

Netzwerktechnologien 3 VO

Dr. Ivan Gojmerac

ivan.gojmerac@univie.ac.at

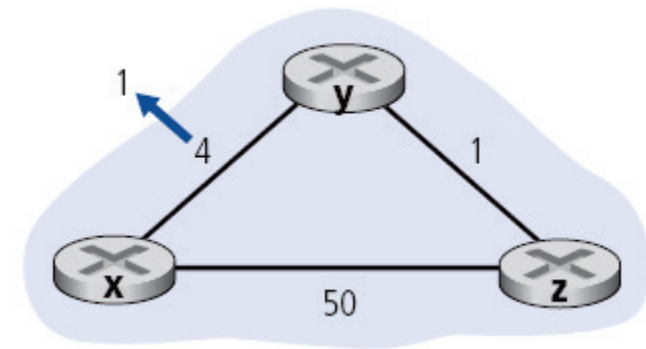
9. Vorlesungseinheit, 22. Mai 2013

Bachelorstudium Medieninformatik
SS 2013

4.5 Distance-Vector: Änderungen der Link-Kosten

Link-Kosten verringern sich:

- Knoten erkennt lokale Änderung
- Berechnet seine Distanzvektoren neu
- Wenn sich sein DV geändert hat, dann werden die Nachbarn informiert



Entwicklung des Distanzvektors nach x:

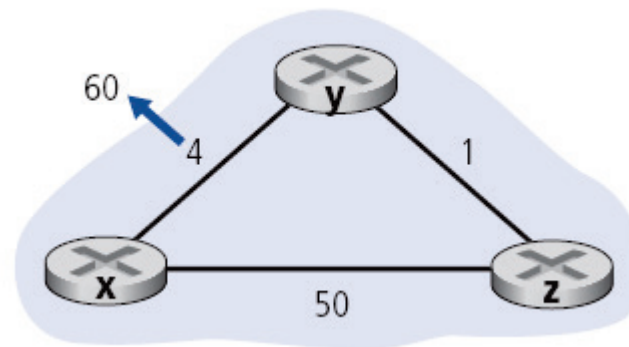
1. Zum **Zeitpunkt t_0** , erkennt y die Verringerung der Kosten, sein DV ändert sich und er informiert seine Nachbarn
2. Zum **Zeitpunkt t_1** , empfängt z den neuen DV von y, sein DV ändert sich, er informiert alle seine Nachbarn

“good news
travels fast”

4.5 Distance-Vector: Änderungen der Link-Kosten

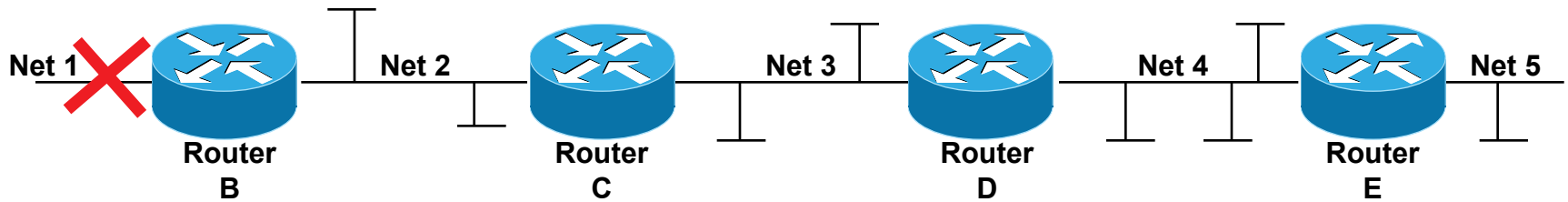
Link-Kosten erhöhen sich:

- Verringerung der Link-Kosten: *Good news travels fast*
- Erhöhung der Link-Kosten: **Bad news travels slow** - “*Count to Infinity*”-Problem!
- 44 Iterationen, bis der Algorithmus zur Ruhe kommt im Beispiel auf dieser Folie



- Ausführlicheres Beispiel mit anderer Topologie auf der nächsten Folie!

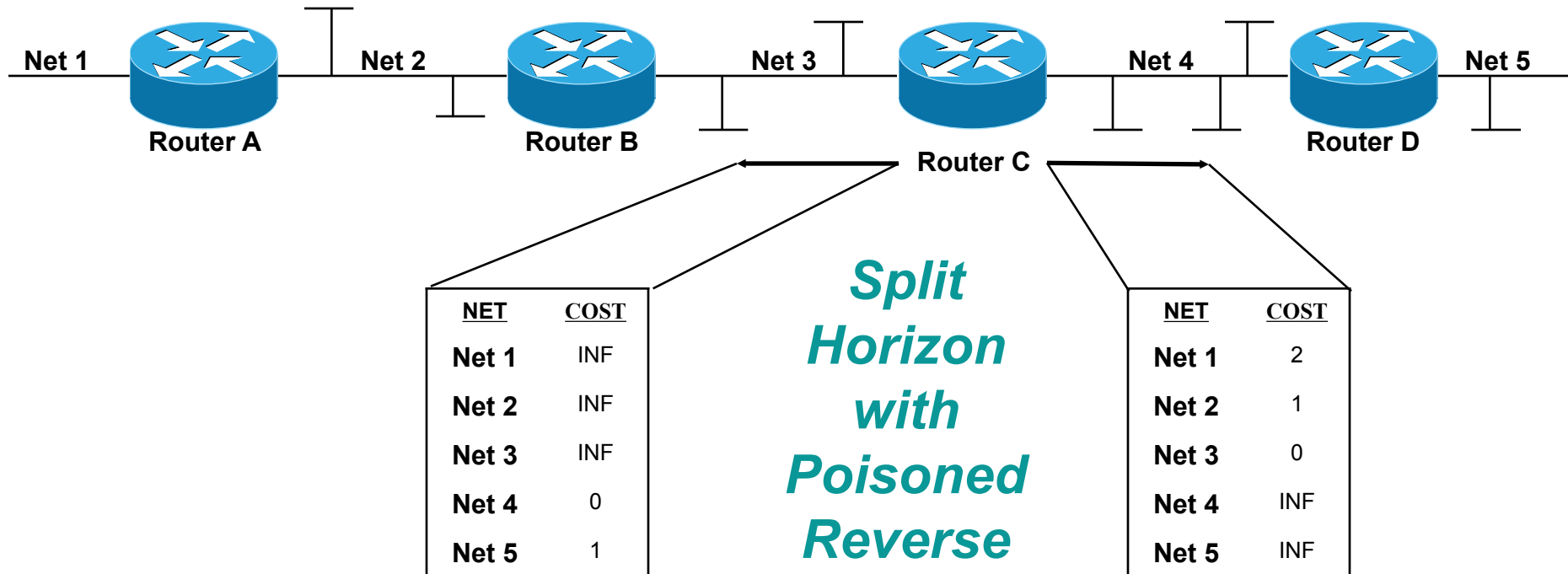
4.5 Distance-Vector: „Count-to-Infinity“-Problem



Pfad-Kosten (Distanz) zu Net 1 in allen Routing Tabellen. Ausfall passiert vor Schritt 1.

Schritt	Distanz zu Net 1	Distanz zu Net 1	Distanz zu Net 1	Distanz zu Net 1
0	1	2	3	4
1	3	2	3	4
2	3	4	3	4
3	5	4	5	4
4	5	6	5	6
5	7	6	7	6
6	7	8	7	8
.
.
X	INF	INF	INF	INF

4.5 Distance-Vector: Split Horizon with Poisoned Reverse



Lösungsansatz → *Split Horizon with Poisoned Reverse*:

- Prinzip: Zum Knoten, über welchen X erreicht wird, eigene Entfernung zu X als unendlich ankündigen!

Wichtig: *Split Horizon with Poisoned Reverse* löst das Problem nicht in allen Fällen!

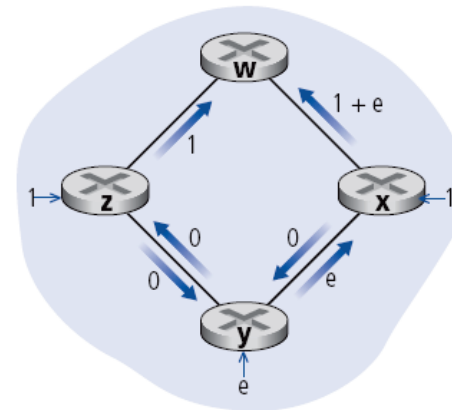
4.5 Link-State- vs. Distance-Vector-Routing

- **Nachrichtenaustausch**
 - LS: Nachrichten werden im ganzen Netz geflutet
 - DV: Nachrichten werden nur mit Nachbarn ausgetauscht
- **Geschwindigkeit der Konvergenz**
 - LS: Schnelle Konvergenz! 😊
 - Fluten der Zustände der Links
 - DV: variiert stark 😞
 - Temporäre Routing-Schleifen sind möglich 😞
 - Count-to-Infinity-Problem 😞
- **Robustheit: *Was passiert, wenn ein Router fehlerhaft ist?***
 - LS:
 - Knoten kann falsche Kosten für einen Link fluten
 - Pfade möglicherweise nicht mehr optimal, das Netzwerk bleibt aber schleifenfrei und verbunden! 😊
 - DV:
 - Router kann falsche Kosten für einen ganzen Pfad ankündigen
 - Fehler propagiert durch das ganze Netzwerk
→ u.U. sehr schädlich 😞

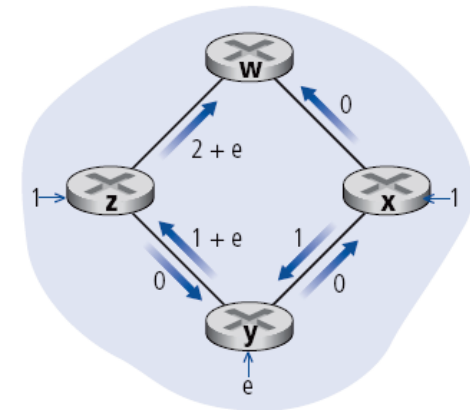
4.5 Oszillationen bei lastabhängigen Linkgewichten

→ Oszillationen sind möglich wenn die Metrik für die Kosten der Links von der Netzwerklast abhängt

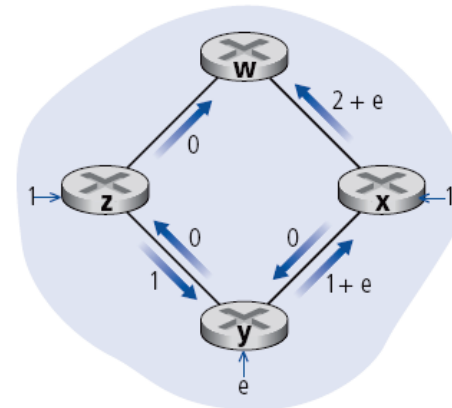
→ Daher ist es sehr ratsam, verkehrsabhängige Link-Metriken strikt zu meiden!



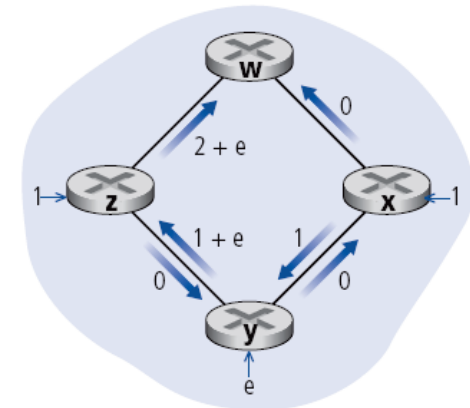
a Anfängliches Routing



b x, y entdecken den im Uhrzeigersinn verlaufenden besseren Pfad nach w



c x, y, z entdecken den gegen den Uhrzeigersinn verlaufenden besseren Pfad nach w



d x, y, z entdecken den im Uhrzeigersinn verlaufenden besseren Pfad nach w

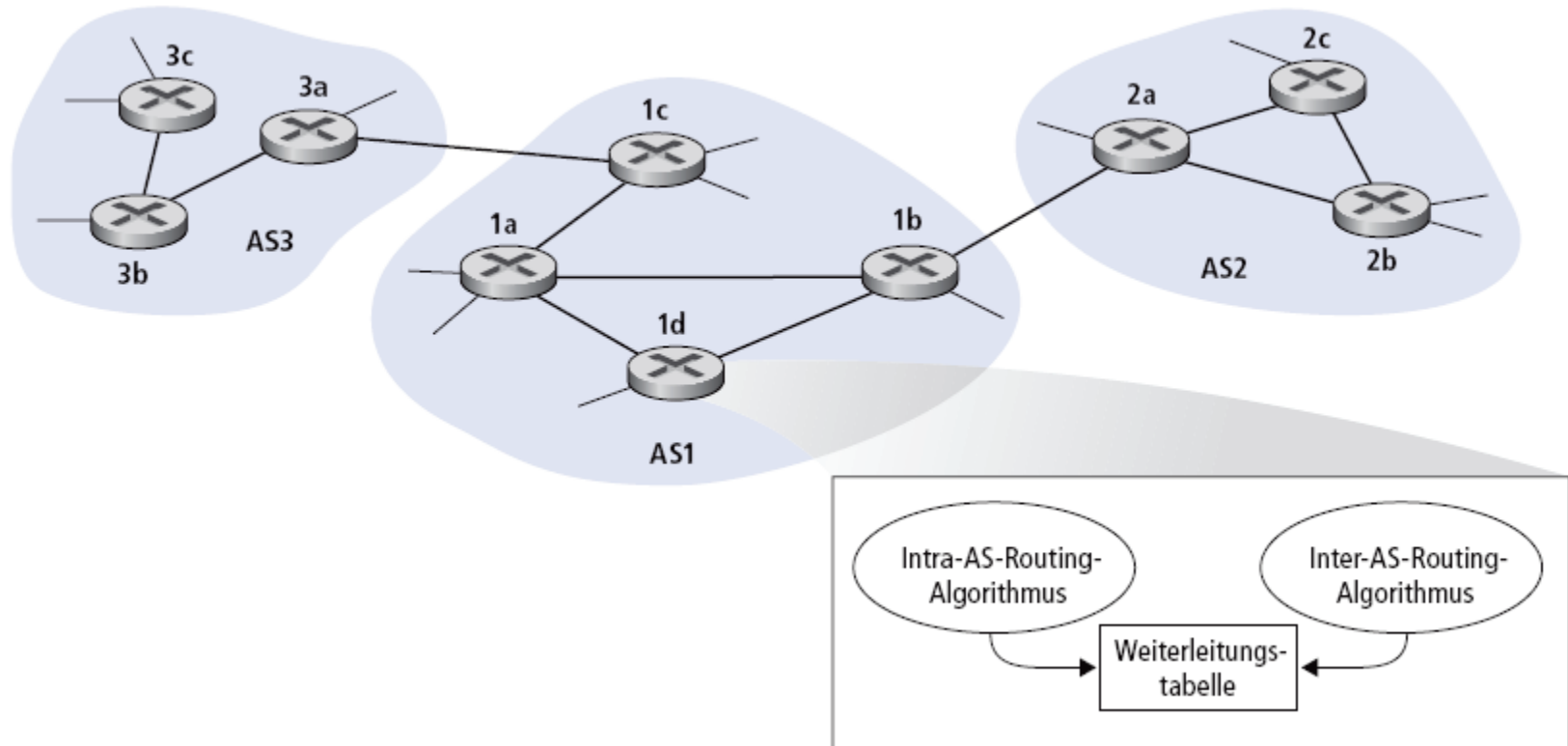
4.5 Hierarchisches Routing im Internet (1)

- Unsere Folien bisher:
 - Alle Router sind gleich
 - Das Netzwerk ist „flach“ und besitzt keine Hierarchie
 - Entspricht nicht der Realität
- Eine Frage der Administration:
 - Internet = Netzwerk von Netzwerken
 - Jede Organisation hat eigene Politiken und Präferenzen bezüglich ihres Netzwerkes
- Eine Frage der Größenordnung:
 - Viele Zielnetzwerke!
 - Es können nicht alle Netzinternen Links weltweit berücksichtigt werden, da die Routing-Protokolle (d.h. der Austausch von Routing-Informationen) alle Links im Internet überlasten würden

4.5 Hierarchisches Routing im Internet (2)

- Router werden zu Regionen zusammengefasst, diese nennt man **Autonome Systeme (AS)**
- Router innerhalb eines AS verwenden ein Routing-Protokoll
 - “Intra-AS”-Routing-Protokoll
 - Router in verschiedenen AS können verschiedene Intra-AS-Routing-Protokolle verwenden
- Manchmal gilt:
 - Eine Organisation = ein AS
 - Es gibt aber auch Organisationen (z.B. einige ISPs), die aus mehreren AS bestehen
- Gateway-Router:
 - Ein Router in einem AS, der eine Verbindung zu einem Router in einem anderen AS hat
- **Routing zwischen AS**
 - „Inter-AS“-Routing-Protokoll

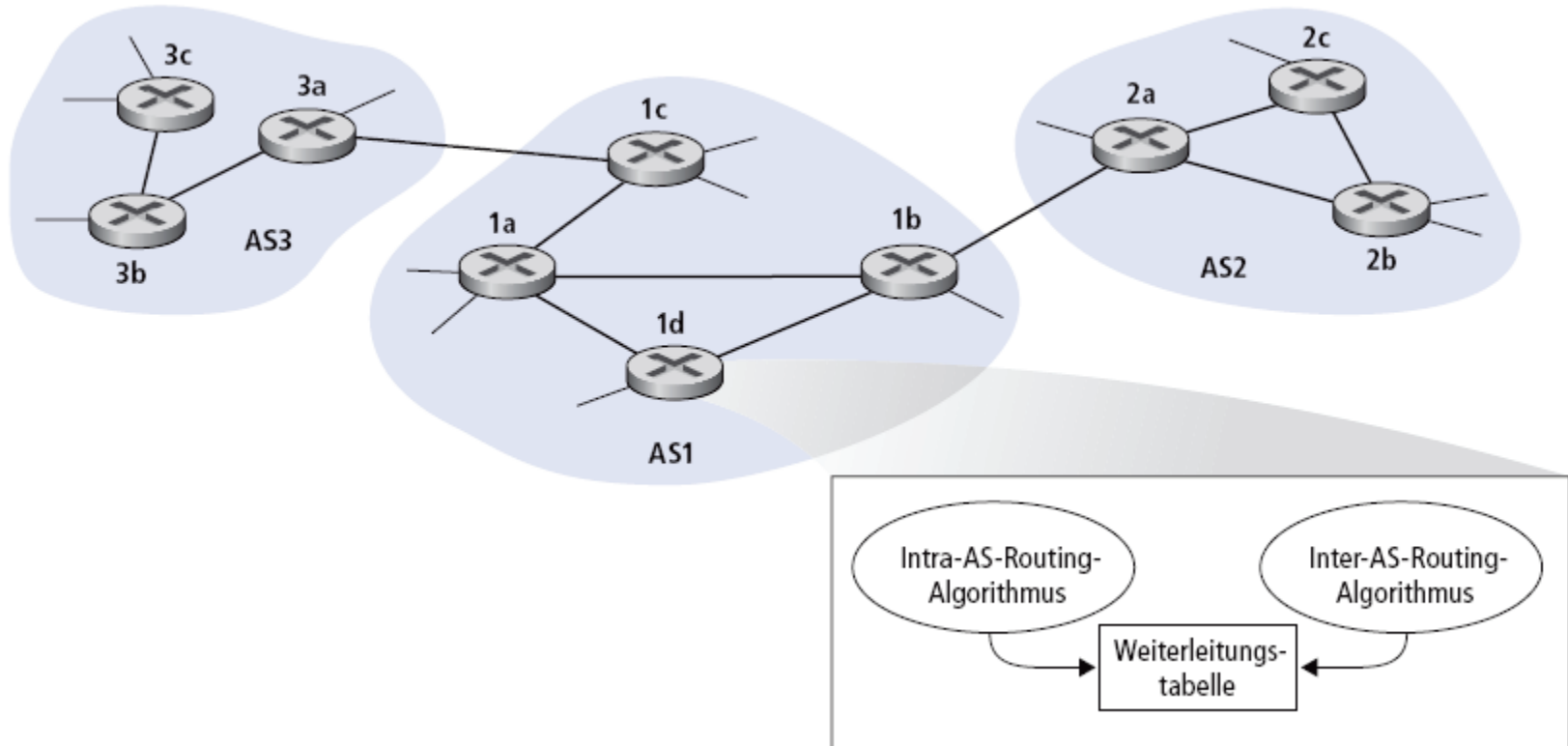
4.5 Verbundene Autonome Systeme



Routing-Tabelle wird durch **Intra-AS-** und **Inter-AS-Routing-Algorithmen** gefüllt:

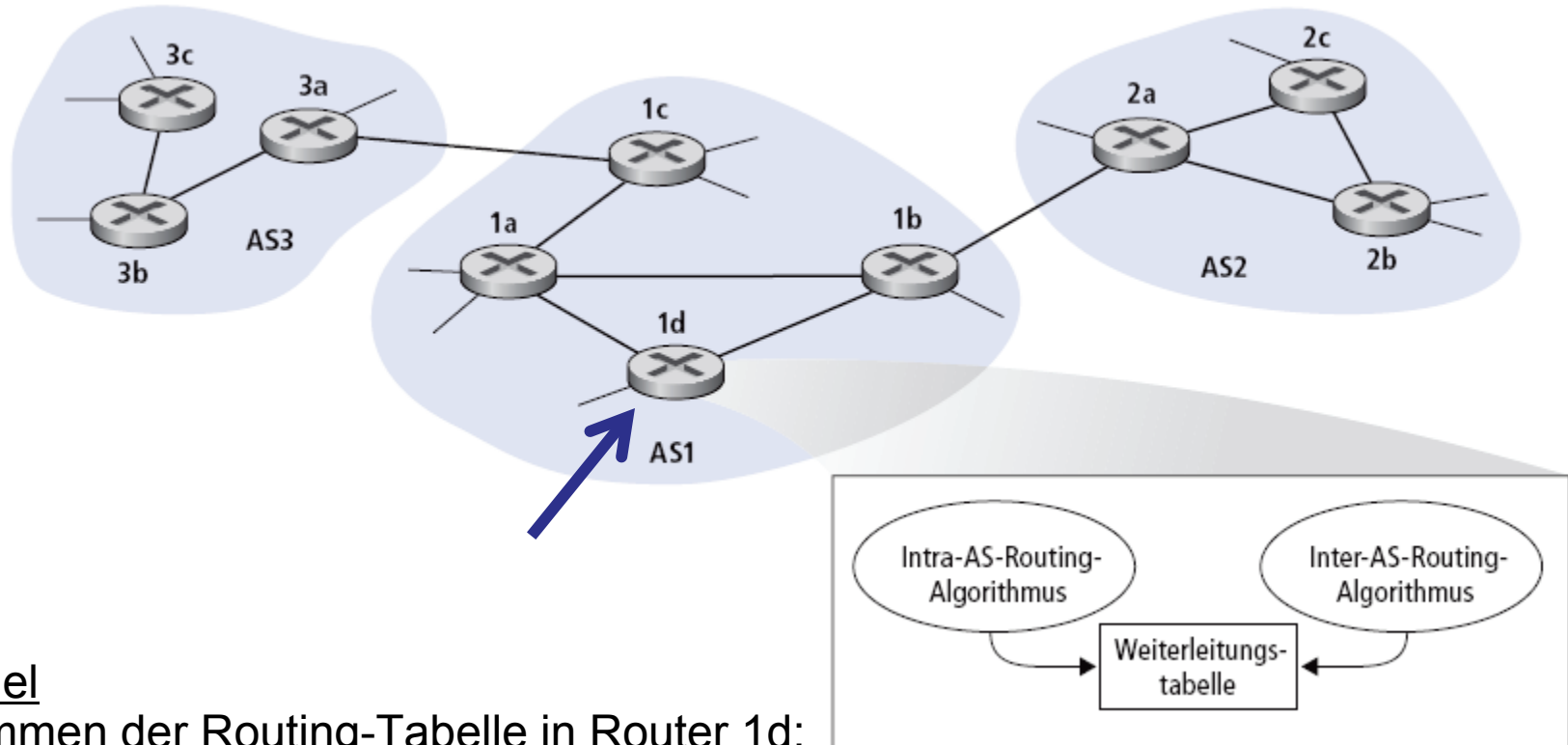
- Intra-AS-Einträge für interne Ziele
- Inter-AS- & Intra-AS-Einträge für externe Ziele

4.5 Aufgaben des Inter-AS-Routing



- Wenn ein Router in AS1 ein Paket für ein Ziel außerhalb von AS1 erhält:
 - Router sollte das Paket zu einem der Gateway-Router in AS1 weiterleiten
 - Aber zu welchem?
- AS1 muss mit Hilfe von Inter-AS-Routing Folgendes tun:
 - Lernen, welche Ziele über die Autonomen Systeme AS2 und AS3 erreichbar sind
 - Verteilen dieser Informationen an alle Backbone-Router in AS1

4.5 Aufgaben des Inter-AS-Routing



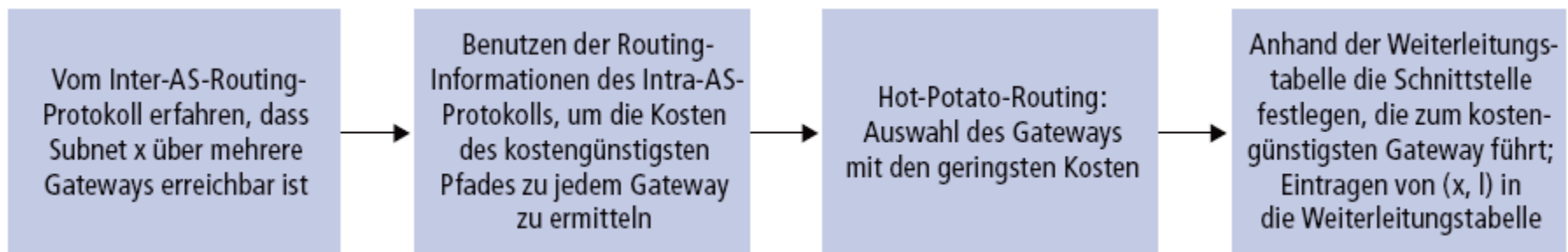
Beispiel

Bestimmen der Routing-Tabelle in Router 1d:

1. Angenommen, AS1 lernt durch das Inter-AS-Routing-Protokoll, dass Netzwerk x von AS3 (Gateway-Router 1c), aber nicht von AS 2 aus erreicht werden kann
2. Inter-AS-Routing-Protokoll propagiert diese Information zu allen internen Backbone-Routern
3. Backbone-Router 1d bestimmt durch das Intra-AS-Routing-Protokoll, dass sein Interface I auf dem kürzesten Pfad zu 1c liegt
4. Router 1d nimmt einen Eintrag (x,I) in der Routing-Tabelle vor

4.5 Beispiel: Alternative Routen

- Angenommen AS1 lernt durch das Inter-AS-Routing-Protokoll, dass Netzwerk X sowohl über AS2 als auch über AS3 zu erreichen ist
- Für den Eintrag in die Routing-Tabelle muss Router 1d sich für einen Pfad entscheiden
- Ebenfalls Aufgabe des Inter-AS-Routing-Protokolls
- Eine Möglichkeit, falls die externen (d.h. BGP-4) Metriken gleich sind, und falls der Betreiber keine explizite Präferenz hat:
 - **Hot Potato-Routing:** Schicke das Paket an den nächsten Gateway-Router, der es in ein anderes AS weiterleiten kann



4.6 Intra-AS-Routing

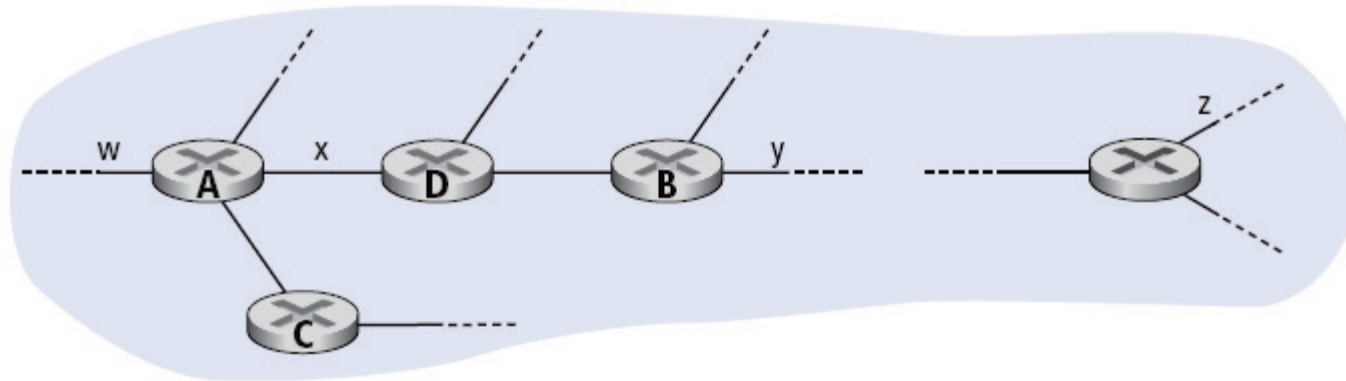
- Auch **Interior Gateway Protocol (IGP)**
- Die bekanntesten Intra-AS-Routing-Protokolle:
 - **RIP**: Routing Information Protocol → *Distance Vector*
 - **IGRP**: Interior Gateway Routing Protocol → *Distance Vector* (Proprietär: Cisco)
 - **OSPF**: Open Shortest Path First → Link State
 - **IS-IS**: Intermediate System to Intermediate System → Link State

4.6 Routing Information Protocol (RIP)

- Version 1 spezifiziert in RFC 1058
- Version 2 (kompatibel mit Version 1) spezifiziert in [RFC 2453](#)
- Distance-Vector-Algorithmus
- War bereits in der BSD-UNIX-Distribution von 1982 enthalten
- Metrik: Anzahl der Hops (Maximum = 15 Hops)

- Distanzvektoren werden zwischen den Nachbarn alle 30 Sekunden per **RIP-Advertisement** ausgetauscht
- Jedes Advertisement enthält eine Liste von bis zu 25 Zielnetzwerken im Inneren des Autonomen Systems

4.6 RIP Beispiel



Routing-Tabelle in D:

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

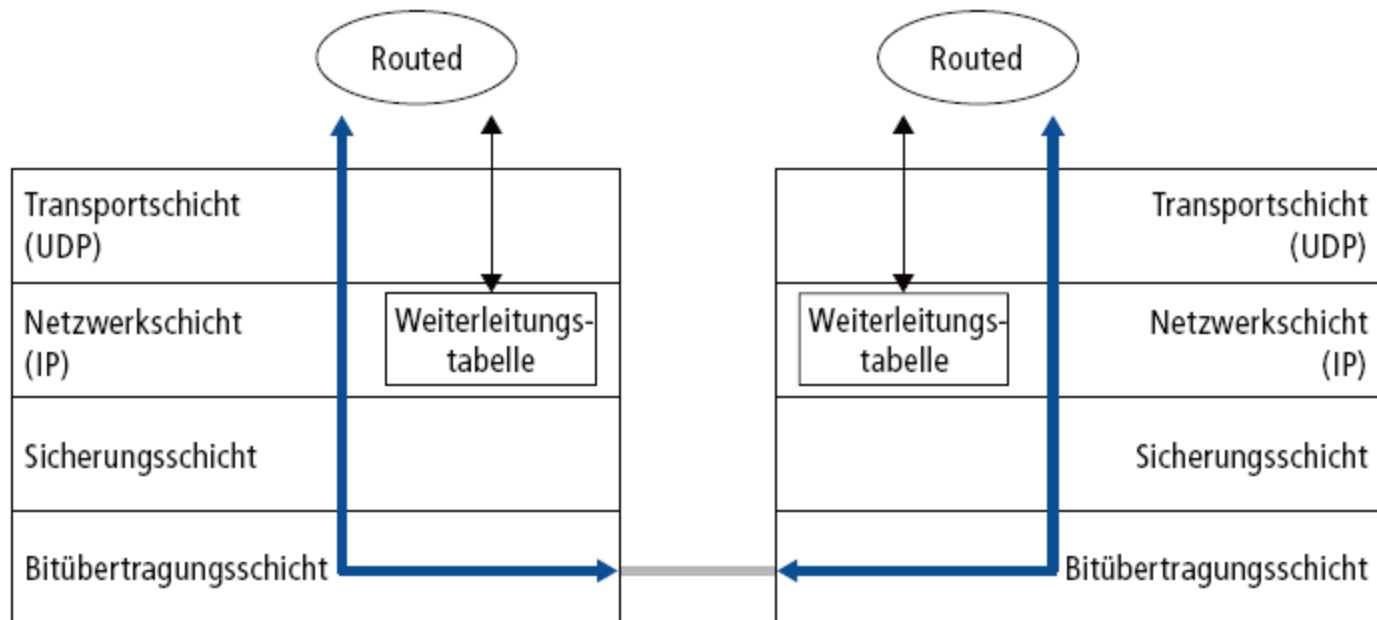
4.6 RIP - Brechen von Links

Wenn von einem Nachbarn 180 Sekunden lang kein Advertisement empfangen wurde, gilt der Nachbar als nicht mehr vorhanden

- Alle Routen über diesen Nachbarn werden ungültig
 - Neuberechnung des lokalen Distanzvektors
 - Verschicken des neuen Distanzvektors (wenn er sich verändert hat)
 - Nachbarn bestimmen ihren Distanzvektor neu und verschicken ihn gegebenenfalls
- ...
- Die Information propagiert schnell durch das Netzwerk
 - *Split Horizon with Poisoned Reverse* wird verwendet, um Routing-Schleifen zu vermeiden (unendlich ist hier 16 Hops!)

4.6 RIP-Architektur

- Die Routing-Tabelle kann von RIP in einem Prozess auf Anwendungsebene gepflegt werden: z.B. *routed* (für „route daemon“)
- Advertisements werden per UDP verschickt



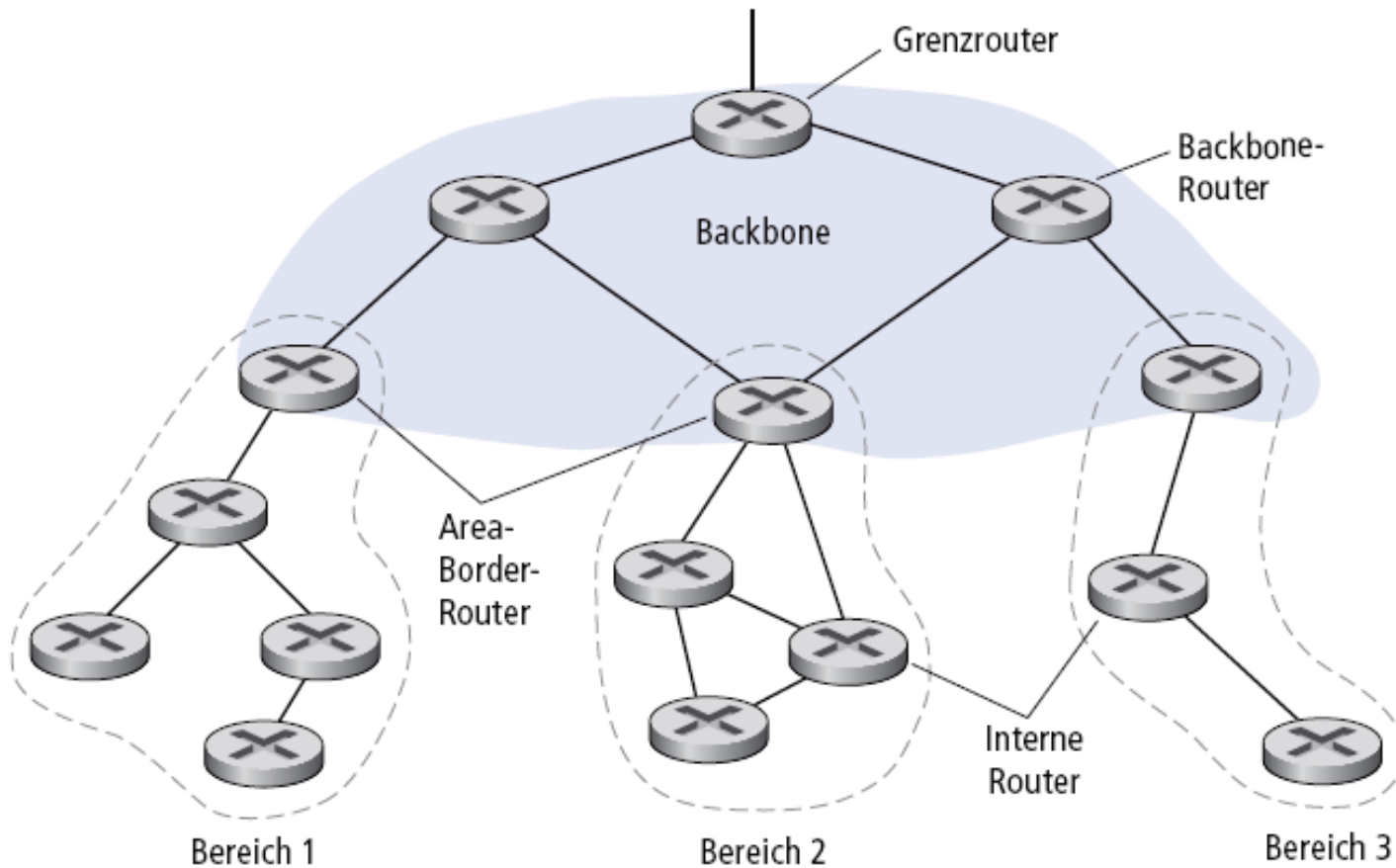
4.6 Open Shortest Path First (OSPF)

- Version 2 spezifiziert in [RFC 2328](#)
- “Open”: frei verfügbar
- „Shortest Path First“: verwendet **Link-State-Routing-Algorithmus**
 - Periodisches Fluten von Link-State-Paketen (LSAs – Link State Advertisements)
 - Jeder Router kündigt seine Links an
 - Diese Ankündigungen werden effizient geflutet
 - Ein Router schickt eine empfangene Ankündigung allen seinen Nachbarn, von denen er dieselbe Ankündigung noch nicht erhalten hat
 - OSPF-Pakete werden direkt in IP-Pakete eingepackt
 - Topologieänderungen (z.B. Linkausfälle) werden schnell mittels LSAs angekündigt
 - Topologie des Netzwerkes in allen Routern bekannt
 - Wird in einer sogenannten „Netzwerkkarte“ oder „Topology Map“ gespeichert
 - Routen werden mit Dijkstras Algorithmus berechnet

4.6 Eigenschaften von OSPF

- Schnell, effizient, schleifenfrei → besitzt alle guten Eigenschaften von Link-State-Routing!
- Sicherheit ist gegeben: Alle OSPF-Nachrichten können authentifiziert werden
- Hierarchisches OSPF in größeren autonomen Systemen

4.6 Hierarchisches OSPF (1)



4.6 Hierarchisches OSPF (2)

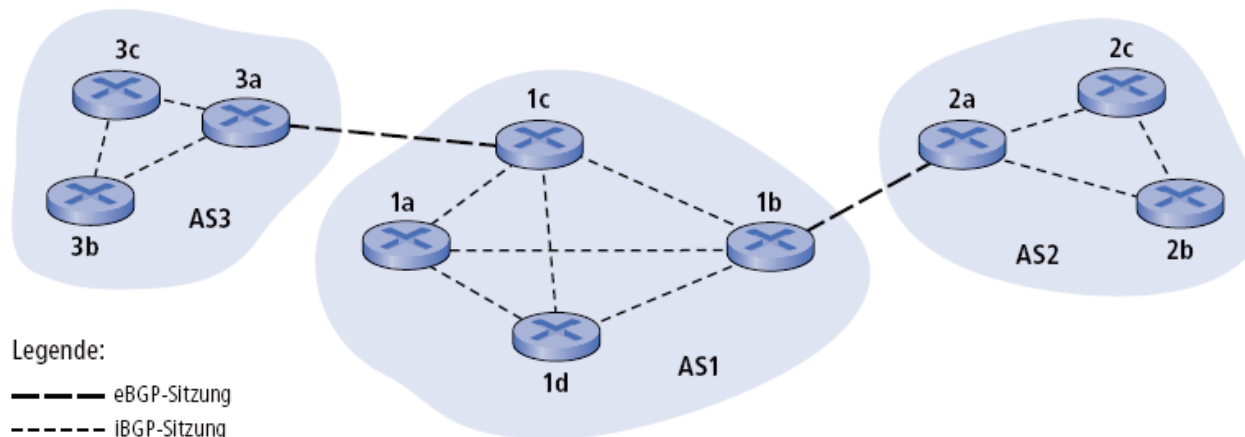
- Zweistufige Hierarchie: Local Area, Backbone
 - Jedes AS hat ein Backbone, das aus mehreren Routern besteht
 - Jeder Router in einer Local Area kennt deren detaillierte Topologie und die Richtung zu den Netzwerken der anderen Local Areas
- Area-Border-Router:
 - Zusammenfassen der Distanzen zu den Netzwerken in der eigenen Local Area
 - Ankündigen dieser Zusammenfassung an die anderen Area-Border-Router (d.h. Backbone-Router)
 - Ankündigungen der Zusammenfassungen der anderen Area-Border-Router (d.h. Backbone-Router) in der Local Area
- OSPF-Backbone-Router: führt OSPF-Routing und *Traffic Forwarding* im Backbone durch
- OSPF-Boundary-Router: stellt eine Verbindung zu anderen Autonomen Systemen her

4.6 Border Gateway Protocol (BGP)

- Das Border Gateway Protocol, Version 4, ist der De-facto-Standard für Inter-AS-Routing im Internet
- Spezifiziert in [RFC 4271](#)
- BGP erlaubt es einem AS:
 - Informationen über die Erreichbarkeit von Netzen von seinen benachbarten Autonomen Systemen zu erhalten
 - Diese Informationen an die Router im Inneren des eigenen AS weiterzuleiten
 - „Gute“ Routen zu einem gewünschten Zielnetzwerk zu bestimmen, wobei die Qualität einer Route von den Informationen über die Erreichbarkeit und Politiken abhängig ist
- Außerdem ermöglicht BGP es einem AS, sein eigenes Netzwerk anzukündigen und so dessen Erreichbarkeit den anderen Autonomen Systemen im Internet mitzuteilen

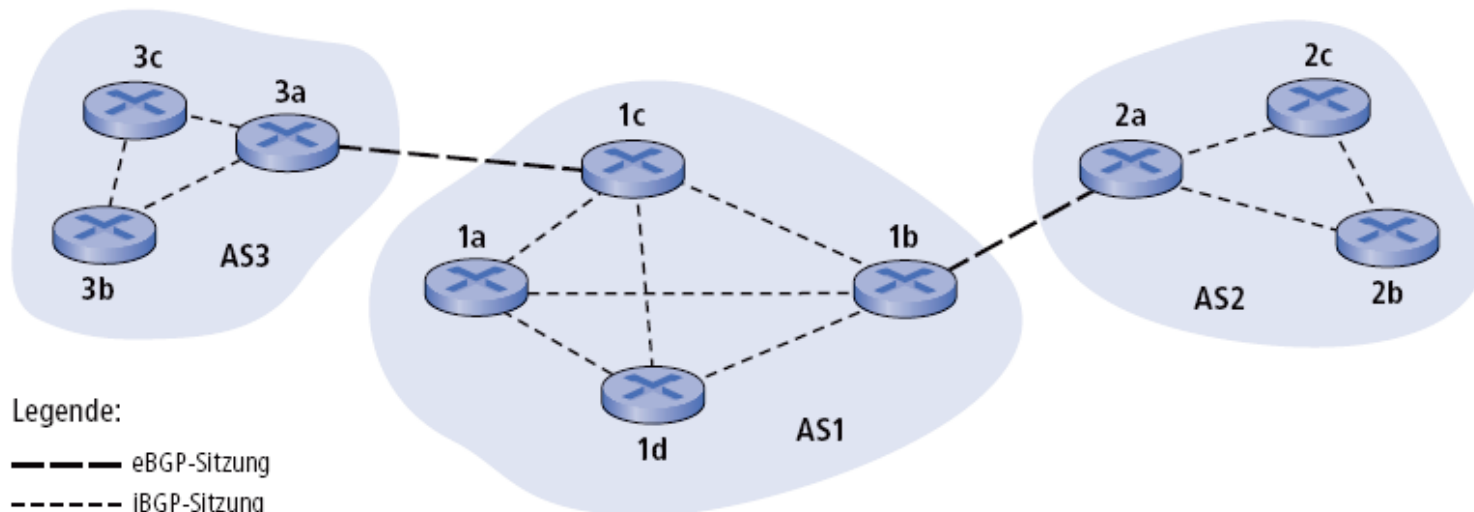
4.6 BGP-Grundlagen

- Paare von Routern (BGP-Peers) tauschen Routing-Informationen über TCP-Verbindungen aus: BGP-Sitzung (engl. BGP session)
 - Wenn beide Router im selben AS sind: **interne** BGP-Sitzung (iBGP)
 - Wenn beide Router in verschiedenen AS sind: **externe** BGP-Sitzung (eBGP)
- Wichtig: BGP-Sitzungen entsprechen nicht notwendigerweise physikalischen Links
- Wenn AS2 ein Präfix mitsamt Pfadattributen an AS1 meldet (z.B. 167.3/16), dann verspricht AS2, dass es alle Datagramme auf dem gemeldeten Pfad weiterleitet, deren Zieladressen zu diesem Präfix passen
 - AS2 kann Präfixe in seinen Ankündigungen aggregieren



4.6 Verbreiten der Erreichbarkeitsinformationen

- AS3 kündigt die Erreichbarkeit von Präfixen über die eBGP-Sitzung zwischen 3a und 1c an
- 1c kann iBGP verwenden, um diese Informationen im AS zu verbreiten
- 1b kann dann diese neuen Informationen über die eBGP-Sitzung zwischen 1b und 2a weitermelden und die Präfixe dem Autonomen System 2 ankündigen



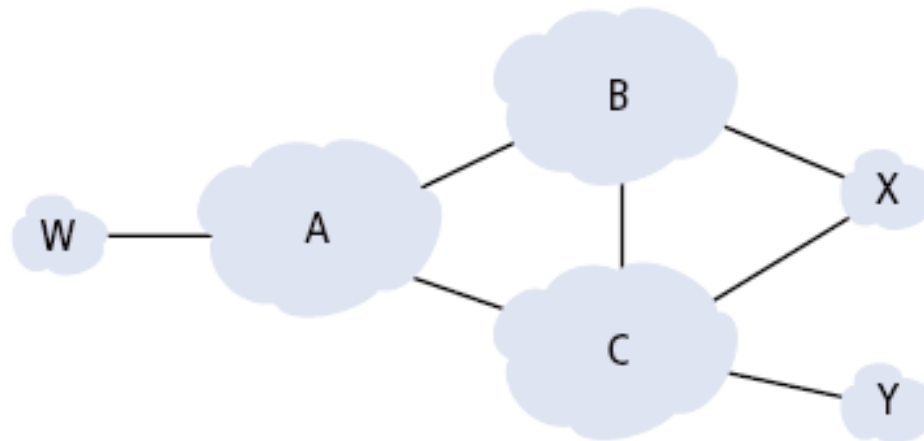
4.6 Pfadattribute und BGP-Routen

- Wenn ein Präfix angekündigt wird, dann beinhaltet diese Ankündigung sogenannte Pfadattribute
 - Präfix + Attribute = Route (in BGP-Terminologie)
- Zwei wichtige Attribute:
 - **AS-PATH**: eine Liste aller AS, durch welche die Ankündigung weitergeleitet wurde: AS 67, AS 17
 - **NEXT-HOP**: zwei Autonome Systeme können über mehr als ein Paar von Gateway-Routern in Kontakt stehen. Um die Ankündigung einem Router im anderen AS zuordnen zu können, schickt der ankündigende Router seine IP-Adresse als NEXT-HOP-Attribut mit
- Der empfangende Gateway-Router entscheidet durch konfigurierbare Politiken anhand der Attribute, ob eine Ankündigung angenommen werden soll bzw. welche Ankündigung bevorzugt wird

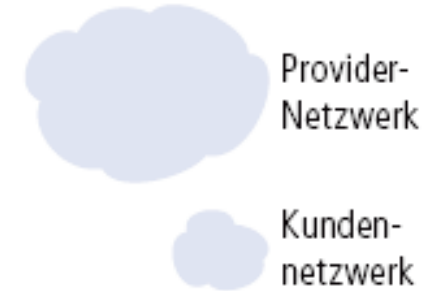
4.6 BGP-Routenwahl

- Router können mehr als eine Route zum Ziel angeboten bekommen. Ein Router muss entscheiden, welche Route verwendet wird.
 - Regeln nach Priorität (nur ein Auszug!):
 1. Lokale Präferenz
 2. Kürzester AS-PATH
 3. Dichtester NEXT-HOP-Router: Hot Potato Routing
 4. Weitere Kriterien (häufigster Fall)
- Die BGP-Routenwahl ist eine firmenpolitische Entscheidung!

4.6 BGP-Routing-Politiken

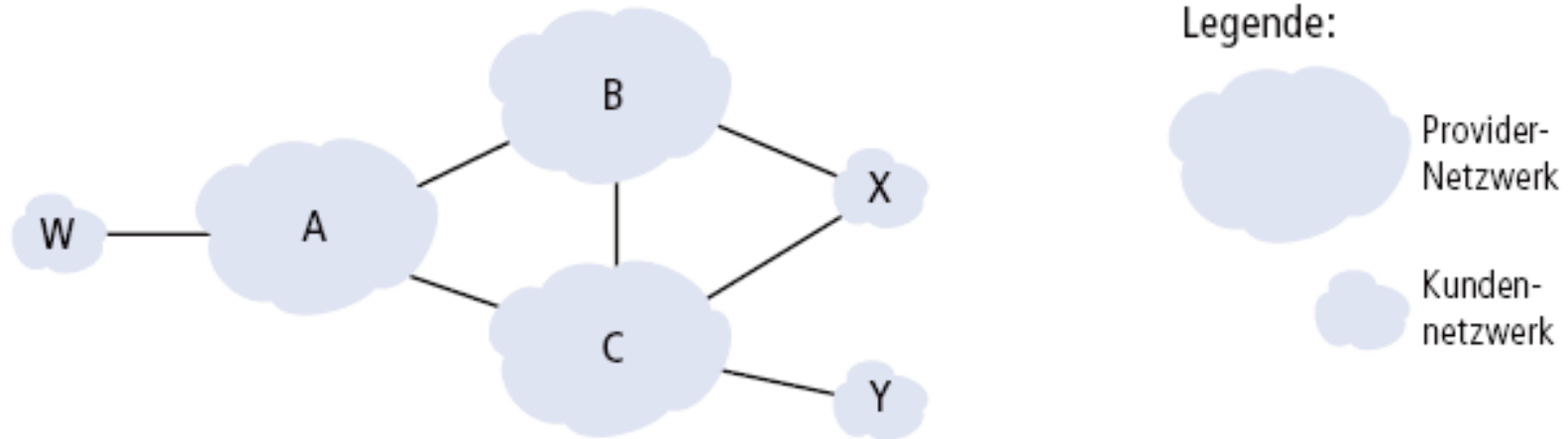


Legende:



- A,B,C sind Netzwerke von Providern
- X,W,Y sind Netzwerke von Kunden (der Provider)
- X ist „dual homed“: an zwei Provider angebunden
 - X möchte keine Daten von B nach C weiterleiten
 - ➔ daher wird X keine Route nach C an B ankündigen

4.6 BGP-Routing-Politiken



- A kündigt B den Pfad AW an (ein Kundenpfad wird angekündigt)
- B kündigt X den Pfad BAW an (alle Pfade werden immer den eigenen Kunden angekündigt)
- Sollte B den Pfad BAW auch C ankündigen?
 - **Nein!** B hätte nichts davon, da weder W noch C Kunden von B sind
 - B und C sind lediglich *Peers*, die nur den eigenen Verkehr und den der eigenen Kunden untereinander austauschen!
 - B möchte, dass C Datenpakete zu W über A (als den Provider von W) leitet
 - B möchte nur Verkehr an oder von seinen Kunden weiterleiten

4.6 BGP-Routing-Politiken

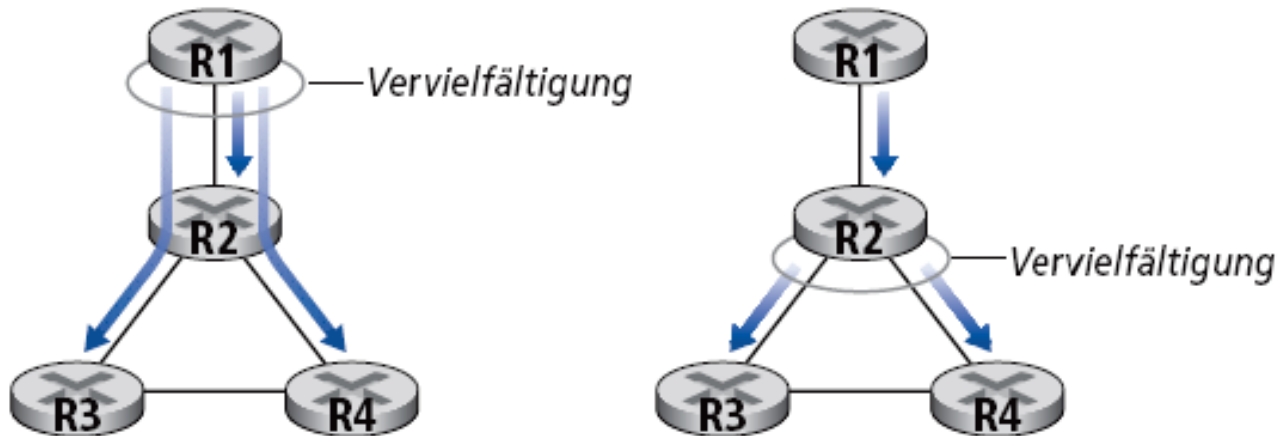
Warum verschiedene Protokolle für Intra-AS- und Inter-AS-Routing?

- Politiken:
 - Inter-AS: Eine Organisation möchte kontrollieren, wie (und ob) der Verkehr anderer Organisationen durch das eigene Netzwerk geleitet wird
 - Intra-AS: eigener Verkehr, eigene Administration, hier sind keine Politiken nötig
- Größenordnung:
 - Hierarchisches Routing reduziert die Größe der AS-internen Routing-Tabellen und reduziert den Netzwerkverkehr für Routing-Updates
 - ➔ Dringend notwendig für Inter-AS-Routing, nicht allzu wichtig in Intra-AS-Routing
- Performance:
 - Intra-AS: kann sich auf Netzwerk-Performance konzentrieren
 - Inter-AS: Politiken können wichtiger sein als Performance

4.7 Broadcast-Routing

- Liefert ein Paket an alle Knoten im Netz aus
- Duplikation in der Quelle ist ineffizient:

Erzeugung bzw. Übertragung von Paketkopien



4.7 Broadcast-Routing: Duplikation im Inneren des Netzwerks

- **Optionen:**
 - **Kontrolliertes Fluten:** Ein Knoten leitet ein Paket nur dann weiter, wenn er es noch nie weitergeleitet hat
 - Jeder Knoten merkt sich die Pakete, die er weitergeleitet hat
 - Oder er verwendet **Reverse Path Forwarding (RPF)**: Pakete werden nur dann weitergeleitet, wenn sie auf dem kürzesten Pfad von diesem Knoten zum Sender angekommen sind
 - **Spannbäume**
 - Redundante Pakete werden vollständig vermieden

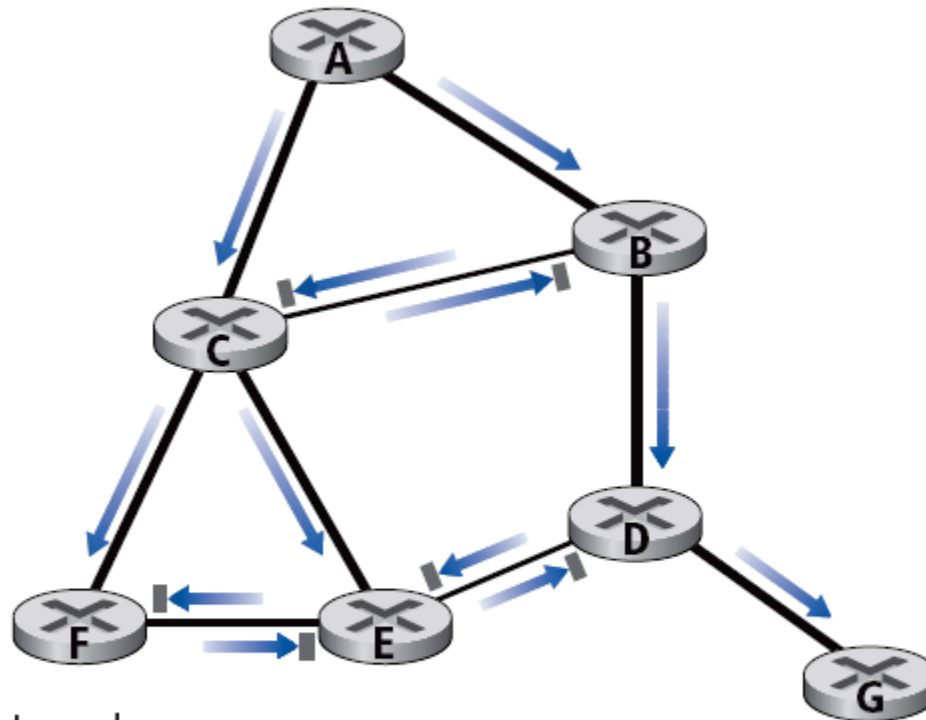
4.7 Broadcast-Routing: Reverse Path Forwarding (RPF)

- Verwendet das Wissen eines Routers bezüglich der kürzesten Unicast-Pfade **von diesem Router zum Sender**.
- Jeder Router führt folgenden Algorithmus aus:

```
if (Multicast-Datagramm wurde auf dem eingehenden Link, der auf dem kürzesten Pfad zum Sender liegt, empfangen)
```

```
    then flute das Datagramm auf alle ausgehenden Links  
    else ignoriere das Datagramm
```

4.7 Broadcast-Routing: Reverse Path Forwarding (RPF) - Beispiel



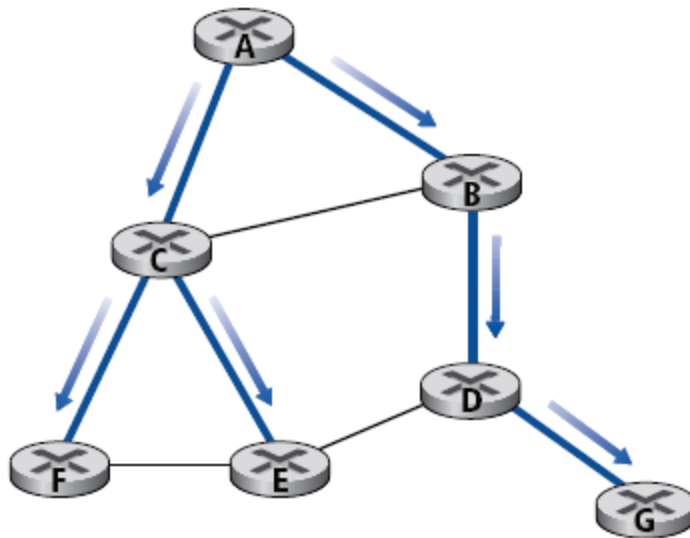
Legende:

➡ Paket wird weitergeleitet

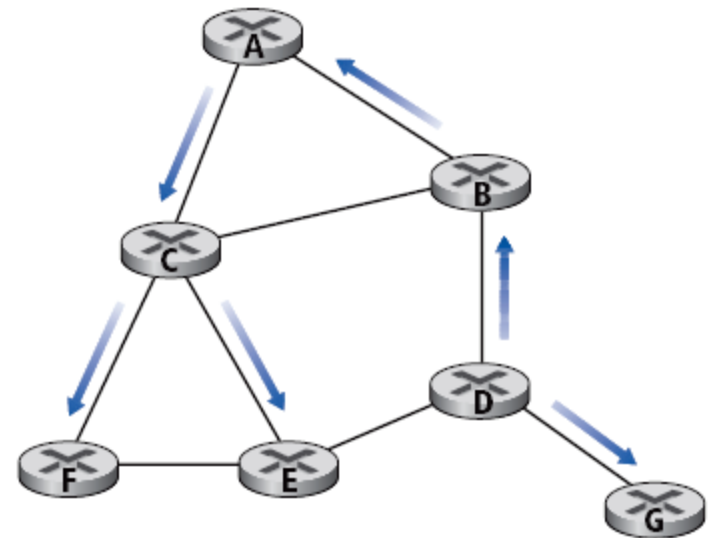
➡▬ Paket wird vom empfangenden Router nicht weitergeleitet

4.7 Broadcast-Routing: Spannbaum

- Zunächst wird ein Spannbaum angelegt
- Dann können beliebige Knoten die Daten entlang dieses Baumes verteilen



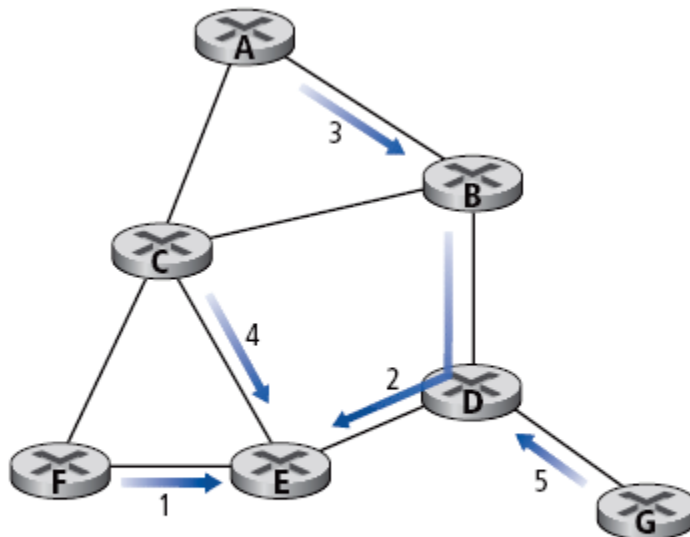
a Ein von A initiiertes Broadcast



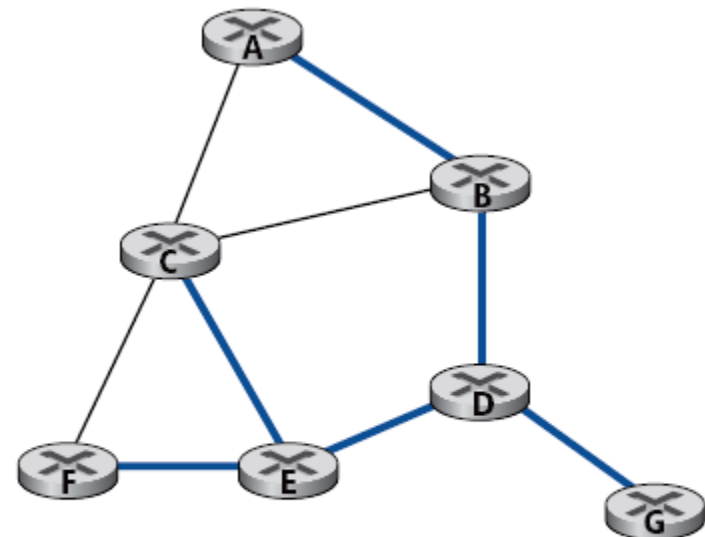
b Ein von D initiiertes Broadcast

4.7 Broadcast-Routing: Duplikation im Inneren des Netzwerks

- Wahl eines Zentrums (im konkreten Beispiel: Knoten E)
- Jeder Knoten sendet per Unicast eine Join-Nachricht in Richtung des Zentrums
 - Die Nachricht wird weitergeleitet, bis sie zu einem Knoten kommt, der schon Bestandteil des Spannbaumes ist



a Schrittweise Konstruktion eines Spannbaumes

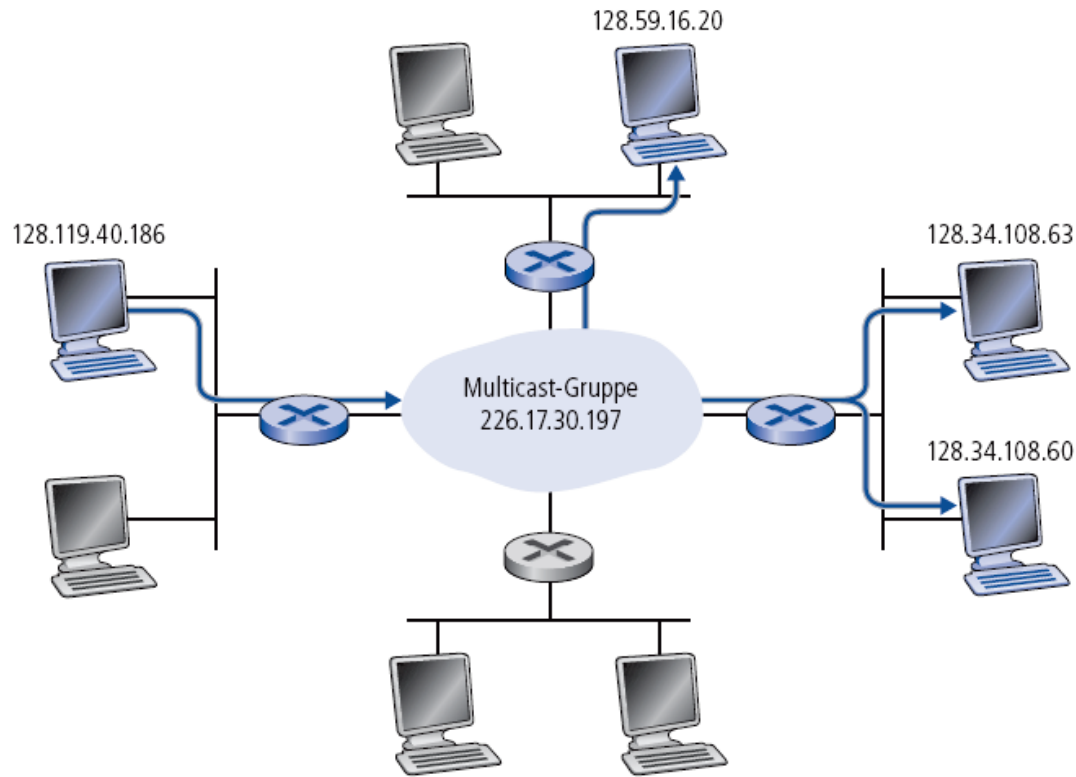


b Fertiger Spannbaum


4.7 IP-Multicast-Konzept

- Ein Sender schickt Pakete an eine Multicast-Adresse:
224.0.0.0 – 239.255.255.255.
- Multicast-Adressen bezeichnen eine (dynamische) Gruppe von Empfängern und sagen nichts darüber aus, wo diese Empfänger zu finden sind!
 - Dies bezeichnet man auch als **Adressindirektion**
- Ein Empfänger teilt den lokalen Routern mit, dass er die Pakete einer Multicast-Adresse empfangen möchte
- Multicast-fähige Router arbeiten mithilfe von Multicast-Routing-Protokollen zusammen, um die Pakete effizient vom Sender zu allen Empfängern zu befördern
- IP Multicast ist anonym, d.h., ein Sender kennt die Empfänger nicht

4.7 IP-Multicast Beispiel



Legende:

 Router mit angeschlossenem Gruppenmitglied

 Router ohne angeschlossenes Gruppenmitglied

4.7 Multicast-Unterstützung im Endsystem

Problem: Wie erfährt ein lokaler Router, dass sich ein Empfänger für eine gewisse Multicast-Gruppe in einem angeschlossenen Subnetz befindet?

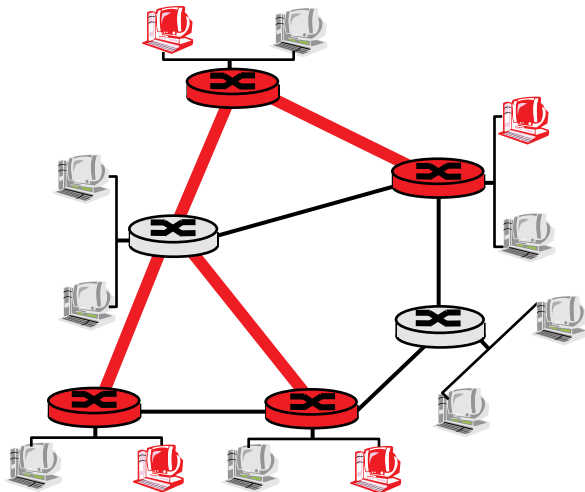
→ Nur wenn er diese Information besitzt, kann der Router mit anderen Routern zusammenarbeiten, um den Empfänger mit den gewünschten Daten zu versorgen

Lösung: Es gibt ein spezielles Protokoll, mit dem Empfänger signalisieren, dass sie den Empfang der Daten wünschen, die an eine bestimmte Multicast-Adresse gesendet werden: Internet Group Management Protocol (IGMP)

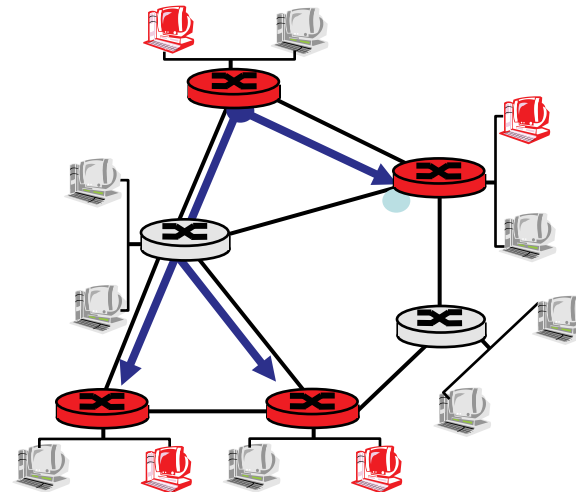
4.7 Multicast-Routing: Ziele

Finde einen Baum (oder mehrere Bäume), welcher die Router verbindet, die Gruppenmitglieder in ihrem lokalen Netzwerk haben

- **Gemeinsam genutzter Baum:** alle Sender verwenden denselben Baum
- **Quellenspezifischer Baum:** für jeden Sender ein eigener Baum



Gemeinsam genutzter Baum



Quellenspezifischer Baum

4.7 Multicast-Routing: Ansätze zur Konstruktion der Bäume

Konkrete Möglichkeiten:

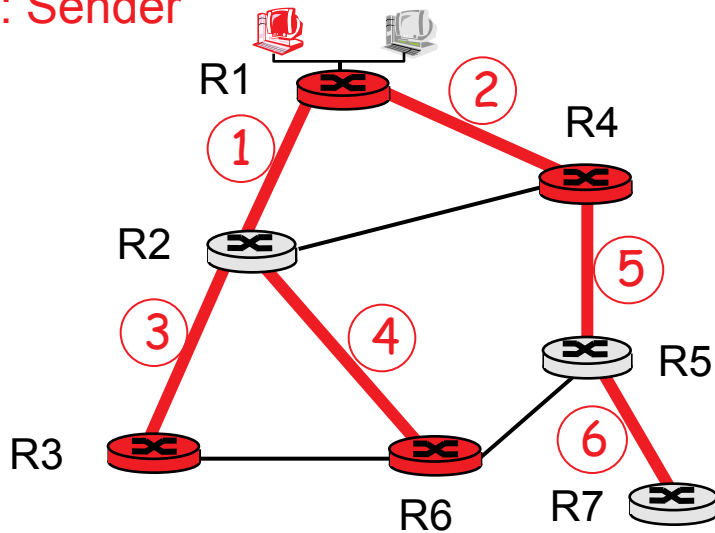
- **Quellenspezifische Bäume** – ein Baum pro Sender
 - Bäume mit kürzesten Pfaden
 - Reverse Path Forwarding
- **Gemeinsam genutzte Bäume** – alle Sender verwenden den gleichen Baum
 - Minimale Spannbäume (Steiner)
 - Zentrumsbasierte Spannbäume

... wir betrachten zunächst die grundsätzlichen Ansätze und dann erst konkrete Protokolle

4.7 Multicast-Routing: Bäume mit kürzesten Pfaden

- Multicast-Baum: Baum der kürzesten Pfade vom Sender zu allen Empfängern
 - Dijkstras Algorithmus

S: Sender



Legende



Router mit Empfängern
im lokalen Netzwerk

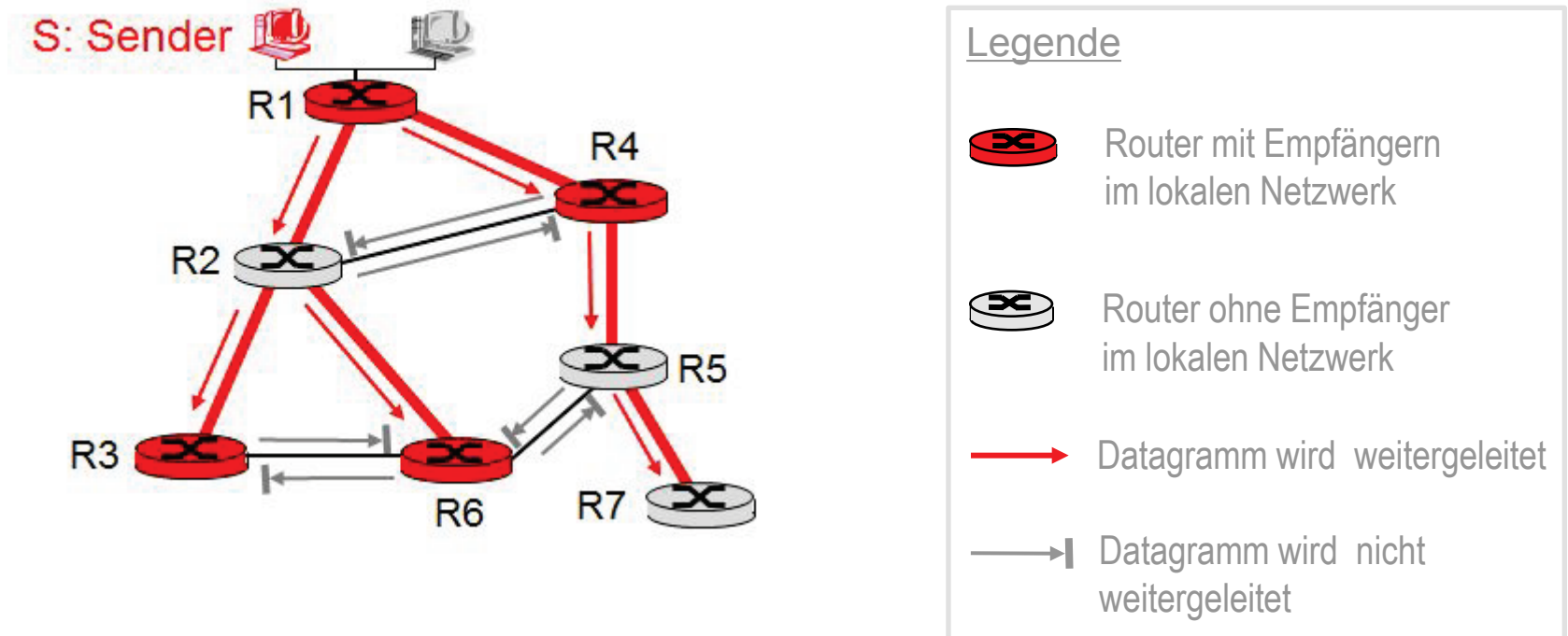


Router ohne Empfänger
im lokalen Netzwerk



Link, der Teil des Baumes ist;
die Nummer gibt an, in
welchem Schritt der Link zum
Baum hinzugefügt wurde

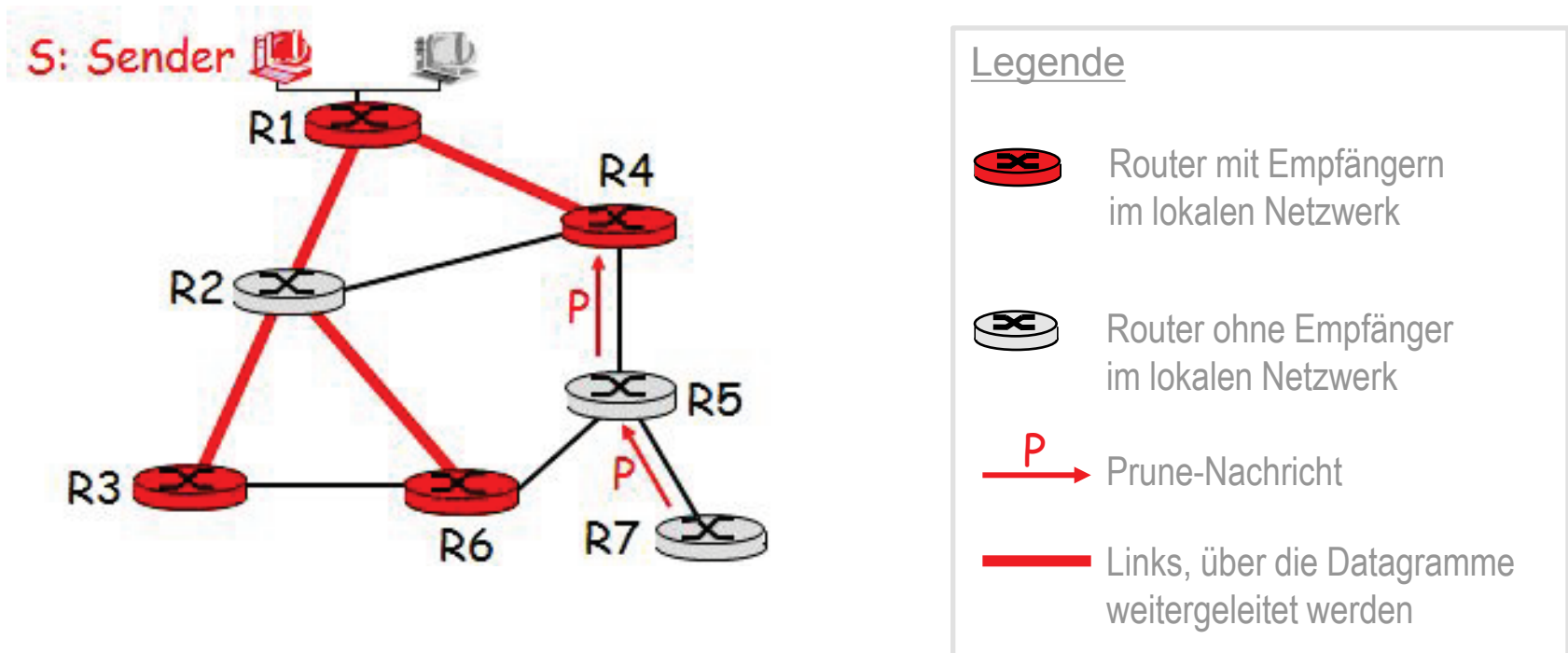
4.7 Multicast-Routing: Reverse Path Forwarding - Beispiel



→ Dies ergibt einen senderspezifischen Baum mit kürzesten Pfaden in der Rückrichtung

4.7 Multicast-Routing: Reverse Path Forwarding - Pruning

- Der Baum kann Teilbäume ohne Empfänger enthalten
 - Es ist nicht notwendig, Datagramme in diese Teilbäume weiterzuleiten
 - **“Prune”-Nachricht** wird in Richtung der Wurzel gesendet, um diese Teilbäume abzuschneiden



4.7 Multicast-Routing: Gemeinsam genutzte Bäume – Steiner-Bäume

Steiner-Baum: Baum mit minimalen Kosten, der alle Knoten verbindet, in deren lokalen Netzen Gruppenmitglieder vorhanden sind

- Konstruktion ist NP-hart
- Es existieren aber sehr gute Heuristiken
- In der Praxis nicht verwendet:
 - Rechenaufwand
 - Müssen jedes Mal neu berechnet werden, wenn ein Knoten hinzukommt oder wegfällt

4.7 Multicast-Routing: Zentrumsbasierte Bäume

- Ein einziger Baum wird von allen Sendern verwendet
- Ein Router ist das Zentrum des Baumes
- Um beizutreten:
 - Lokaler Router sendet eine Join-Nachricht an das Zentrum
 - Die Join-Nachricht wird auf dem Weg zum Zentrum von den dazwischenliegenden Routern untersucht
 - Die Join-Nachricht trifft entweder auf einen Router, der schon im Baum ist, oder sie kommt am Zentrum selbst an
 - Der Pfad, welcher von der die Join-Nachricht zurückgelegt wurde, wird zum neuen Ast im Multicast-Baum

4.7 Multicasting-Routing im Internet: DVMRP

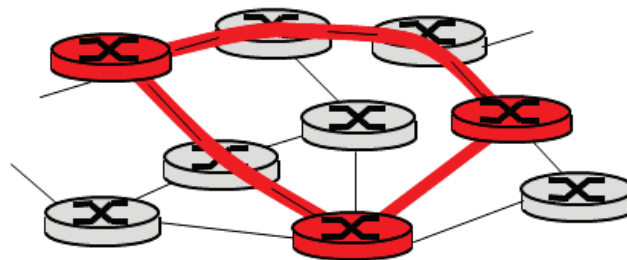
- DVMRP: Distance Vector Multicast Routing Protocol, [[RFC 1075](#)]
- *Flooding and Pruning*: Reverse Path Forwarding, quellenspezifischer Baum
 - RPF-Baum basiert auf einer eigenen (Unicast-)Routing-Tabelle, welche durch die Kommunikation zwischen DVMRP-Routern entsteht
 - Keine Annahmen bezüglich der parallel verwendeten Unicast-Routing-Protokolle, wie z.B. OSPF
 - Initial werden Datagramme per RPF geflutet
 - Wenn ein Router keine Daten für eine Gruppe haben möchte und selbst die Datagramme nicht zu anderen Routern weiterleiten muss: Senden einer Prune-Nachricht in Richtung der Wurzel des Baumes

4.7 Multicasting-Routing im Internet: DVMRP

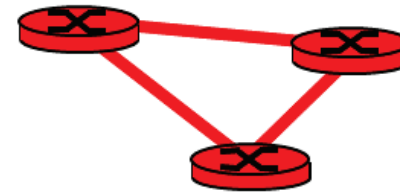
- *Soft State*: DVMRP “vergisst” periodisch, dass Teilbäume abgeschnitten wurden (z.B. einmal pro Minute):
 - Daten fließen wieder in die vorher abgeschnittenen Teilbäume
 - Erneutes Senden einer Prune-Nachricht, wenn die Daten nach wie vor unerwünscht sind
- Sonstiges
 - Häufig in kommerziellen Routern implementiert
 - DVMRP wurde im experimentellen *Mbone* verwendet

4.7 Multicasting-Routing im Internet: DVMRP

Frage: Wie kann man “Multicast-Inseln” über Router hinweg miteinander verbinden, wenn diese Router kein Multicast verstehen?



Reale Topologie



Logische Sicht

- Analog zur Einführung von IPv6:
 - Multicast-Datagramme werden in normale IPv4-Unicast-Datagramme verpackt
 - Diese werden dann von einem Multicast-fähigen Router über das normale Netzwerk an einen anderen Multicast-fähigen Router gesendet
 - Der empfangende Router packt das Datagramm aus und erhält das Multicast-Datagramm

4.7 Protocol Independent Multicast (PIM)

→ Hängt nicht von einem speziellen Unicast-Routing-Protokoll ab

Zwei Varianten für unterschiedliche Szenarien:

- **Dense:**

- Gruppenmitglieder liegen dicht beieinander (d.h. ein sehr großer Teil der Router im Netzwerk möchte die Datagramme empfangen)
- Bandbreite steht in großem Umfang zur Verfügung

- **Sparse:**

- Nur ein kleiner Anteil der Router im Netzwerk möchte die Datagramme empfangen
- Gruppenmitglieder sind weit verteilt
- Bandbreite ist ein Engpass

4.7 Protocol Independent Multicast (PIM)

Auswirkung der Unterscheidung:

- **Dense**

- Es wird angenommen, dass alle Router die Daten bekommen wollen, es sei denn, sie schicken ein explizites Prune
- Der Baum entsteht automatisch durch das Versenden der Datagramme
- Mechanismus: RPF
- Bandbreite und Mehraufwand in Routern sind nicht von großer Bedeutung

- **Sparse:**

- Solange ein Router nicht explizit beitrifft bzw. für das Weiterleiten an andere Router benötigt wird, erhält er auch keine Daten
- Empfängergetriebene Konstruktion des Baumes
- Mechanismus: zentrumsbasierte Konstruktion
- Bandbreite und Mehraufwand in Routern werden minimiert

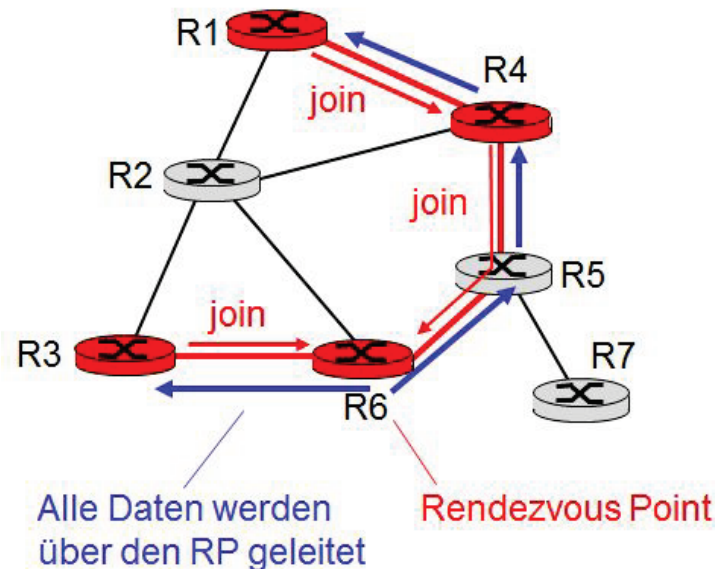
4.7 PIM – Dense Mode

RPF mit Pruning, analog zu DVMRP, aber:

- Das verwendete Unicast-Routing-Protokoll liefert die notwendigen RPF-Informationen

4.7 PIM – Sparse Mode

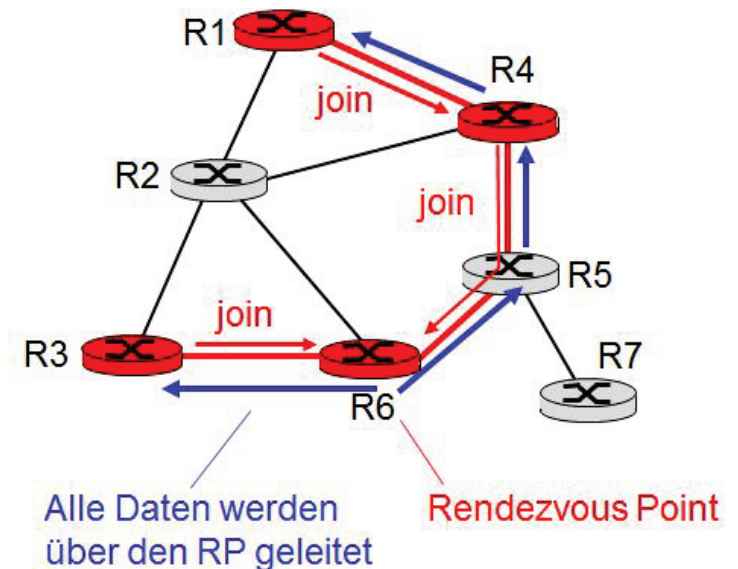
- Zentrumsbasierter Ansatz
- Router sendet Join-Nachrichten an einen Rendezvous Point (RP)
 - Router, die einen Join weiterleiten, merken sich dies und leiten dann den Join weiter



4.7 PIM – Sparse Mode

Sender:

- Schicken ihre Daten per Unicast an den RP
- Der RP verbreitet die Daten dann im Baum
- Der RP kann eine Stop-Nachricht an den Sender schicken, wenn es keine Empfänger gibt



Kapitel 5 – Sicherungsschicht und lokale Netzwerke

- 5.1 Einleitung und Dienste
- 5.2 Fehlererkennung und -korrektur
- 5.3 Protokolle für den Mehrfachzugriff
- 5.4 Adressierung auf der Sicherungsschicht
- 5.5 Ethernet
- 5.6 Switches auf der Sicherungsschicht
- 5.7 PPP
- 5.8 Link-Virtualisierung: ATM, MPLS
- 5.9 Rechenzentren-Netzwerke

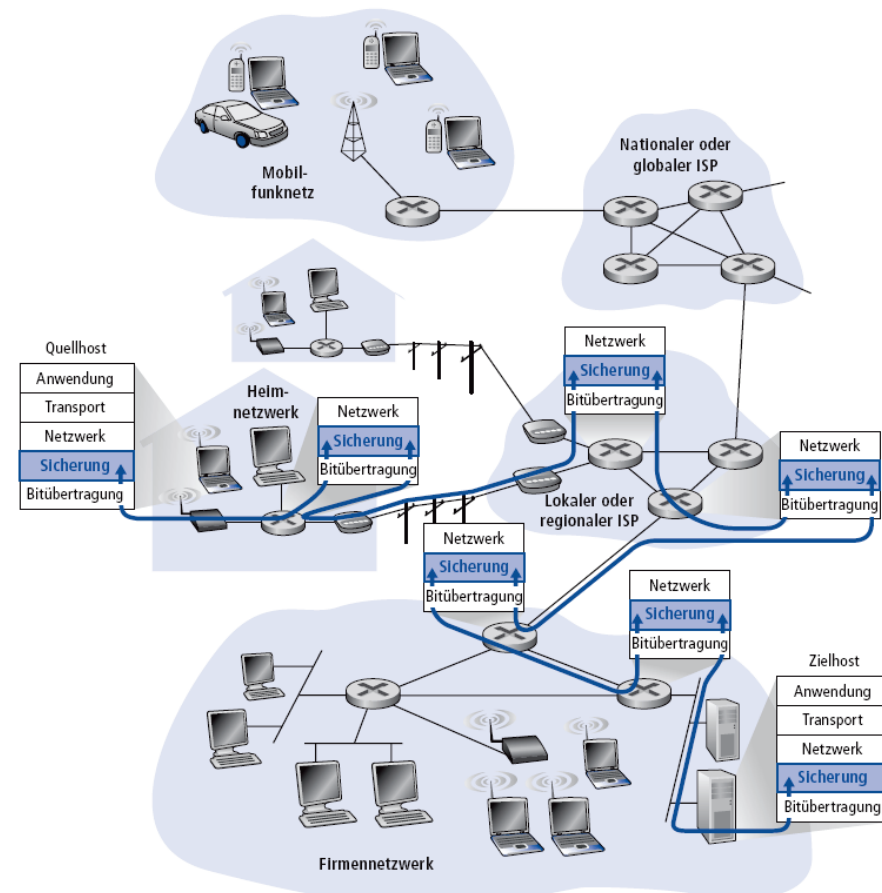
5.1 Einleitung und Dienste

5.1 Sicherungsschicht - Einleitung

Verwendete Terminologie:

- Hosts und Router sind **Knoten**
- Kommunikationskanäle auf dem Weg vom Sender zum Empfänger sind **Links**
 - Kabelgebundene Links
 - Drahtlose Links
 - LANs
- Ein Paket der Sicherungsschicht nennt man **Rahmen** (engl. Frame)
 - Ein Rahmen enthält üblicherweise ein Datagramm der Netzwerkschicht

Die **Sicherungsschicht** (link layer) hat die Aufgabe, Rahmen von einem Knoten über einen Link zu einem direkt benachbarten Knoten zu transportieren.



5.1 Sicherungsschicht - Einordnung

- Ein Datagramm wird von verschiedenen Protokollen der Sicherungsschicht über verschiedene Links transportiert:
 - *Beispiel:* Ethernet auf dem ersten Link, dann Frame Relay, dann IEEE 802.11 WLAN
- Jedes dieser Protokolle kann unterschiedliche Dienste anbieten
 - Diese Protokolle können z.B. zuverlässige oder nur unzuverlässige Übertragung anbieten

Analogie

- Reise von Princeton nach Lausanne
 - Taxi: Princeton zum JFK-Flughafen
 - Flugzeug: JFK-Flughafen nach Genf
 - Zug: Genf nach Lausanne
- Tourist = **Datagramm**
- Reiseabschnitt = **Link**
- Reisebüro = Routing-Protokoll
- Art des Transportes = **Protokoll der Sicherungsschicht**



5.1 Dienste der Sicherungsschicht

- Rahmenbildung und Zugriff auf den Link:
 - Verpacken eines Datagramms in einen Rahmen, Hinzufügen von Header und Trailer
 - Zugriff auf den Kanal (schwierig, wenn dieser von mehreren Knoten verwendet wird)
 - “MAC”-Adressen (Medium Access Control) werden im Header von Rahmen verwendet, um Sender und Empfänger zu kennzeichnen
→ **Verschieden von IP-Adressen!**
- Zuverlässige Datenübertragung zwischen benachbarten Knoten:
 - Seltener Einsatz, wenn der Link sehr wenige Bitfehler verursacht (Glasfaser, Kupferkabel, usw.)
 - Drahtlose Links: hohe Bitfehlerrate

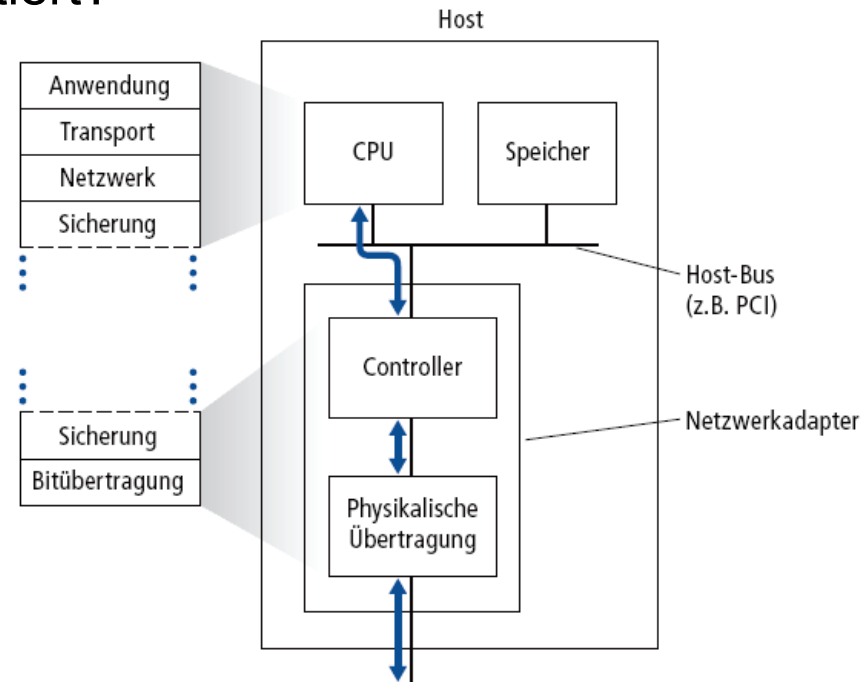
5.1 Dienste der Sicherungsschicht

- Flusskontrolle:
 - Anpassen der Sendegeschwindigkeit an den Empfänger
- Fehlererkennung:
 - Fehler entstehen beispielsweise durch Abschwächen des Signals auf der Leitung und durch Rauschen
 - Der Empfänger sollte Fehler erkennen können, dann:
 - Neuübertragung auslösen...
 - ...oder Rahmen verwerfen
- Fehlerkorrektur:
 - Der Empfänger erkennt und korrigiert Bitfehler, ohne eine Neuübertragung anzufordern
- Halbduplex und Vollduplex:
 - Bei Halbduplex können die Knoten an beiden Enden der Leitung übertragen – jedoch nicht gleichzeitig

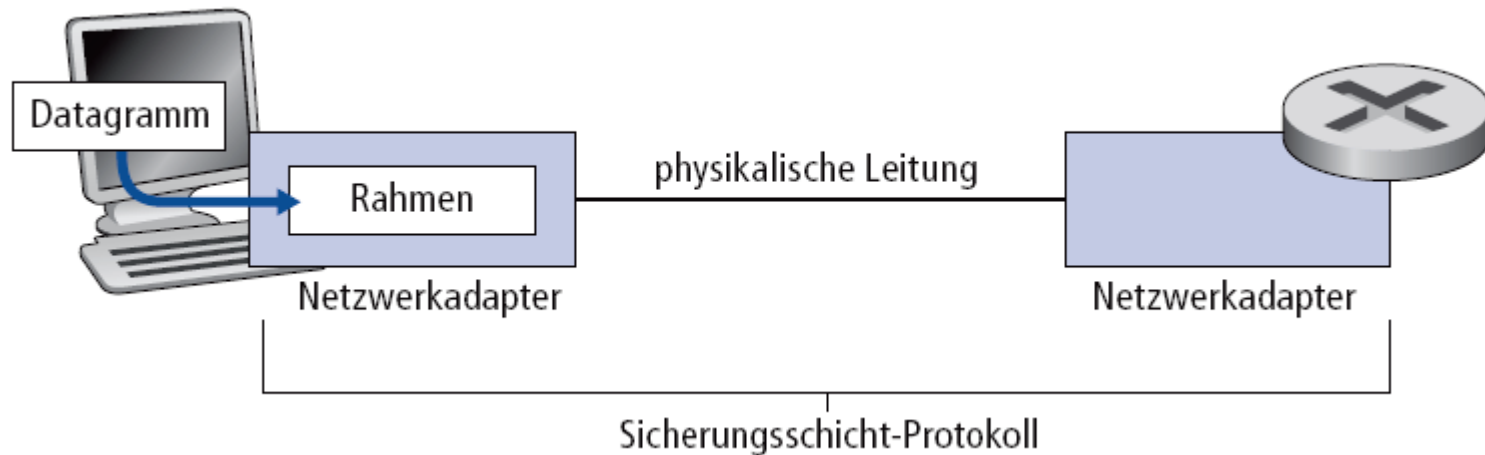
5.1 Sicherungsschicht - Einordnung

Wo ist die Sicherungsschicht implementiert?

- In jedem Host, in jedem Router
- Die Sicherungsschicht ist im Netzwerkadapter (Netzwerkkarte) implementiert
 - Ethernet-Netzwerkkarte, 802.11 WLAN-Karte
 - Enthält Sicherungsschicht und Physikalische Schicht
- An den Systembus des Hosts/Routers angeschlossen
- Kombination von Hardware, Software, Firmware



5.1 Kommunikation zwischen Netzwerkadaptern



Sender:

- Verpacken von Datagrammen in Rahmen
- Hinzufügen von Bits für die Fehlererkennung, die zuverlässige Datenübertragung, Flusskontrolle, usw.

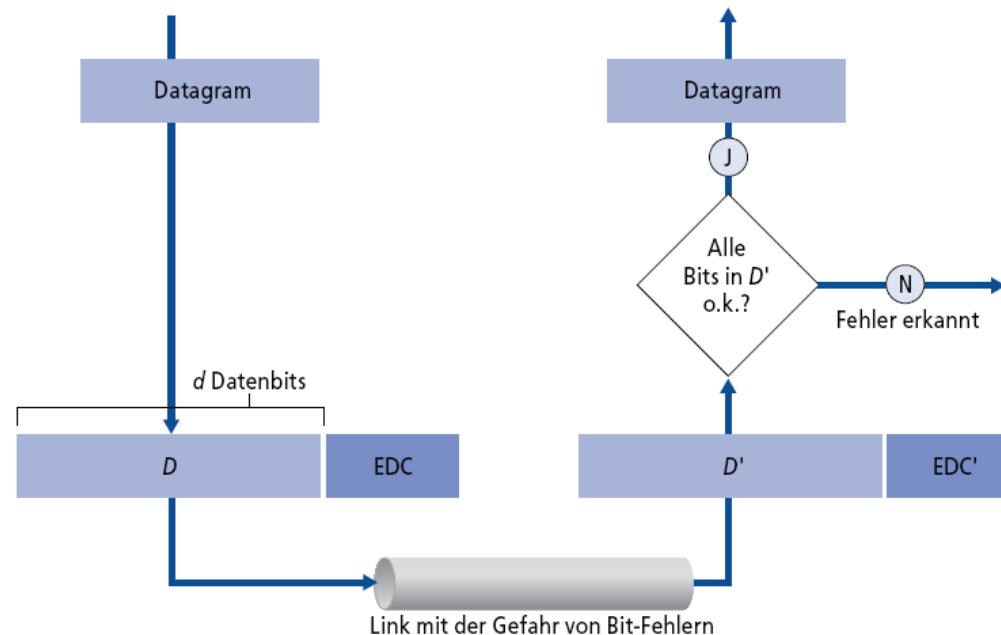
Empfänger

- Überprüfen auf Bitfehler, Flusskontrolle, usw.
- Extrahieren des Datagramms, Ausliefern an die Netzwerkschicht

5.2 Fehlererkennung und -korrektur

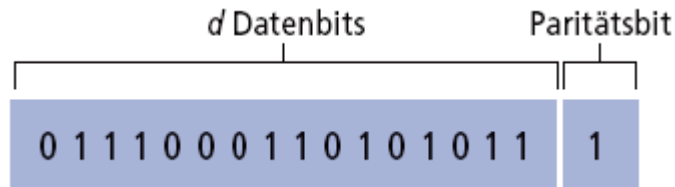
5.2 Fehlererkennung

- **EDC** = Error Detection and Correction Bits (Redundanz)
- **D** = Daten, die durch EDC geschützt sind (kann die Header-Felder einschließen)
- Fehlererkennung ist nicht 100% zuverlässig!
 - Ein Sicherungsschichtprotokoll kann Fehler übersehen (*sehr selten!*)
 - Mehr EDC-Bits führen zu besseren Erkennungsraten

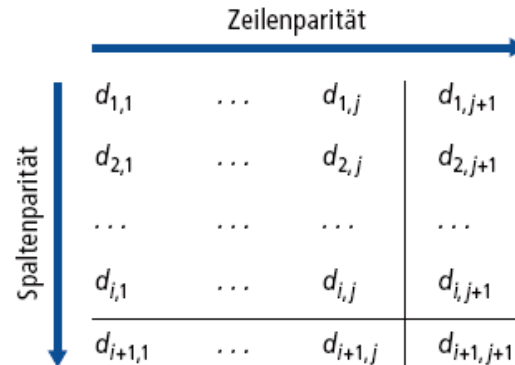


5.2 Paritätsprüfung

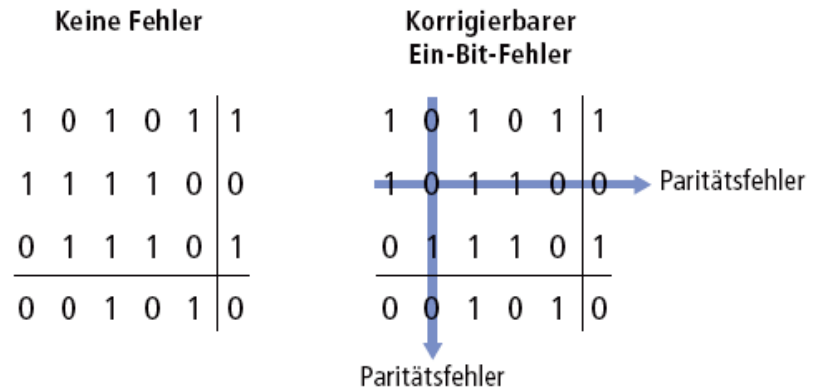
Ein-Bit-Parität:
Erkennt Ein-Bit-Fehler



Zweidimensionale Parität:
Erkennt und korrigiert Ein-Bit-Fehler



Gerade Parität!



5.2 Internetprüfsumme

Ziel: Erkennen von Fehlern in übertragenen Segmenten auf der Transportschicht

Sender:

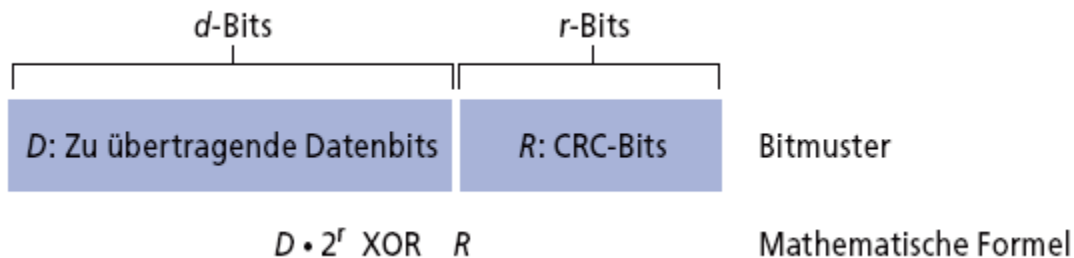
- Betrachte das Segment als eine Folge von 16-Bit-Integerwerten
- Prüfsumme: Addition (im 1er Komplement) der Werte
- Sender schreibt das Ergebnis in das UDP-Prüfsummenfeld

Empfänger:

- Berechne die Prüfsumme
- Passt diese zum Wert im Prüfsummenfeld:
 - Nein – Fehler erkannt
 - Ja – kein Fehler erkannt. Aber es könnten dennoch Fehler vorliegen!

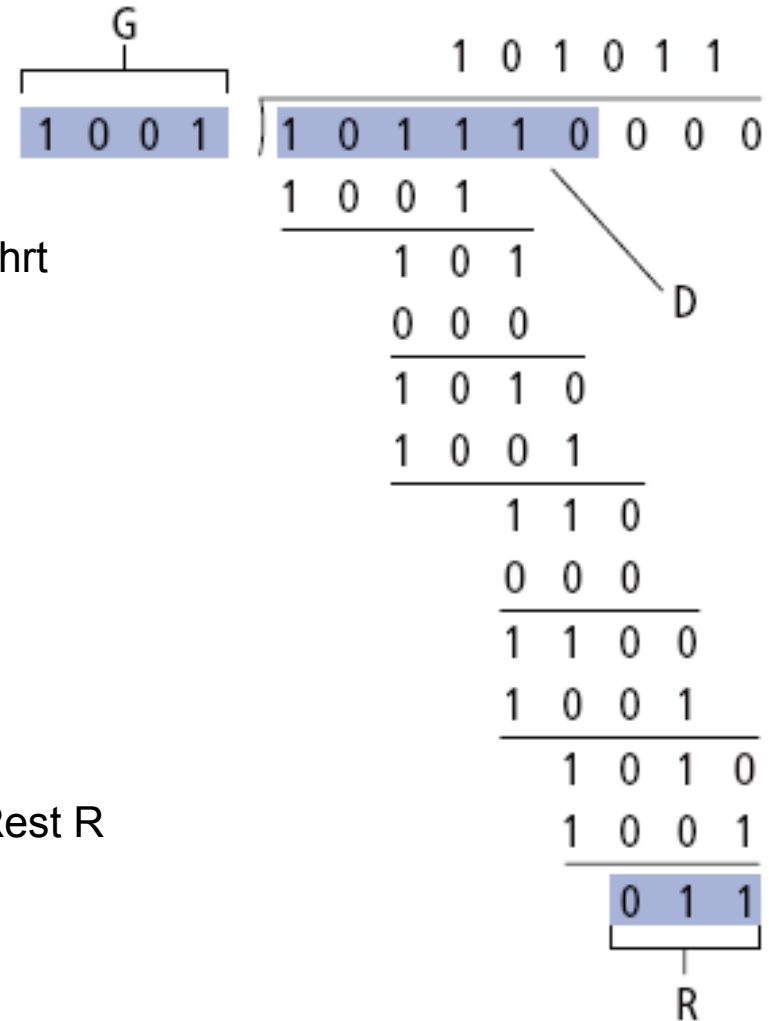
5.2 Bildung von Prüfsummen: Cyclic Redundancy Check (CRC)

- Betrachte die Datenbits (**D**) als eine binäre Zahl
- Wähle ein Bitmuster der Länge $r+1$ (Generator, **G**)
- Ziel: Wähle r CRC-Bits (**R**) so, dass gilt:
 - $\langle D, R \rangle$ ist modulo 2 durch G ohne Rest teilbar
 - Empfänger kennt G und teilt das empfangene $\langle D', R' \rangle$ durch G . Wenn es einen Rest gibt: Fehler erkannt!
- **Kann alle Burst-Fehler erkennen, die kürzer als $r+1$ Bit sind**
- In der Praxis weit verbreitet (IEEE 802.11 WLAN, ATM)



5.2 CRC Beispiel

- Vorbemerkung:
 - Alle Operationen werden modulo 2 durchgeführt
 - Addition und Subtraktion entsprechen der Verknüpfung mit XOR
 - Es gibt keinen Übertrag
- Es soll ein R gefunden werden:
 - $D \cdot 2^r \text{ XOR } R = nG$
- Dies ist äquivalent zu:
 - $D \cdot 2^r = nG \text{ XOR } R$
- Dies ist äquivalent zu:
 - Wenn wir $D \cdot 2^r$ durch G teilen, entspricht der Rest R



$$R = \text{Rest von } \left[\frac{D \cdot 2^r}{G} \right]$$

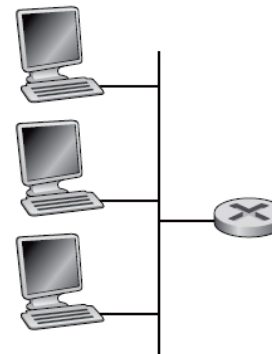
5.3 Protokolle für den Mehrfachzugriff

5.3 Links mit Mehrfachzugriff

Zwei Arten von “Links”:

- **Punkt-zu-Punkt**
 - Einwahlverbindungen
 - Verbindung zwischen Ethernet Switch und Host
- **Broadcast** (gemeinsam verwendetes Medium)
 - Ursprüngliches Ethernet
 - Upstream bei HFC (Internetzugang über das Fernsehkabelnetz)
 - IEEE 802.11 WLAN

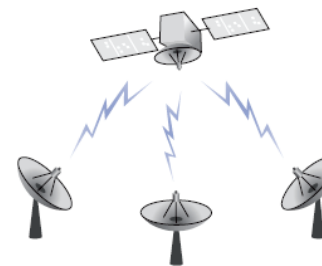
Gemeinsam genutzte Leitung
(z.B. Ethernet)



Gemeinsam genutzter Funkkanal
(z.B. WLAN)



Satellit



Cocktailparty



5.3 Protokolle für den Mehrfachzugriff

- Ein gemeinsam genutzter Broadcast-Kanal
- Mehrere gleichzeitige Übertragungen verschiedener Knoten:
 - **Kollision**, wenn ein Knoten mehrere Signale zur gleichen Zeit empfängt
 - Dadurch werden die Signale unbrauchbar

Protokolle für den Mehrfachzugriff (MAC-Protokolle)

- Verteilte Algorithmen, die bestimmen, wie sich die Knoten den Kanal teilen
- Bestimmen, wer wann senden darf
- Die dazu notwendige Kommunikation muss wiederum über den Broadcast-Kanal selbst abgewickelt werden
 - Kein zusätzlicher Kanal für die Koordination

5.3 Protokolle für den Mehrfachzugriff

Anforderungen an das perfekte Protokoll für den Mehrfachzugriff:

→ Gegeben: Ein Broadcast-Kanal mit R bit/s

1. Wenn nur ein Knoten übertragen möchte, dann kann er mit der Rate R senden
2. Wenn M Knoten übertragen möchten, dann kann jeder mit der Rate R/M senden
3. Dezentral:
 - Kein spezieller Knoten zur Koordination der Übertragungen
 - Keine Synchronisation von Uhren oder Zeitschlitz
4. Einfach