

Vorlesung Einführung in Rechnernetze

7. Netzkopplung und Vermittlung

Prof. Dr. Martina Zitterbart

Dipl.-Inform. Martin Florian, Markus Jung (M.Sc.), Matthias Flittner (M.Sc.)
[zitterbart | florian | m.jung | flittner]@kit.edu

Institut für Telematik, Prof. Zitterbart



© Peter Baumung

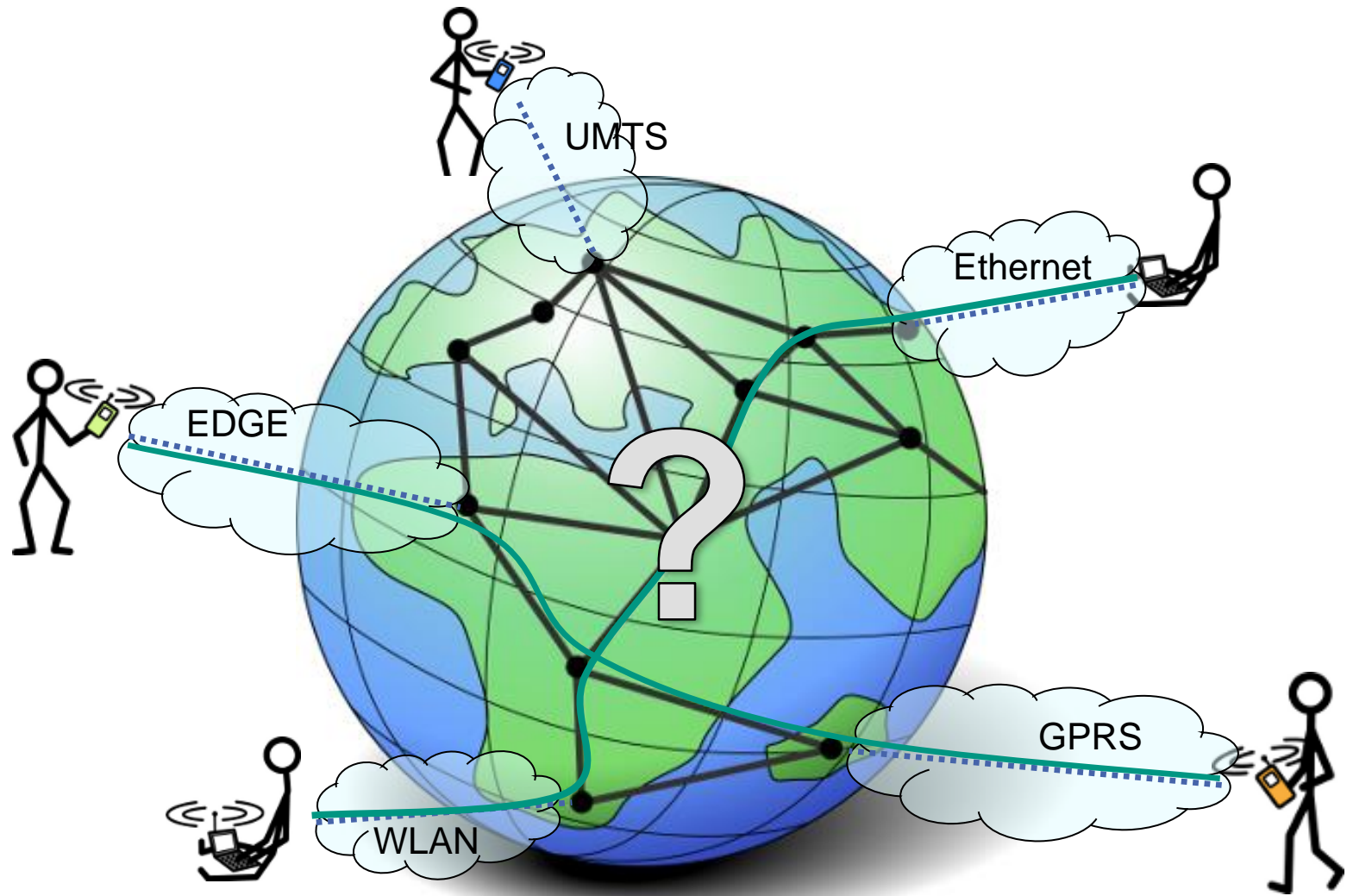
1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
- 7. Netzkopplung und Vermittlung**
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Motivation
2. Vermittlungstechniken
 1. Leitungsvermittlung
 2. Paketvermittlung
 3. Datagrammvermittlung
 4. Virtuelle Verbindungen
 5. Nachrichtenvermittlung
3. Netzkopplung
 1. Repeater
 2. Brücke
 3. Router
4. Vermittlung im Internet

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
- 7. Netzkopplung und Vermittlung**
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Motivation
2. Vermittlungstechniken
 1. Leitungsvermittlung
 2. Paketvermittlung
 3. Datagrammvermittlung
 4. Virtuelle Verbindungen
 5. Nachrichtenvermittlung
3. Netzkopplung
 1. Repeater
 2. Brücke
 3. Router
4. Vermittlung im Internet

7.1 Motivation: Globale Netze und Kommunikation

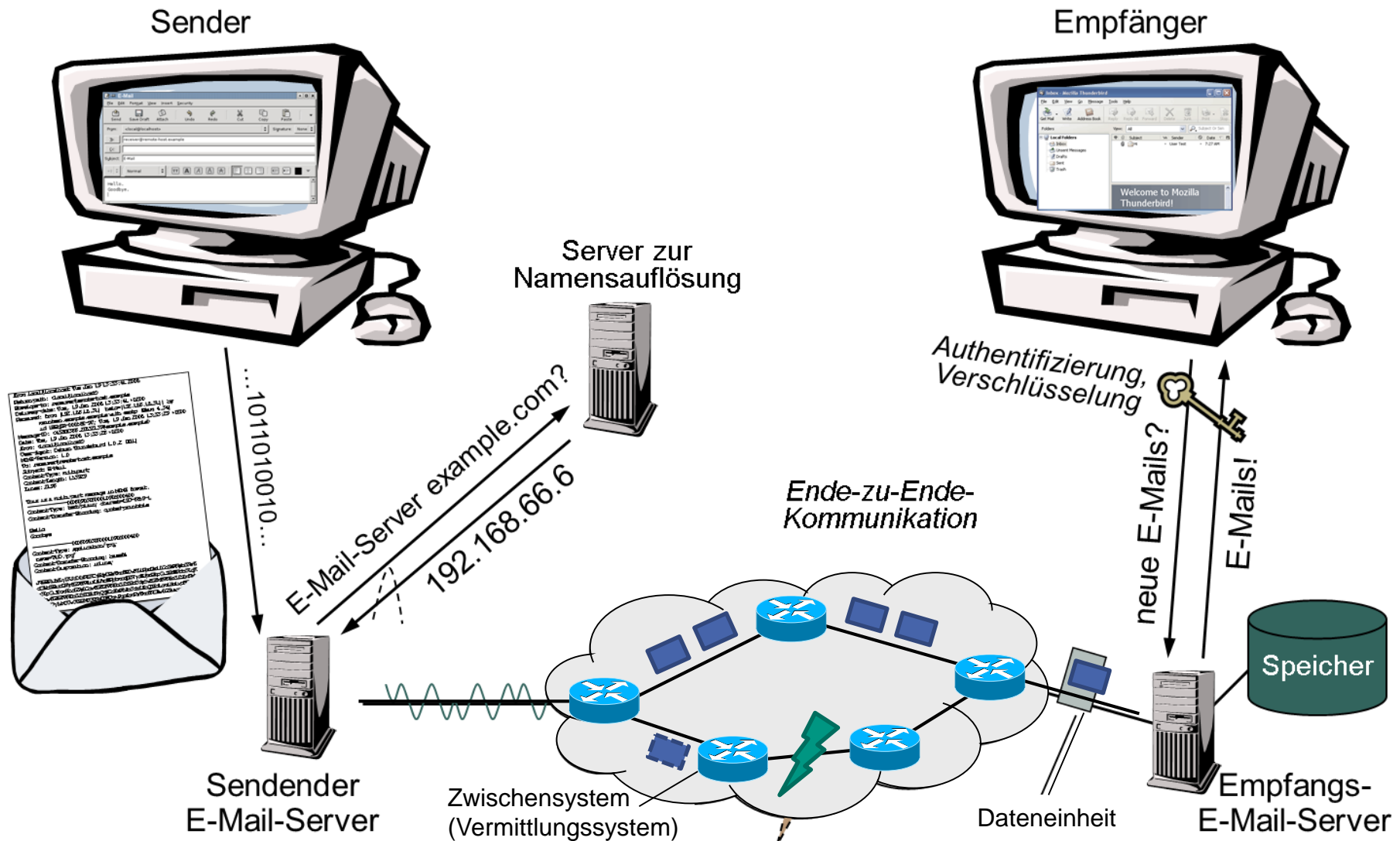


- Wesentliche Herausforderungen
 - **Heterogene Netze**: Wie koppelt man unterschiedliche Netze?
 - Unterschiedliche Adressen, Bandbreiten, Verzögerungszeiten, ...
 - **Wegfindung (Routing)**: Wie findet man den Weg von A nach B?
 - **Weiterleitung**: Wie gelangen die Daten von A nach B?

- Potentielle Lösungsstrategien
 - Zusätzliches Adressierungsschema und -abbildung
 - Einbau von „Vermittlern“
 - Aufbau von Netzkarten / Graphen (vgl. Straßennetz)
 - Nutzung von Graphenalgorithmien (vgl. Navigation im Straßennetz)
 - ...

- Wegfindung? Wieso unterschiedliche Wege?
- Wie ermittelt man die Wege im Internet?

Netzkopplung und Vermittlung in unserem Beispiel

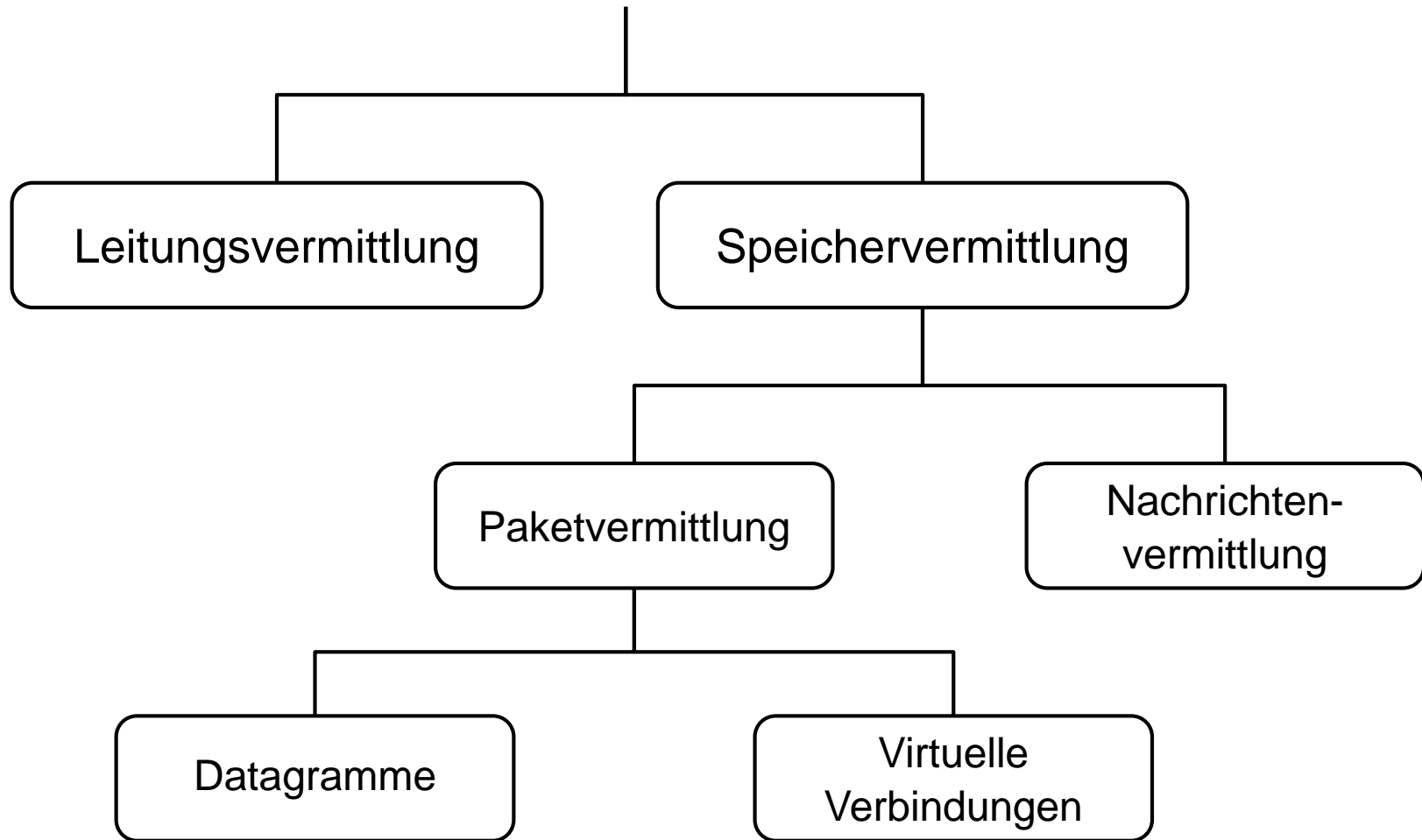


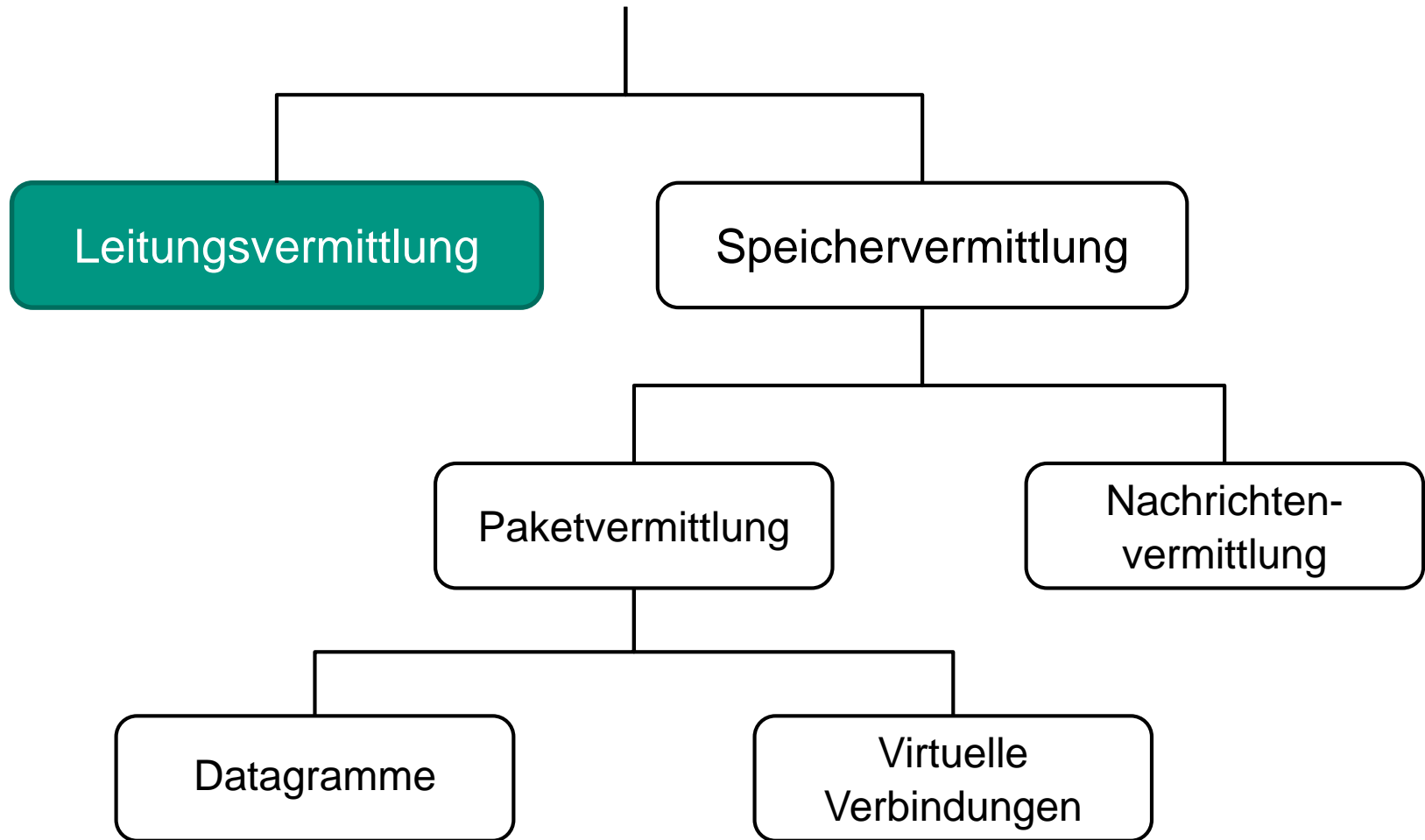
1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
- 7. Netzkopplung und Vermittlung**
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Motivation
2. Vermittlungstechniken
 1. Leitungsvermittlung
 2. Paketvermittlung
 3. Datagrammvermittlung
 4. Virtuelle Verbindungen
 5. Nachrichtenvermittlung
3. Netzkopplung
 1. Repeater
 2. Brücke
 3. Router
4. Vermittlung im Internet

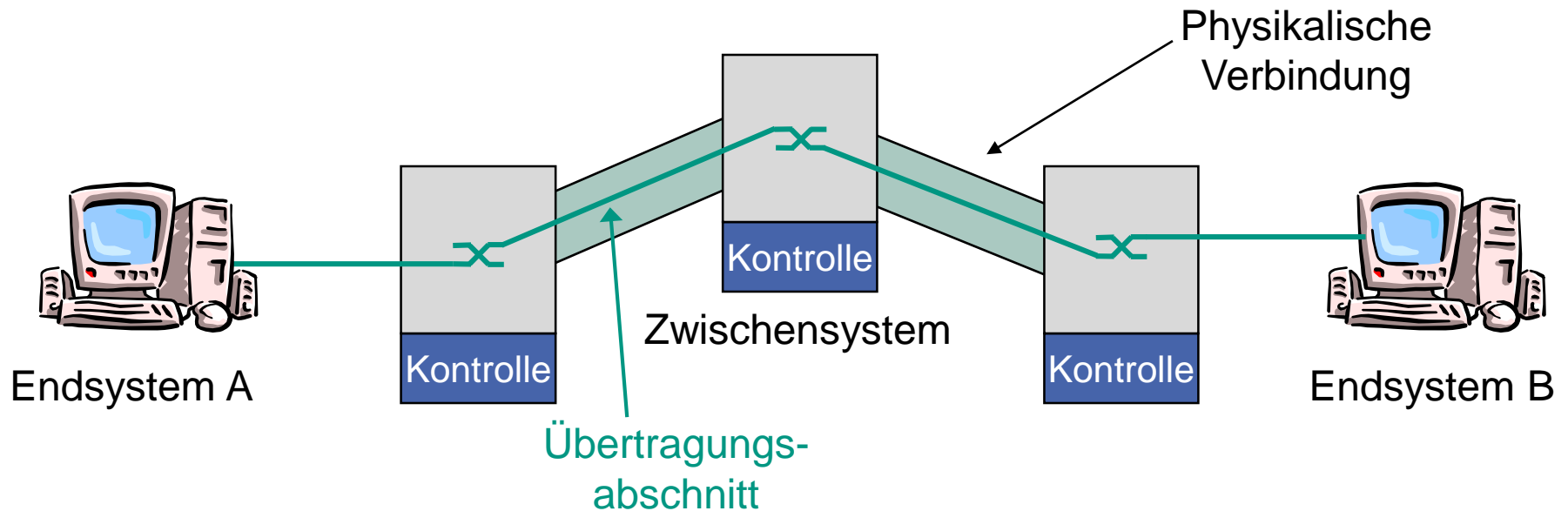
7.2 Vermittlungstechniken

- Folgende Vermittlungstechniken lassen sich unterscheiden





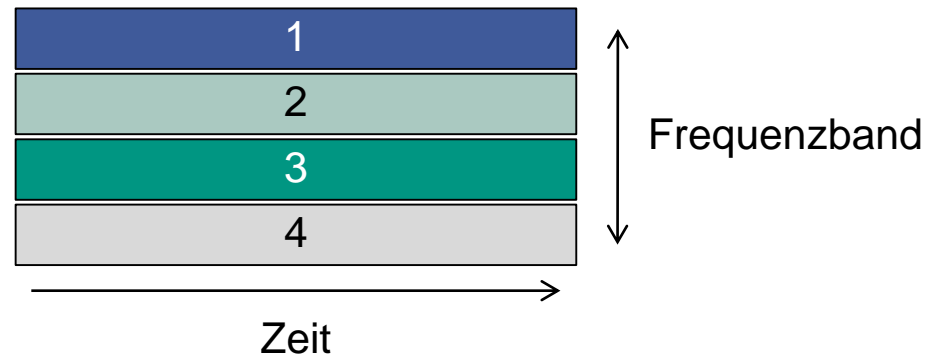
7.2.1 Leitungsvermittlung



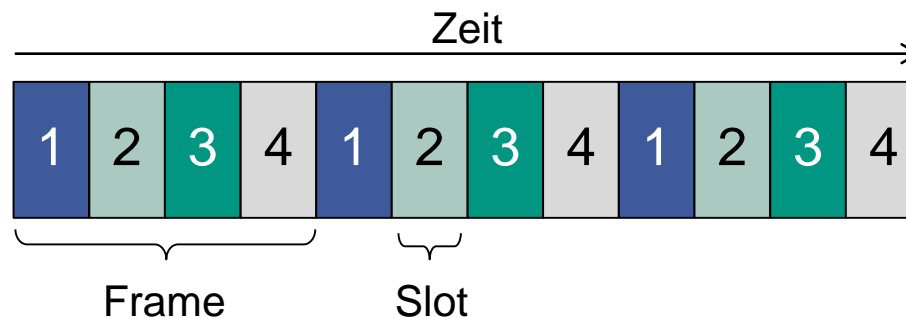
- Verbindungsorientierte Kommunikation
- Verbindung erhält einen durchgehenden Kanal für die ausschließliche Nutzung mit konstanter Bandbreite
 - Hierzu werden aufeinanderfolgende Übertragungsabschnitte miteinander verknüpft

Leitungsvermittlung: Multiplexing

- Bei Leitungsvermittlung ist **starres Multiplexing** möglich
 - **Frequenzmultiplex**
 - Feste Zuweisung von Übertragungskanal und Frequenzabschnitt



- **Zeitmultiplex**
 - Feste Zuweisung von Übertragungskanal und Zeitabschnitt (time slot)

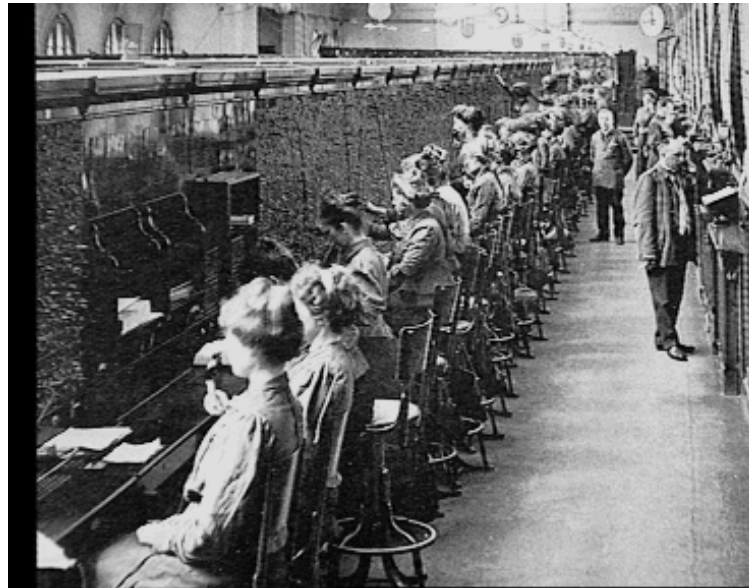


Leitungsvermittlung: Eigenschaften

- Aufbau eines *durchgehenden, nicht-speichernden* Übertragungskanal („Leitung“) zwischen den Endsystemen
- Zugesicherte, feste Bandbreite
- Übertragungsverzögerungen sind auf physikalisch bedingte signaltechnische Laufzeiten beschränkt
- Bitfolgen werden reihenfolgetreu übertragen
 - Absenderreihenfolge beim Empfänger beibehalten (*wire-like feature*)
- Vermittlung in den Zwischensystemen erfordert keine zusätzliche Kontrollinformation zur Adressierung
 - Bei Paketvermittlung ist solche Kontrollinformation erforderlich

Leitungsvermittlung: Telefonnetz

- Einsatzgebiet: „klassische“ Telekommunikationsnetze
 - Analoges Telefonnetz
 - Public Switched Telephone Network, PSTN
... früher mit manueller Vermittlung

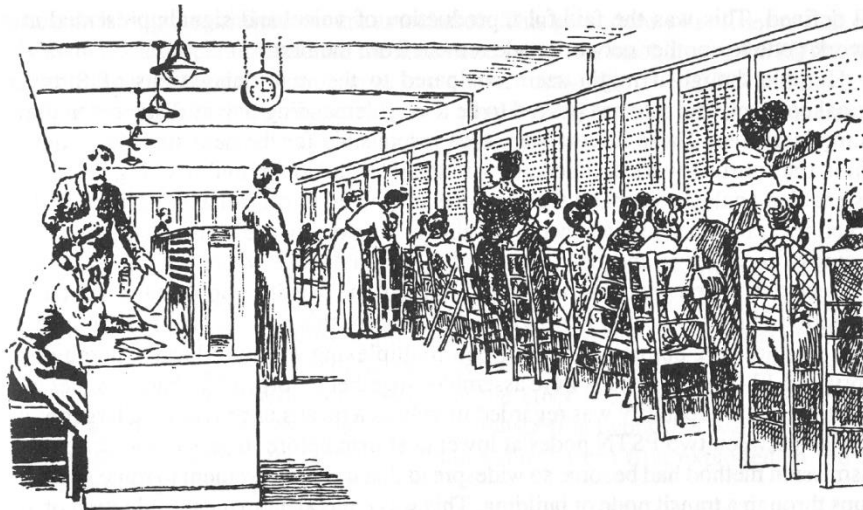


- Digitale Netze zur Sprachkommunikation (nicht nur)
 - ISDN, GSM

Entwicklung der Telekommunikation

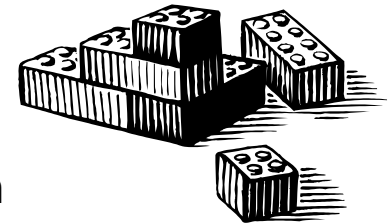
■ Architektur des frühen Telefonnetzes

- Einige hundert Nutzer in einem Umkreis von ca. 1 km des Vermittlungsgebäudes
- Jedes Telefon ist per Kupferkabel angeschlossen
- **Operator** kann Verbindung zwischen zwei Kabeln herstellen



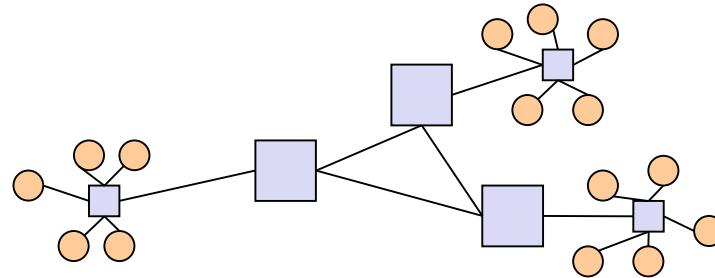
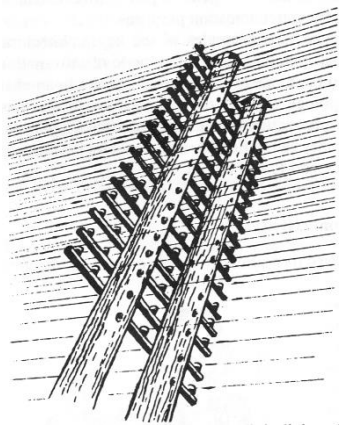
... bereits hier lassen sich Kernkomponenten von Telekommunikationsnetzen erkennen

- Endsysteme, Übertragungsabschnitte, Vermittlungsstellen
Übertragungsnetze



Entwicklung des Telefonnetzes

- Einfluss des **Wachstums** auf Topologie und Kontrolle
 - Lange Übertragungsleitungen ermöglichen Verbindung zwischen Vermittlungsstellen
 - Aus Sternnetz wird ein **vermaschtes Netz**



- Wachsende Teilnehmerzahlen und steigende Zahl von Ferngesprächen
 - Große Vermittlungsräume
 - Netzmanagement wird schnell zum **Engpass**

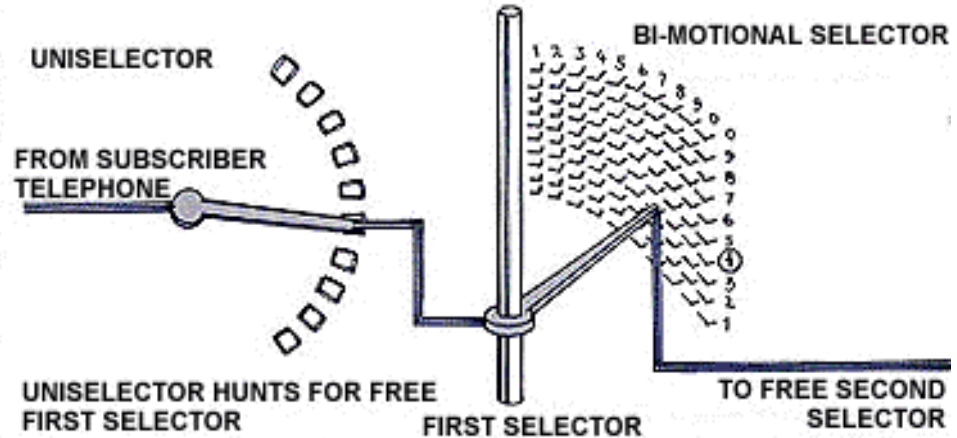
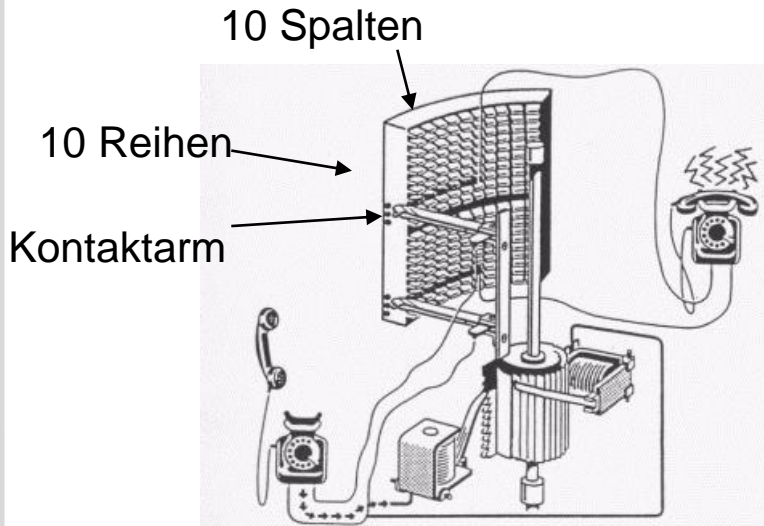
Entwicklung des Telefonnetzes

- Step-by-step Mechanismus von Armon Strowger
 - **Automatisierung des Verbindungsmanagements**
 - Strowger war Bestatter; Frau des Konkurrenten arbeitete in der Vermittlung ... Aufträge gingen an den Konkurrenten
 - Patentierung 1891
 - Gründung der Firma „Strowger Automatic Telephone Exchange“
 - Teilnehmer können Verbindung selbst herstellen
 - Ziffern 0-9
 - 100 Anschlüsse
 - Kaskadierbare für höhere Zahl an Anschlüssen
 - **Elektromechanische Verbindungsvermittlung**
 - → Entwicklung eines standardisierten Nummernplans
 - ... teilweise heute noch im Einsatz

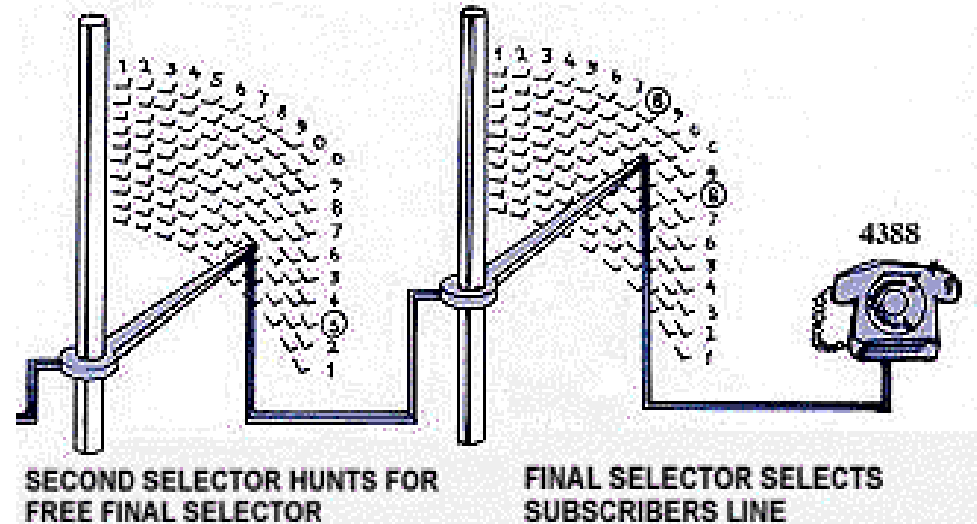
- ... wachsende Anzahl an Teilnehmern und Vermittlungsstellen
 - Keine Änderungen in der Architektur
 - Definition von **User-Network Interface (UNI)** und **Network-Network Interface (NNI)**
 - Vermittlungsstellen je über eine Kupferleitung verbunden



Step-by-step Mechanismus (Hebdrehwähler)



The Step by Step Process Continued



www.sigtel.com/tel_tech_sxs.html

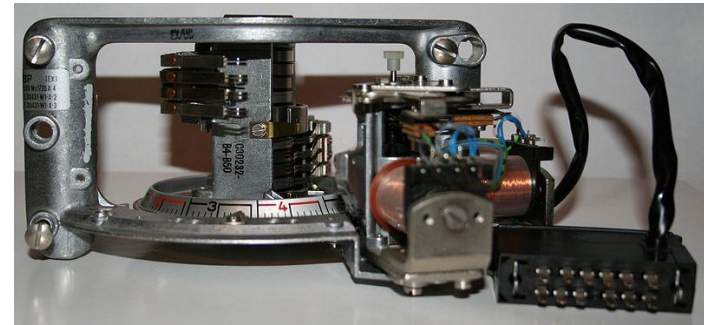
Beispiele



Kleines „Switchboard“
Quelle: phworld.org



Hebdrehwähler
Quelle: wikipedia



Edelmetallkontakt-Motor-Drehwähler
Quelle: wikipedia



**Western Electric Step by Step
Community Dial Office (CDO)**
Quelle: phworld.org

Beispiele

- Erste öffentliche automatische Telefonvermittlung in UK
 - 1912 eröffnet
 - Bis 1995 in Betrieb
 - Genutzt für die Vermittlung innerhalb eines Gebäudes



Batterien →



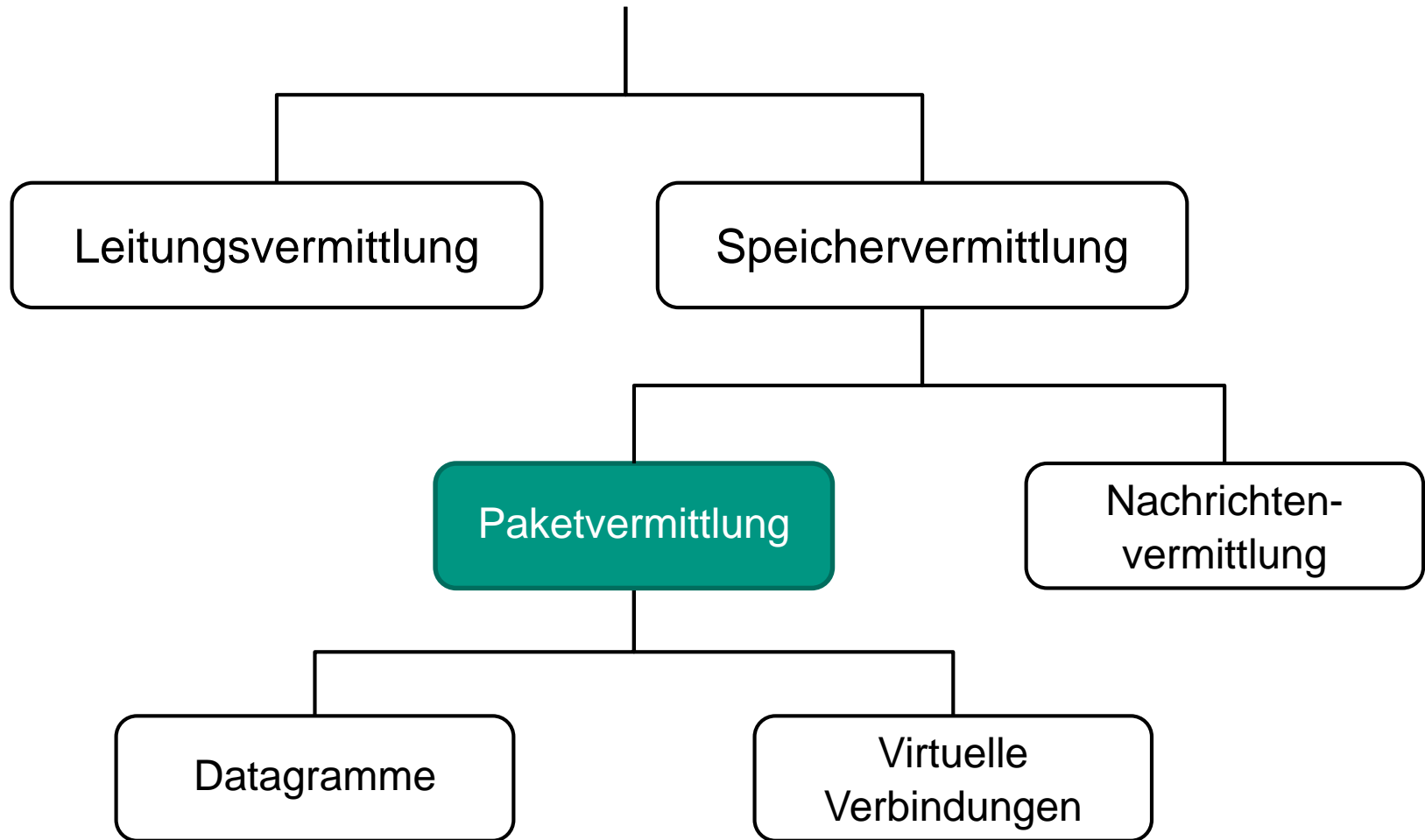
<http://www.seg.co.uk/telecomm/automat1.htm>

Beispiele

■ Heutzutage...



Digitale Vermittlungsstelle
Quelle: wikipedia



7.2.2 Paketvermittlung

- Weiterleitung aufgrund von Kontrollinformation in den Dateneinheiten
 - Zieladresse in Datagrammen
 - Lokale Kennung bei virtuellen Verbindungen
 - Wechselnde Wege für aufeinanderfolgende Dateneinheiten durch das Netz möglich
 - Reihenfolgevertauschungen sind möglich

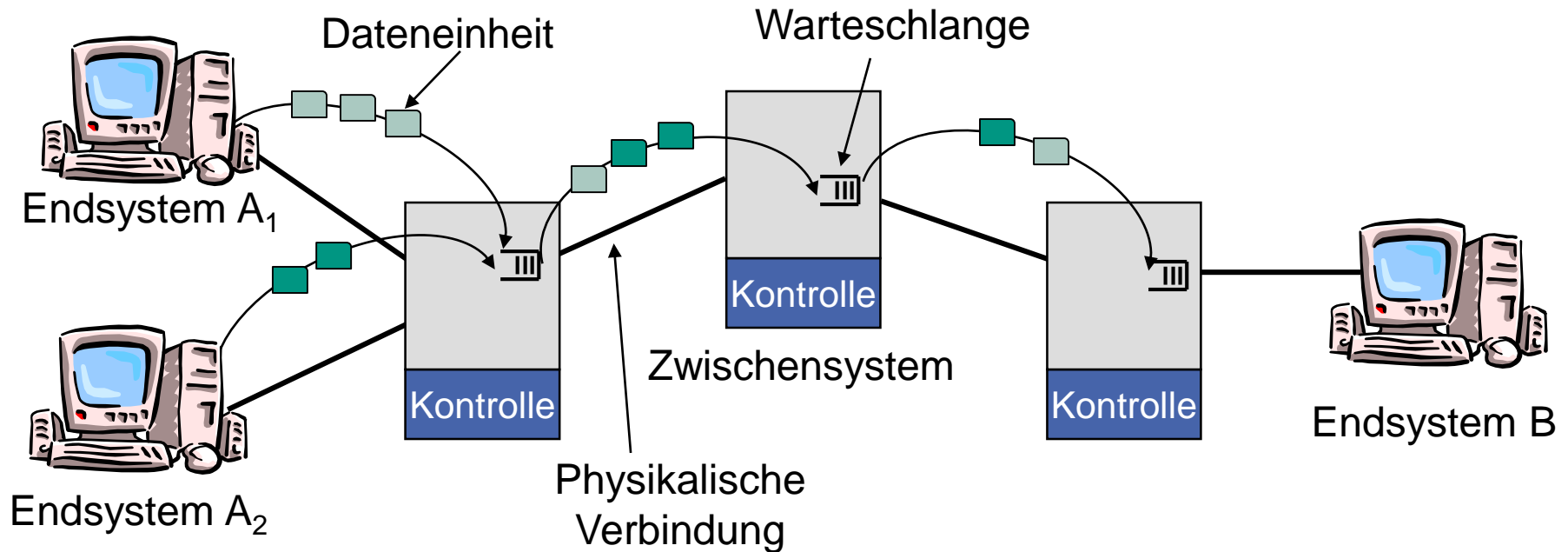
- Zwischensysteme verfügen über Speicher in Form von Warteschlangen
 - Pufferung der Dateneinheiten, falls Ausgangsinterface nicht frei
 - Verlust von Dateneinheiten möglich
 - Begrenzte Pufferkapazität in den Zwischensystemen

- Es besteht im Allgemeinen keine feste Zeitbeziehung zwischen den einzelnen zu vermittelnden Dateneinheiten

Paketvermittlung: Multiplexing

- I.d.R. Verwendung von **Zeitmultiplex**
 - Im Allgemeinen **keine Reservierungen** von Ressourcen (Zeitschlitz, Frequenzen)
- Bei Paketvermittlung spricht man von **statistischem Multiplexing**

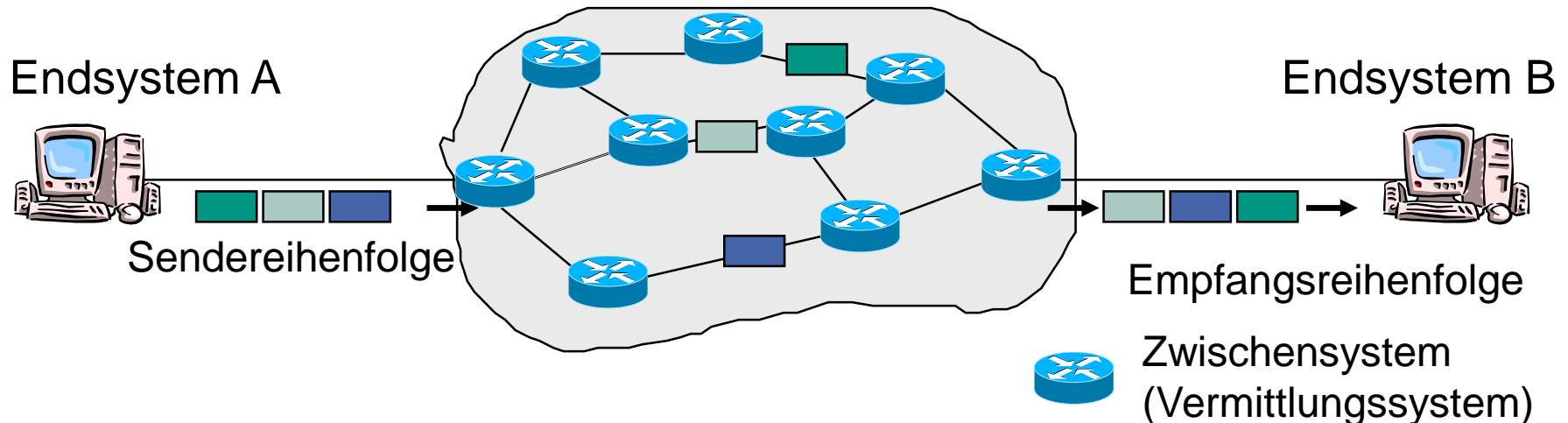
Paketvermittlung: Übermittlung von Dateneinheiten



- Abschnittsweise Übermittlung von Dateneinheiten
- Zwischenspeicherung (Pufferung) erfolgt ggf. in den Zwischensystemen in sog. **Warteschlangen** (engl. *Queue*)
- Varianten
 - Virtuelle Verbindungen - **verbindungsorientiert**
 - Datagramme - **verbindungslos**

7.2.3 Datagrammvermittlung

- Dateneinheiten (sog. **Datagramme**) werden als isolierte Einheiten betrachtet
- Zieladresse in jedem Datagramm enthalten
 - Keine Verbindungsaufbau und -abbau-Phase nötig
 - Keine Information pro Verbindung in den Zwischensystemen
- Dateneinheiten können das Netz auf unterschiedlichen Wegen durchlaufen
 - Überholvorgänge möglich
 - Datagramme können ungeordnet beim Empfänger ankommen

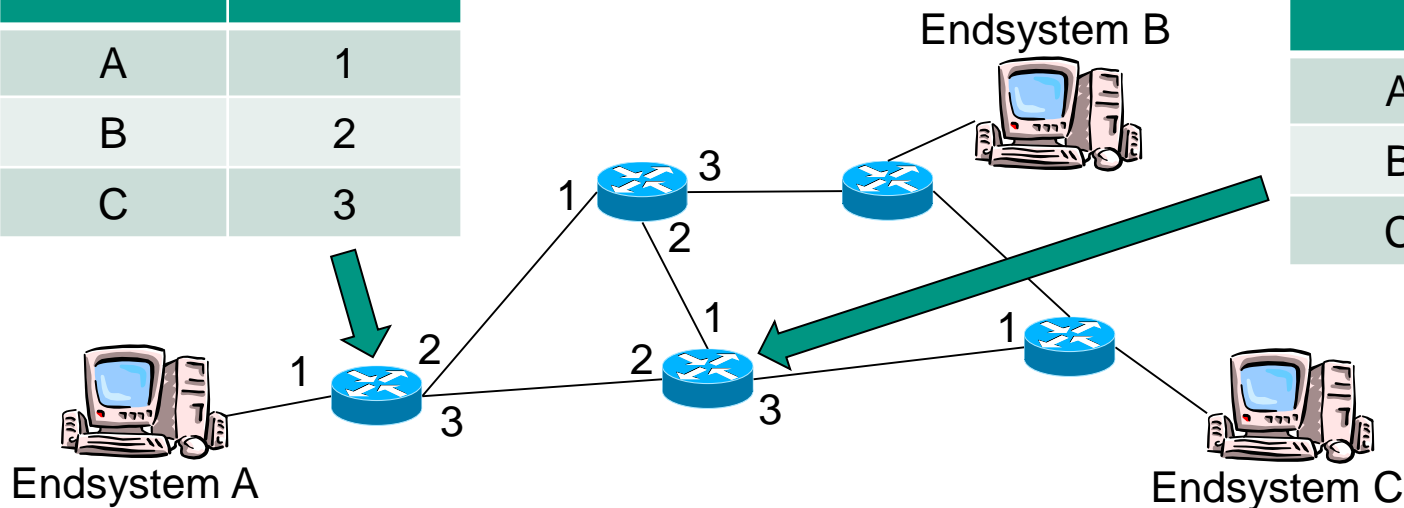


Datagrammvermittlung


- Zieladresse in jedem Datagramm erforderlich
 - Keine verbindungs-spezifische Zustandsinformation in den Zwischensystemen
- Weiterleitungstabelle in den Zwischensystemen
 - Durch Routingprotokolle aufgebaut

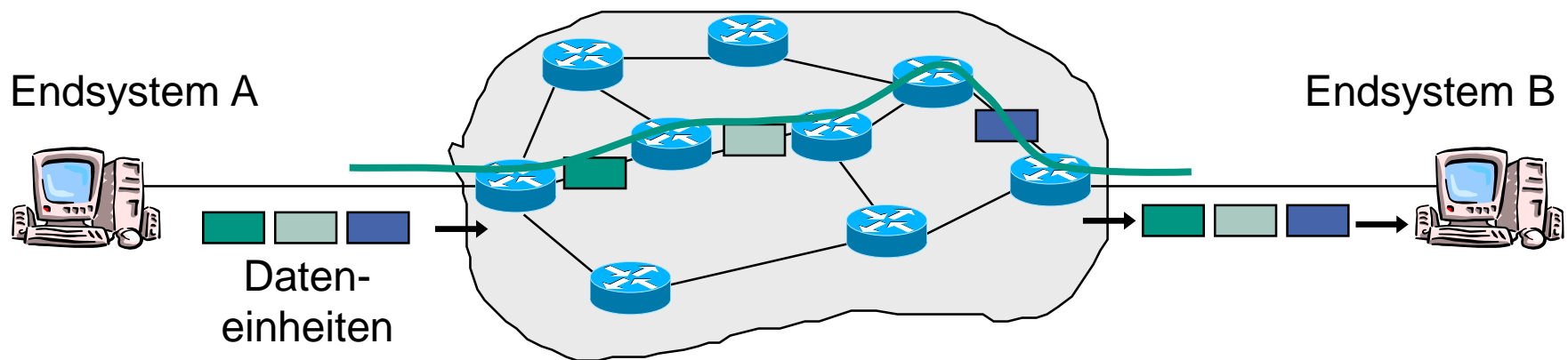
Address	Outgoing Interface
A	1
B	2
C	3

Address	Outgoing Interface
A	2
B	1
C	3



7.2.4 Virtuelle Verbindungen

- „Virtuelle Leitung“ / fester Übertragungsweg zwischen zwei Endsystemen
 - Alle Dateneinheiten folgen dem gleichen Weg, daher **Reihenfolgetreue**
 - Vermittlung von Dateneinheiten anhand von **Kennungen** (sog. *Virtual Circuit Identifier, VCI*)
 - Zieladresse nur beim Verbindungsaufbau nötig
- 3 Phasen
 - Verbindungsaufbau, Datenübertragung, Verbindungsabbau
- Einsatzbeispiel
 - Multiprotocol Label Switching (MPLS) 

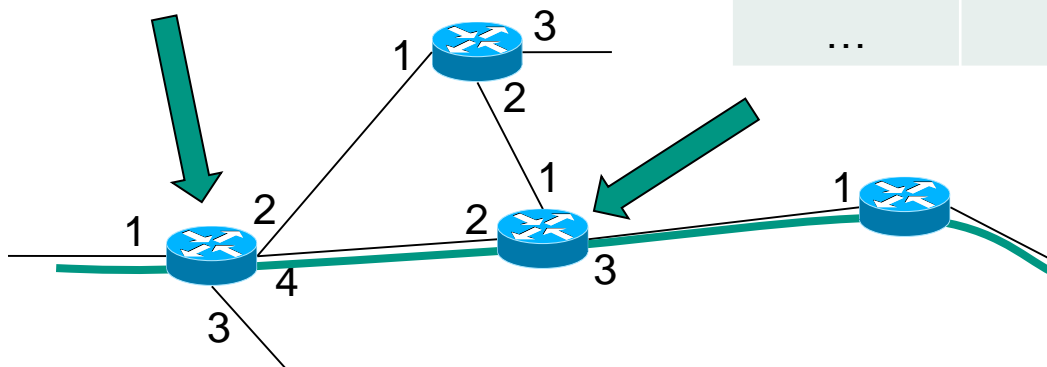


Virtuelle Verbindungen

- Kennungen sind im Allg. nur für einen Übertragungsabschnitt eindeutig
- 3 Phasen
 - Verbindungsaufbau: Virtuelle Verbindung wird durch die Festlegung von Kennungen auf den Zwischensystemen etabliert (Zieladresse nötig)
 - Datenübertragung: Daten werden anhand der Kennungen vermittelt
 - Verbindungsabbau: Virtuelle Verbindung wird abgebaut, d.h. Vermittlungsinformationen in den Zwischensystemen werden gelöscht

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
1	11	4	25
...

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	25	3	12
...



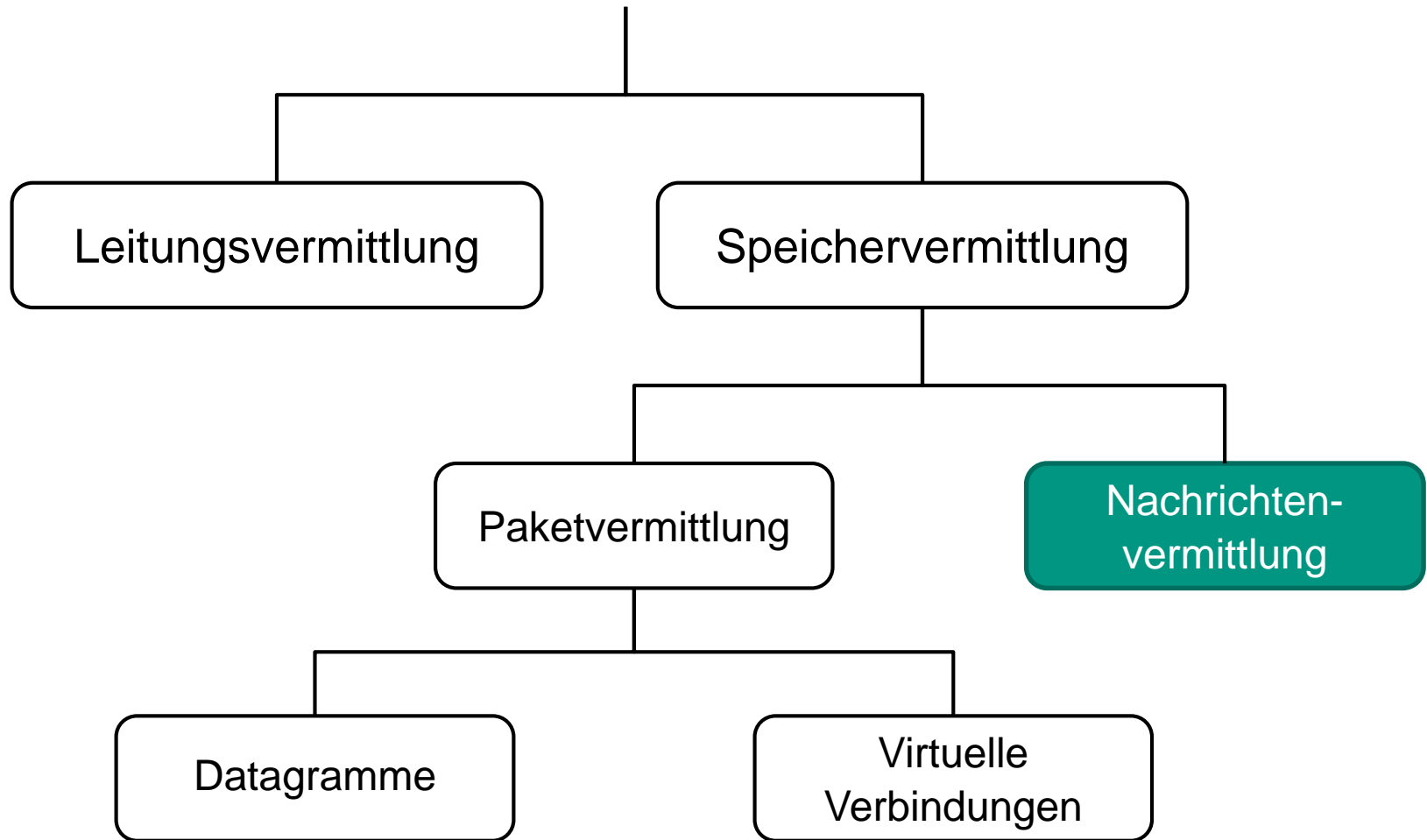
Varianten virtueller Verbindungen

- Feste virtuelle Verbindung (*Permanent Virtual Circuit, PVC*)
 - Längerfristig eingerichteter virtuelle Verbindungen
 - Aufbau in der Regel durch das Netzmanagement bzw. die Netzadministration
 - Vergleichbar mit einer Standleitung in leitungsvermittelnden Netzen

- Gewählte virtuelle Verbindung (*Switched Virtual Circuit, SVC*)
 - Virtuelle Verbindung wird bei Bedarf etabliert
 - Hierzu Signalisierungsprotokoll notwendig
 - Etablierung durch Nutzer ohne Eingreifen der Netzadministration

Virtuelle Verbindung vs. Datagramme

	Virtuelle Verbindung	Datagramme
Zieladresse	Nur während des Verbindungsaufbaus nötig	In jeder Dateneinheit benötigt → Overhead
Reihenfolge	Reihenfolgetreu	Nicht Reihenfolgetreu
Verbindungsaufbau und -abbau	Notwendig → Zeitlicher Overhead (Aufbau) → Verbindungsinformation in Zwischensystemen	Nicht nötig
Netzkomplexität und -funktionalität	→ Quality-of-Service einfacher realisierbar → Mehr Funktionalität im Netz	→ Keine Zustandshaltung im Netz → Mehr Funktionalität im Endsystem
Beispiele	MPLS	Internet



7.2.5 Nachrichtenvermittlung

■ Charakteristika

- Einheiten der Vermittlung: Nachrichten
 - Entsprechen anwendungsorientierten Gesichtspunkten
- Nachricht wird typischerweise mittels mehrerer Dateneinheiten vermittelt
 - Segmentierung und Reassemblierung
- In den Zwischensystemen
 - Reassemblierung der Nachrichten
 - Hierzu erforderlich
 - Alle Dateneinheiten, die zu einer Nachricht gehören, müssen an das gleiche nächste Zwischensystem weitergeleitet werden
- Ende-zu-Ende-Verzögerung ist im Vergleich zur Paketvermittlung deutlich höher

Anwendungsbeispiel: Delay-Tolerant Networks

■ Delay-Tolerant Networks (DTNs)

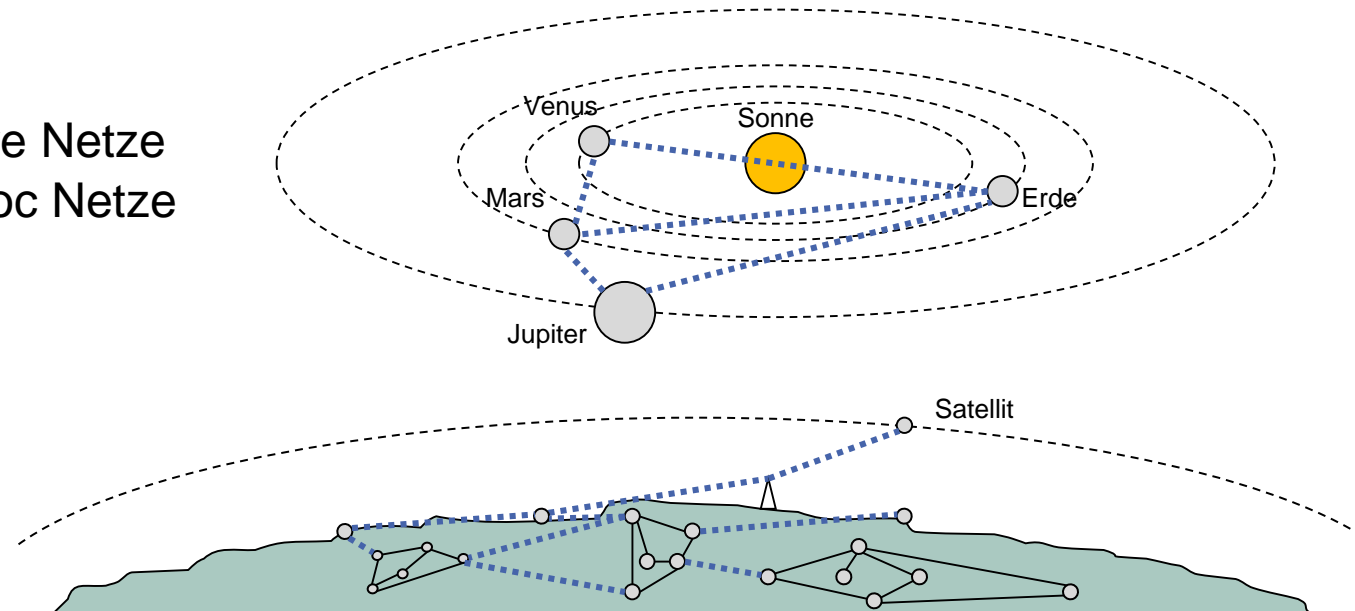


[Wart03] [Zhan06]

- Unterbrochene Verbindungen
 - Große Mobilität einzelner Systeme
 - Gegebenenfalls kein durchgängiger Ende-zu-Ende-Pfad verfügbar
- Lange oder variabel große Verzögerungen
 - Interplanetar bis zu einigen Minuten, trotz Lichtgeschwindigkeit
- Asymmetrische Datenraten
- Hohe Fehlerraten

■ Einsatzgebiete

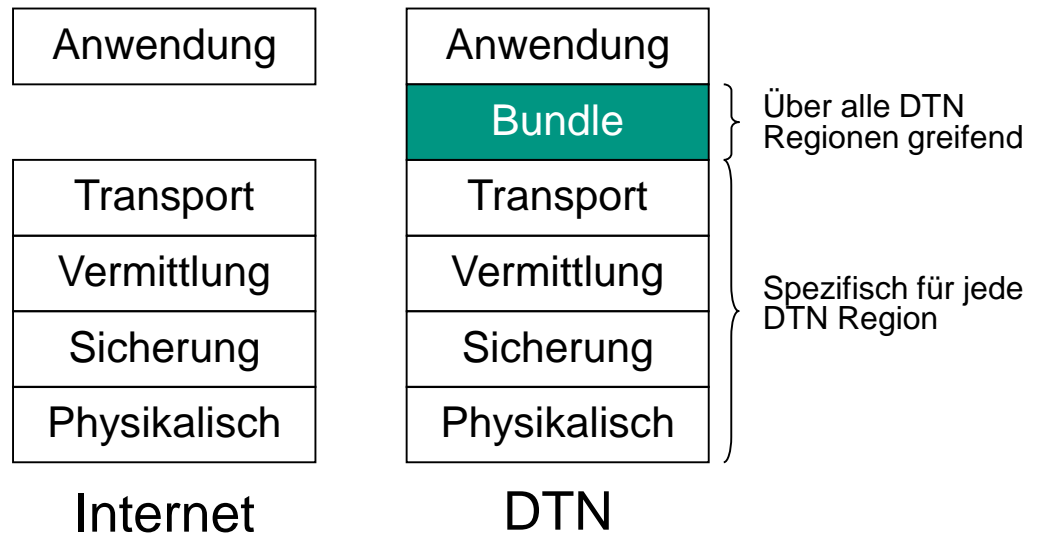
- Interplanetare Netze
- Mobile Ad-hoc Netze



Anwendungsbeispiel: Delay-Tolerant Networks

- Speichervermittelte Übertragung
 - Ermöglicht Kommunikation bei Verbindungen, die zwischenzeitlich unterbrochen werden
- Bilden eines **Overlays** über regionale Netze, inklusive des Internets
 - „Bundle Layer“

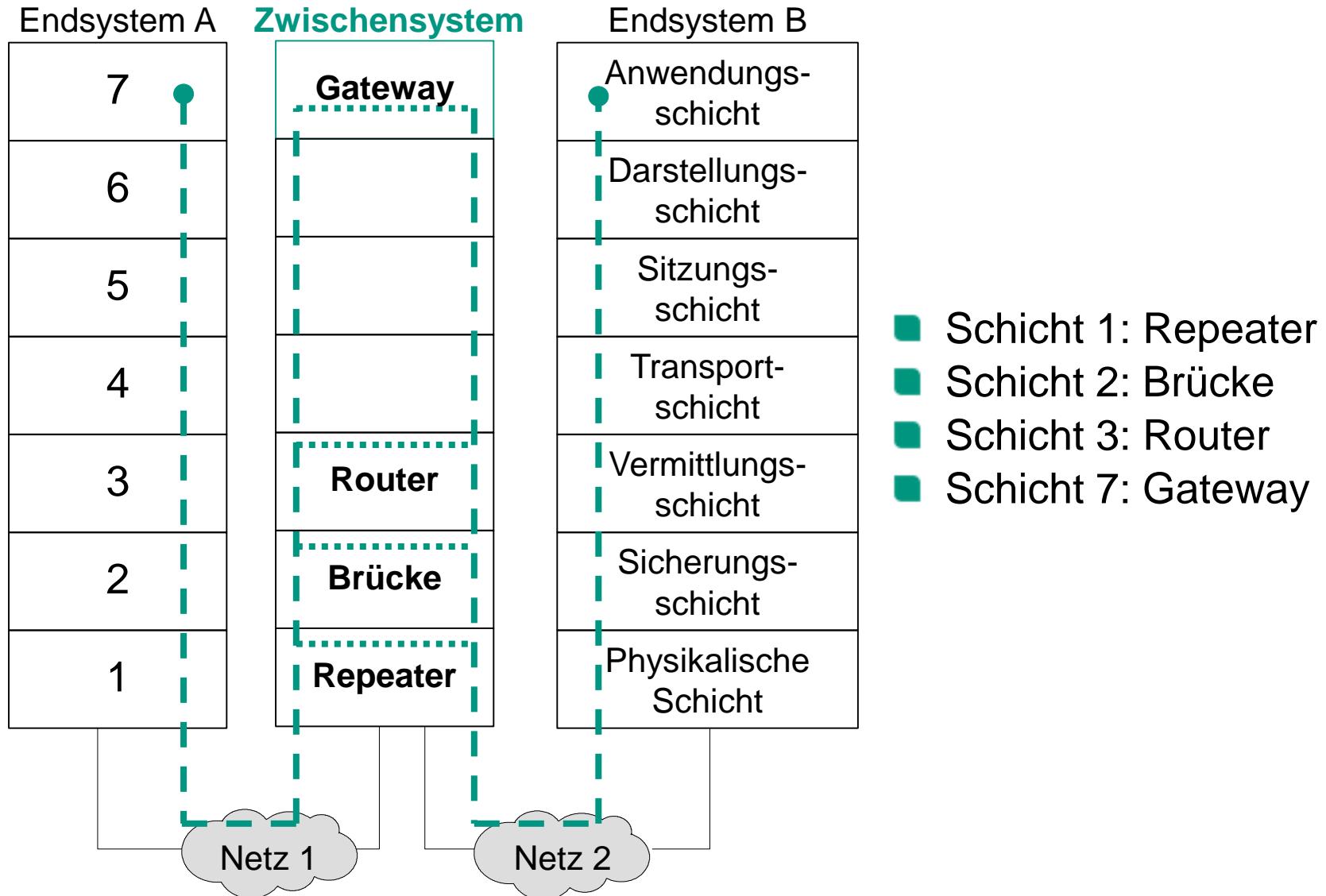
- **Bundle** fasst die regionalspezifischen Schichten zusammen
 - Ermöglicht Kommunikation zwischen verschiedenen Regionen



1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
- 7. Netzkopplung und Vermittlung**
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

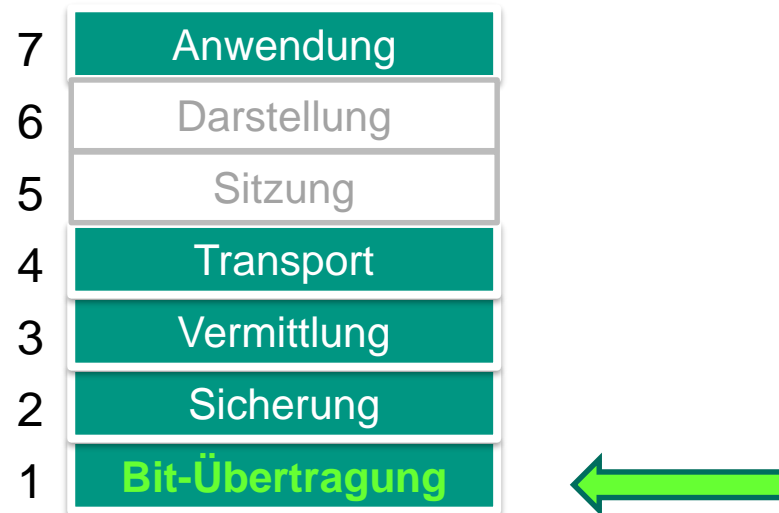
1. Motivation
2. Vermittlungstechniken
 1. Leitungsvermittlung
 2. Paketvermittlung
 3. Datagrammvermittlung
 4. Virtuelle Verbindungen
 5. Nachrichtenvermittlung
3. Netzkopplung
 1. Repeater
 2. Brücke
 3. Router
4. Vermittlung im Internet

7.3 Ebenen der Netzkopplung



Einordnung

- Wir befinden uns auf Schicht 1 des OSI-Referenzmodells



7.3.1 Repeater

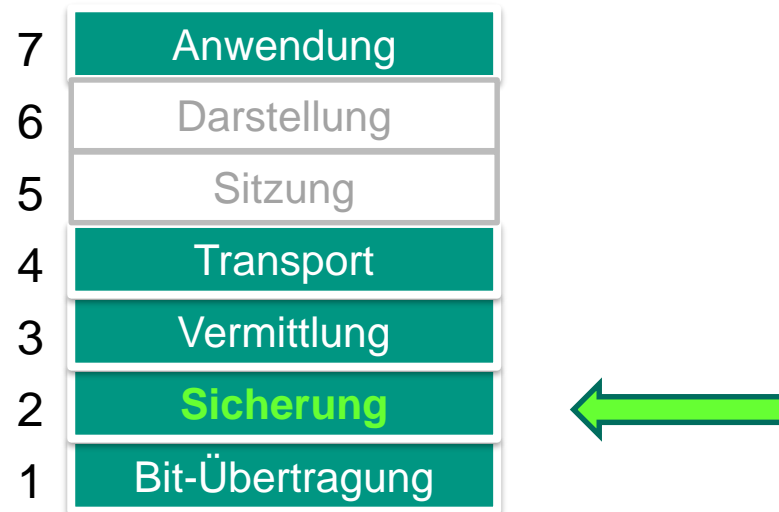
- Kopplung auf Schicht 1
 - Auffrischung des digitalen Signals
 - Medien können unterschiedlich sein (z.B. von Glasfaser auf Kupfer)
 - Größere (nicht beliebige) physikalische Ausdehnung des Netzes möglich

- Voraussetzung: Protokoll auf Schicht 2 muss identisch sein
 - Keine Zwischenspeicherung
 - Keine Bearbeitung von Dateneinheiten
 - Keine Beeinflussung des Verkehrs zwischen Teilnetzen, alle Dateneinheiten werden weitergeleitet

- Einsatzgebiete, z.B.
 - Ethernet
 - Z.B. Reichweite von max. 500 m pro Segment
 - Ausdehnung durch Einbau von max. 4 Repeatern auf bis zu 2500 m
 - Glasfaser (Überland, Unterwasser)

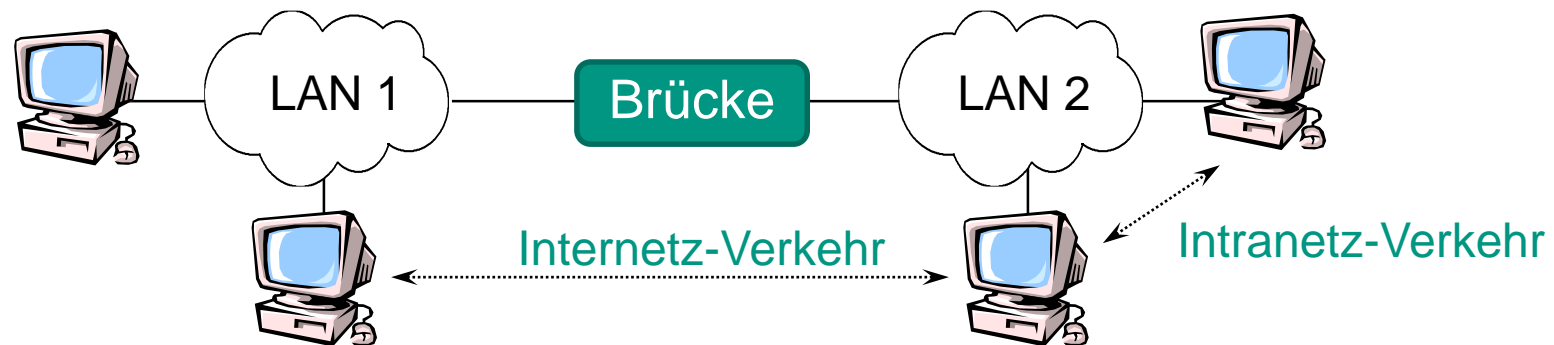
Einordnung

- Wir befinden uns auf Schicht 2 des OSI-Referenzmodells



7.3.2 Brücke

- Ziel: Kopplung von lokalen Netzen (LANs) auf Schicht 2
 - **homogen**: Netzwerke vom gleichen Typ (z.B. IEEE 802.x mit 802.x)
 - **inhomogen**: Netzwerke unterschiedlichen Typs (z.B. IEEE 802.x mit 802.y, $x \neq y$)
- Funktion: in beiden Netzen aktiv
 - Nimmt Dateneinheiten in LAN 1 an und versendet Dateneinheiten in LAN 2 erneut, wenn der Empfänger nicht in LAN 1 liegt
 - Verhält sich wie andere Sender (berücksichtigt ggf. belegtes Medium etc.)



Brücke: Eigenschaften

- Trennen des Intranetz-Verkehrs in einem LAN von dem Internetz-Verkehr zu anderen LANs (Filterfunktion)
- Erhöhung der Netzkapazität großer Netze durch **Partitionierung**
- Keine Filterung von Broadcast-Verkehr

Selbstlernende Switches/Brücken

■ Ziel

- Selbstorganisierende Konfiguration eines Netzes mit Brücken/Switches
 - ... ohne Eingriffe eines Systemadministrators

■ Aufgaben

- Etablierung einer Netztopologie ohne Schleifen
 - Spanning Tree Algorithmus
- Etablierung von Wegen zwischen Endsystemen
 - selbstlernende Brücken

Spanning Tree Algorithmus

■ Ziel

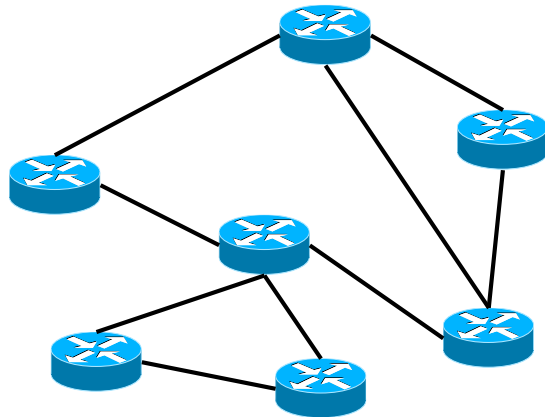
- Aufbau eines minimalen Spannbaums

... hat keine Schleifen

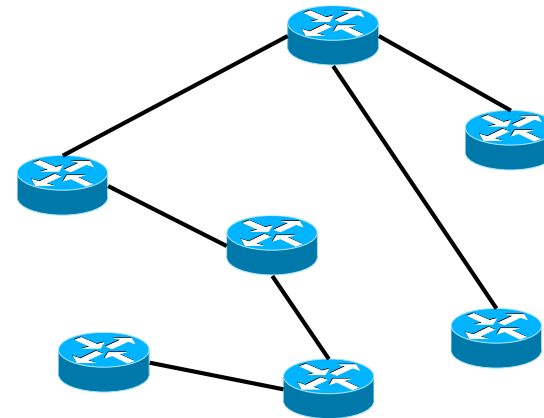
... Details in



■ Beispiel



Graph mit Brücken



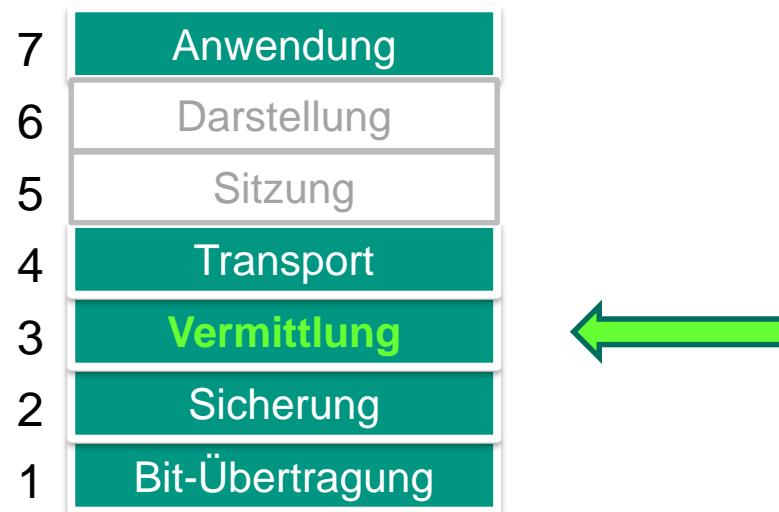
Ein minimaler Spannbaum

- Ziel
 - „Wege“ etablieren

- Vorgehensweise
 - *Kein* extra Protokoll hierfür
 - Brücke empfängt Dateneinheit und kennt Ziel-Adresse nicht
 - Flutet Dateneinheit auf allen aktiven Interfaces
 - Lernt „Lokation“ des Endsystems mit dieser Ziel-Adresse
 - Merkt sich, dass Endsystem über dieses Interface erreichbar ist
 - Brücke kennt Ziel-Adresse
 - Leitet Dateneinheit über entsprechendes Interface weiter

Einordnung

- Wir befinden uns auf Schicht 3 des OSI-Referenzmodells



7.3.3 Router

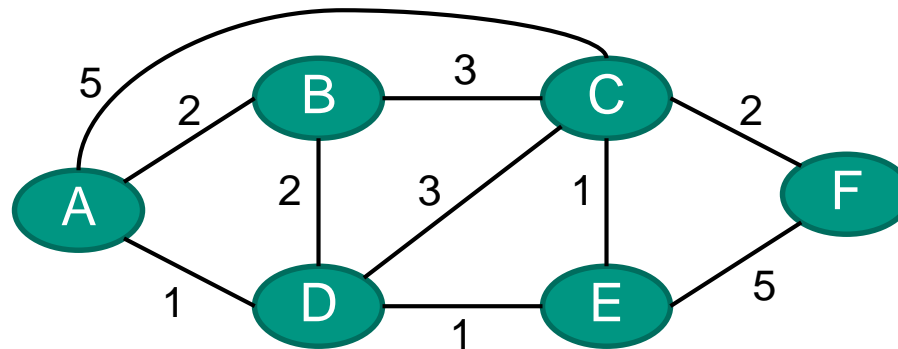
- Ziel
 - Kopplung von (Teil-)Netzen auf Schicht 3

- Eigenschaften
 - Schicht 2-Protokolle der Teilnetze können unterschiedlich sein
 - Teilnetz-übergreifendes Adressierungsschema in Schicht 3
 - Abbildung von Schicht 2 auf Schicht 3-Adressen nötig
 - In der Regel hierarchische Adressierung
 - Filterung von Verkehr möglich, kein Teilnetz-übergreifender Broadcast-Verkehr

- Ziel: Finden eines „guten“ Wegs
 - Typischerweise ist dies der Weg mit den geringsten Kosten
 - Andere Metriken sind denkbar
 - Hier: Betrachtung paketvermittelnder Netze
- Aufgabe der **Routing-Protokolle**

■ Modellierung des Kommunikationsnetzes als Graph

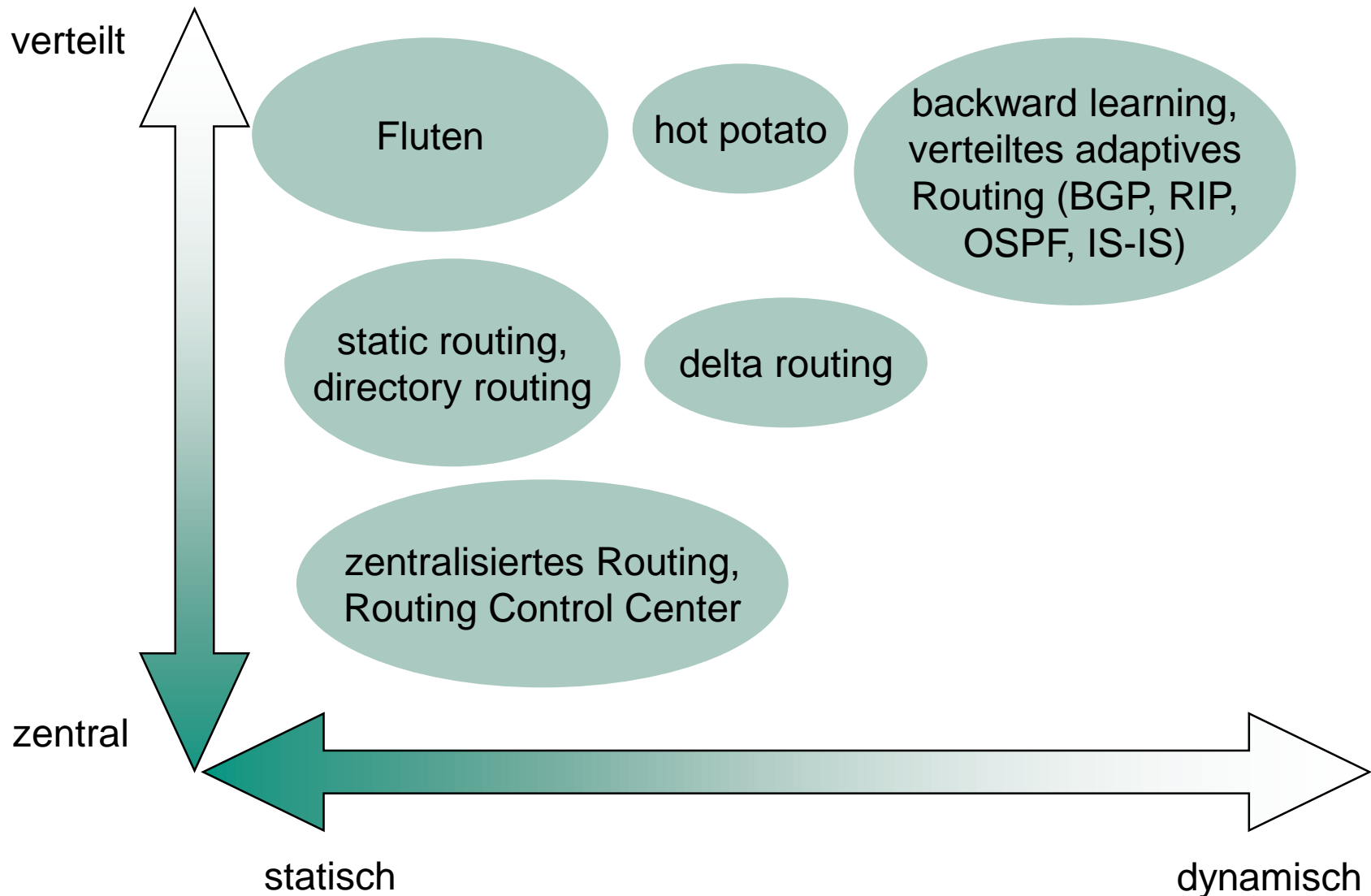
- Router (Zwischensysteme) sind **Knoten**
- Übertragungsabschnitte sind **Kanten**
 - „Kosten“ der Kanten bspw. Verzögerung, Stausituation, Preis ...



■ Pfad

- Folge von Knoten (n_1, n_2, \dots, n_k) ,
wobei $(n_1, n_2), (n_2, n_3), \dots, (n_{k-1}, n_k)$
Kanten des Graphs sind und kein Knoten mehrfach vorkommt
- Pfad mit geringsten Kosten vs. kürzester Pfad

Routing-Verfahren im Überblick



Routing-Verfahren: Zentralisierung

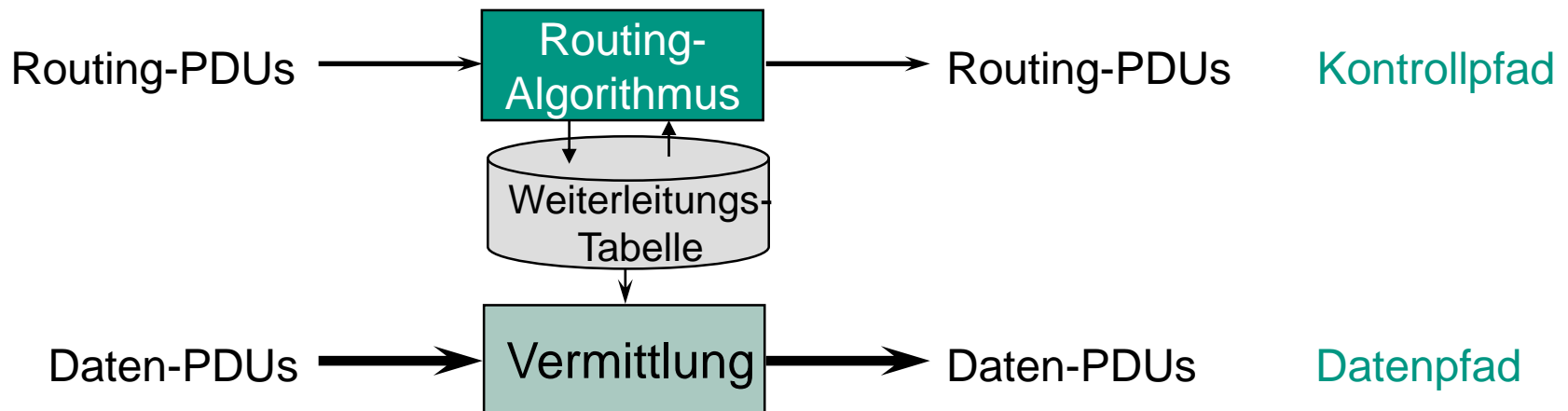
- Wo ist der Routing-Algorithmus lokalisiert?
 - **Zentral** (in einem Netzkontrollzentrum)
 - Zentrale hat Wissen über das komplette Netz
 - Auch als globales Routing bezeichnet
 - Berechnung optimaler Wege möglich

 - **Dezentral** (verteilt auf die Zwischensysteme)
 - Systeme kennen initial nur ihre Nachbarn
 - **Distanz-Vektor-Algorithmen**
 - Kein System hat Wissen über das komplette Netz
 - System kennt nie die komplette Route von einer Quelle zu einer Senke
 - **Link-State-Algorithmen**
 - System hat Wissen über die gesamte Netztopologie
 - Alle Systeme haben (im stabilen Zustand) die gleiche Sicht auf das Netz

- Wie dynamisch ist das Routing-Verfahren?
 - Nicht adaptiv
 - Routen ändern sich nur sehr selten
 - Routenänderungen sind viel seltener als Verkehrsänderungen
 - Adaptiv
 - Routen ändern sich in Abhängigkeit des Verkehrs bzw. der Netztopologie
 - Aktueller Zustand des Netzes wird damit berücksichtigt
 - Schleifen und Oszillationen in Routen wahrscheinlicher als bei nicht- adaptiven Verfahren
 - Können periodisch operieren oder in direkter Reaktion auf Änderungen
 - Zielkonflikt
 - Systeme haben veraltete oder unvollständige Informationen über den Zustand des Netzes
 - Evtl. hohe Belastung durch Austausch von Routing-Informationen

Router – Kontroll- vs. Datenpfad

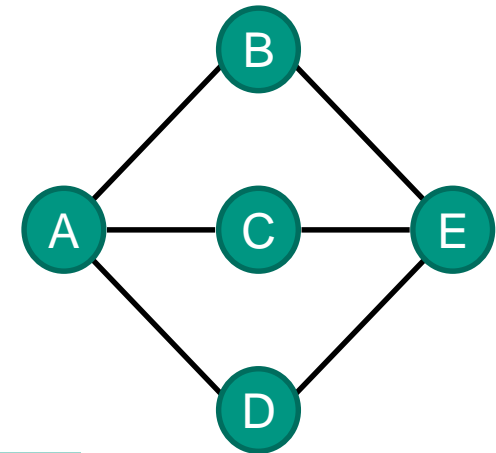
- Datenpfad
 - Vermittlung der Daten auf Schicht 3 (Vermittlungsschicht)
- Kontrollpfad
 - Steuert Vermittlung der Daten
 - Weiterleitungs-Tabelle
 - Enthält Einträge für mögliche Ziele mit Schnittstellen, auf denen Dateneinheiten zum nächsten Zwischensystem in Richtung Ziel weitergeleitet werden
 - Routing-Protokolle sind oberhalb der Schicht 3 angesiedelt
 - Vermittlung/Weiterleitung der Dateneinheiten anhand der Information in der Weiterleitungs-Tabelle



Statisches Routing – Beispiel

■ Beispiel

- Ziehen einer Zufallszahl x mit $1 > x \geq 0$
- Falls $x < 0,6$ dann Weiterleiten nach B
- Falls $0,9 \geq x \geq 0,6$ dann Weiterleiten nach C
- Sonst Weiterleiten nach D

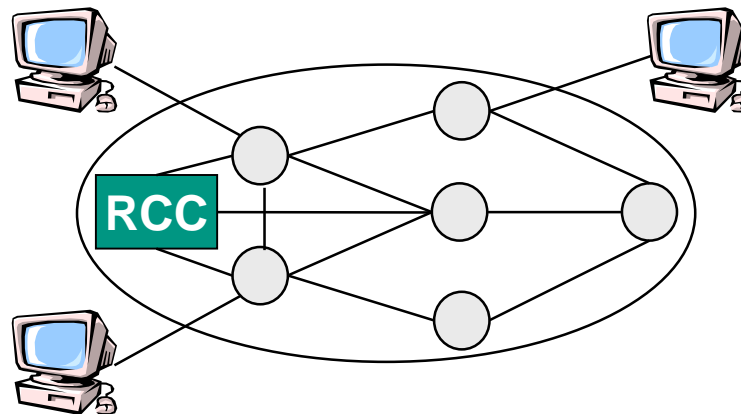


■ Tabelle in System A

Ziel	1. Wahl		2. Wahl		3. Wahl	
	System	Gewicht	System	Gewicht	System	Gewicht
E	B	0,6	C	0,3	D	0,1
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Zentralisiertes Routing

- Adaptives Verfahren
- Zentrales **Routing Control Center (RCC)**
 - Jedes System sendet periodisch Zustandsinformationen an RCC
 - Z.B. Liste aller aktiven Nachbarn
 - Aktuelle Warteschlangenlängen
 - Umfang an Verkehr, der seit dem letzten Bericht abgewickelt wurde
 - RCC berechnet mit diesen Informationen die optimalen Wege zwischen allen Systemen (z.B. kürzeste Wege)
 - RCC verteilt neue Routing-Information an Router
 - Jeder Router trifft Routing-Entscheidungen anhand dieser Information



Zentralisiertes Routing – Vor- und Nachteile

■ Vorteile

- RCC hat theoretisch die vollständige Übersicht und kann perfekte Entscheidungen treffen
- Systeme müssen keine aufwendigen Routing-Berechnungen durchführen

■ Nachteile

- Für große Netze dauert die Berechnung u.U. sehr lange
- Ausfall des RCC lähmt das ganze Netz
 - Erfordert daher möglichst robuste Auslegung
 - z.B. durch Redundanz, etwa Backup-Rechner, etc.
- Inkonsistenzen möglich, da Systeme nahe dem RCC neue Routing-Tabellen wesentlich früher erhalten als die weiter entfernten
- Starke Belastung des RCC durch die zentrale Funktion

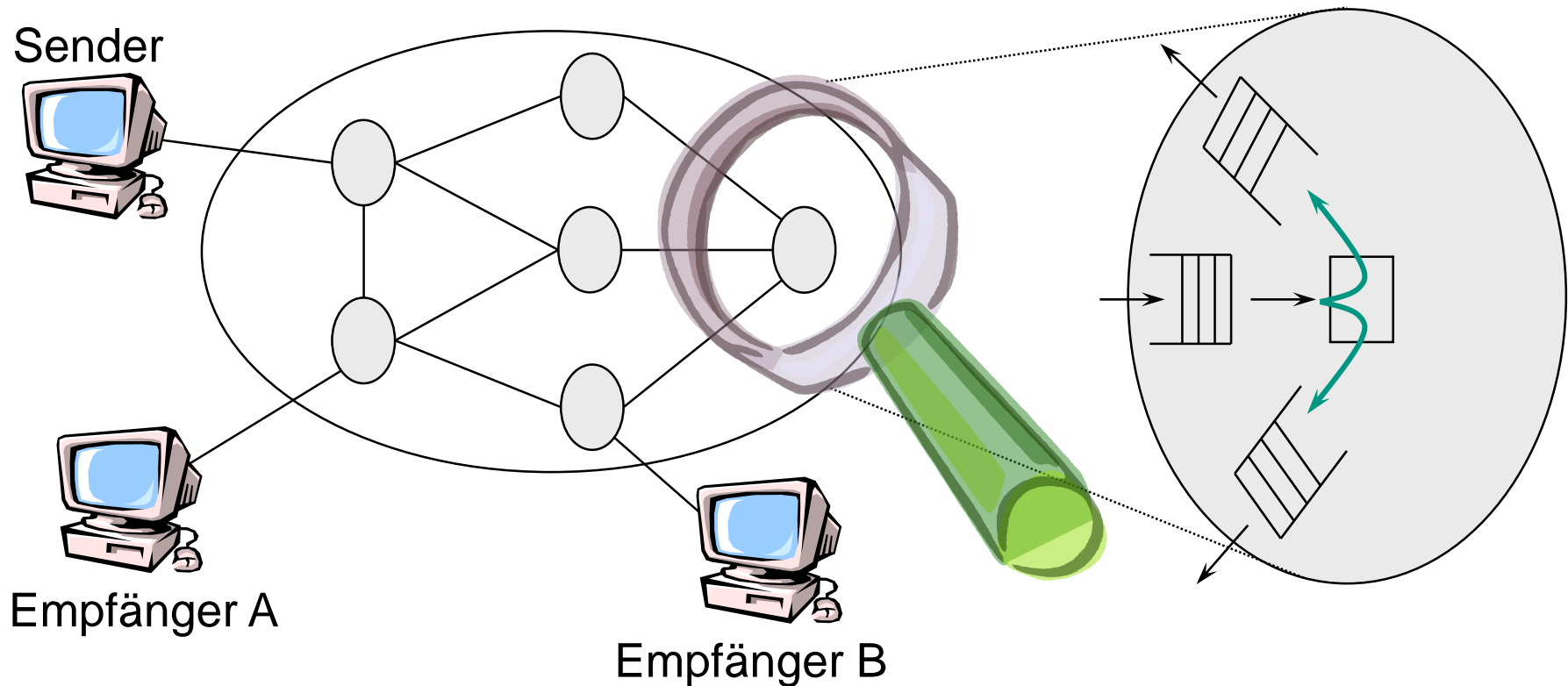
Isoliertes Routing: Überblick

- Jedes System entscheidet nur aufgrund der Information, die es selbst sammelt
- Kein Austausch von Routing-Informationen zwischen den Systemen
- Anpassung an Verkehrs- und Topologieänderungen kann somit nur mit Hilfe beschränkter Informationen erfolgen

- Unterschiedliche Verfahren, z.B.
 - Fluten
 - Hot Potato

Fluten

- Einfachstes Verfahren, nicht adaptiv
- Jede eingehende Dateneinheit wird auf jeder Übertragungsleitung weiter übertragen, außer auf derjenigen, auf der es eintraf

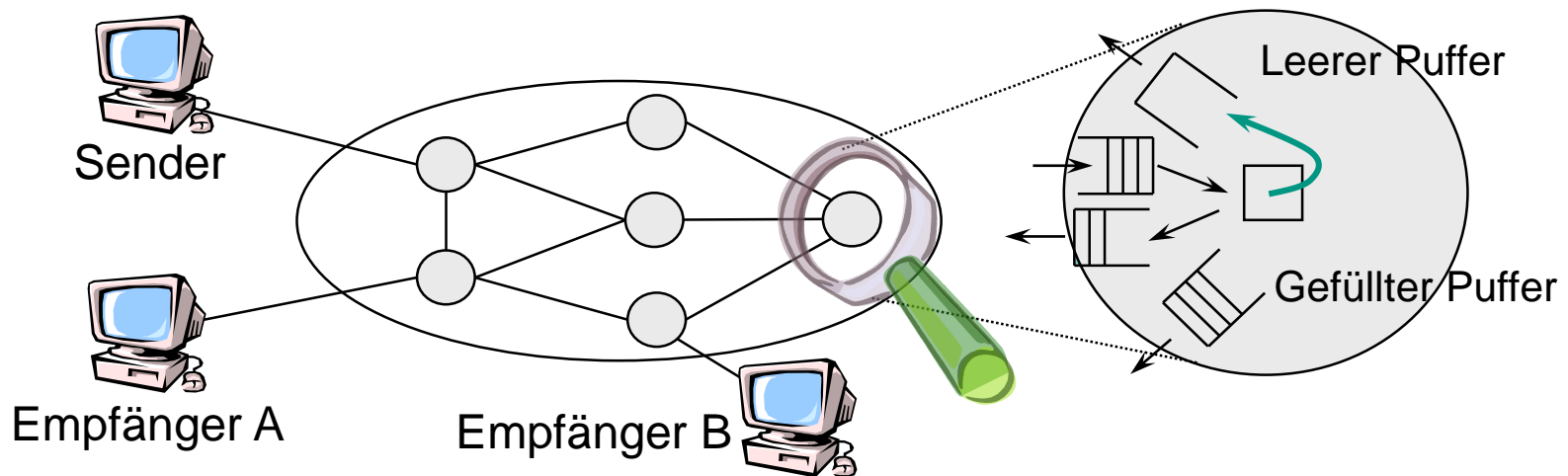


- Maßnahmen zur Eindämmung der Flut
 - Erkennung von Duplikaten durch Sequenznummern
 - Kontrolle der Lebensdauer einer Dateneinheit durch Zählen der zurückgelegten Übertragungsabschnitte (**Hops**)
 - Hop-Zähler wird mit der maximaler Weglänge initialisiert
 - In jedem Router wird der Zähler um 1 dekrementiert
 - Falls der Zähler den Wert 0 erreicht, kann die Dateneinheit verworfen werden
- Varianten
 - Selektives Fluten
 - Weiterleitung nicht auf allen, sondern nur auf einigen Übertragungsabschnitten
 - Random Walk
 - Zufällige Auswahl eines Übertragungsabschnittes
- Einsatzbeispiele
 - Mobile Ad-hoc-Netze (MANETs)
 - Jedes System ist Endsystem und gleichzeitig Router
 - Drahtlose Sensor-Aktor-Netze im Internet of Everything
 - Daten stehen im Mittelpunkt des Interesses



Hot Potato

- Jedes System versucht, eingehende Dateneinheiten so schnell wie möglich weiterzuleiten
 - Wählt Übertragungsabschnitt mit der kürzesten Warteschlange
- Varianten
 - Weiterleitung nie auf dem Übertragungsabschnitt, auf dem die Dateneinheit eintraf
 - Kombination mit statischem Routing
 - Auswahl der besten Übertragungsleitung nach statischem Verfahren, solange Warteschlangenlänge unter bestimmtem Schwellenwert bleibt
 - Auswahl der Übertragungsleitung mit kürzester Warteschlange, falls deren statisches Gewicht nicht zu niedrig ist



Verteiltes adaptives Routing

- Systeme tauschen Routing-Informationen mit Nachbarn aus
- Jedes System unterhält eine Routing-Tabelle. Enthält Informationen über zu erreichende Systeme, z.B.
 - Bevorzugter Übertragungsabschnitt zum Ziel
 - Schätzung über Zeit oder Entfernung zum Ziel, z.B.
 - Anzahl Hops
 - Geschätzte Verzögerung in Millisekunden
 - Geschätzte Anzahl von Dateneinheiten, die entlang des Weges warten
- Schätzungen werden gewonnen aus
 - Zeit oder Entfernung zu den Nachbarn
 - z.B. aus speziellen Echo-Dateneinheiten mit Zeitstempeln
 - Schätzungen der Nachbarn
- Varianten
 - Periodischer Austausch von Routing-Information
 - Austausch nur bei signifikanten Änderungen

■ Distanz-Vektor-Algorithmen

- Routing-Metrik: Distanz
- Jeder Router kennt Distanz zu allen anderen Systemen im Netz
- Hierzu werden die aktuellen Distanzen zwischen den Nachbarn ausgetauscht
- Problem
 - Kürzerer langsamerer Weg wird längerem schnelleren Weg vorgezogen
- Beispiele
 - Routing Information Protocol (RIP), Distance Vector Routing Protocol (DVRP)

■ Link-State-Algorithmen

- Unterschiedliche Routing-Metriken möglich
- Berücksichtigt die aktuellen Zustände der Netzanschlüsse
- Jeder Router kennt komplette Netztopologie und berechnet auf dieser Basis seine Routing-Information
- Link-State-Algorithmen konvergieren im Allg. schneller als Distanz-Vektor-Algorithmen
 - Für größere Netze sind sie damit potenziell besser geeignet
- Beispiele
 - Open Shortest Path First (OSPF), Intra-Domain Intermediate System to Intermediate System Routing Protocol (IS-IS)

7.3.3.1 Distanz-Vektor-Routing

■ Eigenschaften

■ Verteilt

- Jeder Router erhält Information von seinen direkten Nachbarn, führt eine Berechnung durch und verteilt dann neue Information an seine Nachbarn

■ Iterativ

- Das Verteilen und Berechnen von Information geht so lange vor sich bis keine Information mehr ausgetauscht wird

■ Distanz-Vektor-Tabelle

■ Die grundlegende Datenstruktur für Distanz-Vektor-Algorithmen

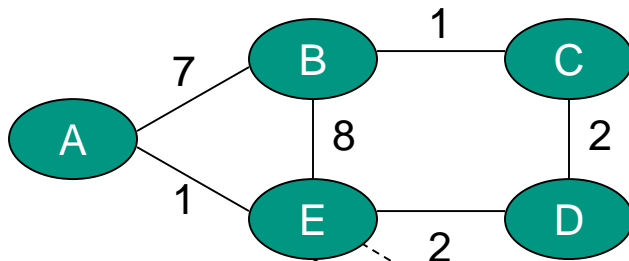
- In jedem System vorhanden
- Zeile für jedes mögliche Ziel
- Spalte für jeden direkten Nachbarn

■ Weiterleiten von Daten

- X will Daten über seinen direkten Nachbarn Z an Y weiterleiten
- $D^X(Y, Z) = c(X, Z) + \min_w \{D^Z(Y, w)\}$

Beispiel: Distanz-Vektor-Tabelle

- Beispiel: $D^E(A, D)$
 - Erster Übertragungsabschnitt ist der von E nach D
 - Eintrag in der Tabelle umfasst Kosten von E nach D (2) plus minimale Kosten von D nach A (3)
 - Minimale Kosten von D nach A über Nachbarsystem von D
 - Wie kommt der Wert von 14 für $D^E(A, B)$ zustande?



Next Hop \ Ziel	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

■ Initialisierung

- Für alle Nachbarn v : $D^X(*, v) = \infty$, $D^X(v, v) = c(X, v)$
- Für alle Ziele y : sende $\min_w D^w(y, w)$ zu jedem Nachbarn, wobei w alle Nachbarn enthält

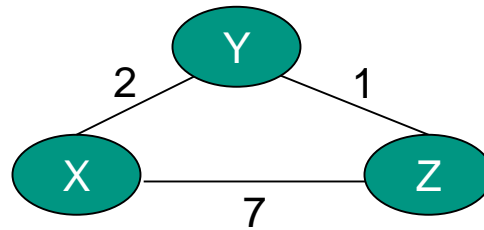
■ Schleife

- Geänderte Übertragungsabschnittskosten:
 $c(X, V)$ ändert sich um den Wert d (positiv oder negativ)
 - Für alle Ziele y : $D^X(y, V) := D^X(y, V) + d$
- Update-Nachricht von einem Nachbarn
 - Kürzester Pfad von V zu einem Ziel Y hat sich geändert zu „neuer Wert“
 - $D^X(Y, V) = c(X, V) +$ „neuer Wert“ für dieses Ziel
- Falls ein neuer $\min_w D^w(Y, w)$ für ein Ziel Y existiert, dann sende diesen Wert zu allen Nachbarn

■ Komplexität: $O(n^3)$ mit $n =$ Anzahl der Knoten

■ Betrachteter Algorithmus: **Bellman-Ford-Algorithmus**

Beispiel



Annahme:
Systeme arbeiten synchron

t_1

t_2

t_3

System
X

D^X	Y	Z
Y	2	∞
Z	∞	7

D^X	Y	Z
Y	2	8
Z	3	7

D^X	Y	Z
Y		
Z		

System
Y

D^Y	X	Z
X	2	∞
Z	∞	1

D^Y	X	Z
X	2	8
Z	9	1

D^Y	X	Z
X		
Z		

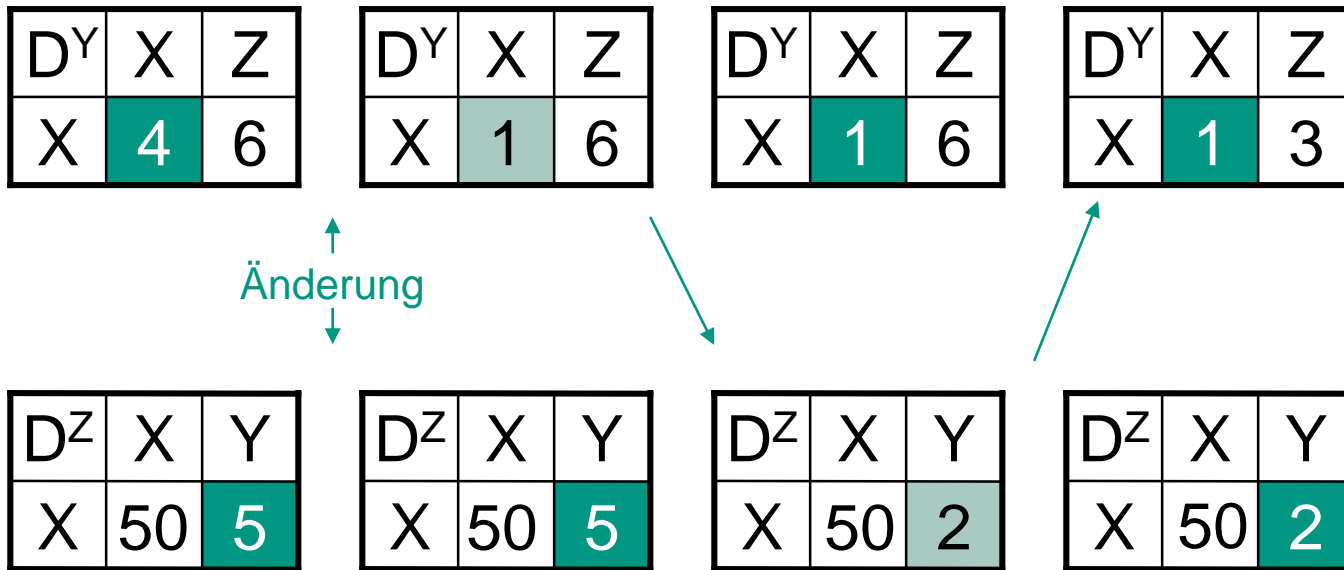
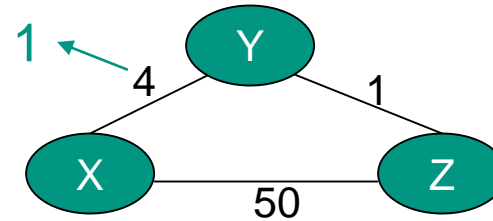
System
Z

D^Z	X	Y
X	7	∞
Y	∞	1

D^Z	X	Y
X	7	3
Y	9	1

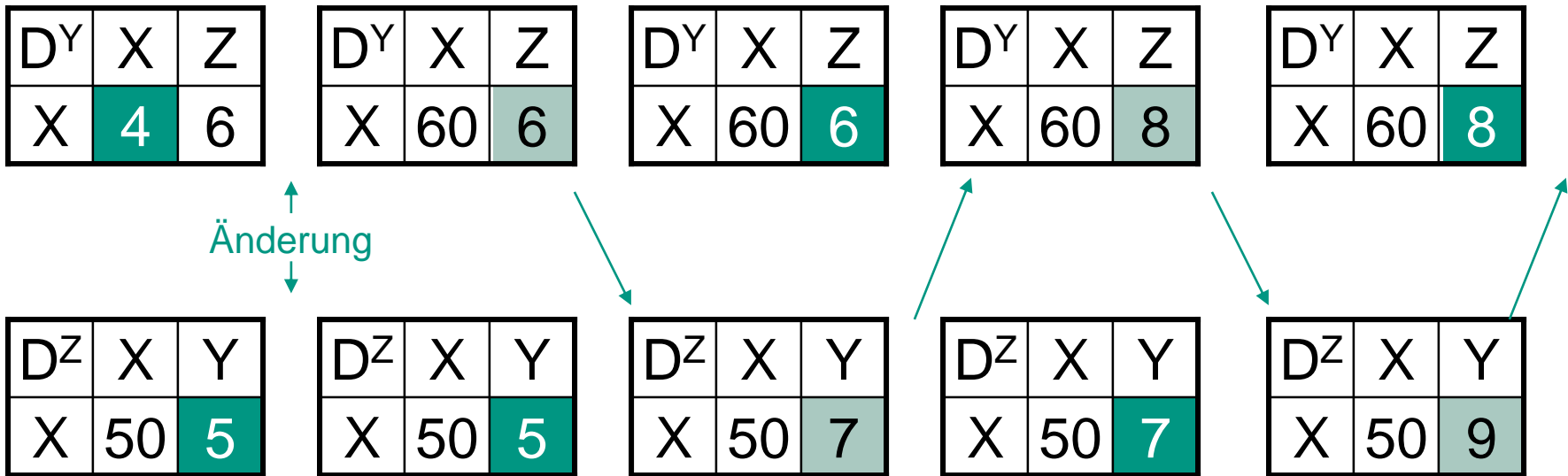
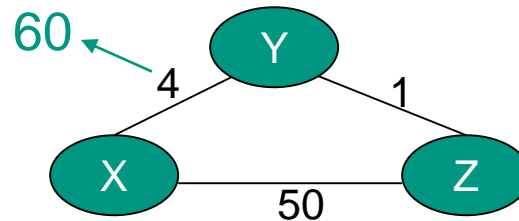
D^Z	X	Y
X		
Y		

Änderung der Linkkosten: Good News



→ Die gute Neuigkeit hat sich schnell im Netz ausgebreitet

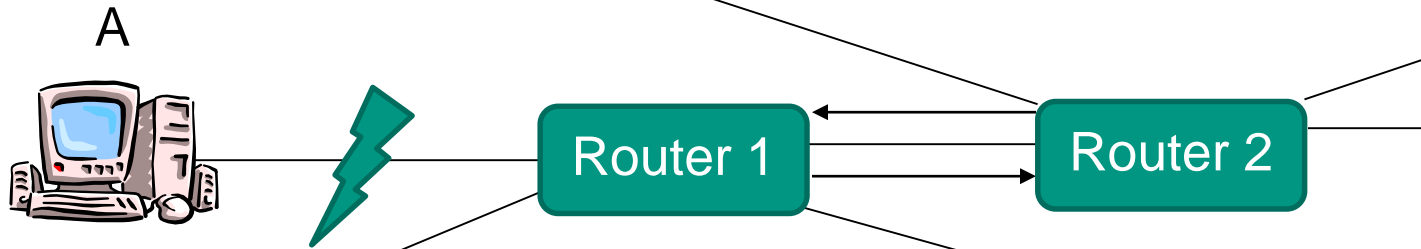
Änderung der Linkkosten: Bad News



- Schlechte Neuigkeit breitet sich relativ langsam aus und führt u.U. zu Routing-Schleifen
 - Schleife hier benötigt 44 Iterationen!
- **Count-to-Infinity**

Count-to-Infinity-Problem

Ziel	Distanz	Nächstes System	Schnittstelle
A	2	Router 1	1
...

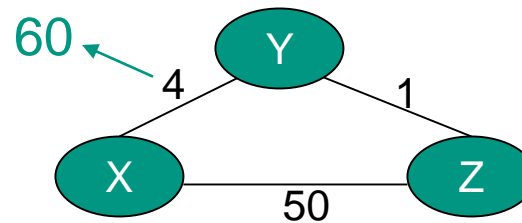


Ziel	Distanz	Nächstes System	Schnittstelle
A	3	Router 2	1
...

■ Zur Vermeidung: **Poisonous Reverse**

Poisonous Reverse

- Ziel
 - Vermeidung von den zuvor beschriebenen Routing-Schleifen
- Vorgehensweise
 - Routing-Information wird Y „vorenthalten“, wenn Weg über Y kürzer



D^Y	X	Z
X	4	∞

D^Y	X	Z
X	60	∞

D^Y	X	Z
X	60	∞

D^Y	X	Z
X	60	51

D^Y	X	Z
X	60	51

D^Z	X	Y
X	50	5

D^Z	X	Y
X	50	5

D^Z	X	Y
X	50	61

D^Z	X	Y
X	50	61

D^Z	X	Y
X	50	∞

Weg über Z kürzer:
 $D(X, Y) = \infty$

7.3.3.2 Link-State-Routing

- Grundlegende Vorgehensweise
 - Systeme müssen am Anfang nur ihre direkten Nachbarn kennen
 - Entdecken neuer Nachbarn mittels spezieller Dateneinheiten
 - z.B. HELLO
 - Bestimmen der Kosten zu den direkten Nachbarn
 - **Link State Broadcast**
 - Identität und Kosten zu den direkten Nachbarn werden an alle Router im Netz weitergeleitet (Fluten)
 - Systeme können Topologie lernen durch die *Link State Broadcasts* der anderen Systeme
 - Ergebnis: Alle Systeme haben *identisches* Wissen über das Netz

- Berechnung der kürzesten Pfade durch Link-State-Algorithmus
 - Jedes System berechnet die kürzesten Pfade
 - Die berechneten Pfade sind aufgrund der identischen Information gleich
 - Nach Fluten und Berechnung der kürzesten Pfade in jedem System ist das Netz schleifenfrei und in stabilen Zustand konvergiert

- Im folgenden betrachteter Algorithmus **Dijkstra-Algorithmus**
 - Berechnet Pfad mit den geringsten Kosten von einem System zu allen anderen Systemen im Netz

- Link-State Routing wird eingesetzt in den Protokollen
 - OSPF (*Open Shortest Path First*) und
 - IS-IS (*Intermediate System to Intermediate System*)

■ Notation

- $c(i, j)$: Kosten von System i zu System j
 - Annahme: $c(i, j) = c(j, i)$
 - Falls i und j nicht direkt verbunden, gilt initial: $c(i, j) = \infty$
- $D(v)$
 - Kosten der Route von der Quelle zur Senke v , die momentan die geringsten Kosten besitzt
- $p(v)$
 - Vorgänger von v auf dem momentan kürzesten Pfad zu v
- N
 - Menge der Systeme, deren kürzester Pfad von der Quelle bekannt ist

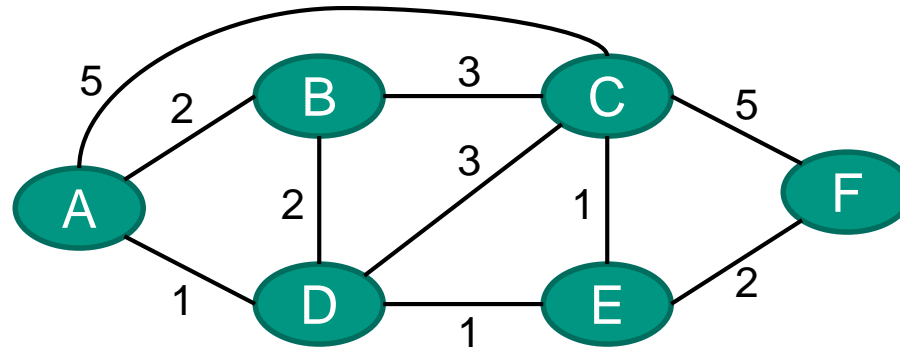
Dijkstra Algorithmus

- Initialisierungsphase
 - $N = \{\text{Quelle } A\}, D(v) = c(A, v)$ für alle direkten Nachbarn von A ,
 $D(v) = \infty$ sonst

- Schleife (wird entsprechend der Anzahl von Systemen im Netz durchlaufen)
 - Finde ein System w mit $w \notin N$ und $D(w)$ ist ein Minimum
 - Füge w zu N hinzu
 - Erneuere $D(v)$ für alle $v \notin N$ und v ist direkter Nachbar von w
 - $D(v) = \min(D(v), D(w) + c(w, v))$

- Komplexität: $O(n^2)$ mit $n = \text{Anzahl der Knoten}$
 - Je nach Implementierung auch $O(n \log n + m)$ mit $m = \text{Anzahl der Kanten}$

Link-State-Routing: Beispiel



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

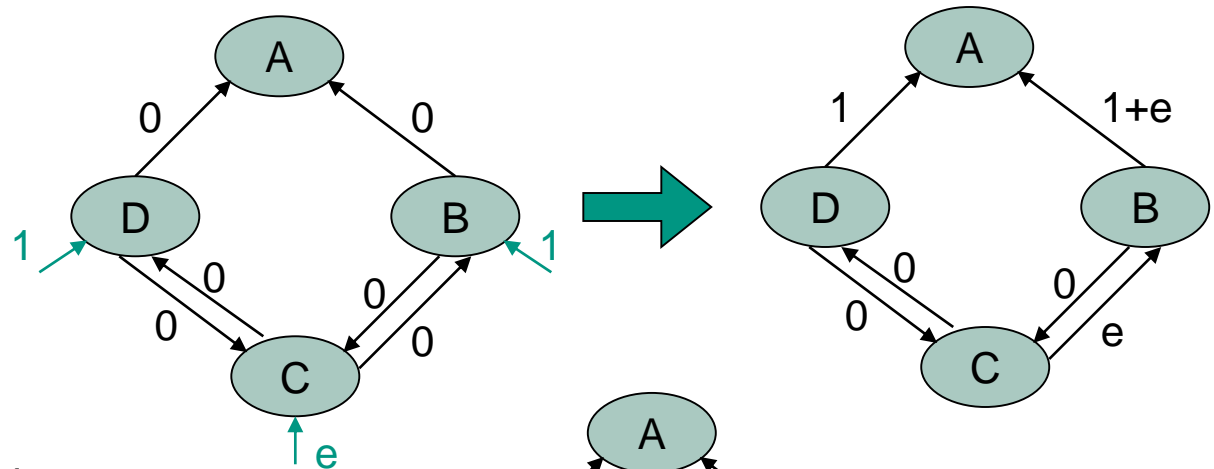
- Initialisierung
 - Die momentan bekannten kürzesten Pfade zu den Nachbarn von A werden gesetzt entsprechend der jeweiligen Link-Kosten (2, 5, 1)
 - Die Pfade zu nicht direkt benachbarten Systemen werden auf ∞ gesetzt
- Erste Iteration
 - System mit den geringsten Kosten wird zu N hinzugenommen: D
 - $D(v)$ wird für alle Systeme erneuert
 - Kosten zu C und E senken sich
- Zweite Iteration
 - System mit den geringsten Kosten wird zu $N = \{A, D\}$ hinzugenommen: E
 - B wäre ebenfalls möglich gewesen
 - $D(v)$ wird für alle Systeme erneuert
 - Kosten zu C senken sich und F ist erstmals mit Kosten geringer als ∞ erreichbar
- ...

■ Beispielnetz

- Linkkosten sind äquivalent zur Last auf dem Weiterleitungsabschnitt
- Die Linkkosten sind hier nicht symmetrisch
- Die Quellen B, C, und D senden 1, e bzw. 1 „Verkehrseinheiten“

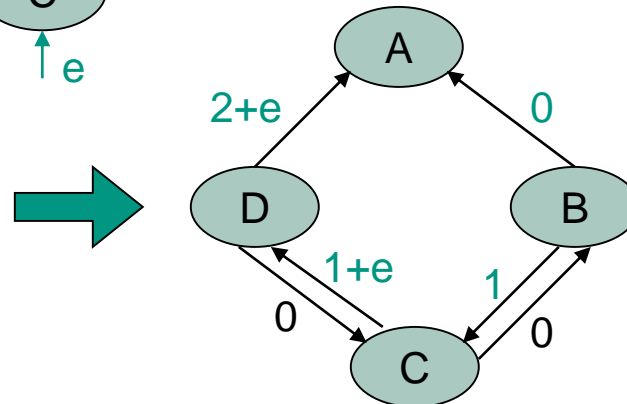
■ Initiales Routing

- $D \rightarrow A$,
- $B \rightarrow A$,
- $C \rightarrow B \rightarrow A$



■ Nächste Runde

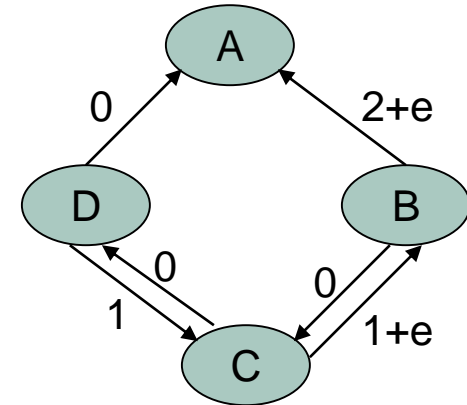
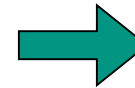
- $D \rightarrow A$,
- $B \rightarrow C \rightarrow D \rightarrow A$,
- $C \rightarrow D \rightarrow A$



Oszillation

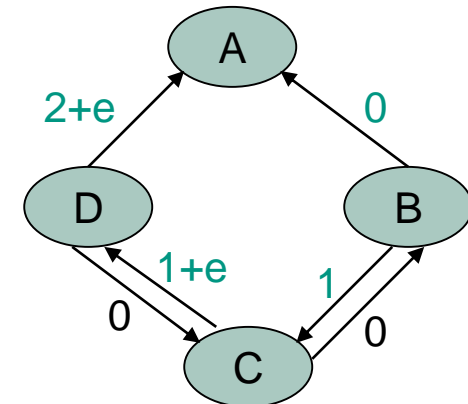
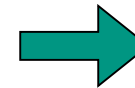
■ Nächste Runde

- Systeme B, C und D entdecken Pfad mit Kosten 0 zu A
- $D \rightarrow C \rightarrow B \rightarrow A$,
 $B \rightarrow A$,
 $C \rightarrow B \rightarrow A$



■ ... Und die nächste Runde

- $D \rightarrow A$,
- $B \rightarrow C \rightarrow D \rightarrow A$,
- $C \rightarrow D \rightarrow A$



■ Was tun?

- Router sollten nicht alle zum gleichen Zeitpunkt den Algorithmus berechnen
- „Self-Synchronization“ sollte möglichst vermieden werden
 - Einführung von Zufälligkeiten („Randomization“)

Link-State vs. Distanz-Vektor

- Komplexität der Kontroll-Dateneinheiten
 - Jedes System muss bei Link-State die Kosten aller Links kennen:
Bei n Systemen und E Links sind $O(nE)$ Dateneinheiten erforderlich
 - Änderungen müssen bei Link-State an alle Systeme gesendet werden
 - Bei Distanz-Vektor werden Änderungen benachbarten Systeme weitergegeben
- Konvergenzgeschwindigkeit
 - Link-State hat eine Komplexität von $O(n^2)$ und benötigt $O(nE)$ Dateneinheiten
 - Schnelle Konvergenz, danach schleifenfrei
 - Oszillationen sind möglich
 - Distanz-Vektor-Algorithmen können langsam konvergieren und können Routing-Schleifen aufweisen
 - Das Count-to-Infinity-Problem kann auftreten
- Robustheit
 - Routenberechnungen sind bei Link-State separiert und stellen somit eine gewisse Robustheit bereit
 - Bei Distanz-Vektor-Algorithmen kann ein System inkorrekte Pfade zu allen Zielen verbreiten
- Gewinner?
 - Link-State konvergiert schneller und ist robuster
 - Distanz-Vektor einfacher zu implementieren

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
- 7. Netzkopplung und Vermittlung**
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Motivation
2. Vermittlungstechniken
 1. Leitungsvermittlung
 2. Paketvermittlung
 3. Datagrammvermittlung
 4. Virtuelle Verbindungen
 5. Nachrichtenvermittlung
3. Netzkopplung
 1. Repeater
 2. Brücke
 3. Router
4. Vermittlung im Internet

7.5 Vermittlung im Internet

- Problem
 - Wie werden Daten im Internet weitergeleitet?

- Verfahren
 - Weiterleitungs-Tabelle liefert Information über nächsten Hop
 - IP-Protokoll
 - Verbindungslos
 - Segmentiert und reassembliert
 - In allen Systemen?
 - Benutzt Internet-Adressierung
 - Unterschied zur MAC-Adressierung?
 - Benutzt weitere Protokolle wie
 - ICMP (Internet Control Message Protocol)
 - ARP (Address Resolution Protocol)
 - IGMP (Internet Group Management Protocol)

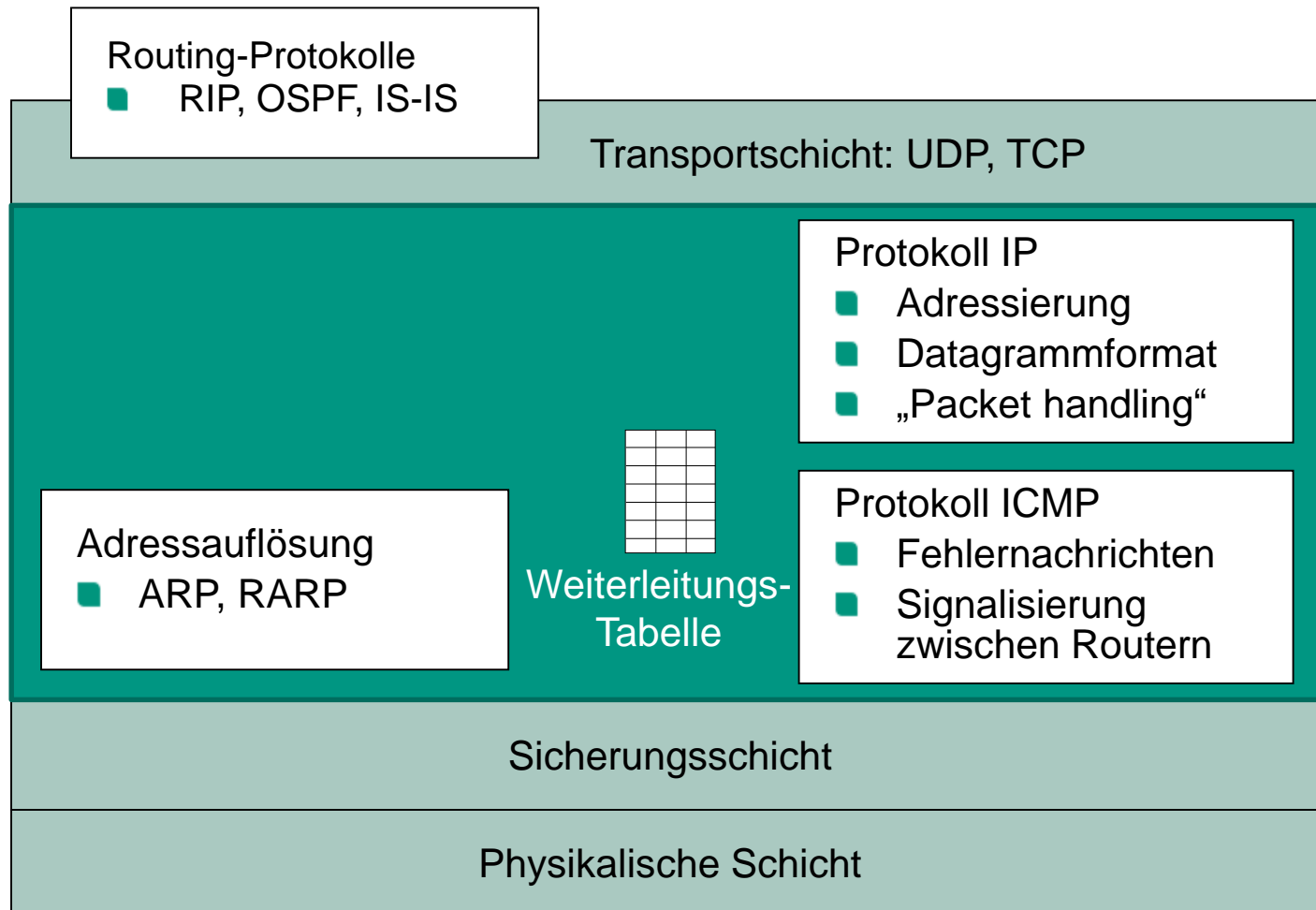
■ Pingo-Link für diese Vorlesung:

→ <http://pingo.upb.de/6466>



<http://pingo.upb.de/>





Aufgaben der Protokolle


- **TCP** (Transmission Control Protocol)
 - Stellt zuverlässigen Transportdienst bereit
- **UDP** (User Datagram Protocol)
 - Stellt unzuverlässigen Transportdienst bereit
- **IP** (Internet Protocol)
 - Unzuverlässige Übertragung/Weiterleitung von Datagrammen
- **ICMP** (Internet Control Message Protocol)
 - Austausch von Kontrollinformationen innerhalb der Vermittlungsschicht
- **IGMP** (Internet Group Management Protocol)
 - Verwaltung von Kommunikationsgruppen
- **ARP** (Address Resolution Protocol)
 - Zuordnung von IP-Adressen zu Adressen der Sicherungsschicht
- **RARP** (Reverse Address Resolution Protocol)
 - Stellt Umkehrfunktion von ARP zur Verfügung

- **Routingprotokolle**
 - BGP (Border Gateway Protocol), **RIP** (Routing Information Protocol), OSPF (Open Shortest Path First)

■ Ziel

- Eindeutige Identifizierung aller im Internet angeschlossenen Systeme bzw. deren einzelner Schnittstellen
 - Ein System kann mehrere Interfaces haben (z.B. Ethernet, WLAN, UMTS)

■ IP-Adressen

- Weltweit eindeutige Adressen auf Schicht 3
- Einfaches, für Maschinen leicht zu verarbeitendes Format
- IPv4
 - Adressen einer Länge von 32 Bit
- IPv6 
 - Größerer Adressraum durch Adressen von 128 Bit Länge

Adressierung bei IPv4

- Ursprünglich unterstützte IP fünf verschiedene Adressklassen



[RFC950]

- Class A für Netze mit mehr als 65.536 Systeme

0	1	2	4	8	16	24	
0	Netz-ID				System-ID		
 - Class B für Netze zwischen 256 und 65.536 Systeme

0	1	2	4	8	16	24	
1	0	Netz-ID				System-ID	
 - Class C für Netze mit weniger als 256 Systeme

0	1	2	4	8	16	24	
1	1	0	Netz-ID				System-ID
 - Class D für Gruppenkommunikation (Multicast)

0	1	2	4	8	16	24
1	1	1	0	Multicast-Adresse		
 - Class E, reserviert für zukünftige Anwendungen

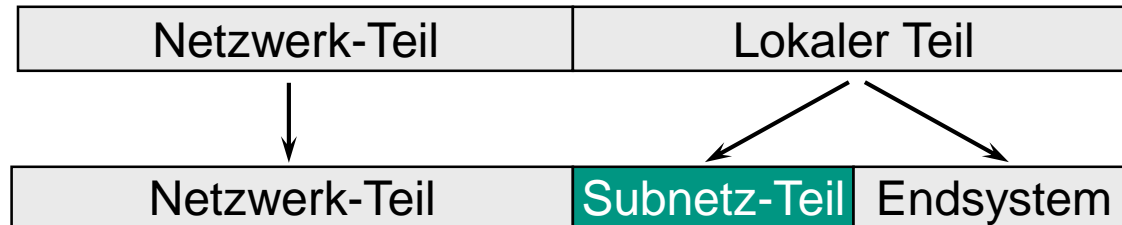
0	1	2	4	8	16	24
1	1	1	1	0	Reserviert	
- Diese Form der Adressierung wird heute nicht mehr direkt eingesetzt
 - Stattdessen **Classless Interdomain Routing (CIDR)**

- Bisher: Drei Adressklassen für Unicast, somit schlechte Ausnutzung durch ungenutzte Adressen („Verschnitt“), z.B.:
 - Größeres Netz mit mehr als 254 Komponenten benötigt Class-B-Adresse
 - Kleines Netz mit 100 IP-Adressen, benötigt Class-C-Adresse
 - 254 Adressen wären verfügbar, damit 154 ungenutzte Adressen

- CIDR
 - Ersetzen der festen Klassen durch Präfixe variabler Länge
 - Beispiele
 - 129.24.12.0/14: Die ersten 14 Bits der IP-Adresse werden für die Netz-Identifikation verwendet
 - 141.3.64.0/21 = 141.3.64.0 bis 141.3.71.255
 - Einsatz in Verbindung mit hierarchischem Routing
 - Backbone-Router, z.B. an Transatlantik-Link, betrachtet z.B. nur die ersten 13 Bits; dadurch kleine Routing-Tabellen, wenig Rechenaufwand
 - Router eines angeschlossenen Providers z.B. die ersten 15 Bit
 - Router in einem Firmennetz mit 126 Hosts betrachtet 25 Bits

Subnetz-Adressen bei IPv4

- Weitere Strukturierung von IP-Adressen



- Subnetzmasken kennzeichnen den Bereich der IP-Adresse, der das Netzwerk und das Subnetzwerk beschreibt. Dieser Bereich wird dabei durch Einsen („1“) in der binären Form der Subnetzmaske festgestellt

- Beispiel

IP-Adresse:	129.	13.	3.	64
Subnetzmaske:	255.	255.	255.	0
=	1111 1111	1111 1111	1111 1111	0000 0000
Netzwerk:	129.	13.		(Class B)
Subnetz:			3.	
Endsystem:				64

- Überdeckt die Subnetzmaske nur den Netzwerk-Teil, dann gibt es keinen Subnetz-Teil (z.B. Subnetzmaske 255.255.0.0 bei Class B)
- Achtung: Systeme, die an mehrere Netze angeschlossen sind (z.B. Router), haben mehrere, netzspezifische IP-Adressen

Zuteilung von Adressen

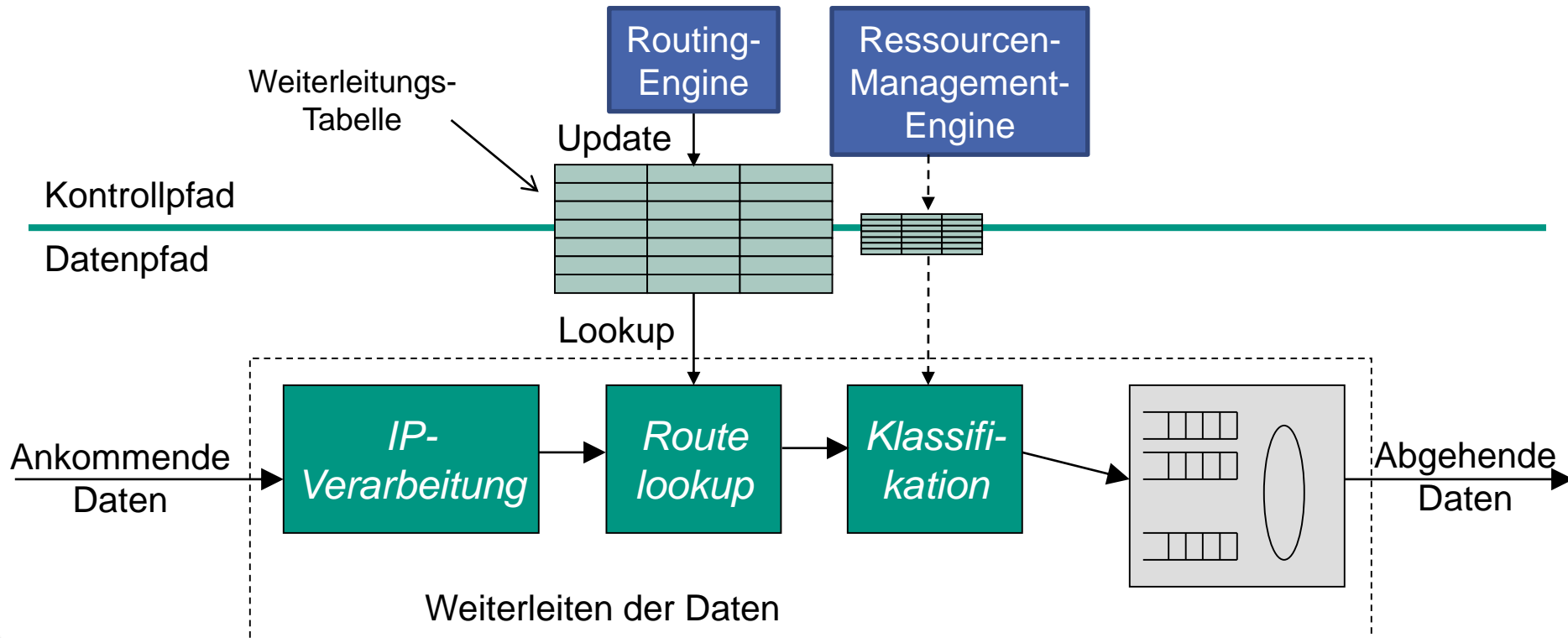
- Es lassen sich die beiden folgenden Varianten unterscheiden
 - Manuelle Konfiguration
 - Dynamische Konfiguration
 - **Dynamic Host Configuration Protocol (DHCP)**
 - DHCP-Server liefert bei Anfrage eine IP-Adresse zurück an den Client

- Wie erhält ein Provider seinen Block von Adressen?
 - Verwaltung der IP-Adressen unterliegt der „Internet Assigned Numbers Authority“ (IANA)
 - Diese untersteht „Internet Corporation for Assigned Names and Numbers“ (ICANN)

- Delegation an regionale Registrierungen („Regional Internet Registries“, RIRs), z.B.
 - APNIC (Asia Pacific Network Information Centre) – Asien/Pazifik
 - ARIN (American Registry for Internet Numbers) – Nordamerika
 - RIPE NCC (Réseaux IP Européens) – Europa, Mittlerer Osten und Zentralasien

Das Internet Protocol (IP)

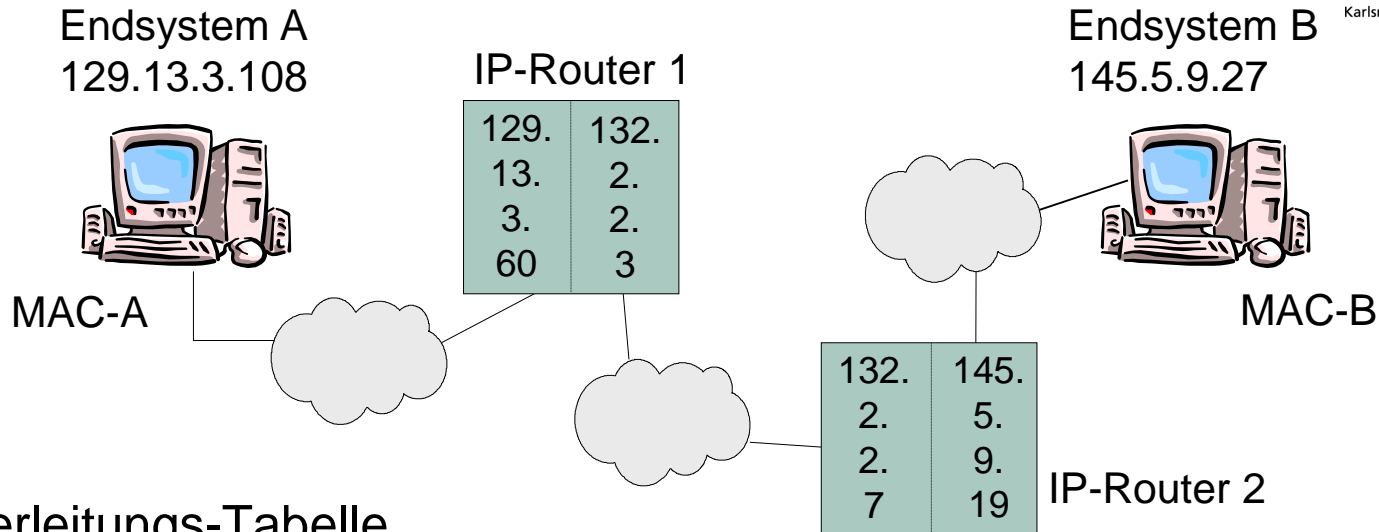
- IP (Internet Protocol): „Das“ Protokoll im Internet
 - Verbindungsloser und unzuverlässiger Dienst
 - Kein IP-basierter Kontext in End- und Zwischensystemen
 - Verantwortlich für das Weiterleiten von Dateneinheiten im Internet
 - Dateneinheiten werden hier als IP-Datagramme bezeichnet



- Für ein Endsystem
 - Ist Rechner mit Zieladresse direkt mit dem Endsystem verbunden (Punkt-zu-Punkt oder gleiches lokales Netz), wird die IP-Dateneinheit direkt zu diesem Empfänger geschickt
 - Ansonsten wird die Dateneinheit an ein voreingestelltes Zwischensystem (*Default-Router*) weitergegeben

- Grundlage für die Weiterleitung: **Weiterleitungstabelle** mit
 - Angabe der Zieladresse (Endsystem- oder Netzadresse)
 - Angabe des „Next-Hop“ Routers
 - Flags, welche die beiden oberen Angaben genauer klassifizieren
 - Angabe der Netzschnittstelle, auf die eine für die Zieladresse bestimmte Dateneinheit ausgegeben werden soll

Weiterleiten im IP-Router



■ Weiterleitungs-Tabelle

- Erstellt von Routing-Protokollen
- IP-Adresse des nächsten Systems und Kennung des Ausgangs

■ Adressumsetzungstabelle

- Erstellt von ARP
- MAC-Adresse des nächsten Systems für IP-Adresse des Endsystems

■ Beispiel

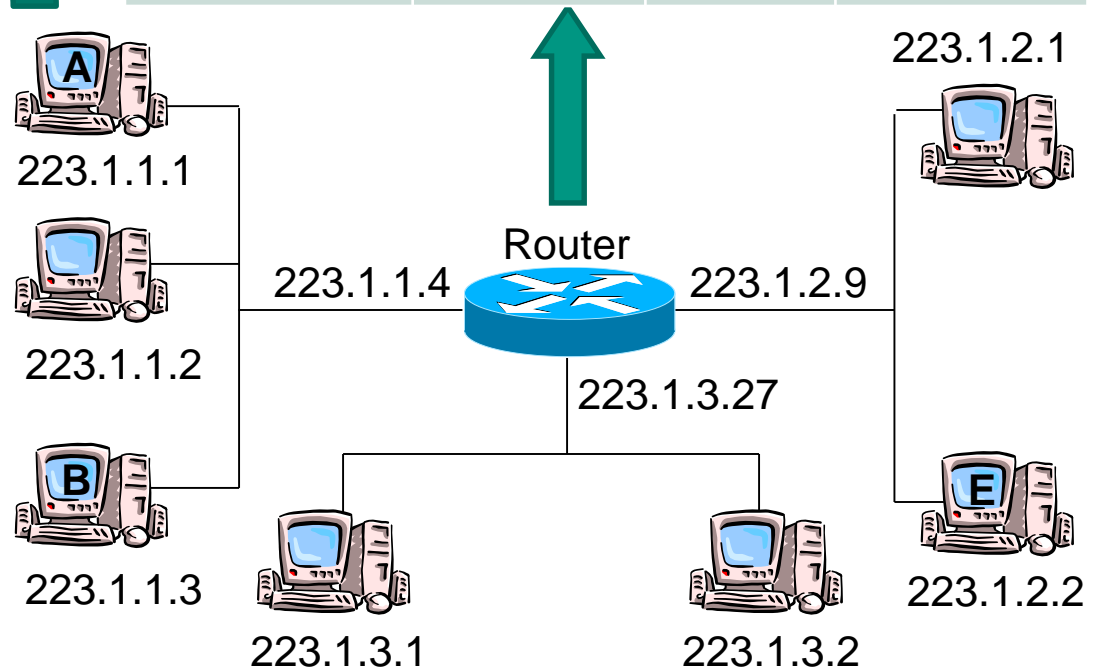
- Ziel: Endsystem B; Quelle: Endsystem A
- Dateneinheit auf dem Weg von Router 1 und Router 2:
 - MAC-Adressen: MAC-Adresse IP-Router 2 (Ziel) und MAC-Adresse IP-Router 1 (Quelle)
 - IP-Adressen: Endsystem B (Ziel), Endsystem A (Quelle)
 - Beachte: IP-Adresse von IP-Router 2 wird in der Dateneinheit nicht transportiert

Tabellen-Lookup

- Aufgabe: Identifikation des nächsten Systems auf dem Weg zum Ziel

Ziel-Netz	Nächster Router	Anzahl Hops	Interface
223.1.1.0/24	–	1	223.1.1.4
223.1.2.0/24	–	1	223.1.2.9
223.1.3.0/24	–	1	223.1.3.27


Ziel-Netz	Nächster Router	Anzahl Hops
223.1.1.0/24		1
223.1.2.0/24	223.1.1.4	2
223.1.3.0/24	223.1.1.4	2



Format einer IPv4-Dateneinheit

Version (4)	Header Length (4)
DSCP* (6)	ECN** (2)
Total Length (16)	
Identifier (16)	
Flags (3)	Fragment Offset (13)
Time to Live (8)	
Protocol (8)	
Header Checksum (16)	
Source Address (32)	
Destination Address (32)	
Options and Padding (variabel)	
Data (variabel)	

* Differentiated Services Code Point

- Kodiert Weiterleitungs-
klasse für die Erfüllung von
Dienstgüte-Anforderungen
(Quality of Service) 

** Explicit Congestion Notification

- Explizite Signalisierung von
Stausituationen 

Segmentieren und Reassemblieren

- Anpassung an verschieden lange maximale Längen der Dateneinheiten unterliegender Netze
- Vorgehensweise

- **Flag-Felder** des IP-Kopfes werden verwendet

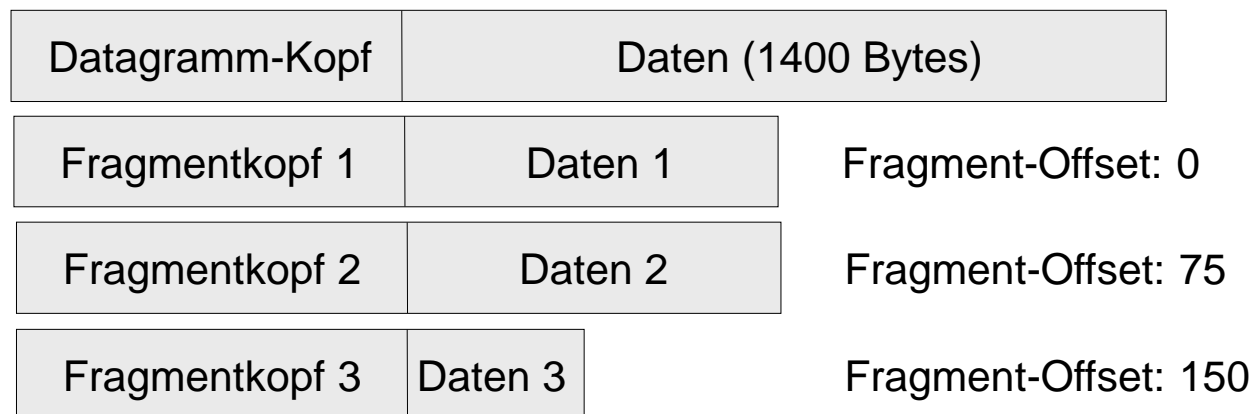
Bit 0: reserviert, muss 0 sein

Bit 1: 0 = darf fragmentiert werden
1 = darf nicht fragmentiert werden

Bit 2: 0 = letztes Fragment
1 = es folgen weitere Fragmente

Fragment-Offset: Stelle, an der empfangenes Fragment in ursprüngliche Dateneinheit eingesetzt werden muss (Basiseinheit: 8 Bytes)

- Beispiel



Empfang einer IP-Dateneinheit

- Folgende Überprüfungen werden durchgeführt
 - Korrekte Länge des Kopfes?
 - IP-Versionsnummer?
 - Korrekte Datagrammlänge?
 - Prüfsumme?
 - Lebenszeit?
 - Protokoll-Identifikation?
 - Adressklasse (Quell- und Zieladresse)?

- Falls ein Fehler erkannt wird
 - Benachrichtigung von ICMP (**Internet Control Message Protocol**)
 - D.h. ICMP wird hier als Funktion aus IP aufgerufen
 - Reagiert möglicherweise mit dem Aussenden einer ICMP-Dateneinheit

<http://pingo.upb.de/>



Das Protokoll ARP

- Ziel-System im gleichen IP-Subnetz: Zustellung über Schicht 2
- ARP: **Address Resolution Protocol**
 - Ist für die Abbildung zwischen IP-Adressen und MAC-Adressen verantwortlich
- Kopf der ARP-Dateneinheit

Netzwerk-Typ		Protokoll-Typ
HLEN	PLEN	Betriebs-Code
MAC-Adresse des Senders		
MAC-Adresse des Senders		IP-Adresse des Senders
IP-Adresse des Senders		MAC-Adresse des Empfängers
MAC-Adresse des Empfängers		
IP-Adresse des Empfängers		

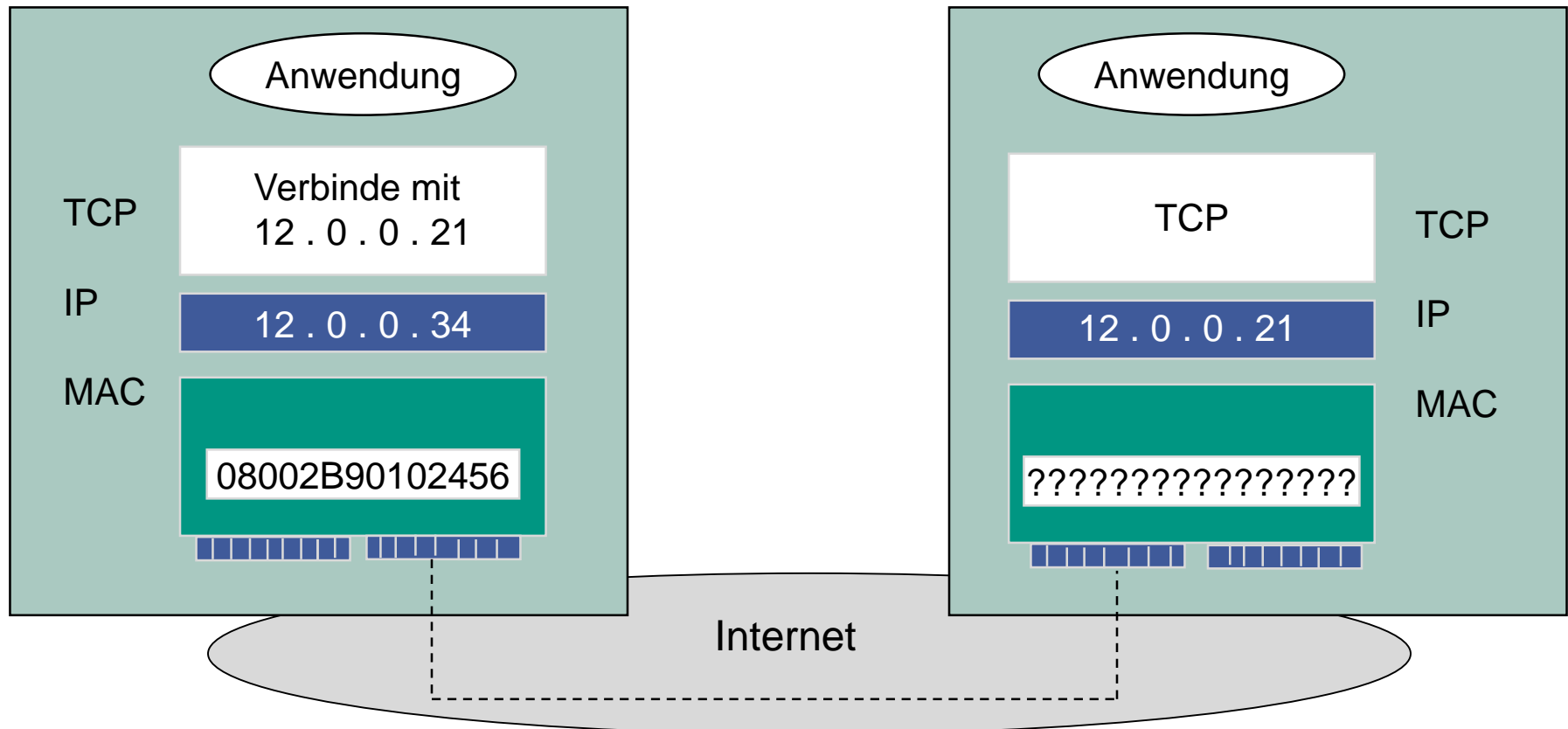


- Netzwerk-Typ: 1 = Ethernet;
 6 = IEEE 802.2
- Protokoll-Typ: 2048 = IP
- HLEN: 2 = 16-Bit MAC-Adresse
 6 = 48-Bit MAC-Adresse
- PLEN: 4 = 32-Bit IP-Adresse
- Betriebs-Code: 1 = Request;
 2 = Reply

HLEN: Header Address Length
 PLEN: Protocol Address Length

Zuordnung von IP- und MAC-Adressen

- Wenn (Ziel-IP-Adresse AND Subnetzmaske) gleich (Eigene IP-Adresse AND Subnetzmaske)
 - Zielsystem ist im gleichen IP-Subnetz
- Welche MAC-Adresse hat das nächste System?



Adressauflösung mit ARP

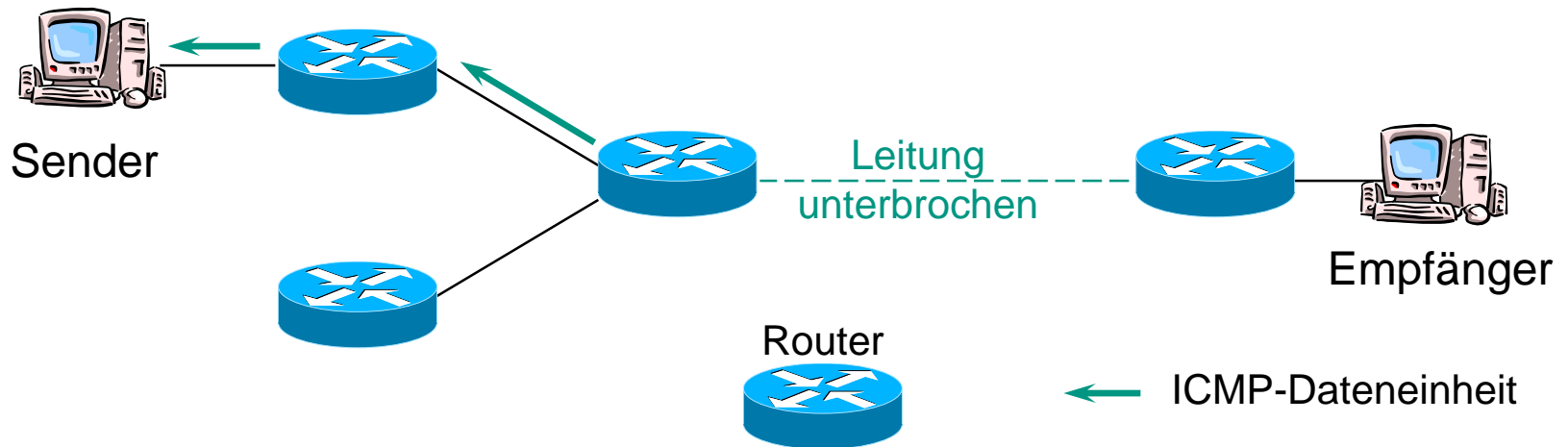
- Dynamisches Lernen von Adresszuordnungen
 - Kleine Tabellen (ARP-Cache bzw. ARP-Table)
 - Maximale Lebensdauer der Einträge (typischerweise 20 Minuten)
 - Hohe Flexibilität

- Nutzt Broadcasting-Fähigkeit der lokalen Netze aus

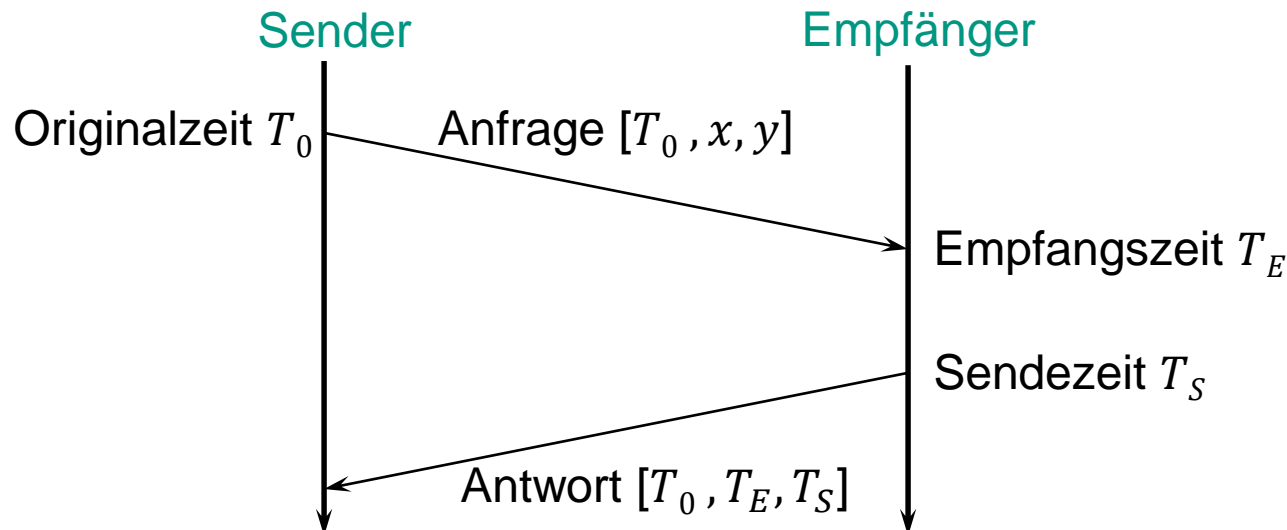
- Falls kein Eintrag im lokalen ARP-Cache vorhanden ist
 - Broadcast eines ARP-Request (enthält Ziel-IP-Adresse)
 - Jedes Endsystem liest ARP-Request und überprüft IP-Adresse
 - Falls eigene IP-Adresse, dann ARP-Reply
 - Suchende Instanz trägt Information in ihren ARP-Cache ein
 - Optional: Andere Endsysteme merken sich ebenfalls Adresszuordnung der suchenden Instanz aus der Anfrage

Internet Control Message Protocol (ICMP)

- Einzelne Verluste von Dateneinheiten werden im Normalfall von IP nicht gemeldet (unzuverlässiger Datagrammdienst)
- Schwerwiegende Probleme (z.B. Unterbrechung einer Leitung) werden zur Vermeidung von Folgefehlern mittels ICMP den Kommunikationspartnern mitgeteilt
- ICMP unterstützt den Austausch von Fehlernachrichten, Statusanfragen und Zustandsinformation



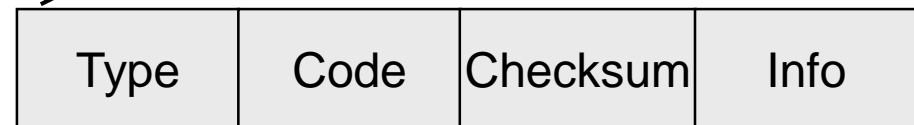
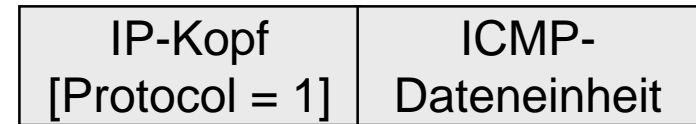
- Echo und Echoantwort (*echo and echo reply*)
 - Dient der Überprüfung der Aktivität von Kommunikationssystemen
 - Der Empfänger einer Echo-Anfrage sendet in der Echo-Antwort die erhaltenen Daten an den Kommunikationspartner zurück
- Zeitstempel und Zeitstempelantwort (*timestamp and timestamp reply*)
 - Dient der Bestimmung von Umlaufzeiten (engl. *Round Trip Time*, RTT)
 - Die Dateneinheiten umfassen mehrere Felder zur Aufnahme von Zeitstempeln, anhand derer die Bearbeitungszeiten beim Empfänger und die Verzögerung im Netz bestimmt werden können



Format von ICMP-Dateneinheiten

■ Übertragung der ICMP-Dateneinheiten

- ICMP-Dateneinheiten werden im Datenteil von IP-Dateneinheiten übertragen und durch den Wert „1“ im Protocol-Feld des IP-Kopfes kenntlich gemacht



■ Format der ICMP-Dateneinheit

- *Type*: Typ der Dateneinheit (z.B. Type = 3 entspricht „Zieladresse nicht erreichbar“)
- *Code*: Genaue Beschreibung der Dateneinheit (z.B. „Netzwerk nicht erreichbar“)
- *Checksum*: Prüfsumme über die gesamte ICMP-Dateneinheit
- Der Inhalt des *Info*-Teils ist abhängig vom Typ der ICMP-Dateneinheit
 - z.B. Felder für Zeitstempel bei Dateneinheit „Zeitstempel und Zeitstempelantwort“

- Probleme mit der bisherigen Version von IP führten zur Weiterentwicklung zu IPv6 (IP Version 6)
 - Unterstützung von Milliarden von Endsystemen (128 bit Adressen)
 - Reduzierung des Umfangs der Routing-Tabellen
 - Vereinfachung des Protokolls zur Effizienzsteigerung
 - Höhere Sicherheit
 - Unterschiedliche Dienstarten, z.B. Echtzeitunterstützung
 - Bessere Multicast-Unterstützung
 - Integrierte Unterstützung mobiler Teilnehmer
 - Koexistenz der alten und neuen Version



Übungen (1)

- 7.1 Nennen Sie verschiedene Vermittlungstechniken und diskutieren Sie deren Eigenschaften, sowie deren Vor- und Nachteile.
- 7.2 Erläutern Sie den Unterschied von virtuellen Verbindungen zum Versand von Datagrammen.
- 7.3 Welche Techniken werden zur Netzkopplung auf den Schichten 1, 2 und 3 eingesetzt und wie funktionieren diese?
- 7.4 Was sind die Vor- und Nachteile von Brücken gegenüber Repeatern?
- 7.5 In welche Kategorien lassen sich Routing-Verfahren unterteilen?
- 7.6 Was ist die Aufgabe eines Routing-Protokolls und wo wird es im Schichtenmodell angesiedelt?
- 7.7 Wie funktionieren Distanz-Vektor-Routing-Protokolle?
- 7.8 Worin unterscheidet sich die Verbreitungsgeschwindigkeit von „guten“ bzw. „schlechten“ Änderungen der Linkkosten in Link-State-Protokollen?

Übungen (2)

- 7.9 Was versteht man unter dem Count-to-Infinity-Problem und durch welche Technik wird ihm begegnet?
- 7.10 Wie funktionieren Link-State-Routing-Protokolle?
- 7.11 Welche Aufgaben hat das Protokoll IP und durch welche Eigenschaften zeichnet es sich besonders aus?
- 7.12 Geben Sie für die IP-Adresse 129.13.6.34 und die Subnetzmaske 255.255.128.0 an, welche IP-Adressen lokal erreichbar sind.
- 7.13 Welches Protokoll kommt zum Einsatz, falls eine Zieladresse im eigenen lokalen Netz erreichbar ist, um die Dateneinheit auf Schicht 2 korrekt zustellen zu können und wie funktioniert dieses Protokoll?
- 7.14 Beschreiben Sie detailliert, wie eine Dateneinheit auf dem Weg zu seiner Zieladresse seinen Weg durch das Internet findet und woher die jeweils benötigten Informationen stammen.
- 7.15 Wozu dient das Protokoll ICMP und in welchem „Verhältnis“ steht es zu IP?



- [Hals05] F. Halsall; **Computer Networking and the Internet**, 5/e; 2005
- [ITWi14] <http://www.itwissen.info/definition/lexikon/Vermittlungsstelle-switching-center-Vst.html>
- [KuRo12] James Kurose, Keith Ross, **Computer Networking**, 6/e, Pearson; 2012
- [PeDa11] L. Peterson, B. Davie; **Computer Networks: A Systems Approach**, 5/e; Morgan Kaufmann; 2011
- [RFC950] J. Mogul, J. Postel; **Internet Standard Subnetting Procedure (RFC 950)**; IETF, August 1985
- [RFC2131] R. Droms; **Dynamic Host Configuration Protocol (RFC 2131)**; IETF, März 1997
- [Sext97] M. Sexton, A. Reid, **Broadband Networking - ATM, SDH and SONET**, Artech-House, 1997
- [Stal10] W. Stallings; **Data & Computer Communications**, 9/e, Prentice Hall; 2010
- [Wart03] F. Warthman; **Delay-Tolerant Networks (DTNs): A Tutorial**; März 2003
 - Online: http://www.ipnsig.org/reports/DTN_Tutorial11.pdf
- [Zhan06] Z. Zhang; **Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges**; IEEE Communications Service; 1st Quarter 2006