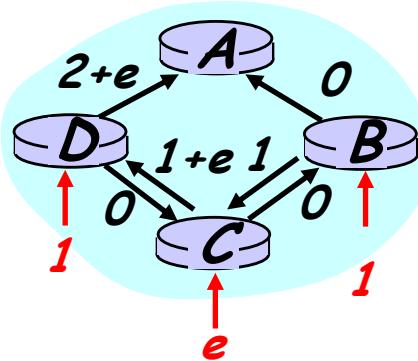
Zielknoten (Zielnetz)	Nächster Router	Kosten
B	B	2
C	D	3
D	D	1
E	D	2
F	D	4

Dijkstra's Algorithmus: Diskussion

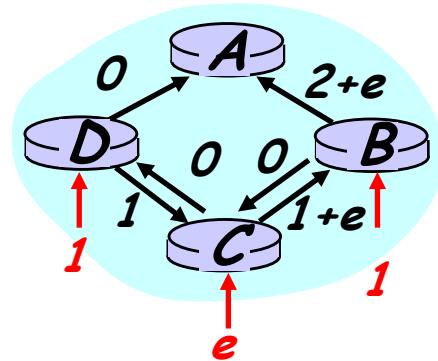
- Komplexität des Algorithmus bei n Knoten
 - jede Iteration: alle Knoten, die nicht in N sind, müssen geprüft werden
 - $n*(n+1)/2$ Vergleiche: $O(n^2)$
 - effektivere Implementierungen erreichen: $O(n * \log n)$
- Oszillationen sind möglich, wenn die Kosten sich über die Zeit ändern
z.B. wenn Verbindungskosten = aktuelle Verkehrslast auf der Verbindung



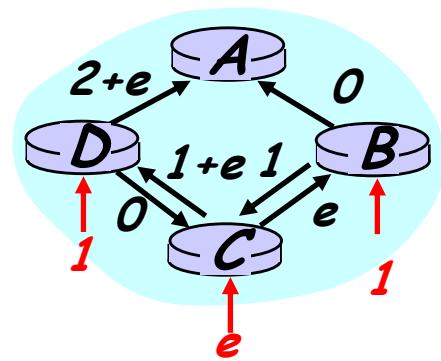
Anfangsrouting
→ B,C,D erzeugen
Last für A



B,C erkennen
besseren
Pfad zu A
im Uhrzeigersinn



B,C,D erkennen
besseren
Pfad zu A
entgegen dem
Uhrzeigersinn



B,C,D erkennen
besseren
Pfad zu A
im Uhrzeigersinn

Distanz-Vektor Routing: Überblick

Verteilt:

- jeder Knoten kommuniziert nur mit seinen direkten Nachbarn

Iterativ:

- stoppt, wenn kein Knoten mehr Infos austauscht
- Selbstterminierend: kein “Stop”-Signal notwendig

Asynchron:

- Austausch muss nicht synchron getaktet sein!

Distanz-Vektor Routing: Grundlage

Bellman-Ford Gleichung

Sei

$d_x(y) :=$ Kosten für den günstigsten Pfad von x nach y

dann gilt

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

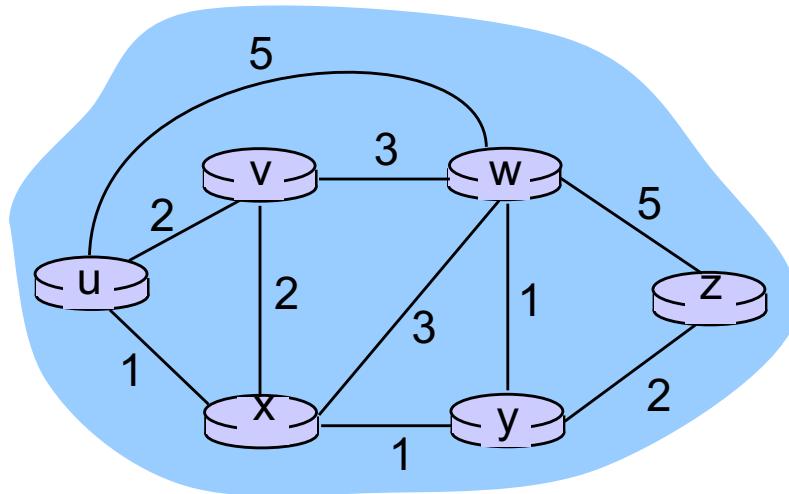
v

Kosten für den günstigen Pfad von Nachbarn zu y

Kosten von x um Nachbarn

Minimum über alle Nachbarn v von x

Beispiel zur Bellman-Ford Gleichung



Es gilt: $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Bellman-Ford Gleichung besagt

$$\begin{aligned}d_u(z) &= \min \{ c(u,v) + d_v(z), \\&\quad c(u,x) + d_x(z), \\&\quad c(u,w) + d_w(z) \} \\&= \min \{ 2 + 5, \\&\quad 1 + 3, \\&\quad 5 + 3 \} = 4\end{aligned}$$

Idee DV Algorithmus: Jeder Knoten erhält d von seinen Nachbarn und berechnet sein d auf Basis der Bellman-Ford Gleichung

Distance Vector (DV) Algorithmus

Distanzvektor: $D_x(y)$ = minimalen Kosten von x nach y (anfänglich eine Abschätzung)

Datenstrukturen pro Knoten x

- $D_x = [D_x(y): y \in N]$
- Für alle Nachbarn v: $c(x,v)$ – Kosten des Links zum Nachbarn v
- Für alle Nachbarn v: Der Distanzvektor von v: $D_v = [D_v(y): y \in N]$

Verteilter Algorithmus

- Die Kosten müssen konstant sein.
- Immer wieder sendet jeder Knoten seinen Distanzvektor zu den Nachbarn
- Trifft eine Distanzvektor ein, dann aktualisiert x seinen DV:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

Distance Vector (DV) Implementation

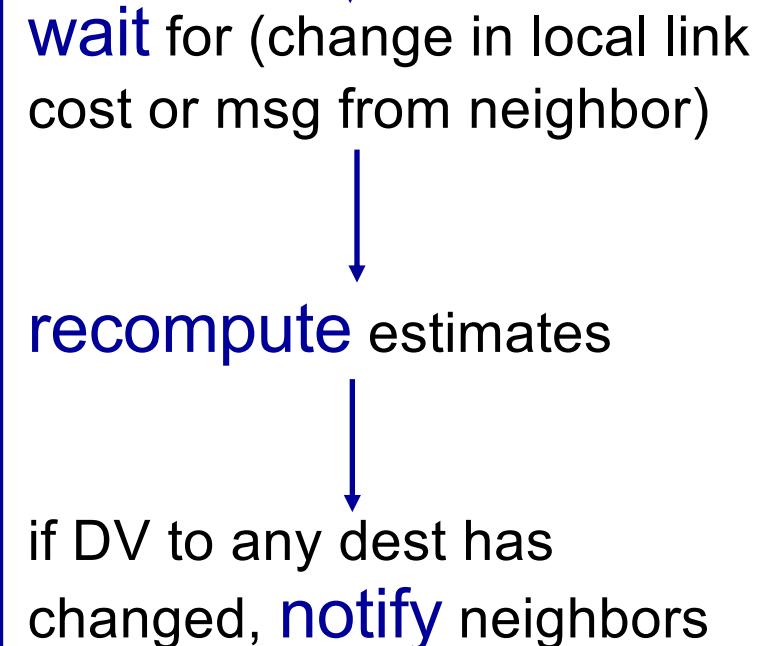
iterative, asynchron:

- Knoten führt seinen Berechnung aus, wenn sich die Kosten des Links oder der Distanzvektor eines Nachbarn geändert hat.

verteilt:

- Ein Knoten verteilt seinen Distanzvektor nur, wenn er sich geändert hat.

For each node:



Beispiel

node x table

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
z		∞	∞	∞

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
z		7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
z		3	1	0

node y table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
z		∞	∞	∞

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
z		7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
z		3	1	0

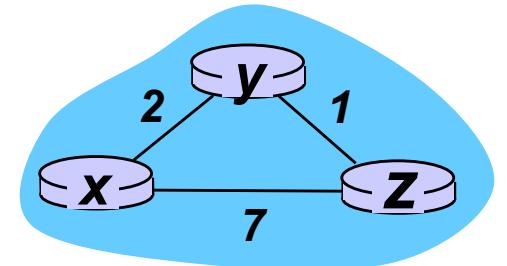
node z table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
z		7	1	0

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
z		3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
z		3	1	0

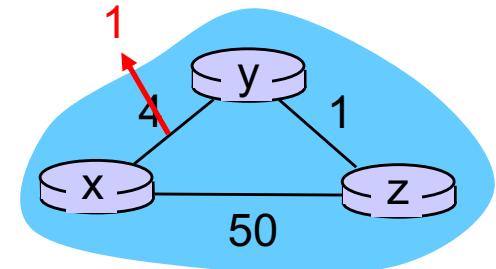
time



DV Algorithmus: Änderungen der Link Kosten

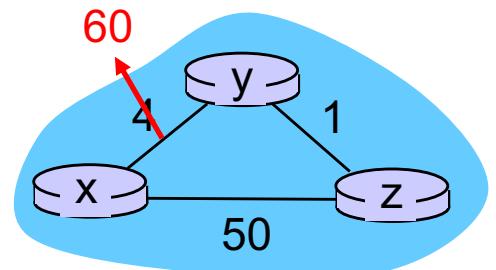
Änderungen der Kosten eines Links

- Knoten erkennt Änderungen der Link Kosten
- Neue Berechnung des DV
- Informiere Nachbarn, falls DV sich geändert hat.



“good news travels fast”

- Durch die Bildung des Minimums in
- $$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$
- werden Verbesserungen schnell propagiert.
 - Werden Verschlechterungen langsam in vielen Iterationen propagiert. -> **Count-to-Infinity-Problem**

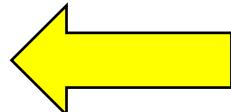


“bad news travels slow”

Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6
- Paketfilterung (Firewalls)
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS
- Zusammenfassung



Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

Routing: Praxisprobleme

Bisher idealisierte Situation

- Alle Router sind identisch
- Netzwerk ist “flach”

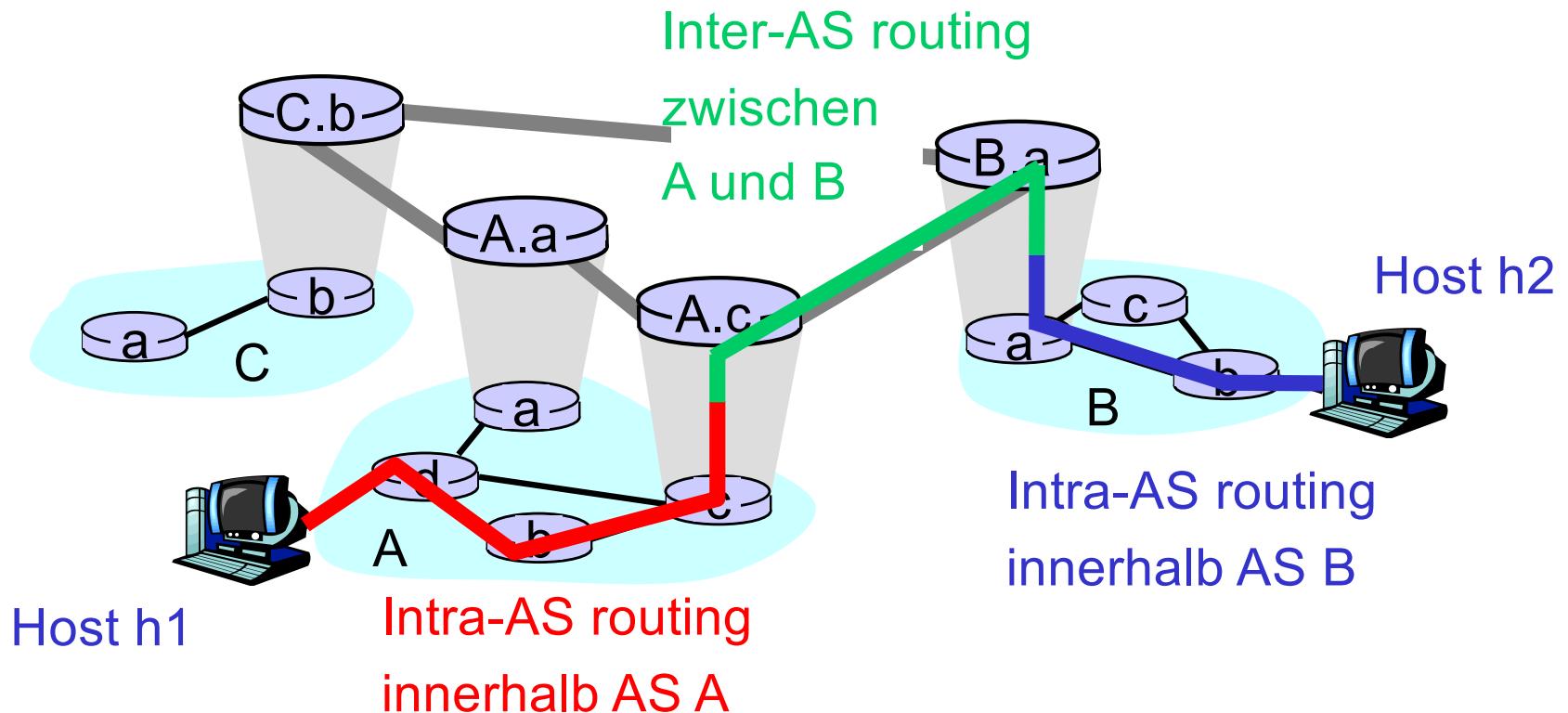
Praxis

- Größe: mit > 100 Millionen Zieladressen:
 - Nicht alle Ziele können in einer Routing-Tabelle gespeichert werden!
 - Austausch von Routing-Informationen würde das Netz überlasten!
- Administrative Autonomie
 - Internet = Netzwerk von Netzwerken
 - Jeder Netzwerk-Admin möchte Routing im eigenen Subnetz beeinflussen können
- **Lösung:** Hierarchie

Hierarchisches Routing: Idee

- **Autonome Systeme (AS):** Zusammenschluss von Routern zu Regionen
 - Router in demselben AS benutzen das gleiche Routing-Protokoll
 - **“Intra-AS”-Routing-Protokoll**
 - Router in unterschiedlichen AS können unterschiedliche Intra-AS-Routing-Protokolle benutzen
- **Gateway Router** verbinden die AS untereinander
 - Spezielle Router im AS
 - “Sprechen” Intra-AS- Routing-Protokoll mit allen anderen Routern im AS
 - AS übergreifenden Routing wird durch **Inter-AS-Routing-Protokoll** realisiert.
 - Gateway Router sind für Routing zu Zielen außerhalb des AS verantwortlich
 - Nutzen Inter-AS-Routing-Protokoll mit anderen Gateway-Routern

Beispiel: Intra-AS und Inter-AS Routing



- Drei AS: A, B, C
- Gateway Router: C.b, A.a, A.c, B.a
- Beispiel: Ein Datagramm wird von h1 zu h2 geroutet.

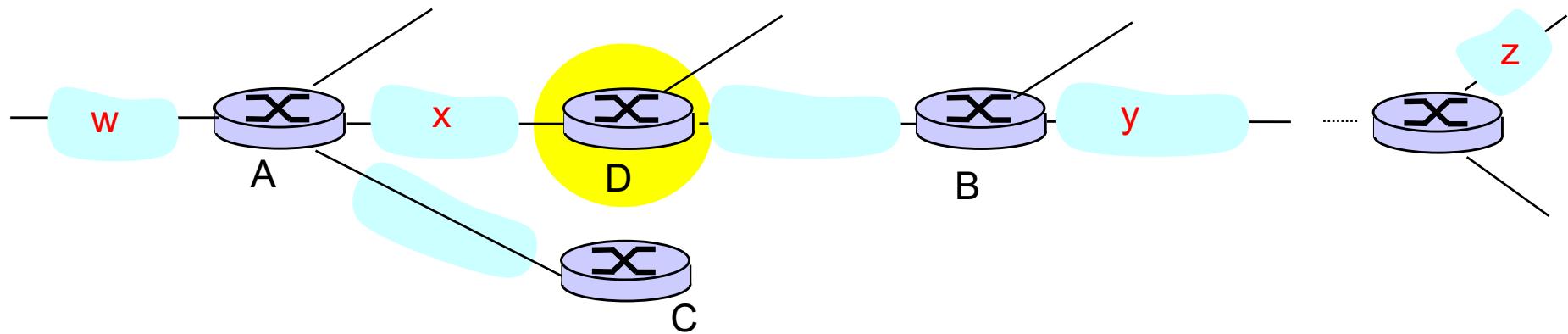
Intra-AS Routing Protokolle

- Auch genannt **Interior Gateway Protocols (IGP)**
- Wichtigste Protokolle:
 - **RIP**: Routing Information Protocol (RFC 1058)
 - **OSPF**: Open Shortest Path First (RFC 2178)
 - **IGRP**: Interior Gateway Routing Protocol
(proprietäres Cisco-Protokoll)

RIP (Routing Information Protocol)

- Distanzvektor-Algorithmus
- Pfadkosten: Anzahl Hops
 - Count-to-Infinity-Problem tritt nur noch beim Wegfall einer Verbindung auf.
 - Kosten jeder Verbindung = 1
 - Maximal 15 Hops erlaubt!
- Routinginformationen werden regelmäßig alle 30 Sekunden mittels einer “RIP Response Message” (auch RIP-Advertisement genannt) ausgetauscht
- Jedes Advertisement enthält die Routingtabelle des Senders für bis zu 25 Zielnetzwerke innerhalb des AS
- Enthalten in Standard-UNIX Distributionen
- CIDR-Unterstützung erst ab Version 2
- Wird in Routern nur noch selten benutzt

Beispiel RIP



Routingtabelle von D

Destination Network	Next Router	Num. of hops to dest.
w	A	1
y	B	1
z	B	6
x	D	0
....

RIP Beispiel: Advertisement von A erreicht D

Alte Routingtabelle von D

Destination Network	Next Router	Num. of hops to dest.
w	A	1
y	B	1
z	B	6
x	D	0
....

Advertisement = Routingtabelle von A

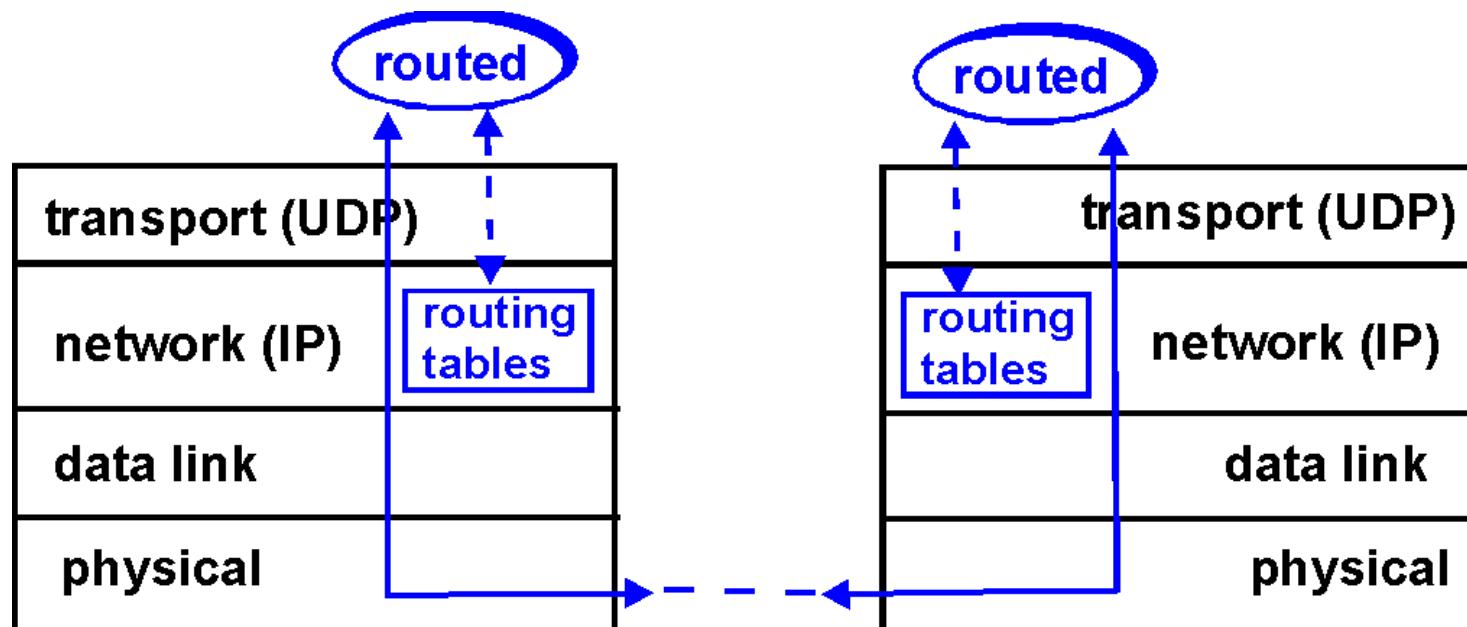
Destination Network	Next Router	Num. of hops to dest.
z	C	3
w	A	0
x	A	0
....

Neue Routingtabelle von D

Destination Network	Next Router	Num. of hops to dest.
w	A	1
y	B	1
z	A	4
x	D	0
....

RIP-Implementierung in UNIX

- RIP ist in Unix als ein Hintergrund-Anwendungsprozess namens **routed** implementiert (“route-daemon”)
- Der routed-Prozess darf die Routingtabellen im Kernel aktualisieren!
- Advertisements werden in UDP-Paketen versendet (Port 520)



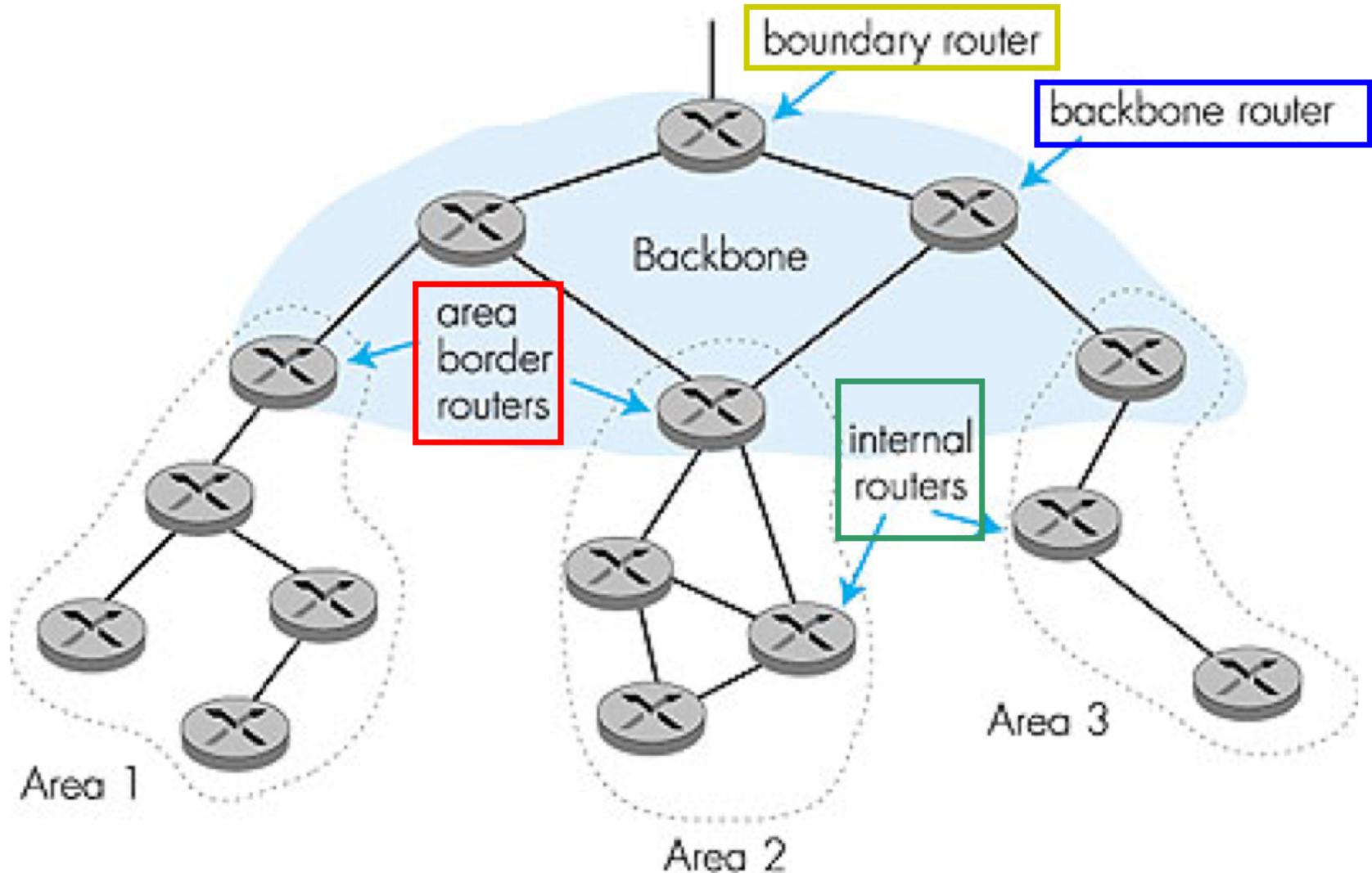
OSPF (Open Shortest Path First)

- “Offener” Standard seit 1990 (RFC 2178)
- Nachfolger von RIP
- Benutzt “Link State” Algorithmus
 - Zustandsinformationen werden an **alle** Router über periodische “Link state broadcast” – Nachrichten verbreitet
 - Ein OSPF-Advertisement enthält nur die aktuellen Verbindungskosten zu den direkten Nachbarn
 - Jeder Router kennt die gesamte Topologie des AS inkl. Verbindungskosten
 - Routenberechnung gemäß Dijkstra-Algorithmus

OSPF: Zusätzliche Eigenschaften

- **Sicherheit:** Alle ausgetauschten OSPF-Nachrichten werden authentifiziert und über TCP-Verbindungen gesendet
- Mehrere Pfade mit denselben Kosten können parallel verwendet werden (in RIP nur ein Pfad) → *Lastverteilung!*
- Für eine Verbindung zwischen zwei Routern können verschiedene Verbindungs-kosten in Abhängigkeit vom TOS-Wert (Type of Service) im IP-Datagramm definiert werden
- Integrierte Unterstützung von Unicast- und Multicast-Routing
- Hierarchische Strukturierung großer Autonomer Systeme
 - Aufteilung in Bereiche (“Areas”), innerhalb derer ein eigener Link-State-Algorithmus angewendet wird

OSPF-Hierarchie: Beispiel



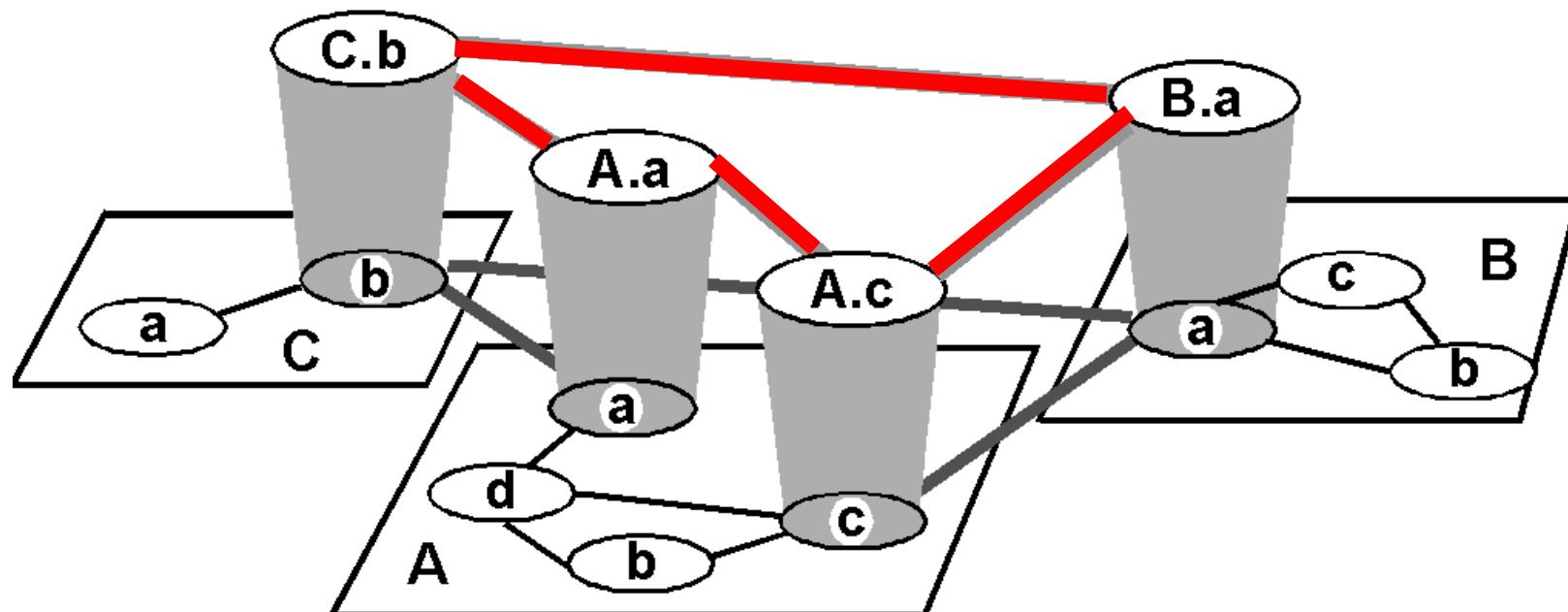
OSPF-Hierarchie

- **Zwei-Ebenen-Hierarchie:** Mehrere “lokale” Bereiche (Areas) und zusätzlich ein “Backbone”-Bereich
- Link-State-Informationsaustausch nur innerhalb eines Bereichs
- Der Backbone-Bereich dient nur zur Weiterleitung zwischen den lokalen Bereichen oder für den Verkehr nach “Außen”
- Typen von OSPF-Routern:
 - **Interne Router:** Router für den Intra-AS-Verkehr innerhalb eines lokalen Bereichs
 - **Area Border Router:** Gehören zu einem lokalen Bereich und zum Backbone → dienen als Gateway für den bereichsübergreifenden Verkehr
 - **Backbone Router:** Gehören zum Backbone und führen das Routing innerhalb des Backbone durch
 - **Boundary Router:** Gehören zum Backbone und sind mit Routern aus anderen AS verbunden → dienen als Gateway für den externen Verkehr mit anderen autonomen Systemen

IGRP (Interior Gateway Routing Protocol)

- Proprietäres CISCO-Protokoll
- Ebenfalls als RIP-Nachfolger entwickelt
- Distanz-Vektor-Protokoll (wie RIP)
- Konfigurierbare Kostenmetriken (z.B. Verzögerung, Übertragungskapazität, Verfügbarkeit, Last, ..)
- Verwendet TCP statt UDP
- Updateinformationen werden nur bei Veränderungen ausgetauscht (nicht periodisch wie bei RIP)
- Berechnung schleifenfreier Routingpfade aufgrund des “Distributed Updating Algorithmus” (DUAL)

Thema: Inter-AS-Routing

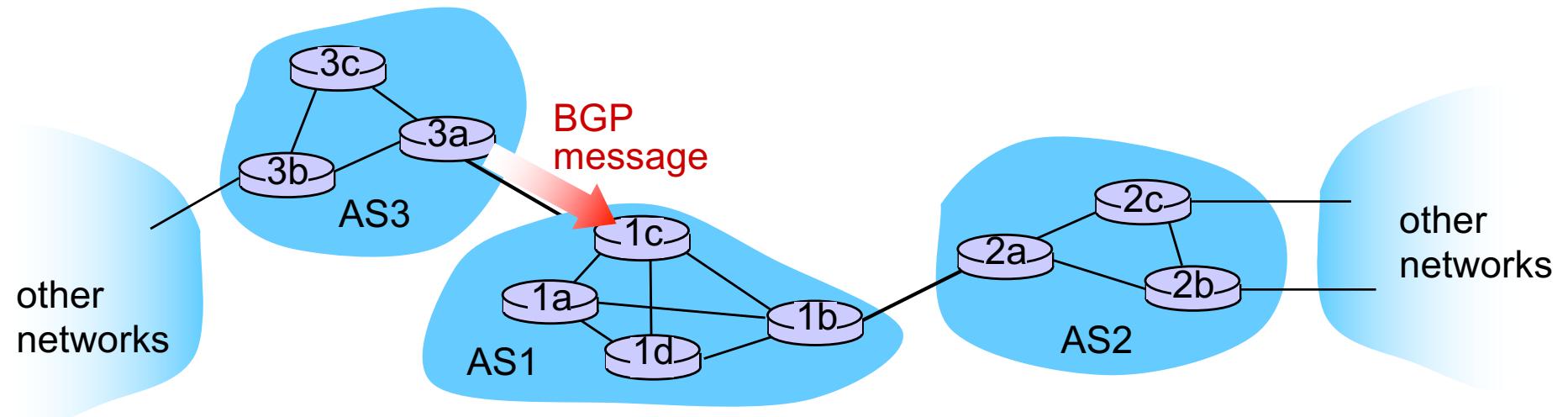


Inter-AS-Routing im Internet: BGP

- **BGP (Border Gateway Protocol):** *Der de-facto-Standard*
- Keine Routenberechnung: Gute Routen werden auf Basis von ISP internen Strategien, ökonomischen und firmenpolitischen Vorgaben erwählt. Natürlich muss die Erreichbarkeit gegeben sein.
- Hauptfunktion: Verteilung von Pfadinformationen
 - Jeder Boundary Router erhält Informationen über erreichbare Subnetze von seinen AS Nachbarn über **eBGP**
 - Jeder Boundary Router propagiert dieser Informationen an alle internen Router des AS über **iBGP**

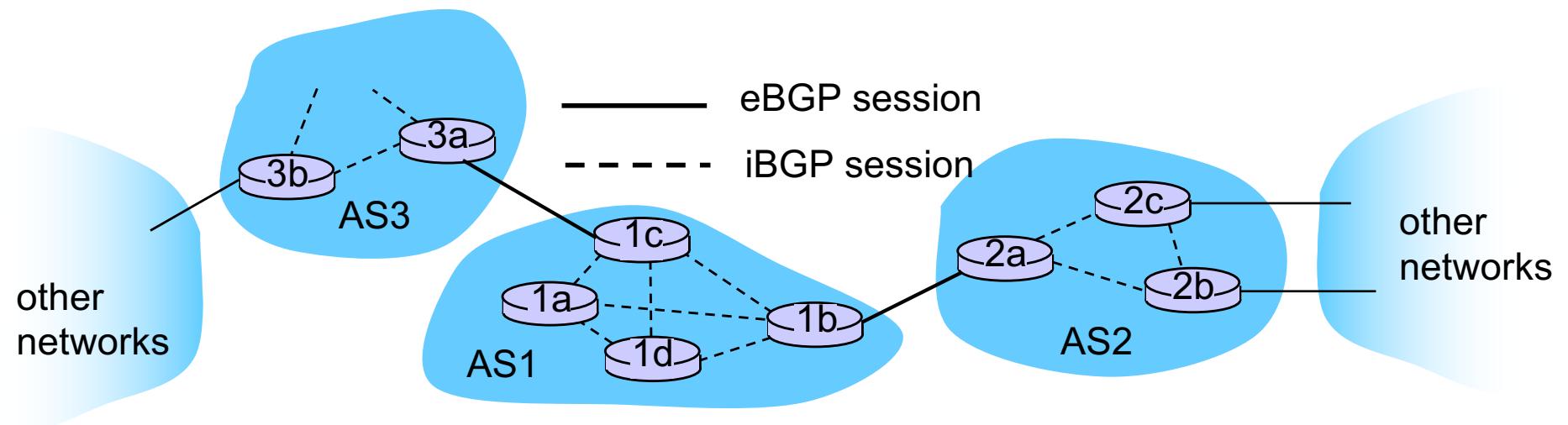
BGP Grundlagen

- **BGP session:** Zwei BGP Router (peers) tauschen BGP Messages aus.
 - Bietet Pfade (Sequenz von eindeutigen AS Nummern) zu unterschiedlichen Subnetzwerken an
 - Basiert auf einer semi-permanenten TCP Verbindung (Port 179)
- Was bedeutet ein Advertisement von AS3 an AS1
 - AS3 verspricht, dass Datagramme bezüglich dieses Prefix weiterleiten wird
 - AS3 kann mehrere Subnetze gemäß CIDR zusammenfassen



BGP: Weiterleiten von Advertisements

- Über eBGP erhält ein Boundary Router ein **Advertisements eines Nachbar AS**
 - Der Boundary Router kann dieses Advertisement via iBGP an alle Router des AS weiterleiten
 - Der Boundary Router kann das Advertisement an andere AS weiterleiten
 - Das Verhalten wird gemäß firmen-politischen und ökonomischen Regeln festgelegt.
 - Gründe: Günstigerer Pfad – was immer günstig bedeutet - vorhanden, Schleifenbildung, Vermeidung eines AS, ...

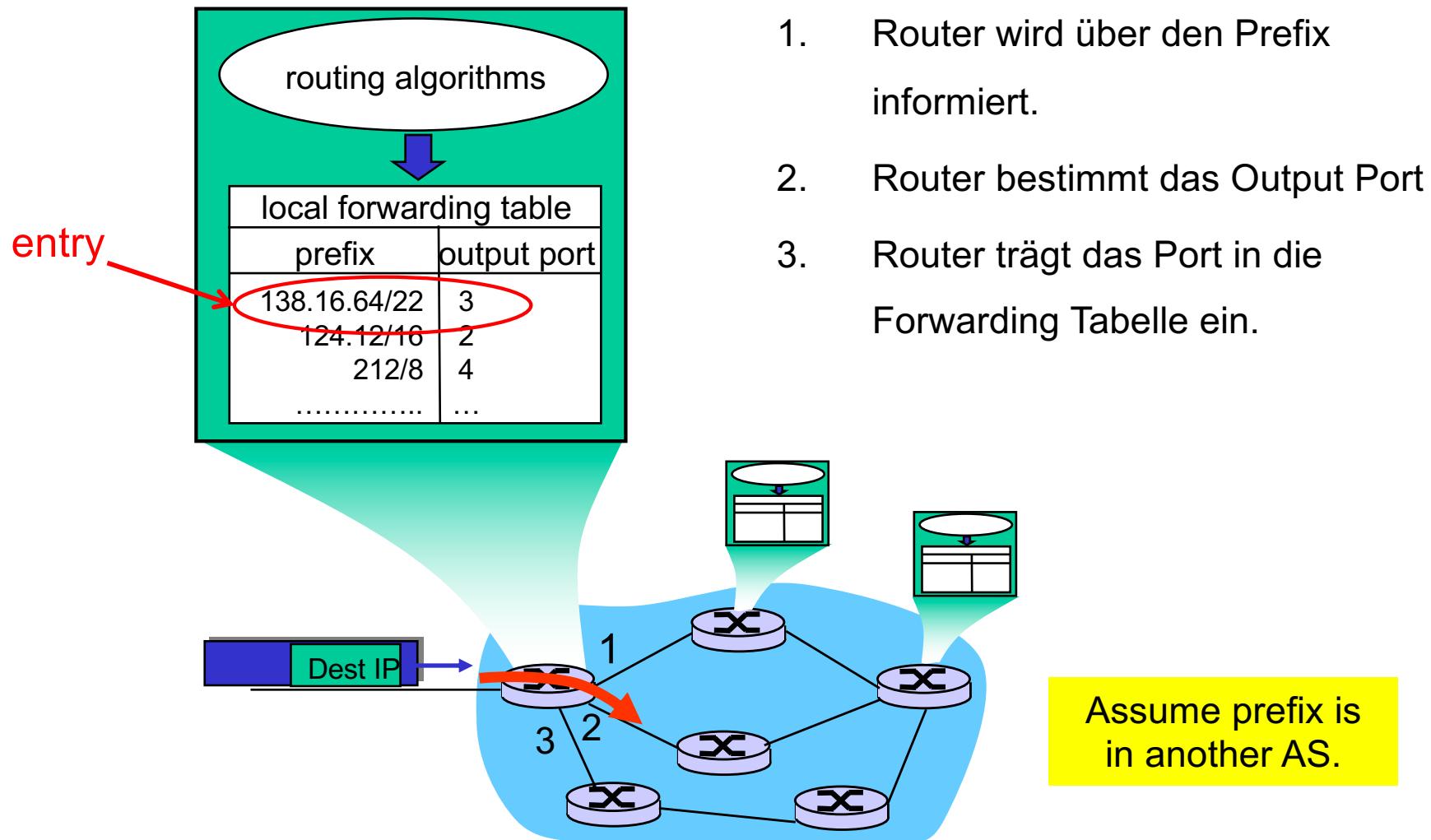


BGP-Nachrichten

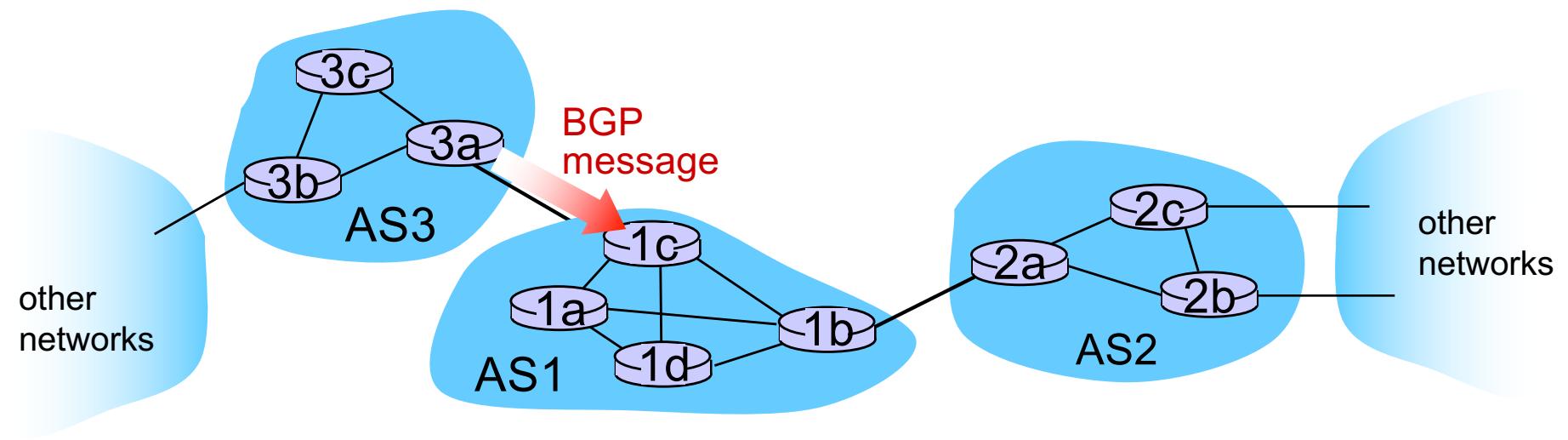
- BGP-Nachrichten werden über TCP (Port 179) ausgetauscht!
- Nachrichtentypen:
 - **OPEN**: Öffnet eine TCP-Verbindung zum Peer und authentifiziert den Sender
 - **UPDATE**: Aktualisiert eine Pfadinformation
 - **KEEPALIVE**: Hält die TCP-Verbindung offen, falls keine Updateinformationen vorliegen
 - **NOTIFICATION**: Fehlermeldung oder Anzeige des Verbindungsendes

Wie wird ein Eintrag der Forwarding Table erstellt?

Zusammenfassung

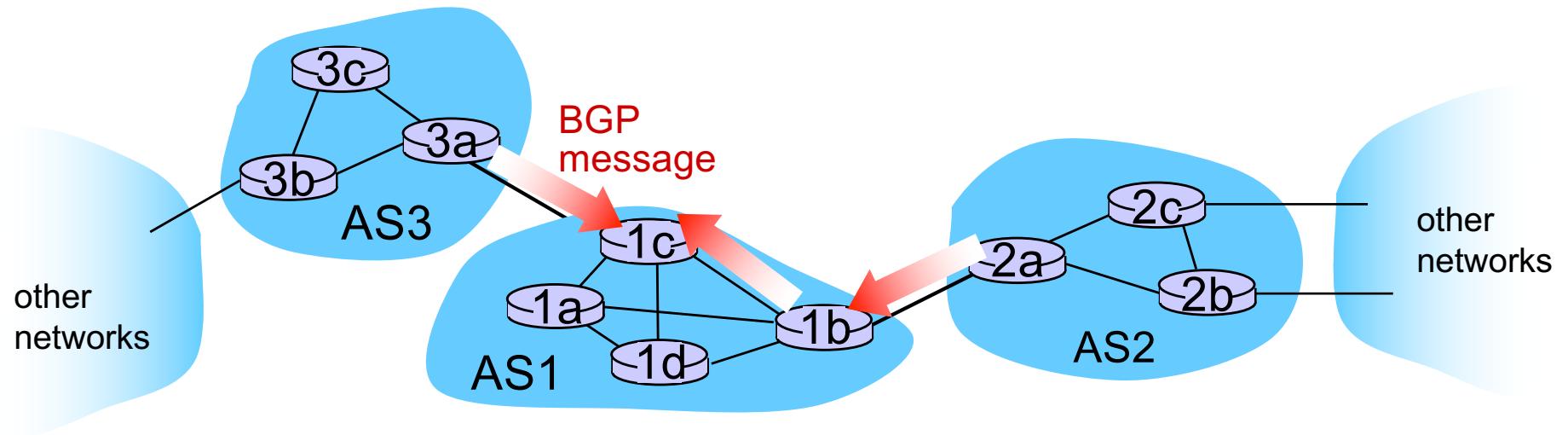


Router wird über den Prefix informiert



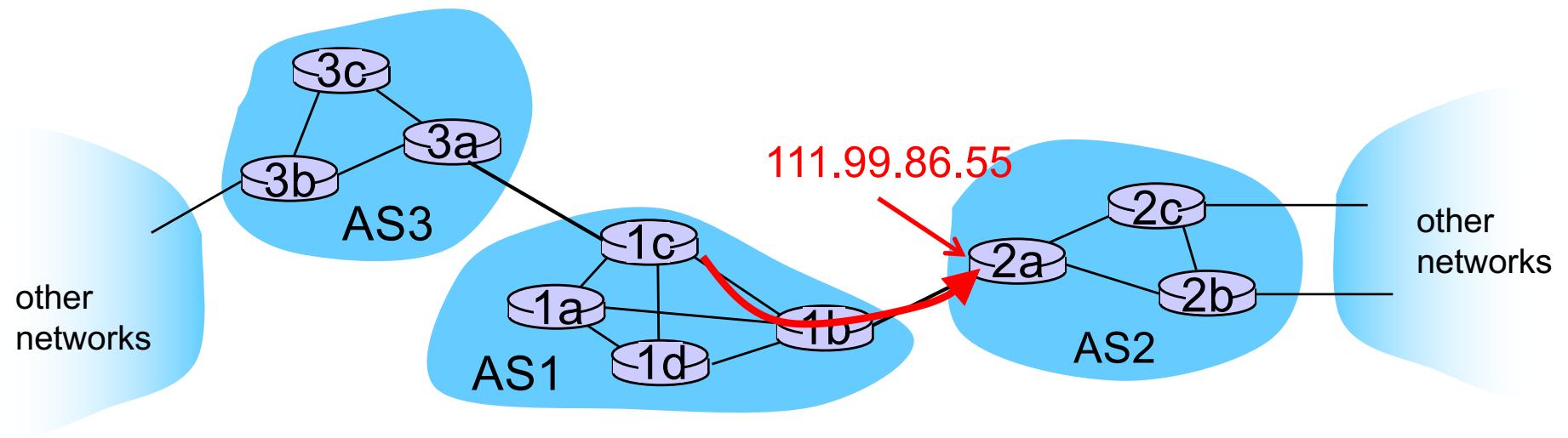
- Eine BGP Message enthält eine Menge von Routen
- Eine Route besteht aus dem AS-Pfad (die AS, die auf der Route liegen) und dem NEXT-HOP
- Beispiel
 - Prefix:138.16.64/22 ; AS-PATH: AS3 AS131 ; NEXT-HOP: 201.44.13.125

Router kann für einen Prefix mehrere Routen erhalten



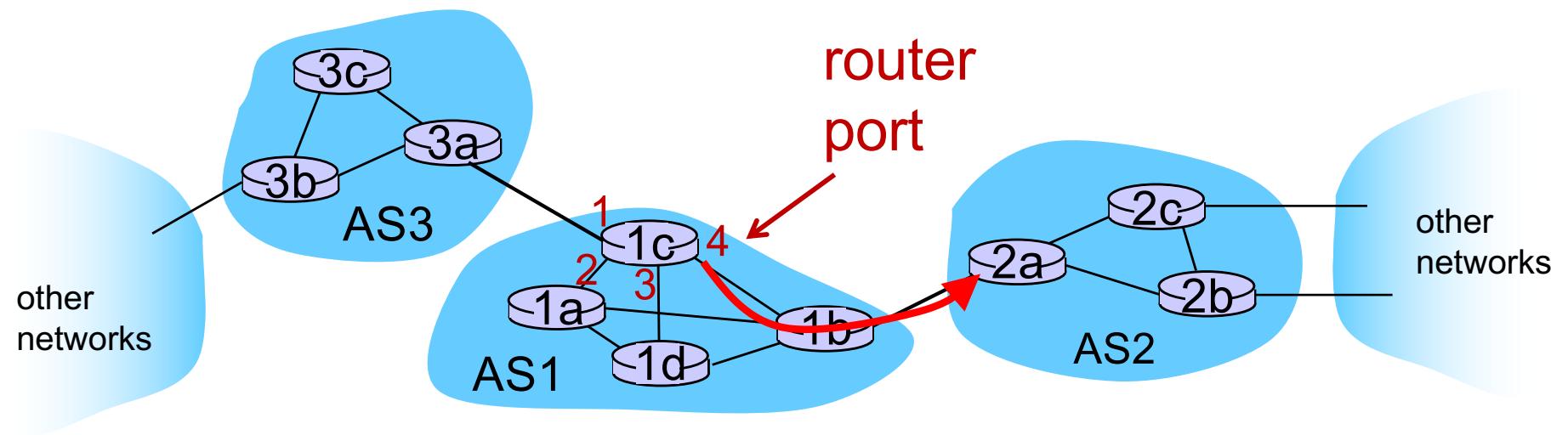
- Wenn ein Router mehrere Routen für einen Prefix erhält, dann wählt er eine aus.
- Beispiel:
 - AS2 ,AS17 to 138.16.64/22 ← Wählle diese Route
 - AS3, AS13, AS201 to 138.16.64/22

Finde die beste Intra-AS Route zur gewählten BGP Route



- Verwende das NEXT-HOP Attribute als Ziel für die Berechnung der Intra-AS Route
 - Das NEXT-HOP Attribut ist die IP Adresse des –router Interfaces, bei dem der AS Path beginnt.
- Beispiel
 - AS-PATH: AS2 AS17 ; NEXT-HOP: 111.99.86.55
 - Der Router verwendet OSPF um den kürzesten Pfad von Router 1c zu 111.99.86.55 zu finden.

Trage das Port der Route in der Forwarding Tabelle ein



- Trage das Paar prefix-port in die Forwarding Tabelle von 1c ein.
- Beispiel
 - (138.16.64/22 , port 4)

Warum gibt es unterschiedliche Intra- und Inter-AS-Routing-Protokolle?

Steuerung:

- Inter-AS: Gezielte Steuerung des Verkehrs nötig (Kosten, Sicherheit, Verfügbarkeit, Politik, ...)
- Intra-AS: Einheitlicher Administrationsbereich, keine Notwendigkeit der Abgrenzung

Skalierung:

- Eine Aufteilung in überschaubare Bereiche ist wichtig für die Anwendbarkeit der Routingalgorithmen (→ Hierarchisches Routing)

Leistung:

- Intra-AS: Leistungsoptimierung i.d.R. oberstes Ziel
- Inter-AS: Steuerungsaspekte teilweise wichtiger als Leistung

Kapitel 5: Netzwerkschicht & Routing

Gliederung

- Einleitung und Netzwerkdienstmodelle
- Aufbau eines Routers
- Das Internet-Protokoll (IPv4)
- NAT vs. IPv6
- Paketfilterung (Firewalls)
- Routing-Algorithmen
- Routing-Protokolle im Internet
- MPLS 
- Zusammenfassung

Textbuch zu diesem Kapitel: J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz, Kapitel 4

Folien und Abbildung teilweise aus:

J. Kurose & K. Ross: Computernetzwerke – Der Top-Down-Ansatz

MPLS: Multiprotocol Label Switching [RFC 3031]

MPLS

- ist eigenständige Netzwerktechnologie, die zwischen Schicht 2 und 3 angesiedelt ist.
- kann IP Pakete transportieren
- wird i.a. innerhalb eines autonomen Teilnetzes (Intra-AS) eines ISP angewendet
- ist verbindungsorientiert
 - verwendet die Idee der virtuellen Kanäle (VC) für Datagrammnetzwerke:
Eingehende Pakete erhalten eine Verbindungs-ID ("**Label**" genannt) für die Paket-Weiterleitung vom Eingangs- bis zum Ausgangsrouter
 - setzt eine vorherige Konfiguration aller MPLS-fähigen Router eines AS voraus (statisch oder per Protokoll wie z.B. RSVP-TE [RFC 3209])
- kann als virtuelle Verbindung ("logical link") zwischen zwei IP-Routern eingesetzt werden

Warum ist MPLS bei ISPs sehr erfolgreich und beliebt?

Mehr Kontrolle über den Netzwerkverkehr

- MPLS Traffic Engineering wird in RFC 3346 diskutiert
- Dienstgüte (Quality of Service) kann durch VC garantiert werden (z.B. für firmeninterne Videokonferenzen)
- Mehrere alternative Routen gleichzeitig verwendbar

Realisierung von Virtual Private Networks

- Verbinden von Kunden-IP-Netzwerken über ein MPLS-Netz eines ISP
- Das MPLS-Netz verhält sich so, als wäre es ein eigenständiges, privates, vom Internet unabhängiges Netzwerk

Flexibilität

- Es können im MPLS-Netz beliebige Protokolle verwendet werden
- Ressourcen können gleichmäßig ausgelastet werden

Bessere Performance

- Vereinfachte Weiterleitung (z.B. kein Longest-Prefix-Matching)

Zusammenfassung

