Rechnernetze I

SoSe 2020

Roland Wismüller
Universität Siegen
roland.wismueller@uni-siegen.de
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: 12. März 2020

Rechnernetze I

SoSe 2020

6 Routing

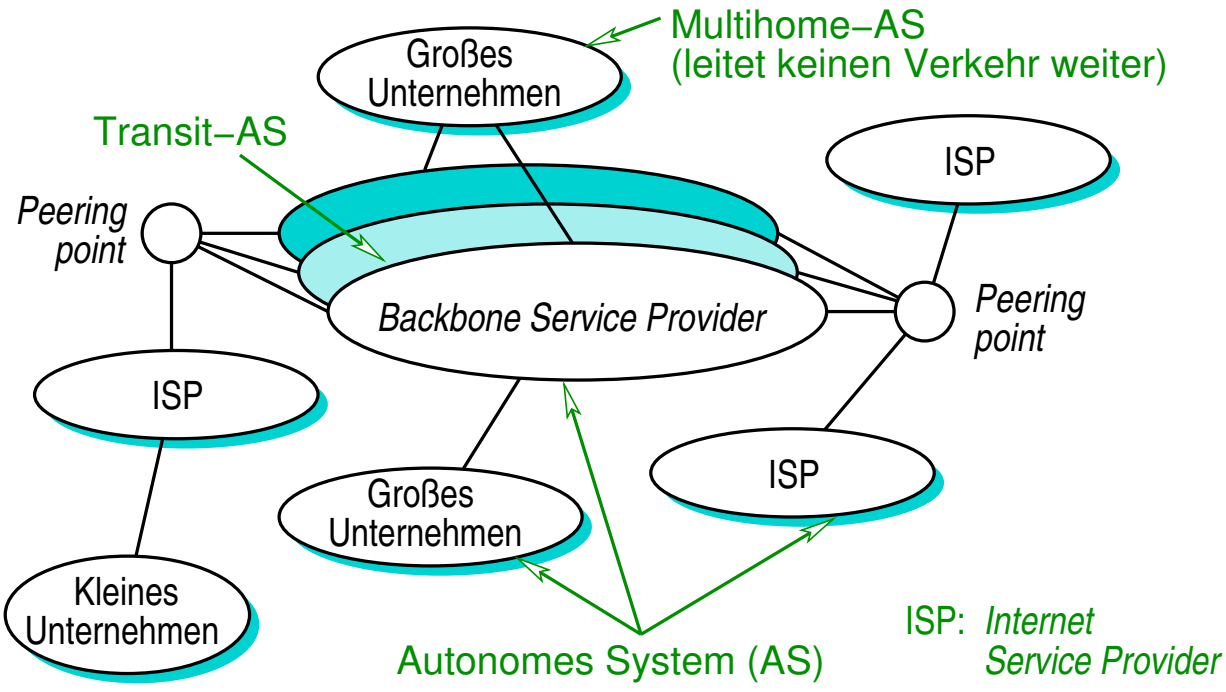
Inhalt

- Einführung
- Routing innerhalb einer Domain
  - *Distance Vector Routing* (RIP, EIGRP)
  - *Link State Routing* (OSPF)
- Interdomain-Routing
  - *Border Gateway Protocol* (BGP)

- Peterson, Kap. 4.2.1 – 4.2.4

6.1 Einführung

Heutige Struktur des Internet



Routing im Internet

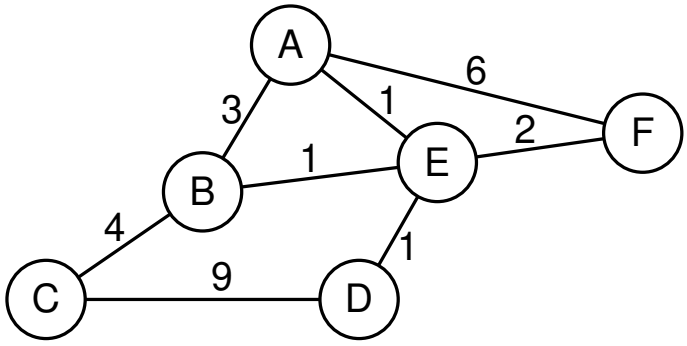
- Heute über 64.000 autonome Systeme (AS)
  - Backbones, Provider, Endbenutzer
  - Netzwerk-Adressen werden an AS zugewiesen
- Probleme: Skalierbarkeit, heterogene Administration
- Hierarchischer Ansatz: ***Routing Domains***
  - Routing innerhalb eines administrativen Bereichs (z.B. Campus, Unternehmen, Provider)
    - *Interior Gateway Protocols* (IGP)
      - z.B. RIP, OSPF, EIGRP
  - Interdomain Routing (zwischen Teilnetzen des Internet)
    - *Exterior Gateway Protocols* (EGP)
      - z.B. BGP (*Border Gateway Protocol*)

6.2 Routing innerhalb eines Bereichs

OSI: 3

Routing als Graph-Problem

- Knoten:
  - Router
  - (Hosts)
  - (Netzwerke)
- Kanten: Verbindungen
  - mit Kosten (Metrik) (symmetrisch oder asymmetrisch)



- Aufgabe des Routings:
  - finde Pfade mit geringsten Kosten zwischen allen Paaren von Knoten

Anmerkungen zu Folie 176:

- Im Graphen werden üblicherweise nur die Router mit den Verbindungen (d.h. Netzen) zwischen ihnen modelliert. An vielen Router sind i.d.R. natürlich noch ein oder mehrere lokale Netze angeschlossen, die im Graphen aber nicht dargestellt sind.
- Zur Bestimmung der Kosten einer Verbindung können typischerweise die Bandbreite, die Latenz und die Auslastung der Verbindung herangezogen werden. Man hat durch Erfahrung gelernt, daß der Wertebereich dabei relativ eng sein muß, damit der billigste Link nicht völlig überlastet wird. Die Bandbreite ist dabei wichtiger als die Latenz, die Auslastung sollte nur bei mittlerer bis hoher Last in die Kosten eingehen.

  Im einfachsten Fall können die Kosten auch für jede Verbindung gleich gewählt werden (i.d.R. als 1).

6.2 Routing innerhalb eines Bereichs ...

Statisches Routing

- Pfade werden manuell bestimmt und in Tabellen eingetragen
- Probleme:
  - Ausfall von Knoten / Verbindungen
  - neue Knoten / Verbindungen
  - dynamische Änderung der Verbindungskosten (Last)

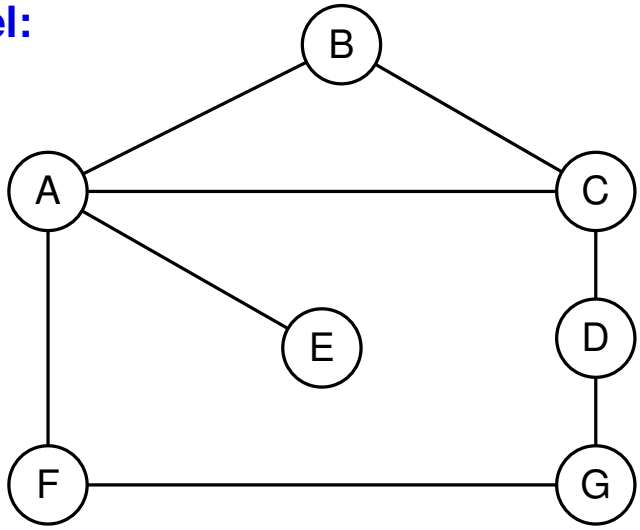
Dynamisches Routing

- Verteilte Algorithmen zum Aufbau der Tabellen
- Anforderungen:
  - schnelle Konvergenz / Skalierbarkeit
  - einfache Administration

6.2.1 *Distance Vector Routing*

- Router kennen nur Distanz und „Richtung“ (*Next Hop*) zum Ziel
  - nur diese Informationen werden (mit Nachbarn) ausgetauscht
- Verteilter Algorithmus zur Erstellung der Routing-Tabellen
  - typisch: Bellman-Ford-Algorithmus

Beispiel:

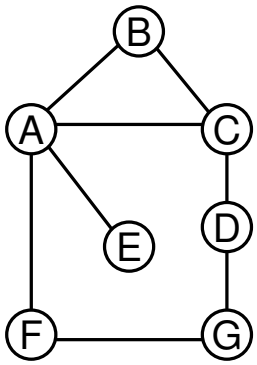


Vereinfachung:
alle Link–Kosten
seien 1

6.2.1 *Distance Vector Routing* ...

Beispiel: ...

- Initial besitzen die Router Information über die folgenden Distanzvektoren (verteilt!)



	Distanz zu Knoten						
Information bei	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Beispiel: ...

- Initiale Routing-Tabelle von A:
  - A kennt nur seine direkten Nachbarn

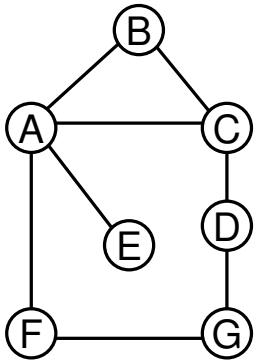
Ziel	Kosten	NextHop
B	1	B
C	1	C
D	∞	––
E	1	E
F	1	F
G	∞	––

Vorgehensweise

- Nachrichtenaustausch zur Erstellung der Routing-Tabellen:
  - Router senden ihren Distanzvektor an alle direkten Nachbarn
  - bessere Routen werden in den eigenen Distanzvektor (und die Routing-Tabelle) übernommen
    - aber auch schlechtere vom Next Hop
- Nach mehreren Runden des Nachrichtenaustauschs konvergieren die Distanzvektoren ... jedenfalls meistens !!
- Der Nachrichtenaustausch erfolgt
  - periodisch (typ. alle 30 s)
  - bei Änderung des Distanzvektors eines Knotens
    - z.B. Ausfall einer Verbindung, neue Verbindung
  - auf Anfrage (z.B. bei Neustart eines Routers)

Beispiel ...

➥ Distanzvektoren nach dem Konvergieren des Verfahrens:

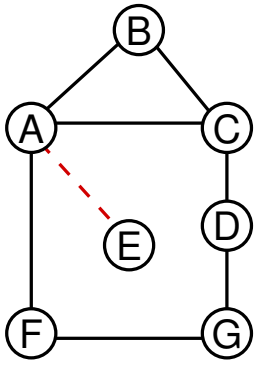


	Distanz zu Knoten						
Information bei	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

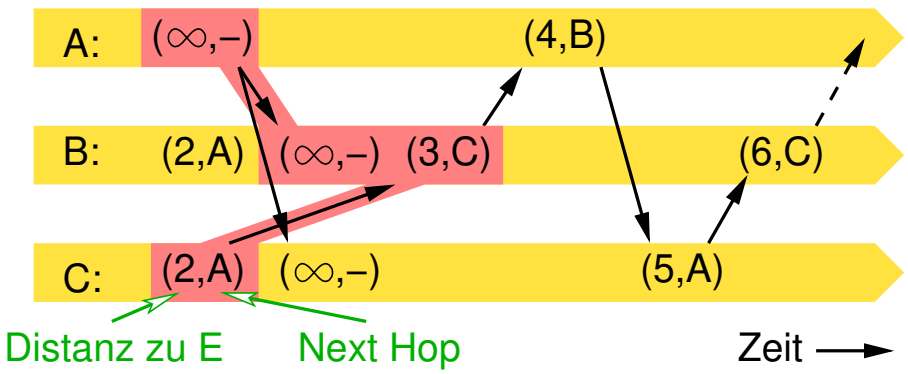
(Animierte Folie)

Problem des Algorithmus: *Count-to-Infinity*

➥ Beispiel: A erkennt Ausfall der Verbindung zu E



Zeitliche Entwicklung der Distanz–Vektoren zu E:



➥ Keine wirklich gute, allgemeine Lösung des Problems

➥ pragmatisch: beschränke ∞ auf den Wert 16

Anmerkungen zu Folie 183:

- Neben dem *Count-to-Infinity*-Problem können beim Distanzvektor-Routing auch Routing-Schleifen entstehen. Im Beispiel entsteht während des Hochzählens eine Routing-Schleife (C→A→B→C), in der Pakete kreisen, bis sie wegen TTL=0 verworfen werden.
- Weitere Maßnahmen, um die genannten Probleme zu verhindern, sind:
  - *Split Horizon*: Information, die über eine bestimmte Schnittstelle gelernt wurde, wird nicht wieder an dieselbe Schnittstelle weitergegeben. Dies kann Routing-Schleifen verhindern, sofern das Netzwerk zyklenfrei ist (was im Beispiel nicht der Fall ist).
  - *Route Poisoning*: Nicht mehr verfügbare Routen werden im Distanzvektor nicht gelöscht, sondern mit einer Metrik von 16 (d.h. ∞) weitergegeben. Im Beispiel auf der Folie wird dies bereits so gemacht, kann in diesem Fall das Problem aber nicht verhindern.
  - *Holddown*-Timer: wenn eine Route ausfällt, wird diese zunächst nur markiert. Bis der Timer abläuft (typ. 180s), werden empfangene Aktualisierungen für die Route ignoriert, die die gleiche oder eine schlechtere Metrik haben wie die Metrik vor dem Ausfall. Im Beispiel würde B die falsche Aktualisierung von C ignorieren, wenn sie vor Ablauf des Timers eintrifft.

6.2.1 *Distance Vector Routing* ...

RIP (*Routing Information Protocol*)

- Einfaches *Distance Vector Routing* Protokoll
- Internet-Standard
- Alle Link-Kosten sind 1, d.h. Distanz = *Hop Count*
- Drei Versionen:
  - RIPv1: leitet keine Präfixlänge weiter
    - Subnetting nur möglich, wenn alle Subnetze des klassenbehafteten Netzes dieselbe Größe haben
    - d.h. Subnetzmaske ist global
  - RIPv2: ermöglicht klassenloses Routing
  - RIPng: unterstützt IPv6

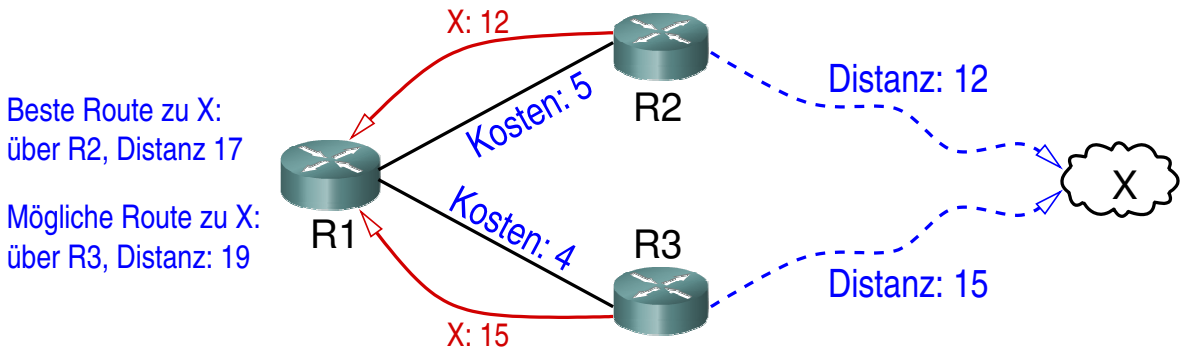
Anmerkungen zu Folie 184:

- RIP-Pakete werden mittels UDP übertragen (in RIPv1 als Broadcast, in späteren Versionen als Multicast).
- RIPv1 ist im Detail im RFC 1058 spezifiziert, RIPv2 im RFC 2453, RIPng im RFC 2080.

6.2.1 *Distance Vector Routing* ...

EIGRP (*Enhanced Interior Gateway Routing Protocol*)

- Erweitertes *Distance Vector Routing* Protokoll
- Cisco-proprietär, seit 2013 offener Internet-Standard
- Link-Kosten berücksichtigen Bandbreite und Latenz
- Unterstützt IPv4, IPv6 und andere Schicht-3-Protokolle
- Updates nur bei Änderungen, kein *Count-to-Infinity*
- Behält alle Routen, nicht nur die beste



Anmerkungen zu Folie 185:

- Im Gegensatz zu RIP behält EIGRP von den eingehenden Distanzvektoren nicht nur den besten, der zum kürzesten Weg zum Ziel gehört, sonden alle, die zu garantiert zyklenfreien Wegen zum Ziel gehören. Um die Zyklenfreiheit zu garantieren, vergleicht EIGRP beim Empfang eines Distanzvektor in einem Router R die darin gemeldete Distanz des Senders S zum Ziel mit der (aktuell bekannten) eigenen Distanz zum Ziel. Hat S eine kleinere Distanz, kann sein bester Weg nicht über R gehen, da die Link-Kosten nicht negativ werden können.

  Wenn im gezeichneten Beispiel R1 den Distanzvektor von R2 erhält (rote Nachricht mit Beschriftung „X: 12“), trägt er als beste Route zu X die über R2 ein, mit einer Distanz von 12+5=17. Beim Erhalt des Distanzvektors von R3 („X: 15“) stellt R1 fest, daß die von R3 gemeldete Distanz 15 kleiner als seine eigene (17) ist, und der Weg von R3 nach X daher nicht über R1 führen kann. R3 ist damit ein möglicher Nachfolger (*feasible successor*) von R1 auf dem Weg zu X. Bei Ausfall der Verbindung zwischen R1 und R2 kann R1 daher die Route in der Weiterleitungstabelle sofort so abändern, daß R3 der *Next Hop* wird. Er muß dazu nicht auf Routing-Nachrichten anderer Route warten.

- Um den Ausfall von Verbindungen zu erkennen, sendet EIGRP regelmäßige *Hello*-Pakete an alle Nachbarn.

- EIGRP-Pakete werden über ein eigenes, zuverlässiges Transportprotokoll (*Reliable Transport Protocol*, RTP) übertragen, das auf IP aufsetzt.

- Den *Informational* RFC zu EIGRP finden Sie unter der URL `https://tools.ietf.org/html/draft-savage-eigrp-00`.

6.2.2 *Link State Routing*

- Grund„problem“ beim *Distance Vector Routing*:
  - Router haben ausschließlich lokale Information
- ***Link State Routing:***
  - Router erhalten Information über die Struktur des gesamten Netzwerks
- Vorgehensweise:
  - Kennenlernen der direkten Nachbarn
    - einschließlich der Link-Kosten
  - Versenden von Link State Paketen an **alle** anderen Router (***Reliable Flooding***)
  - Berechnung der kürzesten Wege mit Dijkstra-Algorithmus

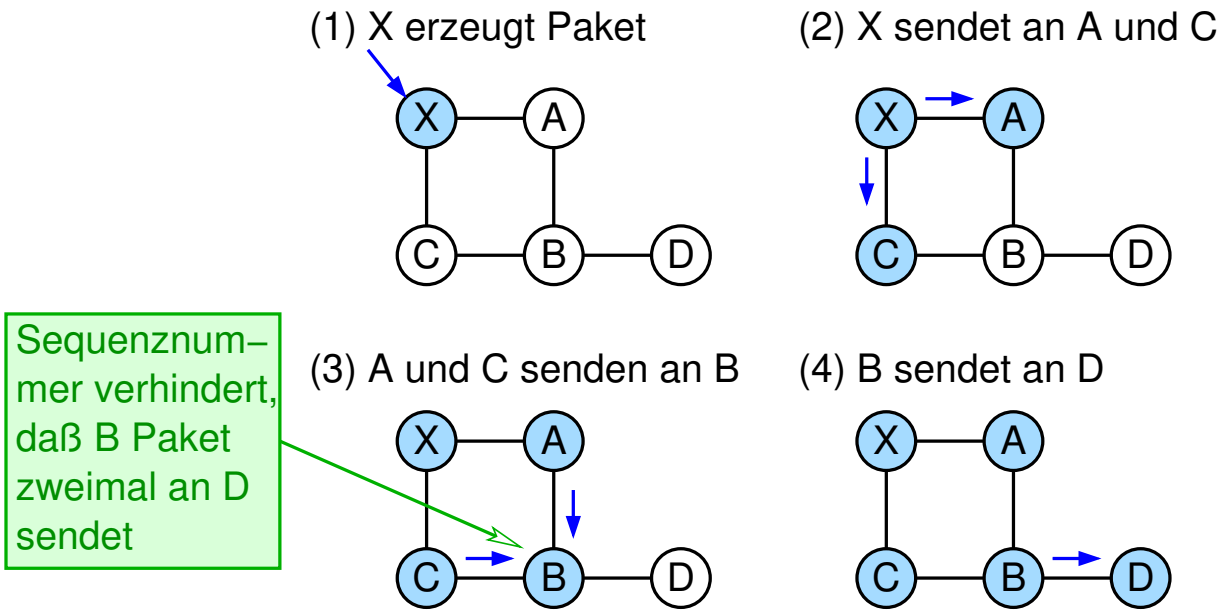
6.2.2 *Link State Routing* ...

Link State Pakete

- Inhalt eines Link State Pakets:
  - ID des erzeugenden Routers
  - Liste der direkten Nachbarn mit Link-Kosten
  - Sequenznummer
    - Paket nur weitergeleitet, wenn die Sequenznummer größer als die des letzten weitergeleiteten Pakets ist
  - *Time-to-Live* (TTL)
    - jeder Router dekrementiert TTL
    - bei TTL = 0 wird das Paket gelöscht
- Versenden der Link State Pakete
  - Periodisch ($\sim$ Stunden) oder bei Topologie-Änderungen

Reliable Flooding

- Flooding mit ACK und ggf. wiederholter Übertragung



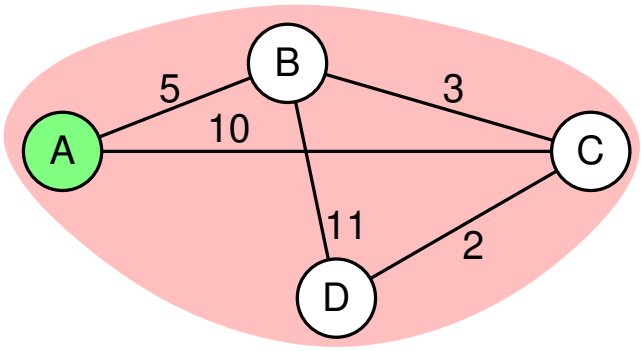
Dijkstra-Algorithmus zur Bestimmung kürzester Wege

- Eingabe: N: Knotenmenge, $l(i,j)$: Link-Kosten, s: Startknoten
- Ausgabe: $C(n)$: Pfad-Kosten von s zu n
- Algorithmus:

$M = \{s\}$
Für alle $n \in N - \{s\}$: $C(n) = l(s,n)$
Solange $M \neq N$:
  Wähle $w \in N - M$ so, daß $C(w)$ minimal ist
  $M = M \cup \{w\}$
  Für alle $n \in N - M$:
    $C(n) = \min(\, C(n),\, C(w) + l(w,n)\,)$

(Animierte Folie)

Beispiel



Schritt	w	M	C(B)	C(C)	C(D)
Initial		{A}	5	10	∞
1	B	{A,B}	5	8	16
2	C	{A,B,C}	5	8	10
3	D	{A,B,C,D}	5	8	10

Anmerkungen zu Folie 190:

Um neben der kürzesten Entfernung auch den nächsten Knoten auf dem Weg zum Ziel (*Next Hop*) zu erhalten, kann der Dijkstra-Algorithmus einfach erweitert werden. Die notwendige Erweiterung ist hier nicht dargestellt, sondern bleibt Ihnen als Übung überlassen.

OSPF (*Open Shortest Path First*)

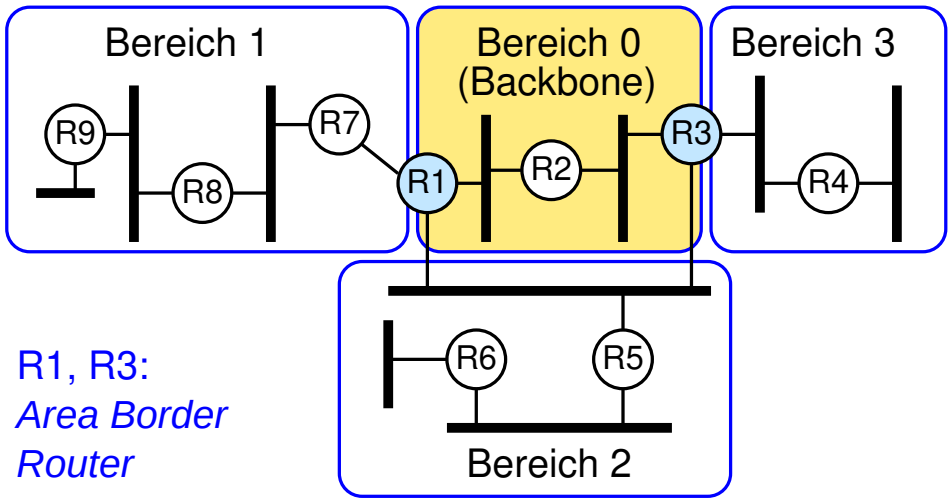
- Weit verbreitetes *Link State Routing* Protokoll
- Internet-Standard
- Link Metrik nicht spezifiziert, in der Praxis: Bandbreite
- Versionen: OSPFv2 für IPv4, OSPFv3 für IPv6
- Besonderheit in Mehrfachzugriffsnetzen:
  - Router wählen einen „designierten“ Router (DR) und einen Backup DR (BDR)
    - bei Ausfall DR: BDR wird neuer DR, Neuwahl BDR
  - Link State Pakete werden nur an diese gesendet und vom DR an alle weiterverteilt
  - verhindert exzessives Flooding im Netz

Anmerkungen zu Folie 191:

- OSPF besitzt fünf verschiedene Pakettypen, die in ein IP-Paket gekapselt und per Multicast versendet werden:
  1. *Hello*-Pakete dienen dazu, die Verbindung zu benachbarten Routern zu testen und in Mehrfachzugriffsnetzen den DR/BDR zu wählen.
  2. *Database Description*-Pakete werden verwendet, um den OSPF-Datenbestand eines Routers mit seinem Nachbarn zu synchronisieren, wenn die Verbindung zu diesem Nachbarn hergestellt wird.
  3. *Link State Request*-Pakete dienen zur expliziten Anforderung von Link State Information.
  4. *Link State Update*-Pakete enthalten ein oder mehrere *Link State Advertisements* (LSA). Ein LSA beschreibt dabei eine Verbindung zwischen zwei Routern.
  5. *Link State Acknowledgements* dienen zur Bestätigung beim *Reliable Flooding*.
- OSPFv2 ist im Detail im RFC 2328 spezifiziert, OSPFv3 im RFC 5340.

Multi-Area OSPF

- Für große *Routing-Domains*:
  - OSPF erlaubt Einführung einer weiteren Hierarchieebene: **Routing-Bereiche** (***Areas***)
- Beispiel:



Multi-Area OSPF ...

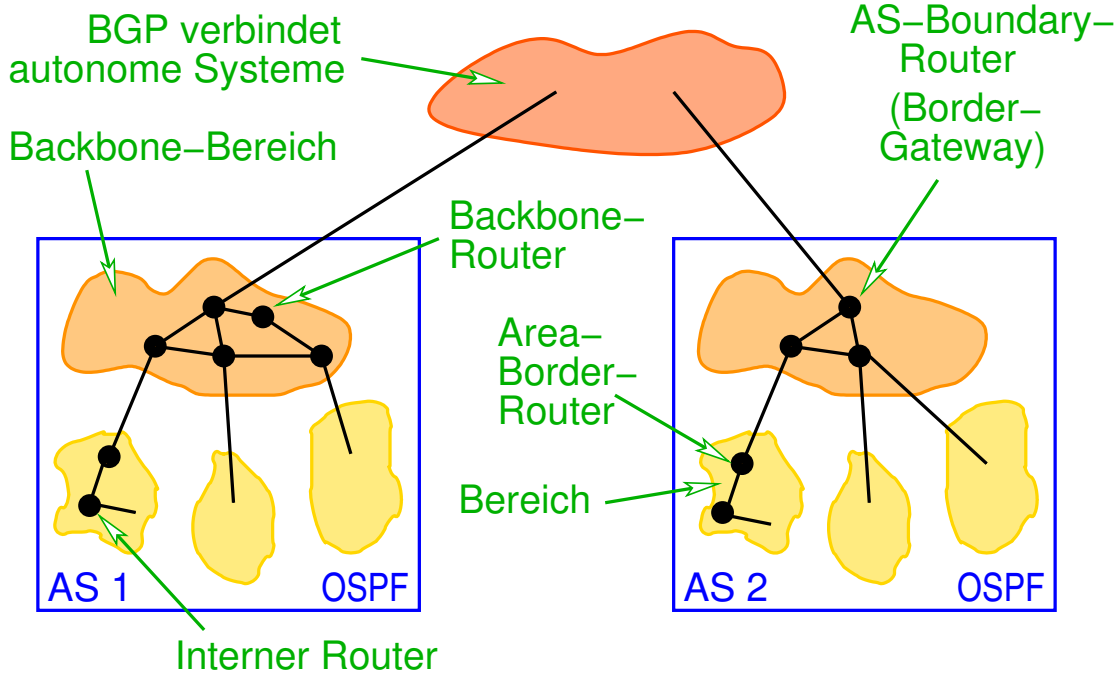
- Innerhalb jedes Bereichs: *Flooding* von Link-State-Paketen
- *Area Border Router* (ABR) geben diese nicht weiter
- Stattdessen: ABR sendet zusammengefaßte Information
  - nur ein Link-State-Paket für den gesamten Bereich
  - ABR spiegelt vor, daß alle Hosts in seinem Bereich **direkt** mit ihm verbunden sind
- Pakete zwischen Bereichen immer über ABR geleitet
  - bei mehreren ABR: automatische Auswahl über die Link-Kosten
- Damit: bessere Skalierbarkeit
  - kleinere Graphen in den einzelnen Routing-Bereichen
  - weniger Neuberechnungen der kürzesten Wege
  - evtl. aber suboptimale Routen

Anmerkungen zu Folie 193:

- OSPF verwendet fünf verschiedene Arten von Link State Advertisements:
  - Typ 1: enthält Information zu allen Router-Schnittstellen eines Bereichs, wird nur innerhalb des Bereichs weitergegeben.
  - Typ 2: wird vom DR an alle Router eines Mehrfachzugriffsnetzes gesendet, Weitergabe nur innerhalb eines Bereichts.
  - Typ 3: Zusammengefasste Information zur Erreichbarkeit von Netzen aus anderem Bereich.
  - Typ 4: Information zur Erreichbarkeit des AS Boundary Routers.
  - Typ 5: Information zur Erreichbarkeit von Netzen ausserhalb des AS (vom AS Boundary Router erzeugt).
- Suboptimale Routen entstehen zum einen durch die von Multi-Area OSPF erzwungene Topologie mit einem zentralen Backbone-Bereich, zum anderen durch die Implementierung des OSPF Protokolls. Details dazu finden Sie im Internet-Draft `http://tools.ietf.org/html/draft-ietf-ospf-shortcut-abr-02`

6.2.2 *Link State Routing ...*

Routing-Hierarchie mit Multi-Area OSPF



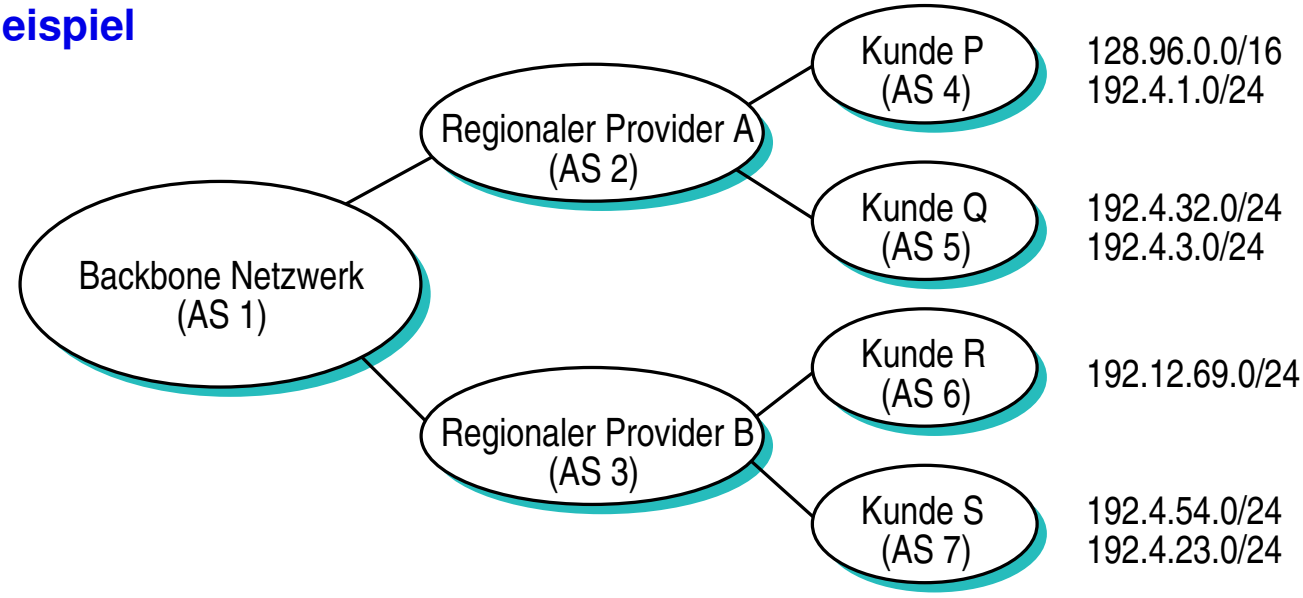
Routing innerhalb und außerhalb von Domains

- Innerhalb einer Domain: RIP, OSPF
  - Bestimmung optimaler Routen
- Zwischen Domains: ***Border Gateway Protocol* (BGP)**
  - Autonome Systeme $\Rightarrow$ keine gemeinsame Metrik
  - Routen werden „politisch“ bestimmt
    - „benutze Provider B für Adressen xyz“
    - „benutze Pfad über möglichst wenige AS“
  - Wichtigstes Ziel: Erreichbarkeit
    - „gute“ Routen sind sekundär

Routing mit BGP

- Jedes Autonome System (AS) hat
  - ein oder mehrere *Border Router*
    - Verbindung zu anderen AS
  - einen BGP Sprecher, der bekanntgibt:
    - lokales Netzwerk
    - über dieses AS erreichbare Netzwerke
      (nur, wenn das AS Pakete an andere AS weiterleitet)
- Bekanntgegeben werden vollständige Pfade
  - zur Vermeidung von zyklischen Routen

Beispiel



- AS 2 gibt (u.a.) bekannt: „ich kann AS 4, AS 5 direkt erreichen“
  - Netze 128.96.0.0/16, 192.4.1.0/24, 192.4.32.0/24, 192.4.3.0/24
- AS 1 gibt (u.a.) bekannt: „ich kann AS 4, AS 5 über AS 2 erreichen“

Anmerkungen zu Folie 197:

Etwas exakter:

- der BGP-Sprecher von AS2 gibt u.a. bekannt:
  - „über den Pfad AS2, AS4 kann ich folgende Netze erreichen: 128.96.0.0/16, 192.4.1.0/24“
  - „über den Pfad AS2, AS5 kann ich folgende Netze erreichen: 192.4.32.0/24, 192.4.3.0/24“
- der BGP-Sprecher von AS1 gibt u.a. bekannt:
  - „über den Pfad AS1, AS2, AS4 kann ich folgende Netze erreichen: 128.96.0.0/16, 192.4.1.0/24“
  - „über den Pfad AS1, AS2, AS5 kann ich folgende Netze erreichen: 192.4.32.0/24, 192.4.3.0/24“

Das bekanntgebende AS ist also immer Teil des Pfades.

6.4 Zusammenfassung

- Routing „im kleinen“ (innerhalb einer Domain):
  - Suche optimale Pfade
  - *Distance Vector Routing* (nur mit lokaler Information)
  - *Link State Routing* (globale Information, *reliable Flooding*)
- Routing „im großen“ (zwischen Domains)
  - *Border Gateway Protocol*
    - Bekanntgabe der Erreichbarkeit
    - Routenwahl ist „politische“ Entscheidung

Nächste Lektion:

- Ende-zu-Ende Prokotolle: UDP, TCP