Rechnernetze und verteilte Systeme (BSRvS II)

Prof. Dr. Heiko Krumm

FB Informatik, LS IV, AG RvS
Universität Dortmund

- Aufgaben
- Virtual Circuit und Datagramm
- Router-Aufbau
- Routingalgorithmen
- Internet-Protokoll IP V4, IP V6
- DHCP und NAT
- Routing im Internet
- Multicast und Broadcast
- Mobile Netze

- **Computernetze und das Internet**
- **Anwendung**
- **Transport**
- **Vermittlung**
- **Verbindung**
- **Multimedia**
- **Sicherheit**
- **Netzmanagement**
- **Middleware**
- **Verteilte Algorithmen**

Vermittlungsschicht (Network Layer)

Aufgabe der Vermittlungsschicht

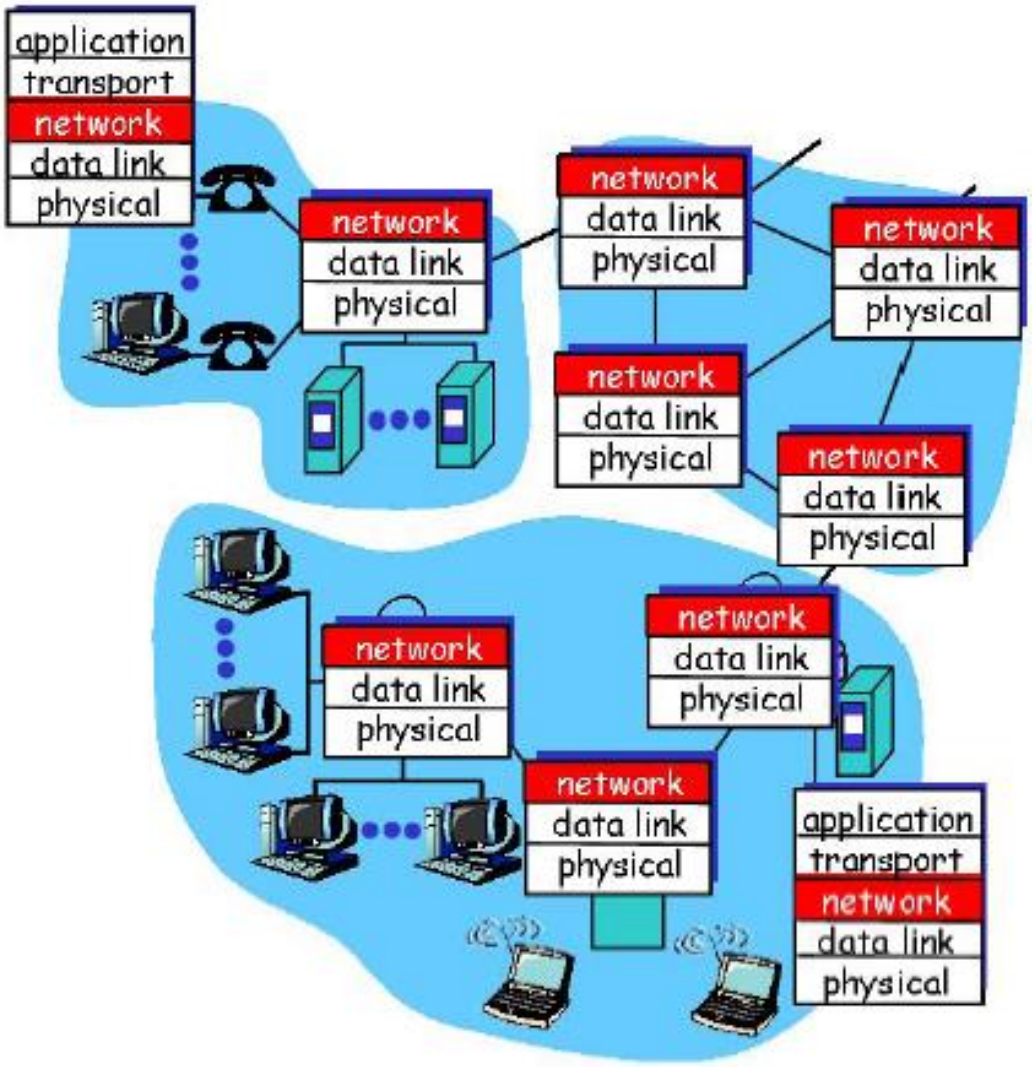
- Nachrichtenweiterleitung

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)



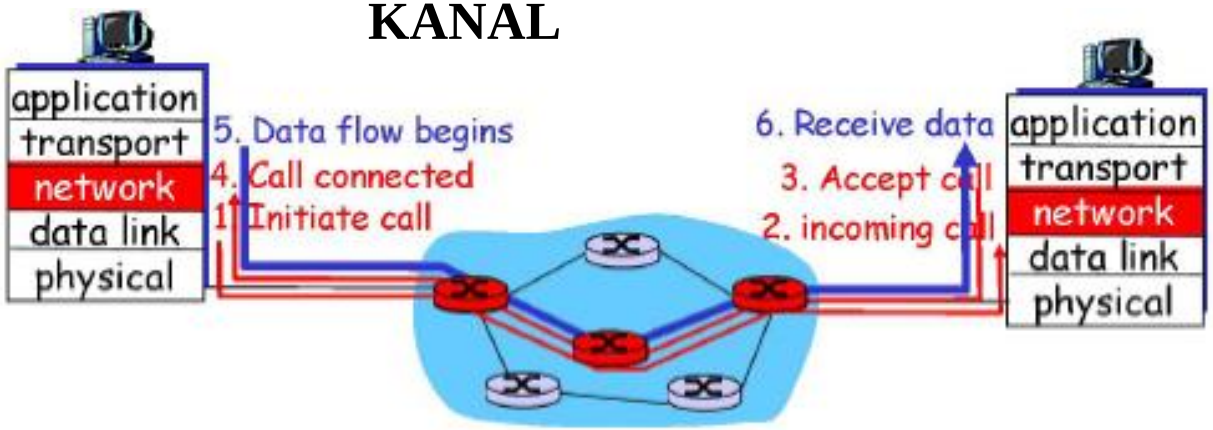
Bilde ein Netz aus Transitknoten und Teilstrecken, so dass Nachrichten zwischen beliebigen Punkten ausgetauscht werden können

Modelle / Paradigmen:

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

Virtueller Kanal (ATM) versus Datagramm-Dienst

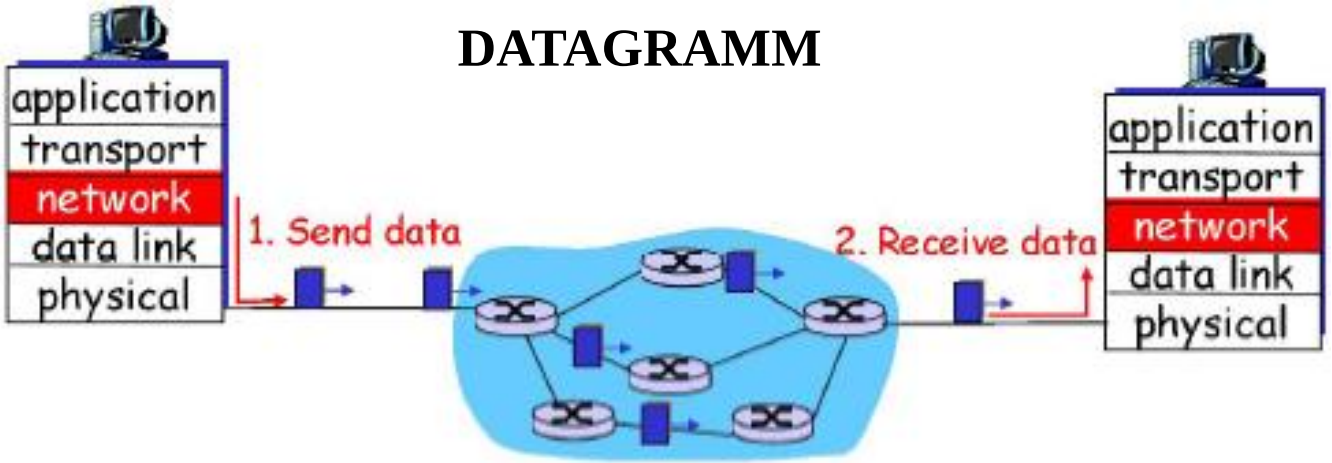
KANAL



2-Wege-Handshake bei Host: Signalisierungnachrichten

1. Initiate call
2. Incoming call from network
3. Accept/acknowledge call (target host)
4. Acknowledgement received by initiating host

DATAGRAMM



Nur Paketsenden u. Empfangen

Virtueller Kanal (ATM) - Vorteile

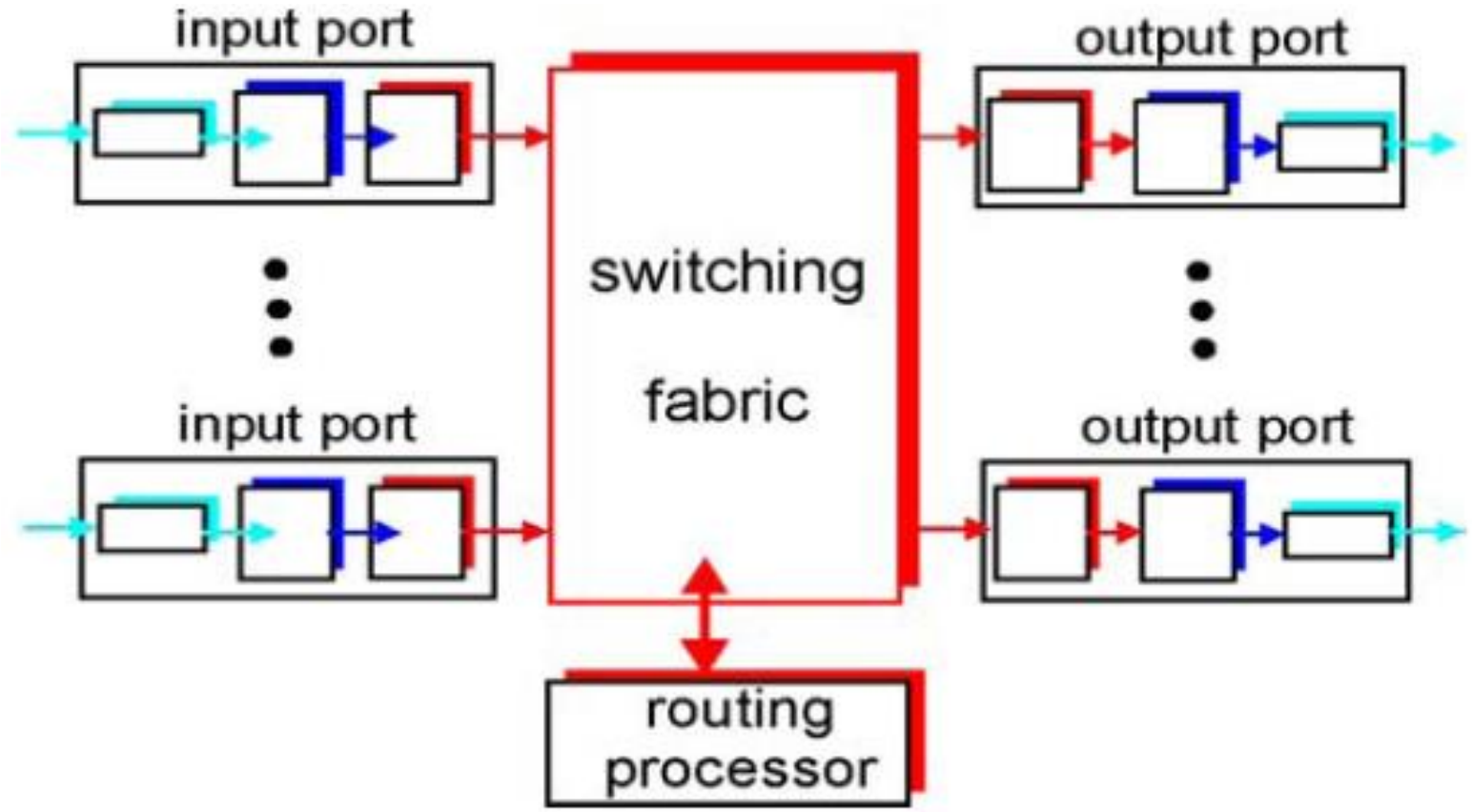
Kontext um für Menge von Pakettransfers gemeinsam Ressourcen zu reservieren und Konfigurationseinstellungen so durchzuführen, dass

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

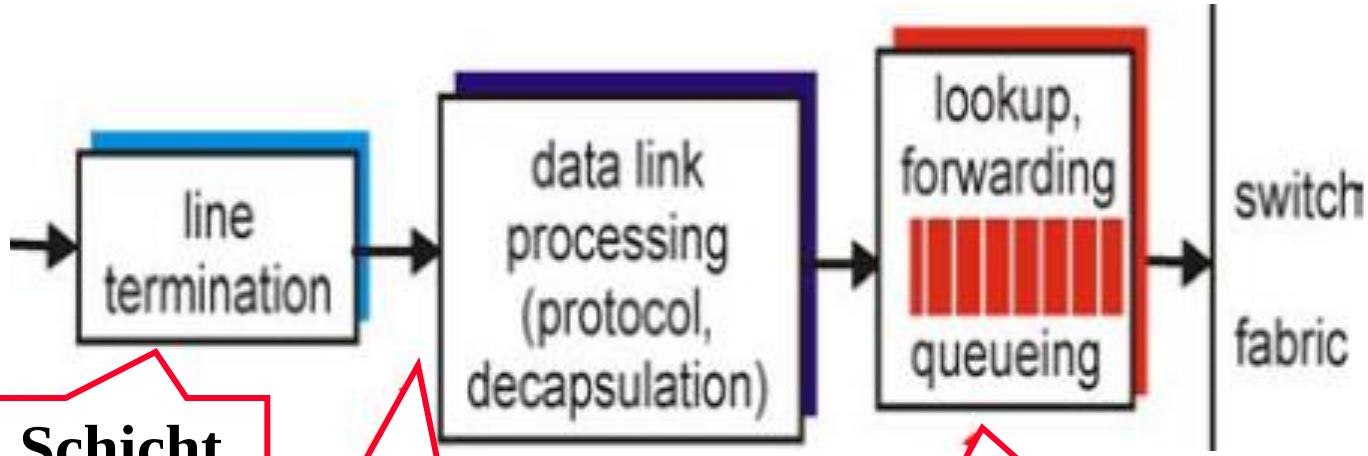
leichter realisiert werden können.

Interner Aufbau eines Routers

- Zwei Hauptfunktionen eines Routers:
  - Vermittle IP-Pakete: IP-Weiterleitung
  - Pflege Routingtabellen per RIP/OSPF/BGP



Input Port Funktionen



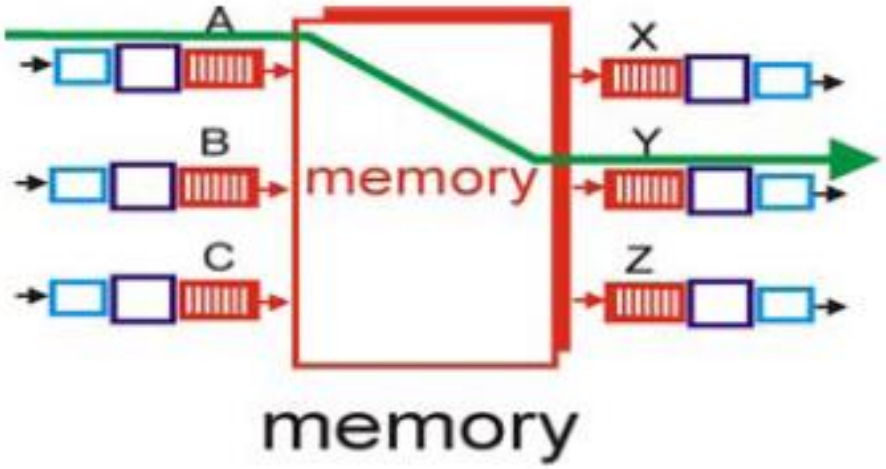
Physik. Schicht

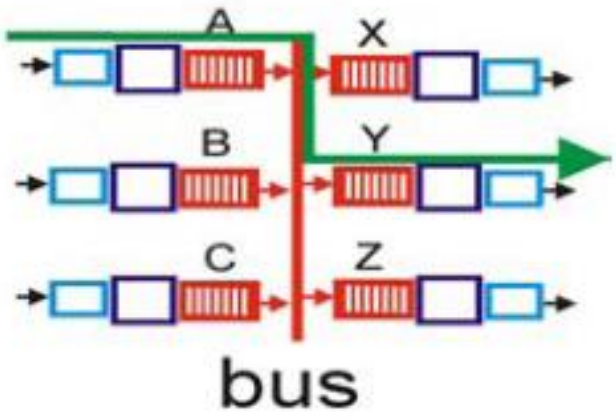
Data Link Protokoll

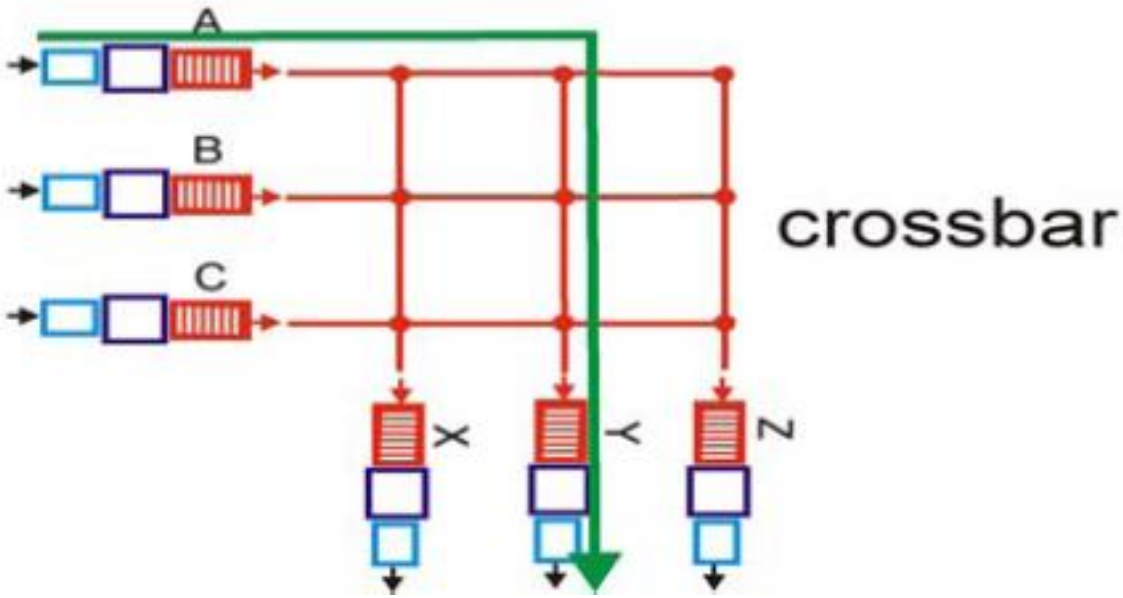
Dezentralisiertes Switching

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

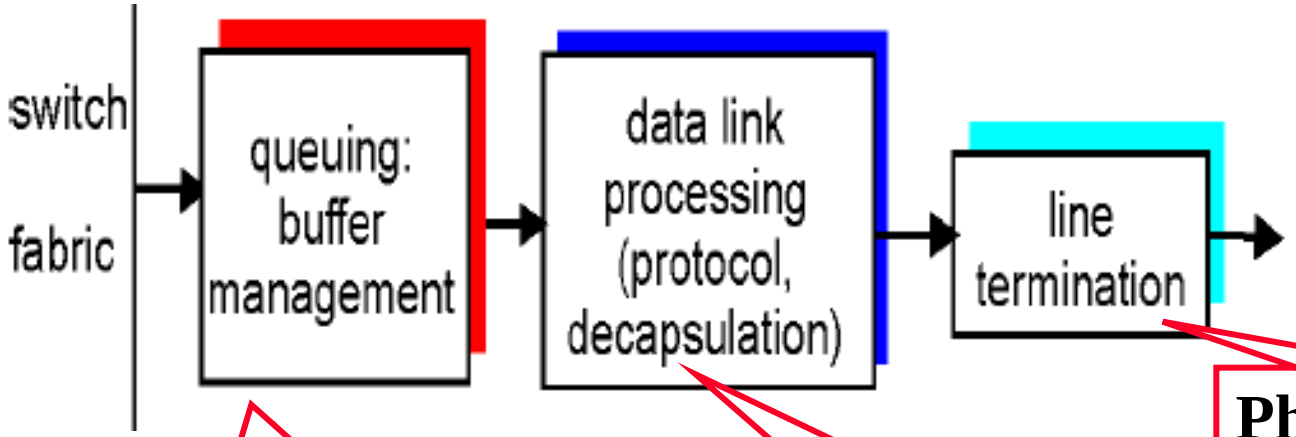
Switching Fabric – Vermittlung: 3 Typen







Output Port Funktionen



Physik. Layer

Data Link Protokoll

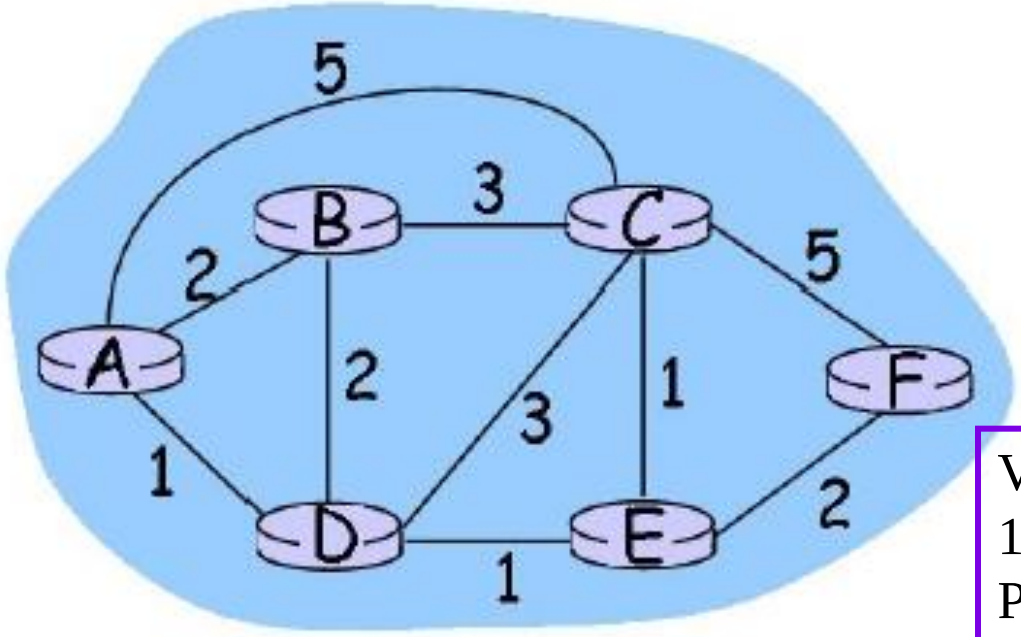
Pufferung von Datagrammen

wenn diese schneller aus Fabric ankommen, als sie über Ausgabeleitung übertragen werden können

- Scheduling zur Auswahl von zu übertragenden Paketen ist notwendig

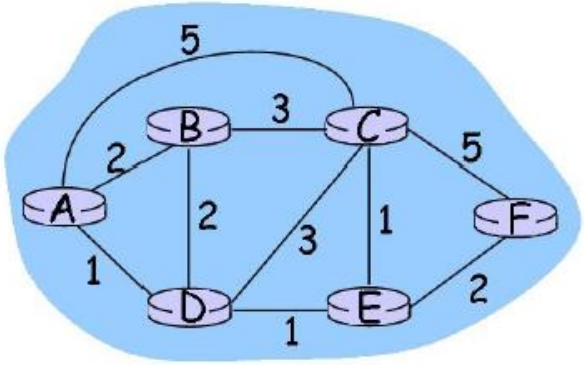
Routing-Algorithmen

- 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 C gibt es hier 17 Pfade. Der kürzeste Pfad ist ADEF

Routing-Algorithmen

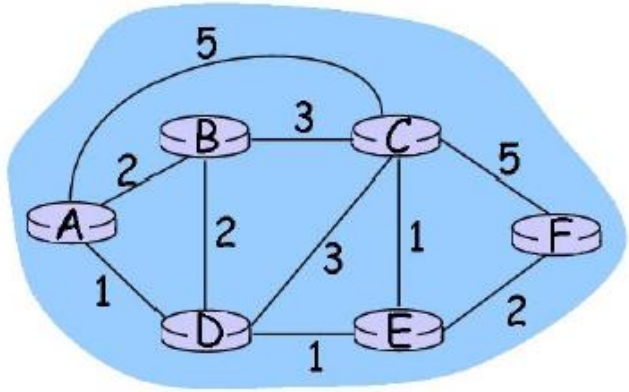


- ***Global***
  je Kante Kantenverlauf und Kosten global bekannt
  Problem: Skalierbarkeit
- ***Dezentral***
  Jeder Router kennt nur die Kanten zu seinen Nachbarn

- ***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** (dyn. global) ↔ **Dist. Vector** (dyn. dezent.)

Link-State: Dijkstra-Algorithmus



Gegeben:

K Knotenmenge eines gegebenen Graphen

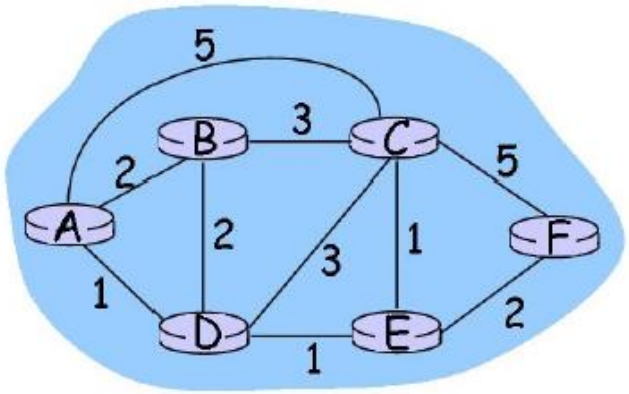
$c(i,j)$ Verbindungskosten von Knoten i nach j

$$c(i,j) \begin{cases} \text{Kosten} & \text{von } v_i \rightarrow v_j \text{ (der Kante)} \\ \infty & \text{falls von } v_i \text{ nach } v_j \text{ keine Kante} \end{cases}$$

$D(o)$ Kosten des (ermittelten) Pfades vom Sender zum derzeitigen Ziel o (hat unter allen 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

Der Algorithmus berechnet einen Weg von einem Sender v_0 zu jedem Ziel v mit minimalen Kosten in $O(n^2)$ Schritten (n = Knotenanzahl).

Der Algorithmus setzt konstante Kosten voraus. In der Praxis unterliegen die Kosten jedoch während des Betriebs Veränderungen.

Der Algorithmus soll hier nicht vertieft behandelt werden:
Typischer Stoff für DAP II.

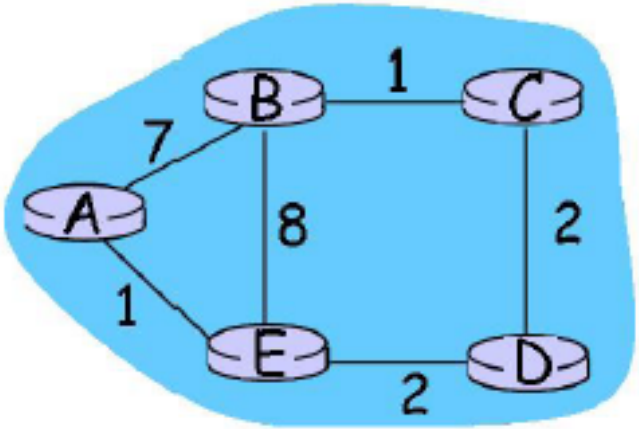
Der nachfolgende Distance Vector Algorithmus hat dagegen eine andere Beschaffenheit:
Es ist ein verteilter Algorithmus.

Distance Vector: Dezentrales Routing

Ziel

Jeder Knoten schickt ein Paket über den direkten Nachbarn, über den die geringsten Kosten entstehen.

Dazu könnte, *wenn sich die Kosten nicht ändern*, eine **Distanz-Tabelle** verwendet werden.



$D^E()$

destination	cost to destination via A	via B	via D
A	1	15	12
B	8	8	5
C	9	9	4
D	10	11	2

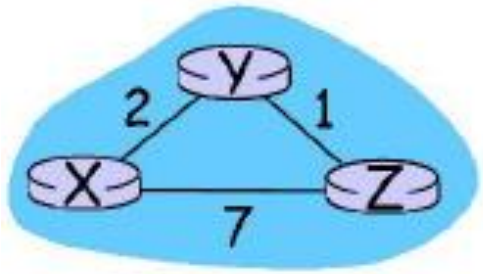
Distance Vector: Algorithmus

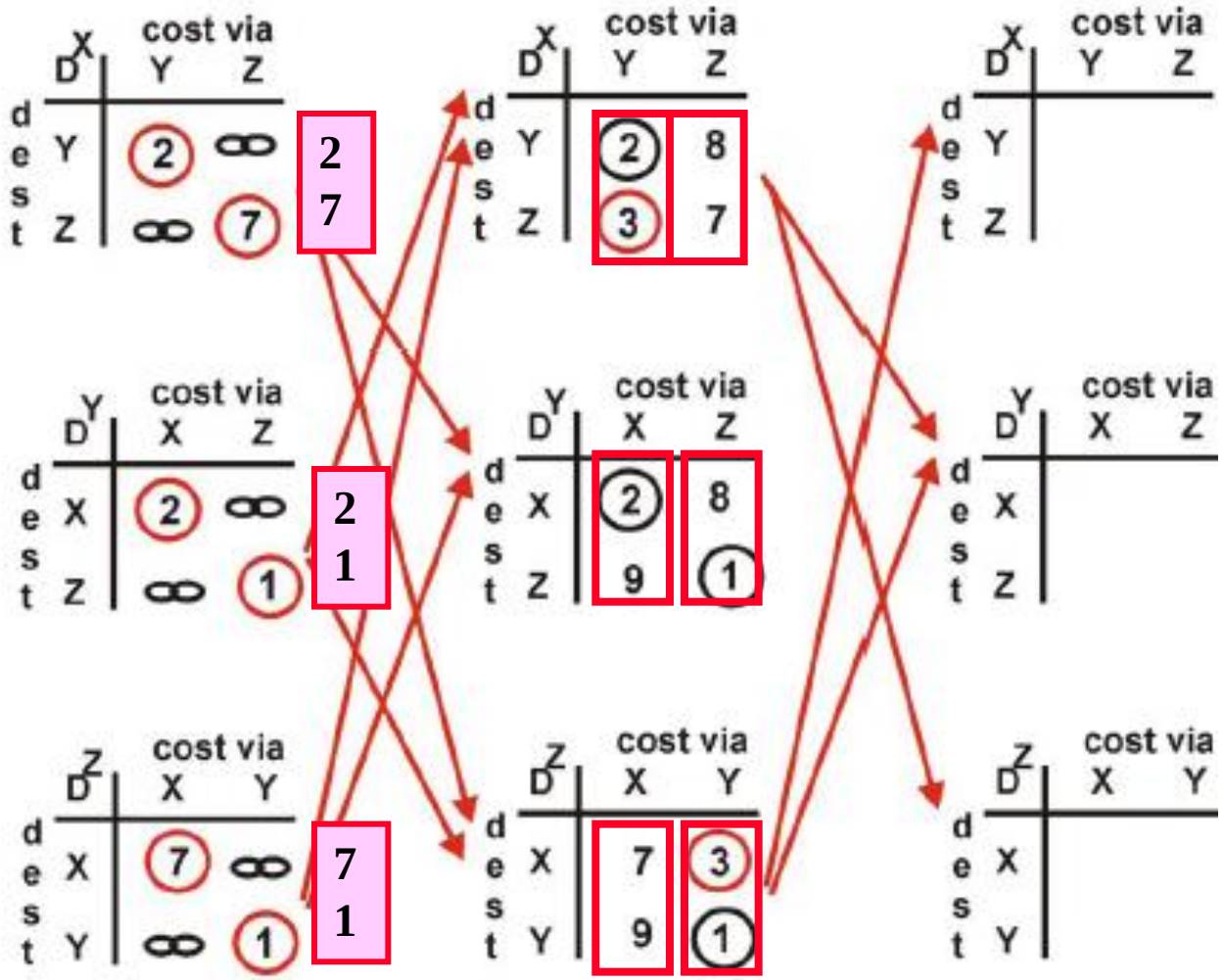
```
1    initialisation:
2    for all adjacent node v
3        D^x(*,v) = ∞ /* * = alle Zeilen */
4        D^x(X,v) = c(X,v)
5    for all destinations, y
6        send min_w D(y,w) to each neighbour
7    loop
8        wait (until I see a link cost change to neighbour V
9              or until I receive an update fom neighbour V)
10       if (c(X,V) changes by d)
11            for all destinations y
12                D^x(y,V) = D^x(y,V) + d
13       else if (update received from V wrt destination y)
14            for singe destination y
15                D^x(Y,V) = c(X,V) = c(X,V) + newval
16       if (we have a new min_w D(Y,w) for any destination Y)
17            send new value of min_w D(Y,w) to all neighbours
```

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

Distance Vector: Algorithmus - Beispiel

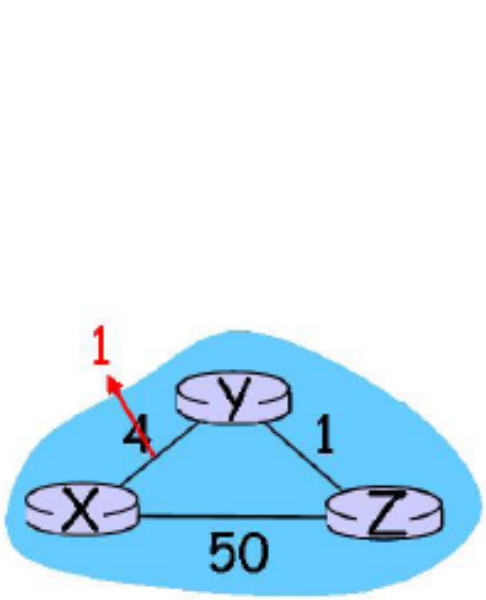
D^X	X	Y	Z
X	-	-	-
Y	-	2	∞
Z	-	∞	7

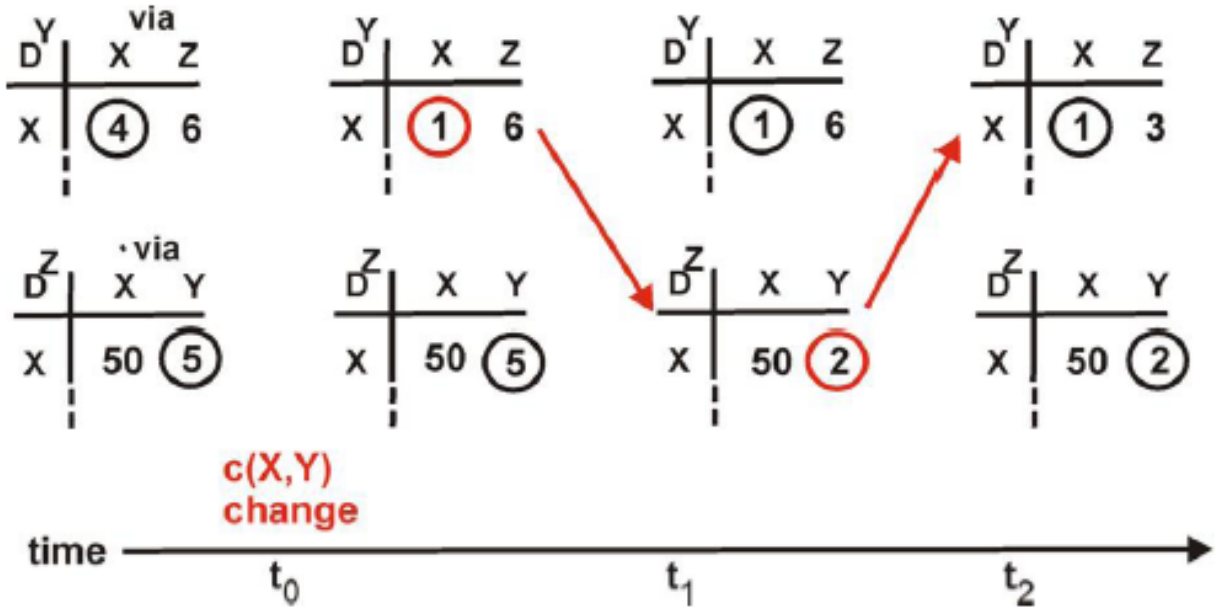




Distance Vector: Algorithmus - Beispiel

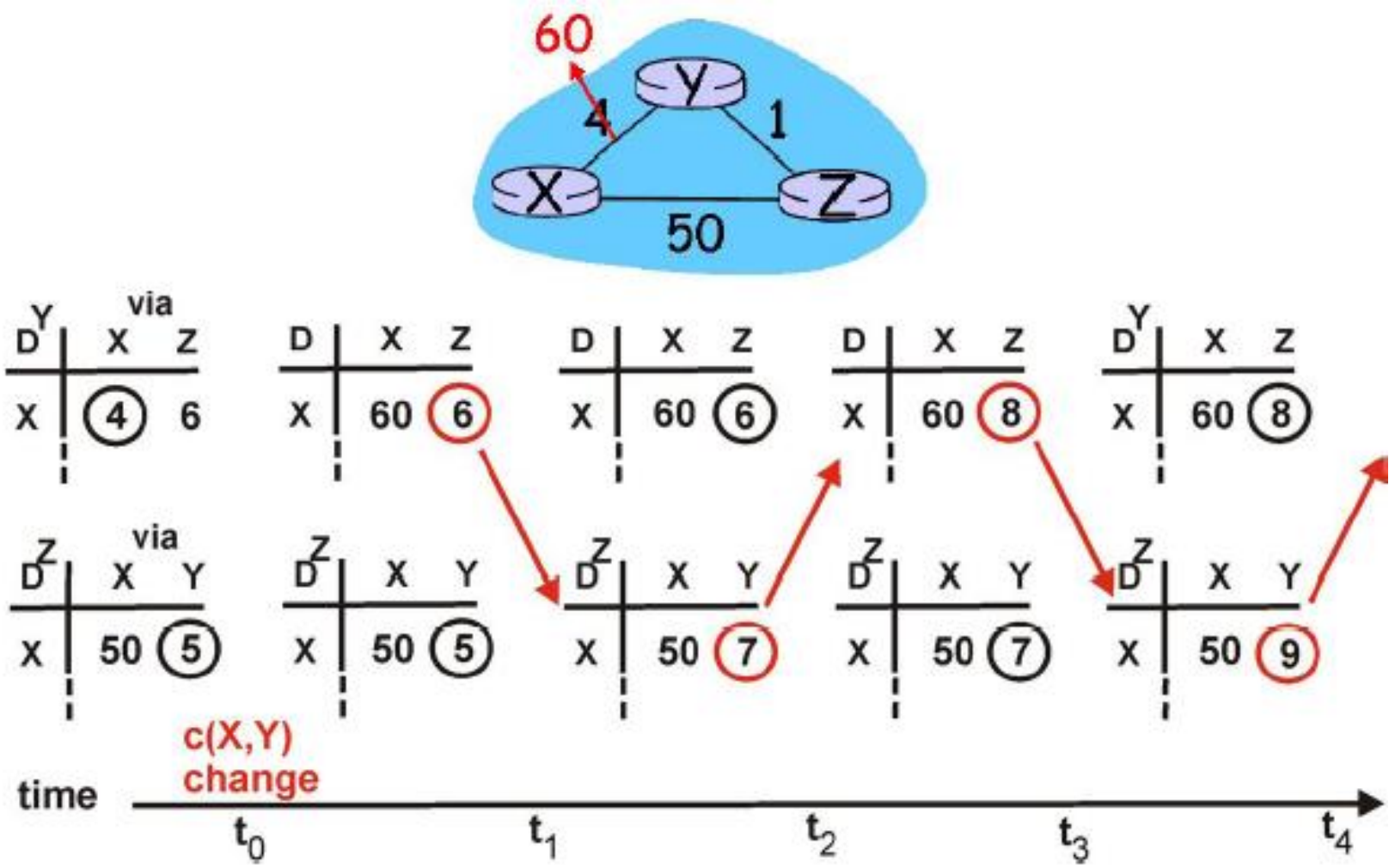
Änderung der Verbindungsleitungskosten





Gute Neuigkeiten verbreiten sich schnell.

Distance Vector: Algorithmus - Beispiel



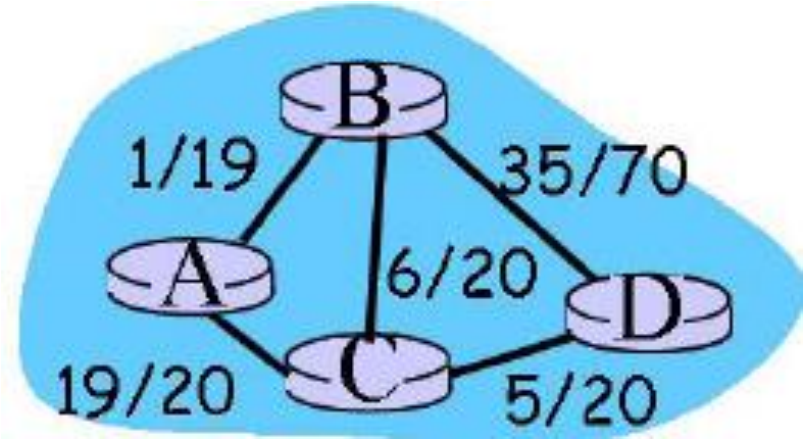
Schlechte Neuigkeiten verbreiten sich langsam.

Leitungsvermittlungsalgorithmen

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

Algorithmen

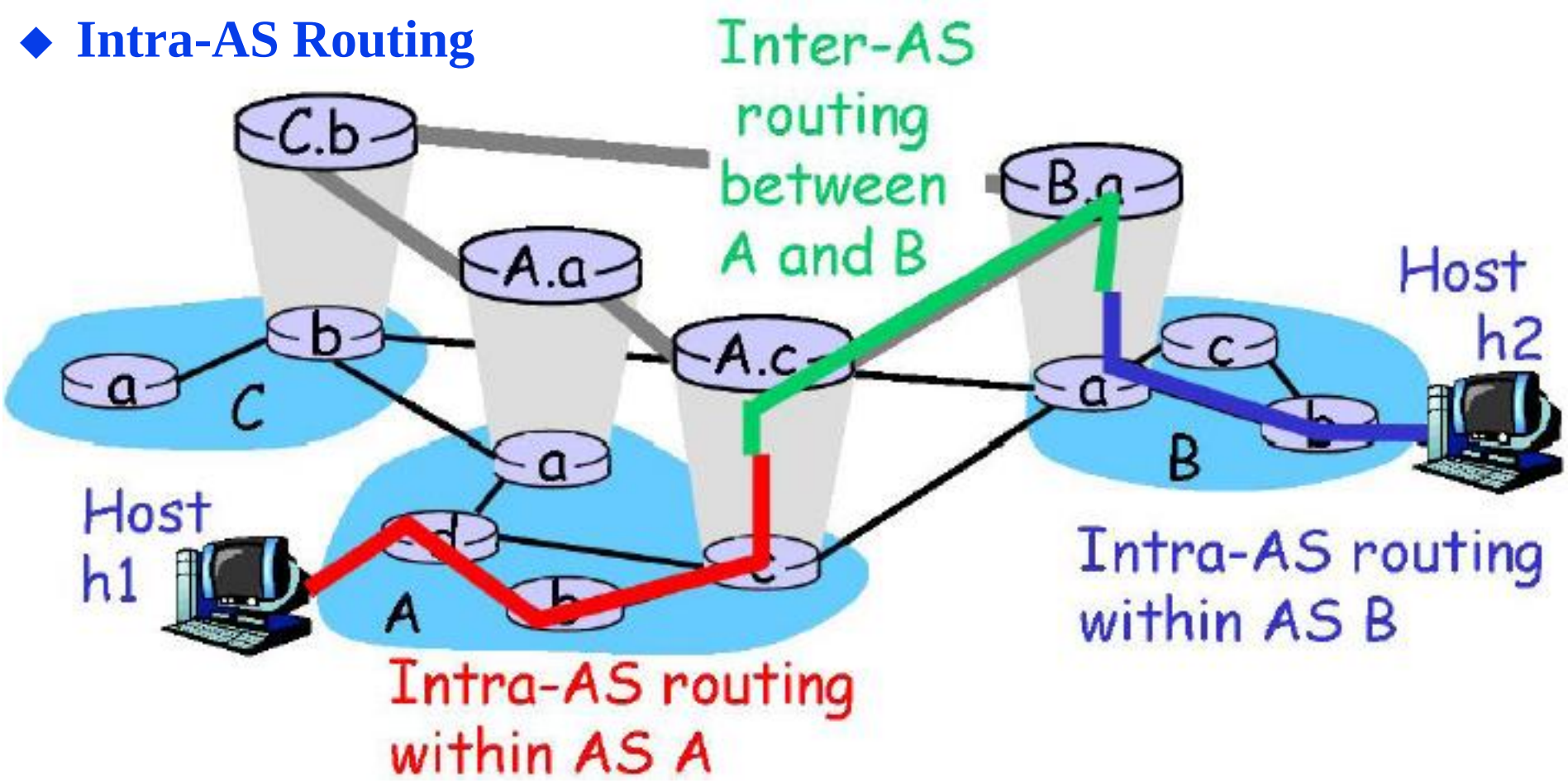
- **Dijkstra** minimale Anzahl Hops (shortest path in terms of hops)
- **Least Loaded Path** (LLP) am wenigsten benutzte Kanäle
- **Maximum Free Circuit** (MFC) jeweils größte Anzahl freier Kanäle



Hierarchisches Routing

Autonome Teilsysteme (AS)

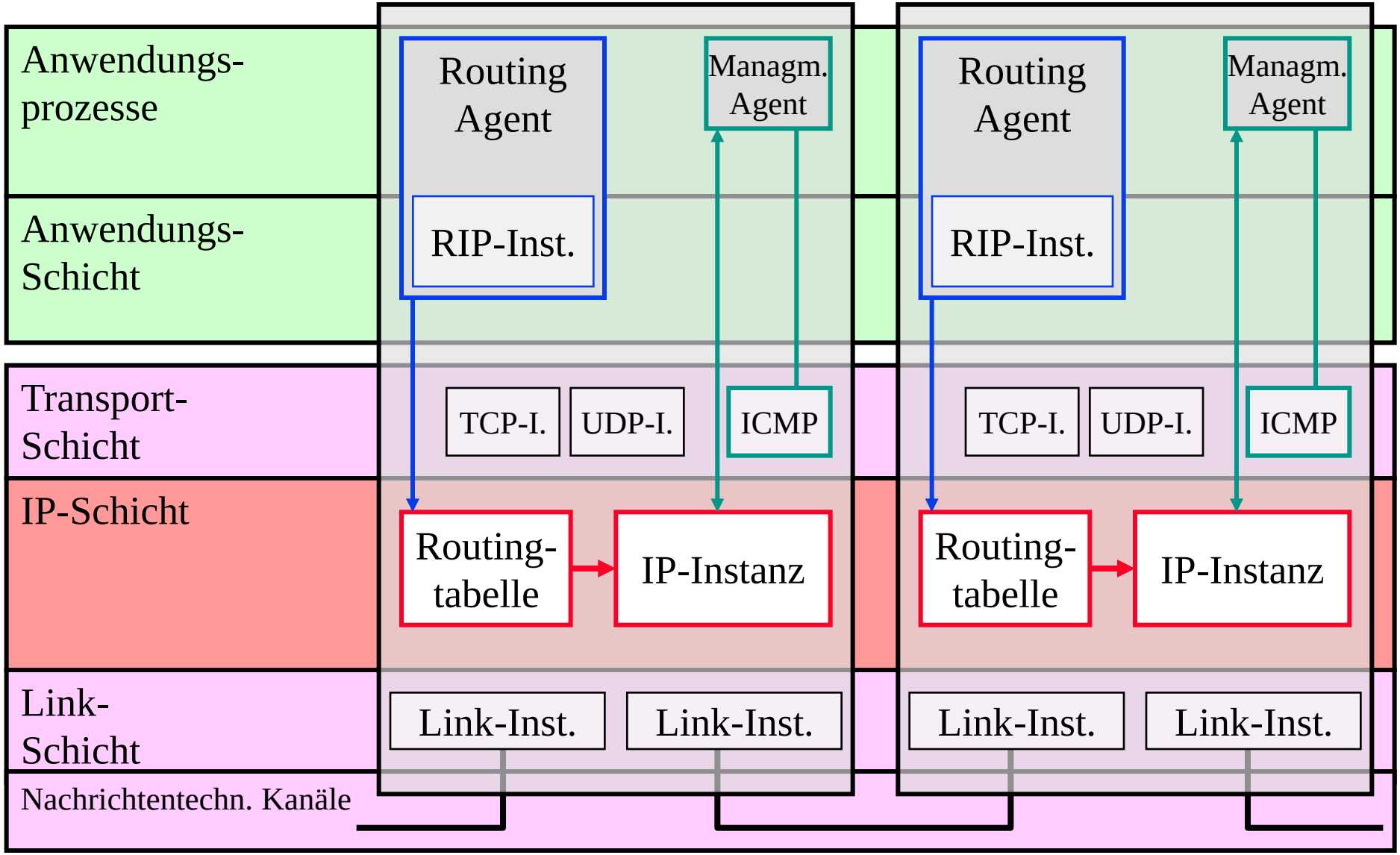
- **Inter-AS Routing**
- **Intra-AS Routing**



Internet-Protokoll: IP V4

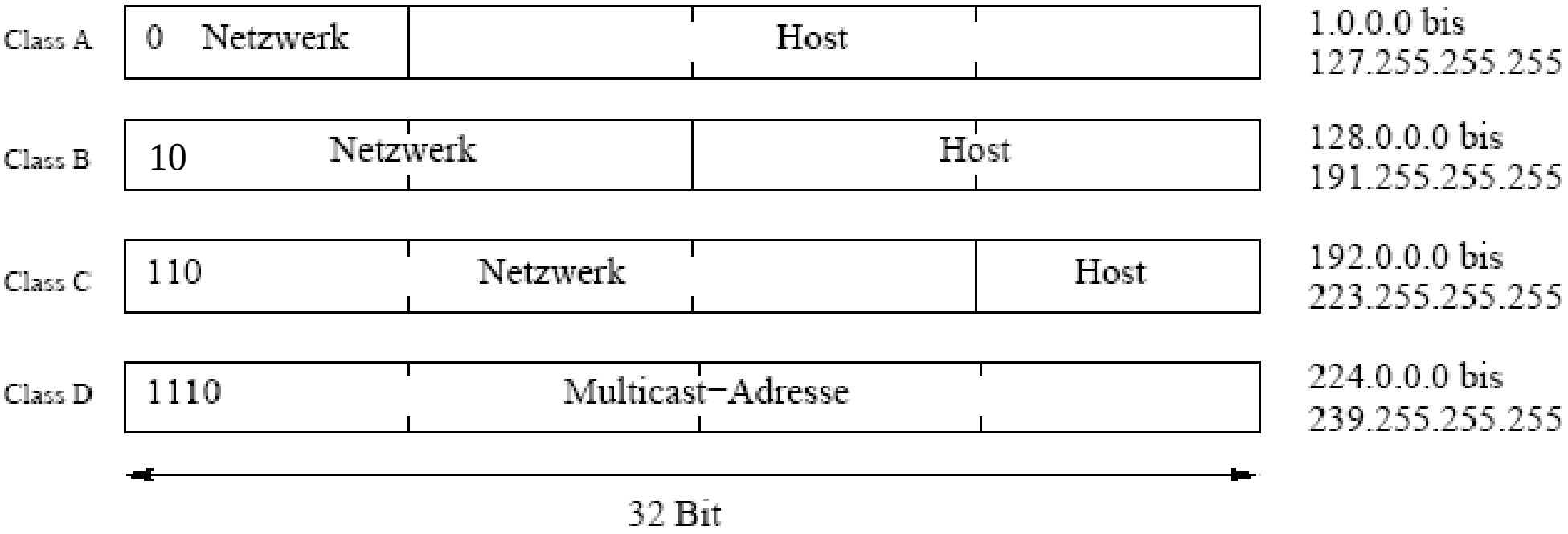
- Verbindungsloser Datagramm–Dienst
- Nachrichten werden im Store-and-Forward-Prinzip von der Quelle zum Ziel weitergeleitet
  - vgl.: Brief oder Postpaket-Transport
- Nachrichten (Nutzdaten, d.h. die PDUs der Transportprotokolle) werden in IP-Pakete (so heißen die PDUs des IP-Protokolls) verpackt.
- 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



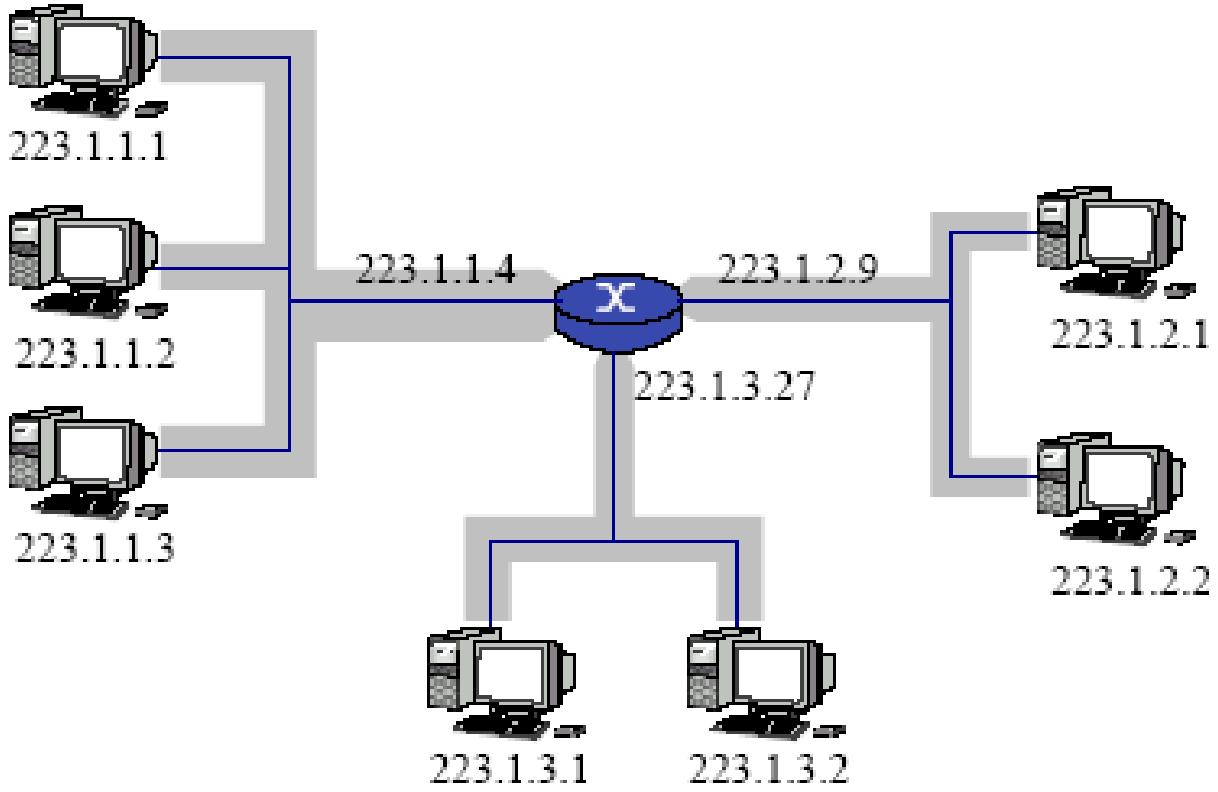
IP v 4: Adressen

- 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



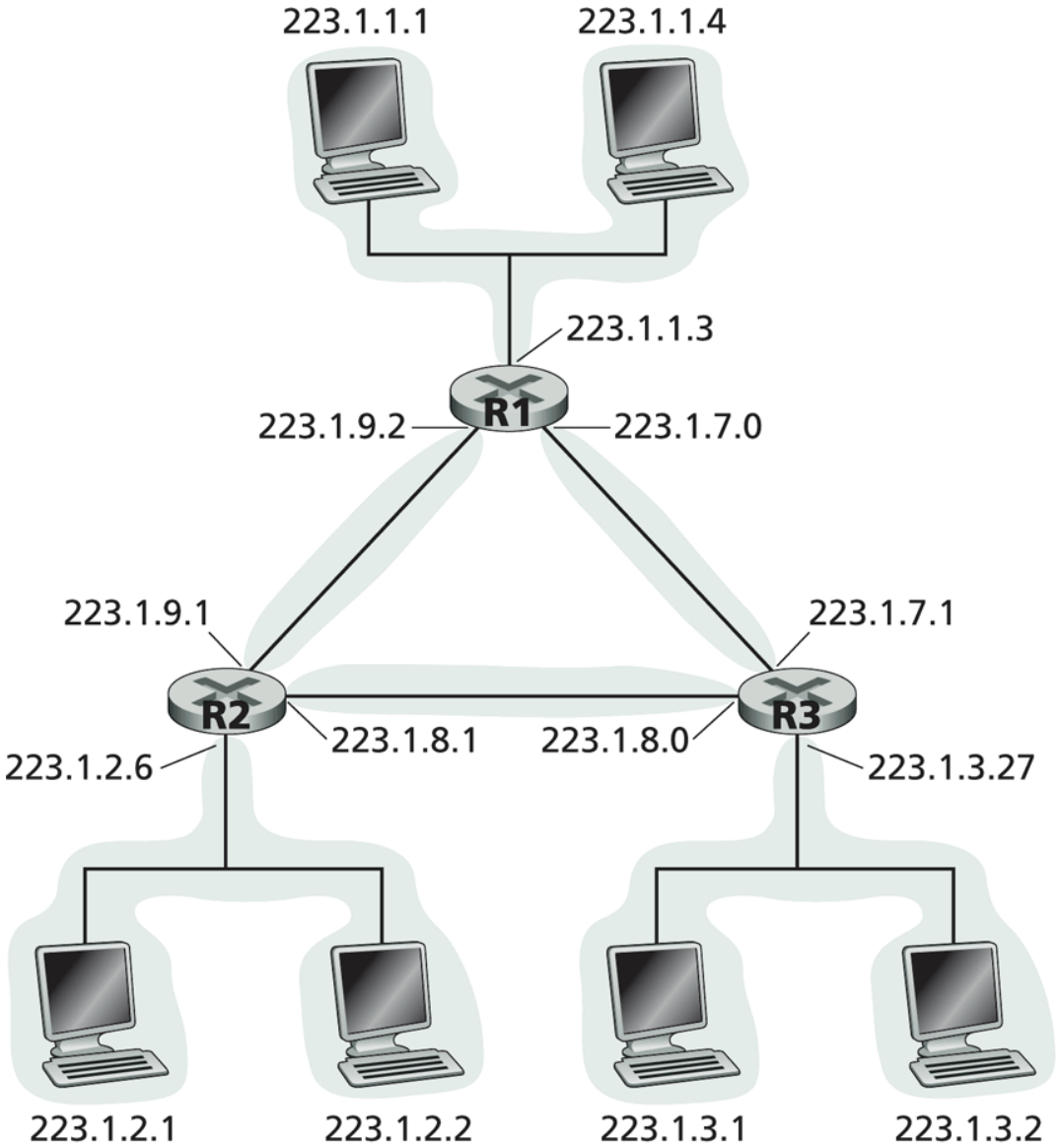
IP: Adressen

- Eine Adresse je Netz-Interface des Knotens



IP: Adressen

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



IP: Interclass Domain Routing

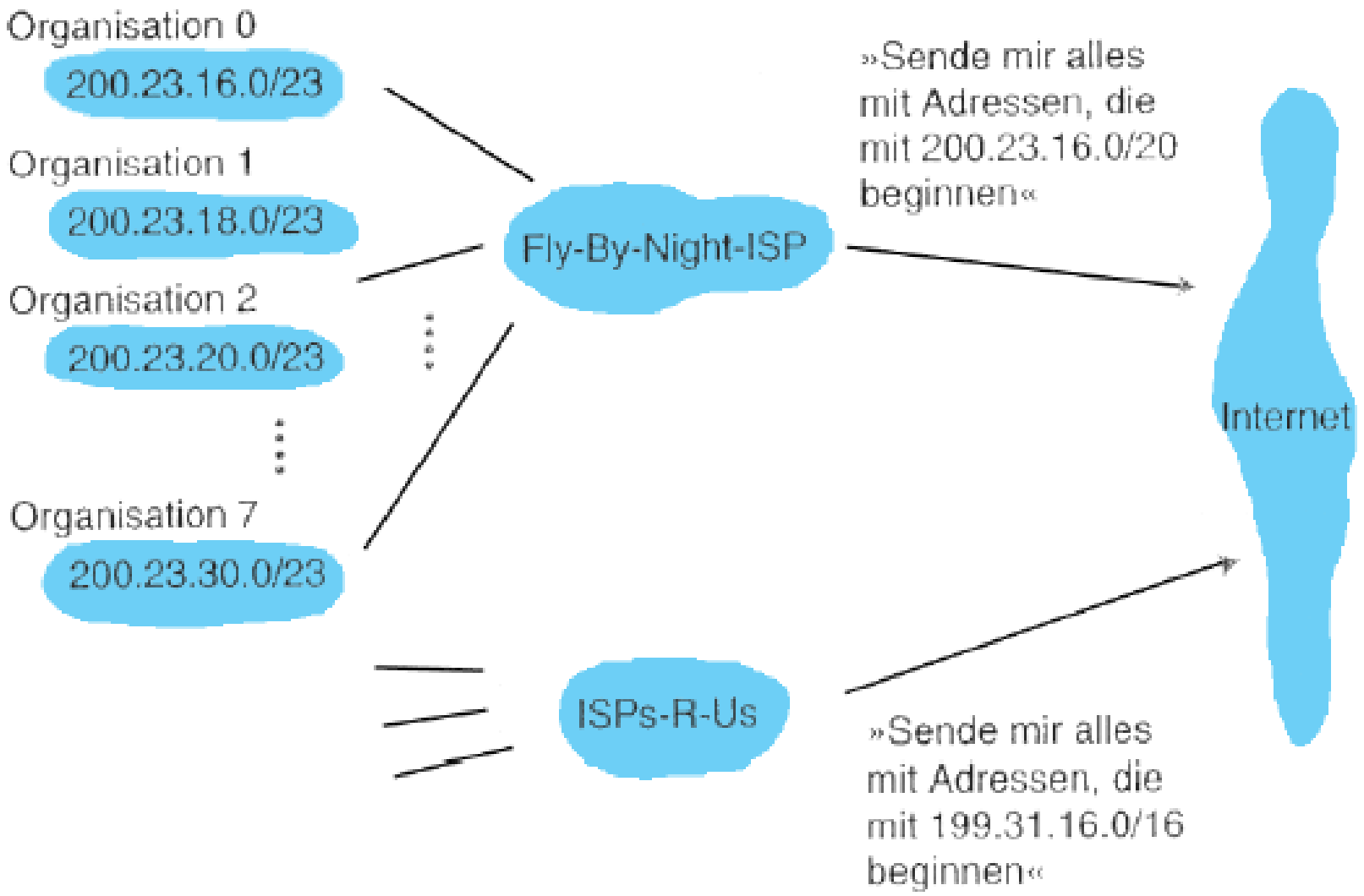
- beliebige Länge der Netzwerkadresse: z.B.: a.b.c.d/21

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
- ...

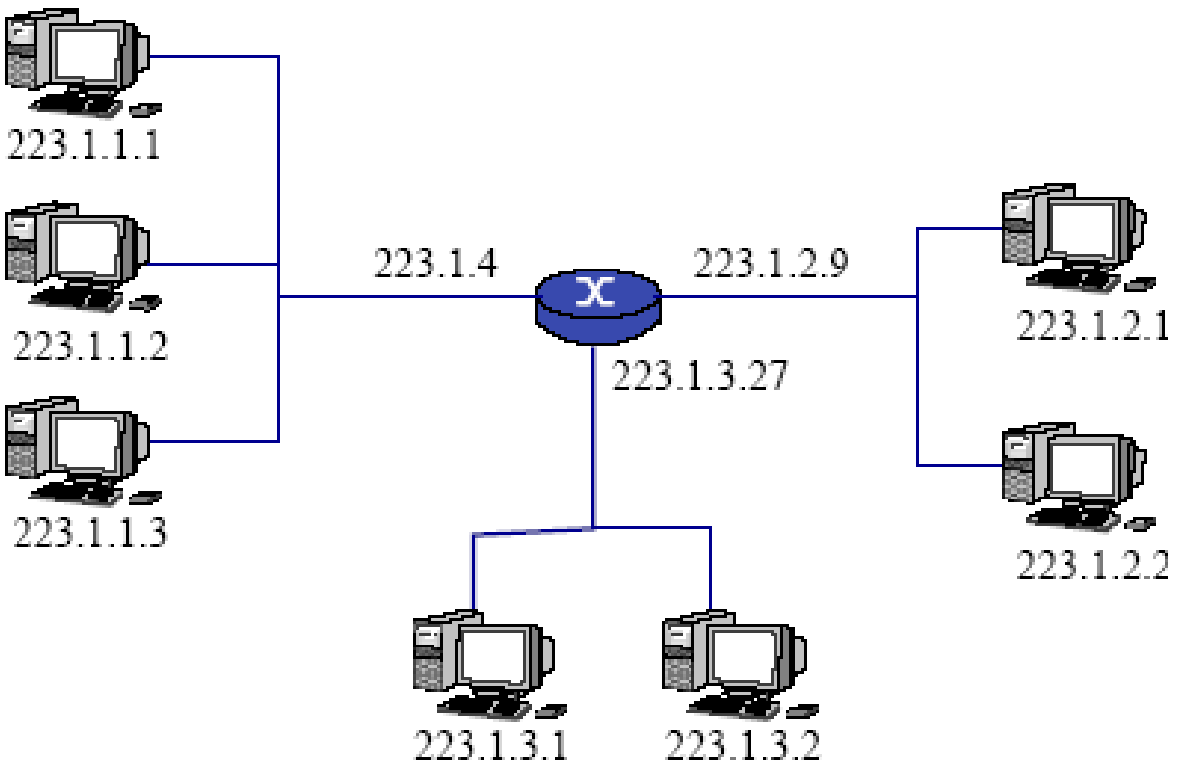
IP: Routingtabellen



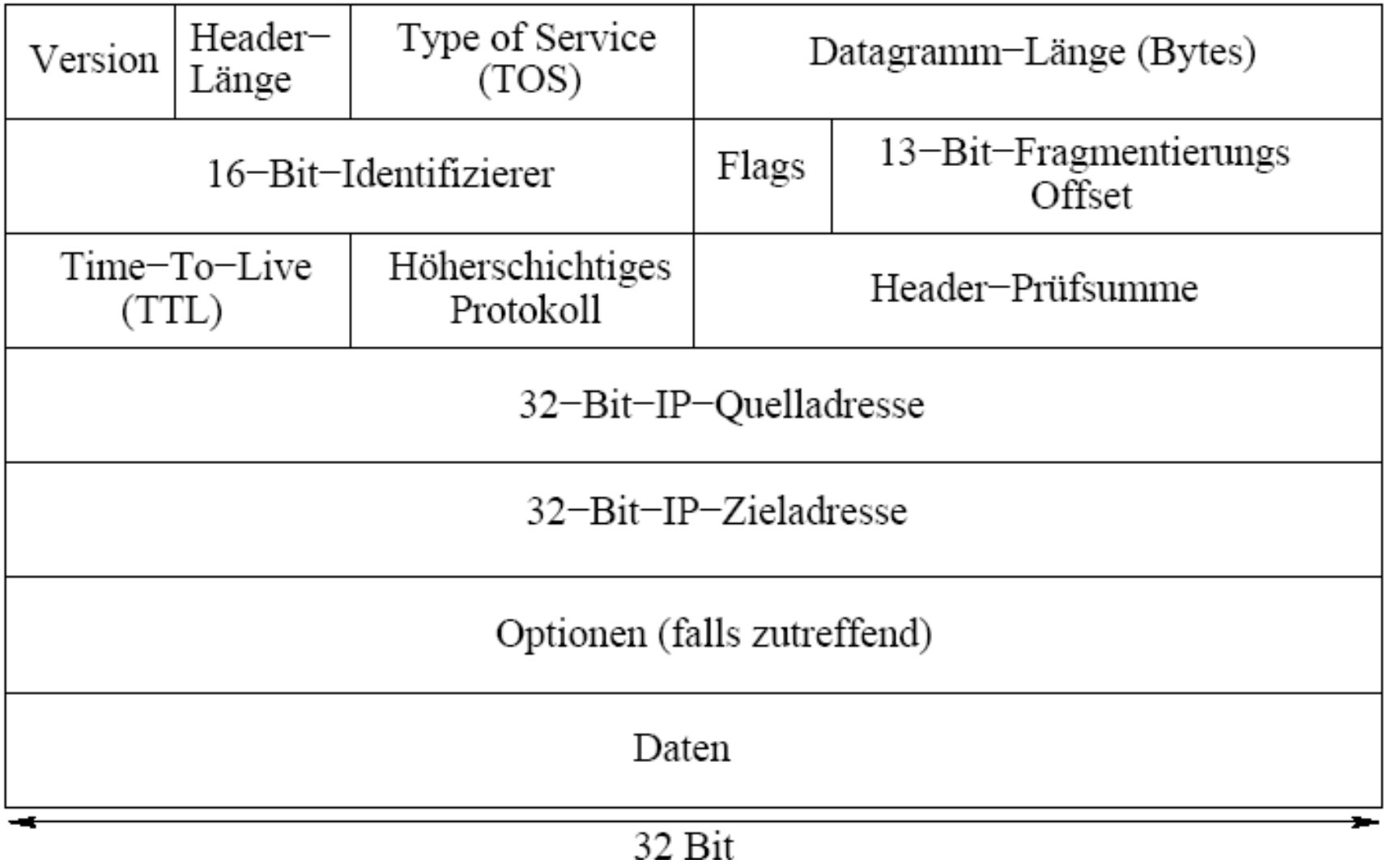
IP: Interdomain-Routing - Routenaggregation

Routing−Tabelle in A

Zielnetzwerk	Nächster Router	Hops
223.1.1.0/24		1
223.1.2.0/24	223.1.1.4	2
223.1.3.0/24	223.1.1.4	2

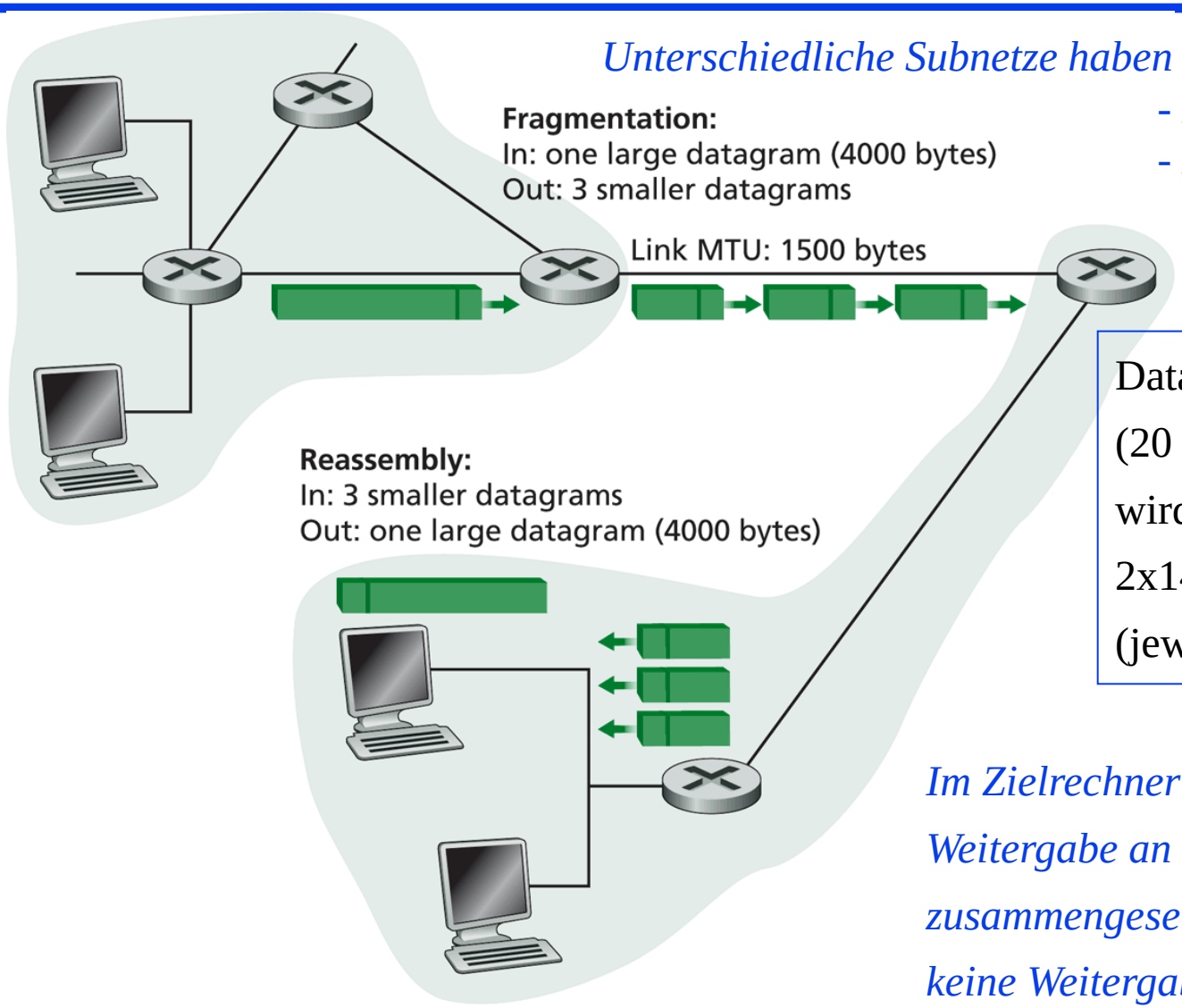


IP: IP v4 - Paketformat



IP: Fragmentierung und Reassemblierung

Unterschiedliche Subnetze haben verschiedene Längenrestriktionen:
- *In Fragmente zerlegen*
- *Fragmente wieder zusammensetzen*



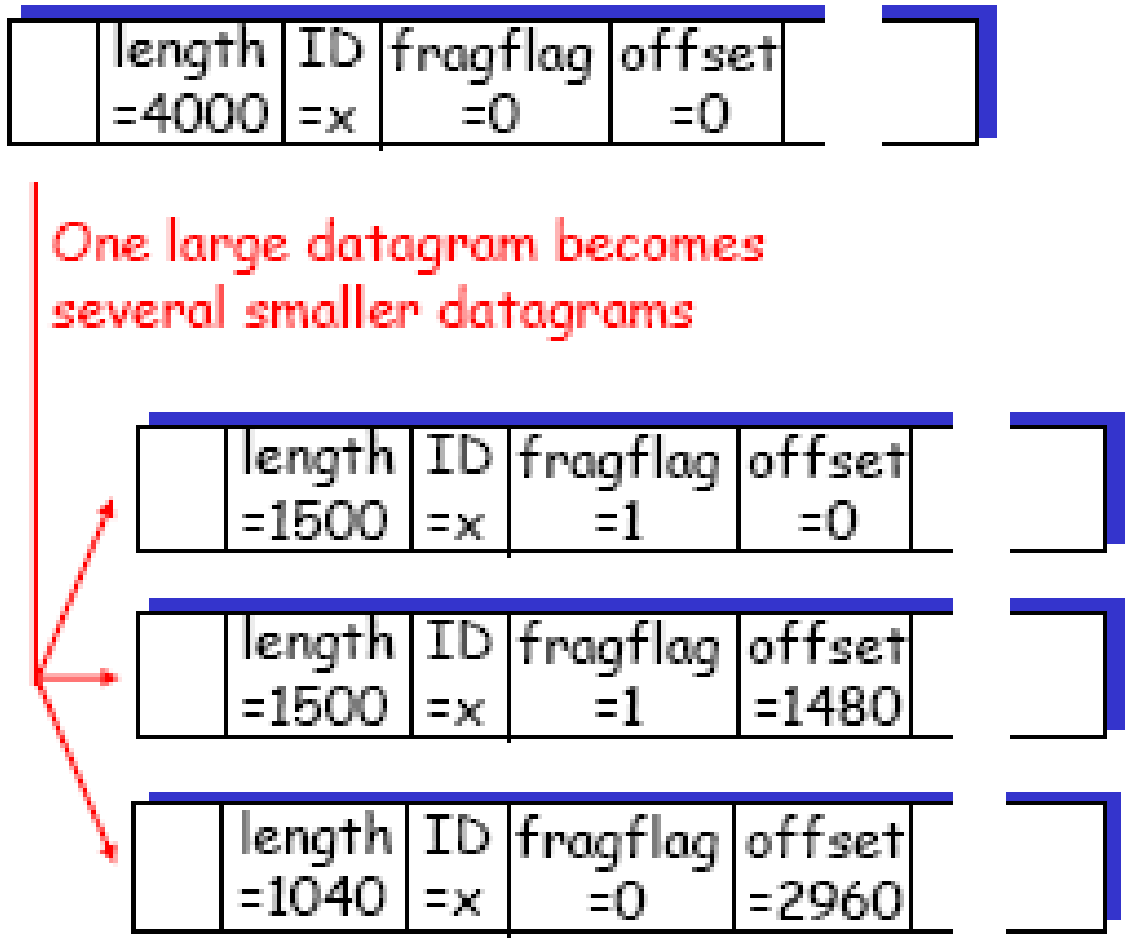
Datagramm 4000 Bytes
(20 Bytes IP-Header +3980 Daten)
wird in 3 IP-Pakete mit
2x1480+1020 Bytes
(jeweils + IP-Header) fragmentiert

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

IP: Fragmentierung und Reassemblierung

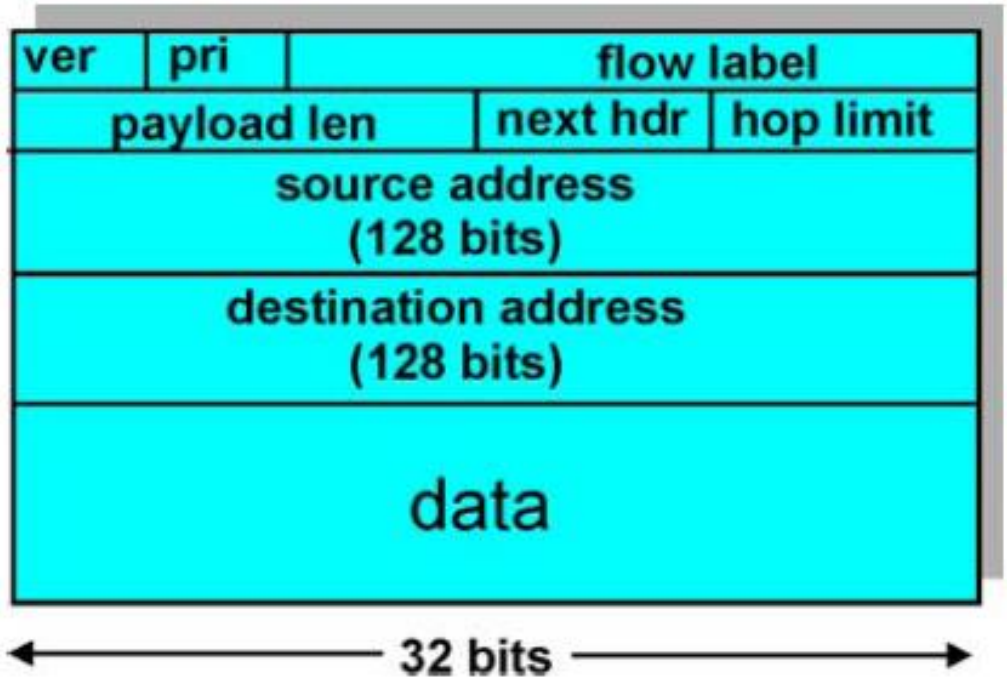
Example

- 4000 byte datagram
- MTU = 1500 bytes



IP v 6: 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
  - Zusatzheader: z.B. zur Verschlüsselung



Priorität

Flusslabel: Id für ausgehandelten Verkehr

Next Hdr: Zusatzheader

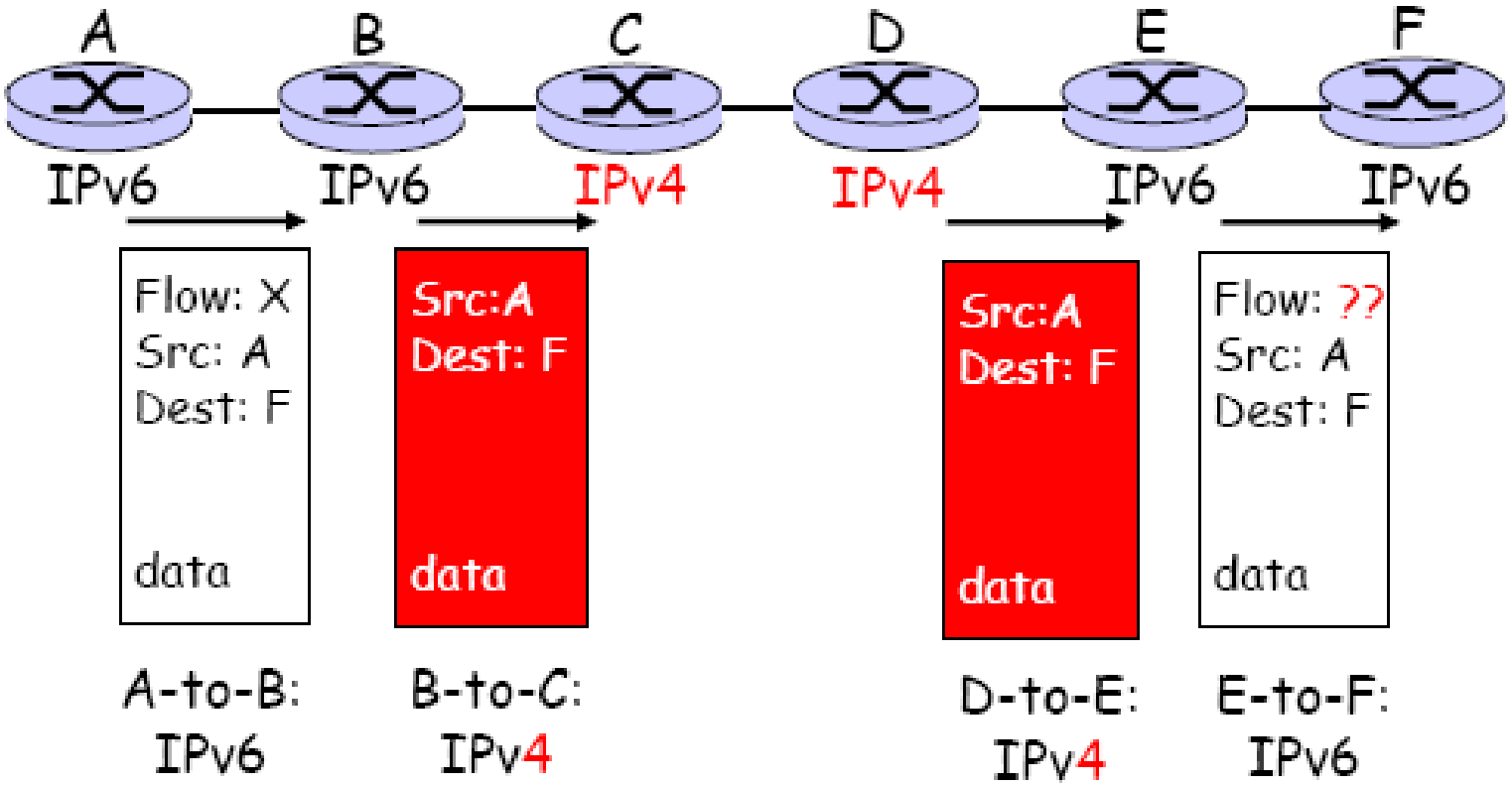
IP v 6: Neue Version des IP

- Weitere Änderungen von IP v6:
  - Keine Prüfsumme in Header (schnelle Verarbeitung)
  - IPsec „IP-Security“
    VPN-Technik

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

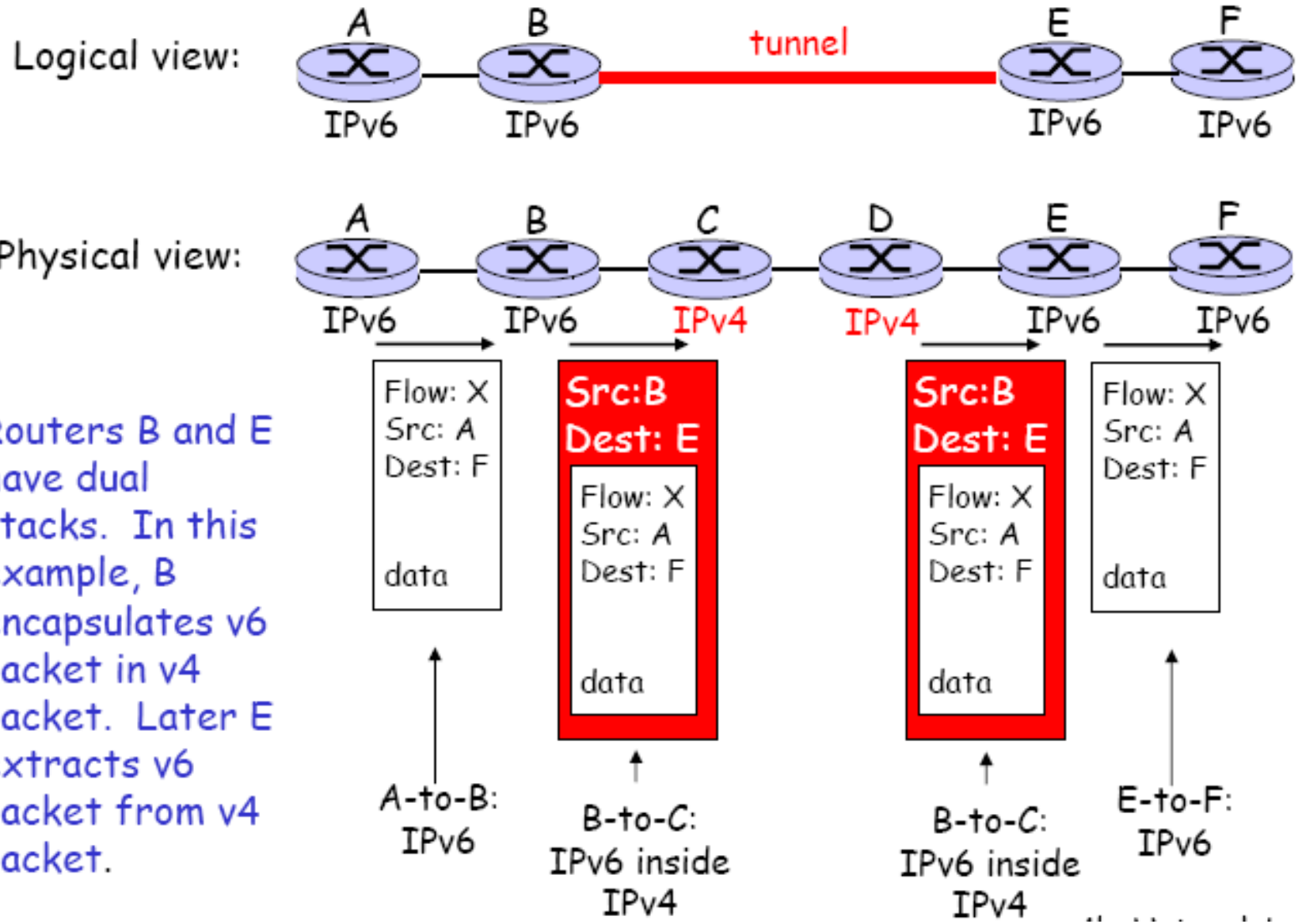
- 2 Möglichkeiten zum gleichzeitigen Betrieb beider Protokolle
  - Dual Stack
    Neue Router können auch IP v4
  - Tunneling
    IP v6 Pakete werden, um IP v4 Netz zu durchlaufen in IP v4 Pakete eingepackt

IP v 6: Dual Stack



- Routers B and E have dual v6 and v4 stacks
- Flow label info lost after translation at B

IP v 6: Tunneling



Routers B and E have dual stacks. In this example, B encapsulates v6 packet in v4 packet. Later E extracts v6 packet from v4 packet.

ICMP Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

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 - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

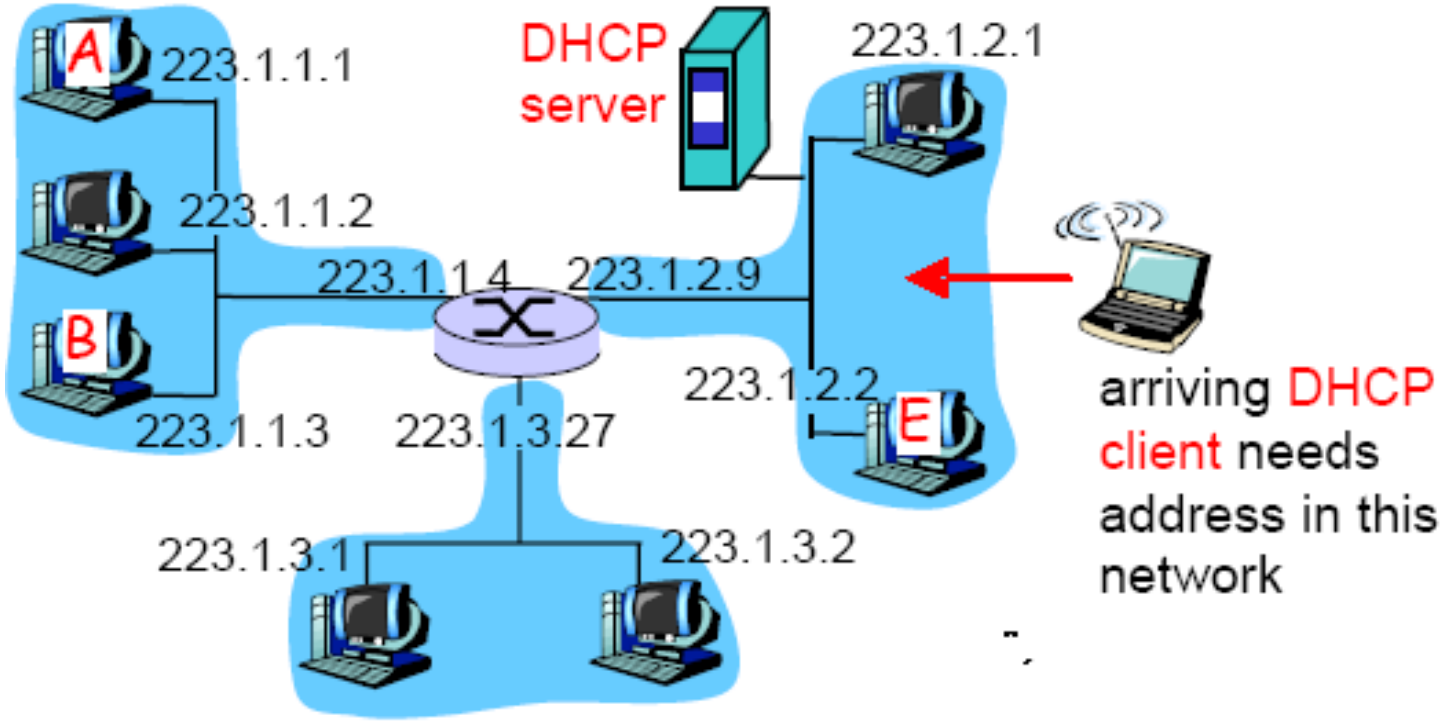
DHCP: Dynamic Host Configuration Protocol

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

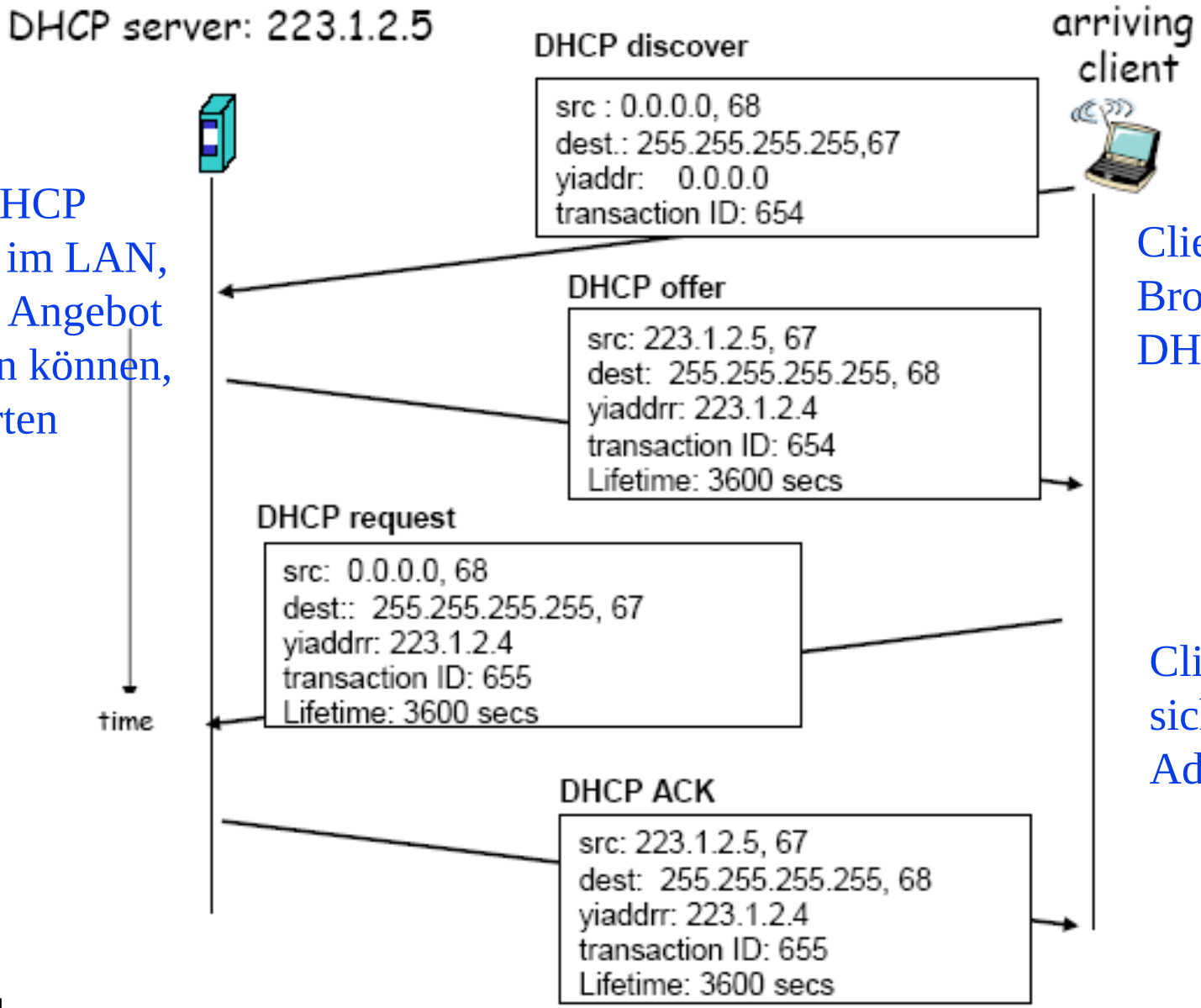
- Dynamische Adresszuweisung

  DHCP Server verteilt Adressen

  Host tritt bei Netzbeitritt als DHCP Client auf



DHCP: Dynamic Host Configuration Protocol



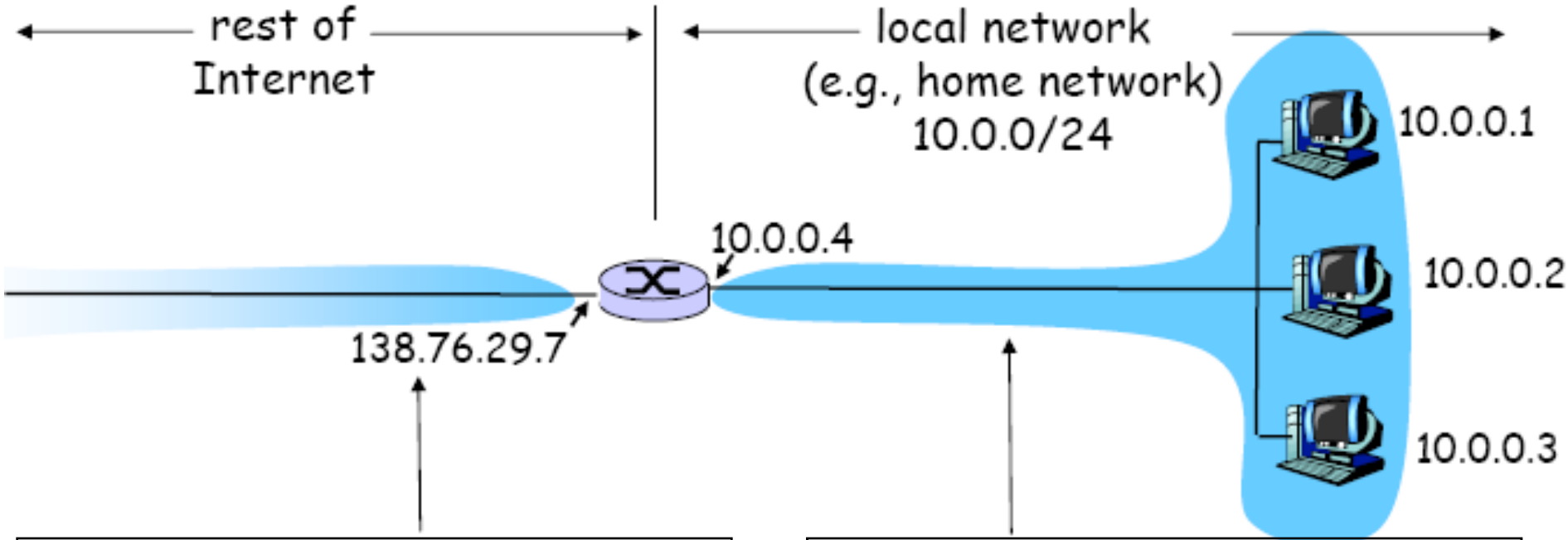
Alle DHCP Server im LAN, die ein Angebot machen können, antworten

Client sucht per Broadcast einen DHCP Server

Client fordert für sich eine IP-Adresse an

NAT: Network Adress Translation

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

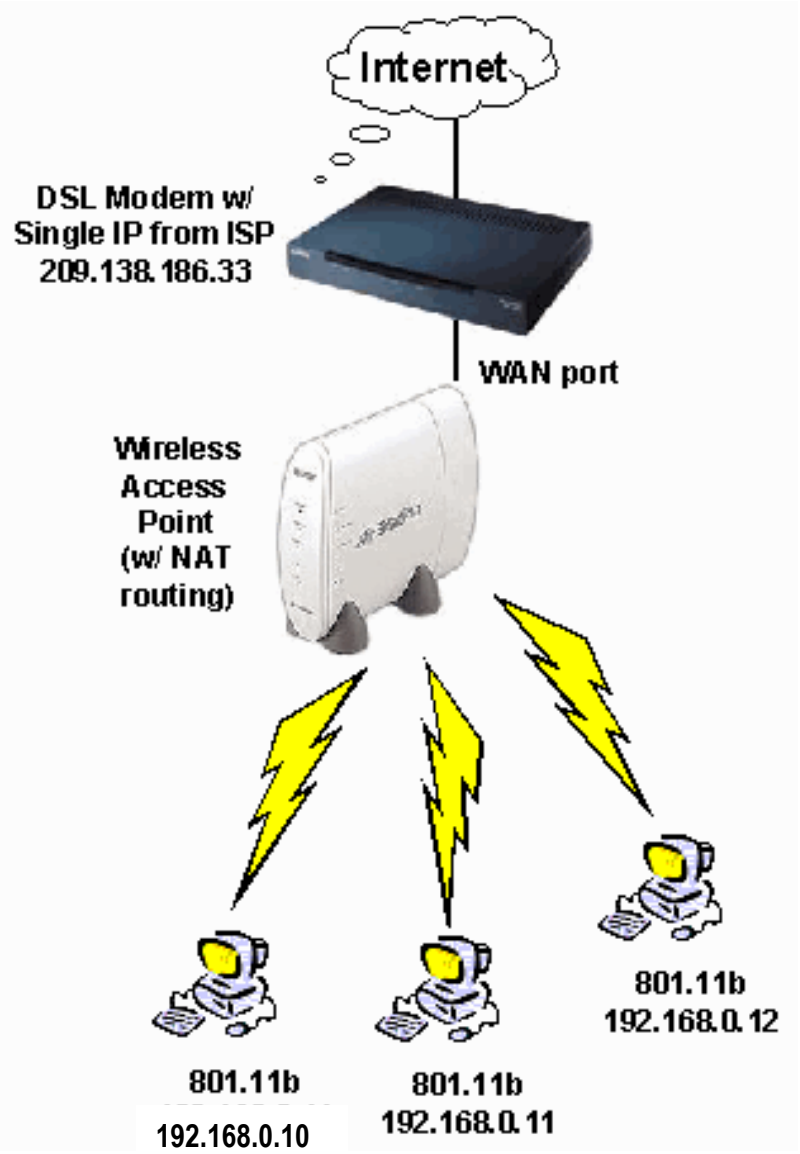


Alle IP-Pakete, die nach außen gehen, haben die Quelladresse 138.76.29.7. Die Port-Nummern werden gespreizt.

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

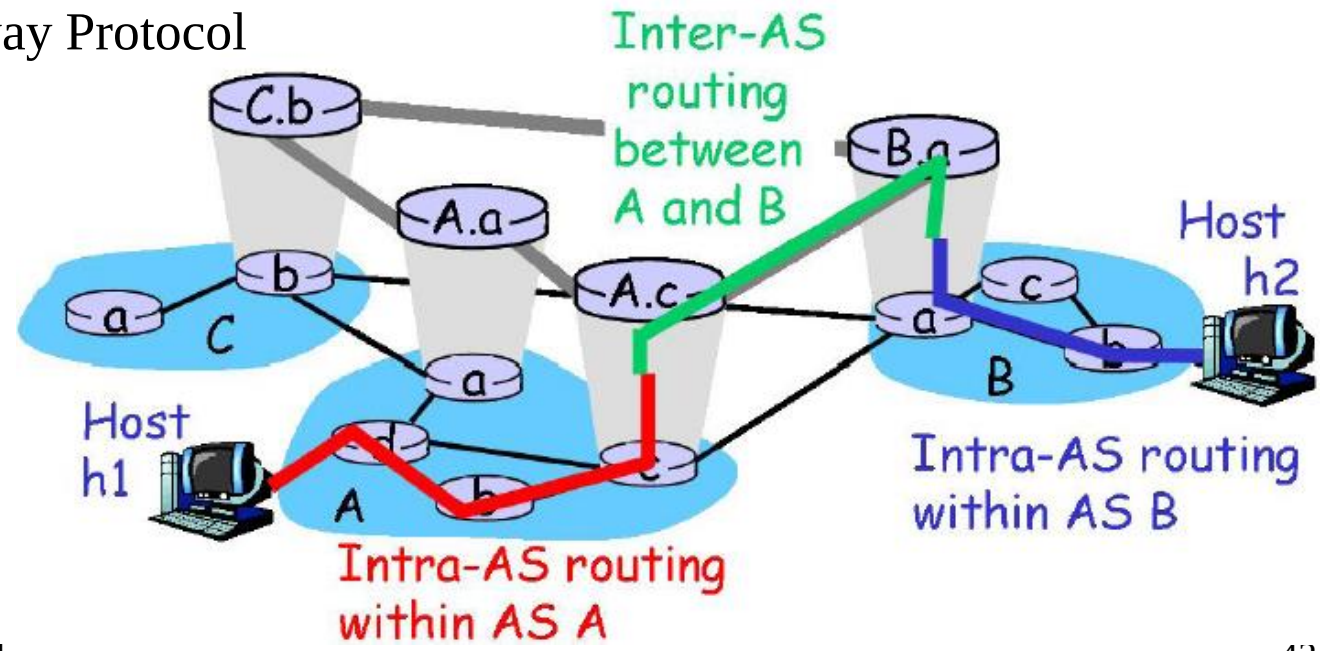
NAT: Network Adress Translation

- 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!

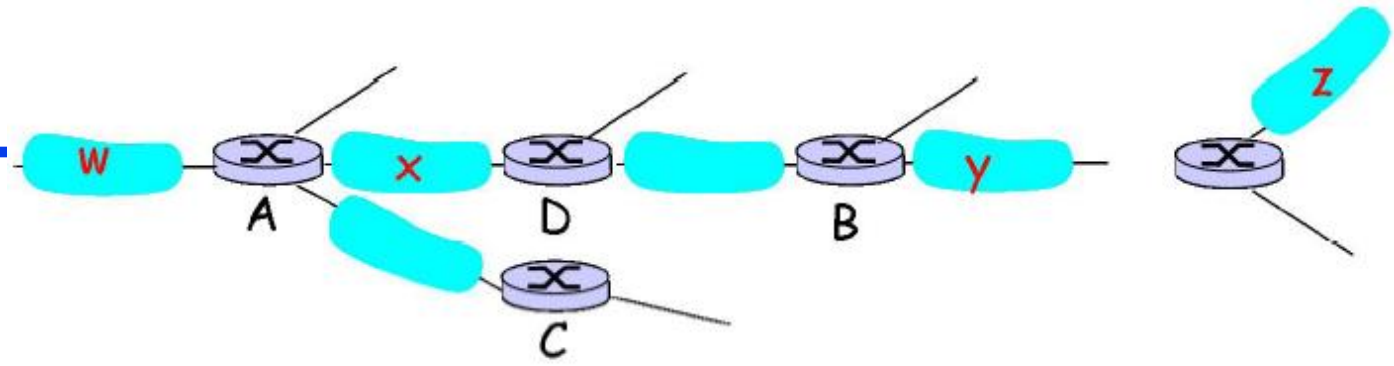


Internet: Routingtabellen-Pflege

- 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



A] RIP

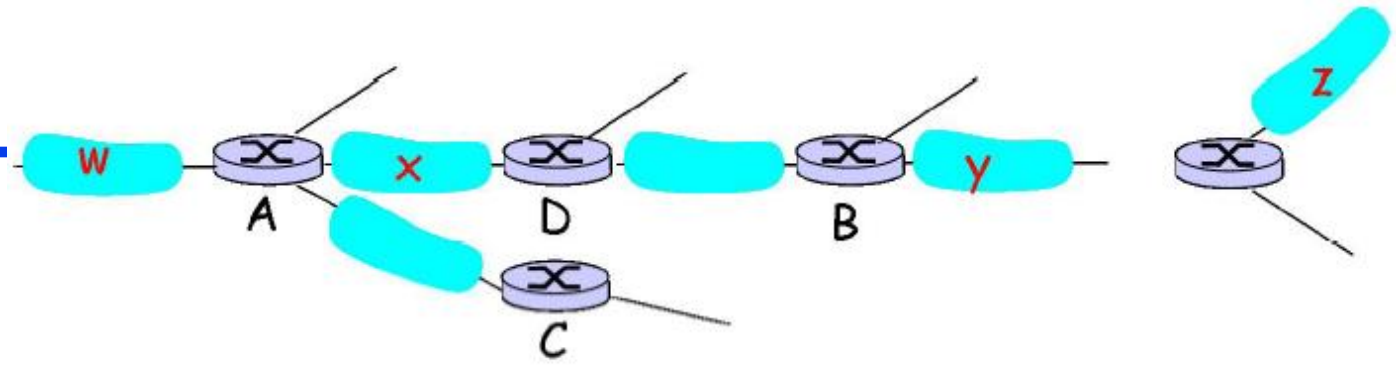


Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

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

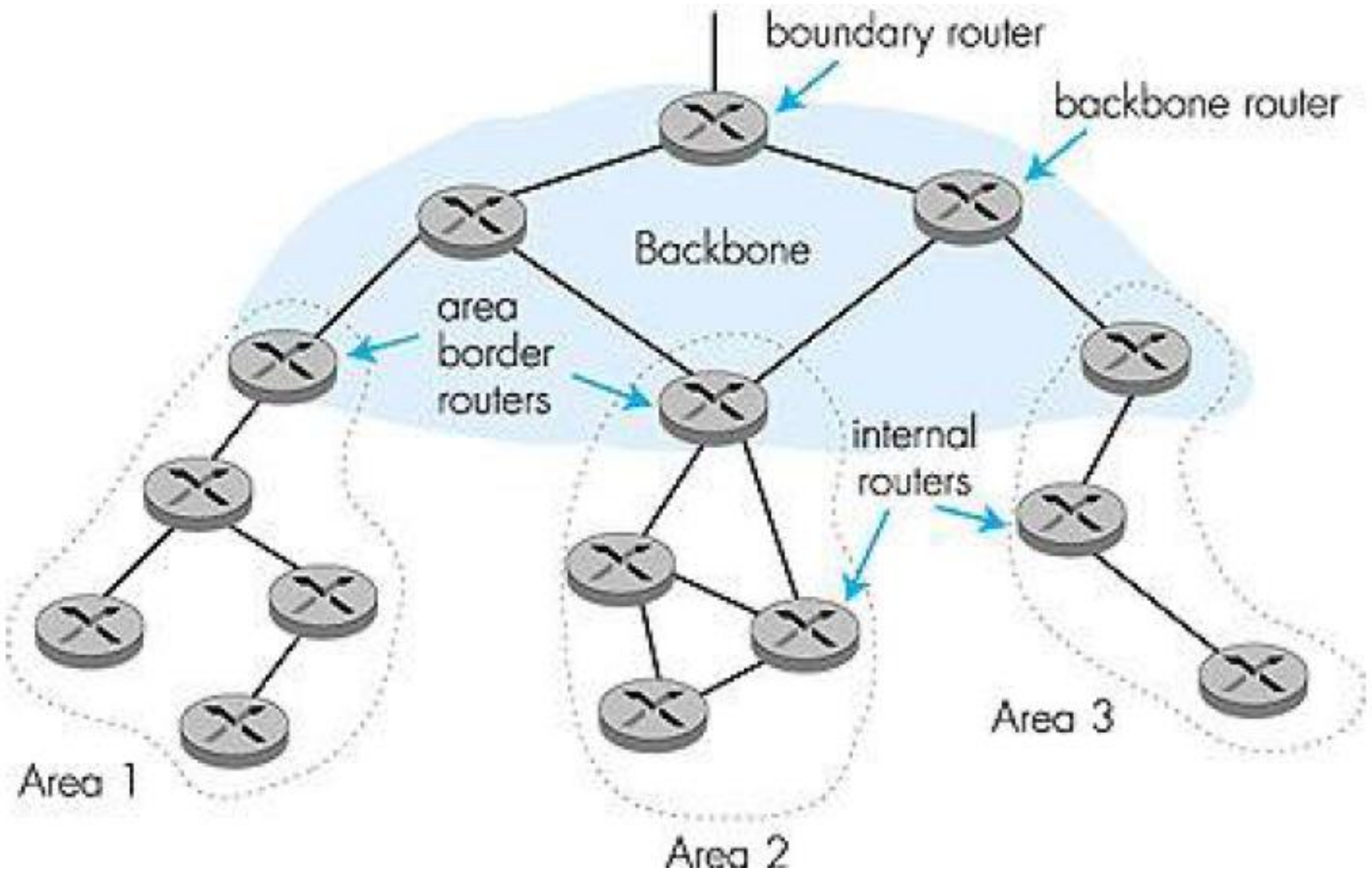
Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	….

Routing table in D

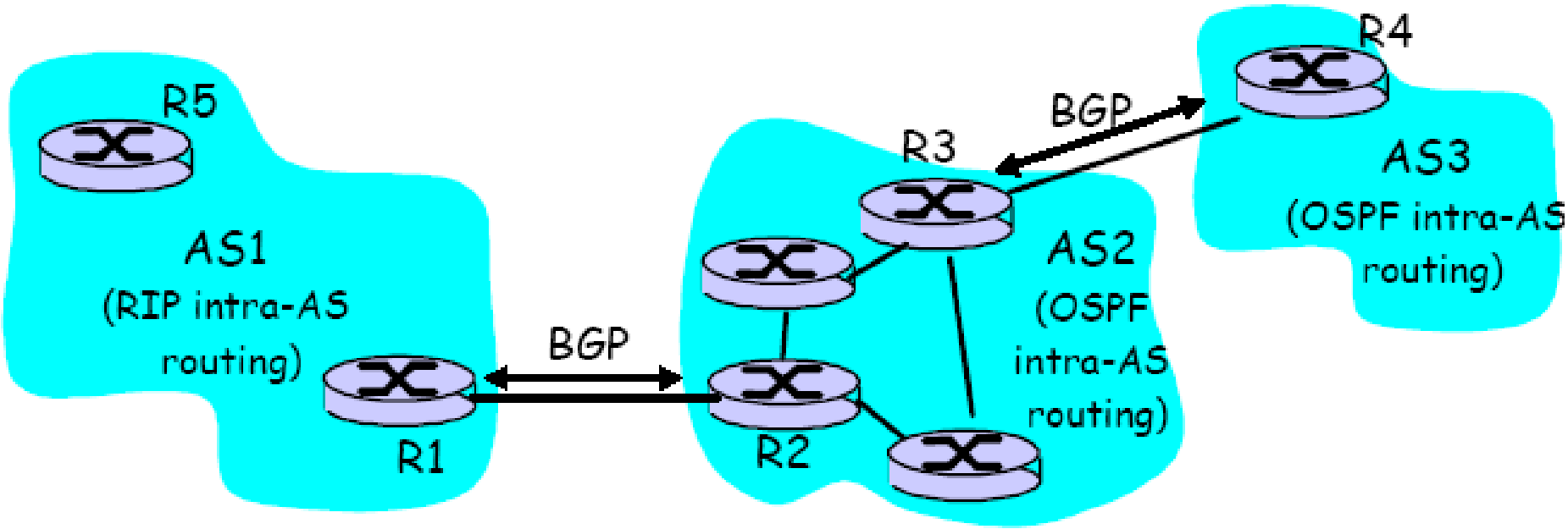
A] OSPF

- 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

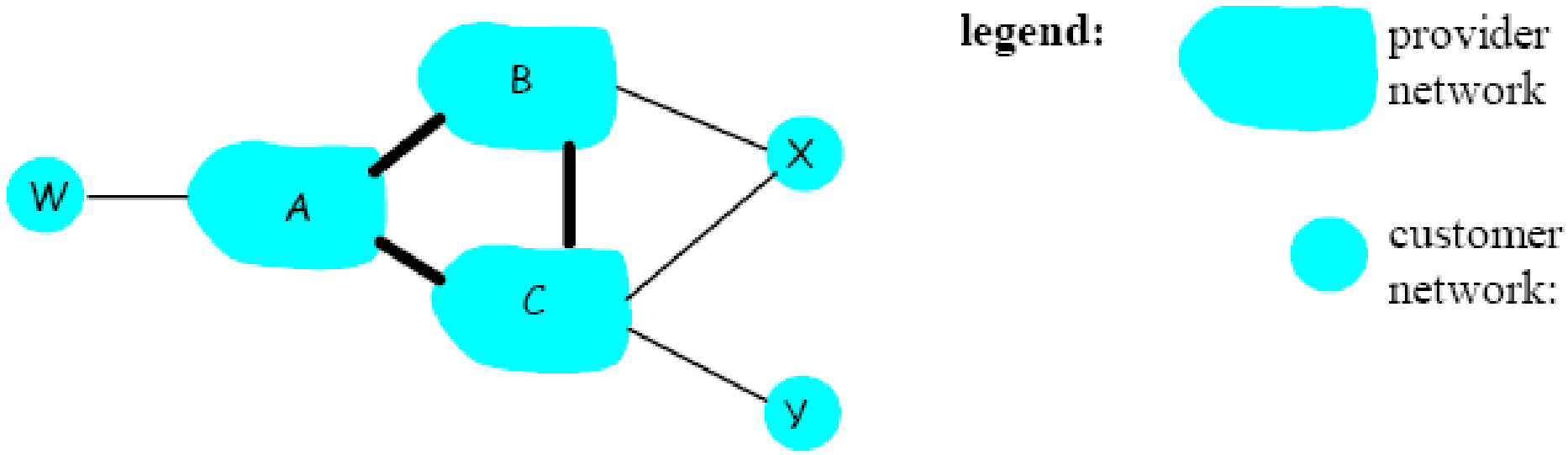


B] Inter-AS Routing im Internet: BGP



- **BGP (Border Gateway Protocol):** the de facto standard
- **Path Vector** protocol:
  - similar to Distance Vector protocol
  - each Border Gateway broadcasts to neighbors (peers) entire path (i.e, sequence of ASs) to destination: *Advertisements and Withdrawels*
    E.g., Gateway X may send its path to dest. Z: **Path (X,Z) = X,Y1,Y2,Y3,…,Z**

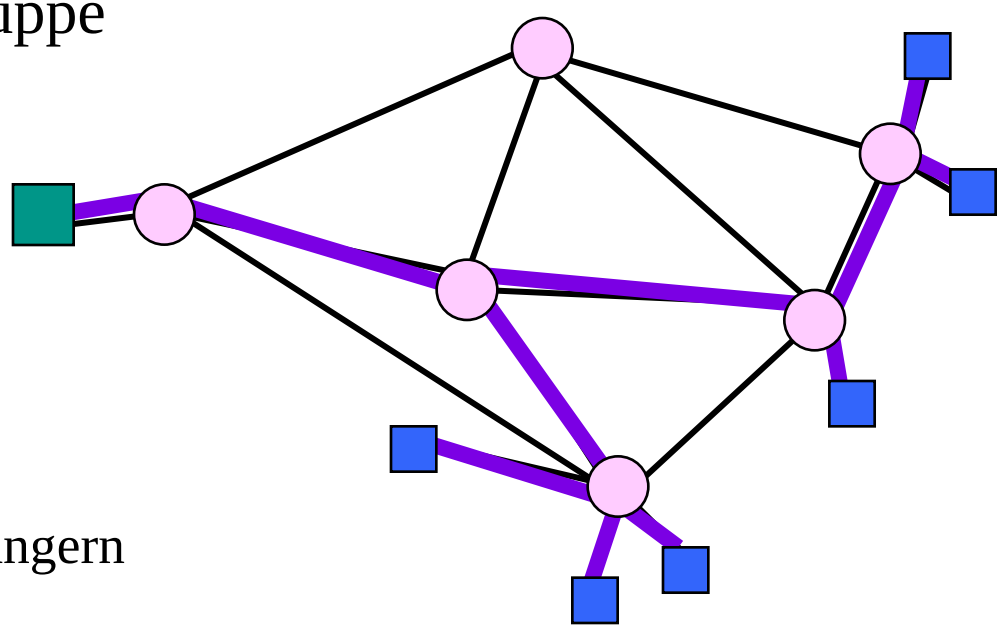
Inter-AS Routing im Internet: BGP



- A,B,C are provider networks
- X,W,Y are customers (of provider networks)
- X is dual-homed: attached to two networks
  - X does not want to route from B via X to C
  - .. so X will not advertise to B a route to C

Multicast-Routing

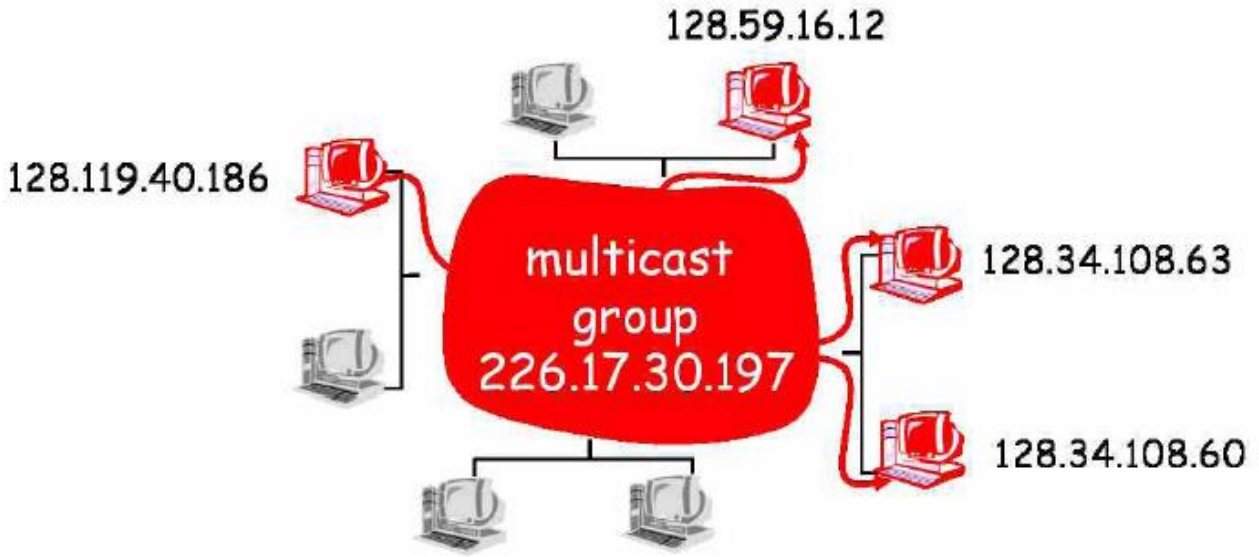
- Gruppen-Kommuikation: Senden an alle Mitglieder einer Empfänger-Gruppe
  - für Software-Verteilung
  - für Konferenz-Übertragung
  - für Telekonferenzen

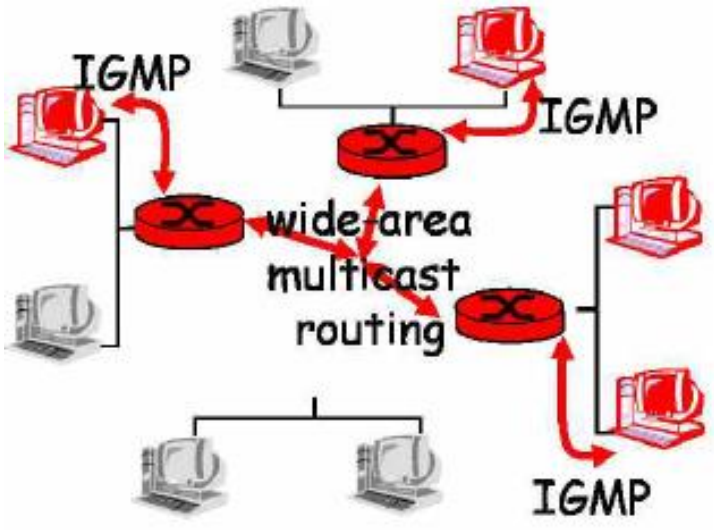


- Grundidee:
  - Spannbaum vom Sender zu Empfängern bestimmen
  - Router als Baum-Zwischenknoten duplizieren die Pakete
- Großes Problem: Zuverlässiger Broadcast so genannte Quittungsimplosion

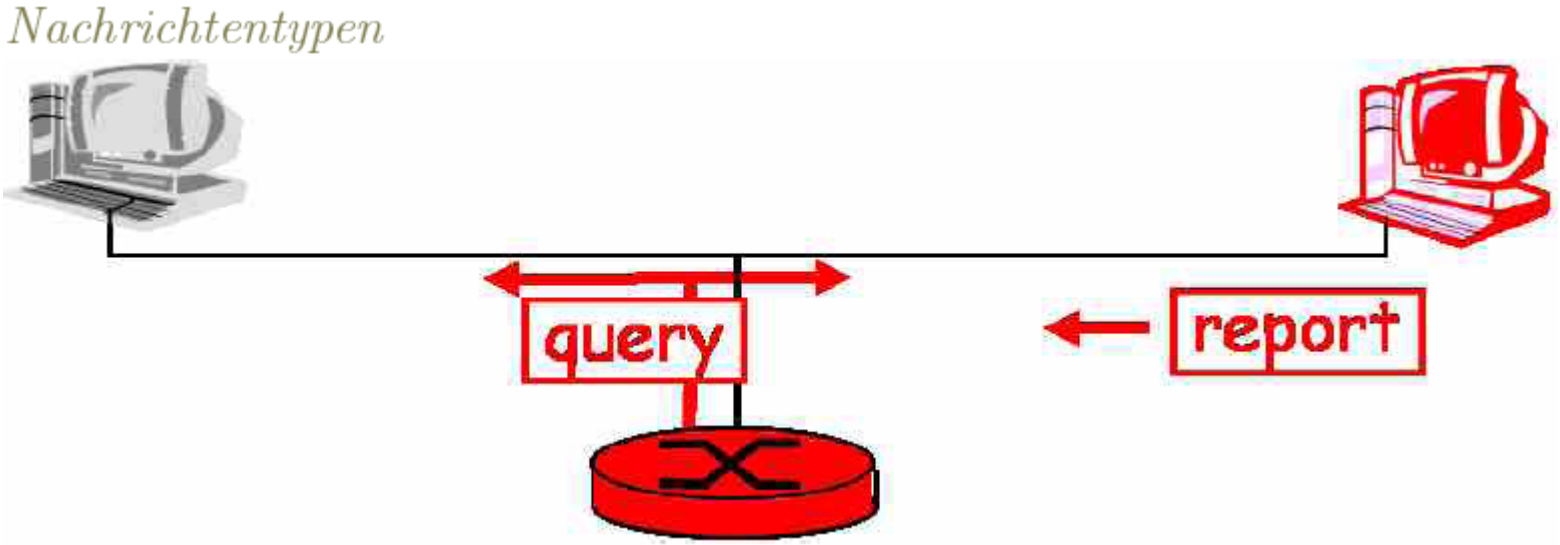
Multicast-Routing: IP-Gruppen-Adressen

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





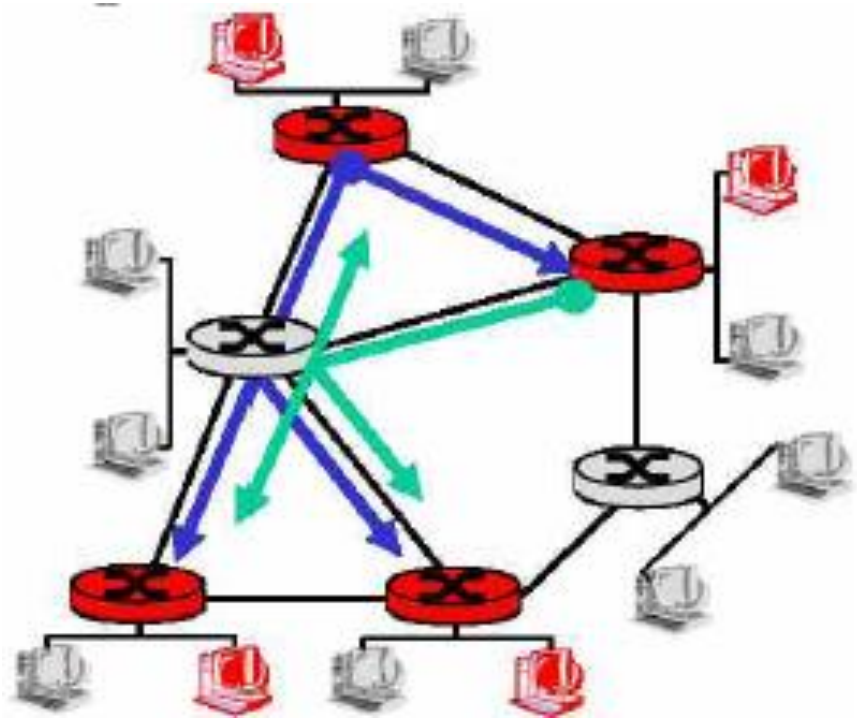
Multicast-Routing: IGMP



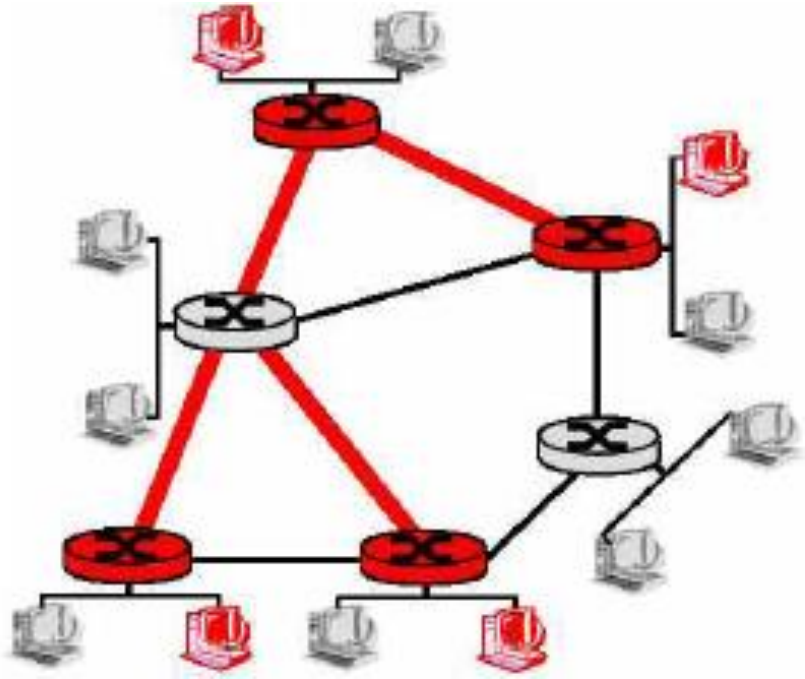
- a) Router frägt „seine“ Hosts, ob sie an Gruppe beteiligt sind
- b) Host sagt seinem Router, dass er an Gruppe beteiligt sein möchte

Multicast-Routing: Wide Area Multicast Routing

A) Je Sender ein Multicast-Baum



B) Gemeinsamer Gruppen-Baum



Multicast-Routing: Wide Area Multicast Routing

- **DVMRP** (Distance Vector Multicast Routing Protocol
  - source-based tree
- **MOSPF** (Multicast Open Shortest Path First)
  - source-based tree
- **CBT** (Core-Based Trees)
- **PIM** (Protocol Independent Multicast)
  - Sparse Mode (similar in spirit to CBT)
  - Dense Mode (similar in spirit to DVMRP)
  - no assumption about underlying unicast routing protocol

Mobile Netze

Home Network
Heimatnetz der mobilen Station (e.g., 128.119.40/24)

Home Agent
Einheit, die Funktionen für die mobile Station ausführt, wenn dieses nicht im Heimnetz ist

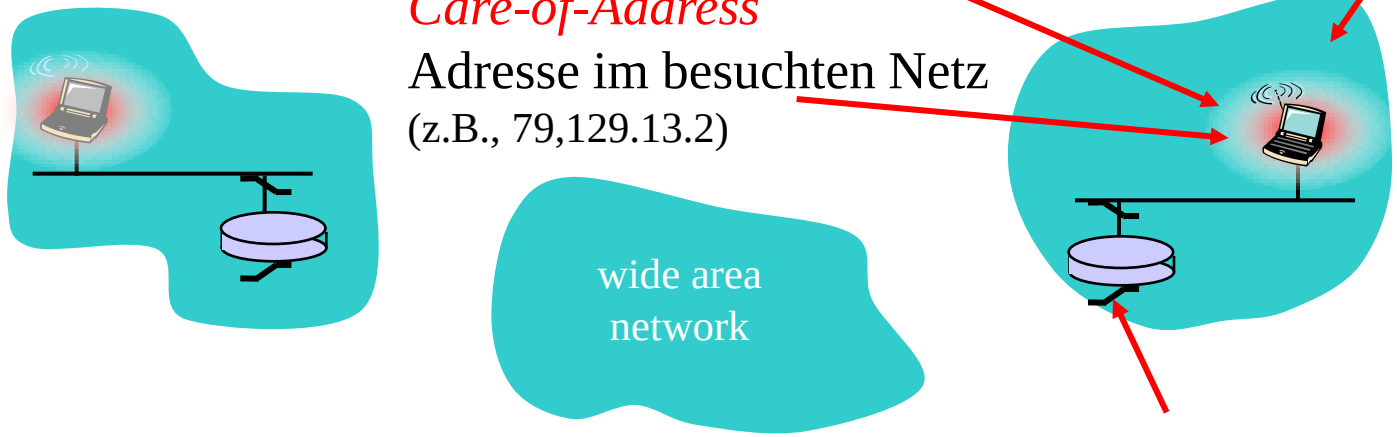
Permanent Address
Adresse im Heimnetz, unter der die mobile Station immer erreichbar sein soll, z.B., 128.119.40.186

Mobile Netze

Permanent Address
bleibt unverändert (z.B., 128.119.40.186)

Visited Network
Netz, in dem sich die mobile Station z.Z. befindet (z.B., 79.129.13/24)

Care-of-Address
Adresse im besuchten Netz (z.B., 79,129.13.2)



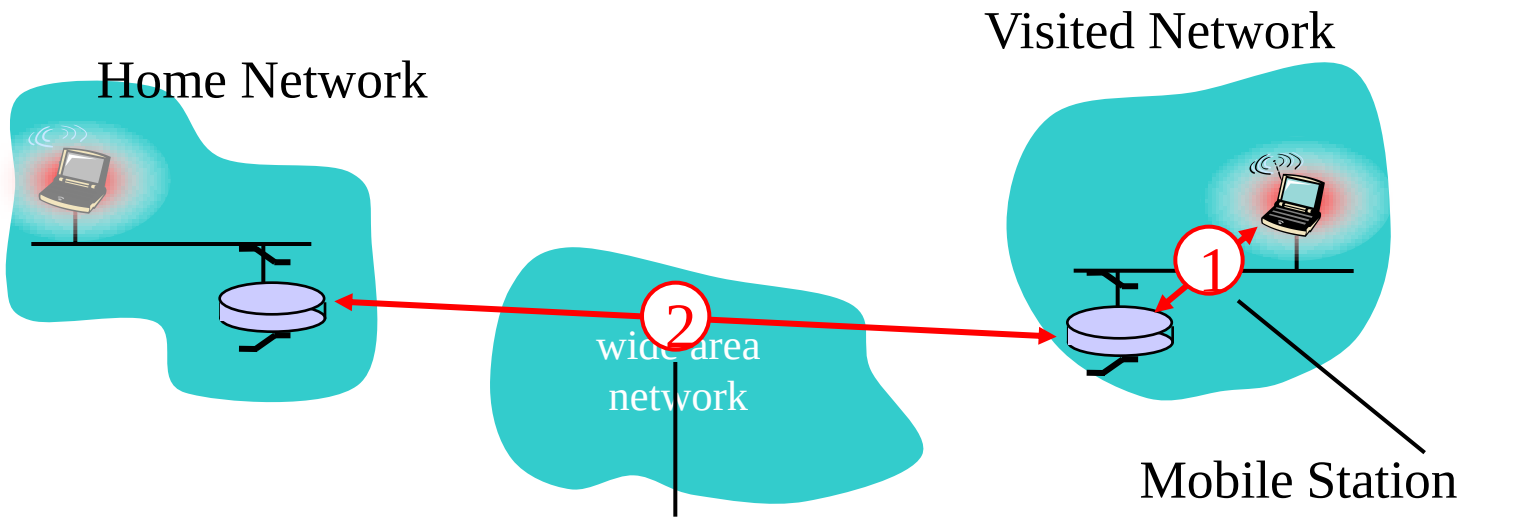
Foreign Agent
Instanz, die mobile Station im fremden Netz unterstützt

Correspondent
Möchte mit der mobilen Station in Verbindung treten

Mobile Netze

- Problem: Wo ist mobile Station momentan?
- Lösung A:
  Routing Algorithmus soll das Problem lösen: Router verteilen den Aufenthaltsort mobiler Stationen mit dem üblichen Austausch von Routing-Informationen,
  - Routing-Tabellen zeigen an, wo sich eine mobile Station befindet
  - Keine Änderungen am Endsystem notwendig
- Lösung B:
  Endsysteme sollen das Problem lösen, Routing bleibt unverändert:
  - indirektes Routing: Kommunikation für die mobile Station läuft über den Home Agent, der die Pakete weiterleitet
  - Direktes Routing: Kommunikationspartner erhält vom Home Agent die aktuelle Adresse der mobilen Station und kommuniziert direkt

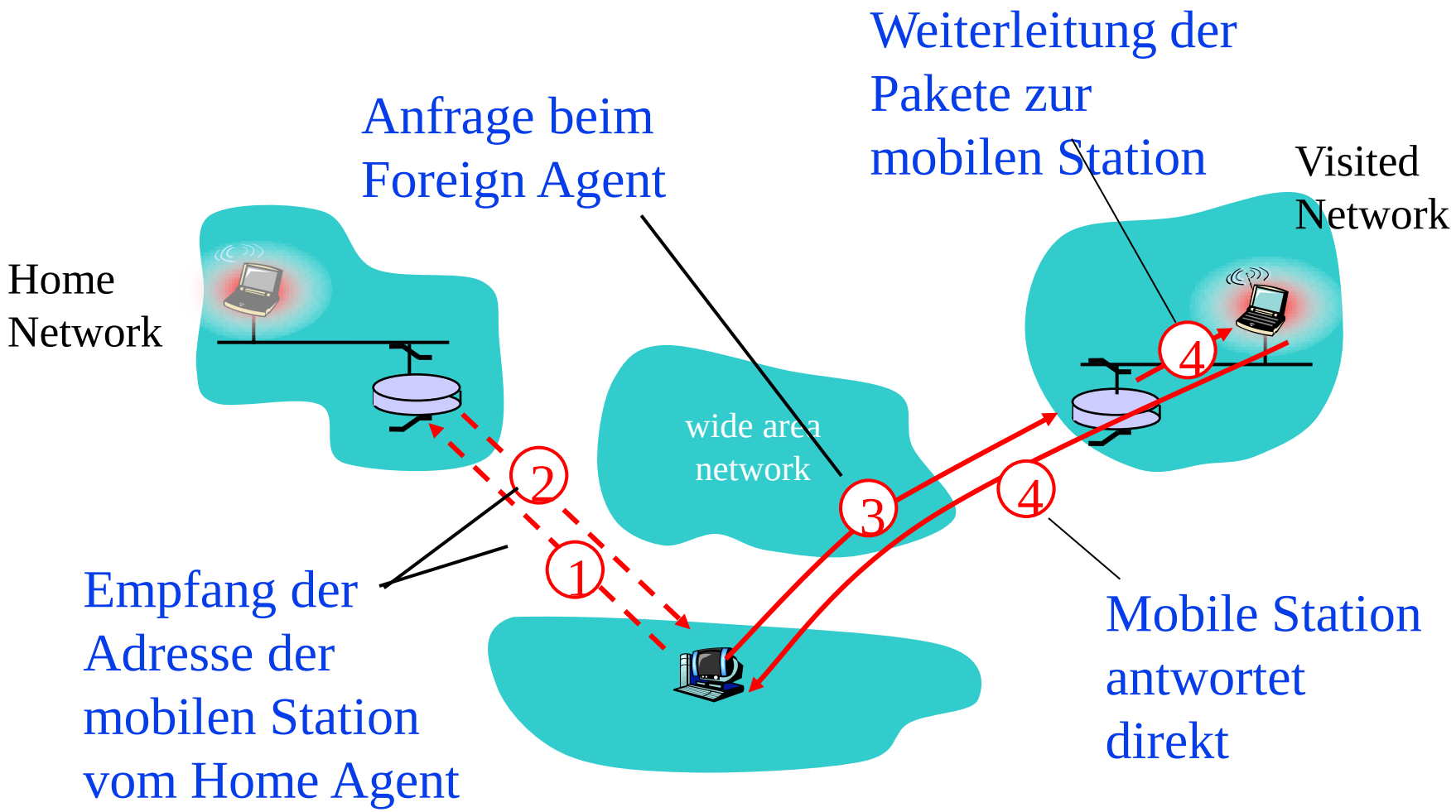
Mobile Netze: Registrierung



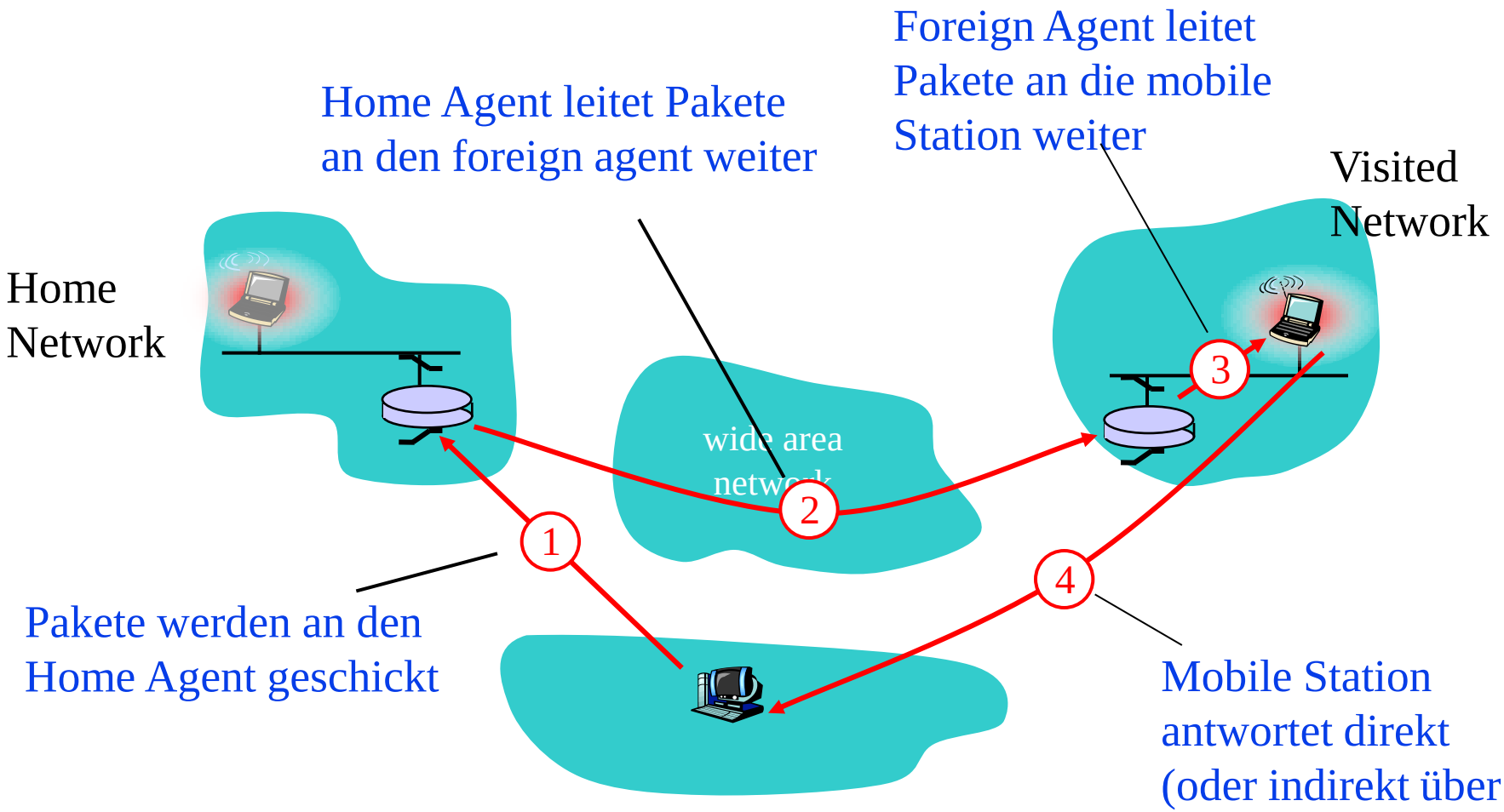
Resultat:

- *Foreign Agent kennt die mobile Station*
- *Home Agent weiß, wo sich die mobile Station befindet*

Mobile Netze: Direktes Routing



Mobile Netze: Indirektes Routing



Mobile IP: RFC 3220

- Hat viele der vorgestellten Eigenschaften:
  - Home Agents, Foreign Agents, Foreign Agent Registration, Care-of-Addresses, Encapsulation (Packet-within-Packet)
- Standardkomponenten:
  - Agent Discovery
  - Registration with Home Agent
  - Indirect Routing of Datagrams
- benutzt ICMP Nachrichten für *Advertisements* und *Registrations*
  - *ICMP Nachrichten Typ 9*

Mobile IP: RFC 3220