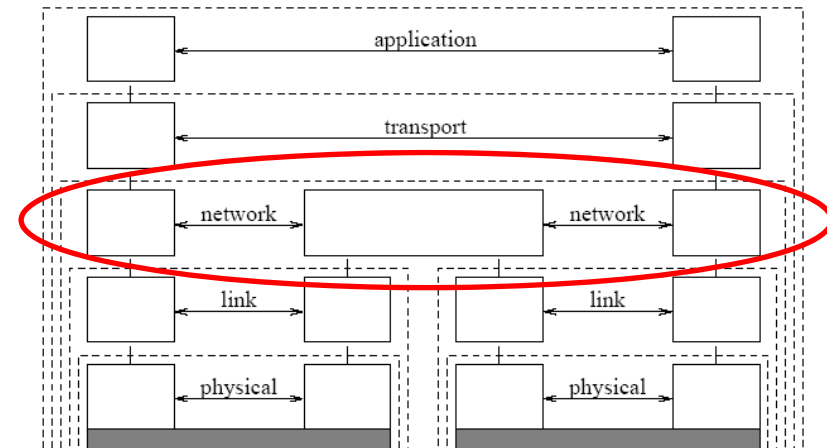


Die Vermittlungsschicht

Gliederung

- Aufgaben
- Virtual Circuit und Datagramm
- Routing
- Internet-Protokoll IP V4, IP V6
- Routing im Internet
- Multicast und Broadcast



Vermittlungsschicht (Network Layer)

Aufgabe der Vermittlungsschicht

- Nachrichtenweiterleitung

basierend darauf, dass mögliche Pfade des Netzes ermittelt wurden und in Weiterleitungstabellen hinterlegt sind.

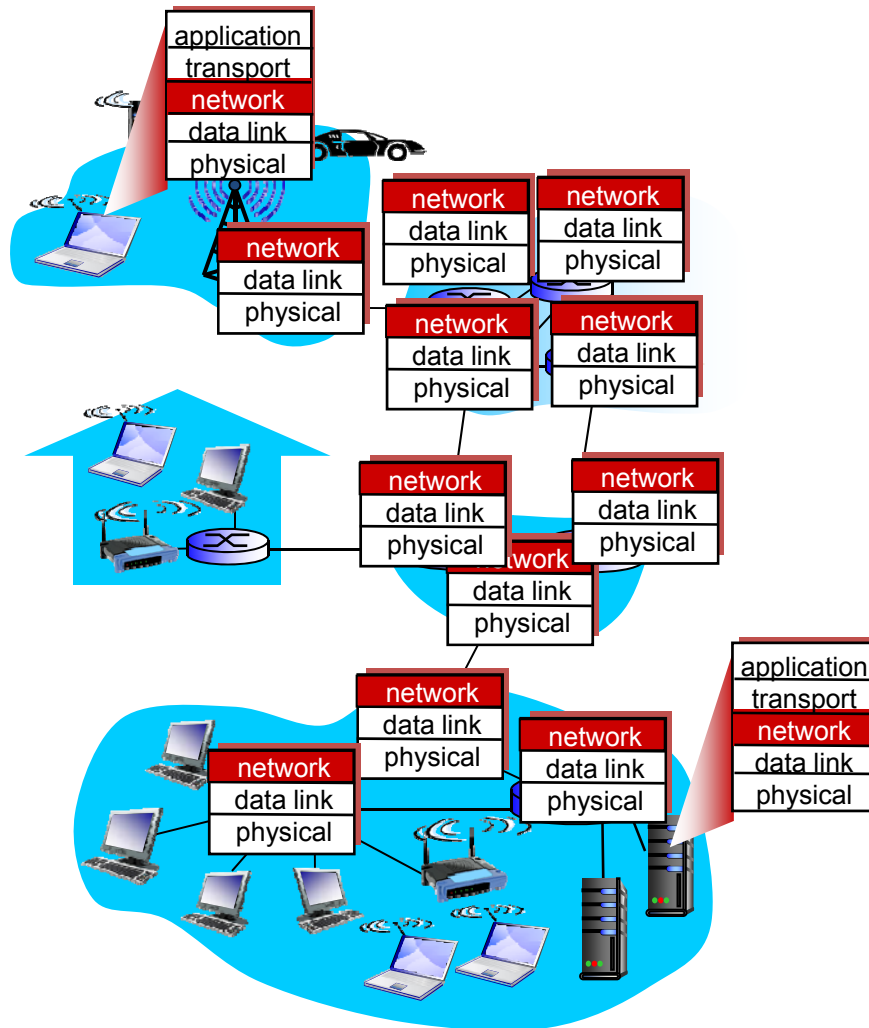


Aufgabe der Anwendung „Routingtabellen-Pflege“

- Netzerkundung
- Pfadermittlung
- Aktualisierung bei Ausfällen, Reparaturen, Überlastsituationen



Vermittlungsschicht (Network Layer)



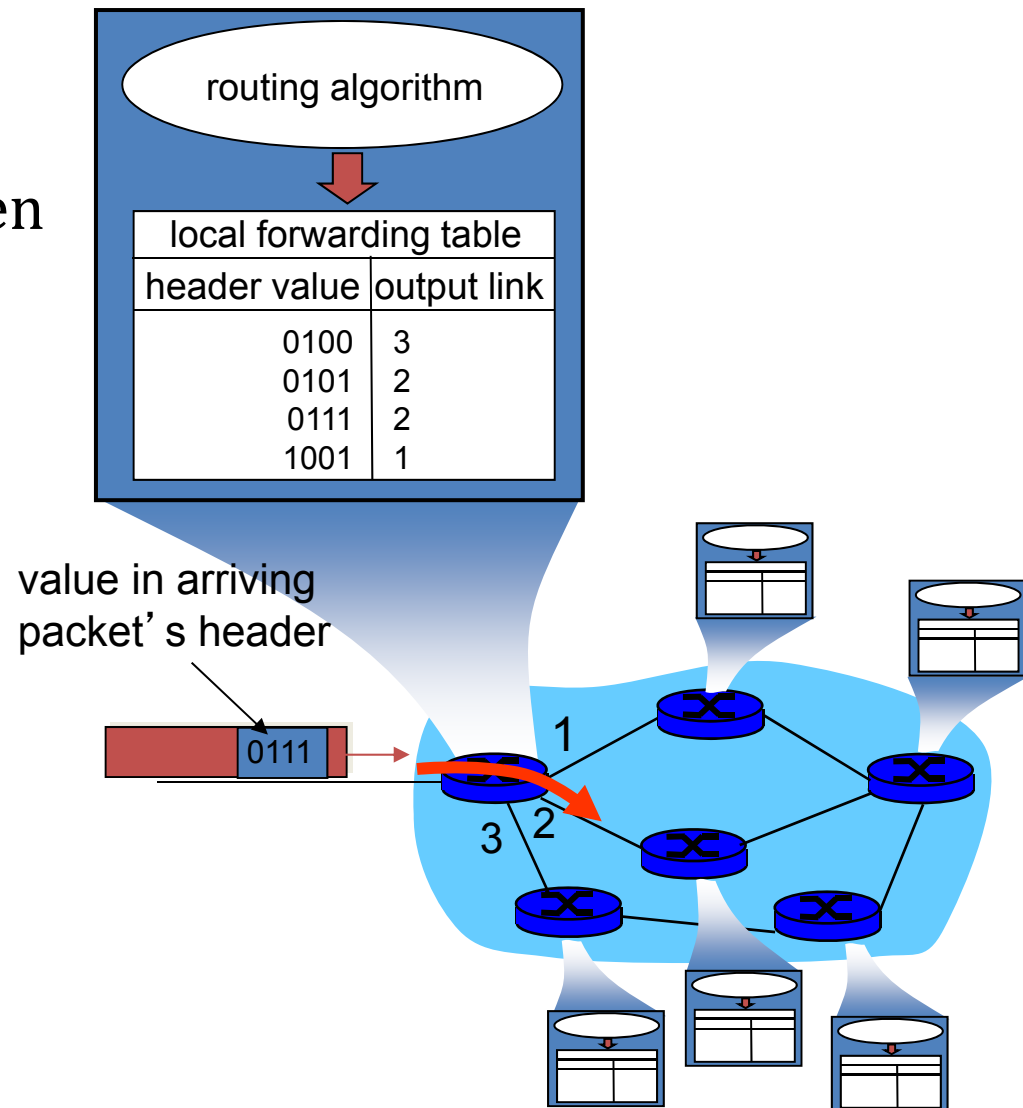
Bilde ein Netz aus Transitknoten und Teilstrecken, so dass Nachrichten zwischen beliebigen Punkten ausgetauscht werden können

Modelle / Paradigmen:

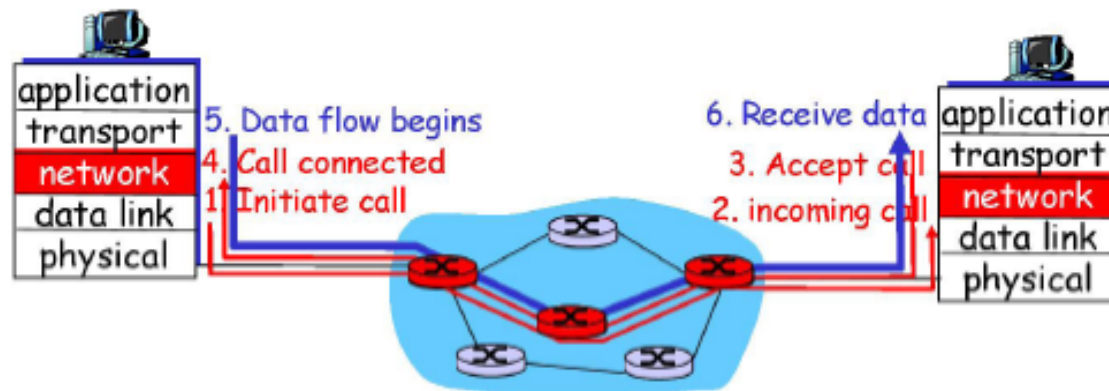
- Virtueller Kanal
Verbindung, z.B. ATM
- Datagramm
einzelne Pakete, Store and Forward, z.B. IP

Routing – Weiterleitung

- *Weiterleitung*: Weitergabe eines Pakets vom Eingang eines Routers zum richtigen Ausgang (nächster Schritt auf dem Weg durch das Netz)
- *Routing*: Bestimmung der Route vom Sender zum Empfänger (Planung des Pfades von der Quelle zum Ziel, Routing-Algorithmus)

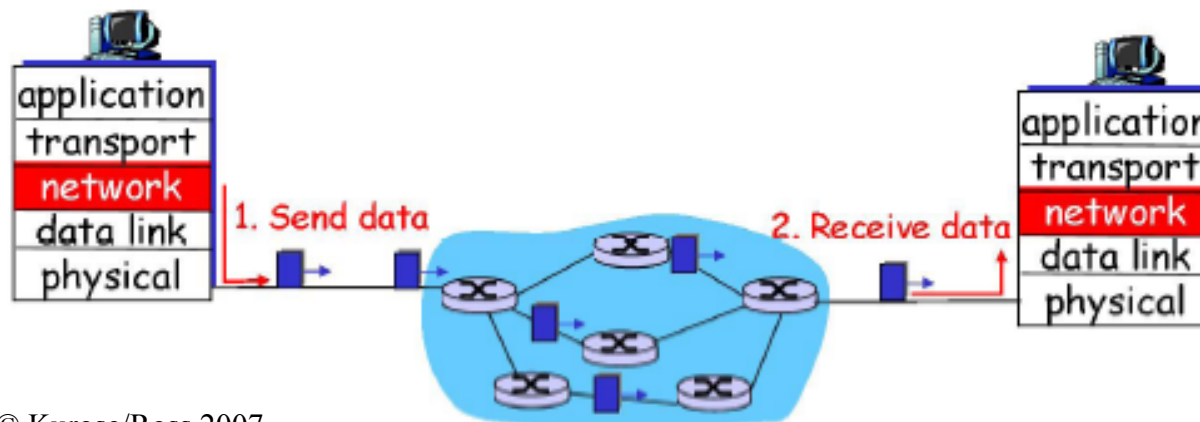


Virtueller Kanal (ATM) versus Datagramm-Dienst



2-Wege-Handshake bei Host:
Signalisierungsnachrichten

1. Verbindung einleiten
2. Ankommende Verbindung
3. Verbindung annehmen (Zielrechner)
4. Verbindung aufgebaut



Nur Pakete senden und empfangen

© Kurose/Ross 2007

© Peter Buchholz 2016 (nach
Kurose/Ross 2003-2013)

RvS Kap. 4 Die Vermittlungsschicht

Virtueller Kanal (ATM) - Vorteile

Virtueller Kanal kann genutzt werden, um vorab für eine Menge von Pakettransfers gemeinsame Ressourcen zu reservieren und Konfigurationseinstellungen so durchzuführen, dass

- Echtzeitgarantien
- Mindestdurchsatz
- Verzögerungszeit-Grenzen
- begrenzte Variation der Verzögerungszeit (Jitter)

leichter realisierbar werden.

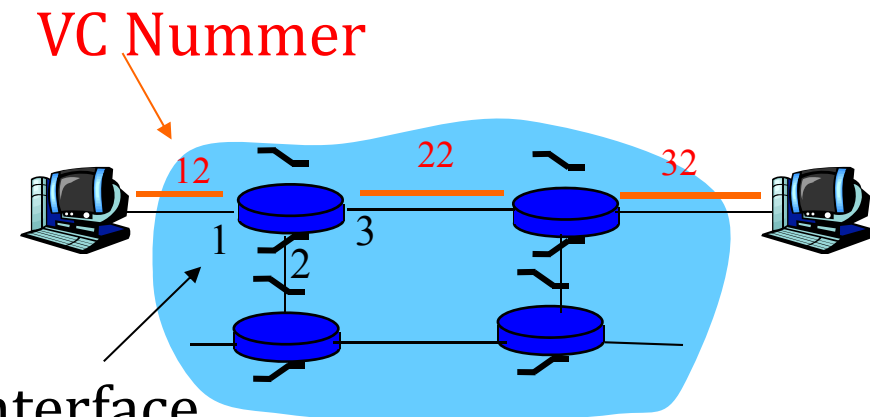
ATM – Dienstklassen versus IP/Best Effort

Architektur	Dienstmodell	Bandbreite	verlustfrei	Reihenfolge	Zeit	Überlasthinweis
Internet	best-effort	keine Zusicherung	nein	beliebig	wird nicht eingehalten	nein
ATM	CBR	konstante Rate	ja	wird eingehalten	wird eingehalten	keine Überlast
ATM	VBR	garantierte Rate	ja	wird eingehalten	wird eingehalten	keine Überlast
ATM	ABR	garantiertes Minimum	nein	wird eingehalten	wird nicht eingehalten	ja
ATM	UBR	keine Zusicherung	nein	wird eingehalten	wird nicht eingehalten	nein

Ablauf Virtual Circuit

Pakete beinhalten VC-Nummern
und nicht die Zieladresse!

Weiterleitungstabelle im Router:



© Kurose/Ross 2009

Eingangsinterface	Eingangs-VC	Ausgangsinterface	Ausgangs-VC
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Router verwalten die Informationen!

Datagramm Dienste

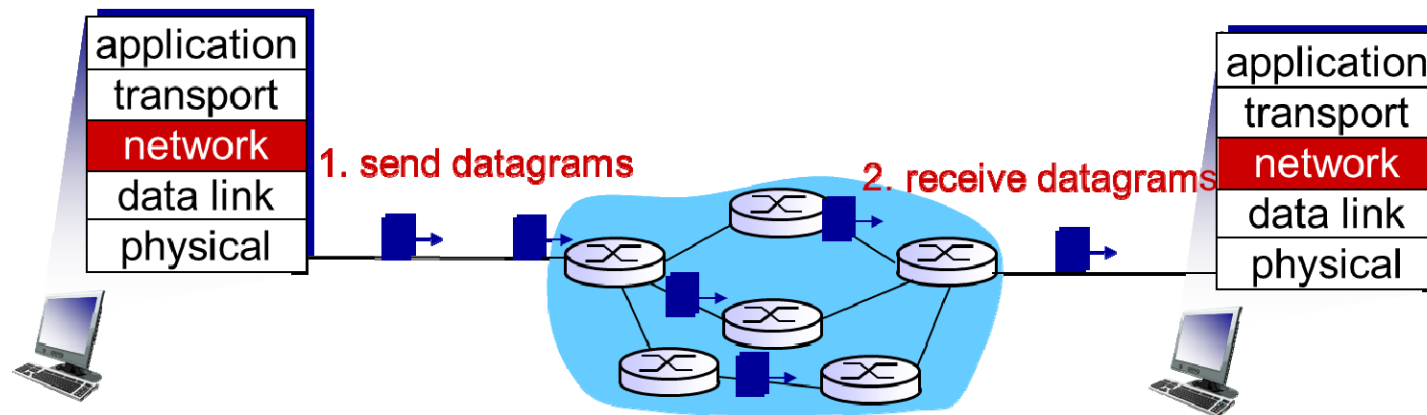
Internet als Datagramm-Netz:

- Jedes Paket muss seinen Weg durch das Netz finden
 - Jedes Paket muss damit eine Zieladresse beinhalten
 - Pakete mit identischer Quell- und Zieladresse können u.U. unterschiedliche Wege wählen
- Jeder Router auf dem Weg zum Ziel entscheidet, wohin das Paket weitergeleitet wird
 - Auf Basis lokaler Informationen über die Netzstruktur und den Netzzustand

✓ Routing Algorithmus

Hilfreiche Abstraktion: Netz als gewichteter Graph!

Datagramm Netze



longest prefix matching

Zieladressbereich	Ausgang
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
sonst	3

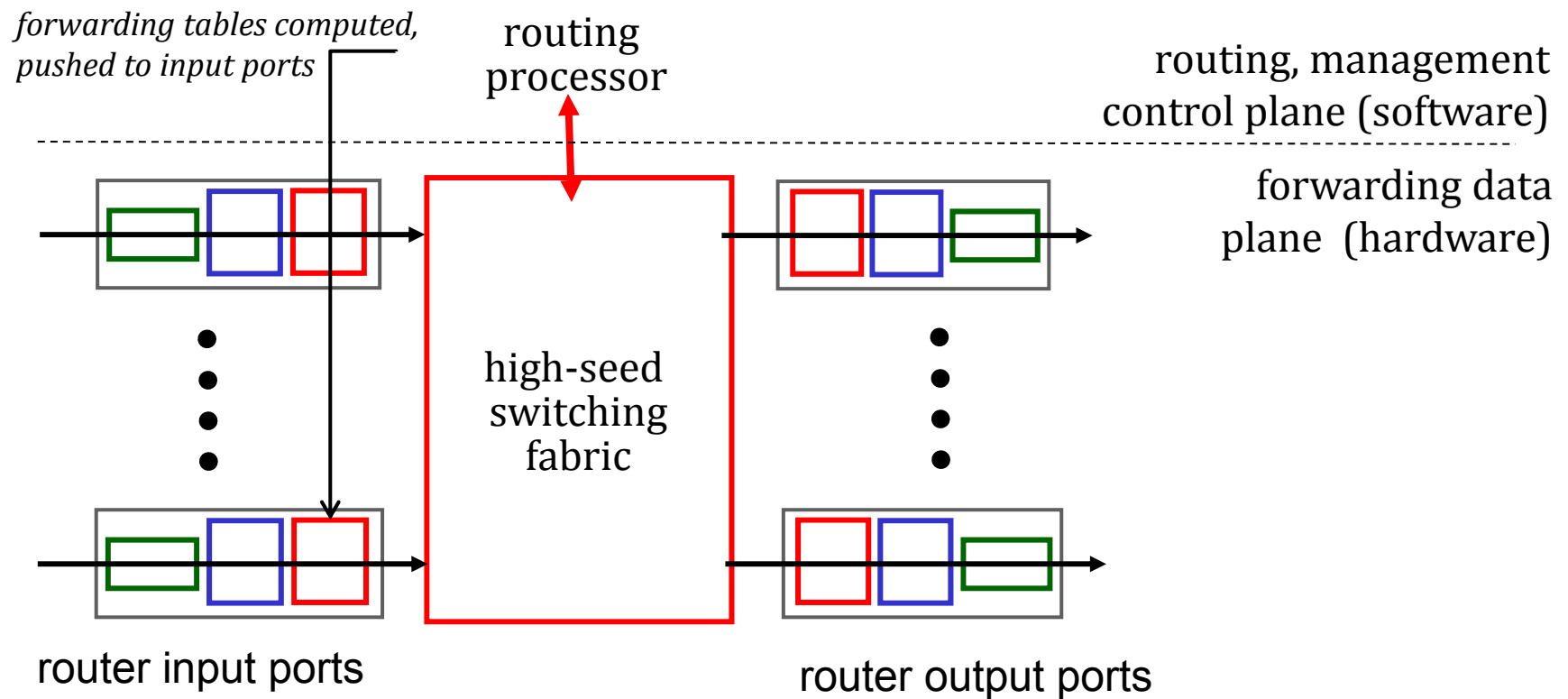
Zieladr.: 11001000 00010111 00010110 10100001

Zieladr.: 11001000 00010111 00011000 10101010

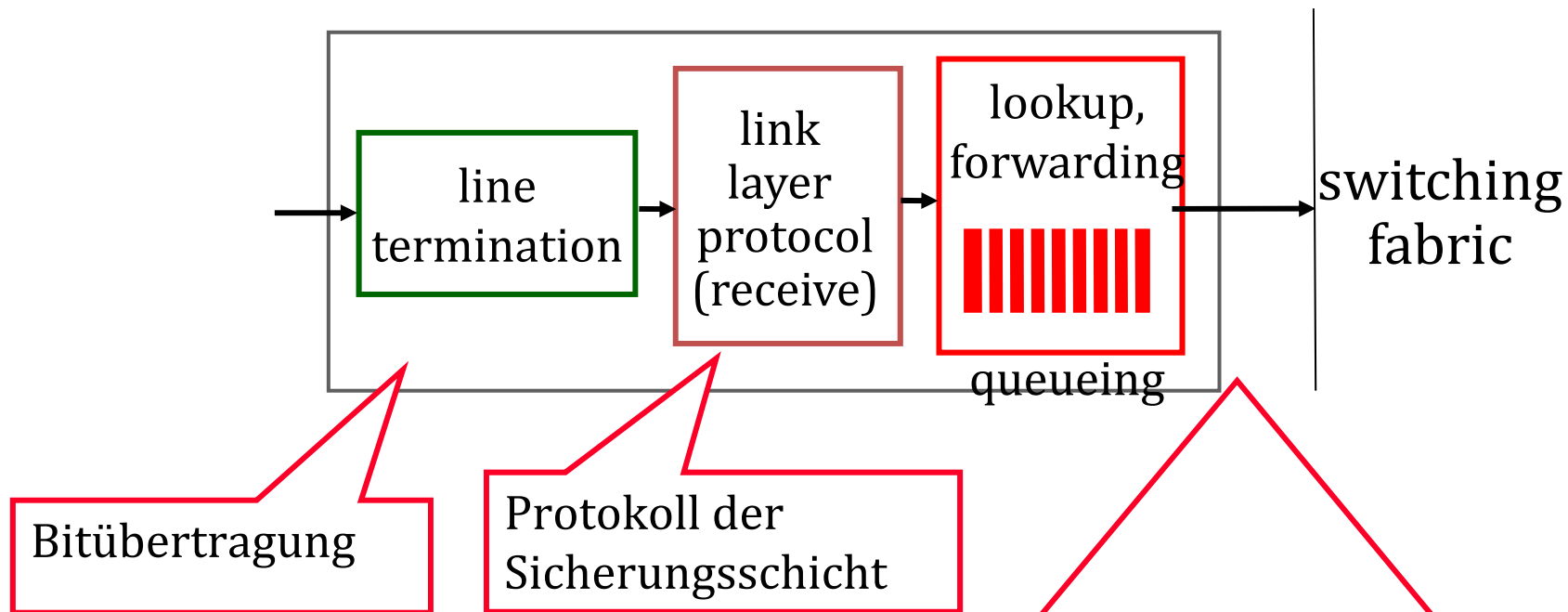
Welcher Ausgang?

Interner Aufbau eines Routers

- Zwei Hauptfunktionen eines Routers im Internet:
 - Vermittle IP-Pakete: IP-Weiterleitung
 - Pflege Routingtabellen per RIP/OSPF/BGP (Beschreibung später)



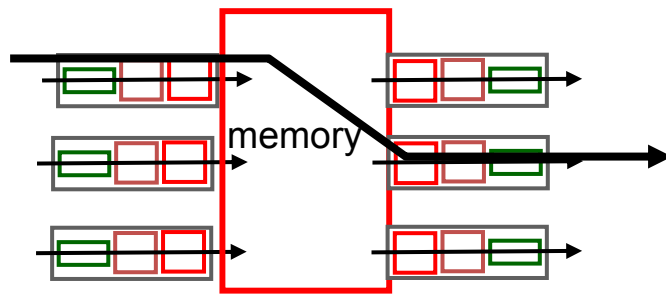
Funktionen des Eingangs-Ports



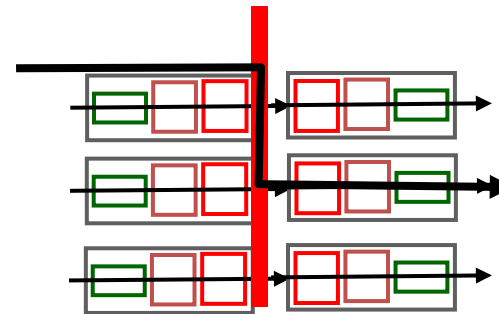
Dezentralisiertes Switching :

- Auf Basis der Zieladresse wird der zugehörige Ausgang aus der Routingtabelle gelesen
- Ziel: Bearbeitung der eingehenden Pakete entsprechend der Geschwindigkeit der Eingangsleitung
- Zwischenspeicherung (queueing): Wenn Pakete schneller ankommen, als sie den Router verlassen

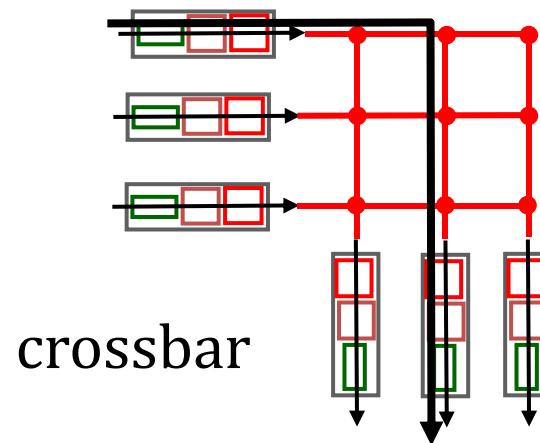
Switching Fabric – Vermittlung: 3 Typen



memory

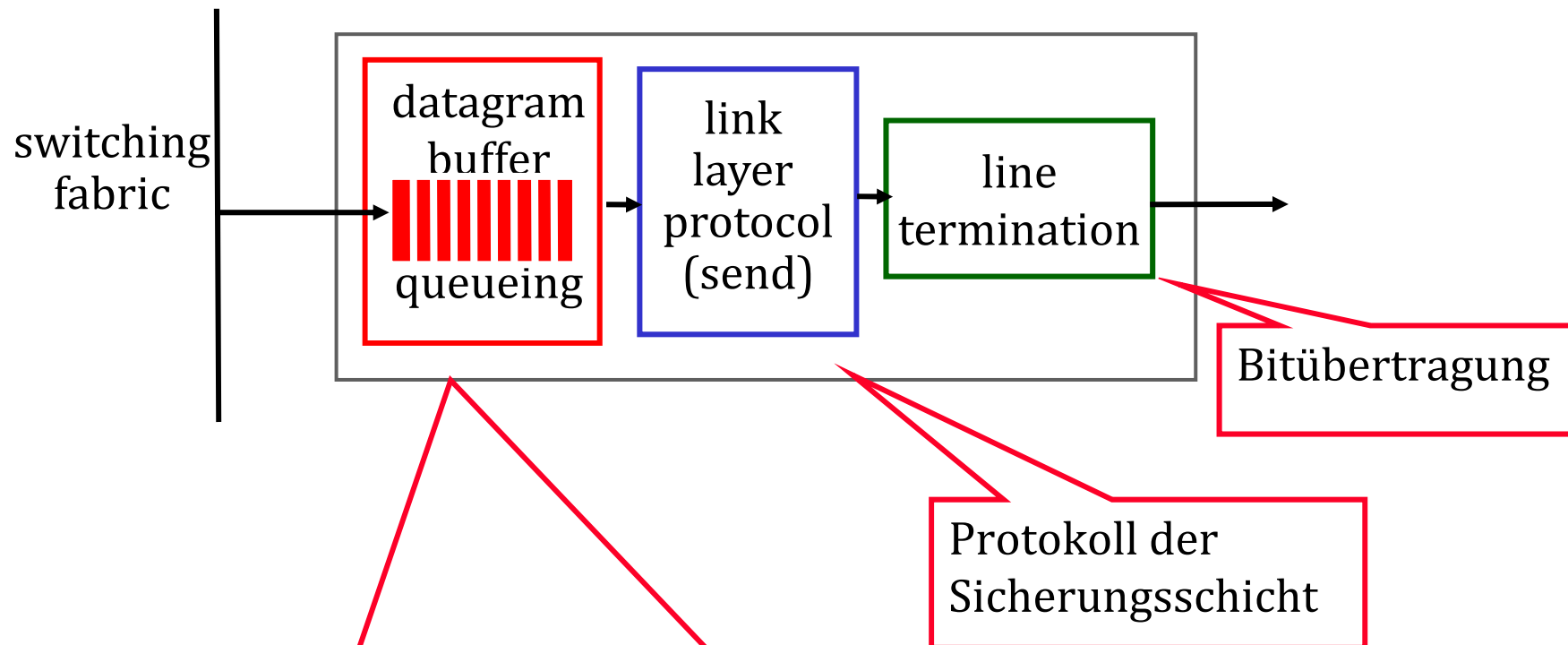


bus



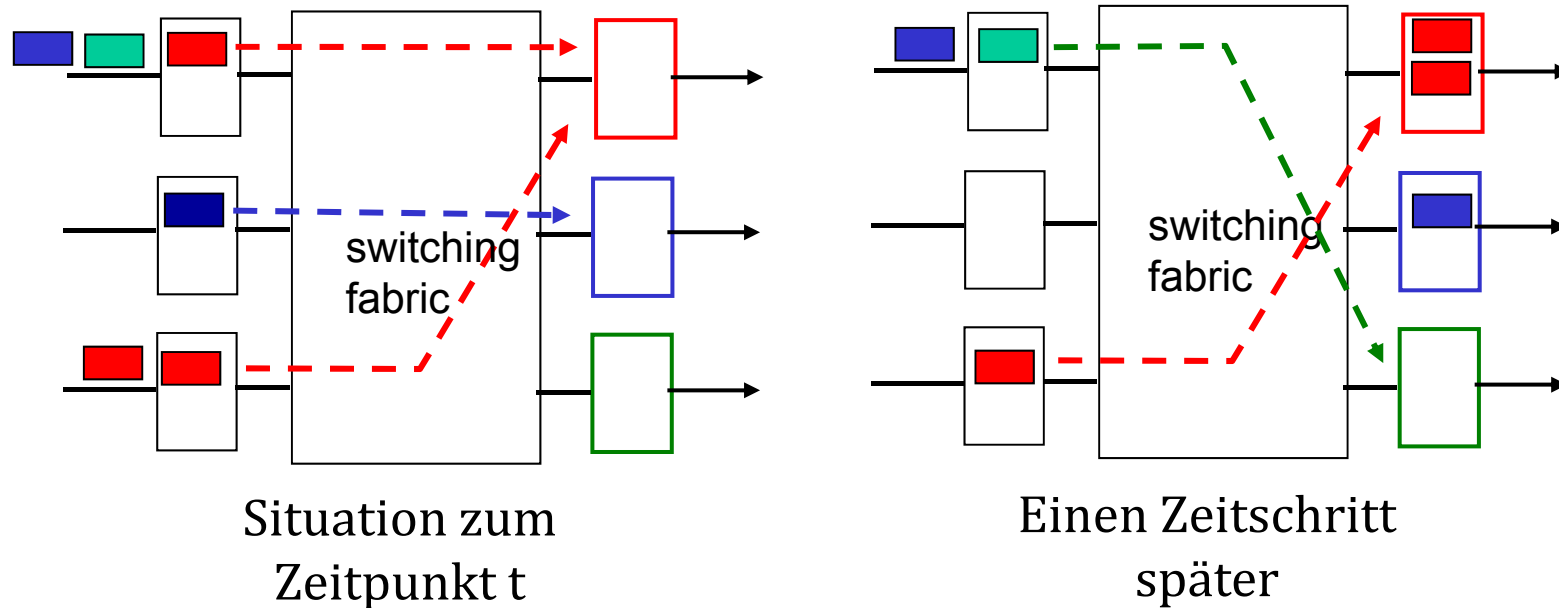
crossbar

Funktionen des Ausgangs-Ports



- Pufferung von Datagrammen, wenn diese schneller ankommen, als sie übertragen werden können
 - Scheduling zur Auswahl von zu übertragenden Paketen ist notwendig

Switching Fabric – Vermittlung: Blockierung

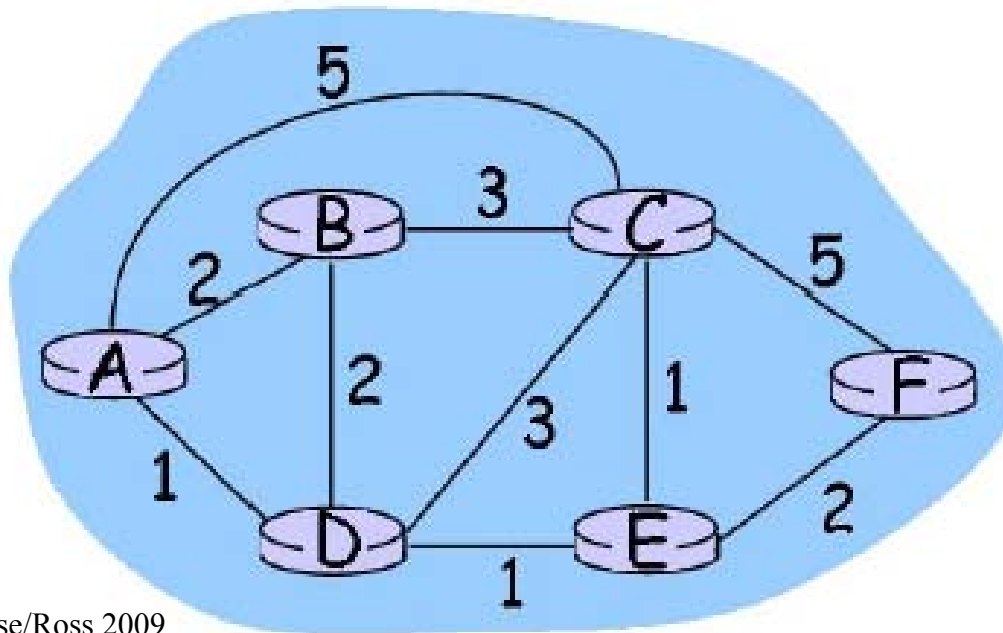


Wartezeiten entstehen am Ausgang, wenn zu viele Pakete für einen Ausgang generiert werden!

u.U. sogar Pufferüberläufe und Paketverluste

Routing-Algorithmen

- ◆ Pfadermittlung im Netz:
Vom *First Hop Router* auf Sendeseite
(*Default Router* des Senders)
zum *Destination Router* des Empfängers
- ◆ Graphentheorie: „Kürzeste Wege“-Algorithmen



Von A nach F gibt es hier 17
Pfade ohne Zyklen. Der
kürzeste Pfad ist ADEF

Routing-Algorithmen

- ◆ **Global**

je Kante Kantenverlauf und Kosten
global bekannt

Problem: Skalierbarkeit, Änderungsaufwand

- ◆ **Dezentral**

Jeder Router kennt nur die Kanten zu seinen Nachbarn

- ◆ **Statisch**

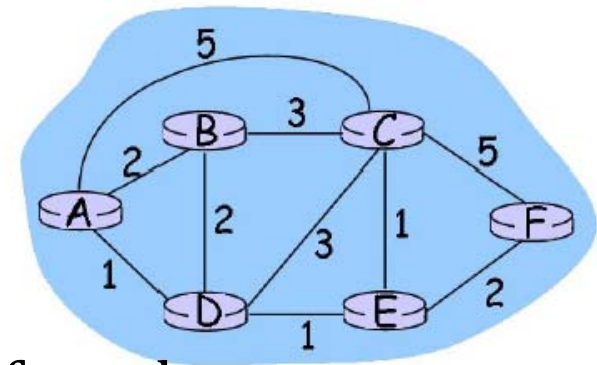
Kantenverlauf und Kosten ändern sich nicht (bzw. kaum)

- ◆ **Dynamisch**

Kanten verschwinden, kommen dazu; Kosten ändern sich

Problem: Schleifenbildung durch dynamische Suche

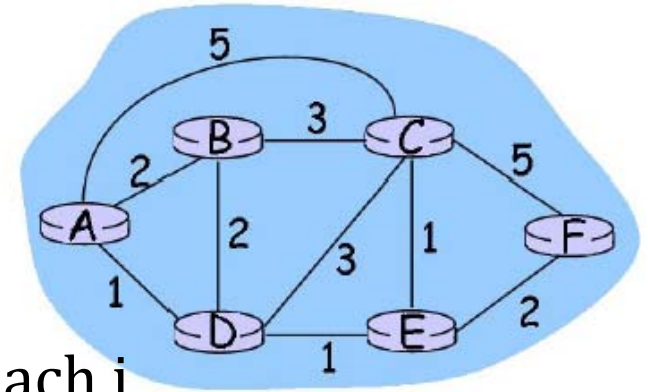
- ◆ Im Internet: **Link State** (dynamisch global) –
Distance Vector (dynamisch dezentral)



Link-State: Dijkstra-Algorithmus

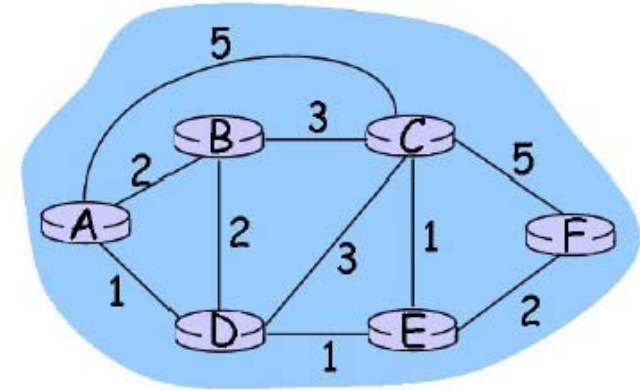
Gegeben:

- K Knotenmenge des Graphen
- $c(i,j)$ Verbindungskosten der Kante von i nach j
wobei $c(i,j) = \begin{cases} \text{Kosten der Kante, falls eine solche existiert} \\ \infty, \text{ falls keine Kante existiert} \end{cases}$
- $D(v)$ Kosten des bisher besten Pfades vom Sender zum derzeitigen Ziel v
(hat unter allen untersuchten Pfaden die geringsten Kosten)
- N Menge der Knoten, bei denen ein Pfad mit geringsten Kosten vom Sender bekannt ist



Link-State: Dijkstra-Algorithmus

```
1  initialisation:
2  N = {v0}
3  for all nodes v
4      if v adjacent to v0
5          then D(v) = c(v0, v)
6          else D(v) = ∞
7
8  loop
9      find w not in N such that D(w) is a minimum
10     add w to N
11     for all v adjacent to w and not in N
12         /* update D(v) */
13         D(v) = min( D(v), D(w)+c(w,v) )
14  until all nodes are in N
```



Link-State: Dijkstra-Algorithmus

Satz:

Der Algorithmus berechnet einen Weg mit minimalen Kosten von einem Sender v_0 zu jedem potentiellen Empfänger v in $O(n^2)$ Schritten ($n = |K|$ die Anzahl der Knoten im Graph).

Algorithmus ist Inhalt von DAP 2 und soll deshalb hier nicht weiter vertieft werden!

Distance Vector: Dezentrales Routing

Basis: Bellman-Ford-Gleichungen (BF)
(dynamischen Programmierung)

Seien

$d_x(y) :=$ Kosten des günstigsten Pfades von x nach y

Dann gilt

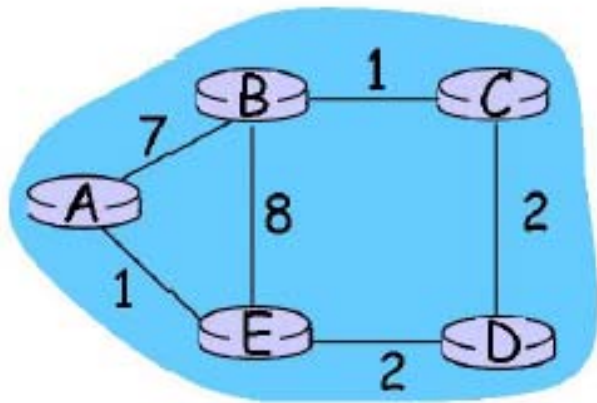
$$d_x(y) = \min \{c(x,v) + d_v(y)\}$$

wobei das Minimum über alle Nachbarn v von x bestimmt wird

Dezentraler Ansatz, jeder Knoten berücksichtigt nur lokale Informationen

Distance Vector: Dezentrales Routing

Idee: Jeder Knoten schickt ein Paket über den direkten Nachbarn, über den die geringsten Kosten entstehen
Solange sich die Kosten nicht ändern kann dazu eine Distanztabelle verwendet werden



Kosten zum Ziel über

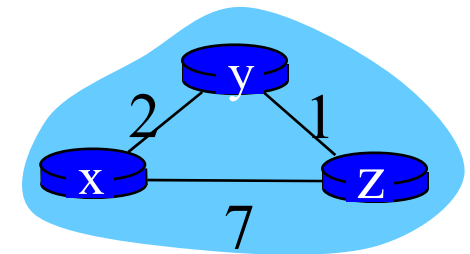
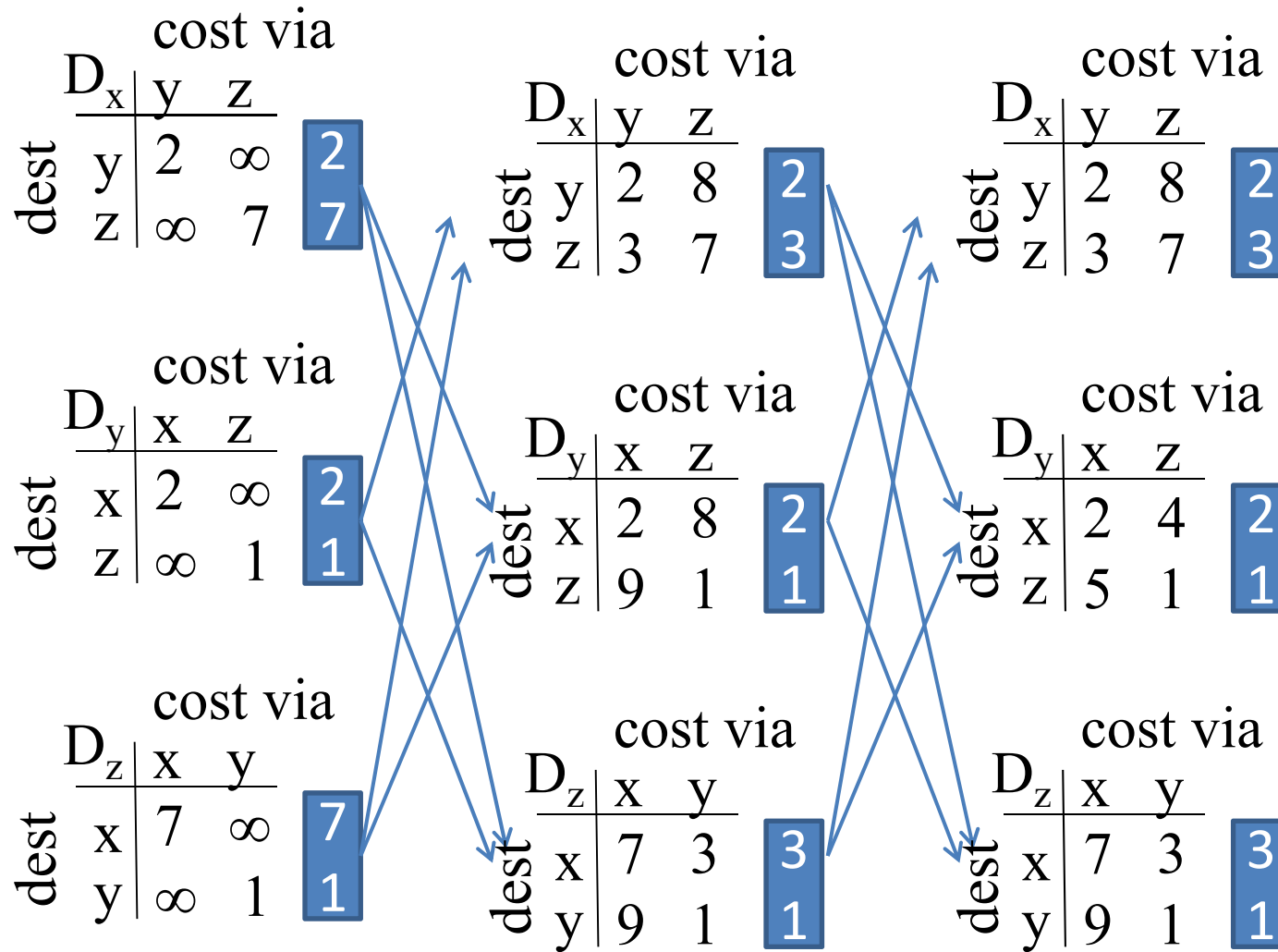
$d_E()$	A	B	D
A	①	14	5
B	8	8	⑤
C	6	9	④
D	5	11	②

Distance Vector: Algorithmus

1. Initialisierung
2. Für alle Matrixeinträge:
3. $D_x(y,z) = \infty$;
4. Für alle Nachbarknoten v:
5. $D_x(v,v) = c(x,v)$;
6. Sende $\min_w D_x(y,w)$ an alle Nachbarn ;
7. Iteriere
8. warte bis sich Leitungskosten $c(x,v)$ ändern oder neue Informationen von einem Nachbarn eintreffen
9. falls sich $c(x,v)$ um d ändert
10. $D_x(y,v) = D_x(y,v) + d$;
11. falls Nachbar v einen neuen Wert newval für seinen Weg zu y schickt
12. $D_x(y,v) = \text{newval} + c(x,v)$;
13. sende neue Werte $\min_w D_x(y,w)$ an alle Nachbarn

Der Algorithmus kommt zum Stillstand, wenn keine Änderungen mehr eintreten. Netzveränderungen können zu Problemen führen: Algorithmus stoppt dann nicht zwangsläufig.

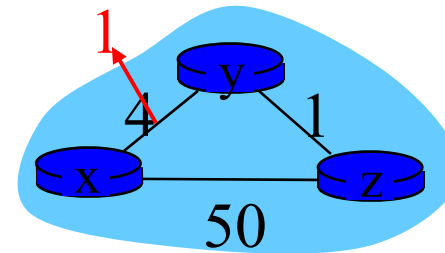
Distance Vector: Algorithmus - Beispiel



Distance Vector: Algorithmus - Beispiel

Veränderungen der Kosten:

- Knoten entdeckt lokale Änderung
- Routinginformation wird geändert und der Distanzvektor (DV) neu berechnet
- Wenn sich DV ändert, wird er an die Nachbarn verteilt



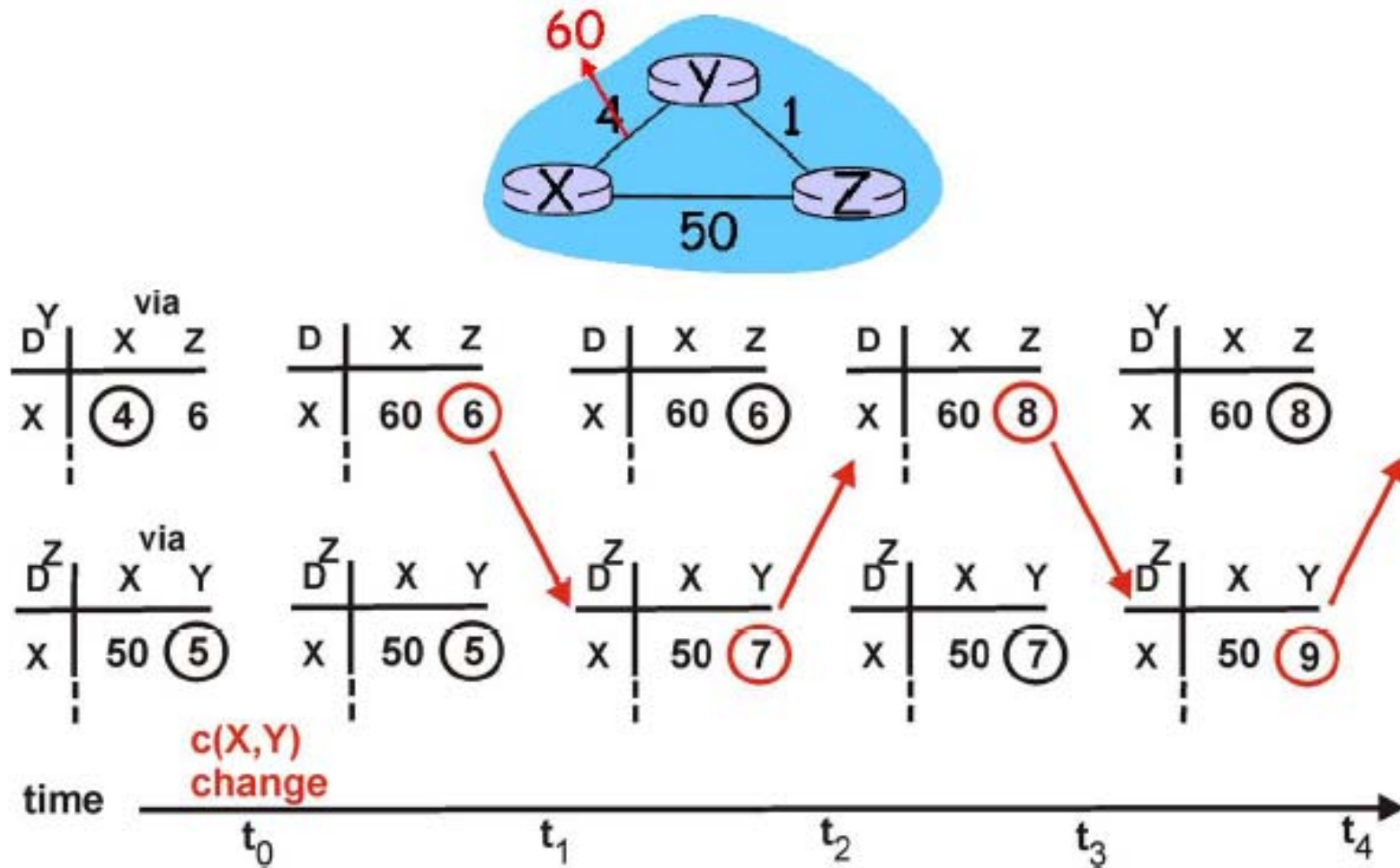
“Gute
Neuigkeiten
verbreiten
sich
schnell”

Zum Zeitpunkt t_0 , entdeckt y die Änderung, berechnet DV neu und informiert seine Nachbarn.

Zum Zeitpunkt t_1 , empfängt z die Änderung von y und aktualisiert seine Routing-Tabelle. Ein neuer kürzester Weg zu x wird berechnet und DV an die Nachbarn verschickt

Zum Zeitpunkt t_2 , empfängt y z 's Änderung und aktualisiert seine Routing-Tabelle. y 's kürzeste Wege ändern sich nicht, so dass keine Nachricht verschickt wird.

Distance Vector: Algorithmus - Beispiel



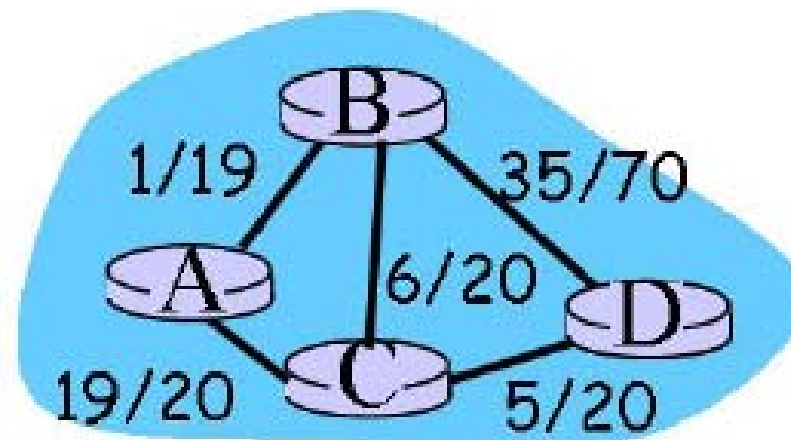
Schlechte Neuigkeiten verbreiten sich langsam.

Leitungsvermittlungsalgorithmen

Bei leitungsvermittelnden Algorithmen (verbindungsorientiert) sind z. B. Kapazitäten und benutzte Kanäle bekannt.

Algorithmen

- **Dijkstra** minimale Anzahl Hops (shortest path in terms of hops)
- **Least Loaded Path (LLP)** am wenigsten benutzte Kanäle
- **Maximum Free Circuit (MFC)** jeweils größte Anzahl freier Kanäle



Routing im Internet

Bisherige Betrachtung des Routing idealisiert sehr stark, da

- alle Router als identisch angenommen wurden
- das Netz als flacher Graph betrachtet wurde

... stimmt nicht in der Realität

Dimension: mehr als 600
Millionen Ziele

- ❖ können nicht alle in Routingtabelle gespeichert werden!
- ❖ schon der Nachrichtenaustausch zum Routing würde das Netz überlasten

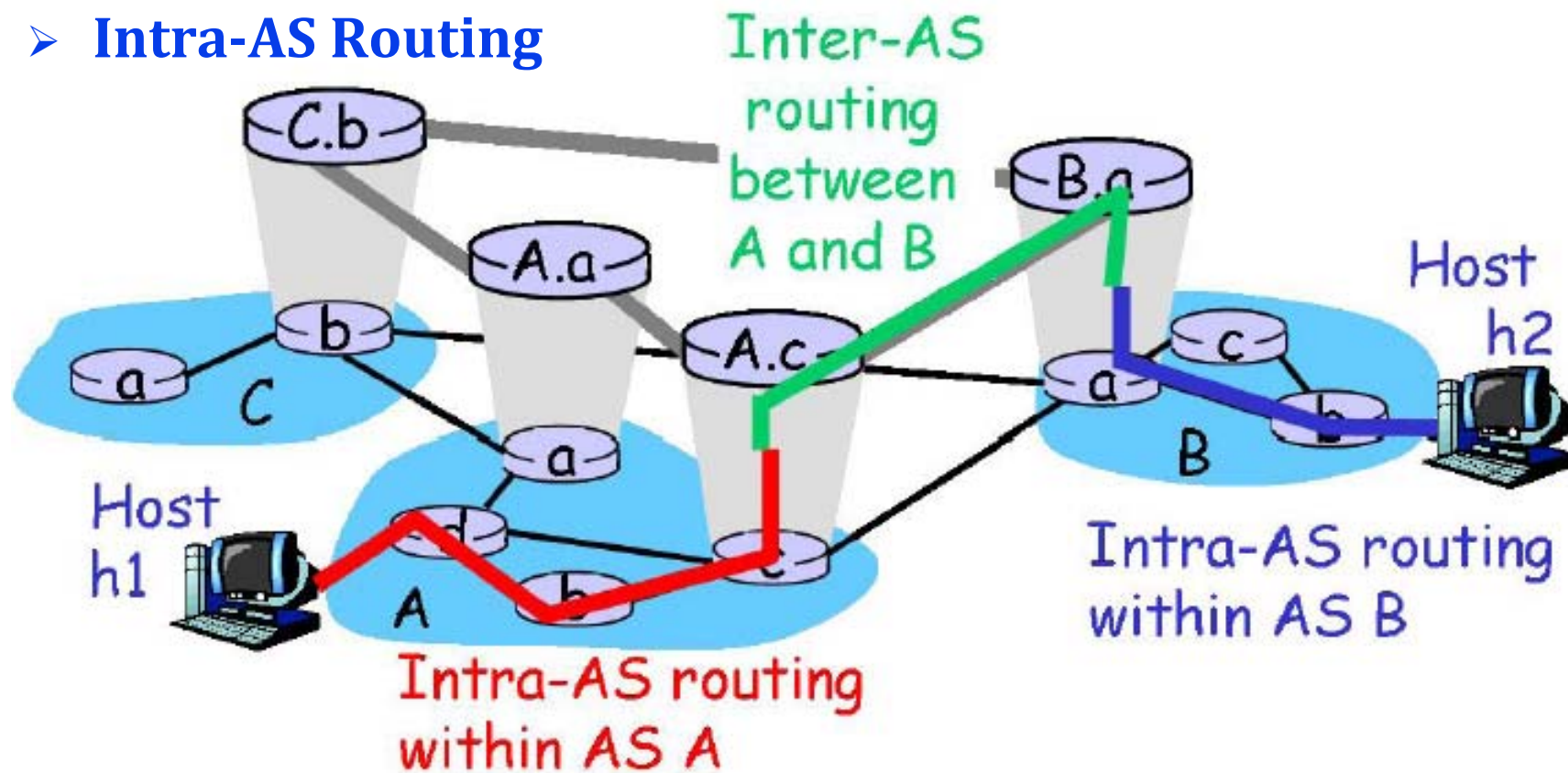
Autonomie:

- ❖ Internet = Netz von Netzen
- ❖ jeder Administrator möchte das Routing in seinem Netz kontrollieren

Hierarchisches Routing

Autonome Teilsysteme (AS)

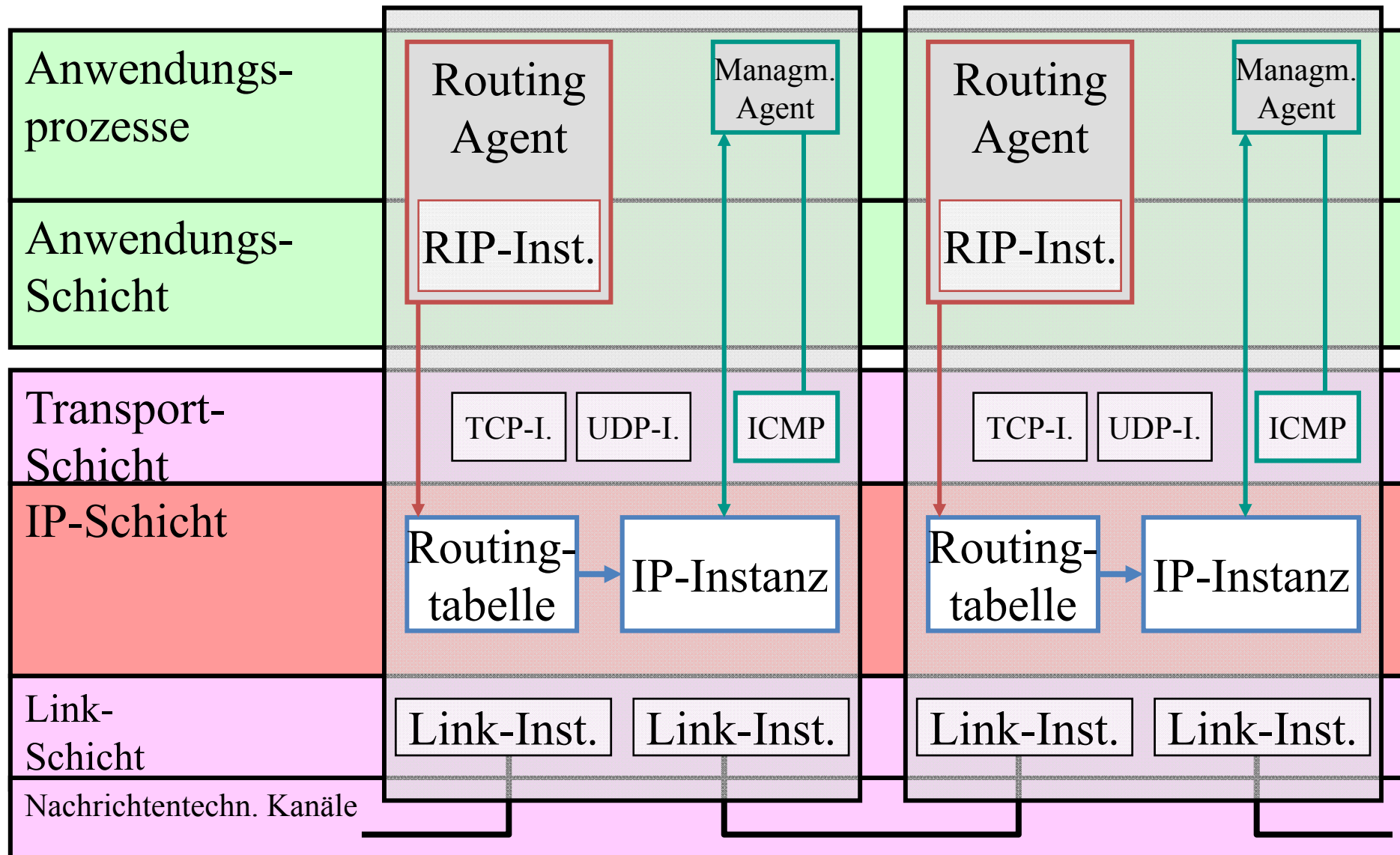
- Inter-AS Routing
- Intra-AS Routing



Internet-Protokoll: IP V4

- Verbindungsloser Datagramm-Dienst
- Nachrichten werden im Store-and-Forward-Prinzip von der Quelle zum Ziel weitergeleitet
 - vgl.: Brief oder Postpaket-Transport
- Nachrichten (Nutzdaten, d.h. die PDUs der Transportprotokolle) werden in IP-Pakete (so heißen die PDUs des IP-Protokolls) verpackt.
- Nachrichten werden u.U. segmentiert und in einer Serie von IP-Paketen hinterlegt.
- Jedes IP-Paket wird separat weitergeleitet.
- **Keine Reihenfolgentreue**
- **Keine Garantie maximaler Latenz**
- **Keine Verlustfreiheit**

Komponenten der IP-Schicht



IP v 4: Adressen

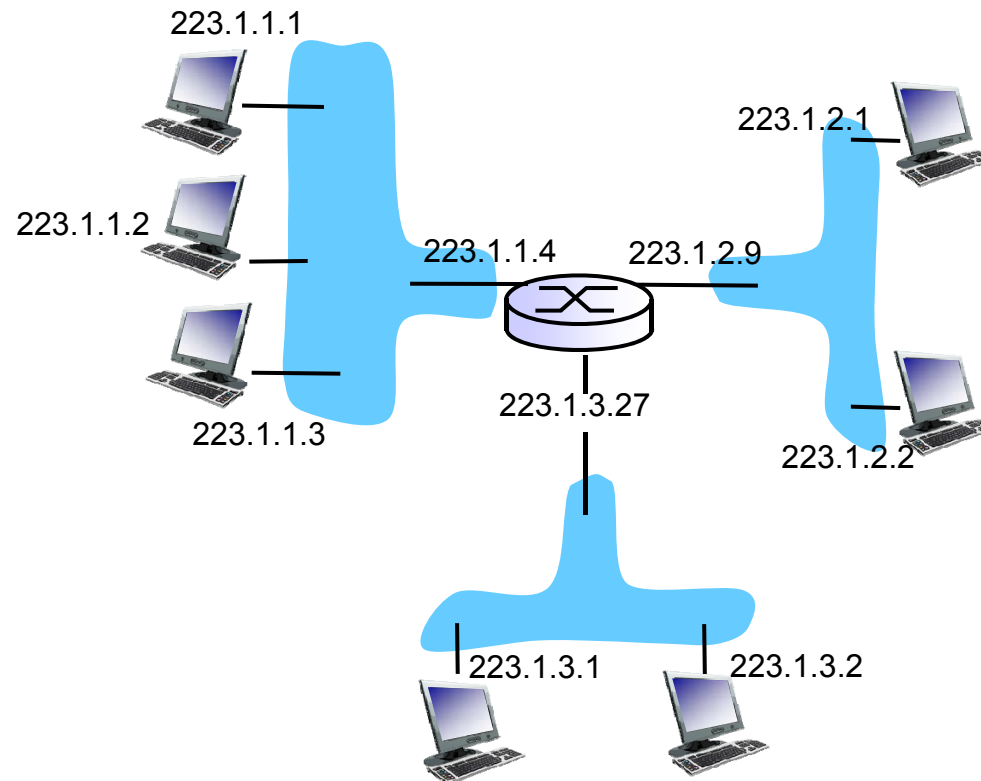
- ◆ 32-Bit Adressen, als 4 Byte-Gruppen
- ◆ 193.32.216.9 == 11000001 00100000 11011000 00001001
- ◆ Wenige große Netze mit sehr vielen Hosts: Class A
- ◆ „Viele“ kleine Netze mit höchstens 256 Hosts: Class C
- ◆ Multicast-Adressen: Vorbereitungsphase reserviert Adresse

Class	Netzwerk	Host	IP-Adresse
Class A	0		1.0.0.0 bis 127.255.255.255
Class B	01		128.0.0.0 bis 191.255.255.255
Class C	110		192.0.0.0 bis 223.255.255.255
Class D	1110		224.0.0.0 bis 239.255.255.255

32 Bit

IP: Adressen

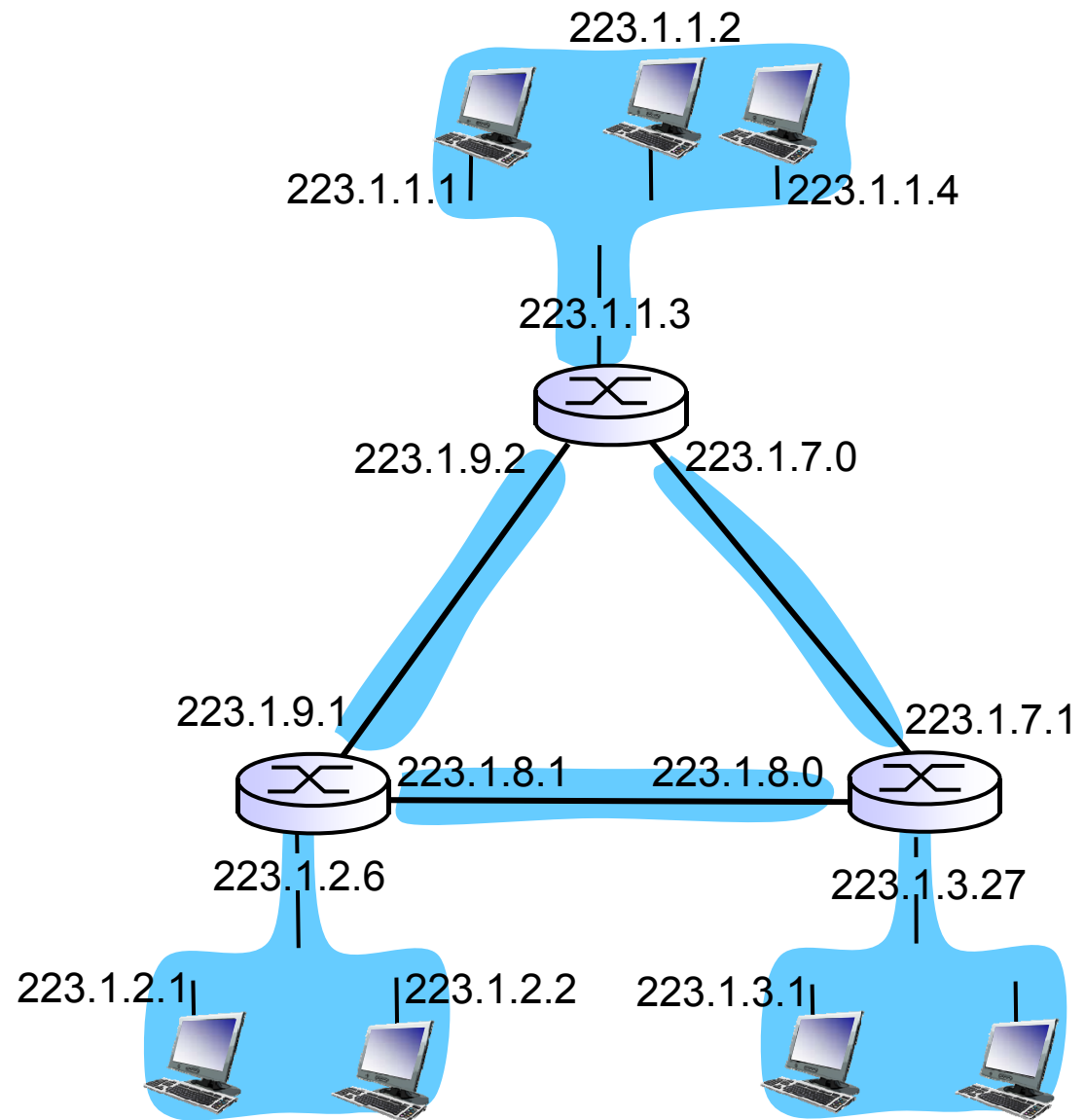
- Eine Adresse je Netz-Interface des Knotens



223.1.1.1 = $\underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_1$

IP: Adressen

- Bei mehreren Routern:
Verbindung von Schnittstellen zwischen Routern im selben Subnetz
- Im Beispiel:
6 Subnetze



IP: Interclass Domain Routing

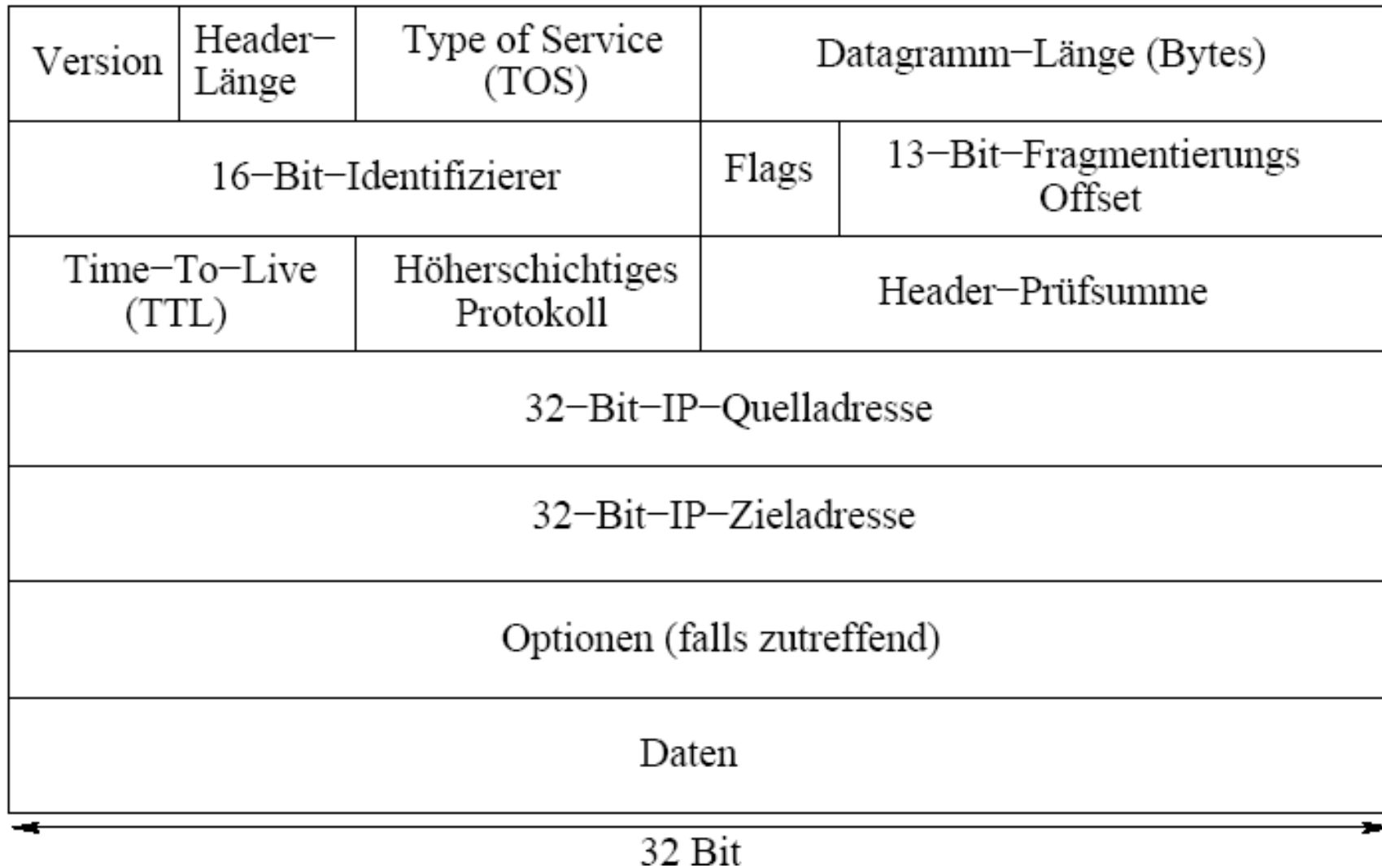
- beliebige Länge der Netzwerkadresse: z.B.: a.b.c.d/21

Adresszuweisung für Organisationen durch ISP:

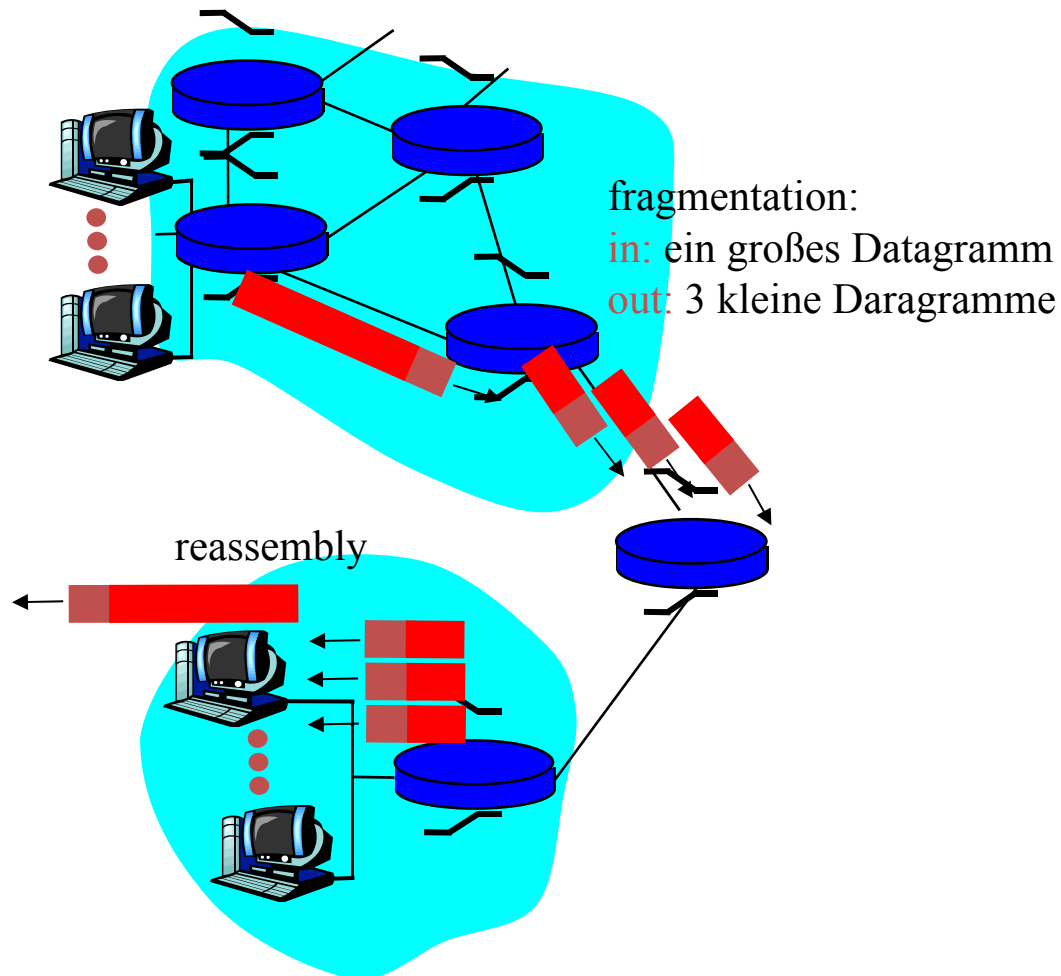
ISP-Block 200.23.16.0/20 (20 Bits für Netzwerkadresse) in 8 gleiche Teile:

- ! jede Organisation hat 23 Netzwerk-Bits:
- ISP-Block 11001000 00010111 00010000 00000000
200.23.16.0/20
- Organisation 0: 11001000 00010111 00010000 00000000
200.23.16.0/23
- Organisation 1: 11001000 00010111 00010010 00000000
200.23.18.0/23
- Organisation 2: 11001000 00010111 00010100 00000000
200.23.20.0/23
- ...
- Organisation 7: 11001000 00010111 00011110 00000000
200.23.30.0/23

IP: IP v4 - Paketformat



IP: Fragmentierung und Reassemblierung



Datagramm 4000 Bytes
(20 Bytes IP-Header
+3980 Daten)
wird in 3 IP-Pakete mit
 $2 \times 1480 + 1020$ Bytes
(jeweils + IP-Header)
fragmentiert

Im Zielrechner werden die
IP-Pakete vor Weitergabe an
die Transportschicht
zusammengesetzt
(bei Verlust eines IP-Paketes
keine Weitergabe)

IP: Fragmentierung und Reassemblierung

Informationen im Paket-Header für unser Beispiel:

Beispiel

- 4000 Byte Datagram
- MTU = 1500 bytes

	length =4000	ID =x	fragflag =0	offset =0	
--	-----------------	----------	----------------	--------------	--

One large datagram becomes
several smaller datagrams

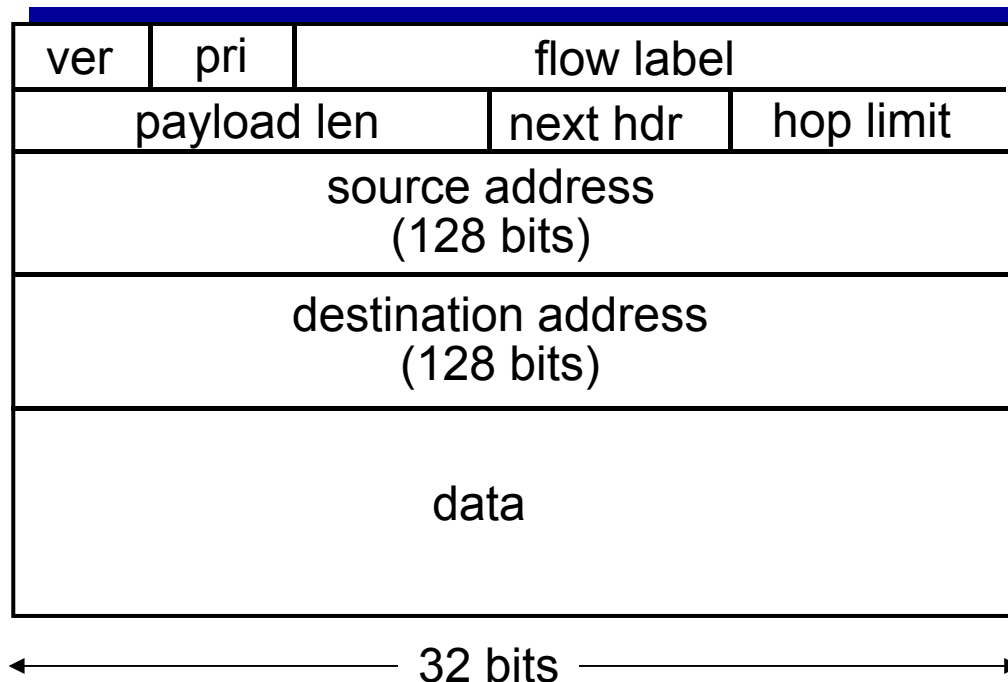
1480 bytes in
data field

offset =
 $1480/8$

	length =1500	ID =x	fragflag =1	offset =0	
	length =1500	ID =x	fragflag =1	offset =185	
	length =1040	ID =x	fragflag =0	offset =370	

IP v6: Neue Version des IP

- Hauptproblem: IP v4: Adressenmangel, 32-Bit Adressen
→ IP v6: 128 Bit Adressen
„Jedes Sandkorn der Erde adressierbar“
- Trotzdem „schlankere“ Header: Zusatzheader-Konzept
 - Header: 40 Byte, unfragmentiert



Priorität

Flusslabel: Id für
ausgehandelten
Verkehr

Next Hdr:
Zusatzheader

IP v6: Adressaufbau

Adressaufbau: x:x:x:x:x:x:x:x

(x = 16 hexadezimal, führende Nullen weglassen)

2001:0000:85A3:08D3:0019:8A2E:0370:7347

⇒ 2001:0:85A3:8D3:19:8A2E:370:7347

Schreibweise für Netzbereich (ähnlich zu IP v4):

2001:0:85A3:8D3::/64

Ersten 64 Bit *Prefix* Rest *Interface*

Adresszuweisung

- Provider-bezogen
- geographisch

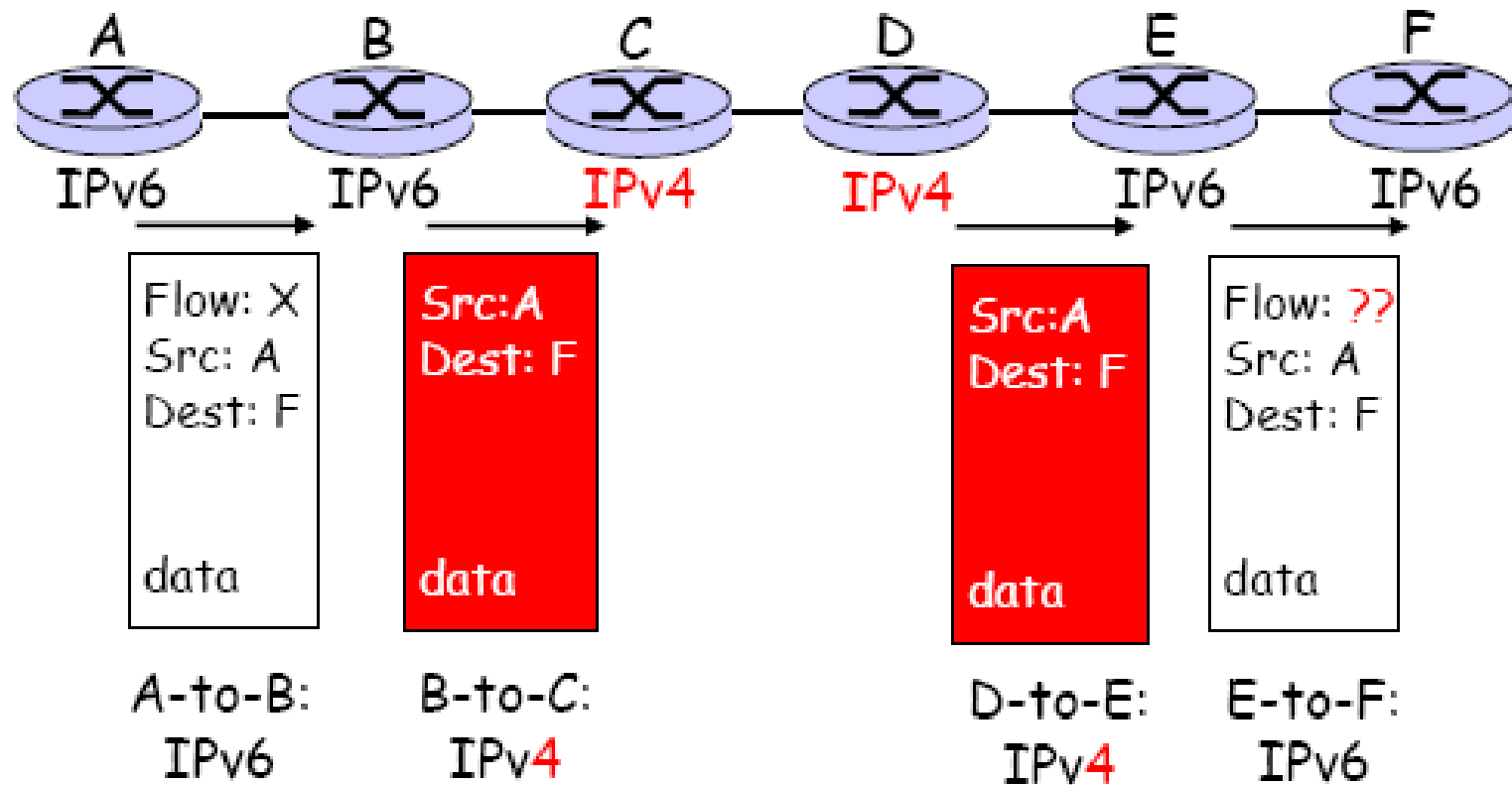
IP v6: Neue Version des IP

- Weitere Änderungen von IP v6:
 - Keine Prüfsumme in Header (schnelle Verarbeitung)
 - IPsec „IP-Security“, VPN-Technik
 - Mobile IP
 - Unterstützung von Multicast und QoS

Übergang von IP v4 nach IP v6: Läuft seit Jahren!

- 2 Möglichkeiten zum gleichzeitigen Betrieb beider Protokolle
 - **Dual Stack**
Neue Router können auch IP v4
 - **Tunneling**
IP v6 Pakete werden, um IP v4 Netz zu durchlaufen in IP v4 Pakete eingepackt

IP v 6: Dual Stack



© Kurose/Ross 2009

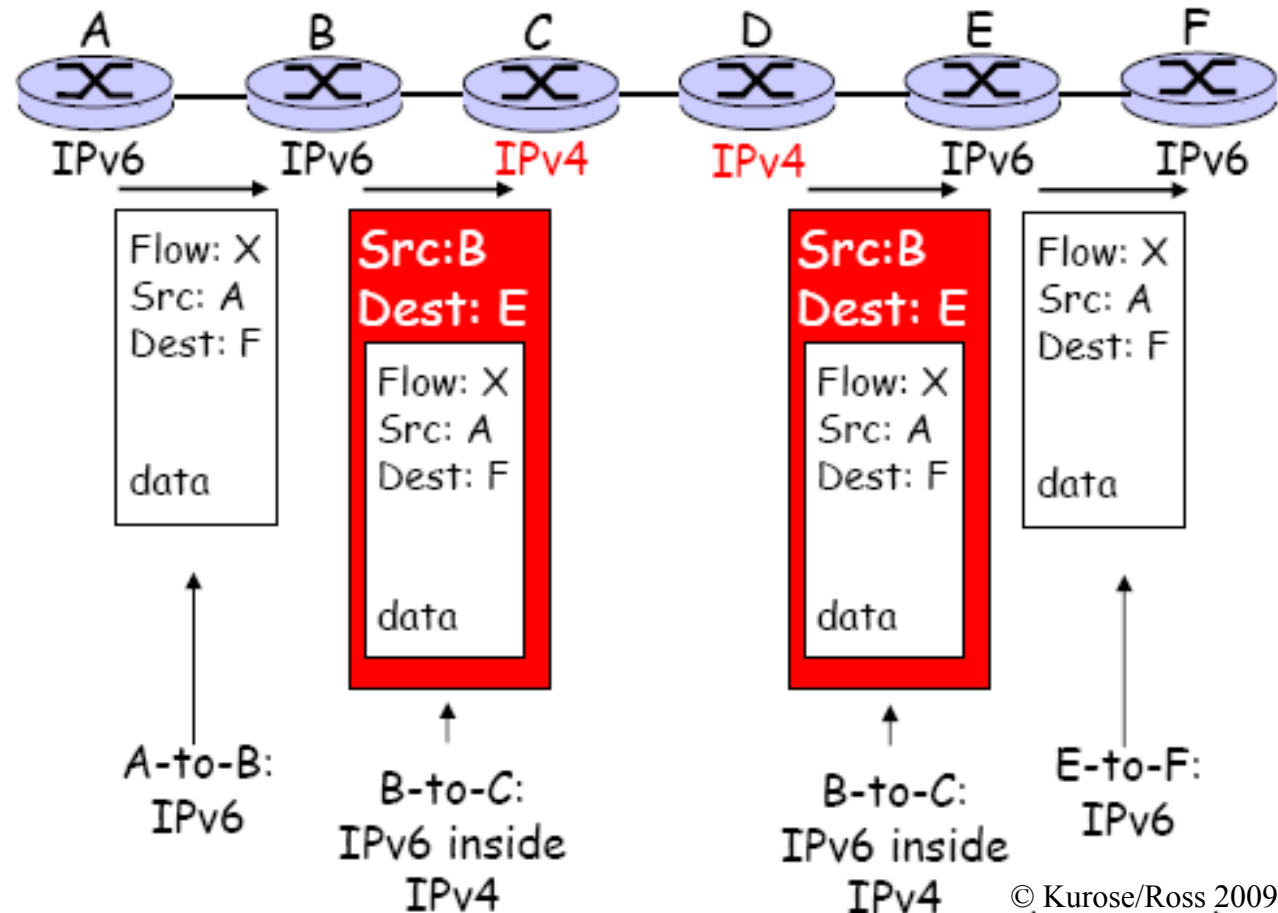
- Router B und E „verstehen“ IPv 4 und 6 und übersetzen! (dual stack)
- Zusätzliche Information von IPv 6 geht verloren!

IP v6: Tunneling

Logische Sicht



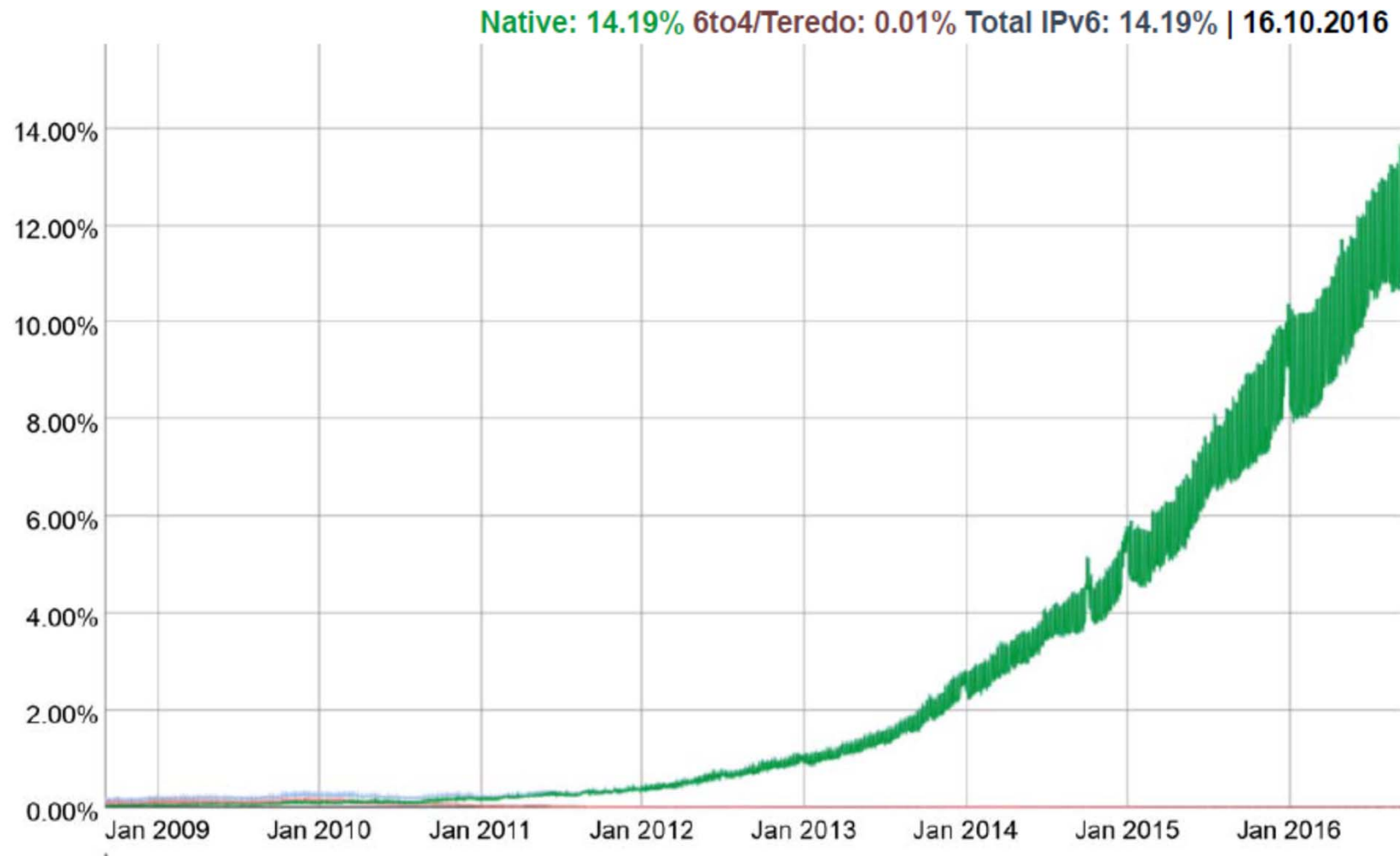
Physikalische Sicht



Router B und E verstehen IP V4 und V6:

- B verpackt V6 Paket in V4 Paket
- E entpackt V6 Paket aus dem empfangenen V4 Paket

IP v6: Verbreitung



© Google 2016

© Peter Buchholz 2016 (nach
Kurose/Ross 2003-2013)

RvS Kap. 4 Die Vermittlungsschicht

44

ICMP Internet Control Message Protocol

- Nutzung durch Hosts, Router, Gateways, um Informationen über das Netzwerk zu verteilen:
 - Fehlerberichte
z.B. unerreichbare Hosts, ...
 - Echo-Nachrichten
z.B. ping
- Eigentlich Protokoll oberhalb von IP (Vermittlungs-Schicht)
 - ICMP-Nachrichten werden in IP-Paketen übertragen
- Aufbau von ICMP-Nachrichten:
 - Type
 - Code
 - Ersten 8 Bytes des IP-Paketes, das den Fehler hervorgerufen hat

<u>Type</u>	<u>Code</u>	<u>description</u>
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 - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

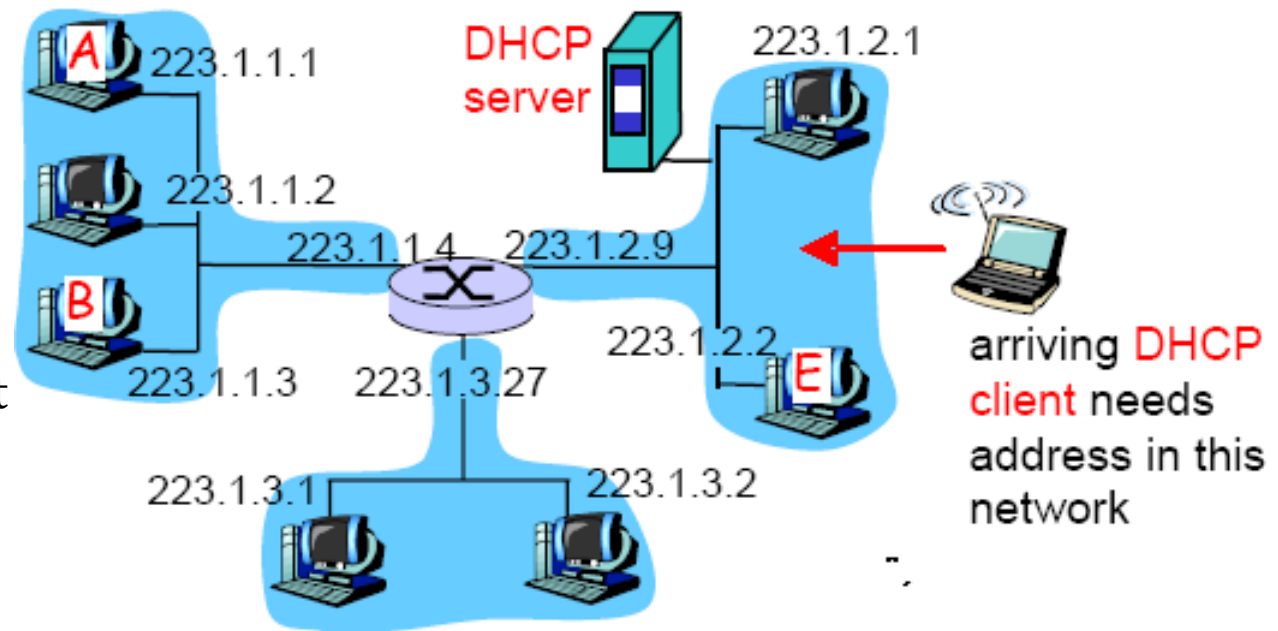
DHCP: Dynamic Host Configuration Protocol

- Lokales Netz:
Hosts kommen dazu, Hosts werden entfernt
Jeder Host braucht eine IP-Adresse → Administrationsaufwand
- WLAN- und ISP-Strukturen: Viele potentielle Hosts

- Dynamische Adresszuweisung

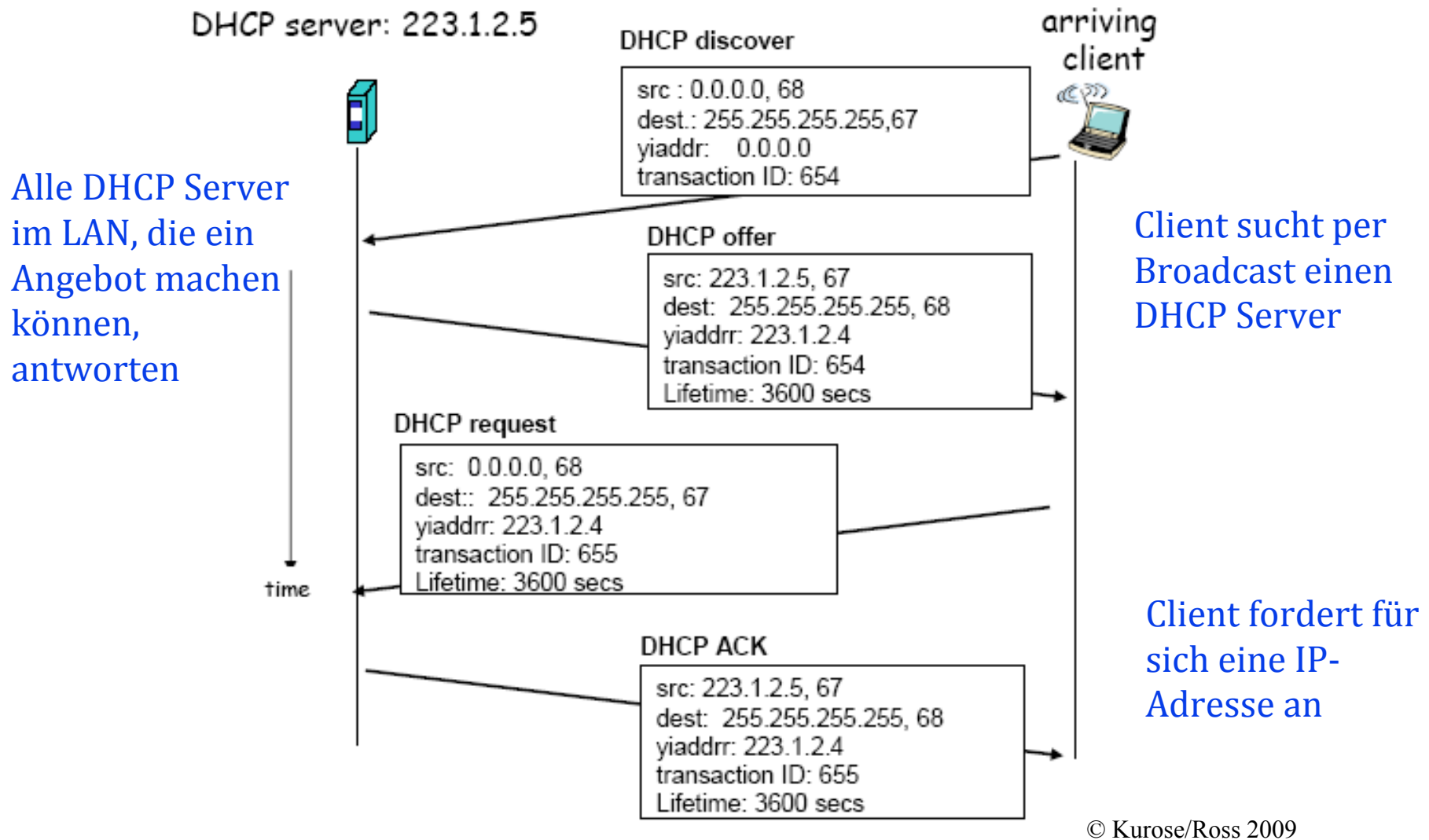
DHCP Server verteilt Adressen

Host tritt bei Netzbeitritt als DHCP Client auf



© Kurose/Ross 2009

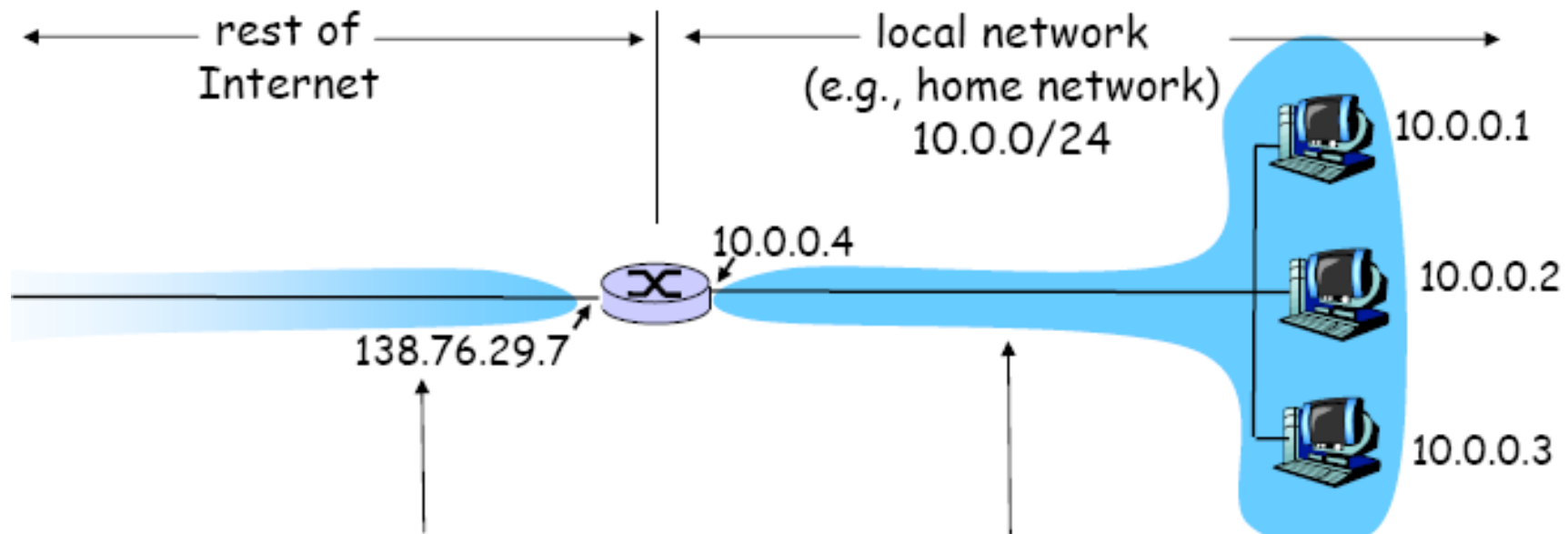
DHCP: Dynamic Host Configuration Protocol



© Kurose/Ross 2009

NAT: Network Address Translation

- „Zu wenig IP-Adressen“ (z.B. ISP weist eine einzige Adresse zu)
- Im öffentl. Internet soll man Hosts des Innennetzes nicht kennen

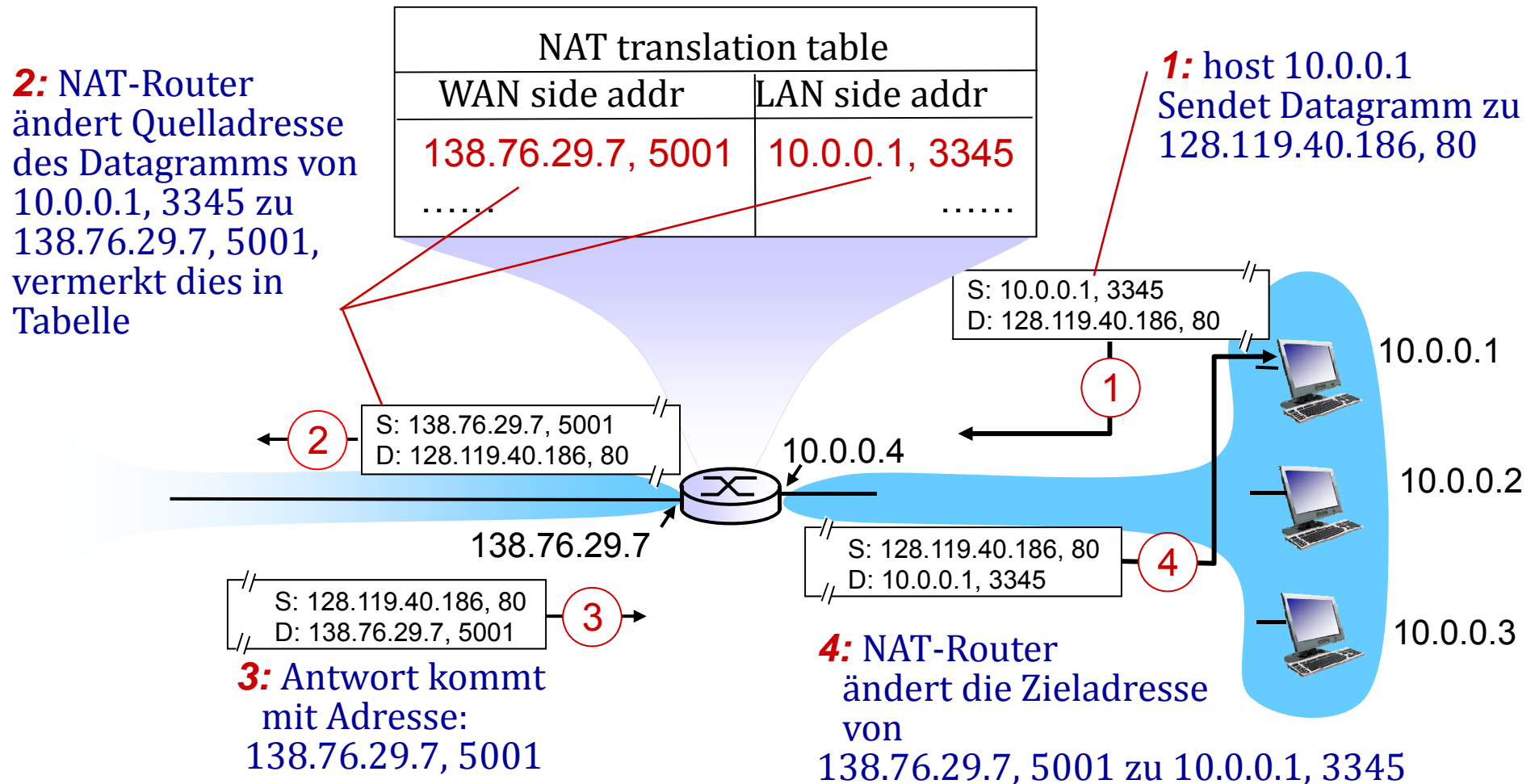


Alle IP-Pakete, die nach außen gehen, haben die Quelladresse 138.76.29.7. Port-Nummern werden genutzt, um interne Hosts zu adressieren

Alle IP-Pakete, die innen bleiben, haben innere Quell- und Zieladressen (10.0.0/24)

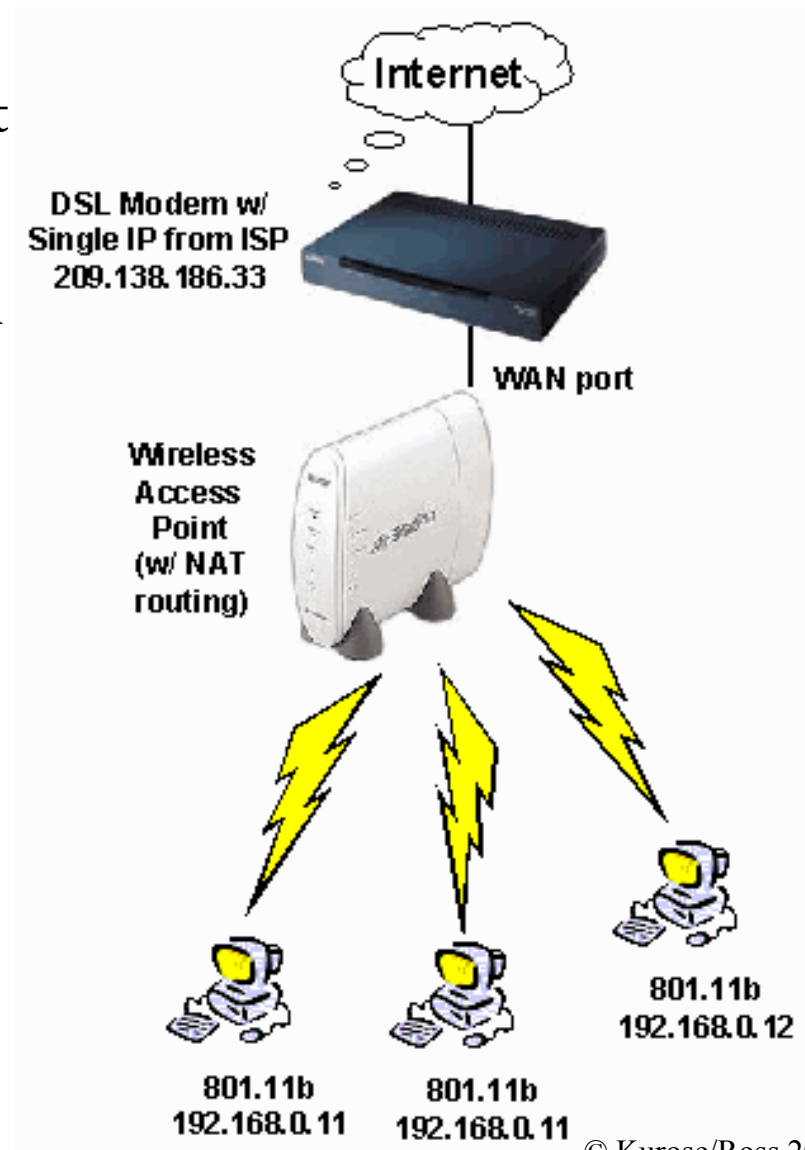
© Kurose/Ross 2009

Adressumsetzung in NAT



NAT: Network Address Translation

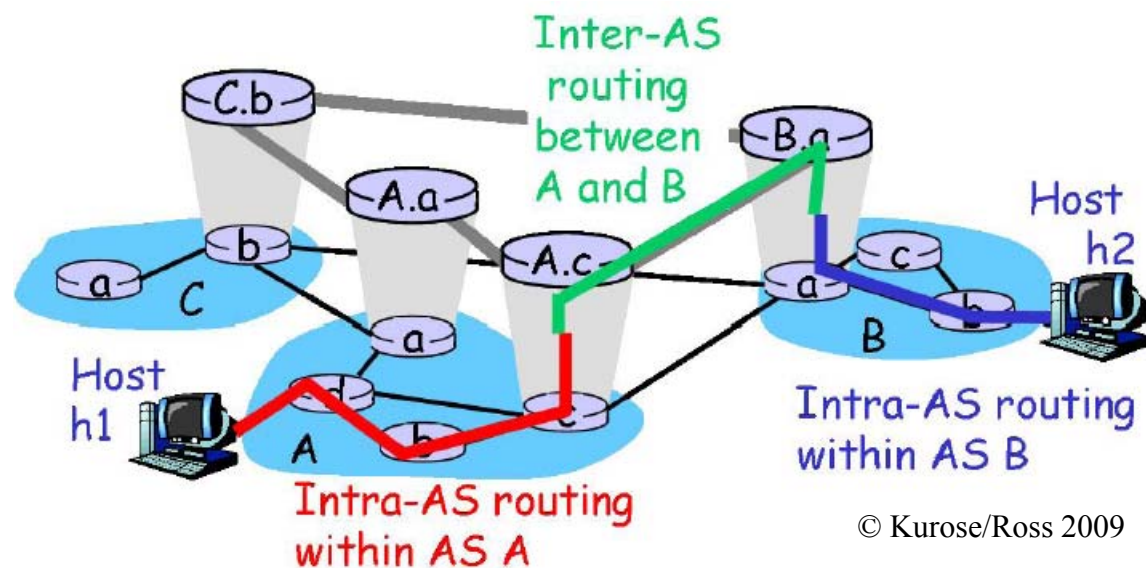
- NAT ist eine Notlösung und verursacht selbst Probleme.
 - Portnummern sollen Anwendungsprozesse adressieren und nicht Hosts!
 - Es gibt Protokolle, wo mehrere Verbindungen im Zusammenhang benutzt werden und Portnummern innerhalb von APDUs ausgetauscht werden (z.B. FTP, VoIP): Hier muss NAT in APDUs reinschauen und dort Portnummern umsetzen.
- Nicht sichtbare Hosts sind noch lange nicht geschützt!



© Kurose/Ross 2009

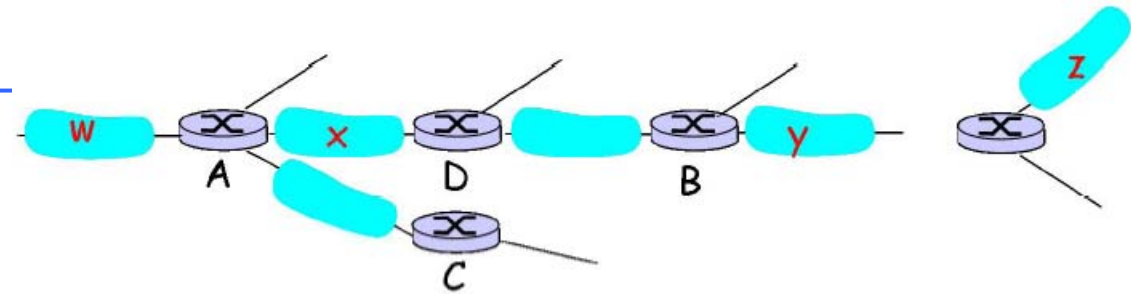
Internet: Routingtabellen-Pflege

- Internet: Netz aus autonomen Subnetzen (AS)
- A) Routing im AS
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First Protocol
 - EIGRP: Enhanced Interior Gateway Routing Protocol
- B) Routing zwischen AS / Inter-Domain-Routing
 - BGP: Border Gateway Protocol



© Kurose/Ross 2009

A) RIP



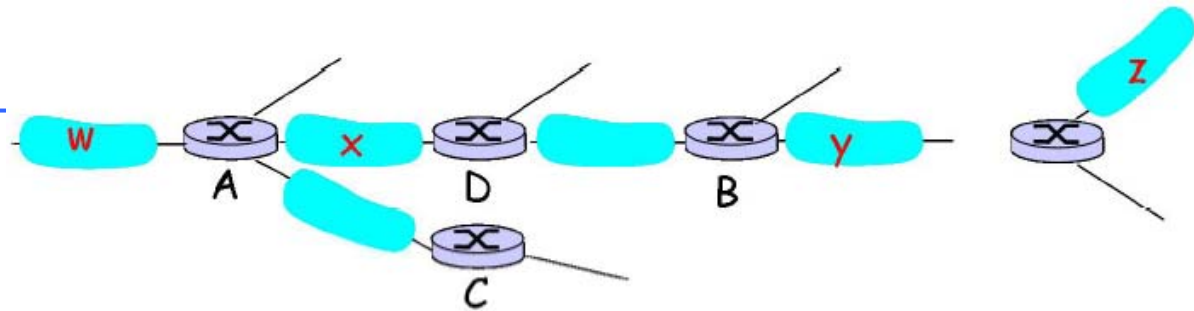
Destination Network	Next Router	Num. of hops to dest.
W	A	2
Y	B	2
Z	B	7
X	--	1
...

Routing table in D

➤ Distanzvektor-Protokoll

- Hop-Anzahl als Weglänge, Begrenzung auf 15 Hops
- Austausch von Routing-Tabellen alle 30 Sekunden
- Maximale Zahl der Einträge: 25
- Austauschnachrichten zwischen Nachbarn (Hopzahl: 1) werden als *Advertisements* bezeichnet.

RIP



Destination Network	Next Router	Num. of hops to dest.
W	A	2
Y	B	2
Z	B	7
X	--	1
...

Routing table in D

Dest	Next	hops
W	-	-
X	-	-
Z	C	4
...

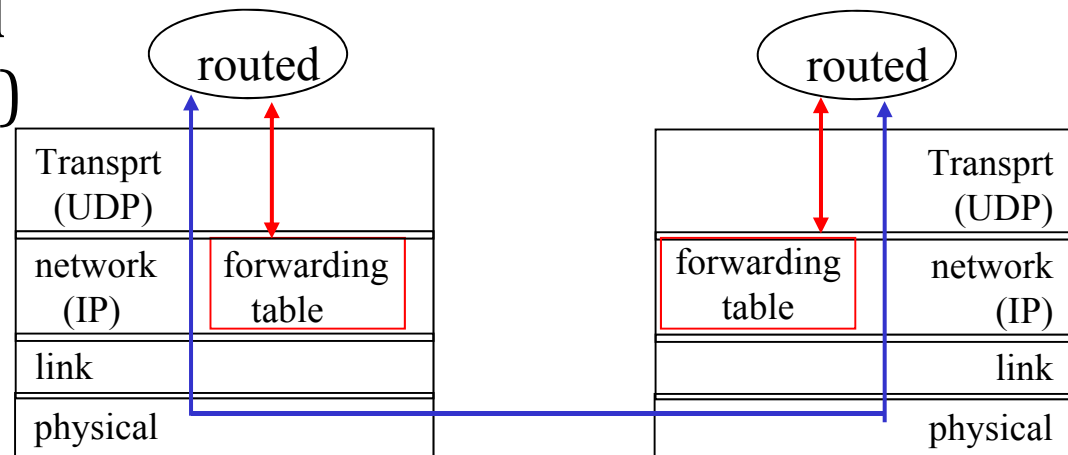
Advertisement von Router A

Destination Network	Next Router	Num. of hops to dest.
W	A	2
Y	B	2
Z	B A	7 5
X	--	1
...

Routing table in D

RIP

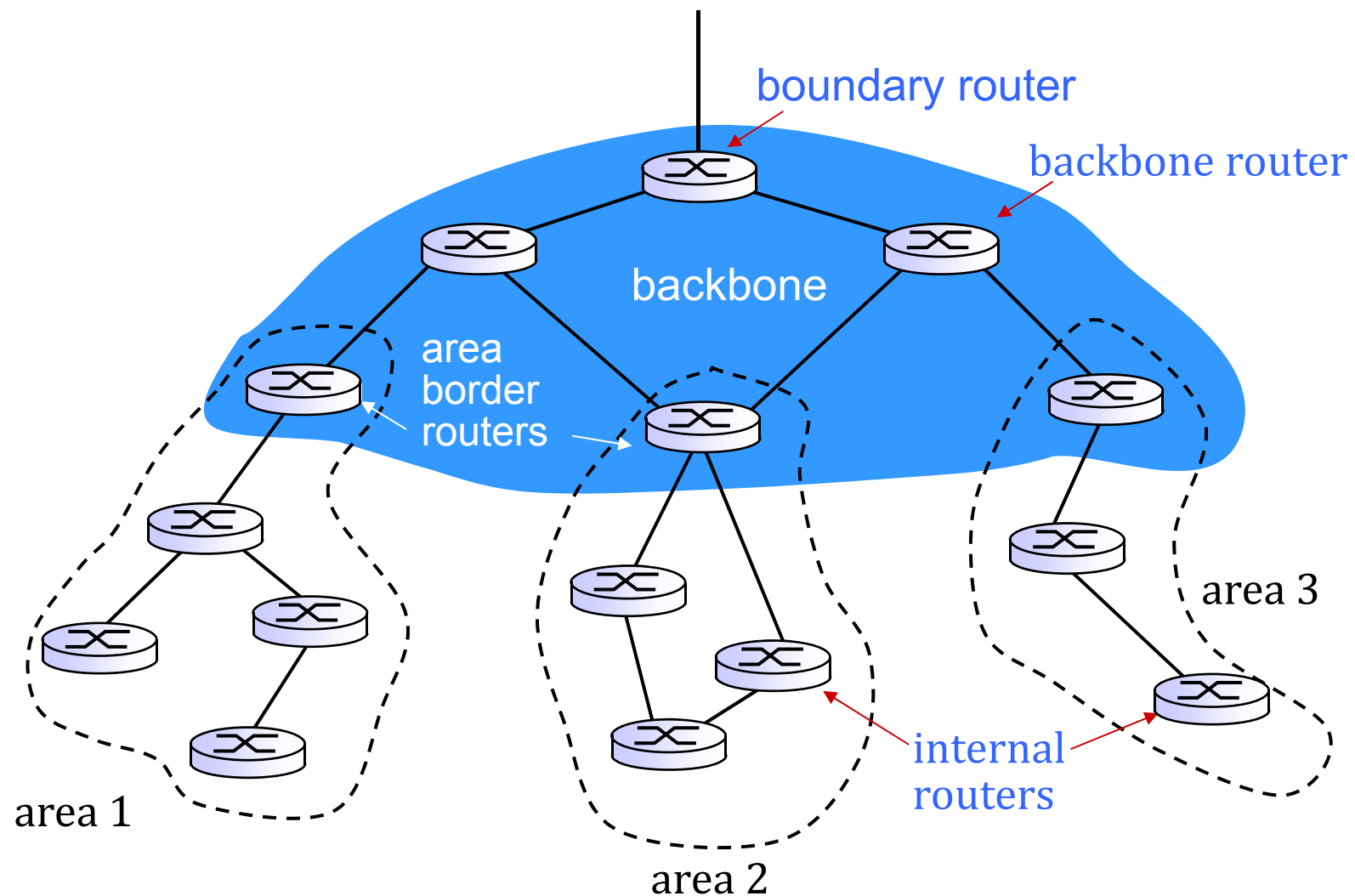
- RIP ist auf der Anwendungsebene implementiert
- RIP-Nachrichten werden üblicherweise in UDP-Datagrammen übertragen
- Falls ein Router innerhalb von 180 Sekunden nicht von einem Nachbarn hört, wird angenommen, dass dieser nicht erreichbar ist
- Router können bei ihren Nachbarn nach den Kosten zu einem bestimmten Ziel fragen (Request/Reply)



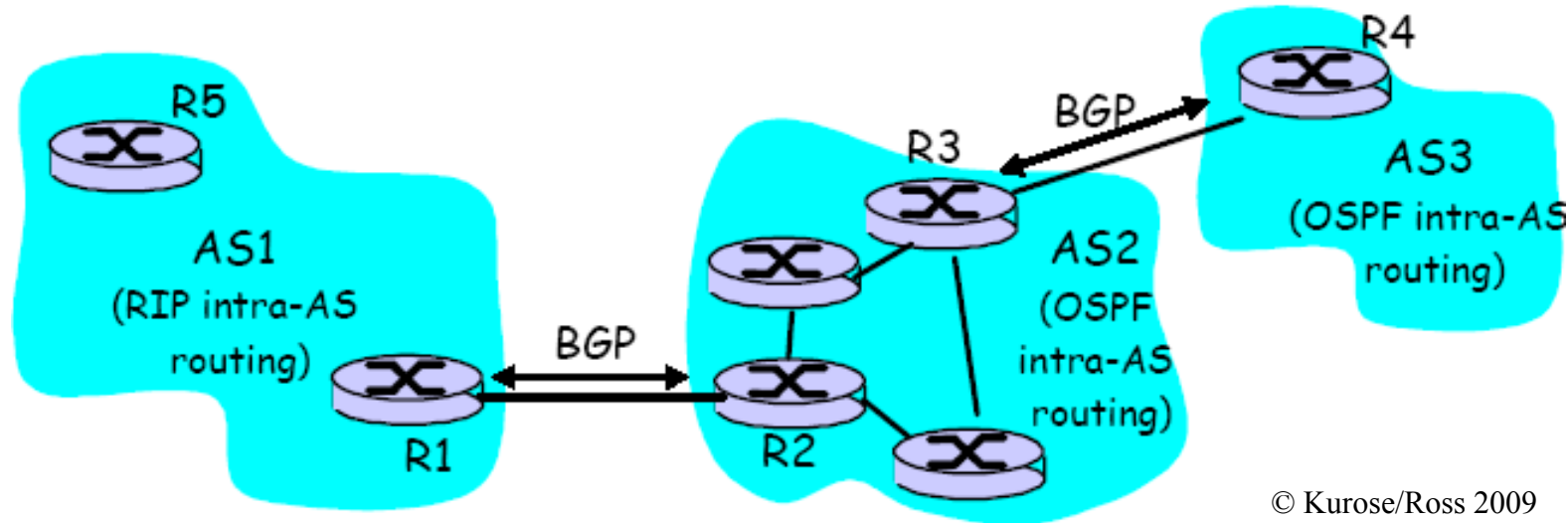
A) OSPF

- Link-State-Protokoll
- Aufbau einer Darstellung der Gesamt-Topologie durch Kommunikation mit allen Routern
- Zentrale Ausführung des Dijkstra-Algorithmus um eine vollständige Kostentabelle pro Router zu bestimmen
- Besonderheiten:
 - Authentifizierung von Routern (Sicherheit)
 - Bei mehreren Pfaden mit gleichen Kosten: Verkehr zwischen A und B über verschiedene Pfade (parallel)
 - Unterschiedliche Kanten-/Verbindungskosten (z.B. höhere Kosten für zeitkritischen Verkehr) (variable Pfadermittlung)
 - Unterstützung von Multi-/Broadcast
 - Unterstützung von hierarchischen Netzstrukturen (verschiedene Rollen für Router)

Hierarchische Netzstrukturen

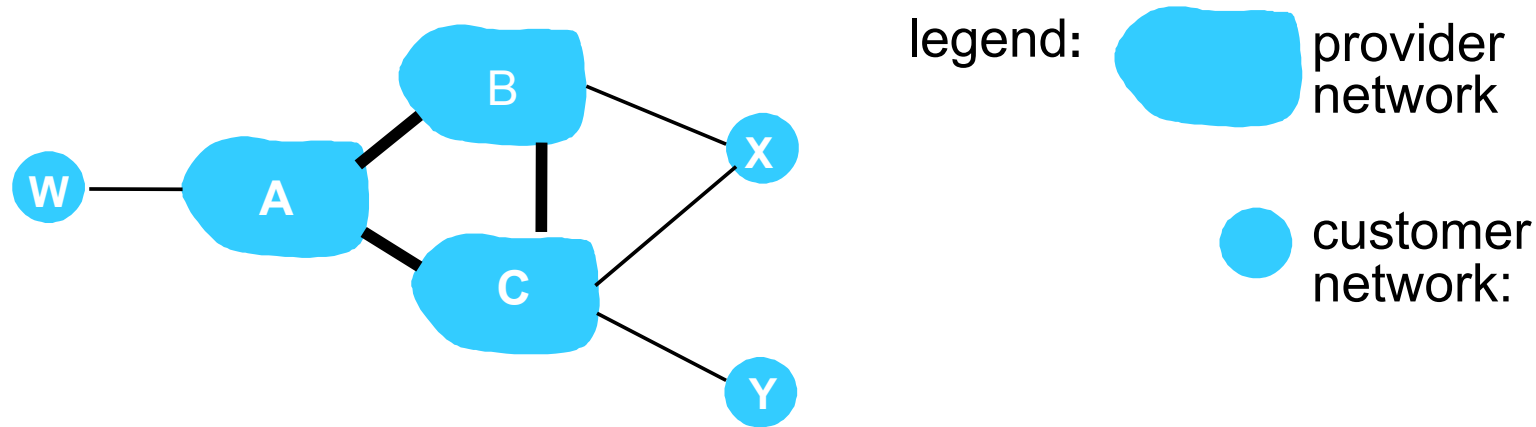


B) Inter-AS Routing im Internet: BGP



- **BGP (Border Gateway Protocol):** De-facto-Standard
- **Pfadvektor-Protokoll:**
 - Ähnlich zum Distanzvektor-Protokoll
 - Jedes (Border) Gateway sendet per Broadcast bekannte Pfade zu seinen Nachbarn (peers): *Advertisements and Withdrawals*
Z.B. Gateway X sendet seinen Pfad zu Z: **Path (X,Z) = X,Y1,Y2,Y3,...,Z**

Inter-AS Routing im Internet: BGP



- A,B,C sind Netze von Betreibern
- X,W,Y Kunden (der Betreiber)
- X ist an B und C angeschlossen
 - Falls X keine Nachrichten von B nach C weiterleiten möchte
 - .. wird X B keine Route nach C anbieten!

Multicast-Routing

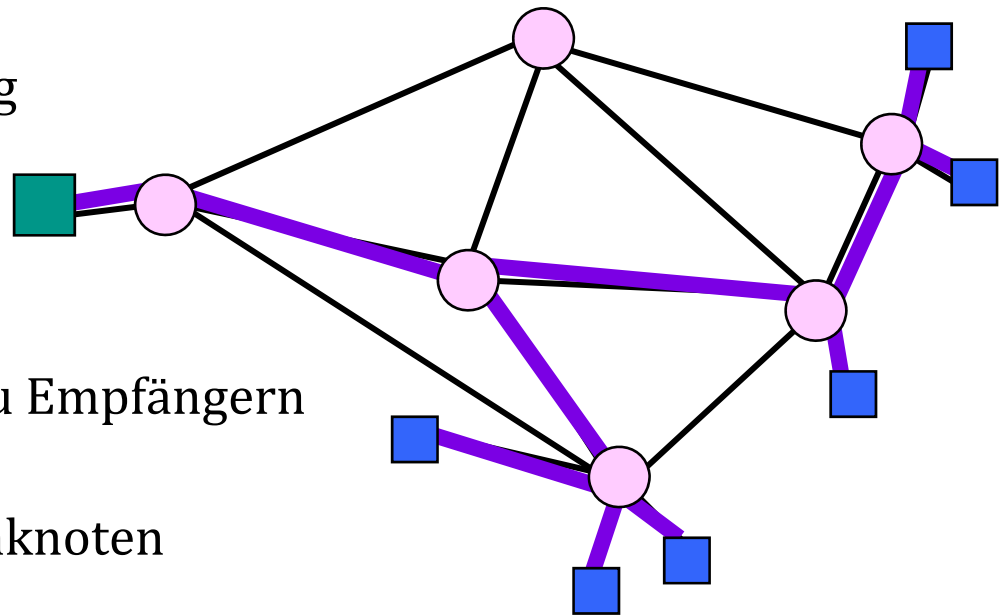
- Gruppen-Kommunikation: Senden an alle Mitglieder einer Empfänger-Gruppe

- für Software-Verteilung
- für Konferenz-Übertragung
- für Telekonferenzen

- Grundidee:

- Spannbaum vom Sender zu Empfängern bestimmen
- Router als Baum-Zwischenknoten duplizieren die Pakete

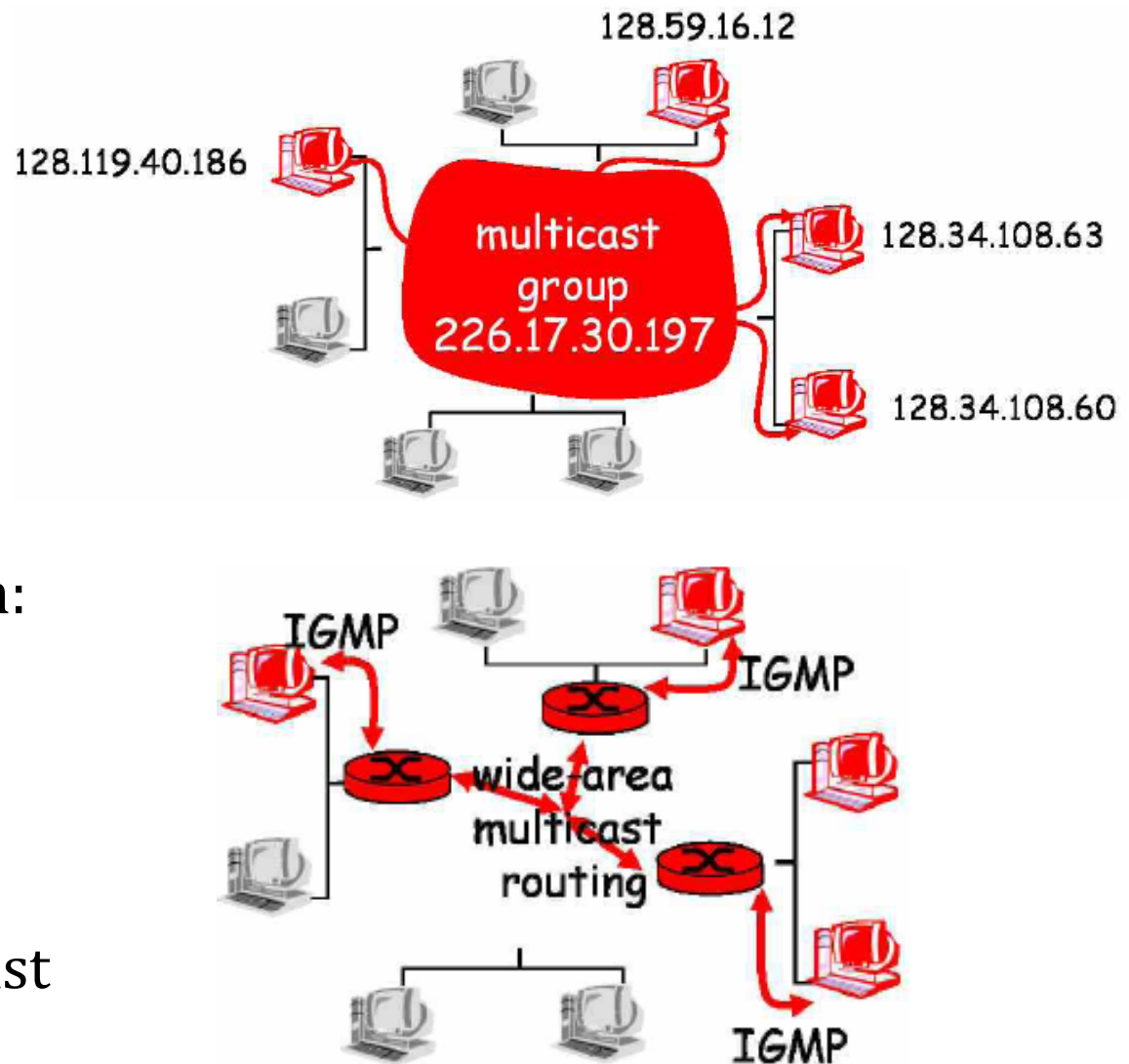
- Großes Problem: Zuverlässiger Broadcast/Multicast wegen so genannter Quittungsimplosion



© Kurose/Ross 2009

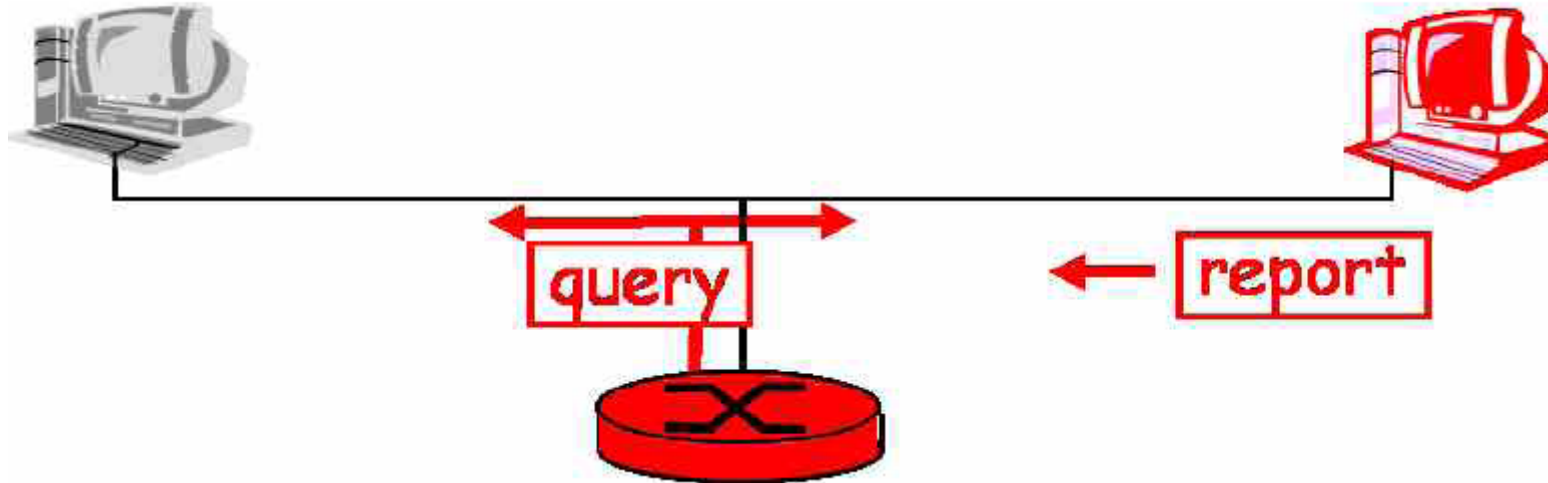
Multicast-Routing: IP-Gruppen-Adressen

- Ziel-Adresse ist Gruppen-Adresse
- Problem:
Adresse und Empfängermenge müssen vorher vereinbart werden:
Internet Group Management Protocol (IGMP) plus
Wide Area Multicast Routing



Multicast-Routing: IGMP

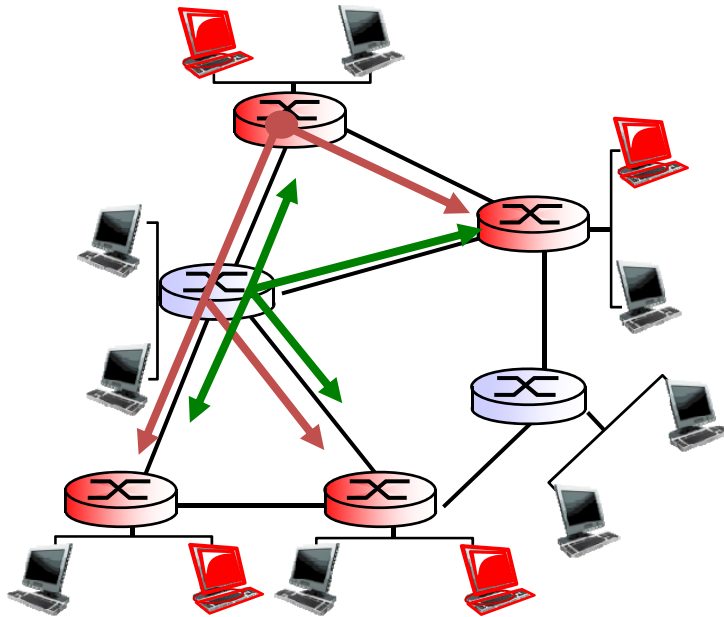
Nachrichtentypen



- a) Router fragt „seine“ Hosts, ob sie an Gruppe beteiligt sind
- b) Host sagt seinem Router, dass er an Gruppe beteiligt sein möchte

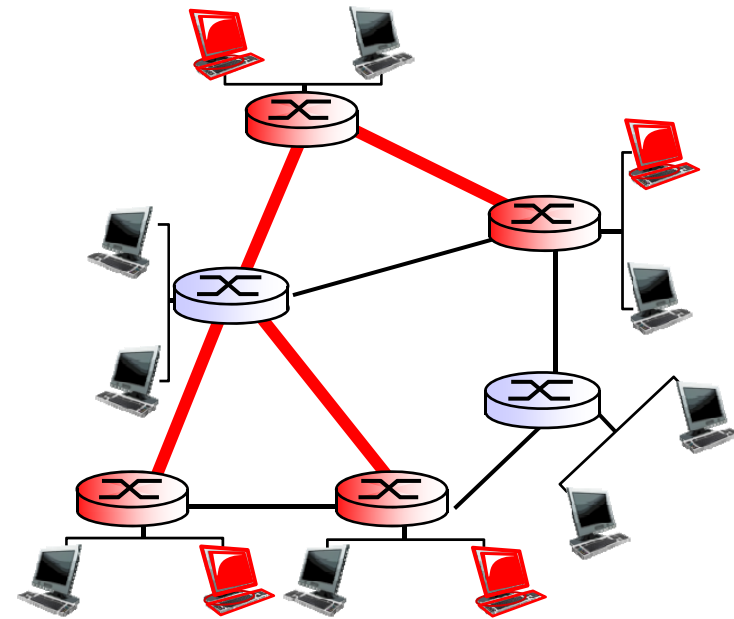
Multicast-Routing: Wide Area Multicast Routing

A) Je Sender ein
Multicast-Baum



source-based trees

B) Gemeinsamer
Gruppen-Baum



shared tree

Multicast-Routing: Wide Area Multicast Routing

- **DVMRP** (Distance Vector Multicast Routing Protocol)
Aufbau quellbasierter Multicast-Bäume mit Hilfe des Distanzvektor-Algorithmus
- **MOSP** (Multicast Open Shortest Path Routing)
Aufbau quellbasierter Multicast-Bäume mit Hilfe des Algorithmus zur Berechnung kürzester Wege
- **CBT** (core Based Trees)
Aufbau eines bidirektionalen Gruppenbaums mit einem Kern (Router)
- **PIM** (Protocol Independent Multicast)
unterschiedliche Modi
 - Dense mode: Teilnehmer nahe beieinander
(Verhalten ähnlich DVMRP)
 - Sparse mode: Teilnehmer verteilt
(Verhalten ähnlich CBT)