

5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.1.1 Übersicht

5.4.1.2 Link-State Routing

5.4.1.3 Distance Vector Routing

5.4.1.4 Routing Protokolle im Internet

5.4.2 Subnetze

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

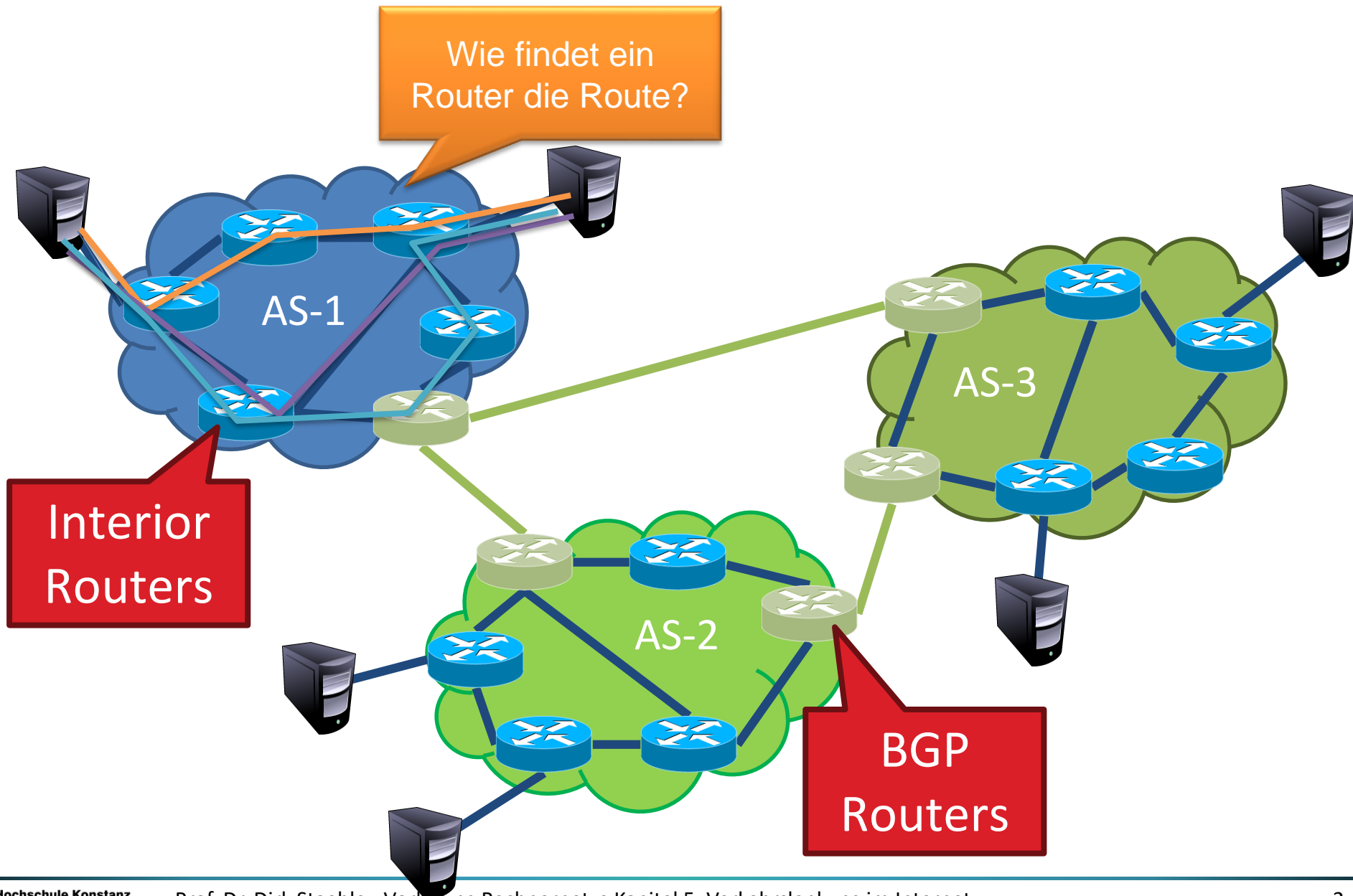
5.7 Network Address Translation (NAT)

5.8 IPv6

5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

Intra-Domain-Routing



5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.1.1 Übersicht

5.4.1.2 Link-State Routing

5.4.1.3 Distance Vector Routing

5.4.1.4 Routing Protokolle im Internet

5.4.2 Subnetze

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

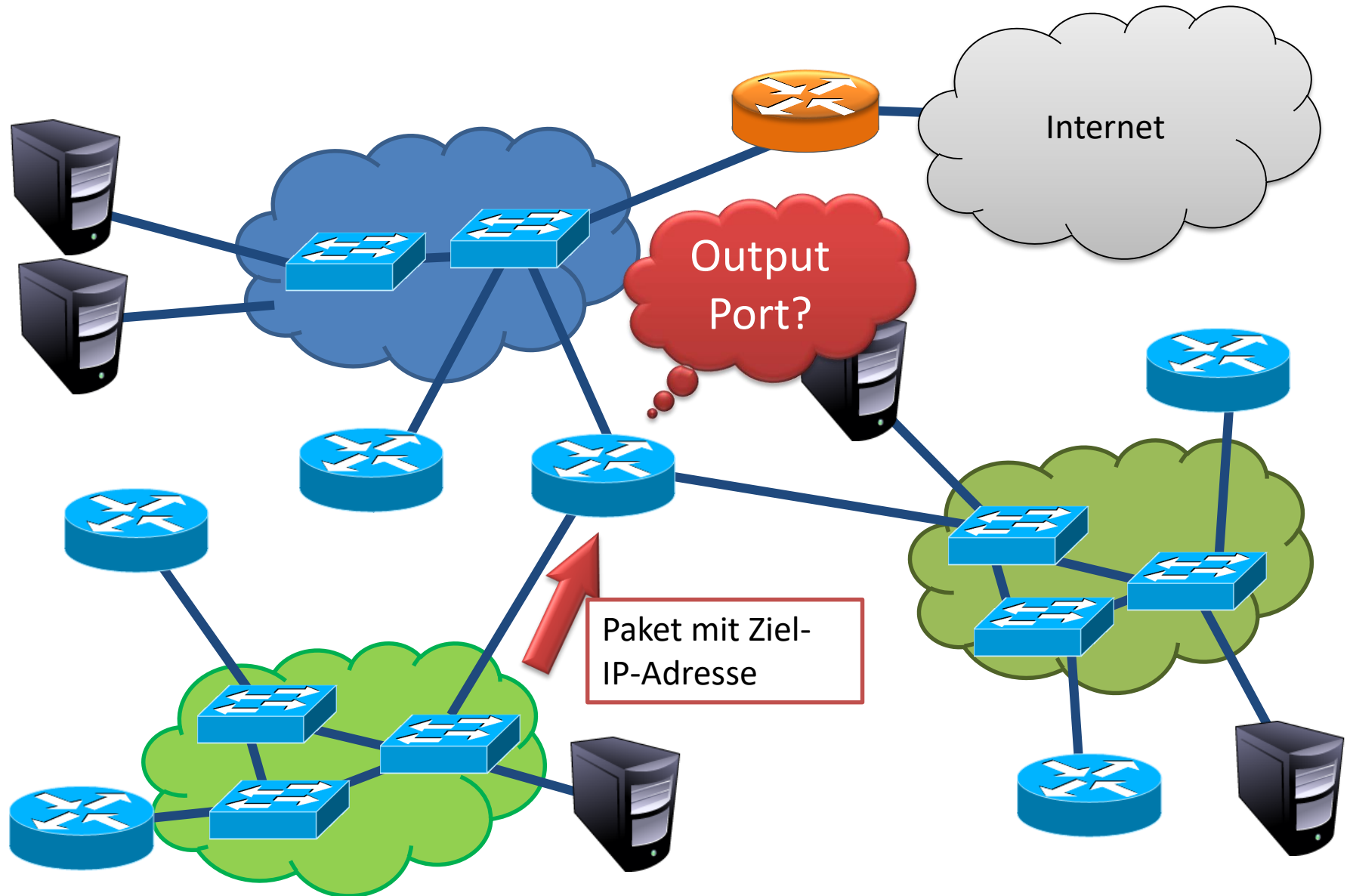
5.7 Network Address Translation (NAT)

5.8 IPv6

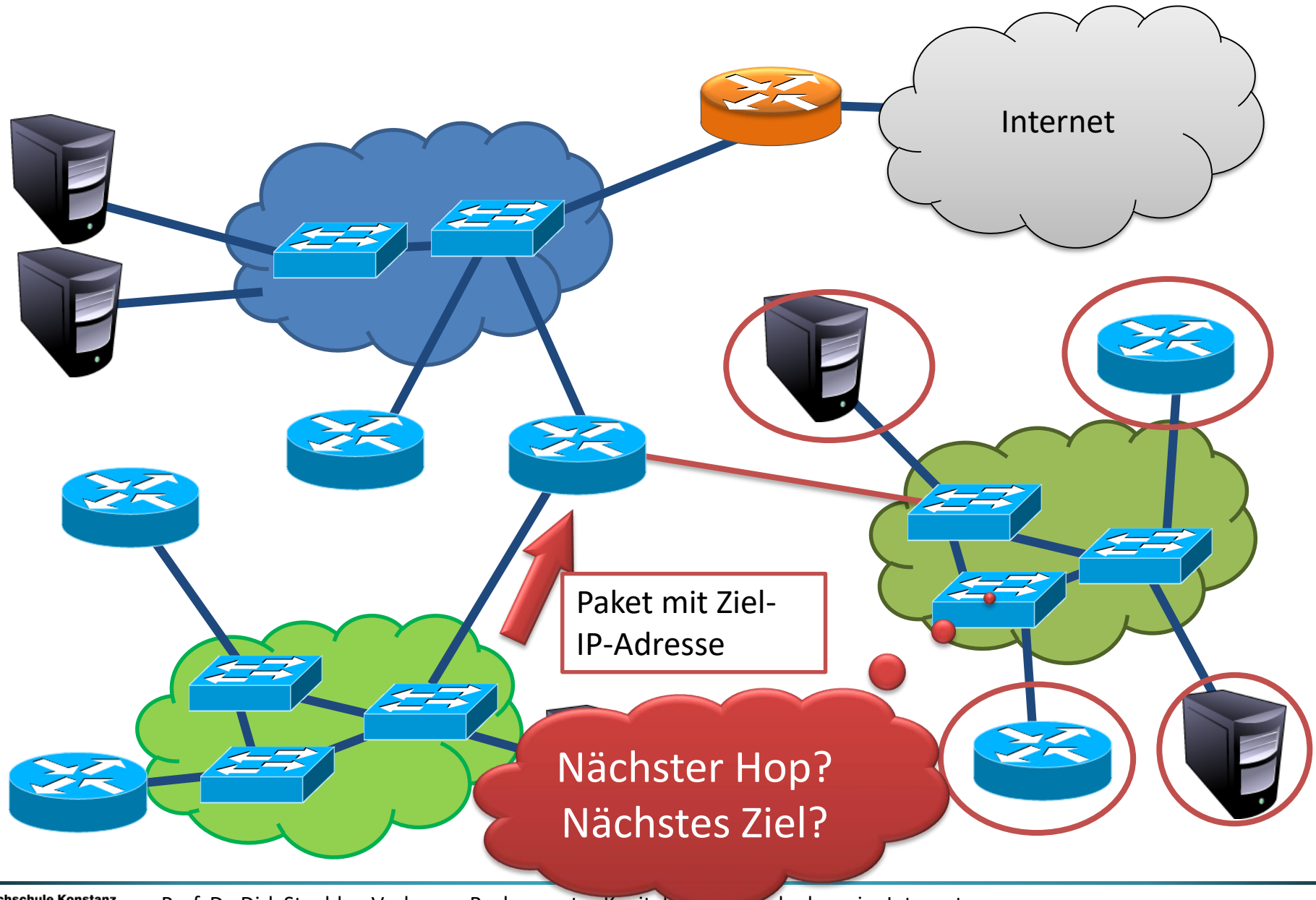
5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

Aufgabe eines Routers

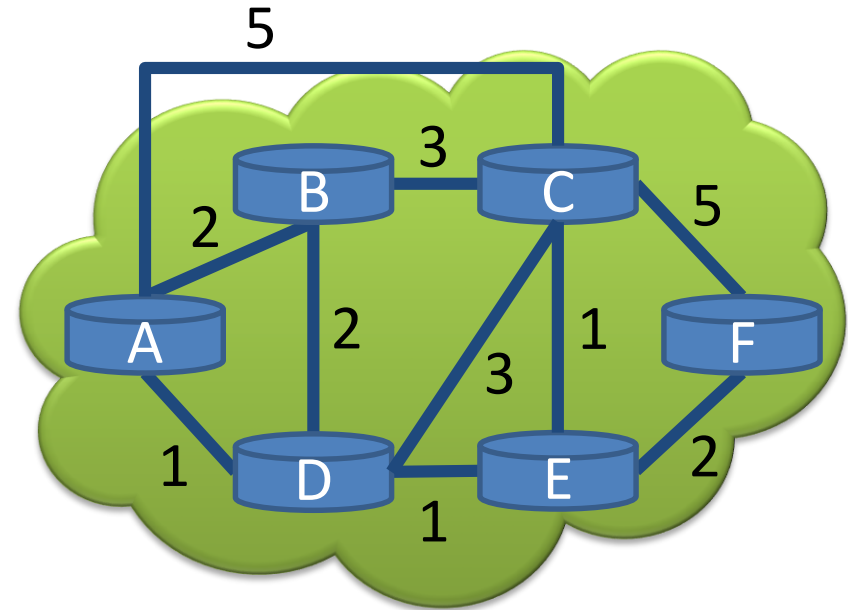
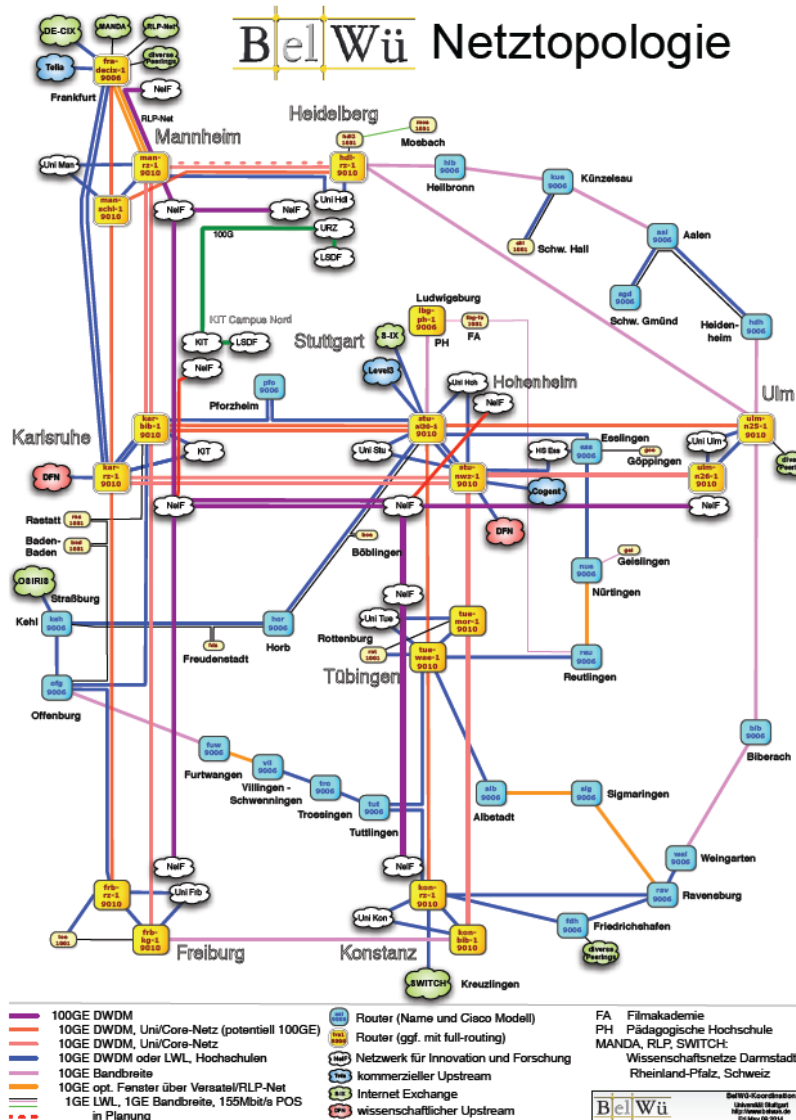


Aufgabe eines Routers



- Routing Protokolle definieren, wie Routen gefunden werden
 - Routing-Algorithmus:
 - **Distance-Vector** oder **Link State**
 - **Routing Metrik:**
 - beschreibt, wie Link-Kosten definiert sind und wie aus mehreren Link-Kosten die Pfad-Kosten für eine Route bestimmt werden
- Routing Protokolle definieren, wie benachbarte Router kommunizieren
 - Protokoll, über das Nachrichten ausgetauscht werden
 - Format und Verfahren zum Austausch von Nachrichten
 - Plug'n'Play: automatisches Finden benachbarter Router
 - Beispiel OSPF:
 - Broadcast von Hello-Paketen alle 10s
 - Broadcast des Link-States bei Veränderung

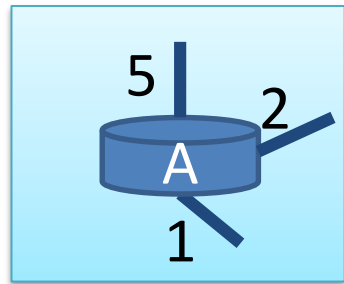
Routing auf Graphen



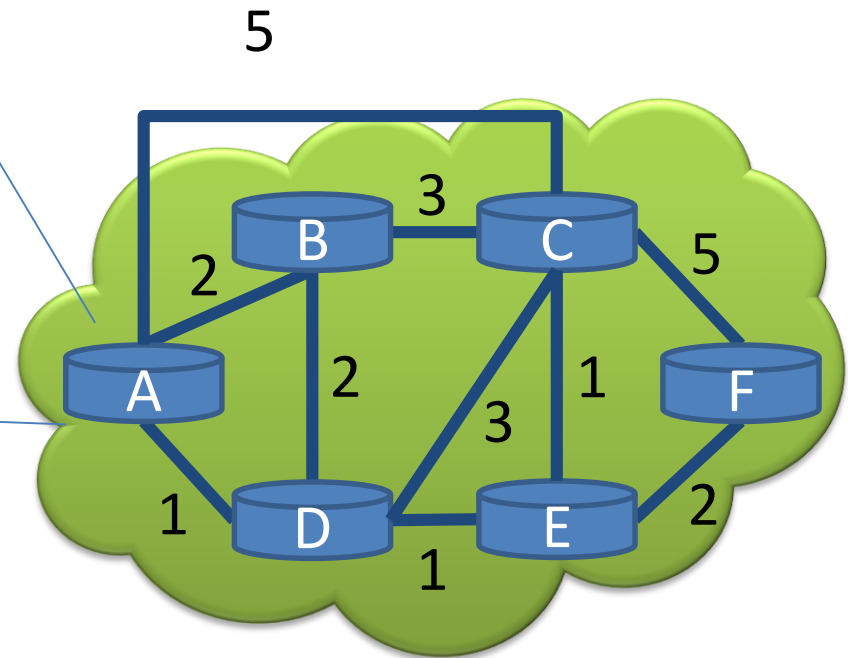
- Abstraktion der Netz-Topologie auf einen Graphen:
 - Router: Knoten
 - Links: Kanten mit Kosten entsprechend einer Routing-Metrik
- Routing:
 - Bestimmen des Pfades mit geringsten Kosten
 - Metriken: Hops, Inverses der Bandbreite, Delay, Energiekosten, Auslastung
 - normalerweise additiv

Routing Algorithmen

- Jeder Knoten kennt seine lokale Nachbarschaft
 - benachbarte Router
 - Link-Kosten entsprechend der Routing-Metrik



- Wie bestimmen die Router die beste Route durch das Netz?
 - Link-State
 - Distance-Vector



- grundlegende Verfahren, auf denen alle Routing-Protokolle aufbauen
- Link-State Routing-Protokolle basieren darauf, dass jeder Router die **gesamte Netztopologie** kennt und Routen **lokal** berechnet.
- In Distance-Vector Routing Protokolle hat jeder Router nur ein **partiell Wissen** über die Netz-Topologie und die Route wird **verteilt** und iterativ berechnet.

5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.1.1 Übersicht

5.4.1.2 Link-State Routing

5.4.1.3 Distance Vector Routing

5.4.1.4 Routing Protokolle im Internet

5.4.2 Subnetze

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

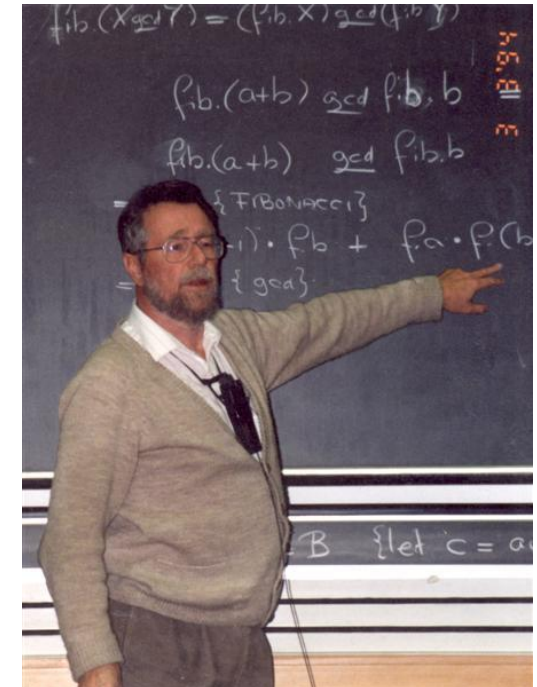
5.8 IPv6

5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

Link-State Routing

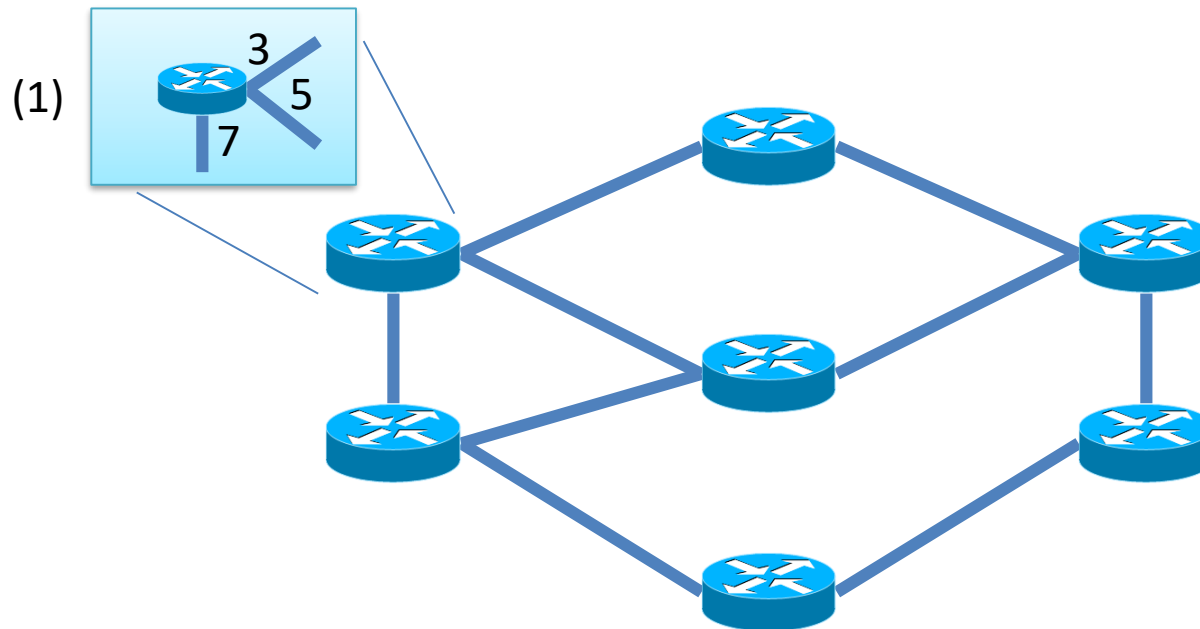
- Jeder Router teilt **allen** anderen Routern regelmäßig seine lokalen Verbindungen mit
 - Flooding: die Routing Information eines Routers wird von den anderen Routern an alle Nachbarn weitergeleitet
- Jeder Router kennt die **gesamte Netz-Topologie**
- Jeder Router berechnet die **beste Route** zu einem Ziel lokal
 - alle Router berechnen die gleichen Routen
 - Dijkstra-Algorithmus
- Wichtigstes Link State Routing Protokoll
 - OSPF (Open Shortest Path First)



Edsger Dijkstra

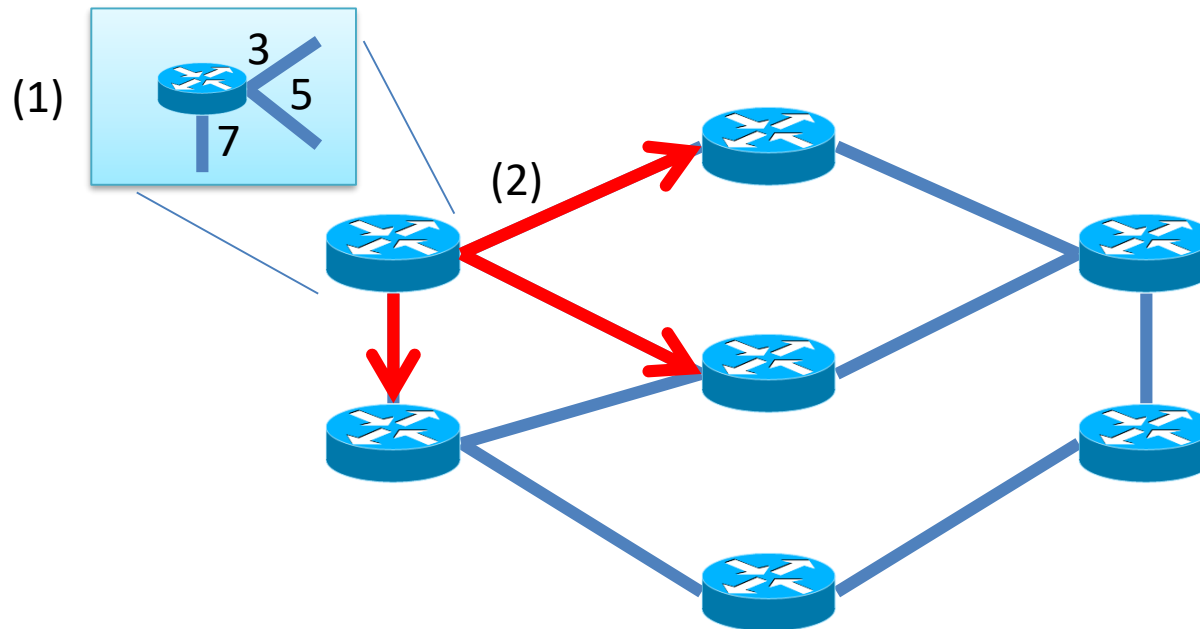
Link-State – Prinzip

- (1) **Jeder Router kennt seine Nachbarn und die Kosten zu seinen Nachbarn**
- (2) Jeder Router sendet diese Information zu allen seinen Nachbarn
- (3) Die Nachbarn leiten die Information weiter, so dass alle Router diese Information erhalten (Flooding)
- (4) Jeder Router kennt alle anderen Router inklusive deren Nachbarschaft
- (5) Berechnung der kürzesten Pfade nach dem Dijkstra-Algorithmus



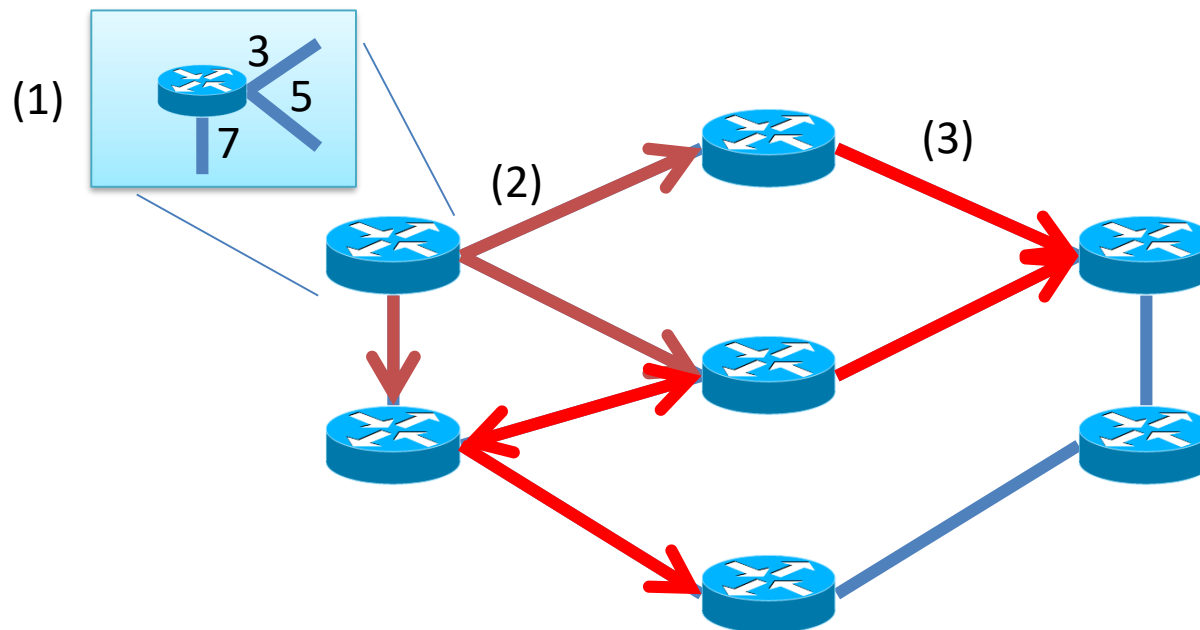
Link-State – Prinzip

- (1) Jeder Router kennt seine Nachbarn und die Kosten zu seinen Nachbarn
- (2) Jeder Router sendet diese Information zu allen seinen Nachbarn**
- (3) Die Nachbarn leiten die Information weiter, so dass alle Router diese Information erhalten (Flooding)
- (4) Jeder Router kennt alle anderen Router inklusive deren Nachbarschaft
- (5) Berechnung der kürzesten Pfade nach dem Dijkstra-Algorithmus



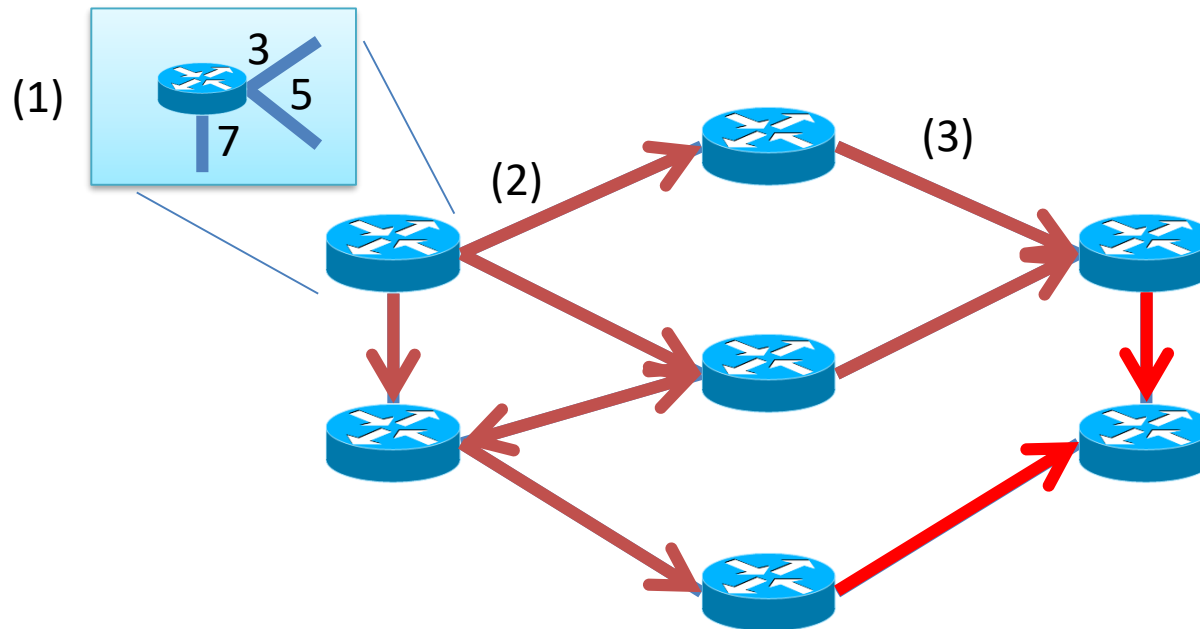
Link-State – Prinzip

- (1) Jeder Router kennt seine Nachbarn und die Kosten zu seinen Nachbarn
- (2) Jeder Router sendet diese Information zu allen seinen Nachbarn
- (3) Die Nachbarn leiten die Information weiter, so dass alle Router diese Information erhalten (Flooding)**
- (4) Jeder Router kennt alle anderen Router inklusive deren Nachbarschaft
- (5) Berechnung der kürzesten Pfade nach dem Dijkstra-Algorithmus



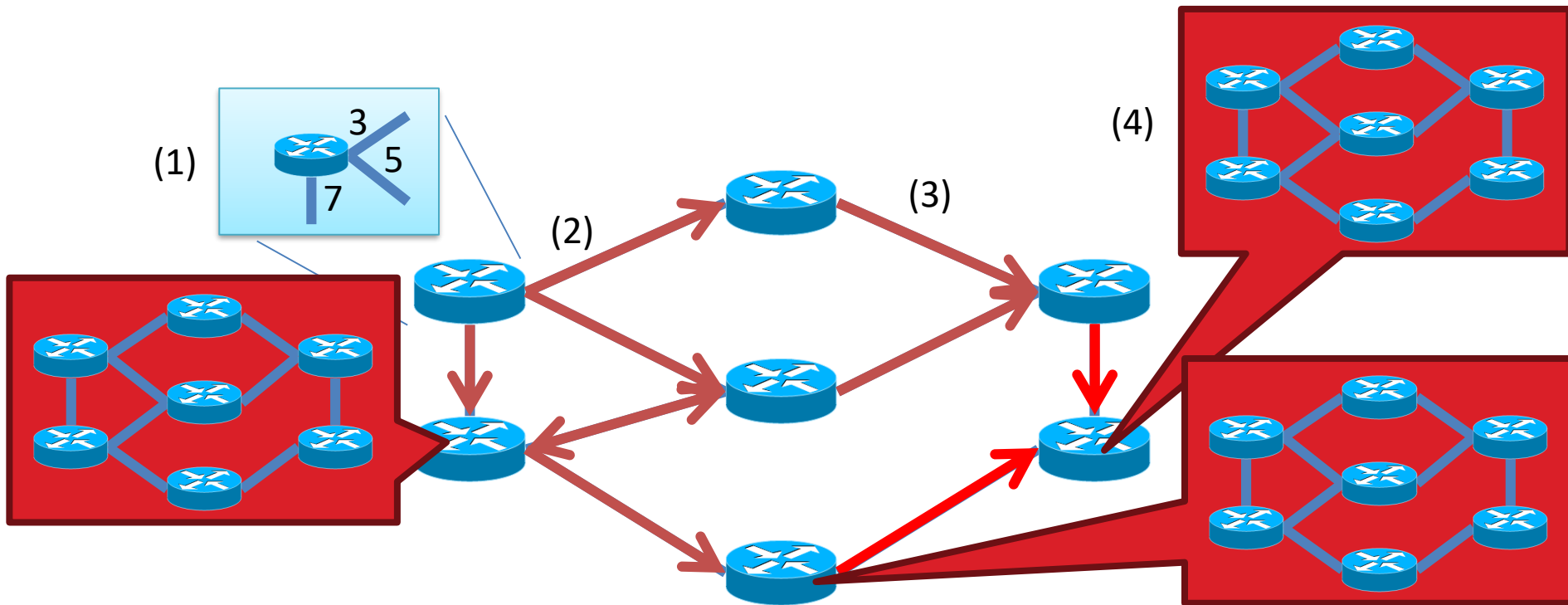
Link-State – Prinzip

- (1) Jeder Router kennt seine Nachbarn und die Kosten zu seinen Nachbarn
- (2) Jeder Router sendet diese Information zu allen seinen Nachbarn
- (3) Die Nachbarn leiten die Information weiter, so dass alle Router diese Information erhalten (Flooding)**
- (4) Jeder Router kennt alle anderen Router inklusive deren Nachbarschaft
- (5) Berechnung der kürzesten Pfade nach dem Dijkstra-Algorithmus



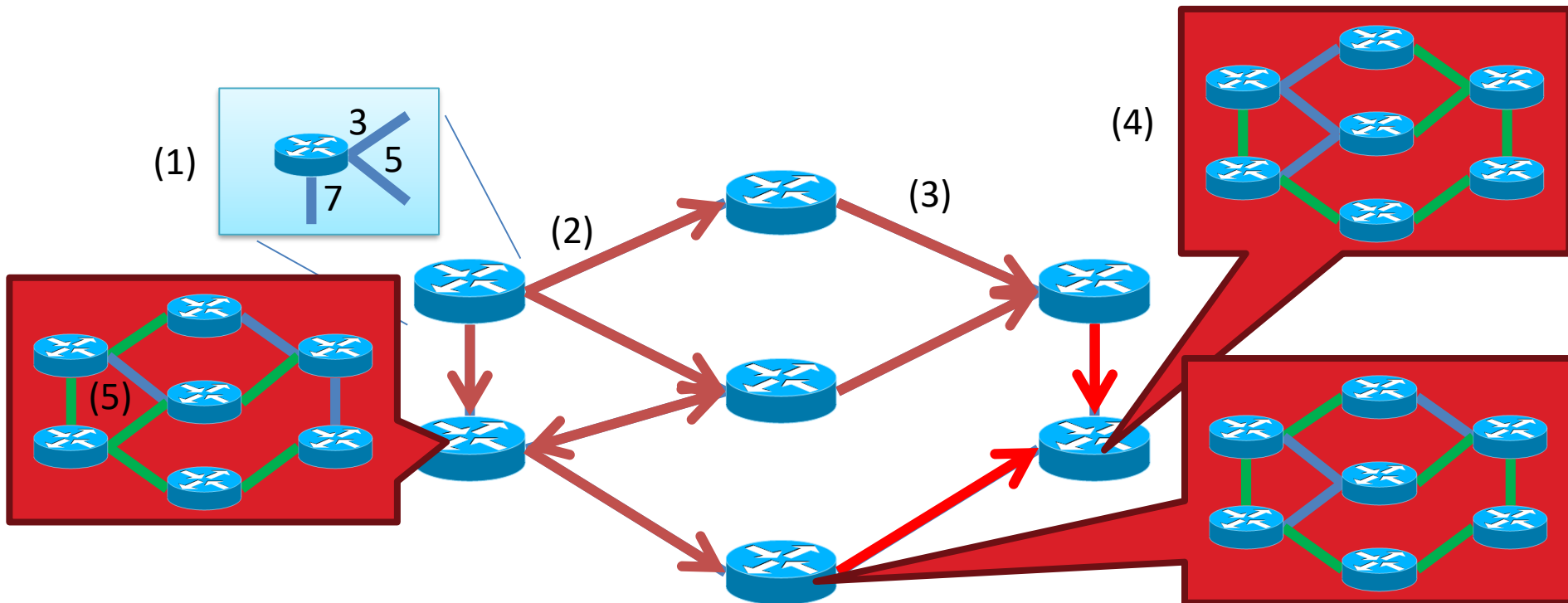
Link-State – Prinzip

- (1) Jeder Router kennt seine Nachbarn und die Kosten zu seinen Nachbarn
- (2) Jeder Router sendet diese Information zu allen seinen Nachbarn
- (3) Die Nachbarn leiten die Information weiter, so dass alle Router diese Information erhalten (Flooding)
- (4) Jeder Router kennt alle anderen Router inklusive deren Nachbarschaft**
- (5) Berechnung der kürzesten Pfade nach dem Dijkstra-Algorithmus



Link-State - Prinzip

- (1) Jeder Router kennt seine Nachbarn und die Kosten zu seinen Nachbarn
- (2) Jeder Router sendet diese Information zu allen seinen Nachbarn
- (3) Die Nachbarn leiten die Information weiter, so dass alle Router diese Information erhalten (Flooding)
- (4) Jeder Router kennt alle anderen Router inklusive deren Nachbarschaft
- (5) **Berechnung der kürzesten Pfade nach dem Dijkstra-Algorithmus**



5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.1.1 Übersicht

5.4.1.2 Link-State Routing

5.4.1.3 Distance Vector Routing

5.4.1.4 Routing Protokolle im Internet

5.4.2 Subnetze

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

5.8 IPv6

5.9 Mobilitätsunterstützung

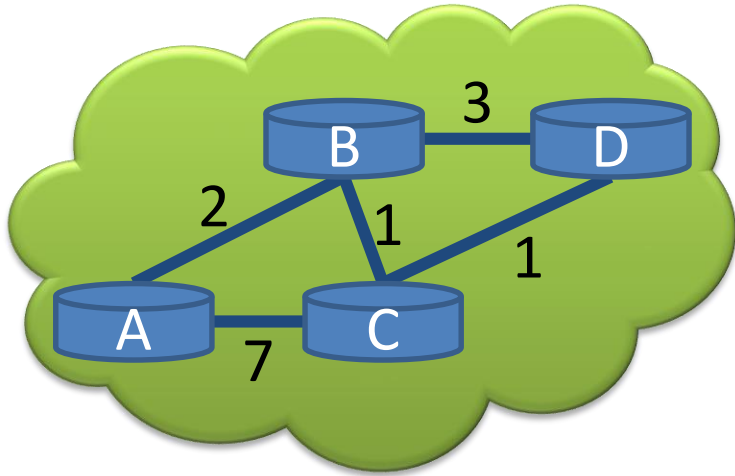
5.10 Zusammenfassung

Distance-Vector Routing

- Jeder Router kennt nur seine **lokale Umgebung**
- Jeder Router tauscht nur mit seinen direkten **Nachbarn** Informationen aus
 - Distanzvektor: enthält für alle bekannten Ziele die Pfadkosten aus Sicht des Absenders
- Jeder Router berechnet den **besten nächsten Hop** für ein Ziel, ohne die gesamte Route zu kennen
 - Kosten einer Route ergeben sich aus den Distanzvektoren der Nachbarn plus den Kosten zum jeweiligen Nachbarn
 - beste Route geht über den Nachbarn mit den geringsten Kosten
 - **Bellman-Ford Algorithmus**
- Wichtigster Vertreter:
 - EIGRP (Enhanced Interior Gateway Routing Protocol, CISCO)

Distance-Vector Beispiel - Initialisierung

- Distance-Vector: lokale Routing-Tabelle mit Ziel, Pfad-Kosten und nächstem Hop als Einträgen
 - initial: nur direkte Nachbarn



Node A

Dest.	Cost	Next
B	2	B
C	7	C
D	∞	

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	3	D

Node C

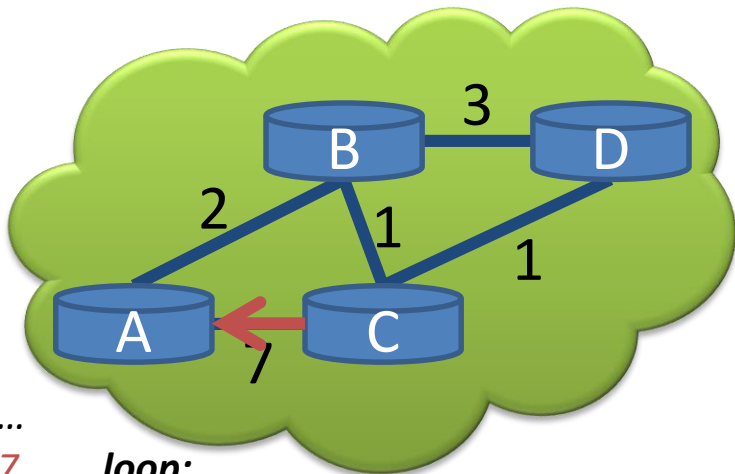
Dest.	Cost	Next
A	7	A
B	1	B
D	1	D

Node D

Dest.	Cost	Next
A	∞	
B	3	B
C	1	C

1. Initialization:
2. for all neighbors V do
3. if V adjacent to A
4. $D(A, V) = c(A, V);$
5. else
6. $D(A, V) = \infty;$
- ...

Distance Vector Beispiel – Austausch mit Nachbarn



Node A

Dest.	Cost	Next
B	2	B
C	7	C
D	8	C

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	3	D

Node C

Node D

$$D(A,D) = \min(D(A,D), D(A,C)+D(C,D))$$

$$= \min(\infty, 7 + 1) = 8$$

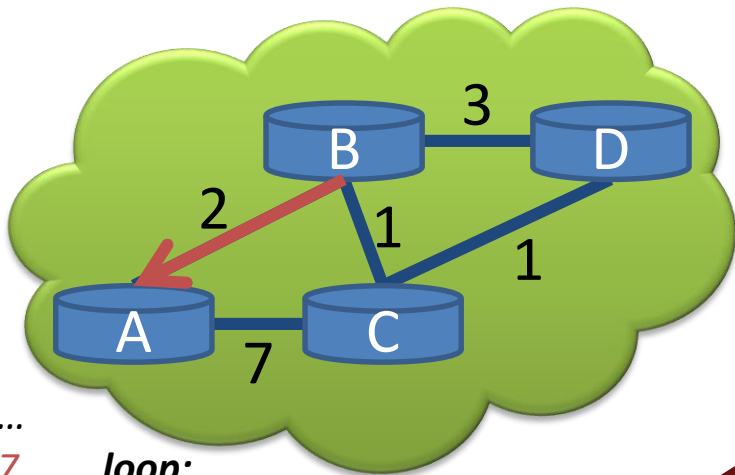
		Next
B	1	B
D	1	D
C	1	C

```

...
7. loop:
...
12. else if (update D(V, Y) received from V)
13.   for all destinations Y do
14.     if (destination Y through V)
15.       D(A,Y) = D(A,V) + D(V,Y)
16.     else
17.       D(A, Y) =
18.         min(D(A, V) + D(V, Y),
19.             if (there is a new min. for dest. Y)
20.               send D(A, Y) to all neighbors
21.         forever

```

Distance Vector Beispiel – Austausch mit Nachbarn



Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	8	C

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	3	D

$$D(A,C) = \min(D(A,C), D(A,B)+D(B,C)) \\ = \min(7, 2 + 1) = 3$$

	Cost	Next
A	7	A
B	1	B
D	1	D

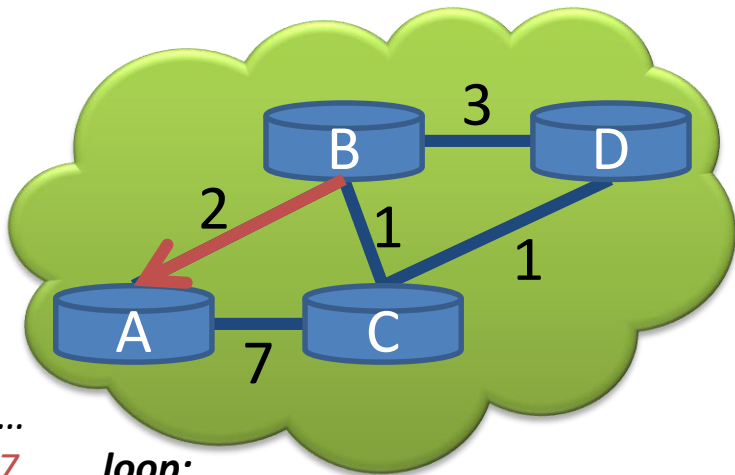
	Cost	Next
A	∞	
B	3	B
C	1	C

```

...
7.  loop:
...
12.  else if (update D(V, Y) received)
13.    for all destinations Y
14.      if (destination Y is a neighbor)
15.        D(A, Y) = D(A, V) + D(V, Y)
16.      else
17.        D(A, Y) =
            min(D(A, Y),
              D(A, V) + D(V, Y));
18.    if (there is a new min. for dest. Y)
19.      send D(A, Y) to all neighbors
20.  forever

```

Distance Vector Beispiel – Austausch mit Nachbarn



Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	5	B

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	3	D

Node C

Dest.	Cost	Next
B	1	B
D	1	D

Node D

Dest.	Cost	Next
B	3	B
C	1	C

$$D(A,D) = \min(D(A,D), D(A,B) + D(B,D))$$

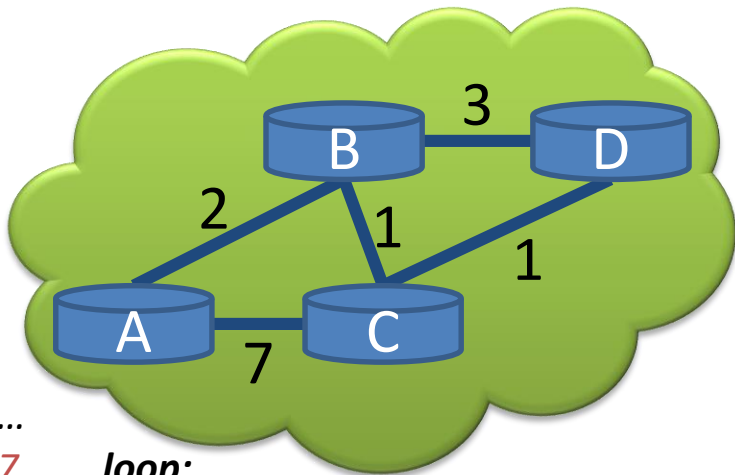
$$= \min(8, 2 + 3) = 5$$

```

...
7.  loop:
...
12. else if (update D(V, Y) received from V)
13.   for all destinations Y do
14.     if (destination Y through V)
15.       D(A,Y) = D
16.     else
17.       D(A, Y) =
18.         m
19.         D(A, V) + D(V, Y));
20.   if (there is a new min. for dest. Y)
21.     send D(A, Y) to all neighbors
22. forever

```

Distance Vector Beispiel – Austausch mit Nachbarn



Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	5	B

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	3	D

Node C

Dest.	Cost	Next
A	7	A
B	1	B
D	1	D

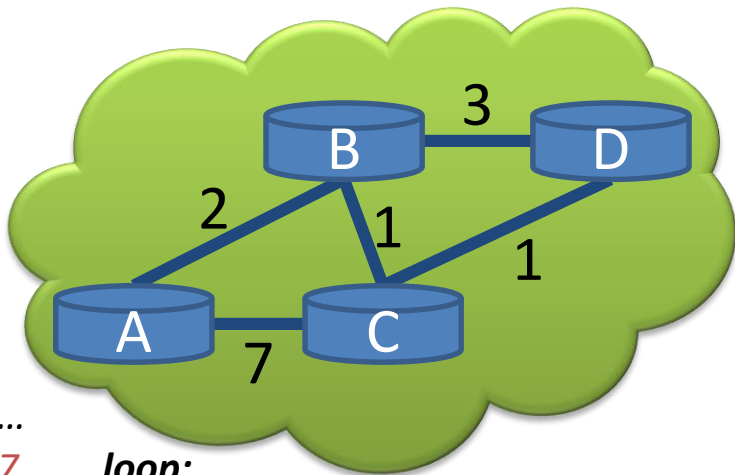
Node D

Dest.	Cost	Next
A	∞	
B	3	B
C	1	C

```

...
7.  loop:
...
12.  else if (update D(V, Y) received from V)
13.    for all destinations Y do
14.      if (destination Y through V)
15.         $D(A, Y) = D(A, V) + D(V, Y)$ ;
16.      else
17.         $D(A, Y) =$ 
           min( $D(A, Y)$ ,
               $D(A, V) + D(V, Y)$ );
18.    if (there is a new min. for dest. Y)
19.      send  $D(A, Y)$  to all neighbors
20.  forever
  
```


Distance Vector Beispiel – Austausch mit Nachbarn



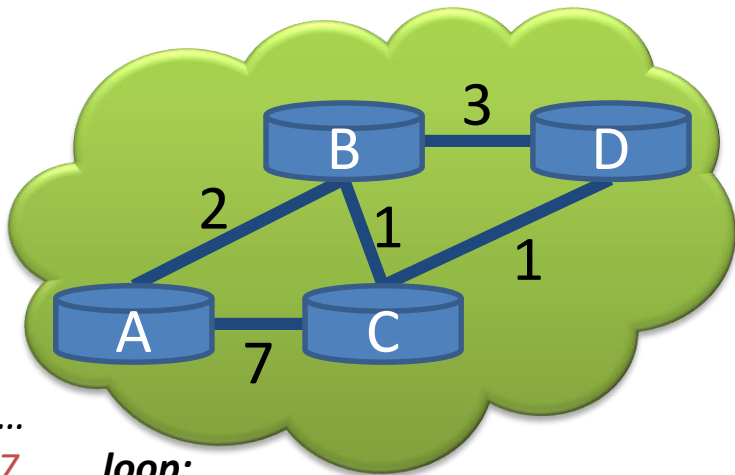
```

...
7.  loop:
...
12.  else if (update D(V, Y) received from V)
13.    for all destinations Y do
14.      if (destination Y through V)
15.         $D(A, Y) = D(A, V) + D(V, Y);$ 
16.      else
17.         $D(A, Y) =$ 
            $\min(D(A, Y),$ 
            $D(A, V) + D(V, Y));$ 
18.    if (there is a new min. for dest. Y)
19.      send D(A, Y) to all neighbors
20.  forever
  
```

Node A			Node B		
Dest.	Cost	Next	Dest.	Cost	Next
B	2	B	A	2	A
C	3	B	C	1	C
D	5	B	D	2	C

Node C			Node D		
Dest.	Cost	Next	Dest.	Cost	Next
A	3	B	A	5	B
B	1	B	B	2	C
D	1	D	C	1	C

Distance Vector Beispiel – Austausch mit Nachbarn



```

...
7.  loop:
...
12.  else if (update D(V, Y) received from V)
13.    for all destinations Y do
14.      if (destination Y through V)
15.         $D(A, Y) = D(A, V) + D(V, Y);$ 
16.      else
17.         $D(A, Y) =$ 
           min( $D(A, Y),$ 
               $D(A, V) + D(V, Y);$ )
18.    if (there is a new min. for dest. Y)
19.      send D(A, Y) to all neighbors
20.  forever
  
```

Node A			Node B		
Dest.	Cost	Next	Dest.	Cost	Next
B	2	B	A	2	A
C	3	B	C	1	C
D	4	B	D	2	C

Node C			Node D		
Dest.	Cost	Next	Dest.	Cost	Next
A	3	B	A	4	B
B	1	B	B	4	C
D	1	D	C	1	C

Red arrows indicate the exchange of distance vector information between adjacent nodes: A ↔ B, B ↔ C, and C ↔ D.

5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.1.1 Übersicht

5.4.1.2 Link-State Routing

5.4.1.3 Distance Vector Routing

5.4.1.4 Routing Protokolle im Internet

5.4.2 Subnetze

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

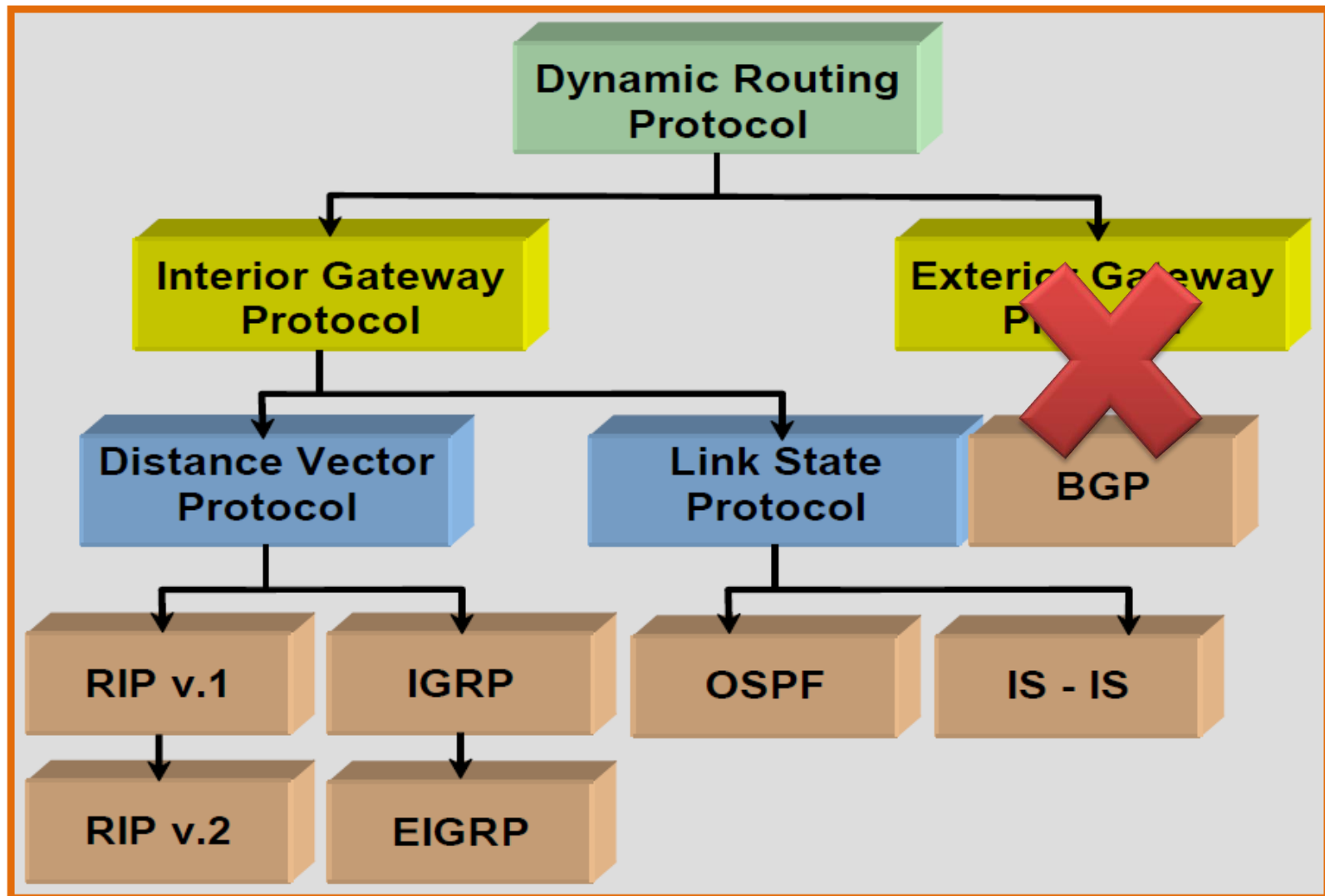
5.7 Network Address Translation (NAT)

5.8 IPv6

5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

Intra-Domain Routing Protokolle



- Distance Vector
 - RIP: Routing Information Protocol
 - veraltet ,aber noch im Einsatz
 - (E)IGRP: (Enhanced) Interior Gateway Routing Protocol
 - CISCO proprietär, soll standardisiert werden
 - von keinem anderen Hersteller unterstützt
 - Enhanced IGRP ist Nachfolger von IGRP, IGRP outdated
 - DUAL Routing erzielt schnellere Konvergenz durch Link-state-Elemente
- Link State
 - OSPF: Open Shortest Path First
 - offen, nicht proprietär
 - IS-IS: Intermediate System to Intermediate System
 - ähnlich zu OSPF
 - vor allem Einsatz in Netzen von ISPs

- Routing Protokolle definieren, wie Routen gefunden werden
 - Routing-Algorithmus:
 - Distance-Vector oder Link State
 - Routing Metrik:
 - beschreibt, wie Link-Kosten definiert sind und wie aus mehreren Link-Kosten die Pfad-Kosten für eine Route bestimmt werden
- Routing Protokolle definieren, wie benachbarte Router kommunizieren
 - Protokoll, über das Nachrichten ausgetauscht werden
 - Format und Verfahren zum Austausch von Nachrichten
 - Plug'n'Play: automatisches Finden benachbarter Router
 - Beispiel OSPF:
 - Broadcast von Hello-Paketen alle 10s
 - Broadcast des Link-States bei Veränderung

Routing Metriken

- RIP
 - 4-Bit-Metrik:
 - Hop-Count, maximal 16

- EIGRP

- 32-Bit-Metrik
- Volle Pfad-Metrik:

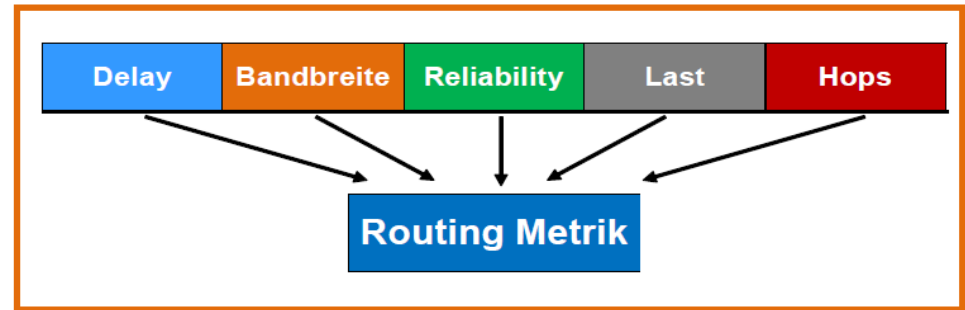
$$256 \cdot \left[(K1 \cdot BW) + K2 \cdot \frac{BW}{256 - Load} + K3 \cdot Delay \right] \cdot \left(\frac{K5}{Reliability + K4} \right)$$

- BW: 10 Gbps/min(Pfad-Bandbreite)
- Delay: Summe der Link-Delays/10μs

- Default Pfad-Metrik: $256(BW + Delay) = 256 \left(\frac{10Gbps}{\min_{links \ell} (Bandbreite_{\ell})} + \sum_{links \ell} \frac{Delay_{\ell}}{10\mu s} \right)$

- OSPF/IS-IS

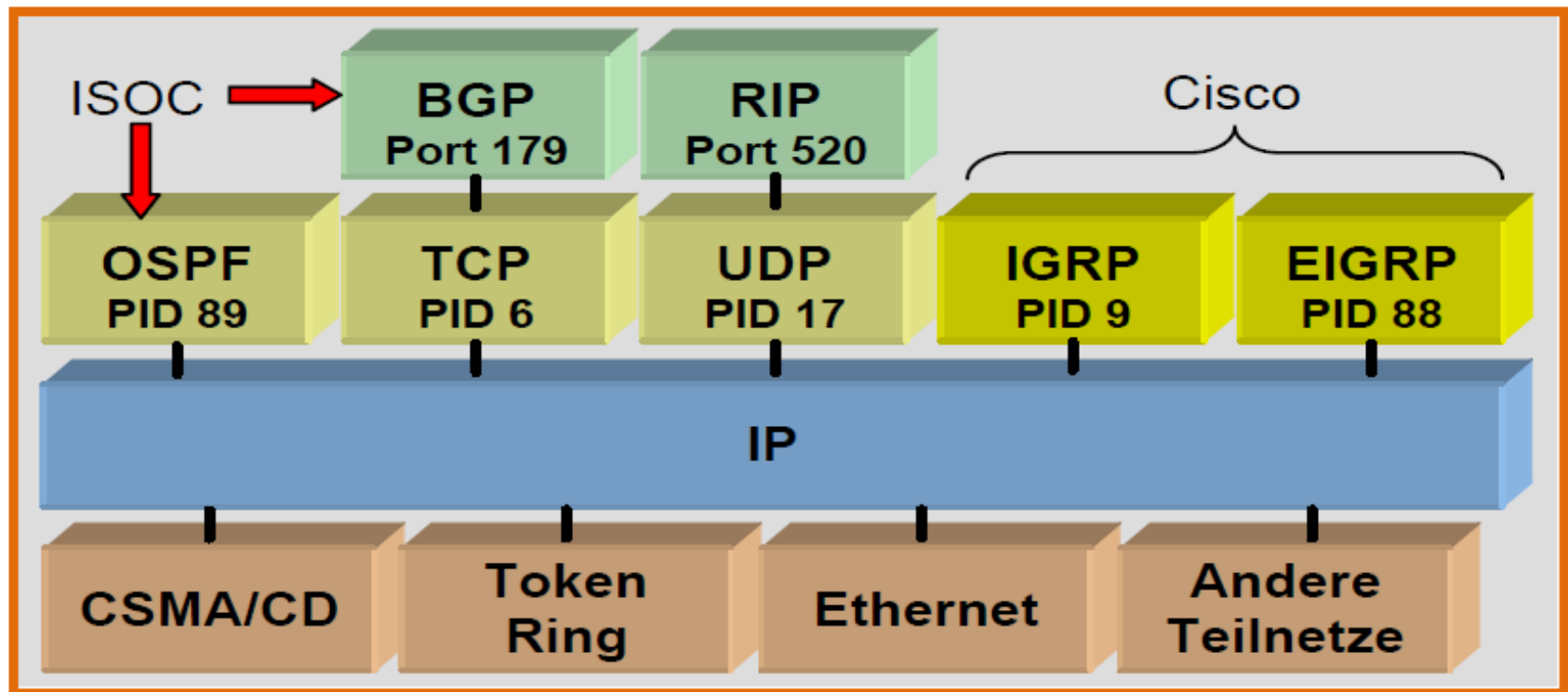
- Konfigurierbare Kosten, default: abhängig von Bandbreite
 - Linkkosten=100 Mbps/Bandbreite



- Plug'n'Play Router:
 - Neighbor discovery: Router lernen ihre Nachbarn automatisch kennen
 - Router tauschen Hello-Pakete aus
 - In LAN Segmenten an IP-Multicast-Adressen
 - OSPF: 224.0.0.5
 - EIGRP: 224.0.0.10
 - RIP: 224.0.0.9
- OSPF Hello Protocol
 - Router senden alle 10 Sekunden Hello Pakete
 - Entdecken von “neuen” Nachbarn
 - Entdecken von “defekten” Nachbarn
 - RouterDeadInterval (default: 40 Sekunden)
 - Wahl eines designierten Routers pro Subnetz
 - Designierter Router sendet LSAs für das Subnetz

Routing-Protokolle im Protokoll-Stack

- Routing-Protokolle können entweder direkt auf IP aufsetzen (OSPF, EIGRP) oder über TCP/UDP übertragen (BGP, RIP)
- EIGRP nutzt proprietäres RTP (Reliable Transport Protocol) zur gesicherten oder ungesicherten Übertragung von Routing-Informationen (z.B. Hello wird ungesichert übertragen, Routing-Tabelle gesichert)



OSPF vs. EIGRP

- Generell kann keine Aussage getroffen werden, ob Link-State-Algorithmen oder Distance-Vector-Algorithmen besser sind
- In der Praxis sind OSPF und EIGRP die relevanten Alternativen und die Entscheidung liegt beim Netzbetreiber
 - EIGRP **war** ein Cisco-proprietäres Protokoll und **wurde** nur von Cisco-Routern unterstützt
 - kein Problem in reinen Cisco-Netzen
 - nicht umsetzbar in Netzen mit Routern mehrerer Hersteller
 - 2013 startet Cisco Standardisierung von EIGRP in IETF
 - inzwischen publiziert als RFC 7868
 - Trend: OSPF hat sich mittlerweile durchgesetzt
 - zunehmender Marktanteil von Huawei und Juniper
 - Trend zu heterogenen Netzen mit Routern mehrerer Hersteller
 - keine Verwendung von EIGRP

5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.2 Subnetze

5.2.4.1 Funktionsweise eines Routers

5.4.2.2 Subnetze und Subnetzmasken

5.4.2.3 Routingtabellen

5.4.2.4 Netzwerksegmente

5.4.2.5 Address Resolution Protocol (ARP)

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

5.8 IPv6

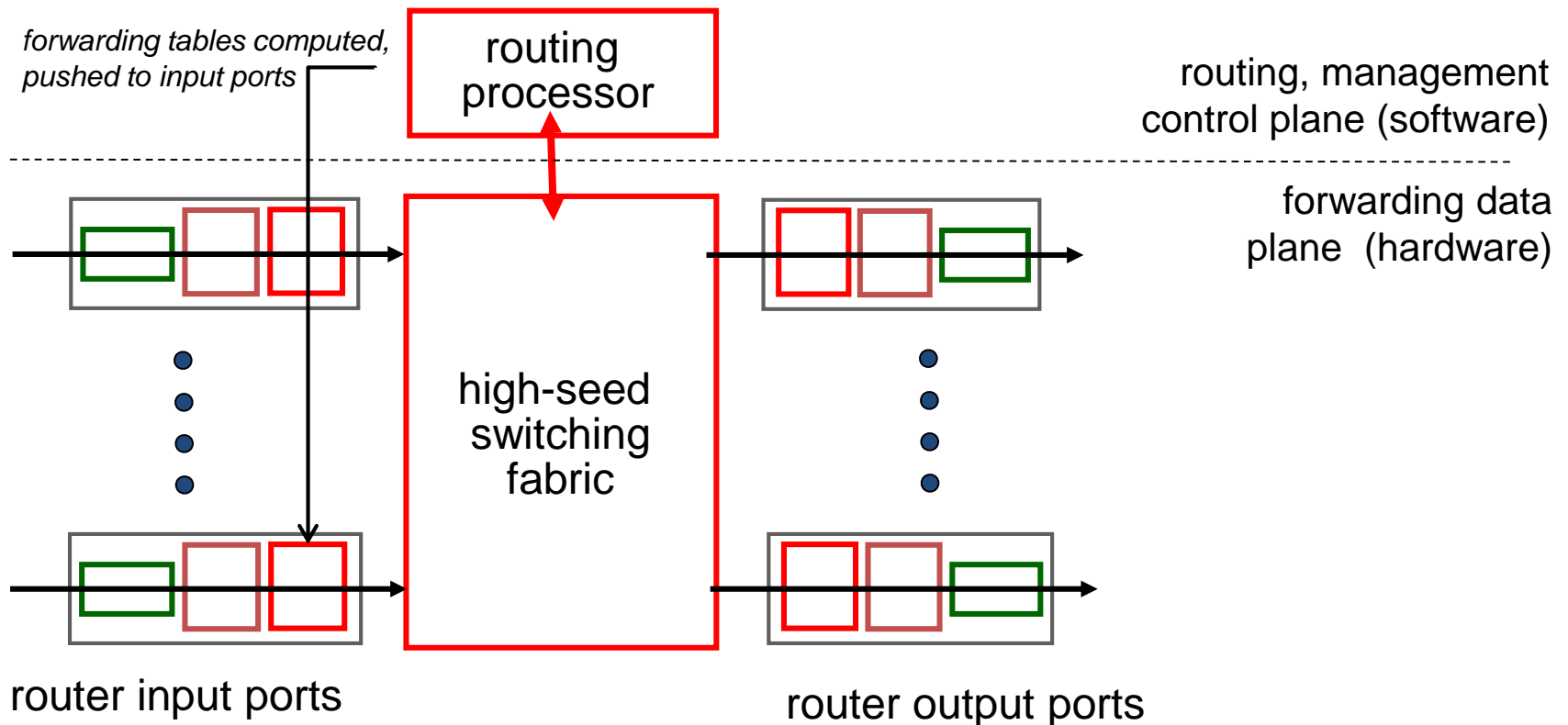
5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

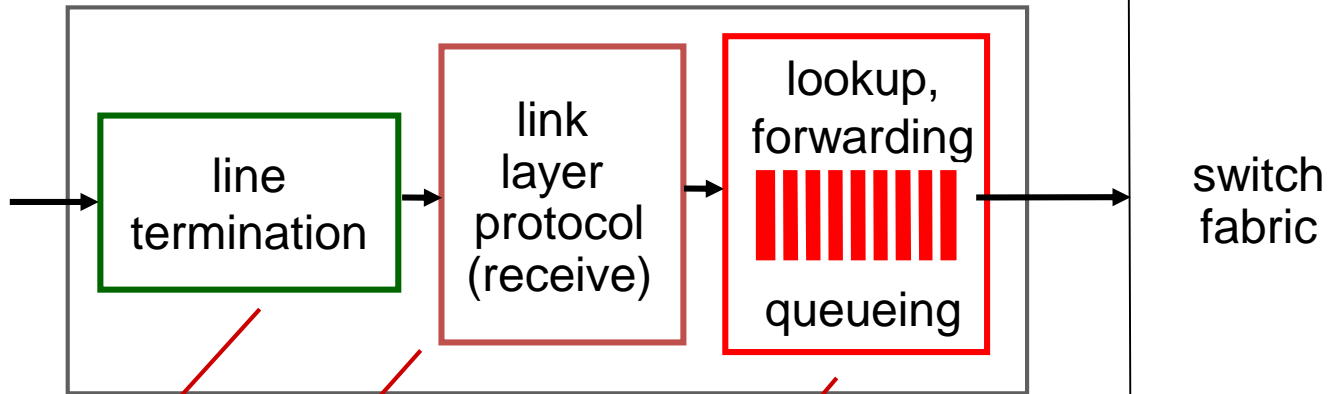
Router Architektur

Hauptfunktionen eines Routers:

- ❖ Routing-Protokoll ausführen (RIP, OSPF, BGP)
- ❖ *Pakete weiterleiten: vom Eingangs- zum (richtigen) Ausgangsport*



Funktion des Input Ports



physical layer:
bit-level reception

data link layer:
e.g. Ethernet

decentralized switching:

- in der Forwarding-Tabelle im Input-Port-Speicher den Ausgangsport zur Ziel-IP-Adresse bestimmen (lookup)
- Ziel: Input-Port Processing sollte mit Line-Speed erfolgen
- Queuing: Pakete werden gepuffert und warten, wenn sie schneller ankommen als sie in die Switch Fabric geschoben werden können

Routingprotokoll und Routingtabellen

- Jeder Router hält eine Routing-Tabelle, die für jede IP-Adresse die Information enthält, wie ein Paket mit dieser Ziel-Adresse weitergeleitet wird.
- Der Router betreibt ein oder mehrere Routingprotokolle, um Informationen über die Erreichbarkeit und die beste Route zu IP-Adressen zu erhalten. Diese Informationen werden in die Routingtabelle geschrieben.
- Bislang haben wir Routingprotokolle unter der Annahme betrachtet, dass jeder Netzknoten ein Router ist und das Routing zwischen Routern betrachtet.
- In der Praxis werden IP-Adressen innerhalb eines Autonomen Systems (innerhalb eines Netzes) strukturiert vergeben, so dass Routingtabellen nicht nur Routen für einzelne IP-Adressen enthalten sondern auch Routen für IP-Adressbereiche oder Subnetze.
- Dies ist notwendig, um die **Größe von Routingtabellen** möglichst klein zu halten. Dadurch
 - sinken die Hardware-Anforderungen und damit auch Kosten für Router
 - verringert sich die Verzögerung für die Ermittlung der richtigen Route

Adress-Typen und resultierende Netze

- Unzureichende Anzahl von Netzen und Hosts pro Netz nach klassischer Aufteilung in Netzklassen
 - Unterteilung von Netzen in **Subnetze** (→ Intra-Domain-Routing)
 - Flexibel skalierbare Netzgrößen mit **CIDR** (→ Inter-Domain-Routing)
 - CIDR=Classless Inter-Domain Routing

Class	Prefix	Network	Number of Classes	Hosts per Class
A	1	7	$2^7 - 2^1 = 126$	$2^{24} - 2^1 = 16,777,214$
B	2	14	$2^{14} = 16,398$	$2^{16} - 2^1 = 65,534$
C	3	21	$2^{21} = 2,097,512$	$2^8 - 2^1 = 254$
			Total: 2,114,036	

viel zu viele

zu viele

zu viele Netze
(siehe Inter-Domain-routing)

zu wenige

¹⁾ (0 and 127 are reserved)

5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.2 Subnetze

5.2.4.1 Funktionsweise eines Routers

5.4.2.2 Subnetze und Subnetzmasken

5.4.2.3 Routingtabellen

5.4.2.4 Netzwerksegmente

5.4.2.5 Address Resolution Protocol (ARP)

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

5.8 IPv6

5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

- Problem:
 - Class A und B Netze sind zu groß für flachen Adressraum
 - zu viele Hosts, zu große Routing-Tabellen
- Lösung:
 - hierarchisch strukturierter Adressraum innerhalb eines Netzes (AS)
 - Sicht von außen: ein Netz, ein Eintrag in globale Routing-Tabellen
 - interne Sicht: Aufteilen des Adressraums in Subnetze durch Subnetzmaske
 - IPv4 Subnetzmaske:
 - besteht aus 32 Bit mit führenden 1en
 - Anzahl führender 1en spezifiziert Größe des Subnetzes
 - auch als $/n$ geschrieben mit n der Anzahl führender 1en



Subnetzmaske
(Subnet Mask):

11111111 11111111 11000000 00000000

Subnetzmaske: Beispiele

Die Subnetzmaske kann generell definiert werden und muss nicht aus führenden 1en bestehen. In der Praxis werden aber vor allem Subnetzmasken mit führenden 1en genutzt und die Subnetzmaske mit 24 1en auch als /24 geschrieben. Generelle Operation zur Bestimmung des Netzes ist IP AND SUBNETMASK und des Hosts IP AND NOT SUBNETMASK. Bei Subnetzmasken mit führenden 1en reduziert sich das dazu, das Netz durch den Teil der Subnetzmaske mit 1en und den Host durch den Teil der Subnetzmaske mit 0en zu identifizieren.

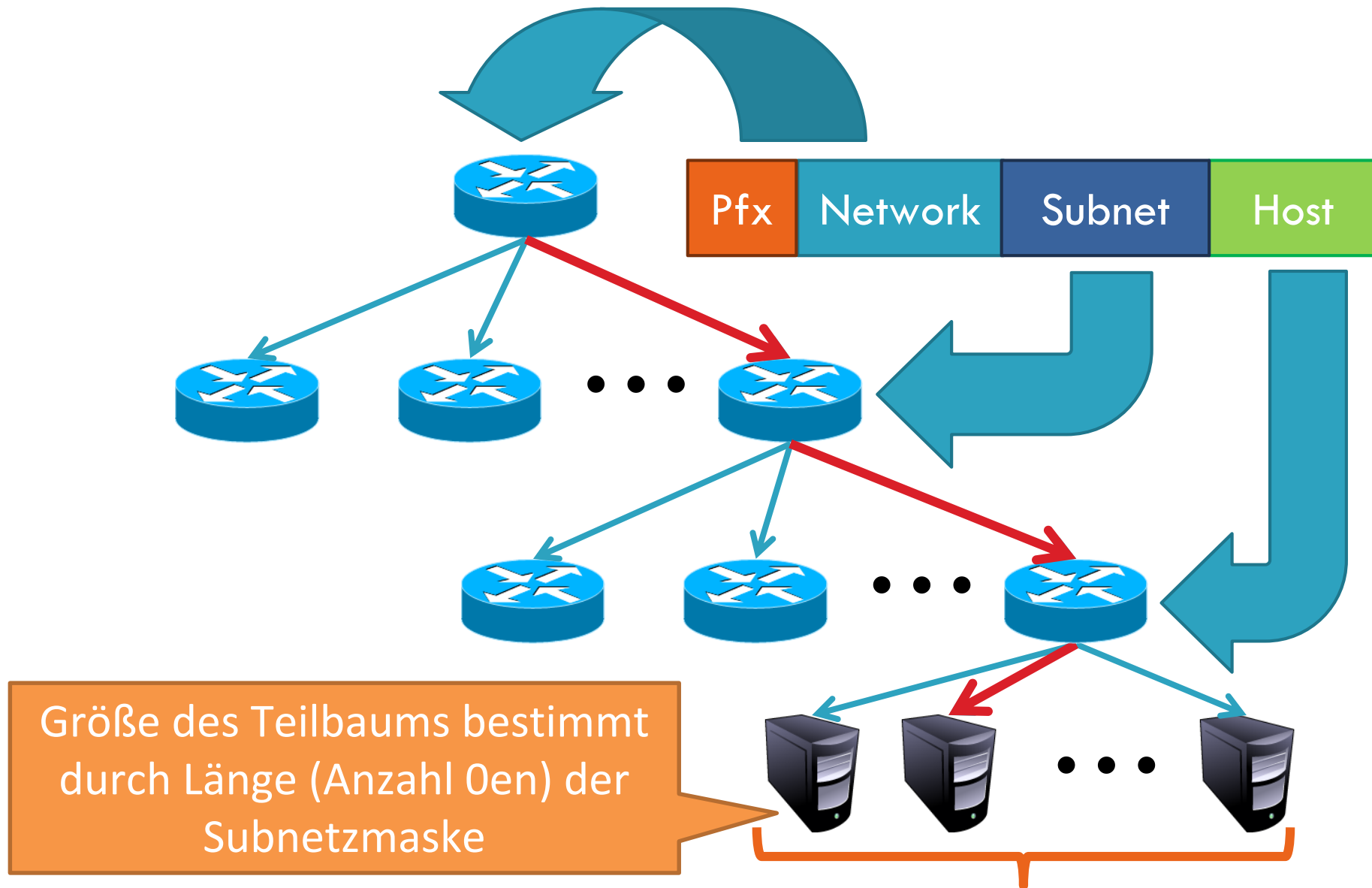
- Netz aus IP Adresse extrahieren:

IP Address:	10110101	11011101	01010100	01110010
Subnet Mask:	& 11111111	11111111	11000000	00000000
Result:	10110101	11011101	01000000	00000000

- Host aus IP Adresse extrahieren:

IP Address:	10110101	11011101	01010100	01110010
Subnet Mask:	& ~(11111111	11111111	11000000	00000000)
Result:	00000000	00000000	00010100	01110010

Hierarchische Strukturierung von Subnetzen



Hierarchische Strukturierung von Subnetzen



Pfx

Network

Subnet

Host

- Baum hat keine feste Tiefe
- Verzweigung durch spezifischere Subnetzmasken (spezifischer=mehr 1en)



...



Größe des Teilbaums bestimmt durch Länge (Anzahl 0en) der Subnetzmaske



...



5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.2 Subnetze

5.4.2.1 Funktionsweise eines Routers

5.4.2.2 Subnetze und Subnetzmasken

5.4.2.3 Routingtabellen

5.4.2.4 Netzwerksegmente

5.4.2.5 Address Resolution Protocol (ARP)

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

5.8 IPv6

5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

- Jeder IP-fähige Netzknoten (Hosts und Router) verwaltet im IP-Layer eine Routing Tabelle. Über die Routing Tabelle kann zu einer Ziel-IP-Adresse bestimmt werden, zu welchem nächsten Netzknoten und über welches lokale Interface das IP Paket weitergeleitet werden soll.
- Ein Eintrag in der Routing-Tabelle beschreibt einen Route zu einem Ziel und besteht aus
 - dem Netzwerkziel und der Netzwerkmaske, die bestimmen welche IP-Adressen über diese Route erreicht werden
 - dem Gateway (nächster Hop) und der Schnittstelle (dem lokalen Interface), auf dem Pakete für diese Route weitergeleitet werden
 - der Metrik, die die Kosten für diese Route bestimmen. Bei mehreren gleichwertigen Routen zu einem Ziel wird die Route mit niedrigerer Metrik gewählt
- Die lokale Routing-Tabelle kann mit den Befehlen "netstat -r" oder "route print" ausgegeben werden. Mit dem "route" Befehl kann die Routingtabelle auch modifiziert werden.
 - unter 0.0.0.0 (alle Internetadressen) wird das Default-Gateway eingetragen.

Beispiel einer Routing-Tabelle (Windows 8)

```

C:\Windows\system32\cmd.exe

C:\HTWG\Vorlesungen\REchnernetze\Labor>netstat -r

=====
Schnittstellenliste
13...00 24 d7 cb 76 59 .....Microsoft Virtual WiFi Miniport Adapter
11...00 24 d7 cb 76 58 .....Intel(R) Centrino(R) Ultimate-N 6300 AGN
10...00 21 cc 63 82 2c .....Intel(R) 82579LM Gigabit Network Connection
1.....Software Loopback Interface 1
18...00 00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
22...00 00 00 00 00 00 00 e0 Microsoft-ISATAP-Adapter #5
23...00 00 00 00 00 00 00 e0 Microsoft-ISATAP-Adapter #6
43...00 00 00 00 00 00 00 e0 Microsoft-ISATAP-Adapter #7
=====

IPv4-Routentabelle
=====
Aktive Routen:
   Netzwerkziel   Netzwerkmaske   Gateway   Schnittstelle   Metrik
   0.0.0.0         0.0.0.0         141.37.28.254   141.37.29.93     20
   127.0.0.0         255.0.0.0       Auf Verbindung  127.0.0.1        306
   127.0.0.1         255.255.255.255 Auf Verbindung  127.0.0.1        306
  127.255.255.255    255.255.255.255 Auf Verbindung  127.0.0.1        306
   141.37.28.0       255.255.252.0   Auf Verbindung  141.37.29.93     276
   141.37.29.93      255.255.255.255 Auf Verbindung  141.37.29.93     276
   141.37.31.255     255.255.255.255 Auf Verbindung  141.37.29.93     276
   224.0.0.0         240.0.0.0       Auf Verbindung  127.0.0.1        306
   224.0.0.0         240.0.0.0       Auf Verbindung  141.37.29.93     276
  255.255.255.255    255.255.255.255 Auf Verbindung  127.0.0.1        306
  255.255.255.255    255.255.255.255 Auf Verbindung  141.37.29.93     276
=====

```

Aufbau eines Routing-Tabellen-Eintrags

- Adressbereich + Subnetzmaske: legt fest, auf welche IP-Adressen die Route zutrifft
 - Adressbereich: IP-Adresse bzw. Adress-Präfix
 - Subnetzmaske: Länge des Adress-Präfixes
 - auf diese Route passen Pakete, bei denen das Präfix der Zieladresse mit dem Präfix des Adressbereichs der Route übereinstimmt
- Next-Hop + Interface: legt fest, wie das Paket weitergesendet wird
 - Interface: legt fest, über welches lokale Interface (welchen Port, welche Netzwerkkarte) des Routers (oder Hosts) das Pakete versendet wird
 - Next-Hop: bestimmt die IP-Adresse des nächsten Zielrechners, an dessen MAC-Adresse das Paket versendet wird. Der Next-Hop muss über die MAC-Adresse (Layer 2) erreichbar sein.
- Metrik: legt die Kosten für eine Route fest. Die Route mit den niedrigsten Kosten wird gewählt.

- Vorgehensweise:
 - entscheidendes Byte bestimmen und Binärdarstellung ermitteln
 - führende Bits mit Netzwerk-/Subnetzmaske vergleichen, bei Übereinstimmung liegt die IP-Adresse im Subnetz/trifft die Route auf die IP-Adresse zu
- Im welchem der folgenden Subnetze liegt die IP-Adresse 23.56.80.37?
 - 23.40.0.0/15
 - 23.56.64.0/23
 - 23.56.64.0/18
 - 23.56.80.38/31

- Lösung:

23.56.80.37:	23.0011 1000.0101 0000.0010 0101
23.40.0.0/15:	23.0010 1000.0 -> keine Übereinstimmung
23.56.64.0/23:	23.56.0100 0000.0 -> keine Übereinstimmung
23.56.64.0/18:	23.56.0100 0000.0 -> Übereinstimmung
23.56.80.38/31:	23.56.80.0010 0110 -> keine Übereinstimmung

Routing-Tabelle: Longest Prefix Match

Address Pattern	Subnet Mask	Destination Router
0.0.0.0	0.0.0.0	Router 4
128.0.0.0	255.0.0.0	Router 2
128.42.0.0	255.255.0.0	Router 3
128.42.128.0	255.255.128.0	Router 5
128.42.222.0	255.255.255.0	Router 1

- Frage: Wohin geht Paket mit Ziel 128.42.222.198?
 - passt auf alle Einträge
- Longest prefix match
 - wähle das passende Adressmuster mit den meisten 1sen in der Subnetzmaske
 - das ist die spezifischste Route
 - Paket geht an Router 1

Beispiel: Routing anhand einer Routing-Tabelle

- auf der linken Seite ist eine Routing-Tabelle gegeben
- gesucht wird die Route, auf der die Pakete auf der rechten Seite weitergeleitet werden

Route	Adressbereich		Interface	Next Hop	Ziel-Adressen	Route?
	Adresse	Subnetzmaske				
1	0.0.0.0	0.0.0.0	32.65.1.1	X	21.7.19.42	
2	32.65.40.0	255.255.252.0	32.65.48.136	H	32.65.41.23	
3	32.65.40.0	255.255.255.0	32.65.48.128	F	32.65.48.140	
4	32.65.43.0	255.255.255.0	32.65.48.136	J	32.65.48.9	
5	32.65.44.0	255.255.252.0	32.65.48.0	C	32.65.47.4	
6	32.65.48.128	255.255.255.240	32.65.48.134	H		
7	32.65.48.0	255.255.255.248	32.65.48.0	C		
8	32.65.48.8	255.255.255.254	32.65.48.8	on route		
9	32.65.48.128	255.255.255.254	32.65.48.128	on route		
10	32.65.48.134	255.255.255.254	32.65.48.134	on route		
11	32.65.48.136	255.255.255.254	32.65.48.136	on route		
12	32.65.48.0	255.255.255.255	32.65.48.0	on route		
13	32.65.48.130	255.255.255.255	32.65.48.128	F		
14	32.65.48.138	255.255.255.255	32.65.48.136	J		

Beispiel: Routing anhand einer Routing-Tabelle

- auf der linken Seite ist eine Routing-Tabelle gegeben
- gesucht wird die Route, auf der die Pakete auf der rechten Seite weitergeleitet werden

Route	Adressbereich		Interface	Next Hop	Ziel-Adressen	Route?
	Adresse	Subnetzmaske				
1	0.0.0.0	0.0.0.0	32.65.1.1	X	21.7.19.42	1
2	32.65.40.0	255.255.252.0	32.65.48.136	H	32.65.41.23	2
3	32.65.40.0	255.255.255.0	32.65.48.128	F	32.65.48.140	6
4	32.65.43.0	255.255.255.0	32.65.48.136	J	32.65.48.9	8
5	32.65.44.0	255.255.252.0	32.65.48.0	C	32.65.47.4	5
6	32.65.48.128	255.255.255.240	32.65.48.134	H		
7	32.65.48.0	255.255.255.248	32.65.48.0	C		
8	32.65.48.8	255.255.255.254	32.65.48.8	on route		
9	32.65.48.128	255.255.255.254	32.65.48.128	on route		
10	32.65.48.134	255.255.255.254	32.65.48.134	on route		
11	32.65.48.136	255.255.255.254	32.65.48.136	on route		
12	32.65.48.0	255.255.255.255	32.65.48.0	on route		
13	32.65.48.130	255.255.255.255	32.65.48.128	F		
14	32.65.48.138	255.255.255.255	32.65.48.136	J		

5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.2 Subnetze

5.4.2.1 Funktionsweise eines Routers

5.4.2.2 Subnetze und Subnetzmasken

5.4.2.3 Routingtabellen

5.4.2.4 Netzwerksegmente

5.4.2.5 Address Resolution Protocol (ARP)

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

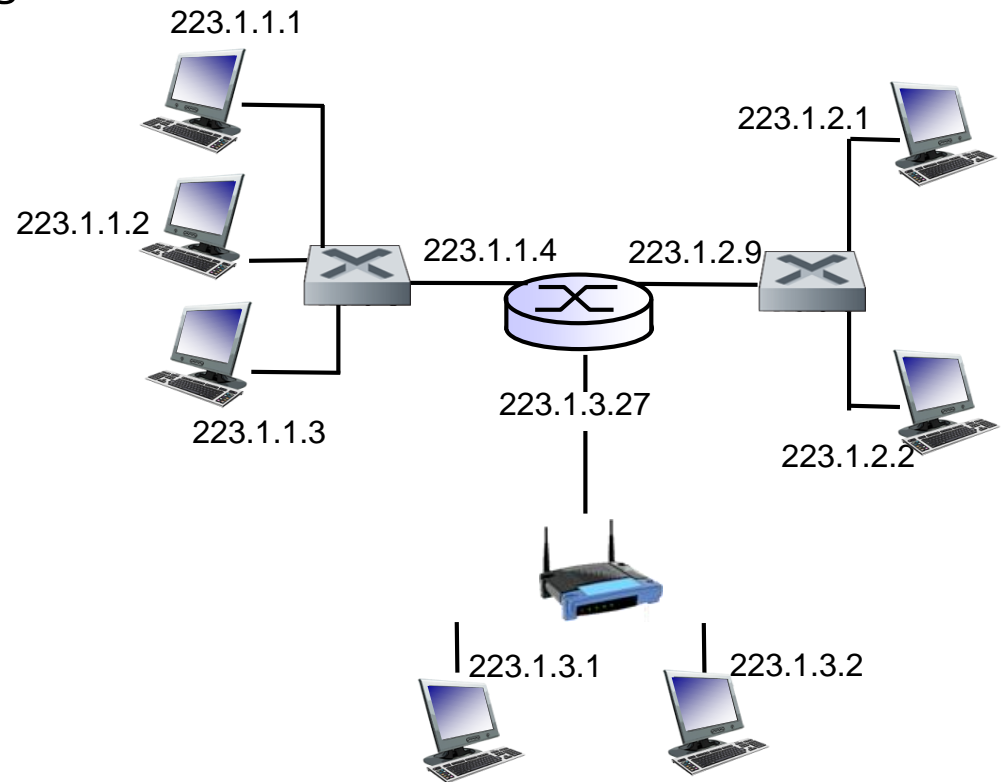
5.8 IPv6

5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

IP Adressen eines Routers

- Ein Router hat mehrere Interfaces (Netzwerkadapter, Port)
 - Interface ist die Verbindung von Host/Router mit physikalischem Link
- Ein Router hat eine IP Adresse pro Interface
 - IP-Adressen der Interfaces eines Routers gehören normalerweise zu unterschiedlichen Netzwerksegmenten



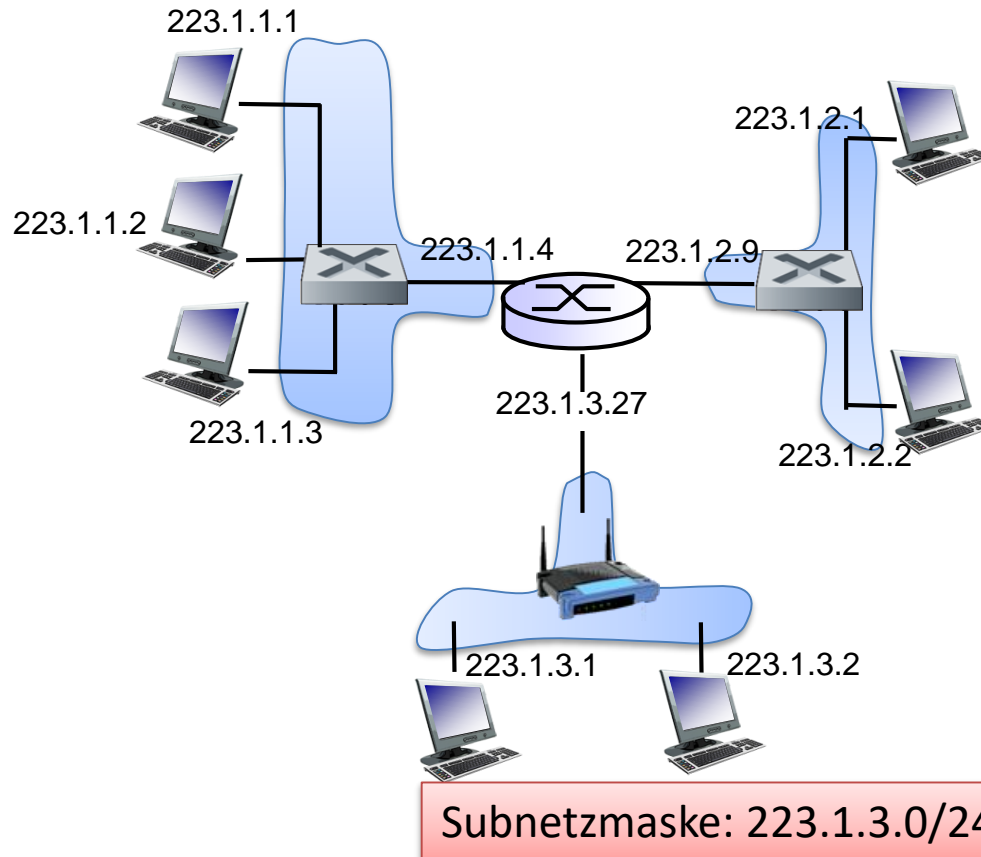
Netzwerksegment:

- Teil des Netzes, in dem alle Rechner (Netzwerkadapter) über die MAC Adresse erreichbar sind
- alle IP Adressen im Netzwerksegment gehören zu einem Subnetz
 - Netzwerksegmente sind die kleinsten Subnetze
 - IP-Adresse und Subnetzmaske eines Rechners (einer Netzwerkkarte) bestimmen das Netzwerksegment des Rechners (der Netzwerkkarte)
 - erste IP-Adresse (Hostteil nur 0en) im Subnetz dient als Netzwerkadresse
 - letzte IP-Adresse (Hostteil nur 1en) im Subnetz dient als Broadcastadresse
- Netzwerksegmente werden durch Router verbunden
 - ein oder mehrere Router befinden sich "am Rand" eines Netzwerksegments
- Zuweisung von IP-Adressen zu Netzwerksegmenten sollte Aggregation von Netzwerksegmenten zu größeren Subnetzen bzgl. der Einträge in Routingtabellen berücksichtigen

Beispiel 1: Netzwerksegmente und deren Subnetzmasken

Subnetzmaske: 223.1.1.0/24

Subnetzmaske: 223.1.2.0/24

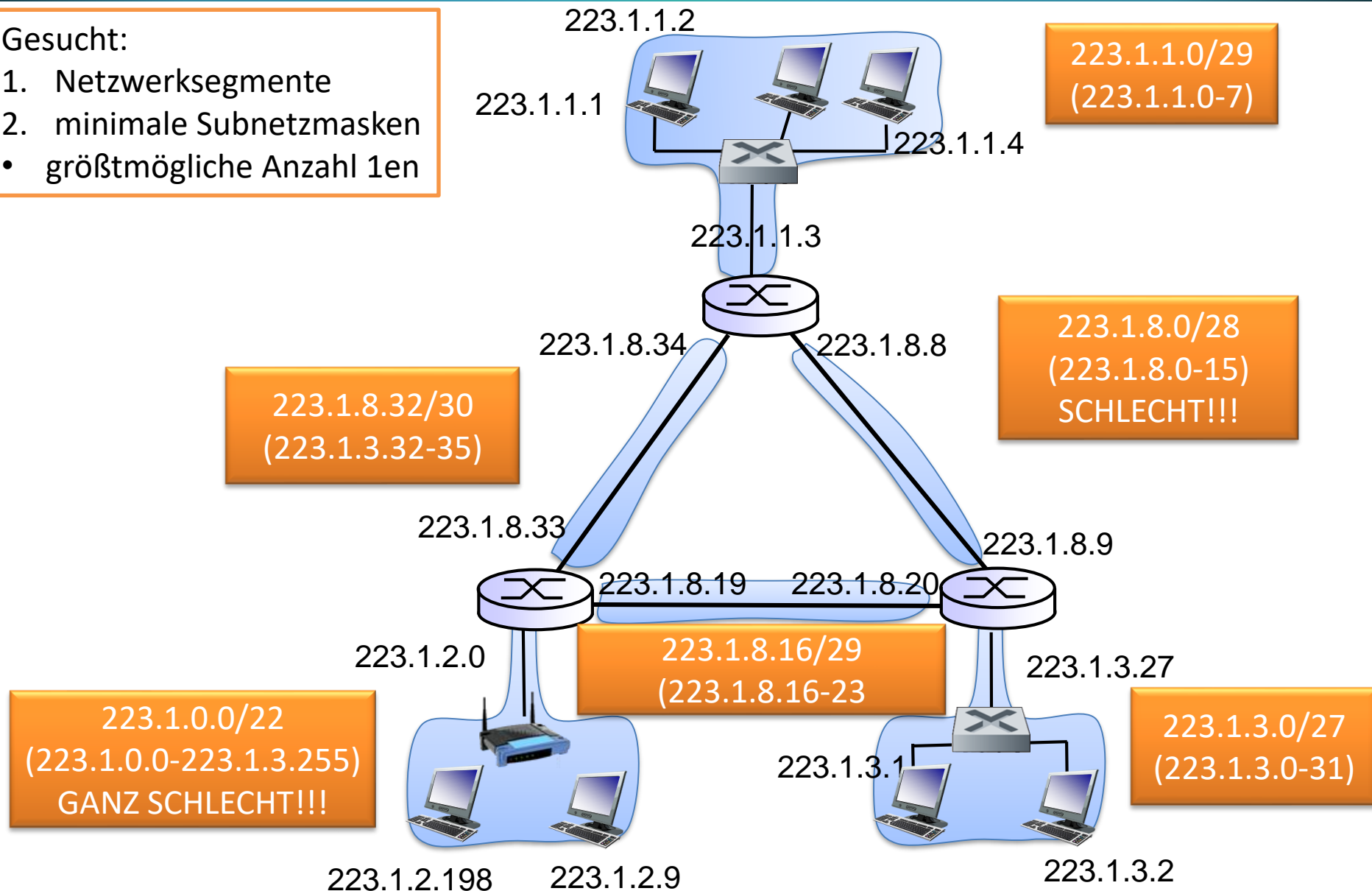


Subnetzmaske: 223.1.3.0/24

Beispiel 2: Netzwerksegmente und deren Subnetzmasken

Gesucht:

1. Netzwerksegmente
 2. minimale Subnetzmasken
- größtmögliche Anzahl 1en



Beispiel 2: Erklärung

Regeln:

- jedes Netzwerksegment beginnt immer bei einem Vielfachen seiner Größe
- Netzwerk- und Broadcast-Adresse des Netzwerksegments dürfen nicht belegt sein
 - nächst größerer Adressbereich, wenn belegt

Netzwerksegment: 223.1.8.33, 223.1.8.34

- Größe: mindestens 4 Adressen → \30
- Adressbereich: 223.1.8.32\30 (ok)

Netzwerksegment: 223.1.1.1, 223.1.1.2, 223.1.1.3, 223.1.1.4

- Größe: mindestens 8 Adressen → \29
- Adressbereich: 223.1.1.0\29 (ok)

Netzwerksegment: 223.1.3.1, 223.1.3.2, 223.1.3.27

- Größe: mindestens 32 Adressen → \27
- Adressbereich: 223.1.3.0\27 (ok)

Beispiel 2: Erklärung

Netzwerksegment: 223.1.8.8, 223.1.8.9

- Größe: mindestens 4 Adressen → 4 Adressen \30
- Adressbereich: 223.1.8.8\30 (geht nicht wegen Netzwerkadresse)
223.1.8.8\29 (geht nicht wegen Netzwerkadresse)
223.1.8.0\28 (ok)

Netzwerksegment: 223.1.8.19, 223.1.8.20

- Größe: mindestens 4 Adressen → 4 Adressen \30
- Adressbereich: 223.1.8.16\30 (geht nicht wegen BC Adresse)
223.1.8.16\29 (ok)

Netzwerksegment: 223.1.2.0, 223.1.2.9, 233.1.2.198

- Größe: mindestens 256 Adressen → \24
- Adressbereich: 223.1.2.0\24 (geht nicht wegen Netzwerkadresse)
223.1.2.0\23 (geht nicht wegen Netzwerkadresse)
223.1.0.0\22 (ok)

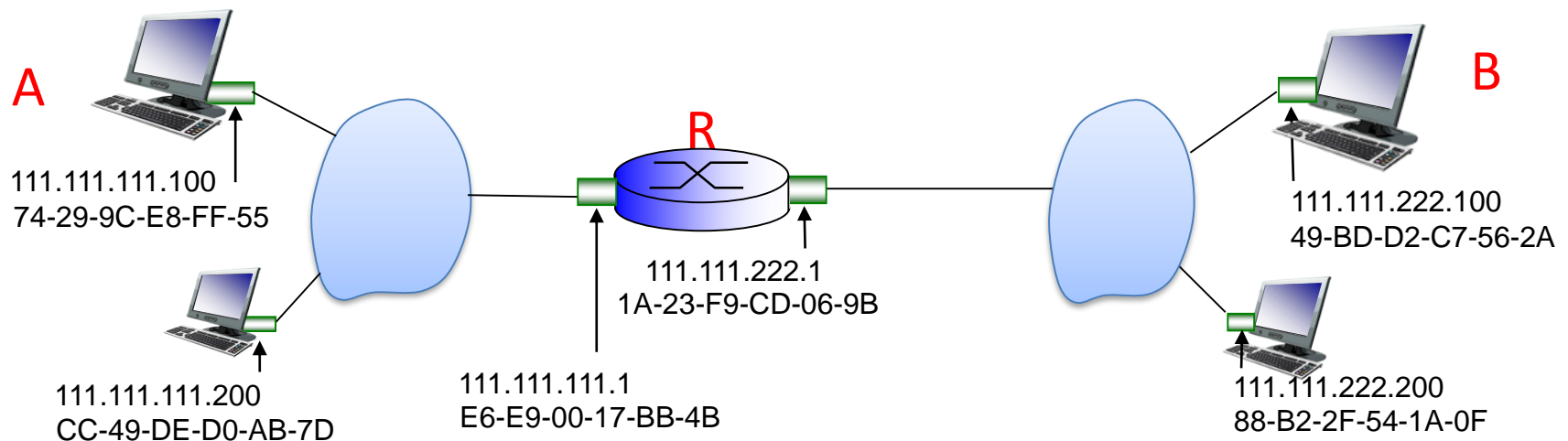
Router verbindet zwei Netzwerksegment (LANs)

Walkthrough:

- Übertragung eines Datagramms von Host A in Netzwerksegment 1 zu Host B in Netzwerksegment 2
- Router R verbindet die beiden Netzwerksegmente

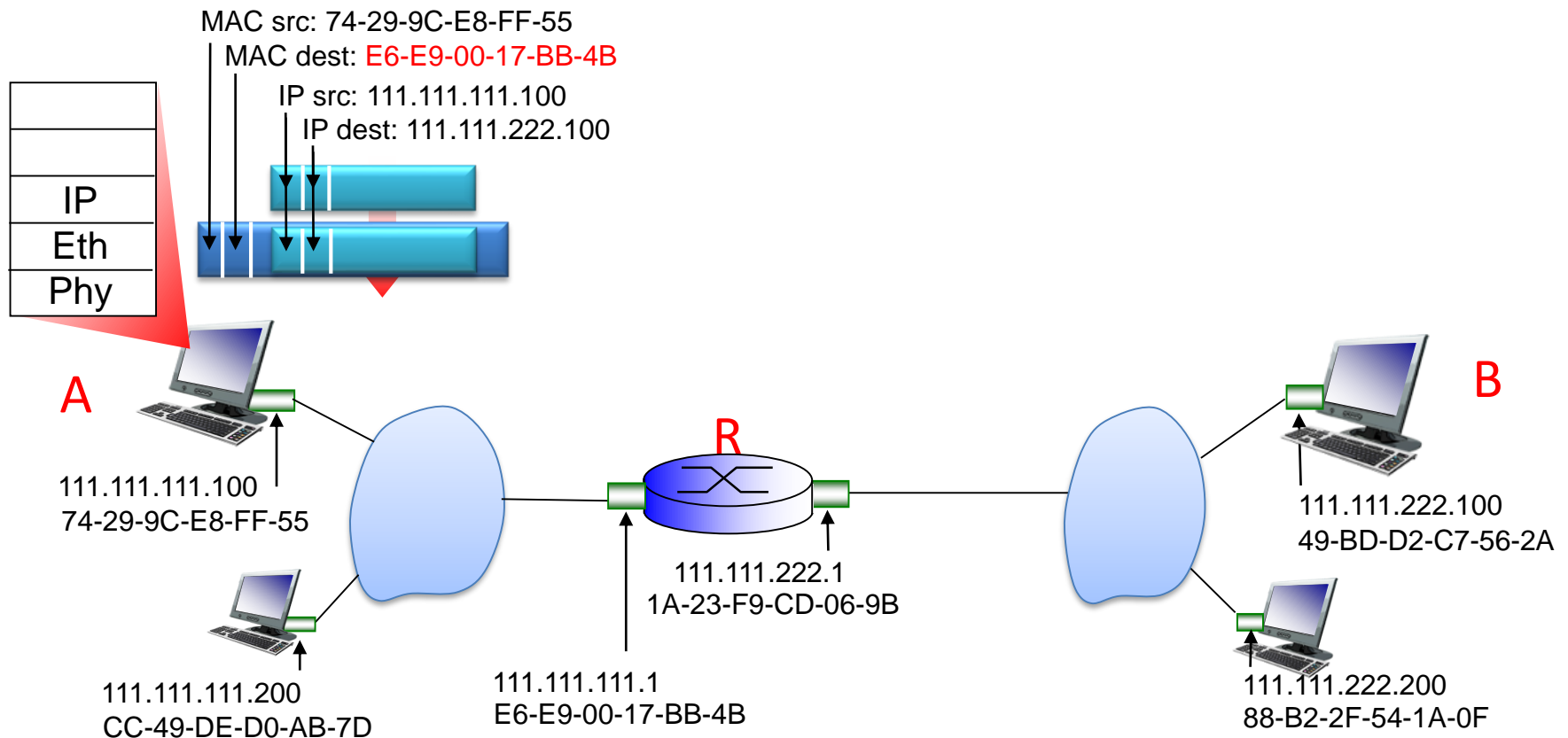
Netzwerksegment 1:
111.111.111.0/24

Netzwerksegment 2:
111.111.222.0/24



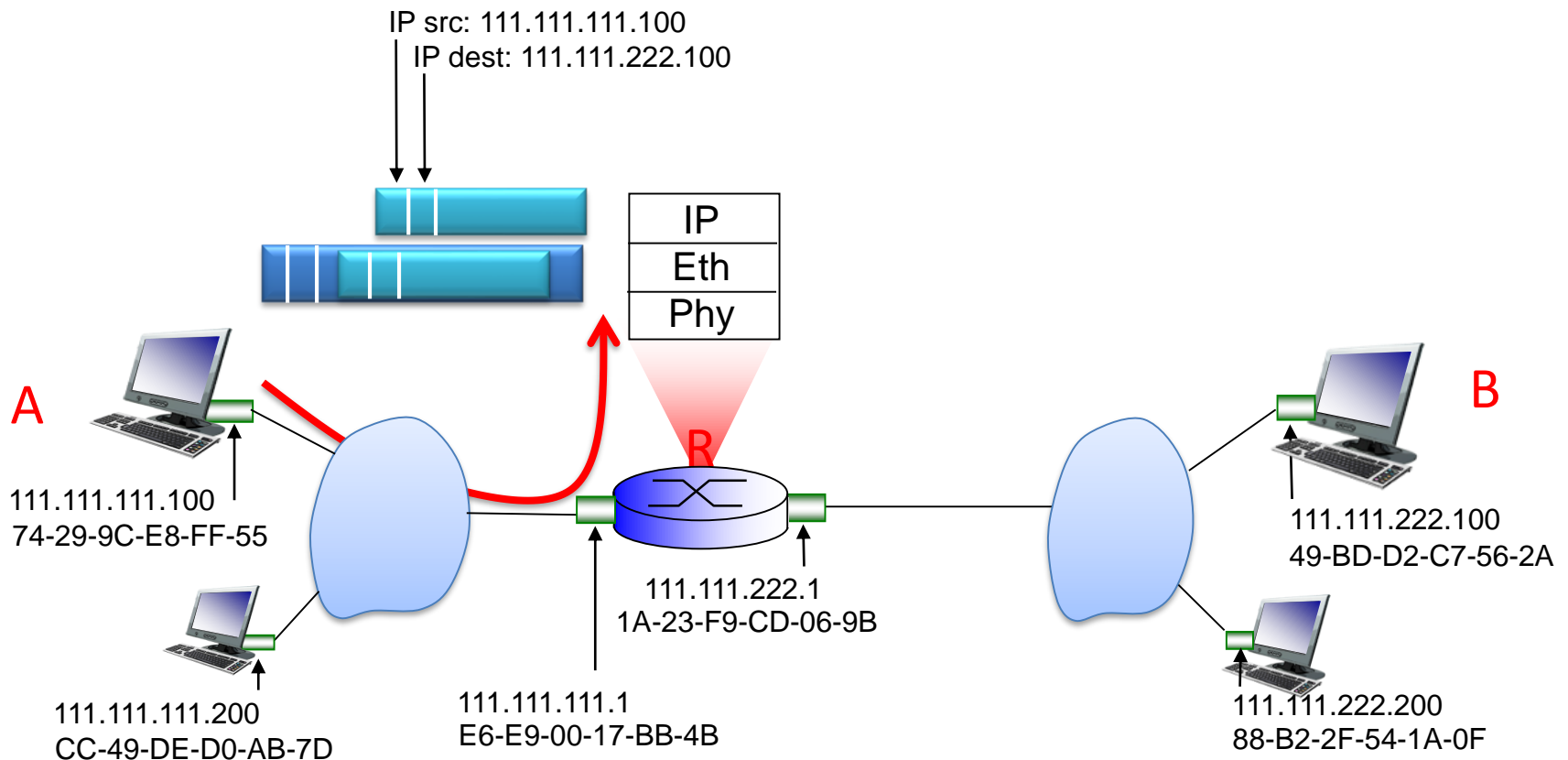
Router verbindet zwei Netzwerksegmente (LANs)

- A erstellt IP Datagramm mit Ziel B
- A packt IP Datagramm in Ethernet-Frame mit MAC-Adresse von R als Ziel ein



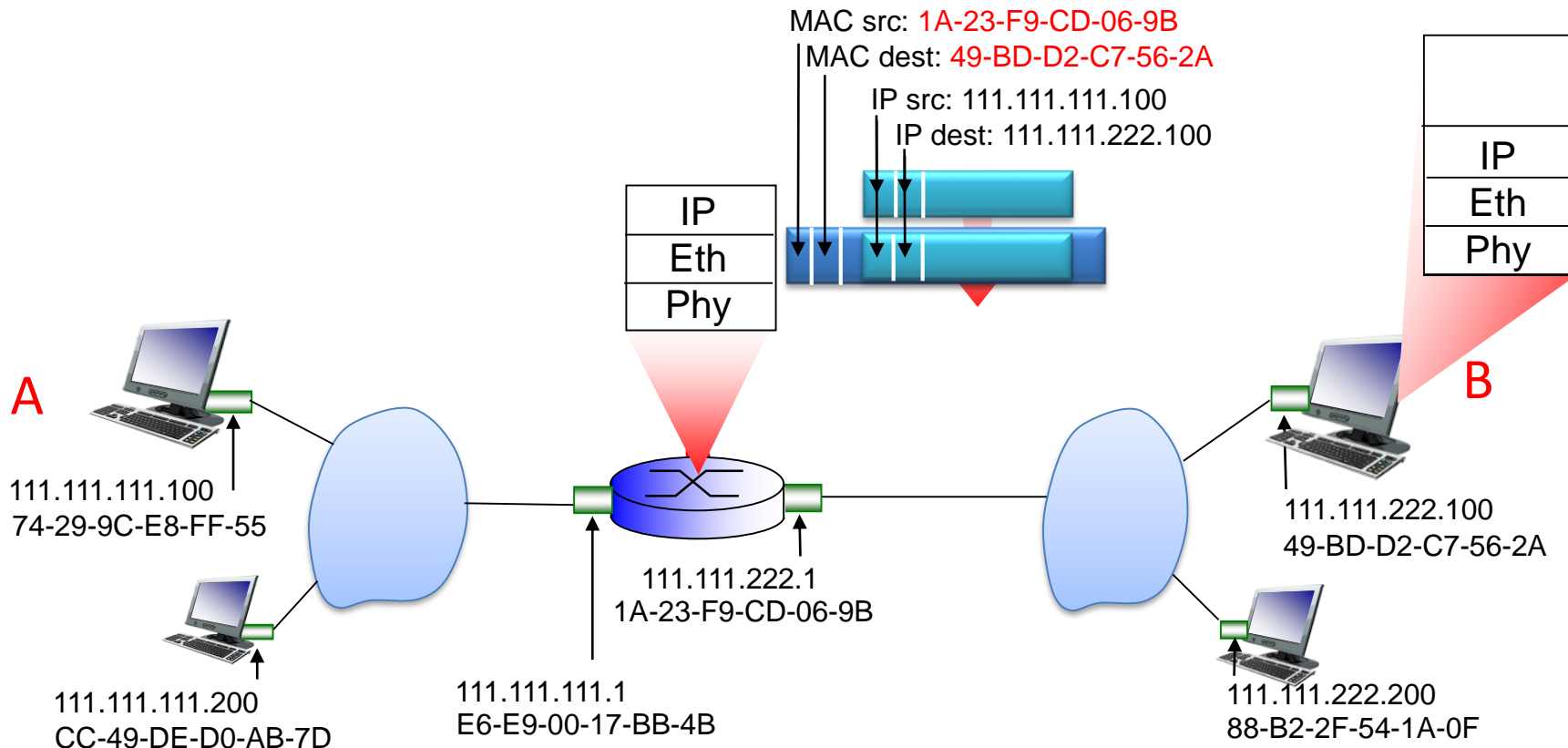
Router verbindet zwei Netzwerksegmente (LANs)

- A schickt Ethernet-Frame an R
- R packet IP Paket aus Ethernet-Frame aus



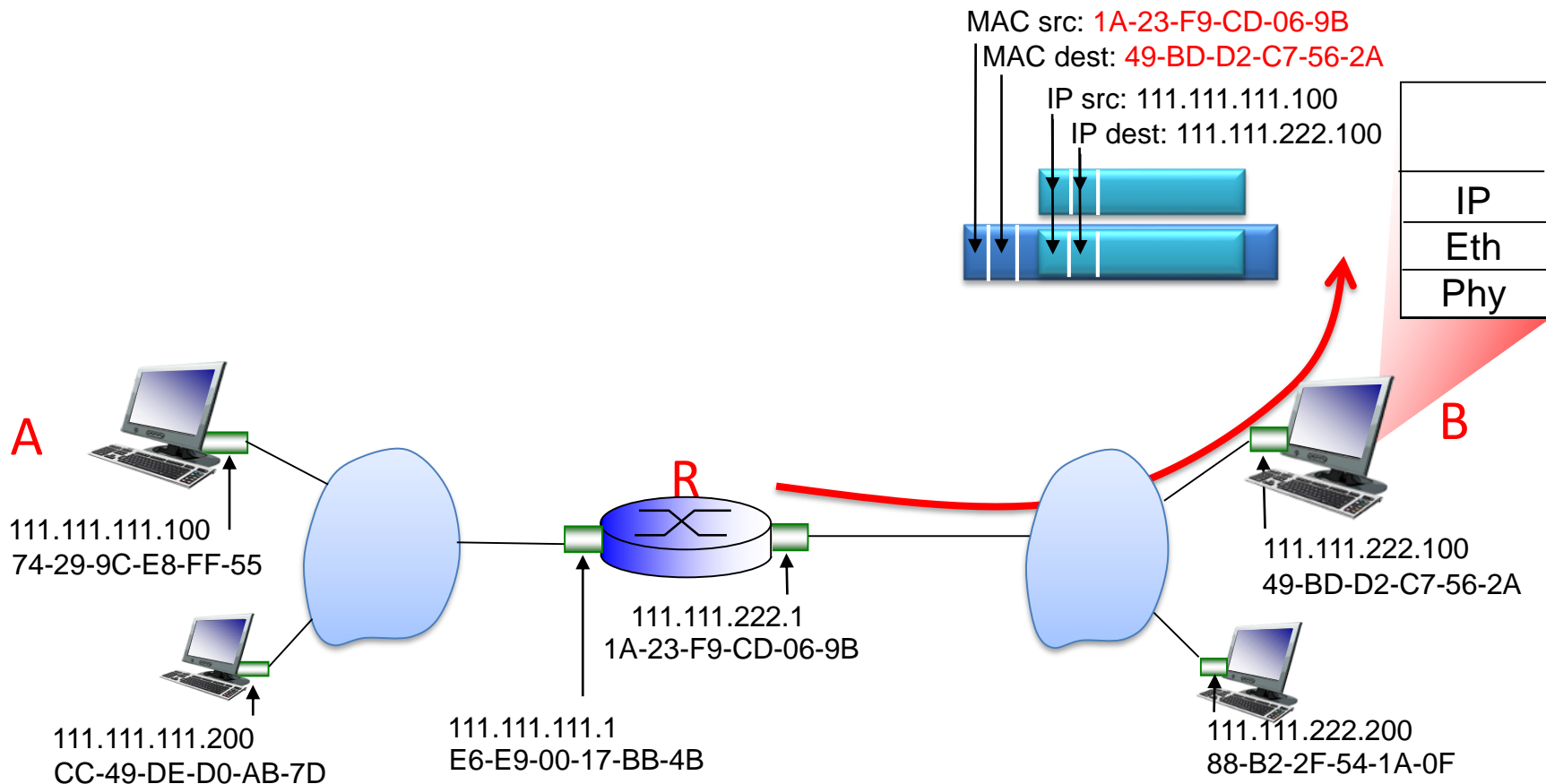
Router verbindet zwei Netzwerksegmente (LANs)

- R schickt das IP Paket an B weiter (Routingtable)
- R packt das IP Paket in einen Ethernet-Frame mit MAC-Adresse von B als Ziel ein



Router verbindet zwei Netzwerksegmente (LANs)

- B empfängt den Ethernet-Frame
- B packt das IP-Paket aus dem Ethernet-Frame aus



5.1 Übersicht

5.2 Adressen

5.3 Lokale Netze: Bridges und Switches

5.4 Intra-Domain Routing

5.4.1 Routing-Protokolle

5.4.2 Subnetze

5.4.2.1 Funktionsweise eines Routers

5.4.2.2 Subnetze und Subnetzmasken

5.4.2.3 Routingtabellen

5.4.2.4 Netzwerksegmente

5.4.2.5 Address Resolution Protocol (ARP)

5.5 Inter-Domain Routing

5.6 Internet Protocol (IP)

5.7 Network Address Translation (NAT)

5.8 IPv6

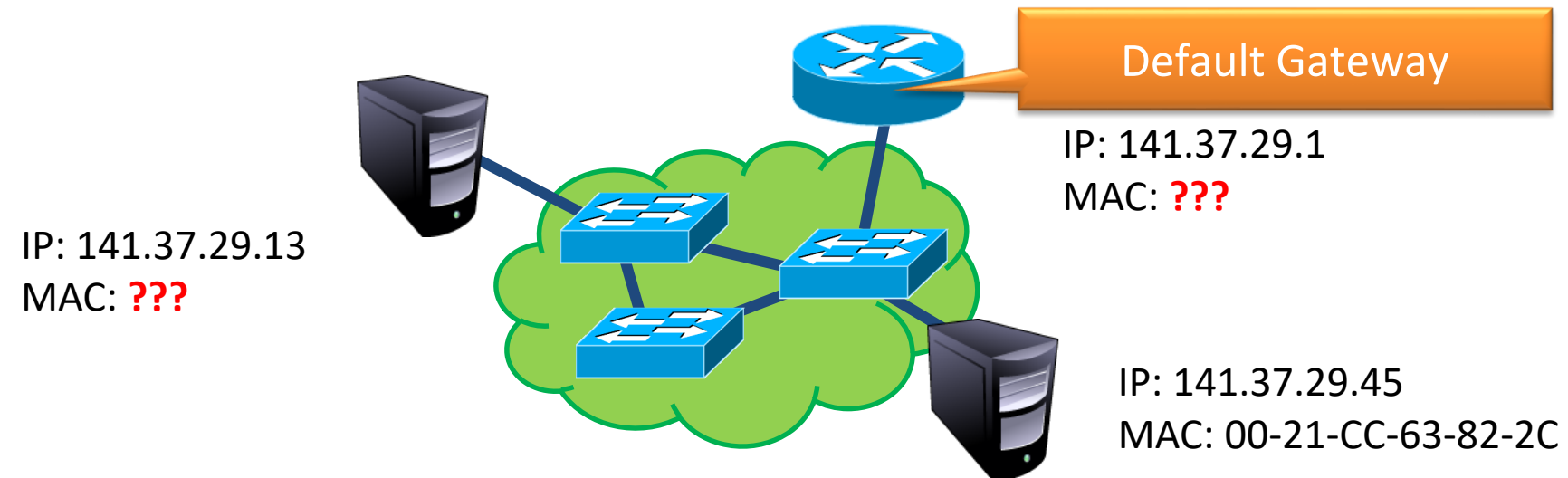
5.9 Mobilitätsunterstützung

5.10 Zusammenfassung

- Verkehrslenkung innerhalb eines Netzwerksegments (LANs) erfolgt auf Schicht 2 (z.B. Ethernet oder WLAN) über die MAC Adresse
 - Zugehörigkeit einer Ziel-IP-Adresse zum eigenen Netzwerksegment erfolgt über eigene IP-Adresse und eigene Subnetzmaske
 - die Routing-Tabelle enthält dazu immer einen Eintrag für das eigene Netzwerksegment mit „On Route“ als nächsten Hop
 - wenn diese Route für die Ziel-IP-Adresse gültig ist, dann ist diese Route spezifischer als die Default-Route zum Gateway und das Pakete wird auf dieser Route direkt (on route) zum Ziel gesendet
- Verkehrslenkung zu Zielen außerhalb des eigenen Netzwerksegments erfolgt über die IP Adresse
 - Ziel der Übertragung innerhalb des Netzwerksegments ist der Router, der als Default-Gateway konfiguriert ist
 - die Route für das eigene Netzwerksegment ist nicht gültig
 - das Default-Gateway wird wiederum über seine MAC Adresse erreicht

Aufgabe von ARP

- Problem:
 - Übertragung zu einer IP Adresse mit unbekannter MAC Adresse
 - Konfiguration im Netz und Einträge in Routingtabellen über IP Adresse nicht über MAC Adresse → DHCP
 - Übertragungen im eigenen Netzwerksegment erfordern aber die Ziel-MAC-Adresse
- Lösung
 - Ermittlung der MAC-Adresse zu einer IP-Adresse im eigenen Netzwerksegment über das Adress-Resolution Protocol (ARP)



- A möchte ein Paket an B schicken, die MAC-Adresse von B ist nicht in der ARP-Tabelle von A
- A schickt eine ARP-Query als Broadcast-Frame mit der IP-Adresse von B als Query
 - Empfänger-MAC-Adresse:
FF-FF-FF-FF-FF-FF (broadcast)
 - alle Systeme im Netzwerksegment (LAN) erhalten diese Anfrage
- Switch leitet ARP Querys auf allen Ports (des VLANs) weiter
 - Switches verbreiten ARP Querys im ganzen Netzwerksegment
 - mit ARP werden alle lokalen IP Adressen (lokal heißt aus dem gleichen Netzwerksegment) aufgelöst
 - jeder Rechner im Netzwerksegment muss die ARP Querys erhalten
- B empfängt die ARP-Query, erkennt seine IP-Adresse und antwortet A mit seiner eigenen MAC-Adresse
 - Empfänger-MAC-Adresse = MAC-Adresse von A
- ARP ist eines der Probleme in großen LANs (Netzwerksegmenten), in dem die Rechner viel intern kommunizieren und zahlreiche lokale Adressen auflösen müssen
 - spezielle Probleme treten auf wenn z.B. ein Server nicht mehr auf ARP Querys antwortet und diese dann von vielen Rechnern kontinuierlich gesendet werden

ARP Query

The screenshot shows the Wireshark 1.12.1 interface with the filter 'arp' applied. The packet list shows several ARP packets, with packet 6383 selected. The packet details pane shows the structure of the selected ARP request packet.

No.	Time	Source	Destination	Protocol	Length	Info
6284	246.530849000	72:dd:d9:7c:59:c3	Broadcast	ARP	60	who has 141.37.28.254? Tell 141.37.31.226
6291	249.559135000	4a:0c:8d:2b:85:61	Broadcast	ARP	60	who has 141.37.31.55? Tell 141.37.31.227
6308	249.689653000	ca:57:ee:08:bb:48	Broadcast	ARP	60	who has 141.37.28.254? Tell 141.37.29.140
6383	250.906950000	Flextron_63:82:2c	Broadcast	ARP	42	who has 141.37.29.1? Tell 141.37.29.93
6384	250.907209000	Hewlett-_e4:dc:47	Flextron_63:82:2c	ARP	60	141.37.29.1 is at 00:19:bb:e4:dc:47
6449	253.575521000	Hewlett-_98:10:45	Broadcast	ARP	60	who has 141.37.31.3? Tell 141.37.31.15
6491	255.907142000	Hewlett-_e4:dc:47	Flextron_63:82:2c	ARP	60	who has 141.37.29.93? Tell 141.37.29.1
6492	255.907177000	Flextron_63:82:2c	Hewlett-_e4:dc:47	ARP	42	141.37.29.93 is at 00:21:cc:63:82:2c
6542	258.346363000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10
6546	259.343391000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10
6599	260.343354000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10
6689	261.345988000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10

Frame 6383: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0

- Ethernet II, Src: Flextron_63:82:2c (00:21:cc:63:82:2c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Source: Flextron_63:82:2c (00:21:cc:63:82:2c)
 - Type: ARP (0x0806)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (1)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (1)
 - Sender MAC address: Flextron_63:82:2c (00:21:cc:63:82:2c)
 - Sender IP address: 141.37.29.93 (141.37.29.93)
 - Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
 - Target IP address: 141.37.29.1 (141.37.29.1)

File: "C:\Users\dstaehle\AppData\Local\Te... Packets: 6... Profile: Default

ARP Response

Wireshark 1.12.1 (v1.12.1-0-g01b65bf from master-1.12)

Filter: **arp** Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
6284	246.530849000	72:dd:d9:7c:59:c3	Broadcast	ARP	60	who has 141.37.28.254? Tell 141.37.31.226
6291	249.559135000	4a:0c:8d:2b:85:61	Broadcast	ARP	60	who has 141.37.31.55? Tell 141.37.31.227
6308	249.689653000	ca:57:ee:08:bb:48	Broadcast	ARP	60	who has 141.37.28.254? Tell 141.37.29.140
6383	250.906950000	Flextron_63:82:2c	Broadcast	ARP	42	who has 141.37.29.1? Tell 141.37.29.93
6384	250.907209000	Hewlett_e4:dc:47	Flextron_63:82:2c	ARP	60	141.37.29.1 is at 00:19:bb:e4:dc:47
6449	253.575521000	Hewlett_e4:dc:47	Broadcast	ARP	60	who has 141.37.31.3? Tell 141.37.31.15
6491	255.907142000	Hewlett_e4:dc:47	Flextron_63:82:2c	ARP	60	who has 141.37.29.93? Tell 141.37.29.1
6492	255.907177000	Flextron_63:82:2c	Hewlett_e4:dc:47	ARP	42	141.37.29.93 is at 00:21:cc:63:82:2c
6542	258.346363000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10
6546	259.343391000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10
6599	260.343354000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10
6689	261.345988000	d2:f5:4d:15:37:af	Broadcast	ARP	60	who has 141.37.29.42? Tell 141.37.29.10

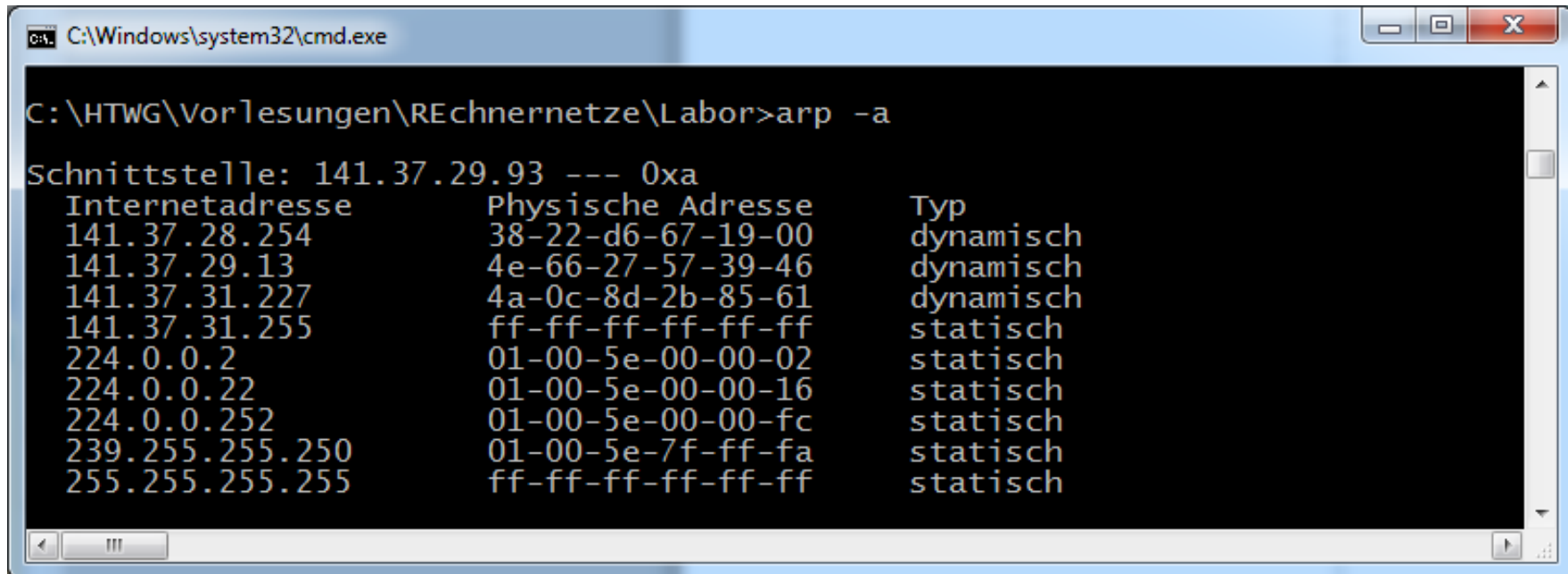
Frame 6384: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

- Ethernet II, Src: Hewlett_e4:dc:47 (00:19:bb:e4:dc:47), Dst: Flextron_63:82:2c (00:21:cc:63:82:2c)
 - Destination: Flextron_63:82:2c (00:21:cc:63:82:2c)
 - Source: Hewlett_e4:dc:47 (00:19:bb:e4:dc:47)
 - Type: ARP (0x0806)
 - Padding: 00000000000000000000000000000000
- Address Resolution Protocol (reply)
 - Hardware type: Ethernet (1)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - opcode: reply (2)
 - Sender MAC address: Hewlett_e4:dc:47 (00:19:bb:e4:dc:47)
 - Sender IP address: 141.37.29.1 (141.37.29.1)
 - Target MAC address: Flextron_63:82:2c (00:21:cc:63:82:2c)
 - Target IP address: 141.37.29.93 (141.37.29.93)

File: "C:\Users\dstaehle\AppData\Local\Te... Packets: 6... Profile: Default

ARP Cache

- Jeder IP-Knoten (Host und Router) verwaltet einen ARP Cache, der Paare von IP und MAC Adressen inklusive einer Gültigkeitsdauer (TTL) enthält
 - < IP address; MAC address; TTL>
- Anzeigen des ARP Caches in Windows: *arp -a*



```
C:\Windows\system32\cmd.exe

C:\HTWG\Vorlesungen\REchnernetze\Labor>arp -a

Schnittstelle: 141.37.29.93 --- 0xa
Internetadresse      Physische Adresse    Typ
141.37.28.254        38-22-d6-67-19-00    dynamisch
141.37.29.13         4e-66-27-57-39-46    dynamisch
141.37.31.227        4a-0c-8d-2b-85-61    dynamisch
141.37.31.255        ff-ff-ff-ff-ff-ff    statisch
224.0.0.2            01-00-5e-00-00-02    statisch
224.0.0.22          01-00-5e-00-00-16    statisch
224.0.0.252         01-00-5e-00-00-fc    statisch
239.255.255.250     01-00-5e-7f-ff-fa    statisch
255.255.255.255     ff-ff-ff-ff-ff-ff    statisch
```

- Subnetze
 - Subnetze sind IP-Adressbereiche, die über eine Adresse plus Subnetzmaske festgelegt werden
 - die Subnetzmaske unterteilt eine Adresse in Subnetzteil und Hostteil
 - Ziele in Routing-Tabellen sind Adressbereiche, die eben über eine Adresse und eine Subnetzmaske angegeben werden
 - kleine Routing-Tabellen durch geeignete Strukturierung von Subnetzen und Netzwerksegmenten
- Netzwerksegmente
 - Netze bestehen aus Netzwerksegmenten, die die kleinsten Subnetze bilden
 - alle Knoten in einem Netzwerksegment können sich über eine Layer-2-Technologie wie z.B. über Ethernet und die MAC-Adresse erreichen
 - ein Knoten kann über seine IP-Adresse und seine Subnetzmaske bestimmen, ob er einen Knoten direkt oder über seinen Default-Router erreicht
- ARP
 - Das Address Resolution Protocol dient dazu, MAC Adressen zu IP Adressen im eigenen Netzwerksegment zu bestimmen