Rechnernetze & Verteilte Systeme

Ludwig-Maximilians-Universität München

Sommersemester 2019

Prof. Dr. D. Kranzlmüller

MNM TEAM

Kapitel 5: Vermittlungsschicht

(Engl. Network Layer)

MNM TEAM

Inhalt von Kapitel 5

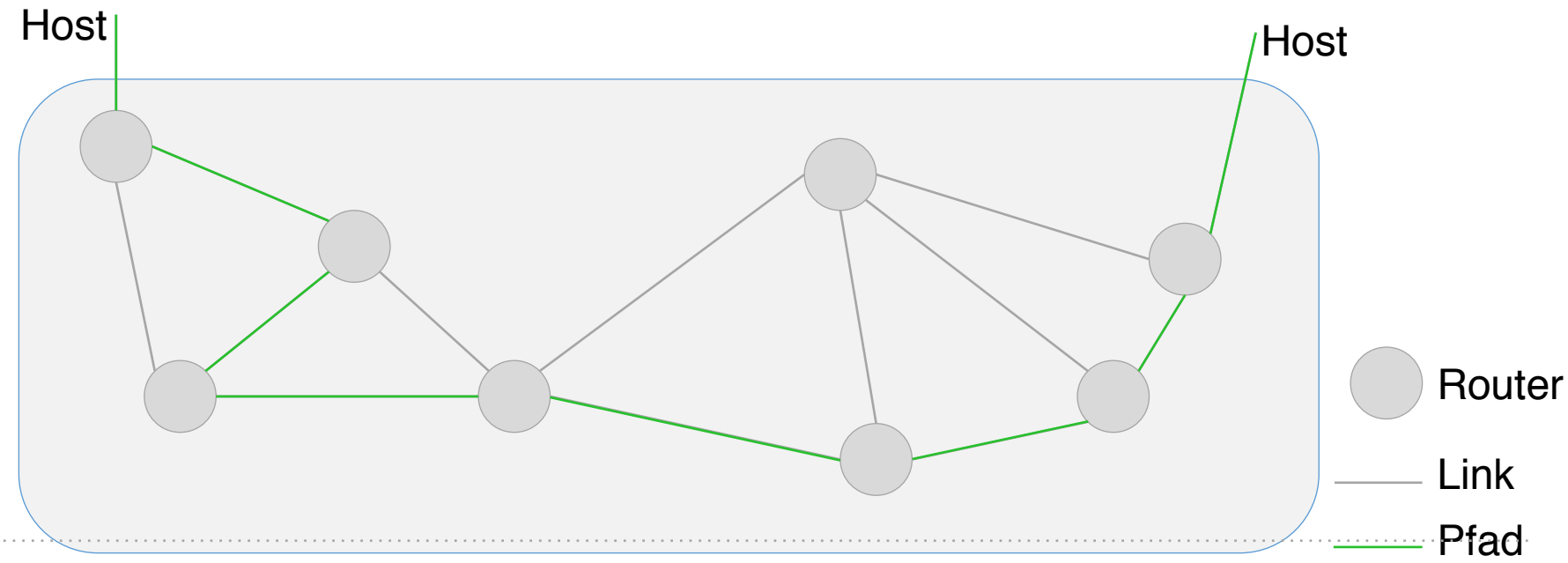
1. Ziele und Rahmenbedingungen
2. Wegewahl auf anderen Schichten
3. Vermittlung (Pfadschaltung)
4. Wegewahl (Routing)
5. Internet Protokoll (v4)
6. Internet Protokoll (v6)

Einordnung von Kapitel 5

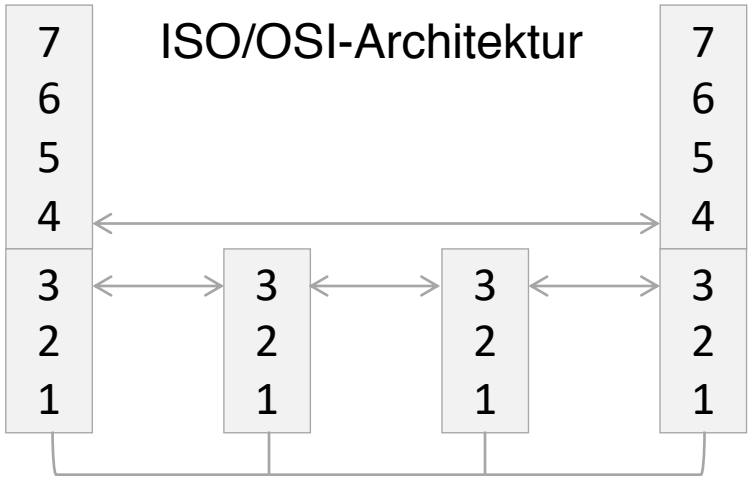
- Im Kapitel 4 haben wir uns mit der Transportschicht (Schicht 4) beschäftigt: Netzunabhängiger Transport von Nachrichten zwischen Endsystemen
- Kapitel 5: Vermittlungsschicht: Übertragung von Paketen von der Quelle zum Ziel
- Vermittlung (Pfadschaltung) und Wegewahl (Routing) als Dienst
- Internet Protokoll (IP): v4 und v6

Ziele und Rahmenbedingungen der Vermittlungsschicht

Wegewahl, Dienstgüte, Verbindungsart, Sicherungsschicht als Rahmenbedingung



Hauptziel: Pfadschaltung und Wegewahl



Dienst: Verbindungslos oder verbindungsorientiert? (1/2)

- Prinzipiell: Auch auf der Vermittlungsschicht verbindungslos oder verbindungsorientiert
- Bei einem verbindungsorientierten Vermittlungsprotokoll müssen sich die Netzknoten den Pfad, den Daten einer gegebenen Verbindung nehmen sollen, merken.

  ➔Alle Pakete der Verbindung werden auf demselben Pfad übertragen.
- Beispiel für verbindungsorientierte Vermittlung: Telefonnetz

Dienst: Verbindungslos oder verbindungsorientiert? (2/2)

- Im Internet: Verbindungsloses Internet Protokoll
  - IP-Pakete werden unabhängig voneinander befördert.
  - Zwischenknoten müssen sich keine Zustandsinformationen merken.
- Vorteil: Beim Ausfall einzelner Netzknoten werden Pakete einfach umgeleitet (anstatt alle Verbindungen, die über den Knoten laufen, neu aufzubauen).

Verbindungsart und Ende-zu-Ende Argument

- Entscheidung für verbindungslosen Ansatz im Internet als Folge der Ende-zu-Ende Argumentation
- Wenn Verbindungen als inhärent unzuverlässig angesehen werden und verbindungsorientierte Aufgaben bereits auf der Transportschicht übernommen (TCP) werden, gibt es kaum Vorteile dies auf der Vermittlungsschicht erneut zu implementieren
- Nachteil: Realisierung von Dienstgüte schwierig (z.B.: für Echtzeitdatenverkehr wie Sprache und Video)
  - ➔„Unter der Oberfläche“ entwickelt das Internet durchaus verbindungsorientierte Eigenschaften (Bsp.: VLAN – Schicht 2)

Schicht 3 Dienstgüte (Engl. Quality of Service – QoS)

- Dienstgüte auf Vermittlungsschicht gewinnt an Bedeutung: Streaming-Dienste und IP-Telefonie
- Schicht 3 Dienstgüteparameter:
  - Verbindungsaufbauwahrscheinlichkeit (Blockierwahrscheinlichkeit)
  - Verbindungsaufbauzeit
  - Durchsatz der Schicht-3-Verbindung
  - Nachrichtenübertragungszeit
  - Schwankung in der Übertragungszeit (engl.: Jitter)
  - Restfehlerrate

Die Sicherungsschicht als Rahmenbedingung (1/2)

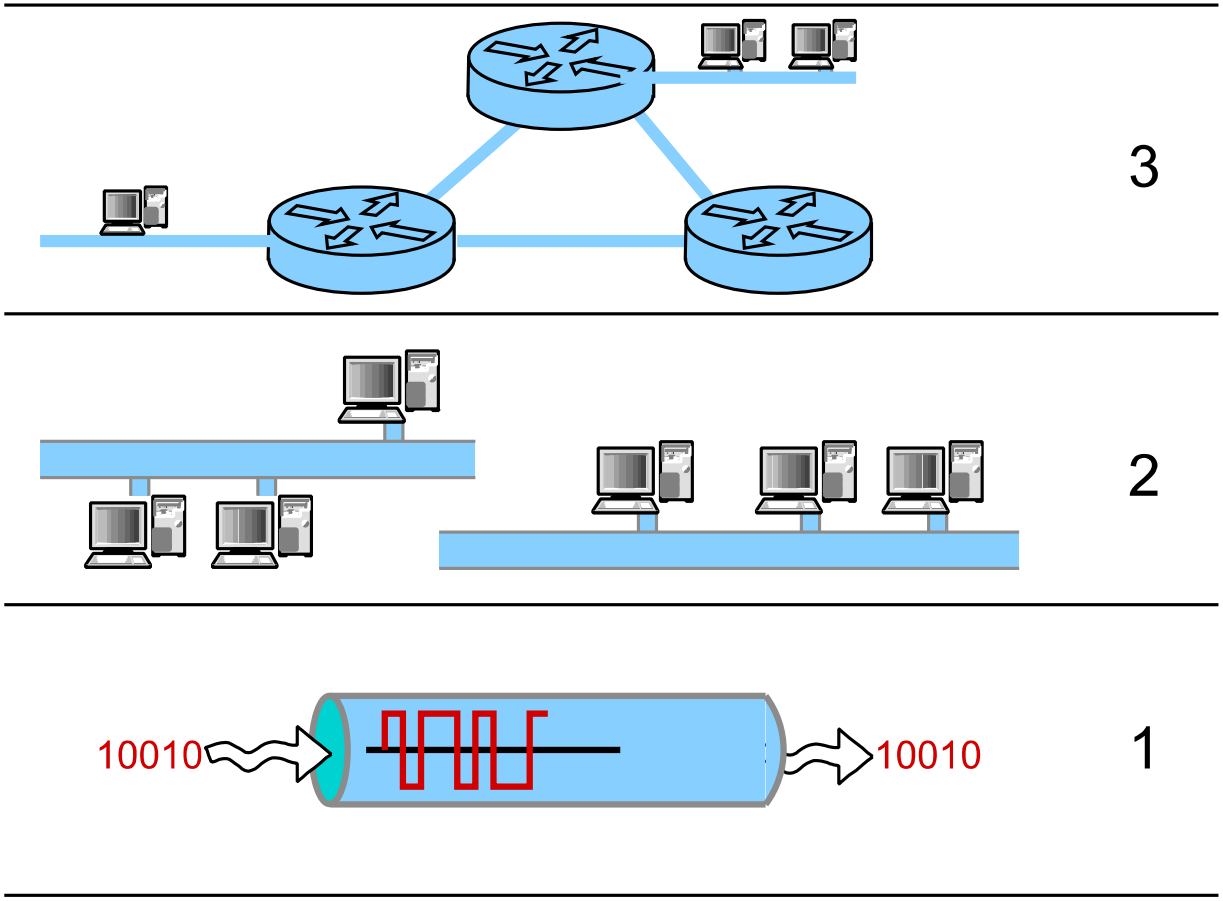
- Vermittlungsschicht bietet ihre Dienste *unter Verwendung der Dienste der nächst tieferliegenden Schicht* an: Schicht 2: Sicherungsschicht
- Verweis auf Kapitel 7 (Hardwarenahe Schichten):

  Übertragung über ein einzelne Wegabschnitte (von einem Knoten über ein Medium zum nächsten Knoten) als *gesicherten Data Link/Logical Link.*

Die Sicherungsschicht als Rahmenbedingung (2/2)

- Übertragung über einzelne Wegabschnitte (von einem Knoten über ein Medium zum nächsten Knoten) als gesicherten Data Link/Logical Link.
  - Medien- und Übertragungstechnik unabhängig
  - Bits zu Blöcken/Rahmen (engl. Frames) zusammengefasst.
  - Fehlererkennung (und ggf. Korrektur) der Rahmenübertragung
- Vermittlungsschicht setzt also voraus, dass Nachrichtenpakete (bis zu einer gewissen Größe – Stichwort MTU) fehlerfrei an Nachbarknoten weitergeleitet werden können.

Netzbildung



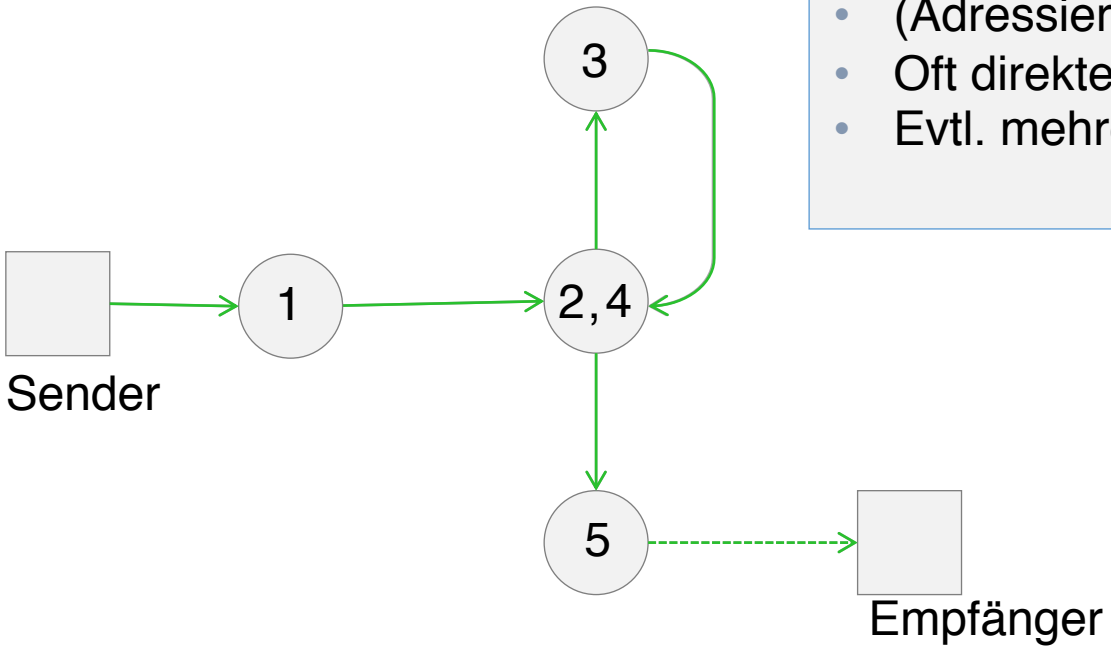
Kapitel 5.1 Wegewahl auf anderen Schichten

E-Mail, Bridges und Switches, IP-Router

Einordnung/Motivation

- Wegewahl und Vermittlung können (technologieabhängig) auch auf anderen Schichten vorkommen (schichtunabhängiges Konzept).
- Überblick anhand von drei kleinen Beispielen:
  - Internet-Schicht 5: E-Mail-Relays
  - Schicht 3: IP-Router
  - Schicht 2: Bridges und Switches

Beispiel: Anwendungsschicht: E-Mail-Relay (ISO/OSI-Schicht 7)



Beispiel: Vermittlung von E-Mails

- Wegewahl anhand von DNS
- (Adressierung: Domain Name)
- Oft direkte Zustellung
- Evtl. mehrere Hops

E-Mail-Relays in der Praxis

Received: from mailrelay1.lrz-muenchen.de (mailrelay1.lrz-muenchen.de [129.187.254.106])
by **mail.nm.ifi.lmu.de** (8.12.11/8.12.11/Linux MNM 0.1) with ESMTP id l9CAC5v7004532
for <danciu@nm.ifi.lmu.de>; Fri, 12 Oct 2007 16:16:05 +0200
Received: from lxmhs06.lrz-muenchen.de (lxmhs06.lrz-muenchen.de [10.156.6.203]) by **mailrelay1.lrz-muenchen.de** with ESMTP for danciu@nm.ifi.lmu.de; Fri, 12 Oct 2007 16:16:04 +0200
Received: from mailrelay1.lrz-muenchen.de ([10.156.6.201])
by **lxmhs06.lrz-muenchen.de** (lxmhs06.lrz-muenchen.de [10.156.6.203]) (amavisd-new, port 10024)
with ESMTP id mDqRYsUc8VAp for <danciu@nm.ifi.lmu.de>;
Fri, 12 Oct 2007 16:16:03 +0200 (CEST)
Received: from mail.gmx.net (mail.gmx.net [213.165.64.20]) by **mailrelay1.lrz-muenchen.de** for danciu@nm.ifi.lmu.de; Fri, 12 Oct 2007 16:16:02 +0200
Received: (qmail invoked by alias); 12 Oct 2007 10:11:01 -0000
Received: from yyyyyyyyyy.dip.t-dialin.net (EHLO ASMCore2Duo) [217.238.48.17x]
by **mail.gmx.net** (mp042) with SMTP; 12 Oct 2007 16:15:01 +0200
From: "NN" <xxxxxx@gmx.de>
To: "'Vitalian A. Danciu'" <danciu@nm.ifi.lmu.de>
Subject: Frage Rechnernetze 1

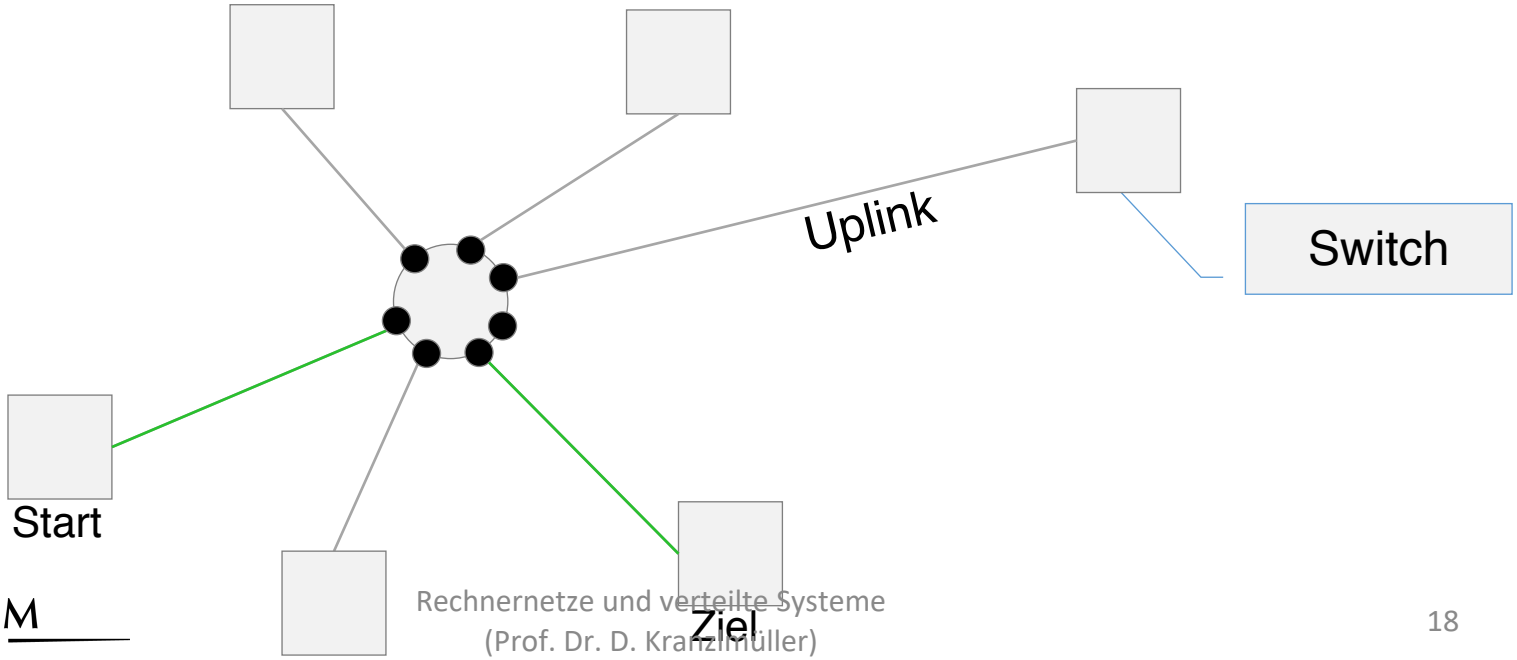
TEXT

Achtung!
Verwechslungsgefahr
Ports auf Schicht 4

Beispiel: Sicherungsschicht: Switch (Schicht 2)

Beispiel: Wegewahl in einem Switch (Schicht 2)

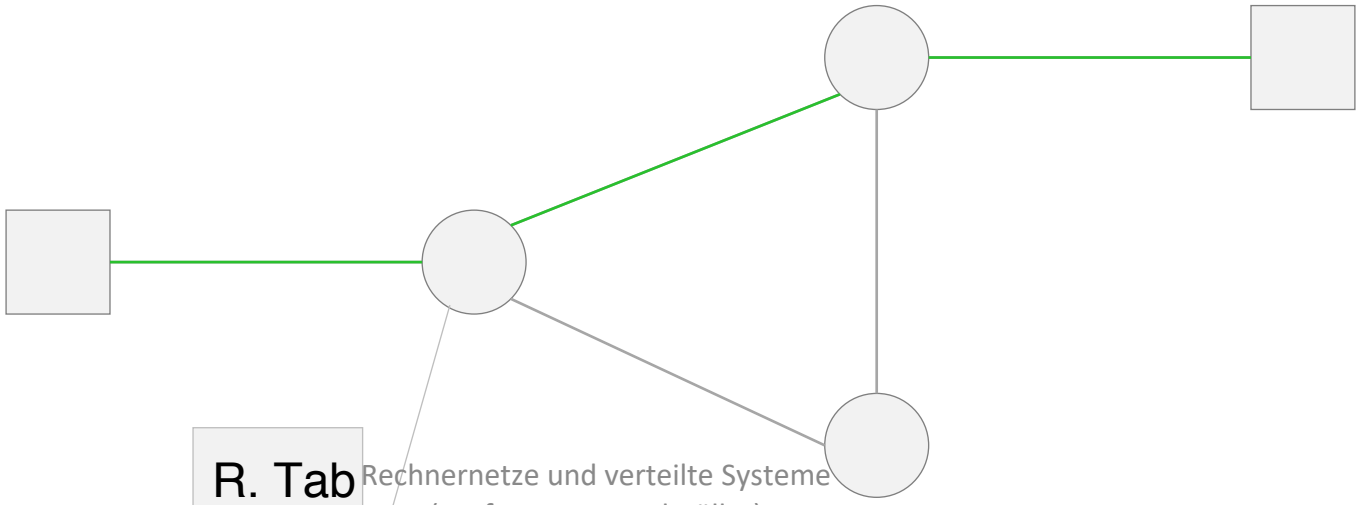
- Ports: Anschlussstellen (Ein-/Ausgänge) für ein Medium
- Switch findet Zuordnung (Adressierung: MAC-Adresse)
  - anhand von Broadcast
  - anhand von bereits beobachtetem Netzverkehr
- Switch erlernt Zuordnung



Beispiel: Vermittlungsschicht: IP-Pakete (Schicht 3)

Beispiel: Wegewahl bei IP-Paketen (Schicht 3)

- Router trifft Entscheidung zur Weiterleitung von Paketen
- Weg bestimmt durch (Adressierung: IP-Adresse)
  - Regeln in Routingtabellen
  - Angaben des Senders (selten)
- Routingprotokolle: zum Aushandeln optimierter Wege
- Nach technischen Gesichtspunkten
- Nach organisatorischen/finanziellen Gesichtspunkten



Kapitel 5.3: Vermittlung (Wiederholung)

Leitungsvermittlung, Nachrichtenvermittlung, Paketvermittlung

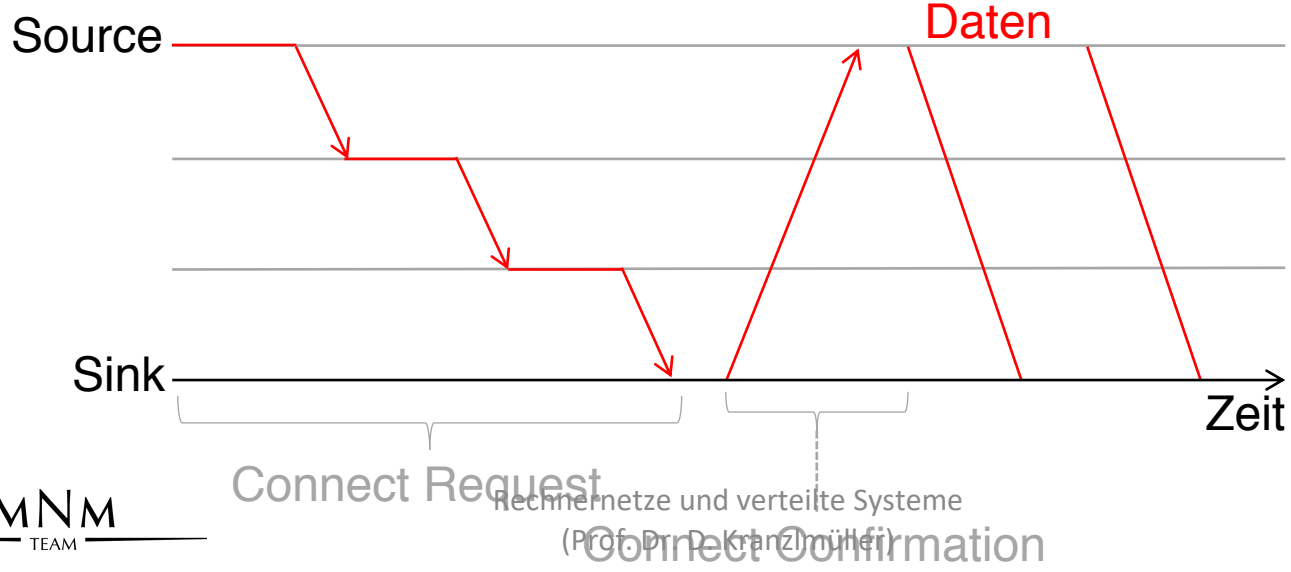
Einordnung/Motivation

- Wegewahl (Routing): Finde anhand von (netzglobalen) Adressen einen Pfad durch das Netz
- Vermittlung (Pfadschaltung): Übertragung der Daten über die einzelnen Wegstücke für einen Ende-zu-Ende Pfad (Beispiele: als eine große Nachricht oder aufgeteilt in viele kleine Pakete).
- Vermittlungsverfahren:
  - Leitungsvermittlung
  - Nachrichtenvermittlung
  - Paketvermittlung

Leitungsvermittlung (Engl. Circuit Switching)

Leitungsvermittlung, Durchschaltung, engl. circuit switching

- Vor Übertragung: Aufbau des Pfades von Sender zu Empfänger
- Dedizierter Kanal für die gesamte Datenphase
- typisch ist gleicher Grenzdurchsatz für alle Links
- typisch gekoppelt mit verbindungsorientiertem Dienst
- Beispiel: *POTS, ISDN-B-Kanäle, Telefonnetz (früher)*



Vor- und Nachteile der Leitungsvermittlung

Nachteile

- Ende-zu-Ende Pfad muss vollständig eingerichtet sein, bevor Daten übertragen werden können
- Bricht eine Vermittlungseinrichtung zusammen, gehen auch alle Verbindungen, die darüber laufen, verloren.
- Wenn Kanal vollständig vergeben ist, kann er von sonst niemandem mehr verwendet werden (Besetztzeichen).

Vorteile

- Ist der Pfad gefunden, besteht ein dedizierter Ende-zu-Ende Kanal.

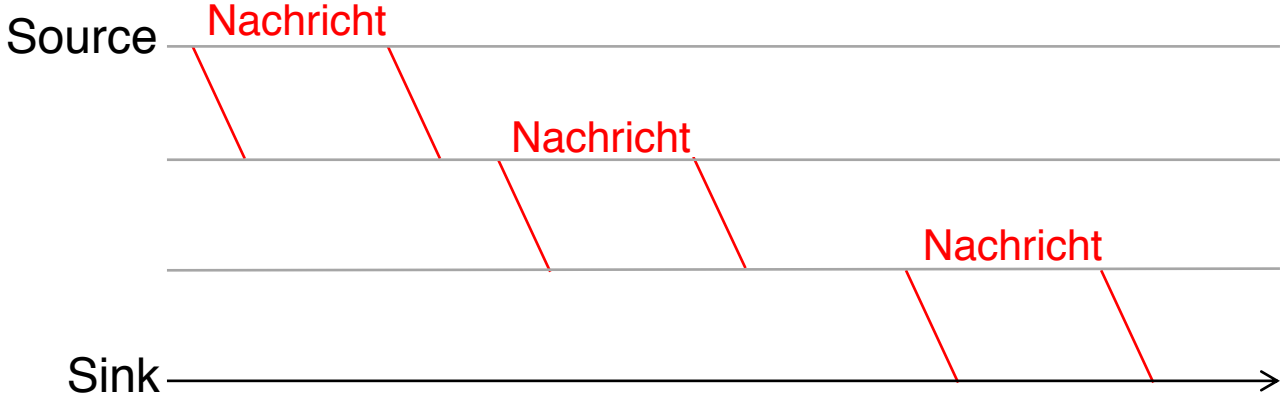
➔Es gibt keinerlei Warteschlangen in den Knoten.

➔Es ist einfach eine geforderte Dienstgüte zu garantieren.

Nachrichtenvermittlung (Engl. Message Switching)

Wird auch als „Store and forward“ Verfahren bezeichnet

- Die ganze Nachricht wird zum jeweils nächsten Knoten geschickt und dort zwischengespeichert, bis dieser sie wieder weiterleitet.
- Streckenstücke sind im Allgemeinen nicht homogen.
- Schwankende Verarbeitungs- und Warteschlangenverzögerungen.
- Beispiel: *E-Mail*



Vor- und Nachteile der Nachrichtenvermittlung

Nachteile

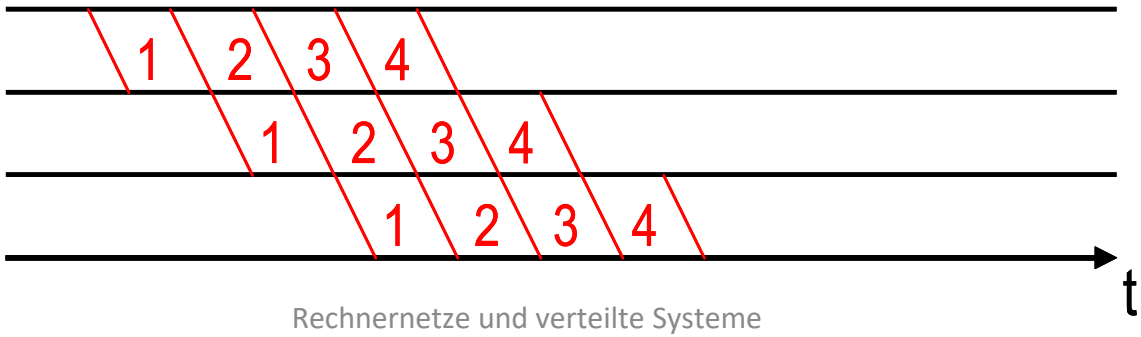
- Übertragung auf dem jeweils nächstem Wegstück beginnt erst, wenn die Nachricht vollständig eingetroffen ist
- Lange Nachrichten können einzelne Wegstücke (für andere Nutzer) lange blockieren.
- Notwendigkeit großer Zwischenspeicher in den Knoten.

Vorteile

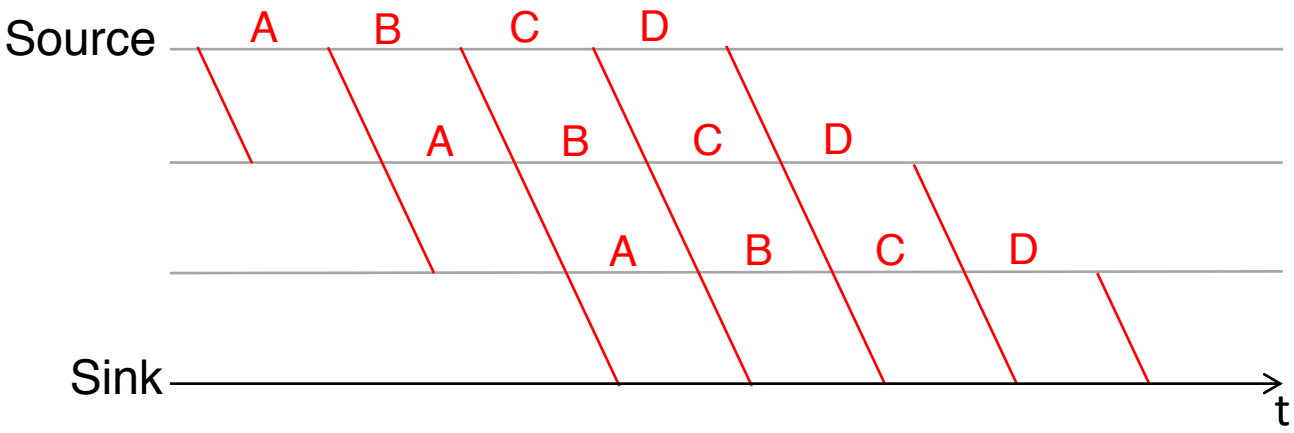
- Nachrichtenvermittlung vor allem dann sinnvoll, wenn die Konnektivität zwischen den Vermittlungseinrichtungen nicht dauerhaft gegeben ist verzögerungstolerante Netze
- Bsp.: Low-Earth-Orbit Satelliten, U-Boote

Paketvermittlung (Engl. Packet Switching)

- Zerlegung der Nachrichten in etwa gleichlange Pakete.
- Senden der Nachrichten **paketweise** nach dem *Store and Forward* Verfahren (vorherige Folie).
- Bessere Leitungsauslastung und geringere Latenz (bzgl. Eintreffen des 1. Pakets) durch *Pipelining Effekt*
- Reihenfolgeproblem
- Verbindungsorientiert sowie verbindungslos möglich
- Beispiel: Internet , ATM



Paketvermittlung (Animation)



Source muss noch senden: A, B, C, D

Unterwegs:

Sink hat erhalten: A, B, C, D

Vor- und Nachteile der Paketvermittlung

Nachteile

- Aufteilung der Nachricht erfordert Duplizieren der Header-Informationen
- Reihenfolgeprobleme
- Geforderte Dienstgüte schwierig zu garantieren.

Vorteile

- Begrenzung der Paketgröße kann sicherstellen, dass kein einzelnes Paket eine Leitung zu lange belegt.
- Pfade müssen nicht vor der Datenübertragung eingerichtet werden

Begriffe

- Routingalgorithmen beschreiben Wegewahlverfahren
- Verfahrensauswahl und -ausprägung hängt ab
  - von der Routingstrategie (engl. Policy)
  - von den Ziel-/Kostenfunktionen
  - Beispiele:
    - geringe Übertragungskosten
    - geringe Übertragungszeiten
    - gute Leitungsauslastung
    - großer Durchsatz

Begriffe

- Routingverfahren möglichst
  - einfach (algorithmische Komplexität, Netz-Overhead)
  - adaptiv (Last, Topologie)
  - robust (bei Fehlern)
  - fair
- Grundlage ist die Routingtabelle

Probleme

- Zielkonflikte
- Beschreibung der Topologie
  - wie beschrieben (Leitungen, Knoten, Kosten)
  - vollständig / partiell
- Berechnung
  - Wo? (zentral / dezentral)
  - Welche Informationen werden vom Algorithmus benötigt?
  - Wie werden diese Informationen bereitgestellt?
  - Welche Ereignisse stoßen die Berechnung an?
  - Wann wird ein neuer Weg aktiviert?

Verkehrsschild als Basis für Wegewahlentscheidung

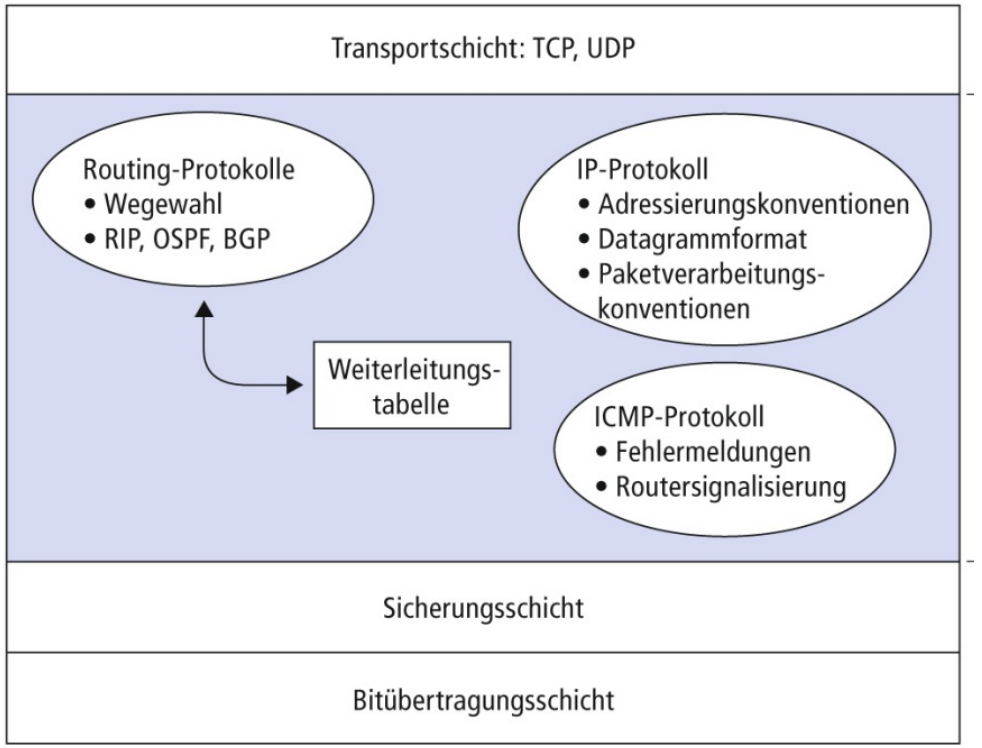
Beschriftung ist nur am Standort des Schildes sinnvoll.

- verschiedene Wege (geradeaus A6, rechts A73)
- indirekt erreichbare Ziele (Regensburg über A3)
- direkt erreichbare Ziele (Flughafen, Messe)

Kapitel 5.2
Das Internet Protokoll (IP)

Forwarding und *Adressierung* im Internet

Vermittlungsschicht im Internet



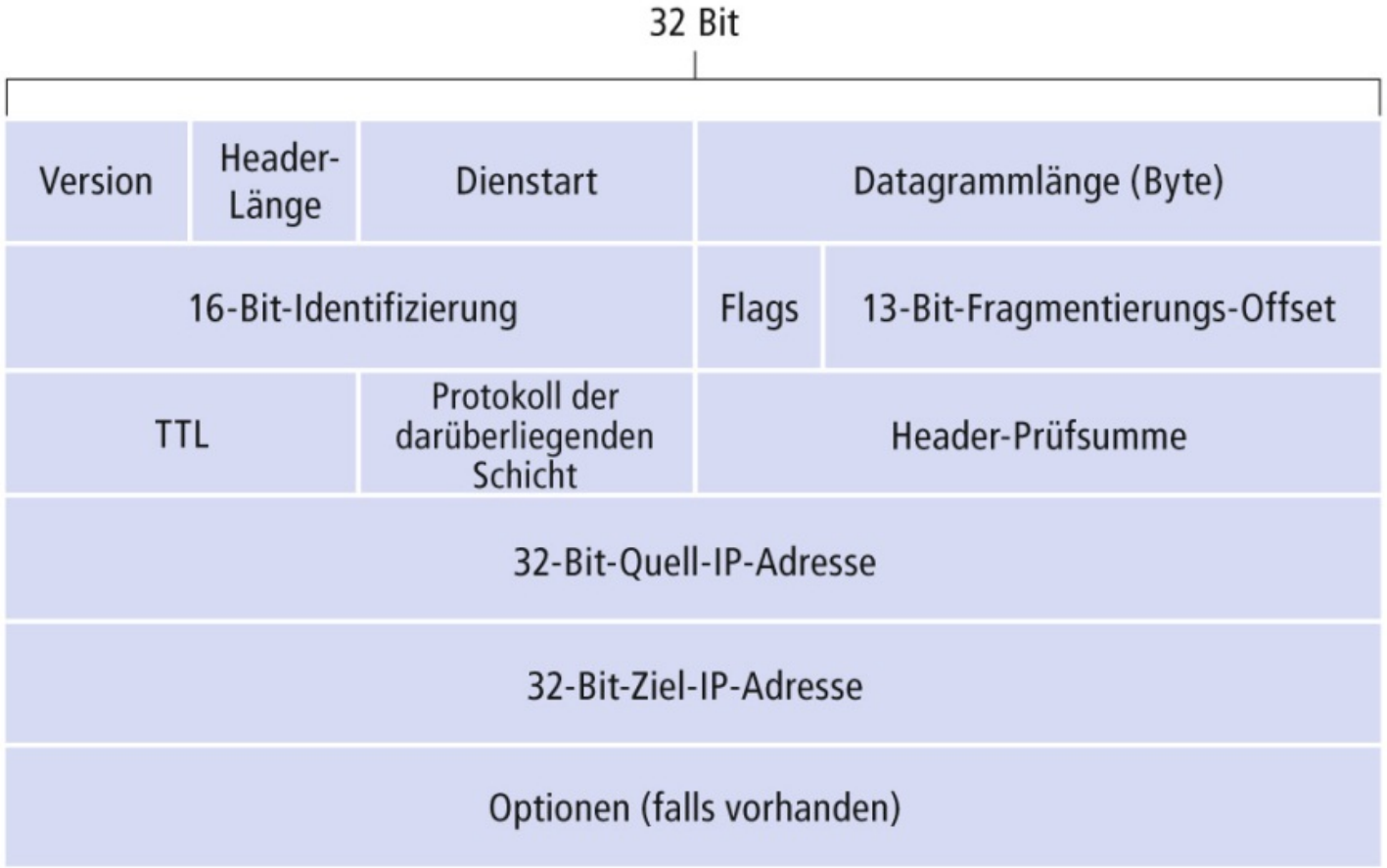
Die Vermittlungsschicht im Internet hat drei Hauptbestandteile:

- Internet-Protokoll
- Routing-Protokolle
- ICMP

Überblick IP

- Verbindungsloser Dienst mit zwei Dienstprimitiven (SEND, DELIVER)
- Netzglobale Adressierung mit Hilfe von IP-Adressen
- Protocol: ID des IP-Nutzers (SAP-Adresse)
  Bsp.: 1 ICMP, 6 TCP, 9 UDP, 46 RSVP
- Unterstützt Fragmentierung und Reassembly (zum Umgang mit MTUs)
- Time-to-Live (TTL), gemessen in „Hops“ (Wegstücken)
- IP-Datagramm bestehend aus Header- und Datenteil
  - IP-PDUs heißen *Pakete*

Der IP-Header (PCI) (1/2)



Der IP-Header (PCI) (2/2)

- **Fragmentierungs-Offset** (13 Bit): Die Stelle, die ein Fragment im originalen Paket inne hält, gemessen in 8-Byte Blöcken.
- **Time to Live** (8 Bit): Anzahl hops, die das Datagramm machen darf bevor es verworfen wird.
- **Protokoll** (8 Bit): ID des IP-Nutzers (für welchen SAP ist das Paket bestimmt?)
- **Prüfsumme** (16 Bit): Einerkomplement der Summe der 16-Bit-Halbwörter des Headers. (Muss nach jeder Teilstrecke neuberechnet werden, da sich die TTL ändert.)
- **Quell-/Zieladresse** (je 32 Bit): IP-Adressen der Netzschnittstellen von Quelle und Ziel.
- **Optionen** (n * 32 Bit): Wählbare Eigenschaften, wie z.B. ein Timestamp.

Fragmentierung (1/2)

In Kapitel 6 werden wir sehen, dass nicht alle Protokolle der Sicherungsschicht die gleiche Größe von Paketen übertragen können. Beispiel:

Ethernet kann 1500 Bytes je PDU (Frame) übertragen. WiFi hingegen 2312 Bytes.

Die Menge an Daten, die ein Frame eines Schicht-2 Protokolls übertragen kann, heißt *Maximum Transmission Unit* (MTU). Die MTU setzt also eine obere Grenze für die Größe von IP-Paketen.

Problem: jeder Link eines Pfades kann eine eigene MTU haben → die Pakete werden *fragmentiert*

Fragmentierung (2/2)

Ablauf der Fragmentierung von IP-Paketen:

Soll ein Paket mit Größe $S_P = S_H + S_D$ Bytes über einen Link mit MTU $M < S_P$ Bytes gesendet werden, muss das Paket in $\frac{S_D}{M - S_H}$ *Fragmente* aufgeteilt werden werden.

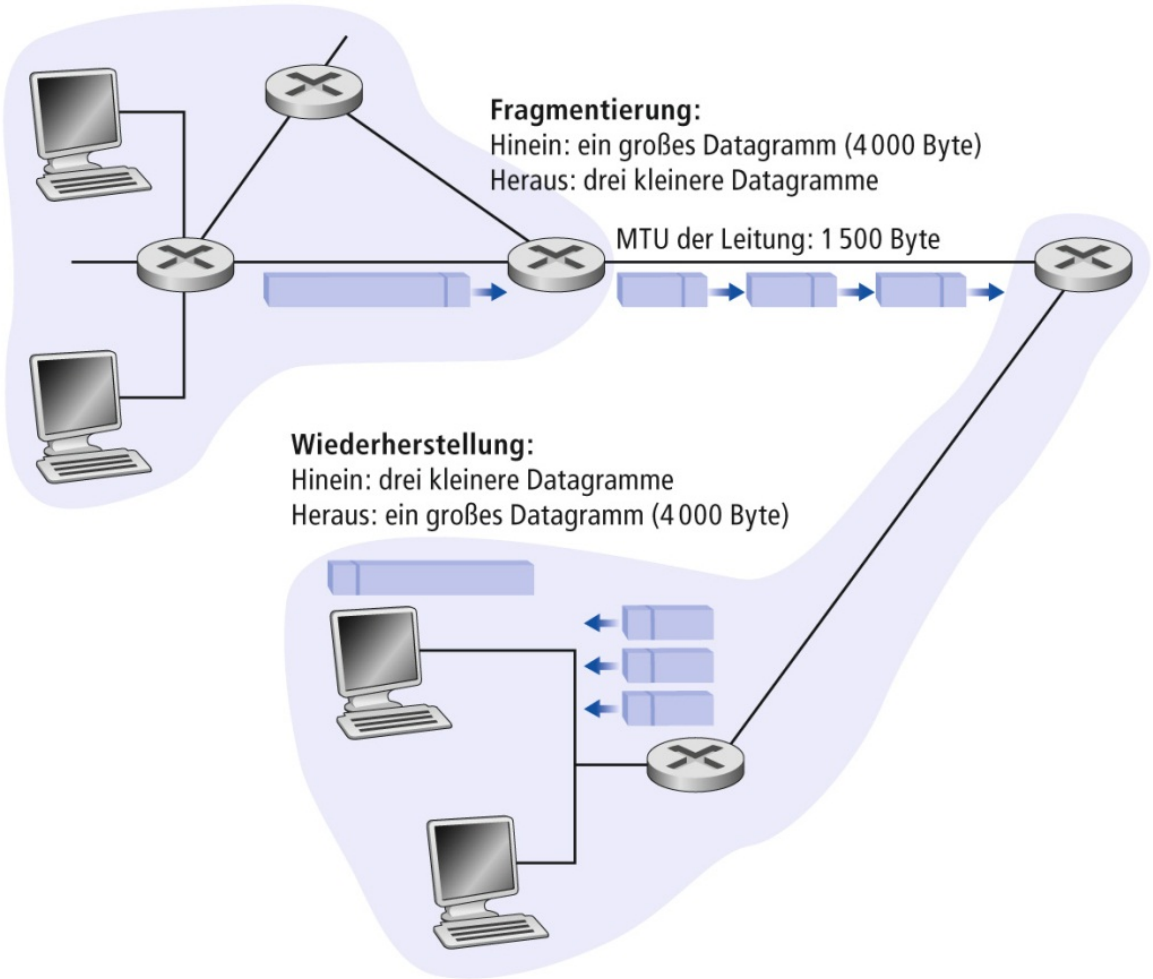
Der Sender (oder Router am jeweiligen Hop) teilt das Paket in Fragmente auf:

Identifizierungsnummer wird vom Paket übernommen.

Fragmentierungs-Offset wird je Fragment gesetzt.

Flag ist 1, falls weitere Fragmente folgen und 0 sonst.

Fragmentierung: Beispiel



IP und unterstützende Protokolle

- ICMP (engl. Internet Control Message Protocol),
  - RFC 792, 1122
  - Benachrichtigungsprotokoll, dient dem Austausch von Informationen und Fehlermeldungen über IP.
  - Bsp.: Destination Unreachable, Time Exceeded, Redirect, usw.
  - Basis für traceroute
- DHCP (engl. Dynamic Host Configuration Protocol)
  - Dynamische temporäre Zuweisung von IP-Adressen
- DNS (engl. Domain Name System)
  - Abbildung von Namen auf IP-Adressen
  - Interaktion zwischen Resolver (Client) und Server (Directory)
- ARP (engl. Address Resolution Protocol), RFC 826
  - Abbildung von IP-Adressen auf MAC-Adressen (Schicht 2)
- IPsec (engl. Internet Protocol Security):
  - Authentifizierung und Verschlüsselung von IP-Paketen

Adressierung und Subnetze im Internet Protokoll

Adressen und Interfaces

Ein Host ist üblicherweise über einen Link mit einem Netz verbunden. Der Übergang von Host zu Netz heißt Interface oder Netzschnittstelle.

Router haben zwei oder mehr Interfaces.

Damit auf einem Interface IP-Pakete empfangen und gesendet werden können, benötigt es eine eigene IP-Adresse.

Diese bestehen bei IP (v4) aus 32 Bits (4 Bytes)

→ es existieren $2^{32} = 4.294.967.296$ IPv4-Adressen.

IPv4-Adressen: Notation

IP-Adressen werden üblicherweise in der sog. Dezimalpunktnotation (dotted-decimal notation) geschrieben:

- Jedes Byte wird dezimal getrennt von einem Punkt notiert
- Beispiel:

  11000001 00100000 11011000 00001001 entspricht
  193.32.216.9

- Binärzahl-Darstellung oft hilfreich (Stichwort: Netzmasken)

Sonderadressen

- alles 0: dieser Host
- alles 1: Broadcast innerhalb des lokalen Netzes
- 127.*.*.*: Loop-Back-Adressen (Schleifentest)

Später: 2 spezielle Adressen

- Netzadresse:
  niedrigste Adresse (Host-ID = alles 0)
- Broadcast-Adresse:
  höchste Adresse (Host-ID = alles 1)

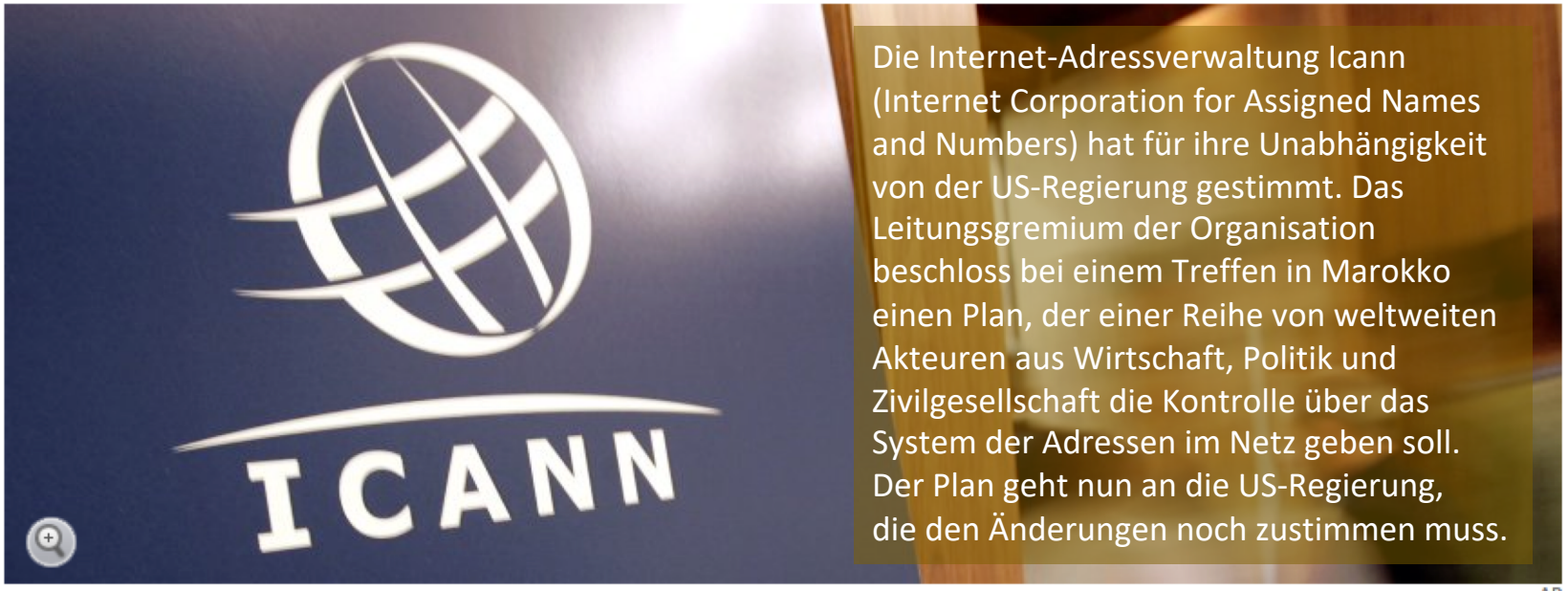
Vergabe

- IPv4-Adressen werden hierarchisch verwendet.
  - IPv4-Adressen bestehen aus Netzteil (Netz ID), der das Netz adressiert, und Hostteil (Host ID), der den Host adressiert
- Einzelne IP-Adressen haben Sonderbedeutungen.

- Internationale Vergabe durch die IANA (Internet Assigned Numbers Authority)
  - Delegiert an nationale Organisationen.
  - Abteilung der ICANN (Internet Corporation for Assigned Names and Numbers)
  - Buchhalter für die Registrierungen

Politik und ICANN

Internet-Adressen: Icann sagt sich von US-Abhängigkeit los



AP

Icann-Logo

Die Icann ist die Hüterin über die Adressen im Internet. Bisher stand die Adressverwaltung unter der Aufsicht des US-Handelsministeriums. Jetzt stimmte das Gremium für neue Kontrollmechanismen.

http://www.spiegel.de/netzwelt/netzpolitik/icann-sagt-sich-von-us-abhaengigkeit-los-a-1081731.html

Klassenbasierte Adressierung (Engl.: Classful Networking)

Grundidee:

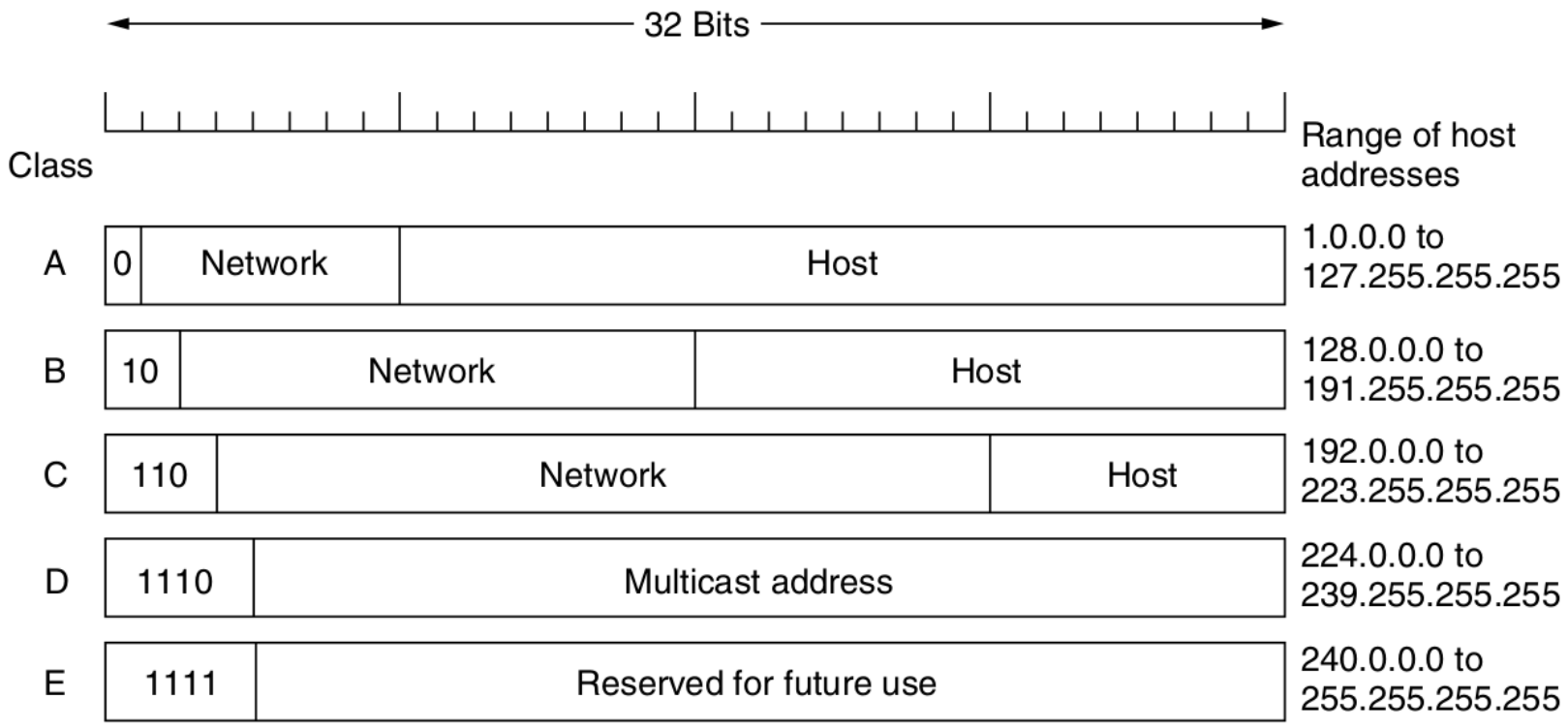
- Einteilung in Klassen anhand des Verwendungszwecks:
  - Netzgrößen für Unicast (A,B,C)
  - Multicast Network (D)
  - Future and Experimental Use (E)
- Grenze zwischen Netz-Teil und Host-Teil verläuft entlang der Byte-Grenzen ➔ Softwareimplementierung
- Netz-Teil wird in Präfix und Netz-ID unterteilt
  - Erlaubt schnelle Erkennung der Klasse anhand weniger Bits
  - Schnelle Routing-Entscheidungen anhand Klasse und Netz-ID

Klassenbasierte Adressierung (Engl.: Classful Networking)

https://de.wikipedia.org/wiki/Netzklasse

Klasse	Präfix	Länge Netz ID	Länge Host ID	Anzahl Netze	Hosts pro Netz
A	0	7 bit	24 bit (3 byte)	126	16 777 214
B	10	14 bit	16 bit (2 byte)	16 382	65 534
C	110	21 bit	8 bit (1 byte)	2 097 152	254
D	1110	Verwendung für Multicast-Anwendungen			
E	1111	Reserviert (für zukünftige Zwecke)			

Klassenbasierte Adressierung (Grafik)



Quelle: Tanenbaum, Computer Networks 5th Edition

Klassenbasierte Adressierung Grundproblem

- Adressen werden in Blöcken vergeben (z.B.: alle Adressen mit einer gegebenen Netz-ID)
- Wird z.B. ein Klasse B Netz an eine Organisation vergeben welche deutlich weniger als 65000 Hosts besitzt, so bleiben viele Adressen ungenutzt.
- (Oft reicht bereits die Erwartung, dass 254 Hosts einmal nicht reichen könnten, für die Anforderung eines Klasse B Adressblocks.)
- Es gibt nur 16382 Klasse B Netze.
- Lösung: Netz IDs mit variabler Länge → CIDR

CIDR (Engl.: Classless Inter-Domain Routing)

- Mit RFC 1517 bis 1520 ab 1993 in Verwendung.
- Grundidee: Die Grenze zwischen Netz-Teil und Host-Teil verläuft fließend.
  => Routing-Protokolle müssen die Länge der Netz-ID (Netzpräfix) zusätzlich zur Adresse übertragen.
- Verwendung von Subnetz-Masken
- Notation: <IP Adresse>/<Präfixlänge>

- Beispiel: 192.168.121.0/26
  (die ersten 26 Bit sind Netz-ID, der Rest Host-ID)

Private IP-Adressen (1/2)

- I.d.R. vergeben ISPs (engl.: Internet Service Providers) an Privatkunden oder kleine Firmen nur eine einzige öffentliche IP Adresse (unabhängig davon wie viele Hosts diese in ihrem LAN anschließen).
- Innerhalb ihres LANs (engl.: Local Area Network) verwenden die Hosts sogenannte private IP-Adressen (nur innerhalb des LANs eindeutig)

Private IP-Adressen (2/2)

- Folgende Adressblöcke sind als privat reserviert:
  - 10.0.0.0/8 (Klasse A Adressblock)
  - 172.16.0.0/12 (16 Klasse B Adressblöcke)
  - 192.168.0.0/16 (256 Klasse C Adressblöcke)

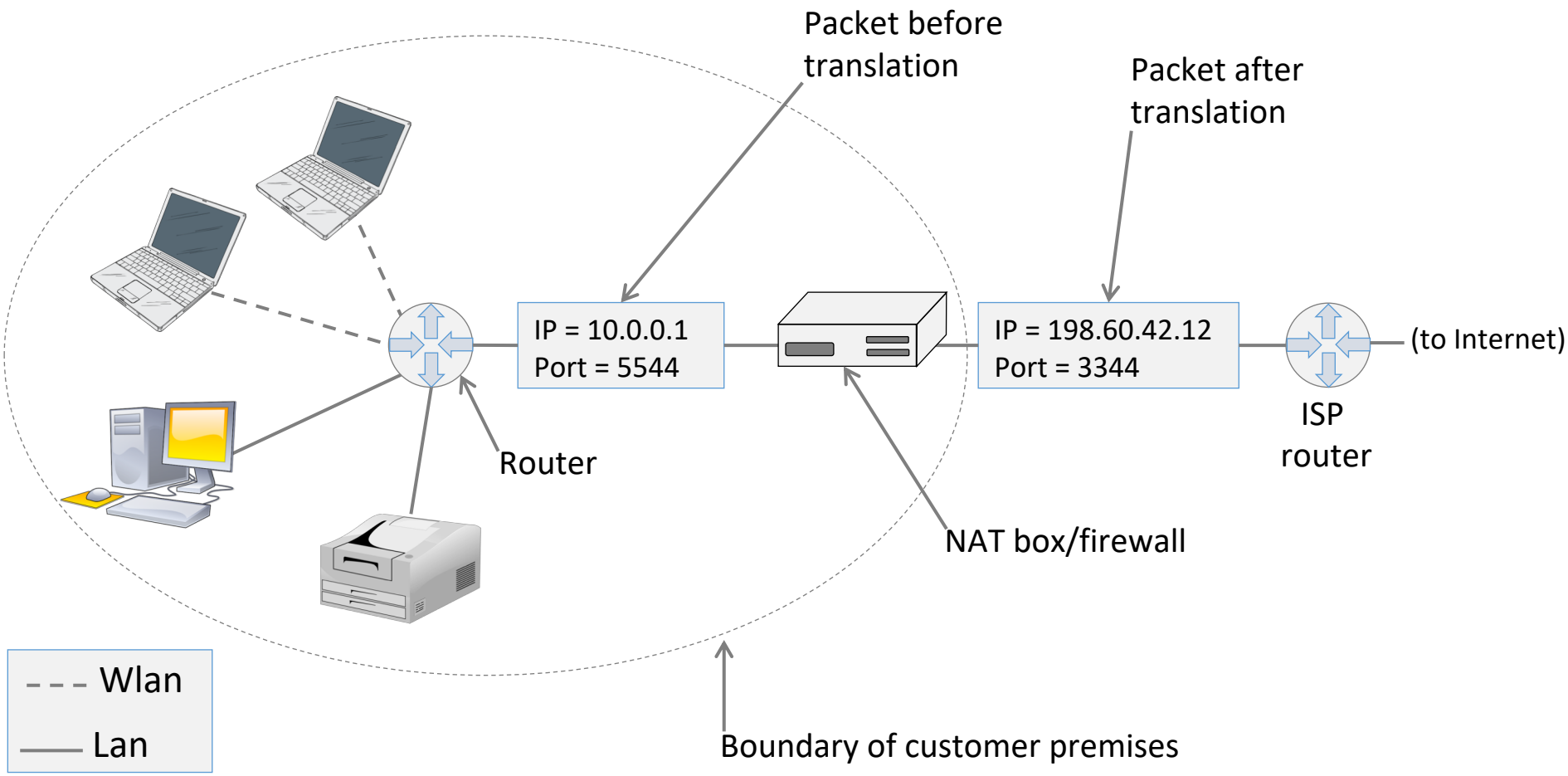
- Private Adressen werden nicht im Internet vermittelt
- Pakete mit privaten Adressen als Ziel oder Absender werden von Routern verworfen

NAT (Network Address Translation)

(Vergleiche auch „Kapitel 1.4: Ein Einführendes Beispiel“)

- Bevor Router ein Paket aus dem privaten LAN ins Internet weiterleitet, wird Absenderadresse in öffentliche IP-Adresse geändert
  ➔ *„Address Translation“*
- Router merkt sich die Änderung der Adresse
- Zusätzlich zur Adresse wird auch der verwendete Port gemerkt (siehe Sockets ➔ Kapitel 3)
- Auf „Rückweg“ erfolgt umgekehrte Übersetzung.

NAT/Mascquerading (als Grafik)



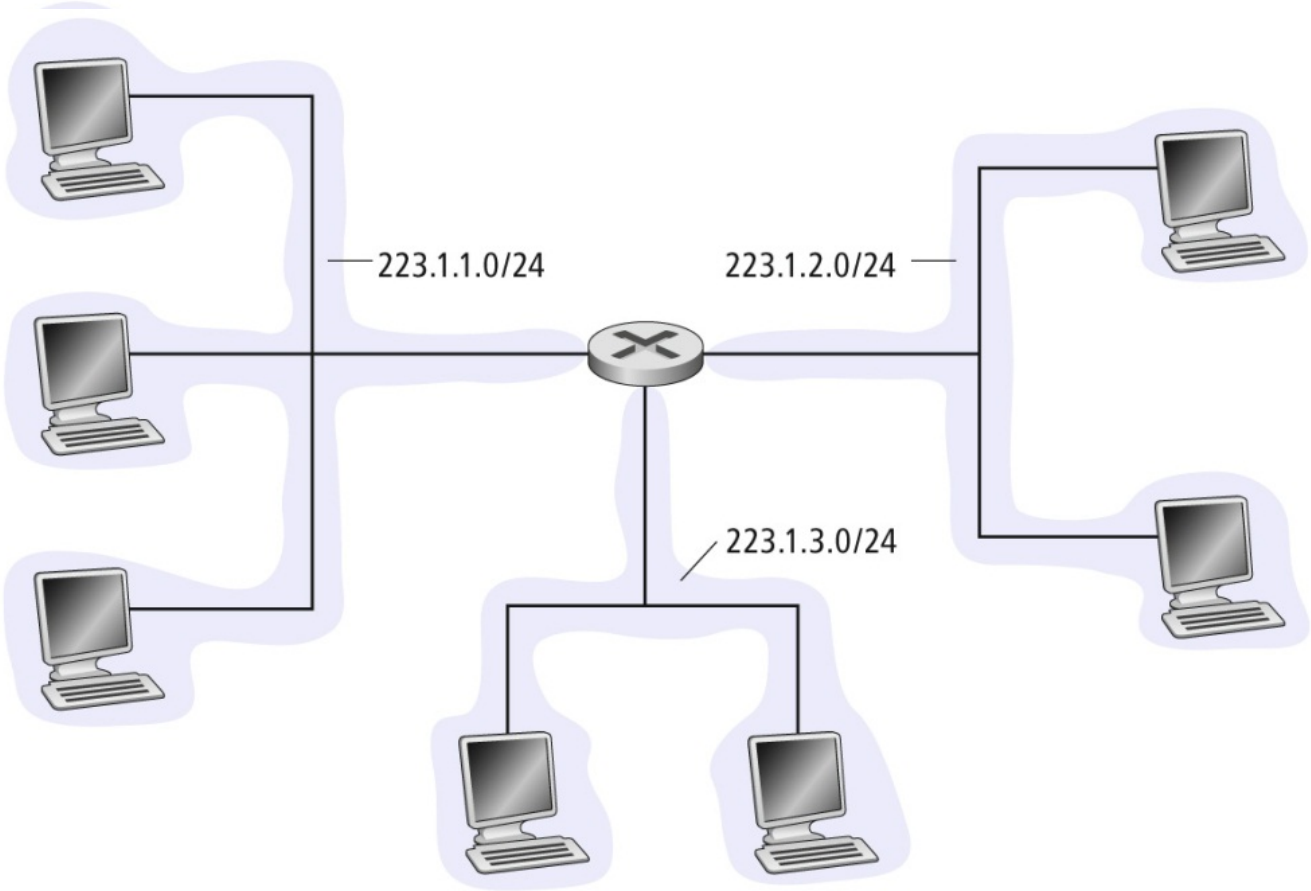
Subnetting

- Sowohl bei klassenbasierter Adressierung, als auch bei CIDR
- Grundidee: Host-Teil eines Adressblocks wird weiter unterteilt in Subnetz-ID-Teil und Host-ID-Teil.

Netz ID	Host ID	
Netz ID	Subnetz ID	Host ID

- Eine Organisation kann mit einem einzigem Adressblock mehrere eigene Netze bedienen

Subnetting: Beispiel 1



Subnetting: Beispiel 2

Eine Organisation bekommt den Adressblock 131.42.0.0/16 zugewiesen und benötigt

- 1 Subnetz mit bis zu 32000 Hosts
- 15 Subnetze mit bis zu 2000 Hosts
- 8 Subnetze mit bis zu 250 Hosts

Machen Sie Vorschläge für eine Aufteilung in geeignete Subnetz-Adressen

ICMP

Das Internet Control Message Protocol

ICMP

Der Zweck von ICMP ist die Kommunikation von Informationen *über* die Vermittlungsschicht.

ICMP gilt als Teil der Vermittlungsschicht, obwohl die PDUs wie TCP-Segmente oder UDP-Datagramme als Nutzlast in IP-Paketen verschickt werden.

Durch ein jeweils 8 Bit breites *Type*- und Code-Feld, wird der Typ des ICMP-Pakets festgelegt.

Das Protokoll bildet z.B. die Grundlage für die Programme *ping* und *traceroute* (siehe Grundlagen).

ICMP-Typen

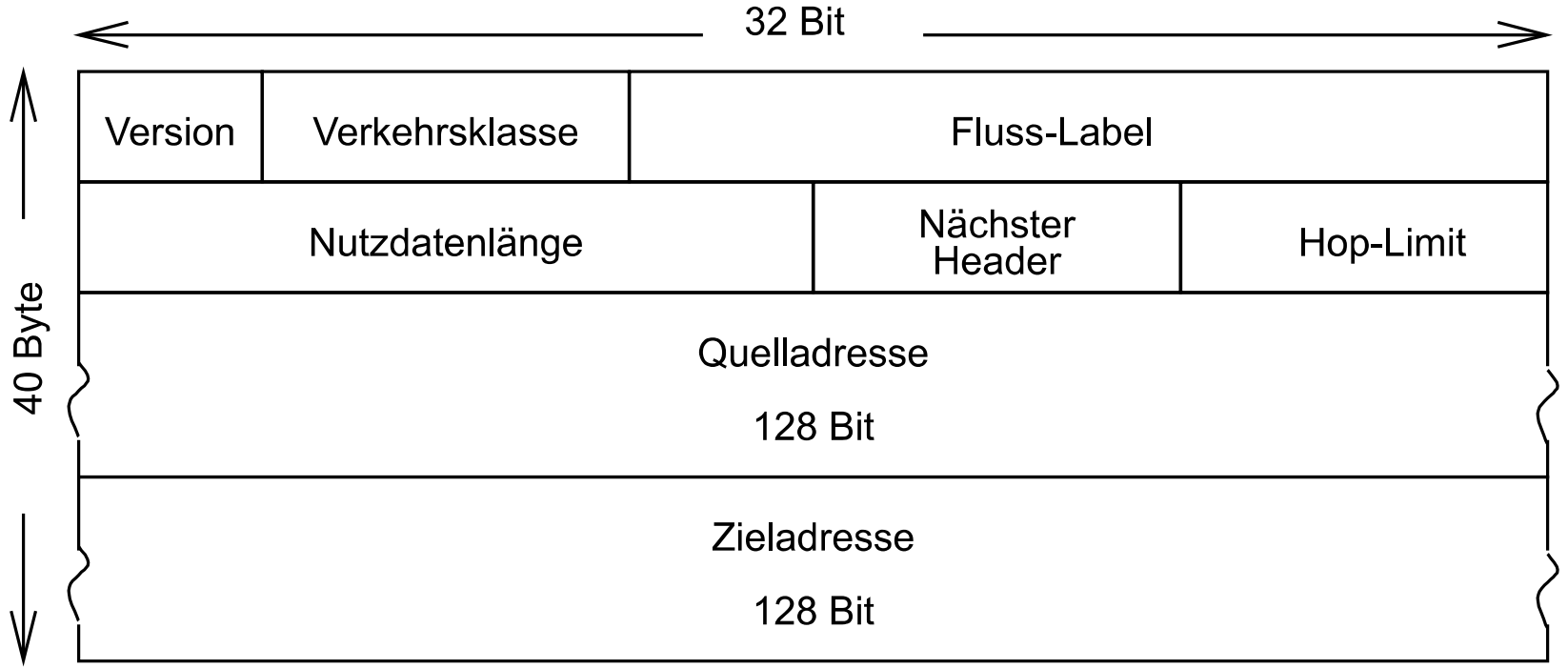
ICMP-Typ	Code	Beschreibung
0	0	Echo-Antwort (Ping)
3	0	Zielnetz unerreichbar
3	1	Zielhost unerreichbar
3	2	Zielprotokoll unerreichbar
3	3	Zielport unerreichbar
3	6	Zielnetz unbekannt
3	7	Zielhost unbekannt
4	0	Source Quench (Überlastkontrolle)
8	0	Echo-Anforderung (Ping)
9	0	Routerbekanntmachung
10	0	Routersuche
11	0	TTL abgelaufen
12	0	IP-Header fehlerhaft

Internet Protokoll (v6)

Überblick

- Motivation
  - Adressknappheit bei IPv4 bei wachsendem Bedarf an Adressen.
  - 1. Februar 2011: zentraler Vorrat an IPv4-Adressen erschöpft.
- Neue und ausgebaute Funktionalität (im Vgl. zu IPv4)
  - erweiterter Adressraum (128 statt 32 bit)
  - dynamische Adresszuweisung (Mobilitätsunterstützung)
  - Erweiterungen für QoS auch bzgl. Streams wie Sprache, Video teilweise verbindungsorientierte Protokollaspekte
  - Ressource Allocation (Reservierungsprotokolle)
  - eingebaute Sicherheitsmechanismen (IPsec)
  - Router Anweisungen (hop-by-hop-options)
- Status
  - IPv6 wächst; In gängigen Betriebssystemen implementiert.
  - China, Japan sind Vorreiter (unter anderem, weil sie von der IPv4 Adressknappheit besonders betroffen sind).

Der IPv6-Header (PCI) als Grafik

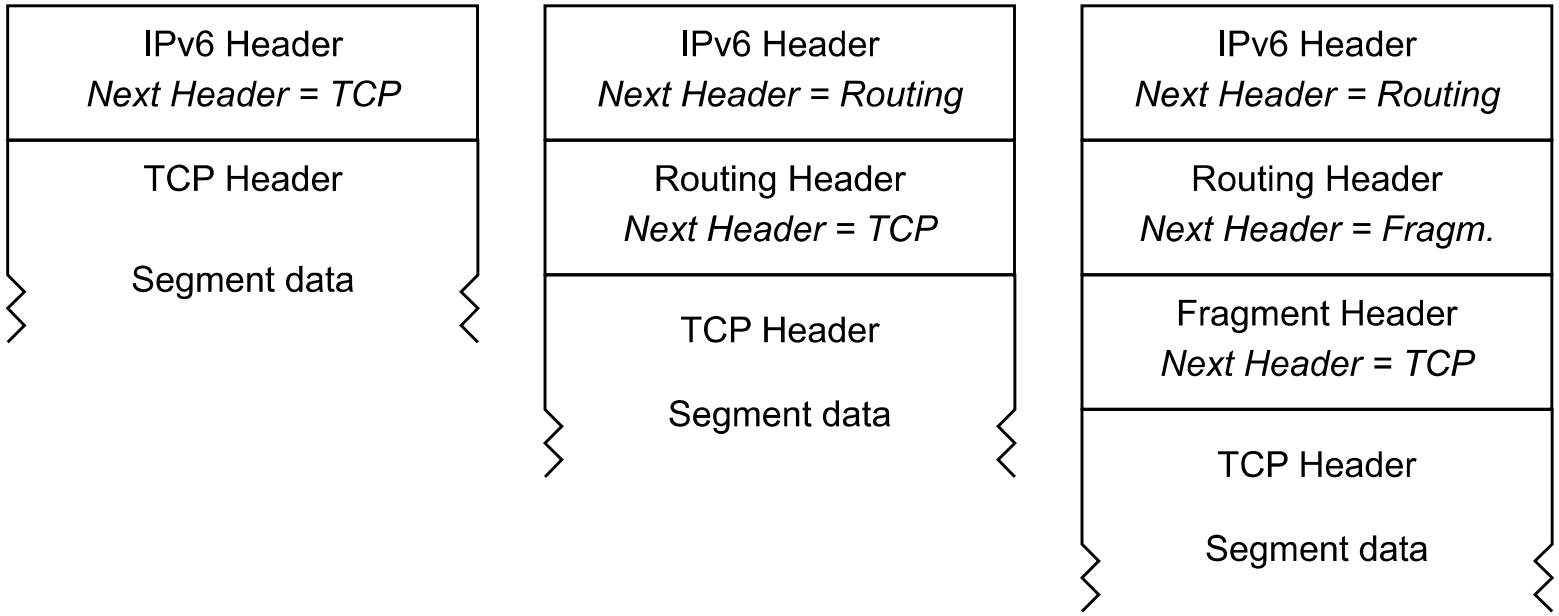


- Der IPv6-Header hat eine feste Größe von 40 Byte.
- 32 dieser Bytes gehen allein an die Quell- und Zieladresse.
- Wenn das nächste Header Feld entsprechend gesetzt ist, können Erweiterungs-Header angehängt werden.

Der IPv6-Header (PCI)

- **Version** (4 Bit): Bei IPv6 immer 6.
- **Verkehrsklasse** (8 Bit): Entspricht dem „Type of Service“ Feld des IPv4 Headers.
- **Fluss Label** (20 Bit): Paket kann als teil eines zusammengehörigen Paketstroms behandelt werden Verbindungsorientierter Protokollaspekt.
- **Nutzdatenlänge** (16 Bit): Anzahl der Nutzdatenbytes im Paket.
- **Nächster Header** (8 Bit): Typ des auf den Header folgenden Erweiterungs-Headers. Wenn keine weiteren Header folgen, entspricht dieses Feld dem Protokollfeld des IPv4 Headers.
- **Hop-Limit** (8 Bit): Entspricht dem Time to Live Feld des IPv4 Headers (der Name wurde geändert um die tatsächliche Nutzung besser widerzuspiegeln).
- **Quell-/Zieladresse** (je 128 Bit): IPv6-Adressen der Netzschnittstellen von Quelle und Ziel.

Erweiterungs-Header

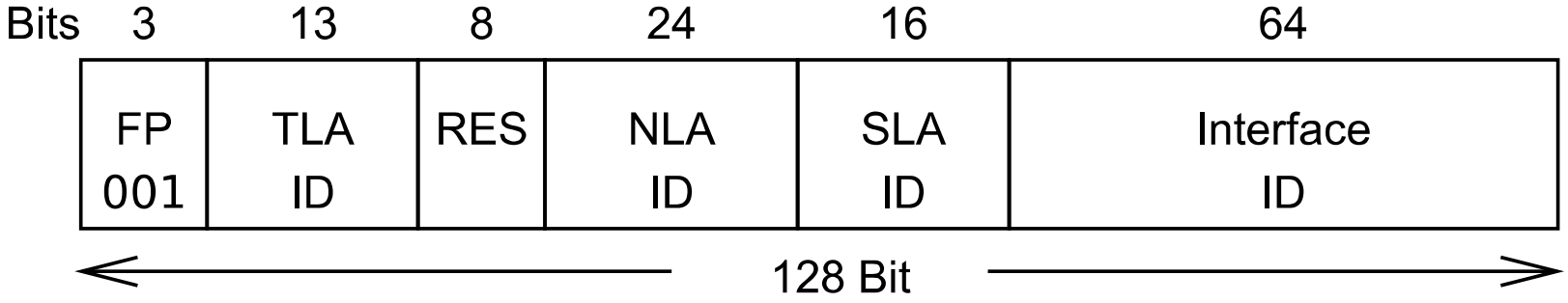


- Original 6 Header-Typen als Teil der IPv6 Spezifikation (RFC 2460)
  Definiert: Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, Encapsulating Security Payload
- Weitere Header-Typen (in der Zukunft) möglich.
- Anzahl (i.d.R. eins) und Reihenfolge der Header können zur effizienten Verarbeitung vorgegeben werden.

IPv6-Adressen

- 128-Bit Unicast-, Multicast- und Anycast-Adressen
  - Unicast: Übertragung von einer Quelle zu einem Ziel
  - Multicast: für Mehrpunktverbindungen
  - Anycast: Adressierung eines beliebigen Hosts einer Gruppe (zur Lastverteilung und Ausfallsicherheit)
- Notation
  - Acht, durch Doppelpunkte getrennte, 16 Bit Hexadezimalzahlen. Bsp.: 1080:0:0:0:8:800:200C:417A
  - Einmal pro Adresse dürfen mehrere Nullwertige 16-Bit-Gruppen durch „::“ zusammengefasst werden: 1080::8:800:200C:417A
  - Präfixschreibweise wie CIDR: 12AB::CD30:0:0:0:0/60
- Sonderadressen
  - Nicht angegeben (Abwesenheit einer Adresse): 0:0:0:0:0:0:0:0
  - Loopback Adresse: 0:0:0:0:0:0:0:1 oder ::1

Globale Unicast-Adressen

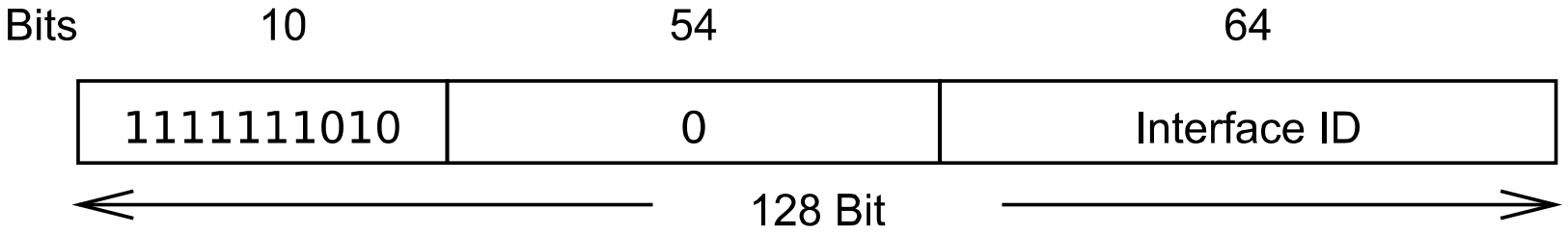


- Adress-Felder:
  - **Format Prefix (FP), Top-Level Aggregation ID (TLA ID), RES, Next-Level Aggregation ID (NLA ID):** Ergeben zusammen den globalen Routingprefix (eine Art strukturierte Netz ID).
  - **Site-Level ID (SLA ID):** Entspricht einer Subnetz ID.
  - **Interface ID**: Adressiert die Netzschnittstelle auf dem Host
- Dreistufige Adresshierarchie

Lokale Unicast-Adressen

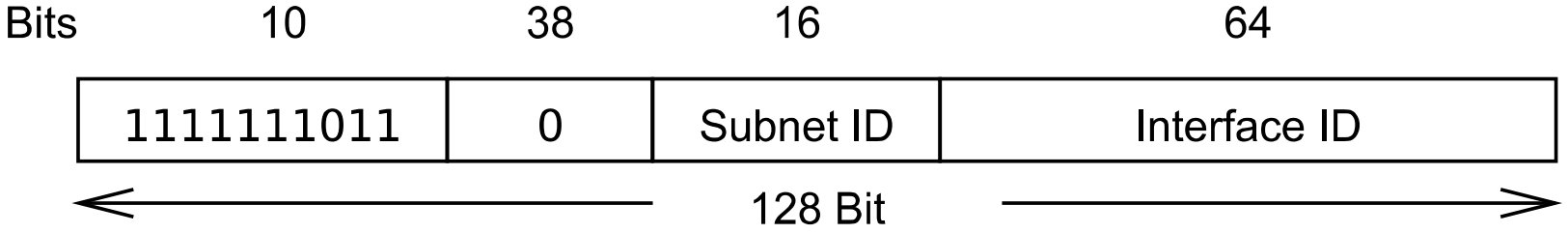
Link-local

- Bezug nur auf einen bestimmten Link
- Einsatz z.B. für Autokonfiguration



Site-local

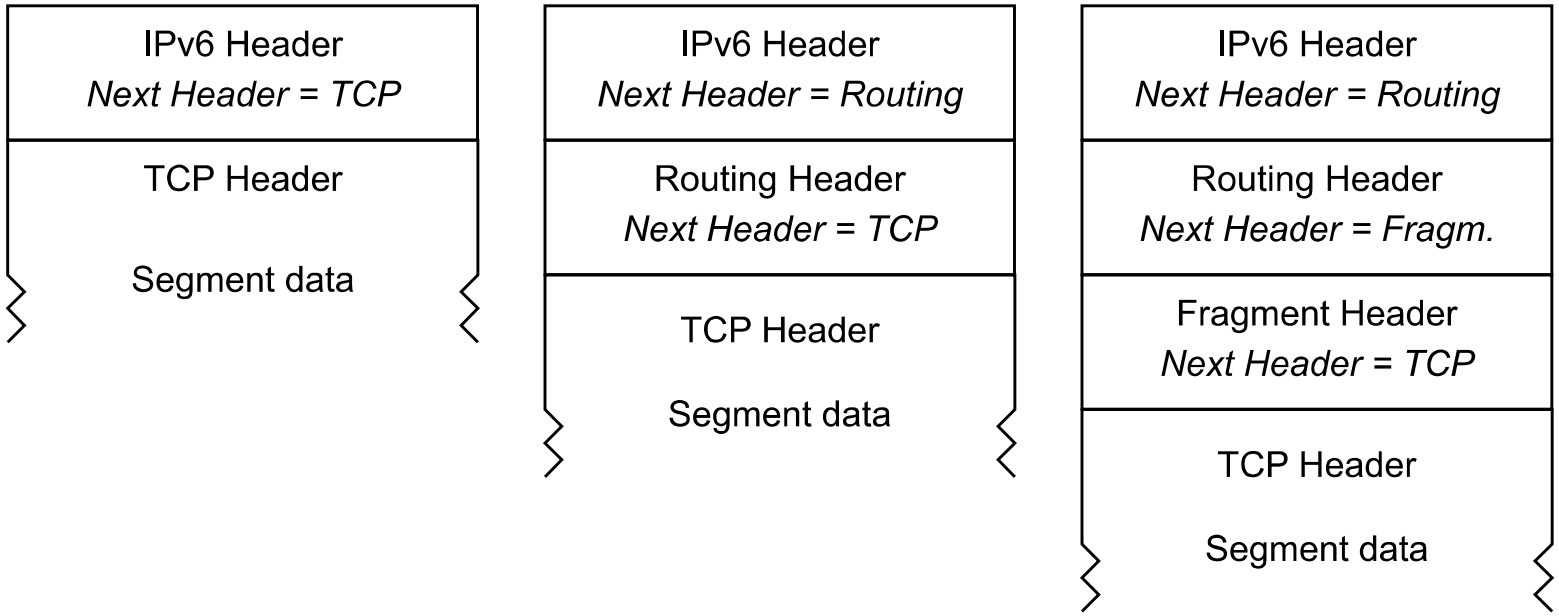
- Geltungsbereich umfasst einen Standort/eine Site
- ähnlich privater Adressblöcke bei IPv4
- seit 2004 abgeschafft („deprecated“), RFC 3879



IPv6-Adressen auf Basis von Schicht-2-Adressen

- Idee: Nutzung gerätegebundener Adressen bei der Bildung von IPv6-Adressen
- Anwendung nur auf Interface Identifier (64 Bits)
- Extended Unique Identifier (EUI-64) (Definition der IEEE)
- z.B. anwendbar mit Ethernet, Token Ring, FDDI . . .
- Beispiel: IEEE 802, 48 Bit auf 64 Bits Abbilden
  00-23 Kennung des Herstellers mit Flag-Bits
  6 universal/local (1 = globaler Geltungsbereich)
  7 individual/group
  24-39 1111111 111111110
  40-63 spezifisch für Geräteinstanz

Erweiterungs Header



- Original 6 Header-Typen als Teil der IPv6 Spezifikation (RFC 2460)
  Definiert: Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, Encapsulating Security Payload
- Weitere Header-Typen (in der Zukunft) möglich.
- Anzahl (i.d.R. eins) und Reihenfolge der Header können zur effizienten Verarbeitung vorgegeben werden.

Extension Header - Beispiele

- Hop-by-Hop Options Header
- Routing Header
- Fragment Header

Der Hop-by-Hop Options Header

- Zweck: Angabe von Optionen, die jeder Knoten auf dem Weg berücksichtigen soll. (z.B. Padding)
- Felder:
  - **Next Header** (8 Bit): Angabe des folgenden Headers (wie bekannt)
  - **HEL (Header Extension Length)** (8 Bit): Länge des Erweiterungs-headers in Bytes.
  - **Optionen:** als (Typ, Länge, Wert) Tupel Codiert:
    - **Typ** (8 Bit): Art der Option
    - **Länge** (8 Bit): Länge des Wertes der Option in Byte
    - **Wert** (bis zu 255 Byte): beliebige Informationen
- Bisherige Optionen:
  - Datagramme mit einer Länge größer als 64 KB.

Der Routing Header

- Zweck: Angabe von Zwischenstationen die das Paket auf seinem Weg durch das Netz der Reihe nach durchlaufen muss.
- Felder
  - **Next Header, Header Extension Length**: wie bekannt
  - **Routing Type**: identifiziert Routing-Header Variante (default: 0)
  - **Segments Left**: Anzahl verbliebener Zwischenstationen, wenn 0 erreicht ist, wird der Routing Header ignoriert.
  - **Typspezifische Daten:** abhängig von Routing Type.

Fragmentierung

- Anders als bei IPv4, wird die Fragmentierung von IPv6 Paketen ausschließlich in den Endsystemen und nicht im Transitsystem durchgeführt.
- Nicht-Fragmentierbarer Teil: IPv6-Header + alle Header mit Ende-zu-Ende Relevanz
- Aufbau von Fragment-Paketen:
  - Nicht-Fragmentierbarer Teil
  - Fragment Header (nächste Folie)
  - Inhalt des ersten Fragments

Der Fragment Header

- Zweck: Senden eines IP-Pakets, das größer ist als die MTU (Maximum Transmissible Unit)
- Felder:
  - **Next Header, Header Extension Length**: wie bekannt
  - **Fragment Offset:** Offset der Daten, die diesem Header folgen
  - **MF Flag (More Fragments)**: wie bei IPv4 (0 bedeutet letztes Fragment, 1 bedeutet weitere Fragmente folgen.)
  - **Identification:** eindeutige Kennung des IP-Pakets, das fragmentiert wird.

IPv4 zu IPv6 Migration

- Betriebliche Aufgabenstellung
  - Möglichst kosteneffiziente Ersetzung der IPv4-Technologie.
  - Koexistenz von IPv4 und IPv6 während der Übergangsphase.
- Prinzipielle Ansätze
  - Dual-Stack: Knoten besitzen sowohl IPv4- als auch IPv6-Implementierung
  - Tunneling (IPv6-Verkehr über IPv4-Netz und umgekehrt)
    - Kapselung von Paketen im tunnelnden Protokoll
    - Managment des Tunnels: Erstellung, Umgang mit Fragmenten
- Übersetzung
  - Protokollübersetzung in Übergangsknoten/Router
  - in Programmbibliotheken (APIs zur Netzprogrammierung)

Kapitel 5.3

Routing Algorithmen

Grundlagen Routing

- Szenario: Zieladresse eines Pakets liegt außerhalb des (Sub-)Netzes eines Hosts

→ Weiterleitung an **Default Router.**

- Grundlegendes Prinzip
  1. Ein zu versendetes Paket wird vom Host (Quell-Adresse) zuerst an den **Default-Router** (Erster Hop) übermittelt.
  2. Default-Router findet **besten** Pfad von **Quell-Router** zu **Ziel-Router**.
  3. Ziel-Router vermittelt das Paket an die Ziel-Adresse.

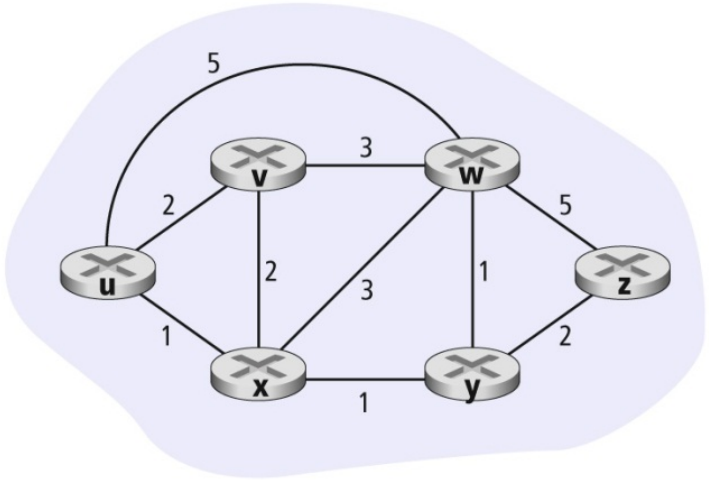
Grundlagen Routing

- Relevante Fragen
  - Wie werden Routing-Tabellen aufgebaut?
  - Was ist der **beste** Pfad von Quell-Router zu Ziel-Router

- Lösung des Problems durch Routing-Algorithmen
  - Link State Routing
  - Distance Vector
  - Autonome Systeme
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Border Gateway Protocol

Grundlagen Graphen-Theorie

- Graph G(N, E)
  - N: Knoten (Nodes)
  - E: Kanten (Edges)
- Kosten-Attribut (C) auf Kanten
  - C(X,Y): „Kosten von X zu Y.“
  - Beispiel: C(U,W) = 5
- Pfad: Sequenz von besuchten Knoten über Kanten.
- Kontext Vermittlungsschicht
  - Router sind Knoten
  - Links (Kabel, Funk,…) sind Kanten



Kosten-Kriterien

- **Geringste Kosten**: Finden einen Pfad von Quelle zu Ziel mit geringsten Kosten.
- **Kürzester Pfad**: Pfad mit geringster Anzahl an Links zwischen von Quelle zu Ziel.

→ Wenn alle Links in einem Netz (Graph) dieselben Kosten haben, entspricht kürzester Pfad auch dem Pfad mit den geringsten Kosten.

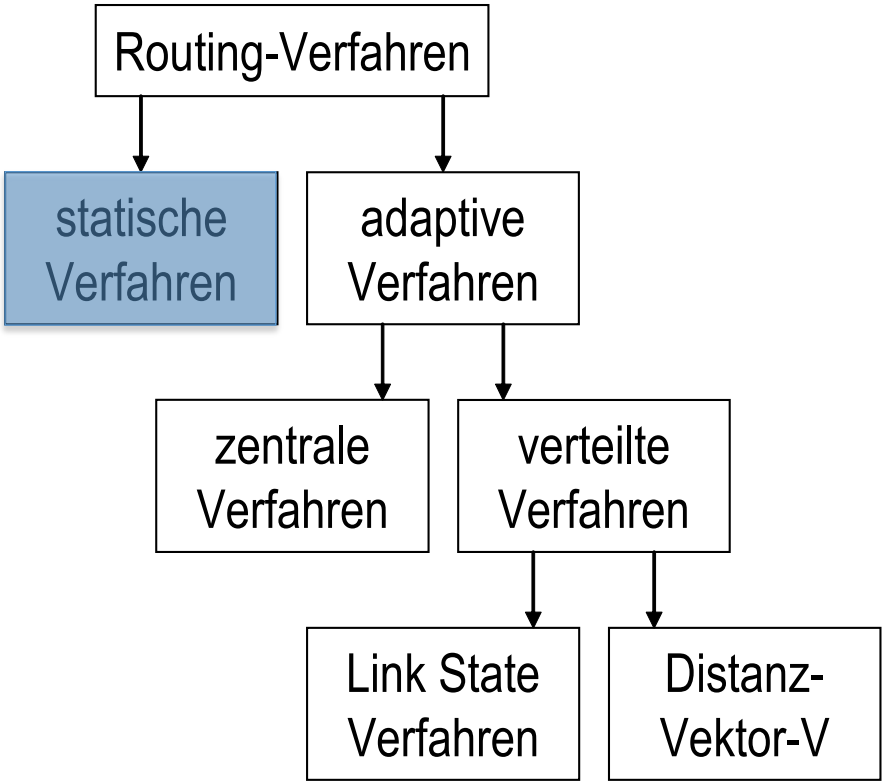
Klassifikation von Routing Verfahren (global vs. dezentral)

- Globales Routing: Berechne den Pfad mit den geringsten Kosten auf Basis des vollständigen Netzes (globales Wissen einer zentralen Komponente). → Beispiel **Link State**
- Dezentralisiertes Routing: Berechne Pfad mit geringsten Kosten schrittweise (iterativ) und verteilt. → Beispiel **Distance Vector**.
  - Kein Knoten hat vollständiges (globales Wissen), nur über seine direkten Nachbarn.

Klassifikation von Routing (statisch vs. dynamisch)

- Statisches Routing: Zustand der **Routing-Tabelle** ändert sich selten (nie) und erfordert manuelles Eingreifen. → Bspw. Administrator
- Adaptives Routing: Änderungen der Routing-Tabelle unter Berücksichtigung dynamischer Faktoren
  - Auslastung (bspw. Netz-Stau)
  - Links, Netz-Topologie (Struktur des Graphen)

Routing Verfahren (Überblick)



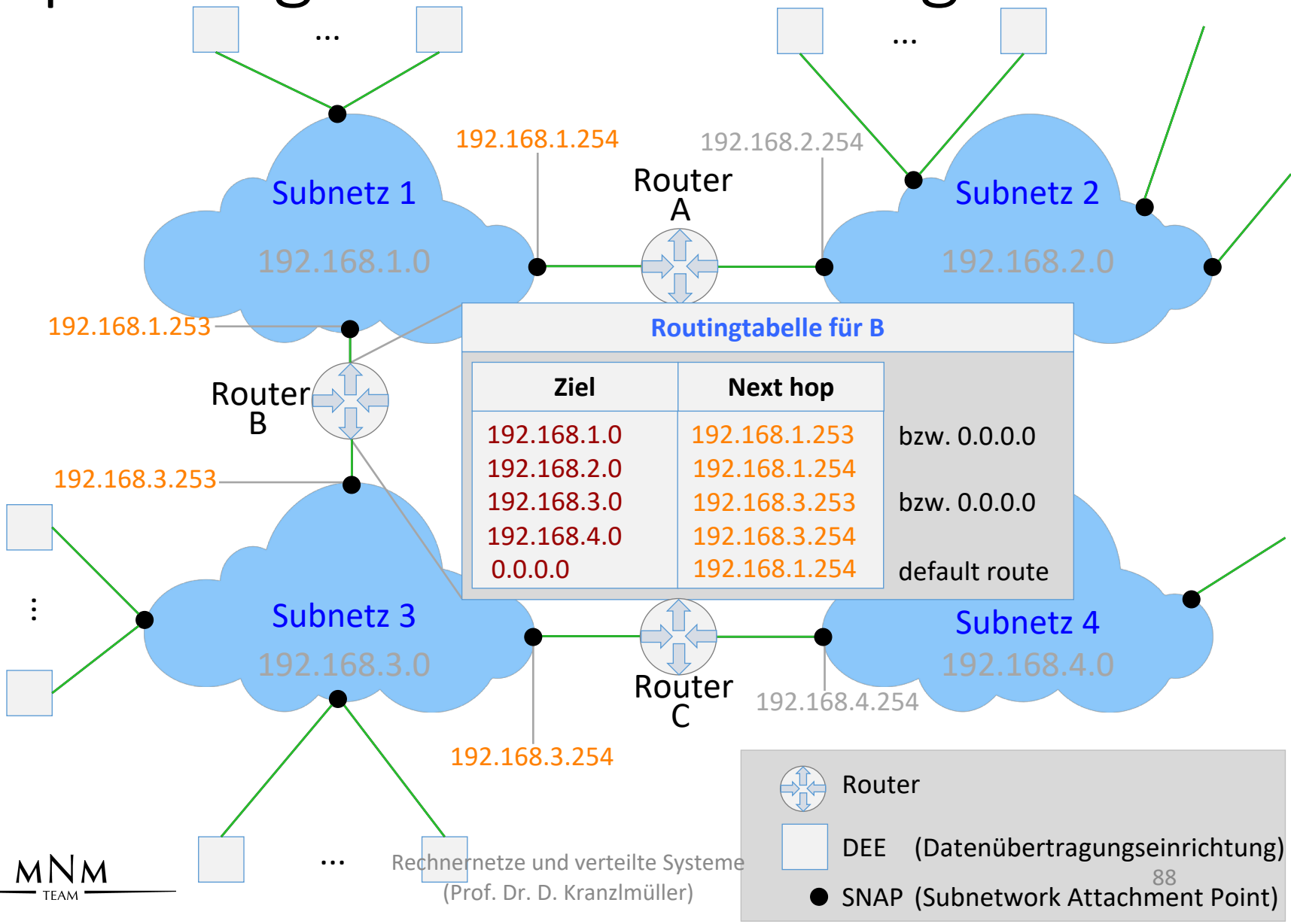
Weitere Verfahren:

- isoliertes Verfahren, nicht adaptiv: Flooding
- isoliertes Verfahren, lastabhängig: Hot Potato

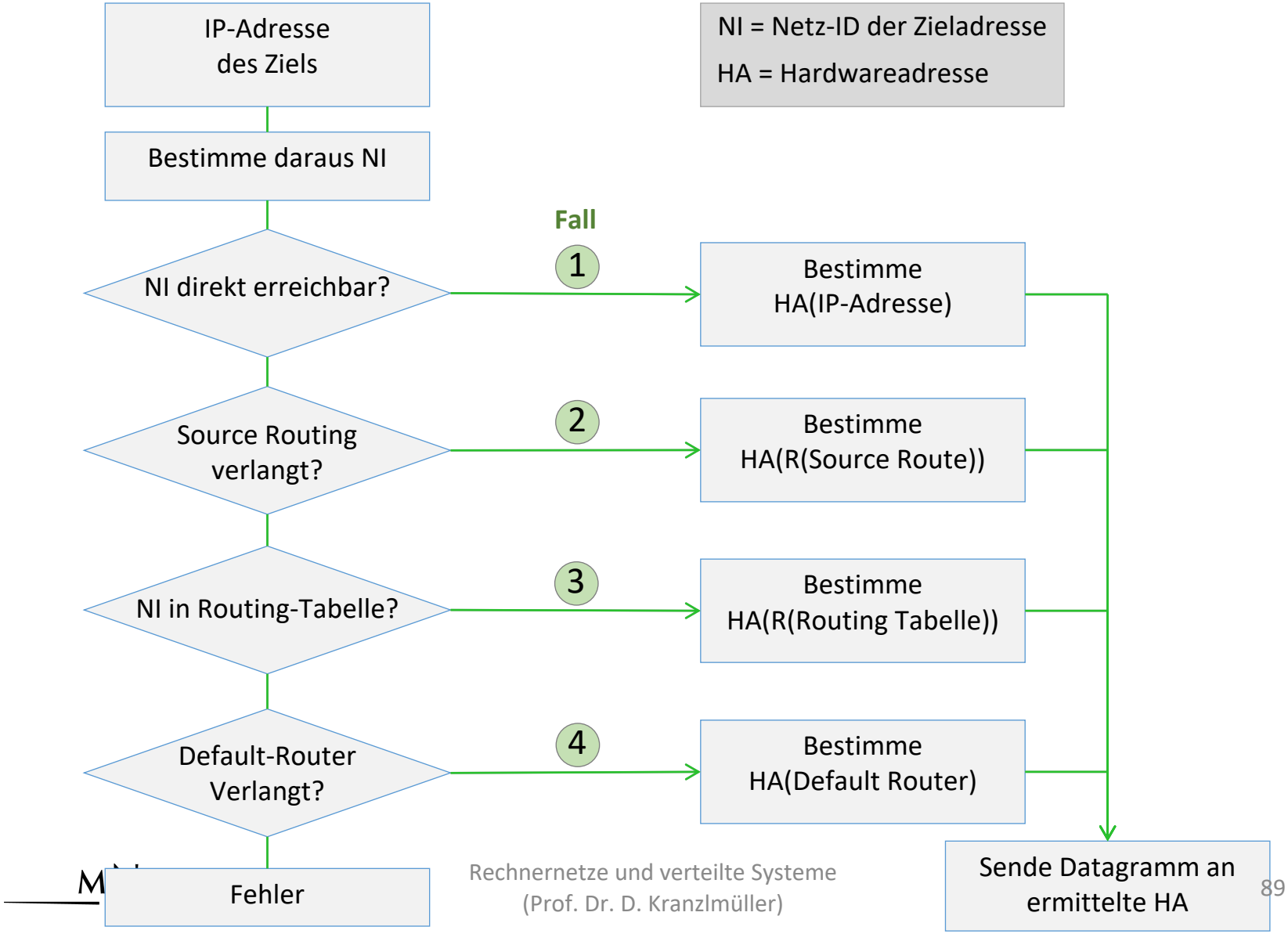
Routingtabellen

- Realisieren Vorgaben zur Nachrichten-Weiterleitung
- Bestimmen die Entscheidungsfindung in IWUs/Routern
- **Default Route**: spezieller Eintrag, wird benutzt, wenn kein anderer Eintrag passt.
- Felder eines Tabelleneintrags (eine Zeile):
  - Ziel (Netz oder Host)
  - Netzmaske (Angabe zur Bestimmung der Netzadresse)
  - Gateway (d.h. Router)
  - Metrik (Kostenwert)
  - Schnittstelle (engl. Interface) (Zielleitung)
- Auf Unix-Derivaten (z.B. Linux): $ route -n

Bsp.: Wegewahl mit Routingtabelle



Wegewahlentscheidung im Router



Link State Routing

- **Annahme**: Globales Wissen über die Netz-Topologie (Kanten, Knoten, Kosten) ist vorhanden.
- Praxis: Erlangen des globalen Wissens über Broadcast (Link State Broadcast)
  - Jeder Router sendet Informationen über die direkt verbundenen Links an alle anderen Router.

  →Link State Broadcast (separater Algorithmus)
- Link State auch bekannt als Dijkstra Algorithmus
  - Benannt nach dem Erfinder Edsger W. Dijkstra [1]

[1] https://en.wikipedia.org/wiki/Edsger_W._Dijkstra

Link State Routing

Jeder Knoten (Quelle) hat folgende Information, die iterativ berechnet wird:

- D(v): Geringste Kosten von dem Knoten zum Ziel-Knoten v auf Basis des aktuellen Wissens (Iteration)
- p(v): Vorheriger (previous) Knoten entlang des Pfades mit den geringsten Knoten zum Ziel v
- N‘: Teilmenge der Knoten N.
  - v ist in N‘ ($v \in N'$), falls der Pfad von der Quelle zu v definitiv bekannt ist.

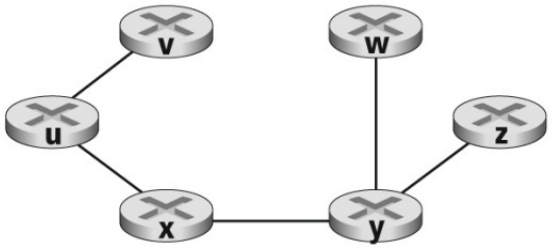
Link State Algorithmus

```
1  Initialization:
2     N' = {u}
3     for all nodes v
4       if v is a neighbor of u
5         then D(v) = c(u,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    update D(v) for each neighbor v of w and not in N':
12          D(v) = min( D(v), D(w) + c(w,v) )
13    /* new cost to v is either old cost to v or known
14     least path cost to w plus cost from w to v */
15 until N'= N
```

Link State Algorithmus: Beispiel

Schritt	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	*u*	*2, u*	*5, u*	*1, u*	∞	∞
1	*ux*	*2, u*	*4, x*		*2, x*	∞
2	*uxy*	*2, u*	*3, y*			*4, y*
3	*uxyv*		*3, y*			*4, y*
4	*uxyvw*					*4, y*
5	*uxyvwz*					

Abb. 1: Link State Algorithmus aus Sicht von Knoten u



Ziel	Leitung
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Abb. 2: Kostengünstigster Pfad und resultierende Routing-Tabelle für Knoten u.

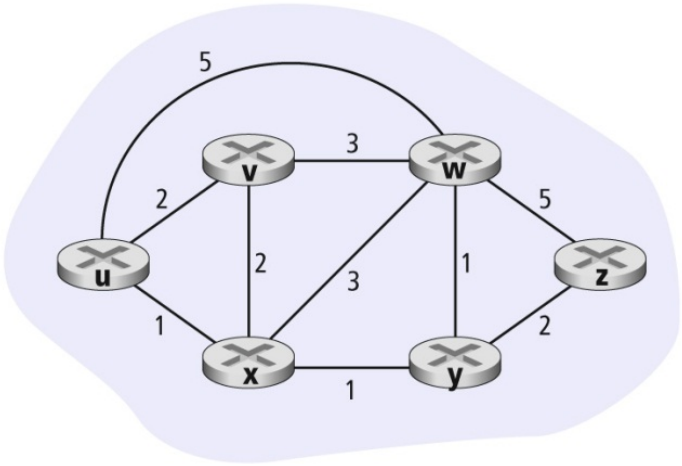


Abb.: gegebener Graph

Distance Vector

- Im Vergleich zu Link State iterativ, asynchron und verteilt (kein globales Wissen).

- **Verteilt**: Jeder Knoten empfängt Informationen von seinen Nachbarn, berechnet ein neues Teilergebnis und verteilt es wieder.
- **Iterativ**: Der Algorithmus berechnet schrittweise neue Information, bis keine neue Information mehr ausgetauscht wird (*self terminating)*
- **Asynchron**: Die Knoten müssen nicht synchronisiert in einem Schritt miteinander kommunizieren.

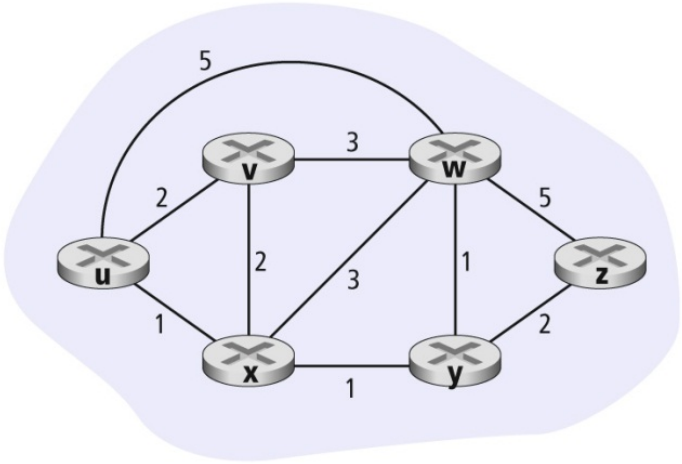
Bellman-Ford Theorem

$d_x(y)$: Pfad mit geringsten Kosten von Knoten x zu y.

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

Wobei min_v über alle Nachbarn von x berechnet wird.

Bellman-Ford (Beispiel)



- Quell-Knoten sei u, Ziel-Knoten z
- u hat 3 Nachbarn: (v,w,x)
- Zu erkennen ist
  - $d_v(z) = 5, d_w(z) = 3, d_y(z) = 3$
- Einfügen in die Gleichung ergibt
  - $c(u,v) = 2, c(u,w) = 5, c(u,x) = 1$
  - $d_u(z) = min\{2+5, 5+3, 1+3\} = 4$

→ Aus Bellman-Ford Gleichung resultiert ein Eintrag der Routing-Tabelle für den Pfad $u \rightarrow z\ via\ x$

Distance Vector Algorithmus

- Jeder Knoten x hält folgende Zustände:
  - c(x, v): Kosten von x zu allen unmittelbaren Nachbarn v.
  - $\boldsymbol{D}_x = \{D_x(y): y \in N\}$: Distanzvektor aktueller Kostenschätzungen von x zu allen anderen Knoten y
  - $\boldsymbol{D}_v = \{D_v(y): y \in N\}$: Distanzvektoren der unmittelbaren Nachbarn v von x
- Jeder Knoten sendet in Zeit-Intervallen eine Kopie seines Distanz-Vektors an die unmittelbaren Nachbarn
- Wenn ein Knoten x einen Distanz-Vektor eines unmittelbaren Nachbarn empfangen hat, wird der eigene Distanz-Vektor mit Bellmand-Ford Gleichung aktualisiert:
  - $D_x(y) = min_v\{c(x,y) + D_v(y)\}\ f\ddot{u}r\ alle\ Knoten\ y \in N$

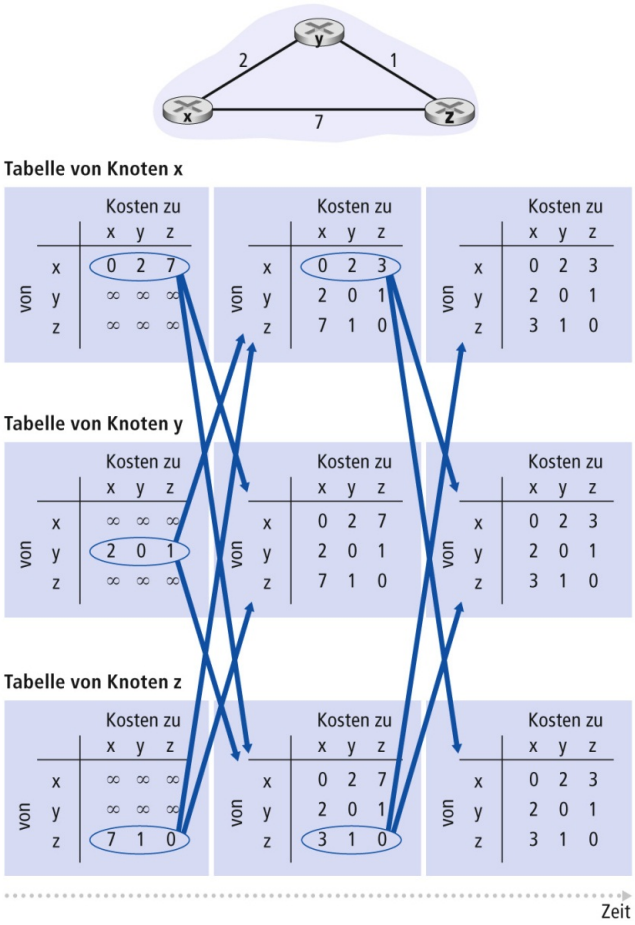
Distance Vektor (Pseudo-Code)

- Alle Knoten führen folgenden Algorithmus aus:

```
1  Initialization:
2     for all destinations y in N:
3        D_x(y) = c(x,y)   /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        D_w(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector D_x = [D_x(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       D_x(y) = min_v{c(x,v) + D_v(y)}
15
16    if D_x(y) changed for any destination y
17       send distance vector D_x = [D_x(y): y in N] to all neighbors
18
19 forever
```

Distance Vektor (Beispiel)

- Linke Spalte sind die initialen Routing-Tabellen
  - Da noch keine Information ausgetauscht wurde, sind bspw. Bei x die y und z-Zeilen leer.
- Anschließend werden die Distanz-Vektoren schrittweise ausgetauscht
- Der Algorithmus terminiert, wenn sich keine Information ändert.



Hierarchisches Routing

- Bisher: Algorithmen berechnen Routing-Information über das gesamte (globale) Netz
- Herausforderungen
  - Skalierung: Informationen über alle Knoten und Kanten in einem umfangreichen Netz (bspw. Internet) zu aufwendig
  - Administrative Unabhängigkeit: Aufteilen des Netzes in administrativ eigenständige Domänen sinnvoll.

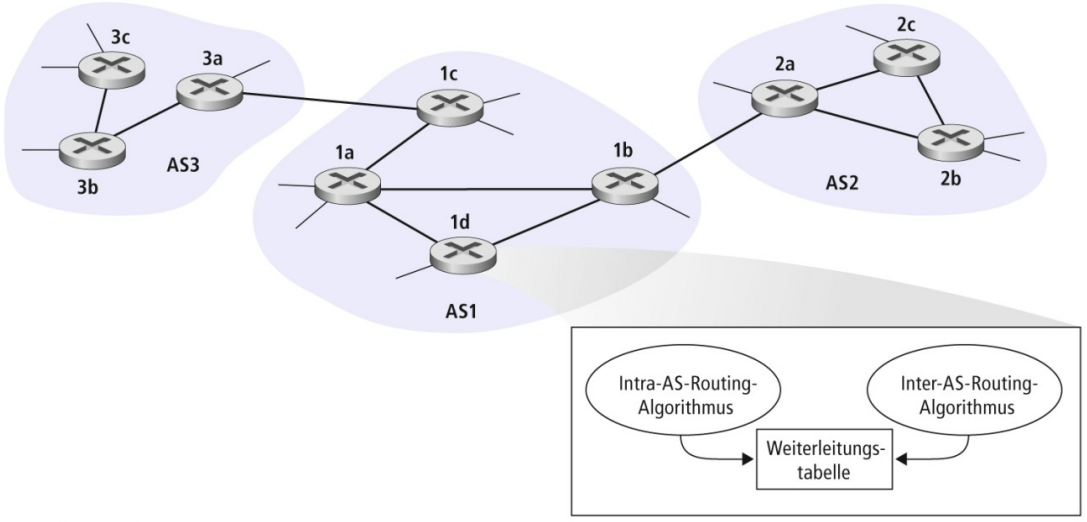
→ Autonome Systeme

Autonome Systeme (AS)

- Alle Router innerhalb eines AS verwenden denselben Routing-Algorithmus (bspw. Die bisher vorgestellten Algorithmen)
- Routing zwischen unterschiedlichen AS erfolgt durch **Gateway Router**.
- **Intra AS** Routing: Propagierung von Routing Informationen innerhalb eines AS.
- **Inter AS** Routing: Propagierung von Routing-Informationen zwischen **verschiedenen** AS

Inter-AS Routing (Szenario 1)

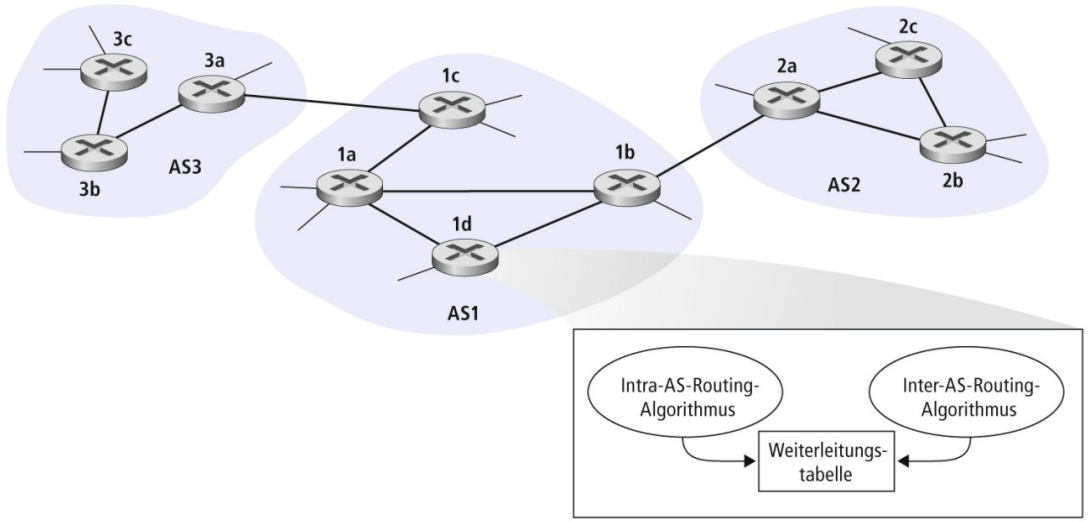
- Szenario 1: Subnetz *x* ist für Router 1d aus AS1 erreichbar via AS3 (Information erhalten durch Inter-AS Routing Protokoll).
  - Router 1d bekommt Information durch das Intra-AS Routing Protokoll
  - Router 1d nimmt einen Eintrag in die entsprechende Routing-Tabelle auf: (x, 1c)



Inter-AS Routing (Szenario 2)

- Szenario 2: Subnetz *x* ist für Router 1d aus AS1 erreichbar via AS3 oder auch AS2 (Information erhalten durch Inter-AS Routing Protokoll).
  - Wie entscheidet Router1d, ob ein Paket zu Subnetz x via AS3 (Gateway 1c) oder AS2 (Gateway 1b) vermittelt wird?
  - Häufige Lösung in der Praxis: **Hot Potato Routing**
    - Ziel: Paket aus interner Queue so schnell wie möglich loswerden.

    → Leite Paket daher an den Gateway-Router mit geringsten Kosten (Distanz), wie aus dem Intra-AS Routing Protokoll ermittelt.



Routing im Internet

- Das Internet ist hierarchisch zusammengesetzt aus mehreren Autonomen Systemen
- Intra-AS Routing Protokolle
  - Routing Internet Protocol (RIP)
  - Open Shortest Path First (OSPF)
- Inter-AS Routing Protokoll
  - Border Gateway Protocol

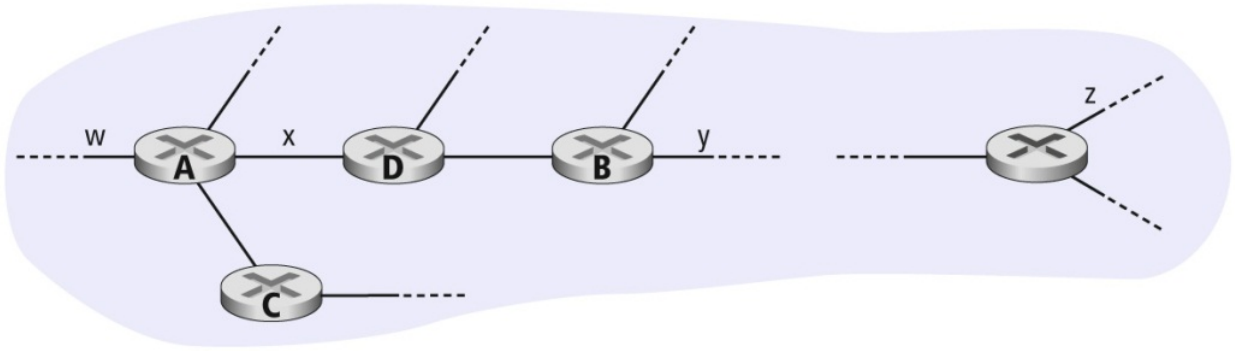
Routing Internet Protocol (RIP)

- Distanz-Vektor (DV) basiertes Protokoll
- Kostenfunktion
  - Klassisches DV: Kosten waren zwischen zwei Routern definiert (der Einfachheit halber).
  - RIP: Kosten sind definiert von Quell-Router zu Ziel-Router.
- Maximale Kosten: 15 Hops
  - Daher kann RIP ausschließlich auf AS mit maximaler Distanz von 15 zwischen zwei Routern eingesetzt werden.
- In RIP kommuniziert ein Knoten x in 30 Sekunden Intervallen mit den unmittelbaren Nachbarn via **RIP Reponse Message** (auch **RIP Advertisement** genannt).
  - Max. 25 Ziel-Subnetze innerhalb des AS mit jeweils den aktuell bekannten Kosten (Distanz in Hops).
- Wenn ein unmittelbarer Nachbarknoten innerhalb von 180 Sekunden kein RIP Advertisement verschickt, wird der Nachbar als *down* markiert.

  → Information wird via RIP Advertisement an die anderen Nachbarn propagiert.

RIP Beispiel (1)

a) Gegeben: Teilansicht eines (beliebigen) autonomen Systems.



b) Initiale Routing Tabelle von Router D.

Zielsubnetz	Nächster Router	Anzahl von Hops zur Zieladresse
w	A	2
y	B	2
z	B	7
x	–	1
...	...	...

RIP Beispiel (2)

c) Router D erhält folgendes Advertisement von Router A:

Zielsubnetz	Nächster Router	Anzahl von Hops zur Zieladresse
z	C	4
w	–	1
x	–	1
...	...	...

d) Router D gleicht lokalen Routing-Tabelle ab und stellt fest, dass die Route zu Subnetz **z** kostengünstiger wurde (via Router A).

Zielsubnetz	Nächster Router	Anzahl von Hops zur Zieladresse
w	A	2
y	B	2
z	A	5
...	...	...

Open Shortest Path First (OSPF)

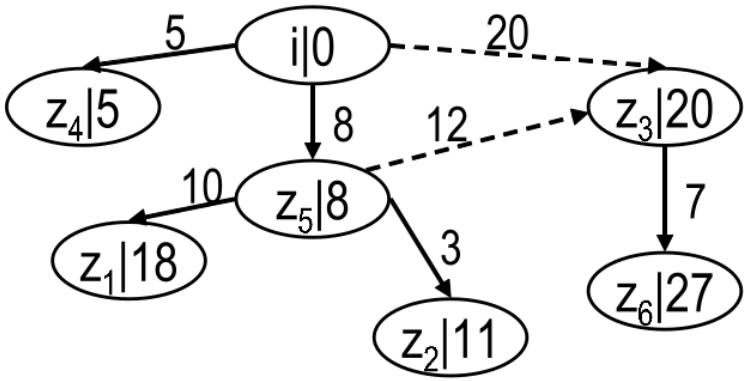
- Wurde als Nachfolger für RIP konzipiert und ist ein offenes Protokoll (RFC 2328).
- Nutzt zwei bisher bekannte Verfahren:
  a) Flooding von Link State Informationen. (Link State Broadcast). Dadurch wird ein vollständiger Graph des Autonomen Systems konstruiert.
  b) Anwendung des Dijkstra Algorithmus zur Berechnung eines Subnetz-Baums mit dem Router selbst als Wurzel-Knoten (**Quelle-Senken Baum**).
- Link-Kosten können durch den Netz-Administrator definiert werden. Beispiele:
  - Alle Link-Kosten auf 1 setzen (Minimum Hop Routing)
  - Umkehrfunktion des verfügbaren Durchsatzes: Links mit **hohem Durchsatz** werden gegenüber denen mit **niedrigem Durchsatz bevorzugt**.

Quelle-Senken-Baum Wegetafel

Wegetafel Knoten i

Ziel	Nachbar	Kosten
z_1	z_5	18
z_2	z_5	11
z_3	z_3, z_5	20
z_4	z_4	5
z_5	z_5	8
z_6	z_3, z_5	27

Quell-Senken-Baum



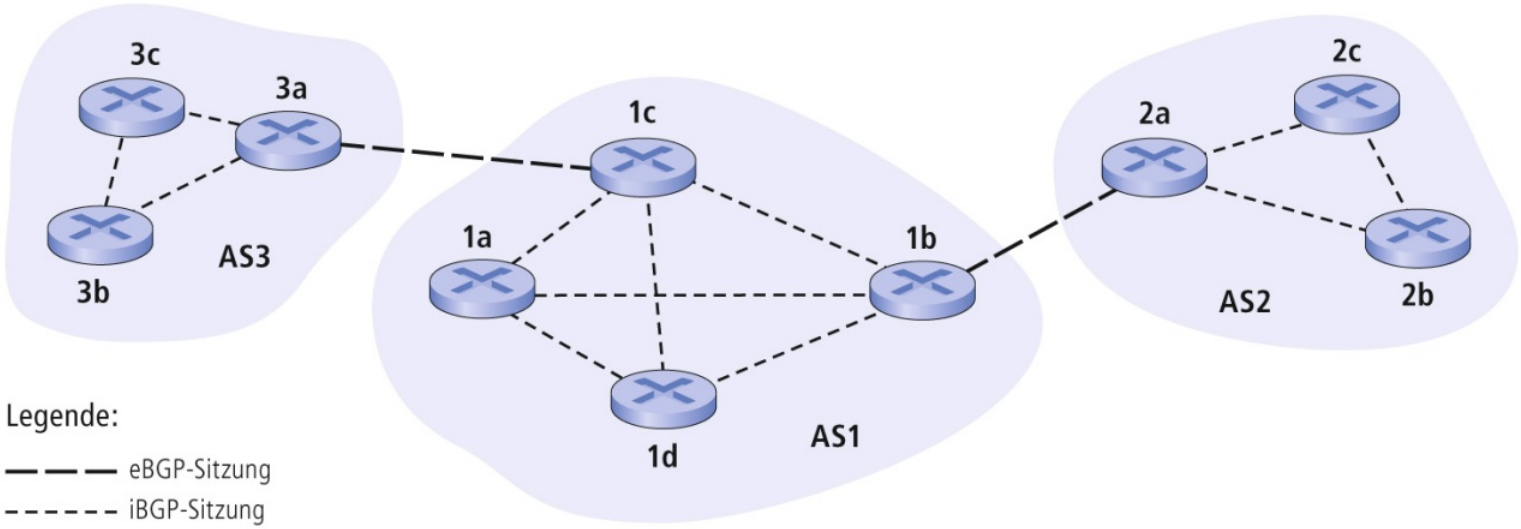
OSPF Flooding

- Jeder Link übermittelt die aktuelle Routing Tabelle an die unmittelbaren Nachbarn
  - Sobald sich in der lokalen Routing Tabelle etwas ändert.
  - Auf jeden Fall alle 30 Minuten (periodisch). Dadurch wird (laut RFC) Robustheit erreicht.
- Kommunikation von OSPF Nachrichten erfolgt basiert auf IP. Heißt also:
  - OSPF muss selbst Funktionalität für zuverlässigen Nachrichtentransfer implementieren (normalerweise Layer 4).

OSPF Vorteile gegenüber RIP

- Mehrere Pfade mit identischen Kosten sind erlaubt. Es musst nicht der vollständige Traffic von Quell- zum Ziel-Router über dieselbe Router vermittelt werden.
- Hierarchien innerhalb eines autonomen Systems.
- Integrierte Unterstützung für Unicast und Multicast (in der Vorlesung nicht im Detail diskutiert).

BGP (Border Gateway Protocol)



- Kommunikationsaustausch zwischen 2 Routern (**BGP Peers**) erfolgt via semipermanenter TCP-Verbindung (**BGP Session**)
  - eBGP: Verbindung zwischen zwei Router unterschiedlicher AS
  - iBGP: Verbindung zwischen zwei Router innerhalb eines AS

BGP Routenaustausch

- Arbeitsweise ähnlich wie bei Distanz Vektor
- Ziele in BGP sind keine Hosts, sondern Subnetze. Ausgetauscht werden daher CIDR Präfixe.
  - Beispiel
    - Ein AS hat die 4 Subnetze 138.16.{64,65,66,67}/24
    - Übermittelt an die Nachbar AS wird dann der längste gemeinsame CIDR Präfix 138.16.64/22
- Wenn ein Gateway Router (in einem AS) einen neuen eBGP CIDR Präfix empfängt, wird dieser Präfix innerhalb des AS weiterpropagiert. Dadurch dehnt sich die Information an weitere AS aus.
- Jeder neu empfangene („gelernte“) CIDR Präfix führt zu einem Eintrag in der Routing Tabelle.
- Die Auswahl des Pfades basiert nicht nur auf einem simplen Metrik (Anzahl Hops), sondern ist regelbasiert (RFC 1930) durch BGP Attribute, die an einen CIDR Pfad

Fragen zu Wegewahl/Routing

- Warum war es sinnvoll, im Internet RIP durch OSPF abzulösen?
- Was ist der Kernalgorithmus bei Link State Routing?
- In welchen Fällen ist Flooding ein sinnvolles Verfahren?
- Nennen Sie Kostenfunktionen, die dem Optimalitätskriterium genügen?
- Was versteht man unter autonomen Systemen?