

Internet Routing

1. Grundprinzipien
2. Dynamisches Routing
3. Distanz-Vektor Routingprotokolle
4. Link-State Routingprotokolle
5. Autonome Systeme, Inter-Domain Routing
6. Border Gateway Protokoll



Zum Inhalt

In diesem Abschnitt wird das Problem der Wegefindung im Internet behandelt. Dabei sollen einerseits eine dynamische Anpassung an die ‚besten‘ der vorhandenen Netzwerkübergänge erreicht, andererseits die hierarchische Struktur des globalen Internet und seine stete Anpassung an sich ändernde Akteure kennen gelernt werden.

Die zugehörigen (Teil-) Kapitel sind im Tanenbaum 5.6, im Bonaventure finden Sie die Inhalte in Teil 2. Eine gründlichere Behandlung des Routings findet sich in Christian Huitema: *Routing on the Internet*, 2nd Ed. Pearson, 2000



1. Routing versus Forwarding

Routing bezeichnet die Wegefindung im Netz

- ▶ Verteilter Prozess zwischen Routern im Netzwerk
- ▶ Basiert auf Topologie-Erkennung und Wegewahl
- ▶ Router haben oft Hilfstabellen hierfür (RIBs)

Forwarding bezeichnet das Weiterleiten von Paketen

- ▶ Lokaler Prozess auf einer Maschine
- ▶ Eingehende Pakete werden zum Ausgangsport geschickt
- ▶ Grundlage sind Weiterleitungstabellen (FIBs)
- ▶ FIBs sind die (lokalen) Ergebnisse des Routings



1. Forwarding im Internet

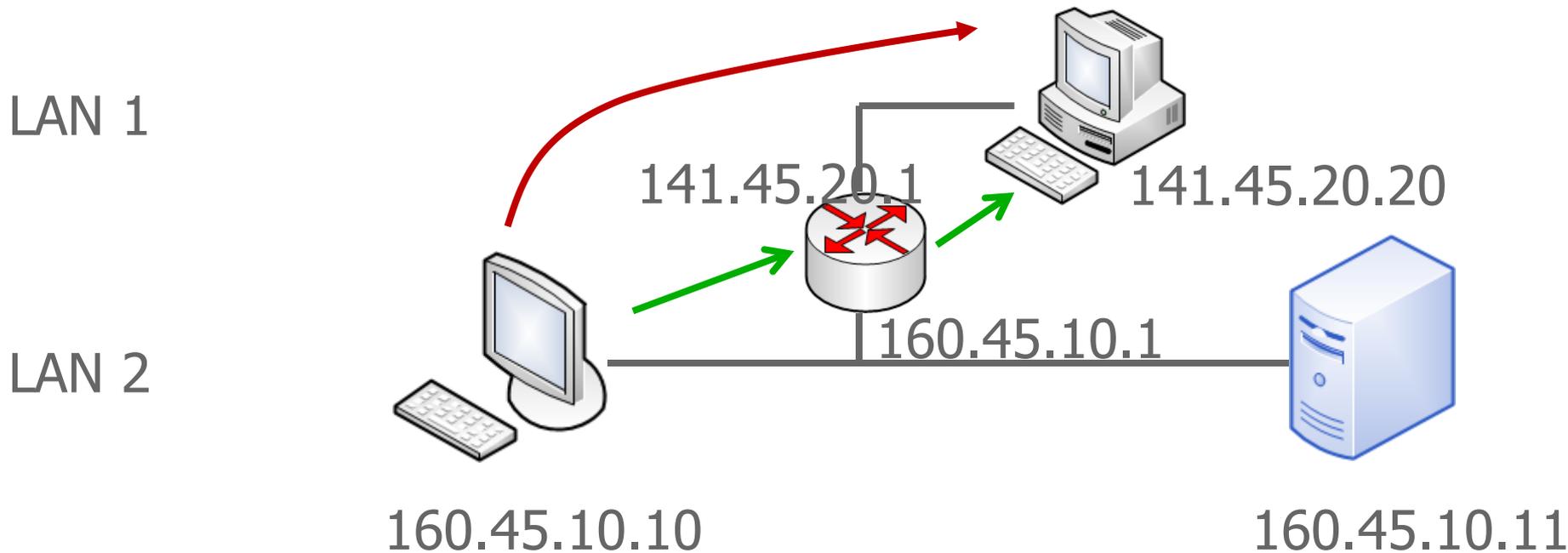
Forwarding bezeichnet die Weiterleitung der Pakete im Internet

Wichtigste Festlegungen:

- ▶ Die Forwarding-Entscheidung basiert allein auf der Zieladresse
- ▶ Jede Komponente bestimmt nur den nächsten Punkt des Weges (next hop), nicht den gesamten Weg zum Ziel
- ▶ Es gibt zwei Arten der Weiterleitung:
 - **Direktes Zustellen:** Der Zielrechner ist im gleichen Netz, d.h. direkt erreichbar
 - **Indirektes Weiterleiten:** Der Zielrechner ist nur über ein Gateway/Router erreichbar, an welchen das Paket zur Weiterleitung geschickt wird (**z.B. Defaultgateway**)



1. Ende-zu-Ende Paketauslieferung



- Welche Ziel-MACs nutzt der Sender?
- Wer spricht ARP mit wem?



1. Routing-Tabellen

Die Entscheidungen des indirekten Routings basieren auf Routing-Tabellen:

- ▶ Gateways und Hosts besitzen Routingtabellen
 - **Host-Tabellen:** meist nur das Defaultgateway
 - **Gateway-Tabellen:** Eintrag für jedes ‚erreichbare‘ Netz
- ▶ Statisches Routing bezeichnet die manuelle Tabellenpflege
- ▶ Automatisiertes Update von Routing-Tabellen zwischen Gateways ist im Internet notwendig \Rightarrow Routing-Protokolle
- ▶ Informationen über (lokale) Routingtabellen erhält man unter UNIX und Windows mit **netstat -r(n)**



1. Routing versus Forwarding

Routing-Tabelle

0.0.0.0	0.0.0.0	1	R3
172.217.20.0	255.255.255.0	1	R3
172.217.20.0	255.255.255.0	2	R4
172.217.23.0	255.255.255.0	2	R3
172.217.23.0	255.255.255.0	2	R4
172.217.0.0	255.255.0.0	1	R3

Zielnetzwerk

Netzmaske

Next hop
Meta-Informationen
(z.B. Pfadlänge)

Forwarding-Tabelle

0.0.0.0	0.0.0.0	R3
172.217.20.0	255.255.255.0	R3
172.217.23.0	255.255.255.0	R4
172.217.0.0	255.255.0.0	R3



1. Routing versus Forwarding

Routing-Tabelle

0.0.0.0	0.0.0.0	1	R3
172.217.20.0	255.255.255.0	1	R3
172.217.20.0	255.255.255.0	2	R4
172.217.23.0	255.255.255.0	2	R3
172.217.23.0	255.255.255.0	2	R4
172.217.0.0	255.255.0.0	1	R3

Welche Wege wählt der Router für die Forwarding-Tabelle?

Forwarding-Tabelle

0.0.0.0	0.0.0.0	R3
172.217.20.0	255.255.255.0	R3
172.217.23.0	255.255.255.0	R4
172.217.0.0	255.255.0.0	R3

An welches Zielnetzwerk wird ein Paket geschickt?



1.1 An welches Zielnetz wird ein Paket geschickt?

- **Problem:** Finde Eintrag mit größter Übereinstimmung mit der Ziel-IP-Adresse.



1.1 An welches Zielnetz wird ein Paket geschickt?

- ▶ **Problem:** Finde Eintrag mit größter Übereinstimmung mit der Ziel-IP-Adresse.

- ▶ Longest Common Prefix Match



1.1 Longest Common Prefix Match: Beispiel

Ziel-Adresse: 172.217.20.78

Forwarding-Tabelle

0.0.0.0	0.0.0.0	R3
172.217.20.0	255.255.255.0	R3
172.217.23.0	255.255.255.0	R4
172.217.0.0	255.255.0.0	R3



1.1 Longest Common Prefix Match: Beispiel

Ziel-Adresse: 172.217.20.78

Forwarding-Tabelle

0.0.0.0	0.0.0.0	R3
172.217.20.0	255.255.255.0	R3
172.217.23.0	255.255.255.0	R4
172.217.0.0	255.255.0.0	R3

255.255.255.0 ^ 172.217.20.78: 172.217.20.0
255.255.255.0 ^ 172.217.20.78: 172.217.20.0
255.255.0.0 ^ 172.217.20.78: 172.217.0.0



1.1 Longest Common Prefix Match: Beispiel

Ziel-Adresse: 172.217.20.78

Forwarding-Tabelle

0.0.0.0	0.0.0.0	R3
172.217.20.0	255.255.255.0	R3
172.217.23.0	255.255.255.0	R4
172.217.0.0	255.255.0.0	R3

255.255.255.0 ^ 172.217.20.78: 172.217.20.0
255.255.255.0 ^ 172.217.20.78: 172.217.20.0
255.255.0.0 ^ 172.217.20.78: 172.217.0.0



1.1 Longest Common Prefix Match: Beispiel

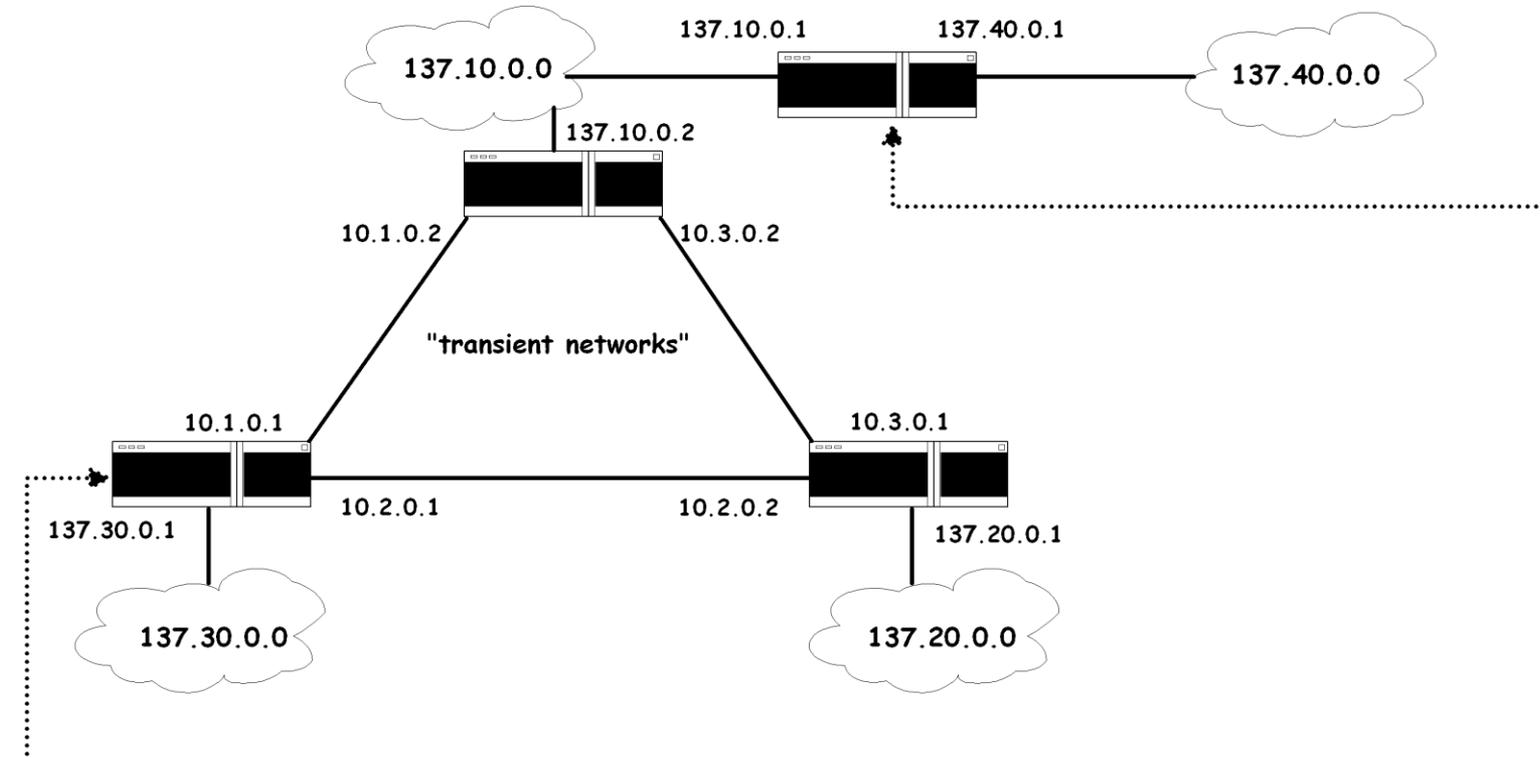
Ziel-Adresse: 172.217.20.78

Forwarding-Tabelle

0.0.0.0	0.0.0.0	R3
172.217.20.0	255.255.255.0	R3
172.217.23.0	255.255.255.0	R4
172.217.0.0	255.255.0.0	R3

255.255.255.0 ^ 172.217.20.78:	172.217.20.0
255.255.255.0 ^ 172.217.20.78:	172.217.20.0
255.255.0.0 ^ 172.217.20.78:	172.217.0.0

1.1 Forwarding im Beispielnetz



TO REACH HOSTS
ON NETWORK

ROUTE TO
THIS ADDRESS

137.30.0.0/16	direct /137.30.0.1
137.20.0.0/16	10.2.0.2
137.10.0.0/16	10.1.0.2
137.40.0.0/16	10.1.0.2

TO REACH HOSTS
ON NETWORK

ROUTE TO
THIS ADDRESS

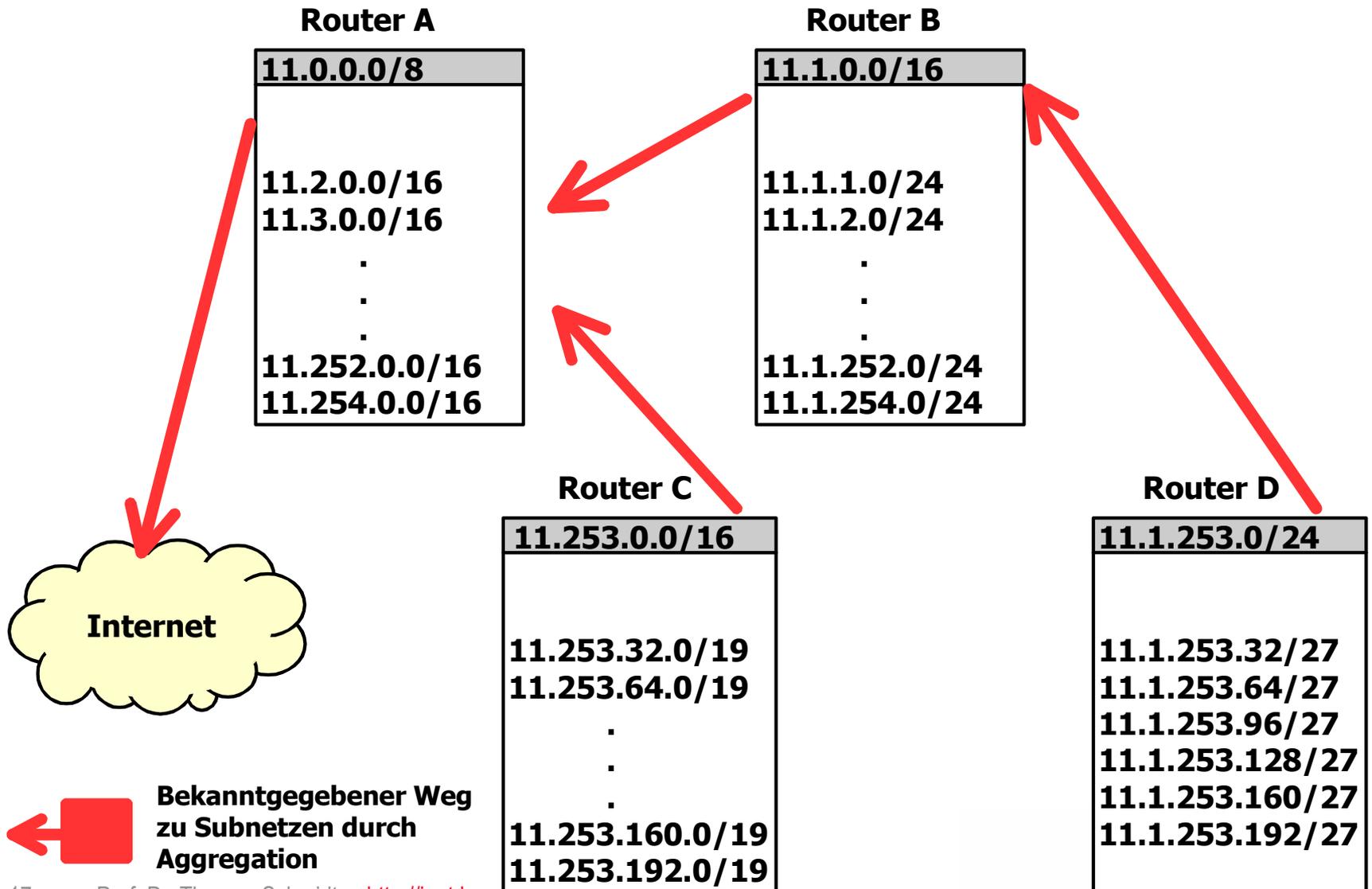
137.40.0.0/16	direct /137.40.0.1
137.10.0.0/16	direct /137.10.0.1
137.30.0.0/16	137.10.0.2
137.20.0.0/16	137.10.0.2

1.2 IP Routing: CIDR

- Statische Subnetzmasken in IP sind nicht flexibel genug, um dem wachsenden Strukturierungsbedarf des Internet zu entsprechen.
- Internet Backbone Router benötigen Methoden zur Verdichtung, um Routingtabellen zu begrenzen:
 - Classless Interdomain Routing (CIDR)
 - Variable Length Subnet Masks (VLSM)
- Ansatz:
 - Vergabe von Blöcken von Netzadressen
 - Bezeichnung durch ‚Supernetting‘ Adresse
- Routing erfolgt entlang längster Netzmaskenübereinstimmung



1.2 Routenverdichtung durch VLSM



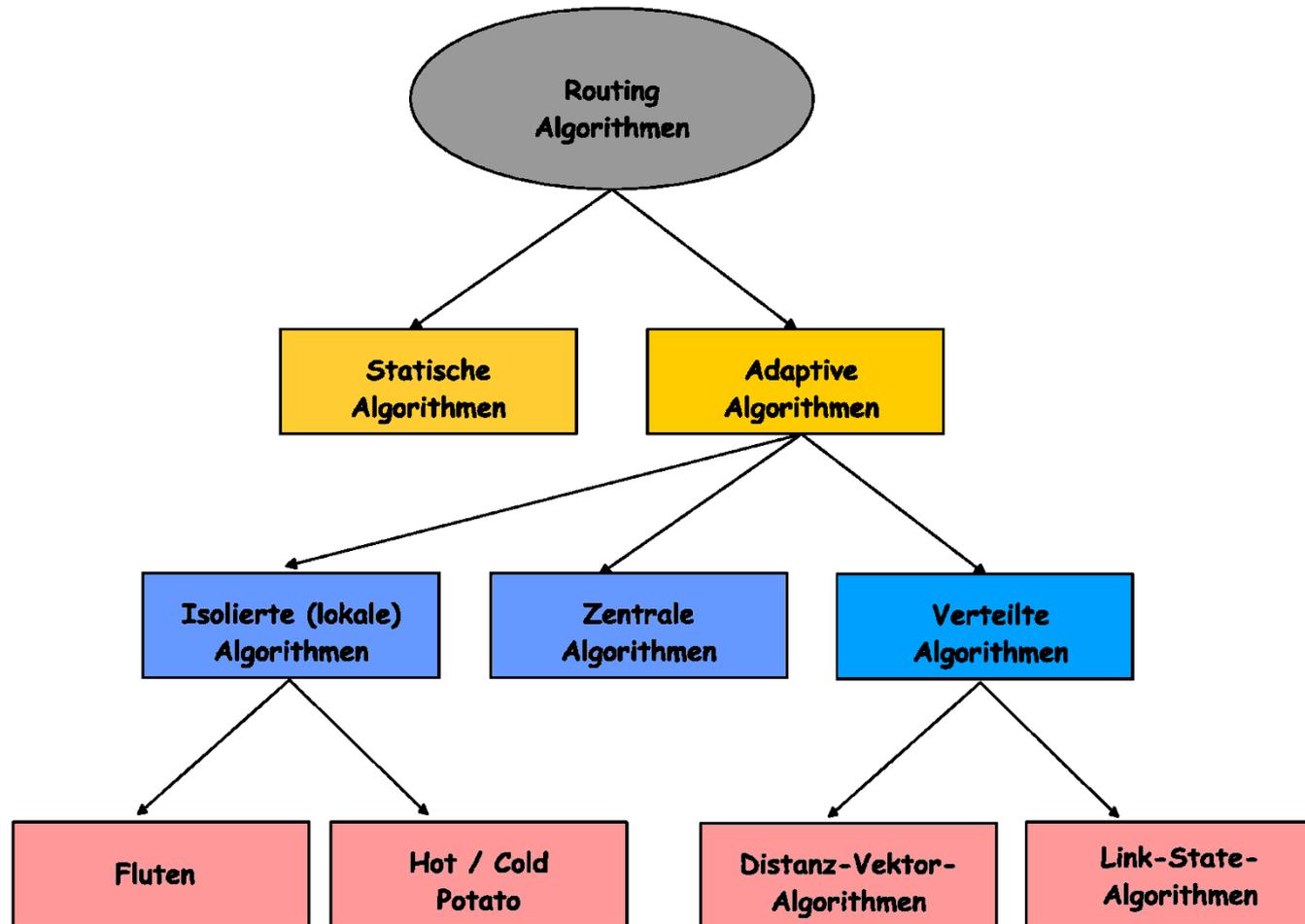
2. Dynamisches Routing

Da statische Routing-Tabellen zu inflexibel für das schnell veränderliche Internet sind, werden diese i.d.R. dynamisch verändert:

- **Routing-Informationen** werden mithilfe von Protokollen automatisch ausgetauscht (Topologie, Güte)
- Hierzu ‚passende‘ **Routing-Algorithmen** übernehmen die Entscheidung über ein Update der lokalen Routing-Tabellen.



2. Routing-Algorithmen



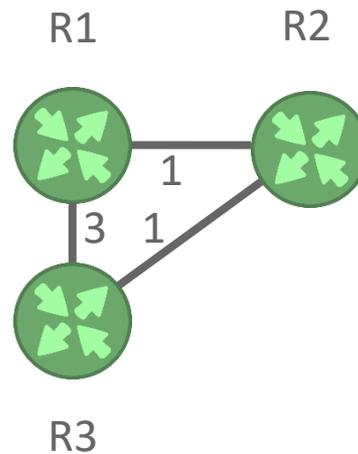
3. Distanz-Vektor Algorithmen

Ziel: Jeder Router verfügt über eine Liste kürzester Distanzen zu allen Netzen der Routing-Domain (Distanzvektor).

- Distanzen werden in einheitlicher, vorgegebener Metrik gemessen.
- Router „broadcasten“ ihre Routingtabellen.
- Update Algorithmus:
Neu erlernte kürzere Wege ersetzen bisherige Einträge.



3. Schritt 0: Initialisierung



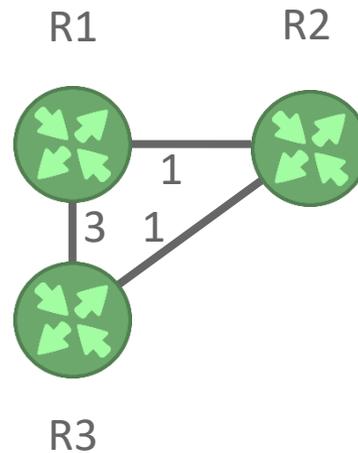
R1: Destination	Next Hop	Costs
R1	Local	0

R2: Destination	Next Hop	Costs
R2	Local	0

R3: Destination	Next Hop	Costs
R3	Local	0



3. Schritt 1: Direkte Nachbarn lernen



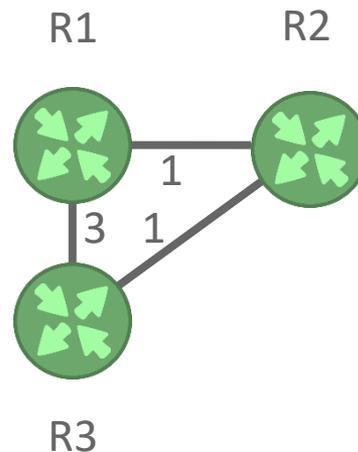
R1: Destination	Next Hop	Costs
R1	Local	0
R2	R2	1
R3	R3	3

R2: Destination	Next Hop	Costs
R2	Local	0
R1	R1	1
R3	R3	1

R3: Destination	Next Hop	Costs
R3	Local	0
R2	R2	1
R1	R1	3



3. Schritt 2: Indirekte Nachbarn lernen



R1: Destination	Next Hop	Costs
R1	Local	0
R2	R2	1
R3	R2	2

R2: Destination	Next Hop	Costs
R2	Local	0
R1	R1	1
R3	R3	1

R3: Destination	Next Hop	Costs
R3	Local	0
R2	R2	1
R1	R2	2



3. Updates?

- ▶ Wie schnell verteilen sich gute und schlechte Nachrichten?



3. Updates

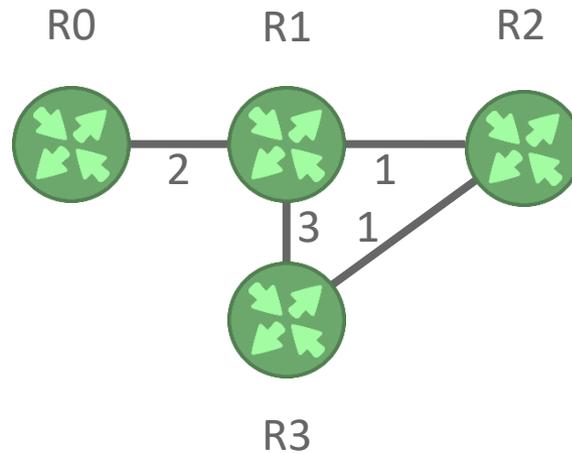
Link-Gewicht sinkt;
neuer Router

➔ Wie schnell verteilen sich **gute** und **schlechte** Nachrichten?

Link-Gewicht steigt;
Router fällt aus



3. Good News: Neuer Router



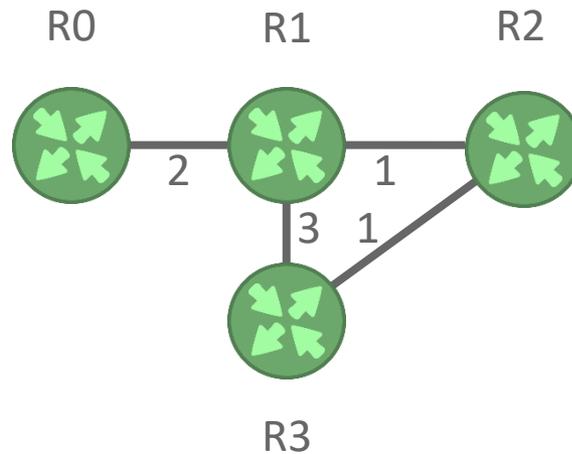
R1: Destination	Next Hop	Costs
R1	Local	0
...		
R0	R0	2

R2: Destination	Next Hop	Costs
R2	Local	0
R1	R1	1
R3	R3	1

R3: Destination	Next Hop	Costs
R3	Local	0
R2	R2	1
R1	R2	2



3. Good News: Neuer Router



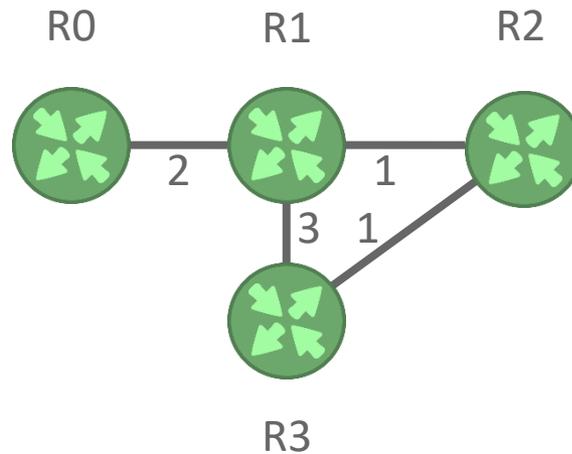
R1: Destination	Next Hop	Costs
R1	Local	0
...		
R0	R0	2

R2: Destination	Next Hop	Costs
R2	Local	0
...		
R0	R1	3

R3: Destination	Next Hop	Costs
R3	Local	0
...		
R0	R1	5



3. Good News: Neuer Router



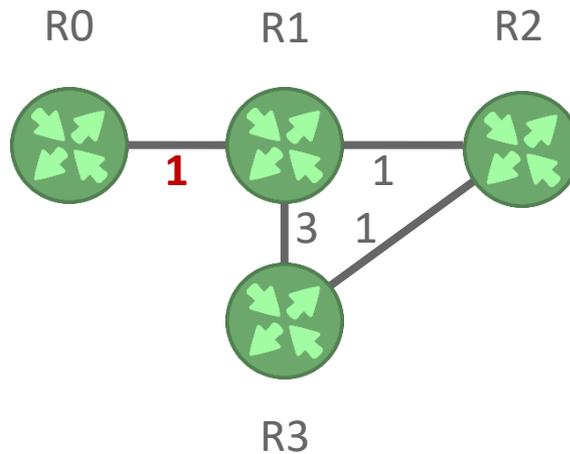
R1: Destination	Next Hop	Costs
R1	Local	0
...		
R0	R0	2

R2: Destination	Next Hop	Costs
R2	Local	0
...		
R0	R1	3

R3: Destination	Next Hop	Costs
R3	Local	0
...		
R0	R2	4



3. Good News: Link-Gewicht sinkt



R1: Destination	Next Hop	Costs
R1	Local	0
...		
R0	R0	1

R2: Destination	Next Hop	Costs
R2	Local	0
...		
R0	R1	2

R3: Destination	Next Hop	Costs
R3	Local	0
...		
R0	R2	3

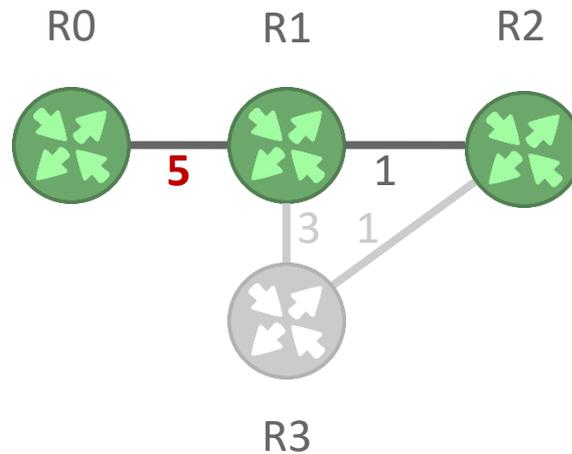


3. „Good news travel fast!“

- ▶ Wie schnell verteilen sich gute und schlechte Nachrichten?
- ▶ Wenn N die maximale Pfadlänge in Hops ist, brauchen wir max. N Schritte.

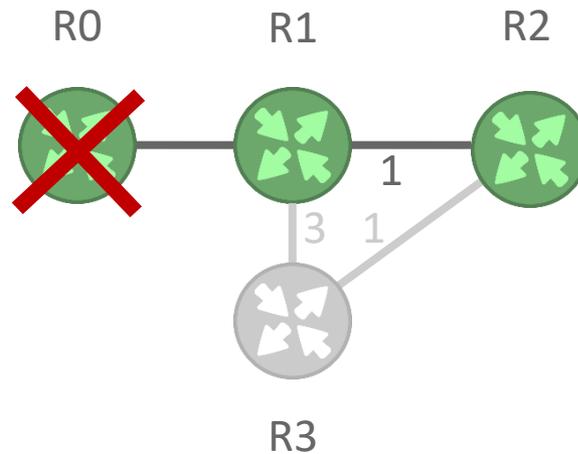


3. Bad News: Link-Gewicht steigt



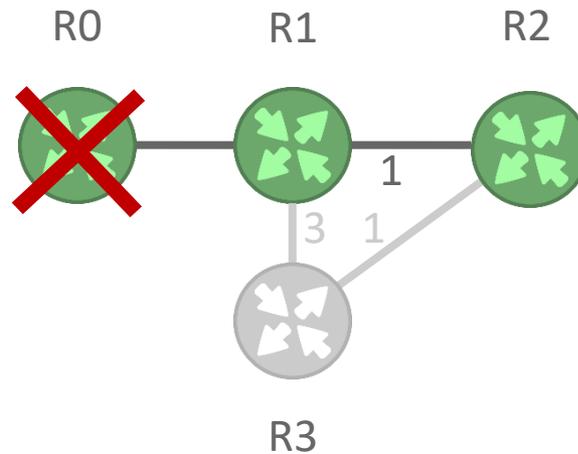
Was passiert, wenn R1 ein Update von R2 zu R0 erhält?

3. Bad News: Router fällt weg



Was passiert, wenn R1 ein Update von R2 zu R0 erhält?

3. Bad News: Router fällt weg



Was passiert, wenn R1 ein Update von R2 zu R0 erhält?

Link-Gewichte werden unendlich hoch („Count to Infinity“).

3. Bad News: Router fällt weg

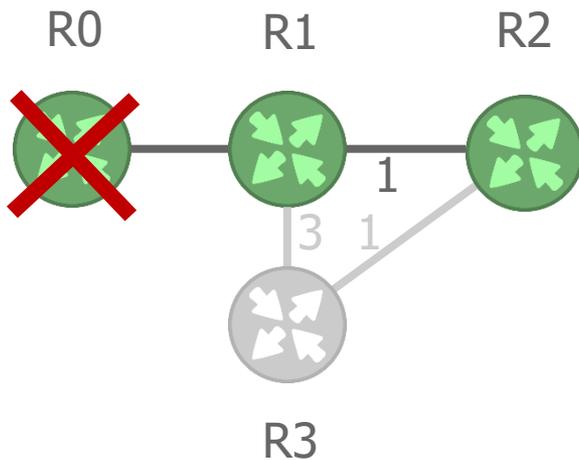
R0 R1 R2

Wir müssen dauerhafte Schleifen vermeiden.

R3



3. Wie vermeiden wir Schleifen?



► Split Horizon

Gebe Route nicht über den Weg bekannt, über den sie gelernt wurde.

► Split Horizon mit Poisoned Reverse

► Gebe Route über den Weg, über den sie gelernt wurden, mit Kosten Unendlich an.



3. Wie vermeiden wir Schleifen?

► Split Horizon

R0

R1

R2

Gebe Route nicht über den Weg

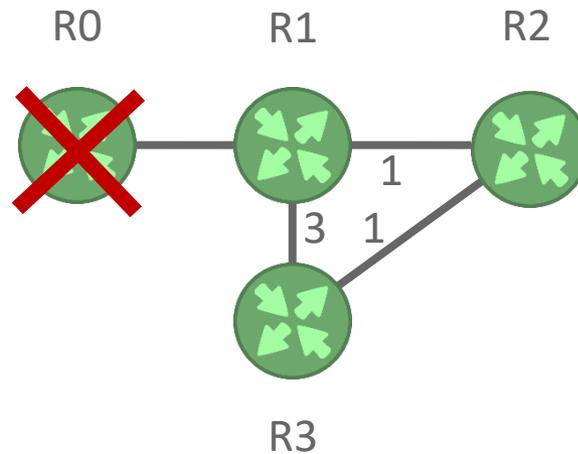
Funktioniert der Ansatz für alle Topologien?

R3

- Split Horizon mit Poisoned Reverse
- Gebe Route über den Weg, über den sie gelernt wurden, mit Kosten Unendlich an.



3. Bad News: Router fällt weg



Trotz Split Horizon mit
Poisoned Reverse kann es zu
dauerhaften Schleifen
kommen.

3. Wie vermeiden wir Schleifen?

- ▶ Split Horizon
- ▶ Split Horizon mit Poisoned Reverse
- ▶ **Vordefinierte maximale Kosten**



3. „Bad news travel slow!“

- Wie schnell verteilen sich gute und schlechte Nachrichten?
- Wenn N die max. möglichen Kosten sind, brauchen wir max. N Schritte.



3. Beispiel: Routing Information Protocol (RIP)

- ▶ Distance-vector Protokoll
- ▶ RFC 1058, viele (auch public domain) Implementierungen
- ▶ Router tauschen mit RIP ihre aktuellen Routing-Tabellen alle 30 Sekunden mit den direkten Nachbarn aus
- ▶ RIP ändert Routing-Einträge auf den Rechnern direkt
- ▶ RIP stützt sich auf eine ‚hop count metric‘
- ▶ RIP benutzt UDP, ab Version 2 Multicast und simple Authentifizierung
- ▶ Die max. Hop-Distanz beträgt 15 ($\infty := 16$)

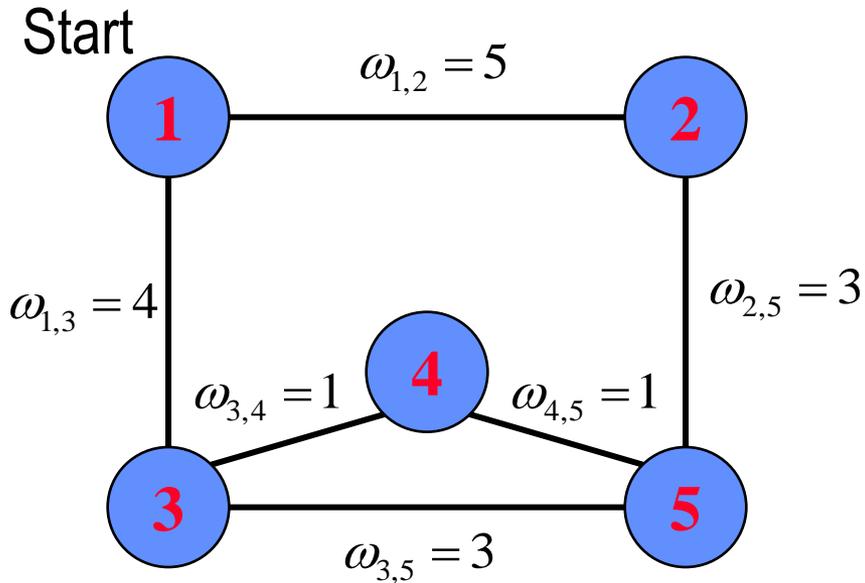


4. Link-State Algorithmen

- Jeder Router testet den Status seiner Links zu den Nachbarroutern.
- Link-Informationen werden an alle Router des Netzwerks weitergegeben (Fluten).
- Jeder Router baut hieraus eine Netztopologie auf.
- Alle Router des Netzwerks berechnen in gleicher Weise die günstigsten Wege und bilden Routing-Tabellen.



4. Graphentheorie: Dijkstra-Algorithmus



Der Dijkstra-Algorithmus funktioniert auch für gerichtete Graphen, d.h. asymmetrische Links.

Ziel: Errechne minimal gewichtete Wege vom Startpunkt aus: $D(k)$, f.a. Knoten k .

Init: $D(k) = \omega_{1,k}$ für k Nachbarn von 1
 $= \infty$ sonst.

$E = \{1\}$ (Knoten mit opt. Pfad)

Loop: Für alle $k \notin E$ finde min. $D(k')$

Füge k' zu E hinzu.

Für alle Nachbarn $i \notin E$:

$D(i) = \min (D(i), D(k') + \omega_{k',i})$

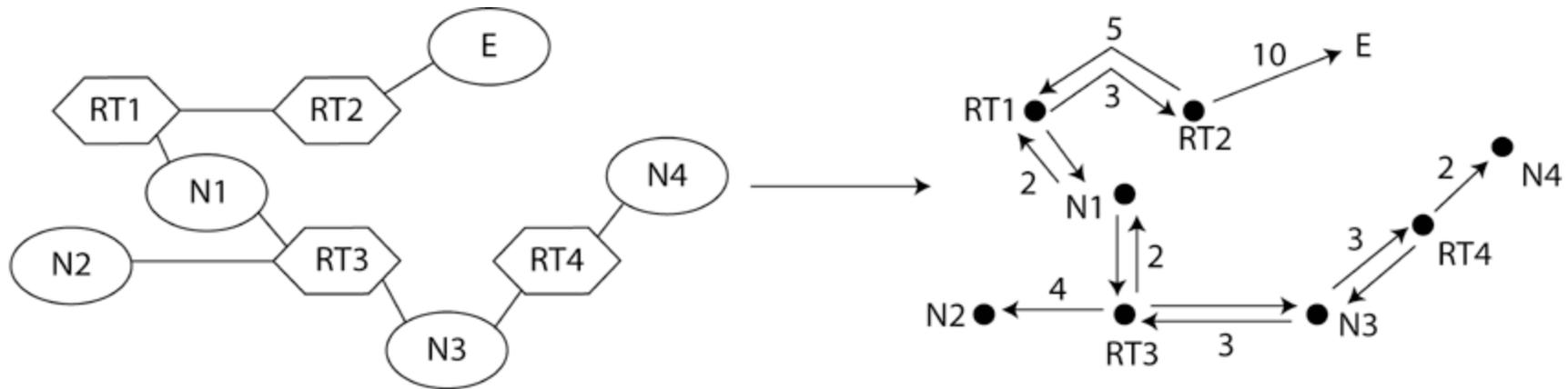


4. Beispiel: Open Shortest Path First (OSPF)

- OSPF (RFC 1247) gehört zur Familie der **Link-State** Protokolle
- OSPF setzt direkt auf IP auf (Nutzung von TOS).
- OSPF konvergiert schnell und unterstützt Load-Balancing.
- OSPF Routen haben Versionsnummern.
- OSPF unterteilt das Netz in **Areas** (IP Teilnetze, nach außen unsichtbar)
- OSPF unterscheidet Punkt-zu-Punkt-, Multi-Access- und Stub-Netze
- OSPF verarbeitet neben Topologieinformationen auch Cost-Gewichte
- Problem: Routing-Entscheidungen können sehr komplex werden.



4. OSPF Topologie

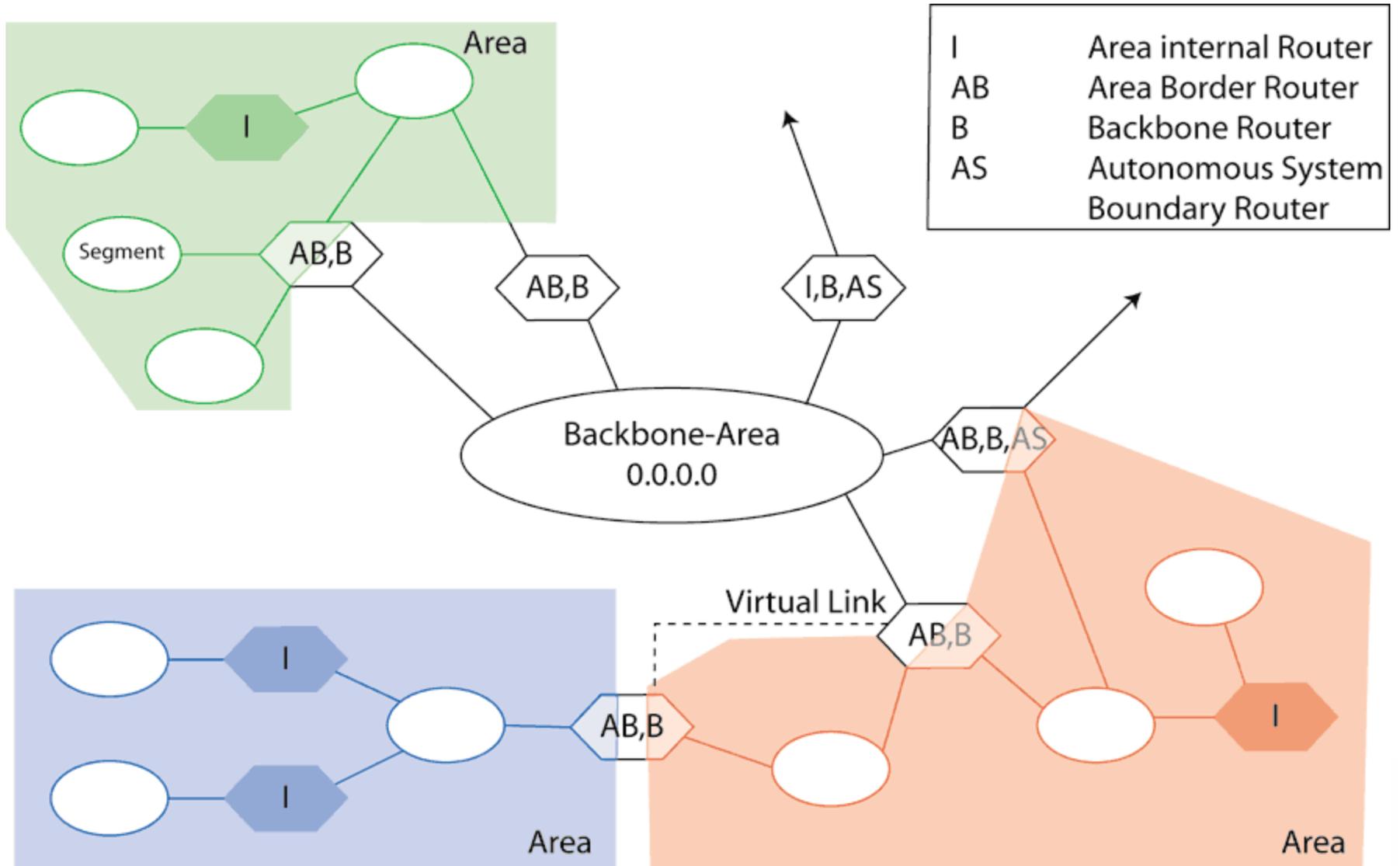


Routing Table at RT3:

Destination Net	N1	N2	N3	N4	E
NextHop	dd	dd	dd	RT4	RT1
Metric	2	4	3	5	15



4. OSPF Areas



4. Konfiguration von OSPF

Folgende Informationen benötigt ein OSPF-Router für seine Konfiguration

1. Konfiguriere IP Interfaces (einschl. Netzmaske)
2. Weise Netze zu Areas zu
3. Konfiguriere Link-Kosten auf dem Forwarding Interface – Achtung, diese sind richtungsgebunden
 - ▶ Typischerweise invers proportional zur Bandbreite:
Hat 100 Mbit/s Cost=100, dann hat 1 Gbit/s Cost=10
 - ▶ OSPF kann Kosten automatisch nach der Bandbreite zuweisen (benötigt Referenz-Bandbreite)



4. Intermediate System to Intermediate System (IS-IS)

- ▶ Ursprünglich OSI Protokoll für „verbindungslose Schicht 3“, RFC 1195 definiert Unterstützung für IPv4, RFC 5308 für IPv6
- ▶ Neutral bzgl. Schicht 3
- ▶ Link-State Protokoll wie OSPF
- ▶ Initiiert Shortest-Path-Forwarding (\Rightarrow Dijkstra)
- ▶ Unterstützt Domain-Bildung (Areas) in zweistufiger Hierarchie (wie OSPF)
- ▶ Hierarchisierung: Level-1 und Level-2 Router
- ▶ Router einer Domain lernen alle die gleiche Netzwerk-Topologie

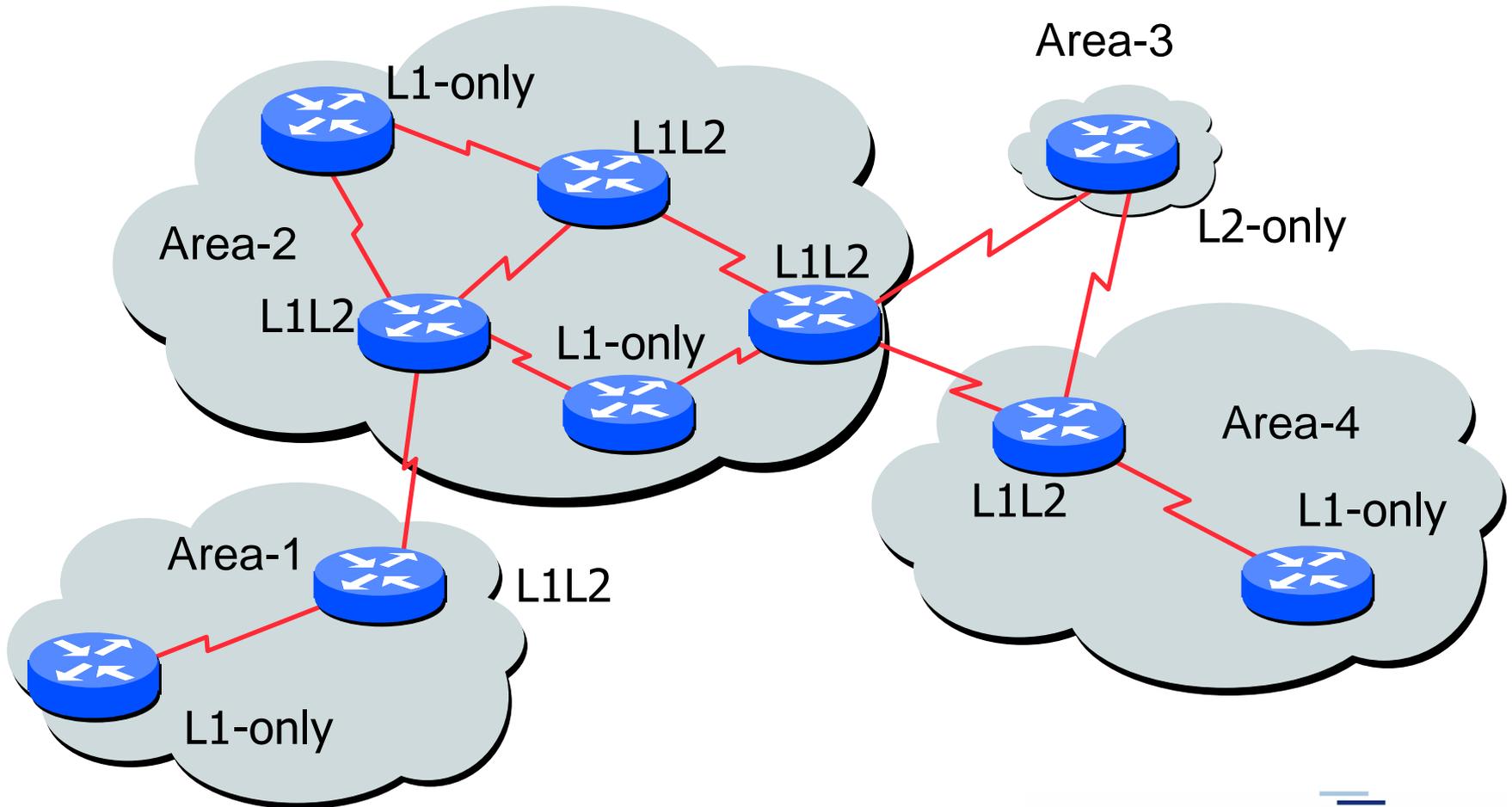


4. IS-IS: Unterschiede zu OSPF

- ▶ Area-Grenzen sind nicht Router, sondern Links
 - ▶ Router sind immer Teil *einer* Area
- ▶ Keine „Backbone“-Area: Backbone bilden alle Level-2 Router
 - ▶ Flexibler als OSPF: L-2 Router müssen nicht an topologischen Grenzen platziert sein
- ▶ Backbone muss zusammenhängend gewählt werden
 - ▶ Design Guide: Starte mit Level-2 Domain
- ▶ Adressierung: „Network Service Access Point (NSAP)“
 - ▶ Einige „Basteleien“ mit alten OSI-Konzepten
- ▶ IS-IS ist wieder populär geworden
 - ▶ Protokollneutralität: IPv6, MAC-Layer (Data Centers)



4. Domains und Router Levels



5. Internet Hierarchien: Autonome Systeme

Grundsätzlich zerfällt das Routing-Problem in

- Routing innerhalb von Netzwerken
- Routing zwischen Netzwerken

Deshalb gliedert sich das Internet in **Autonome Systeme (AS)**, deren innere Struktur nach außen transparent ist.

Router innerhalb eines Autonomen Systems heißen

Interior Neighbours

außerhalb:

Exterior Gateways

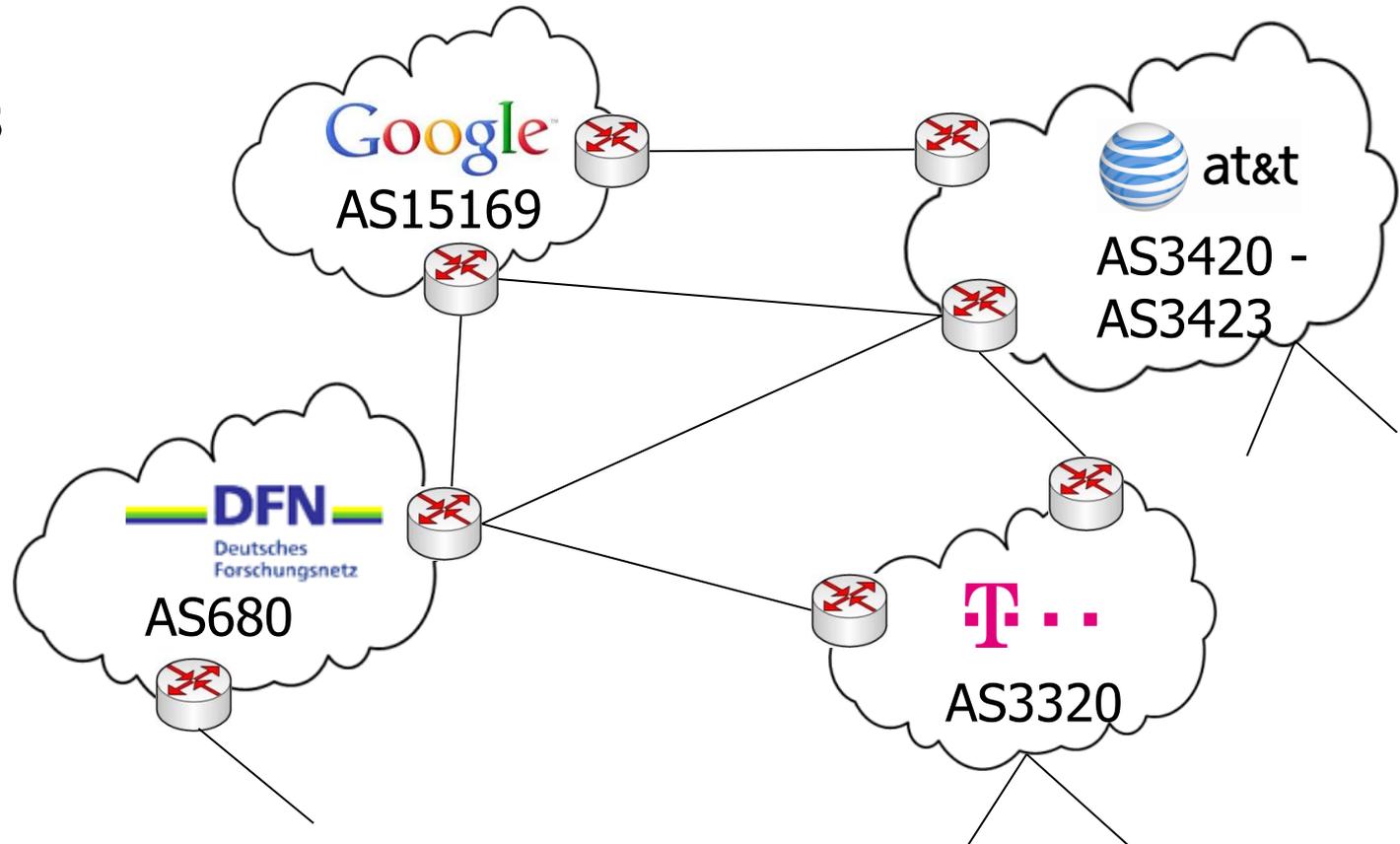
Innerhalb ASe wird **ein** frei wählbares internes RP gesprochen.

ASe besitzen eine global eindeutige AS-Nummer (ASN).



5. Das Internet: Netz der Netze

Autonomous System (AS)
=
One domain



IP Prefixes

160.45.0.0/16

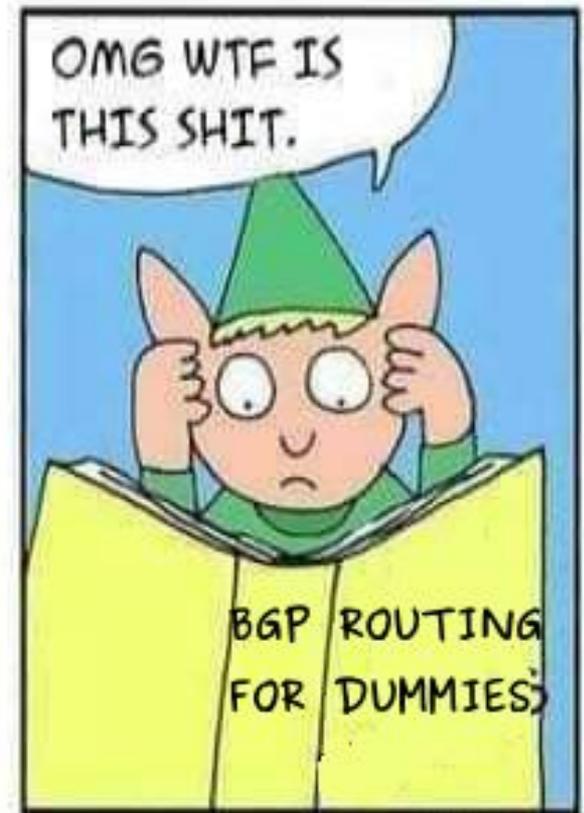
IP Blocks

160.45.10.0/24

160.45.20.0/24



5. Inter-Domain Routing



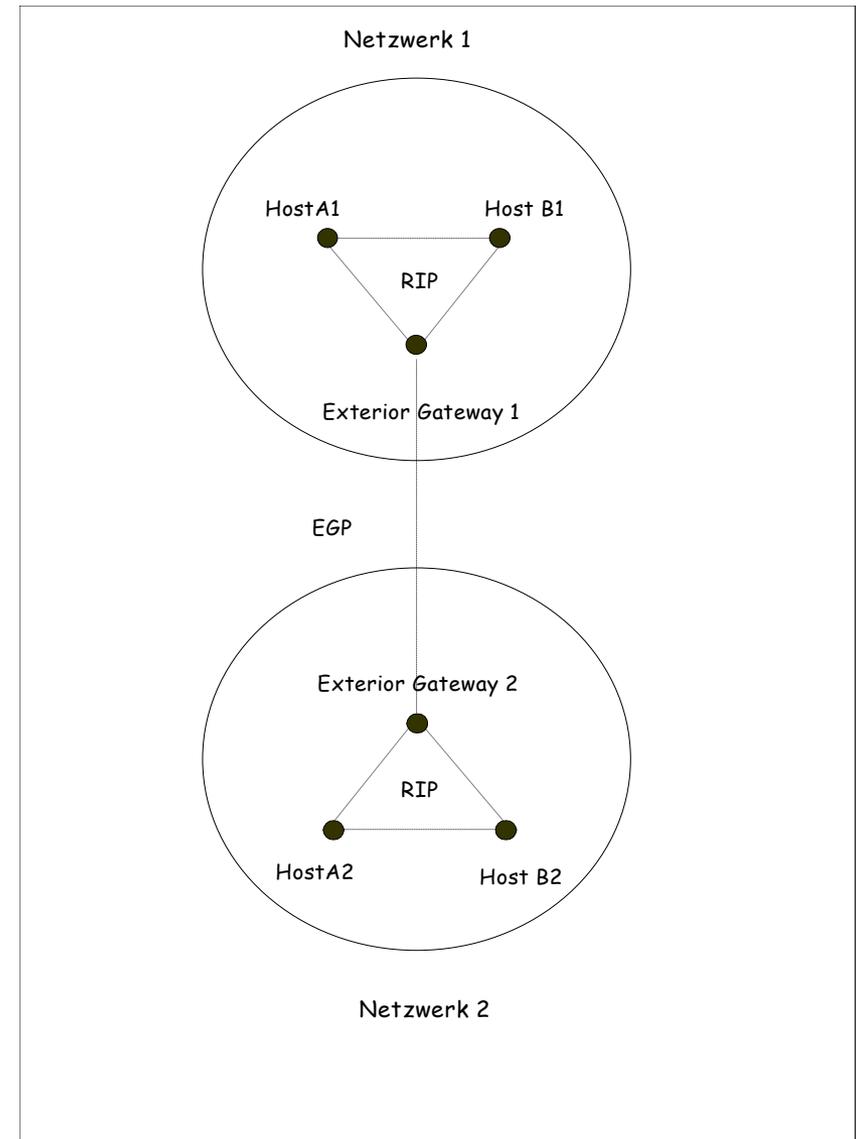
5. Exterior / Interior Gateway Protokolle

Exterior:

- EGP
- BGP

Interior:

- RIP
- RIP-V2
- OSPF
- IS-IS



5. Hierarchien der Internet Topologie

Peering Hierarchy

Business Relations

Tier 1: Global Internet Core

Settlement Free

Tier 2: National/Regional ISPs

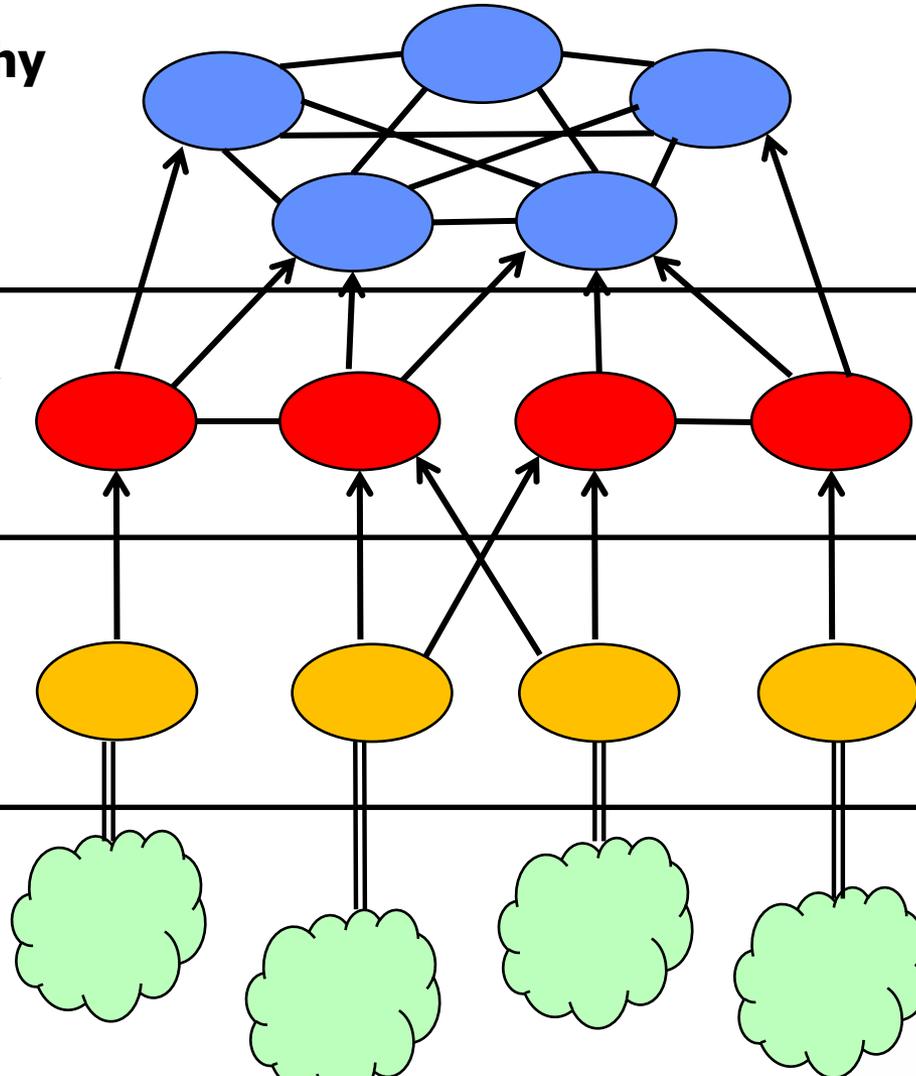
Pay for BW

Tier 3: Stub Networks, Local Eyeballs

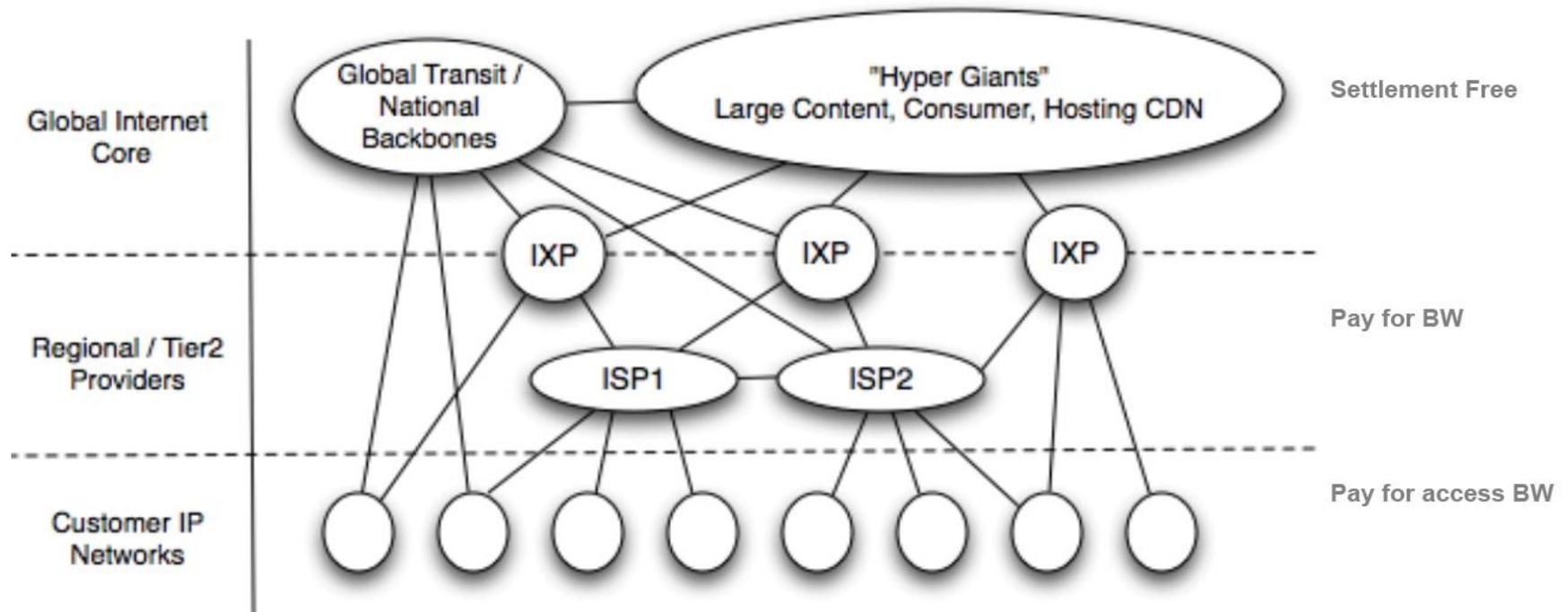
Pay for Upstream

Customer IP Networks (without ASN)

Pay for Access



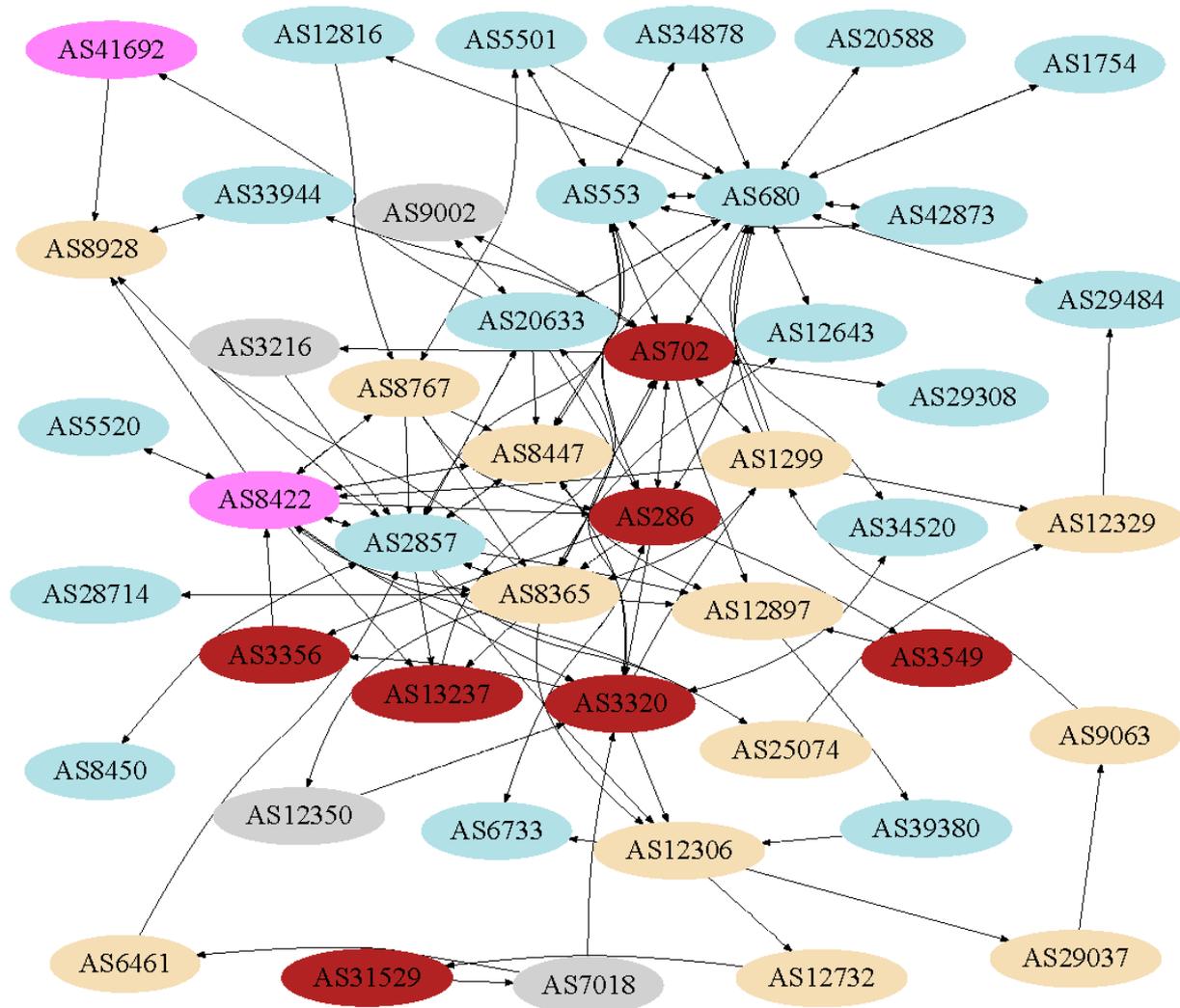
5. Aktualisiertes Internet-Modell



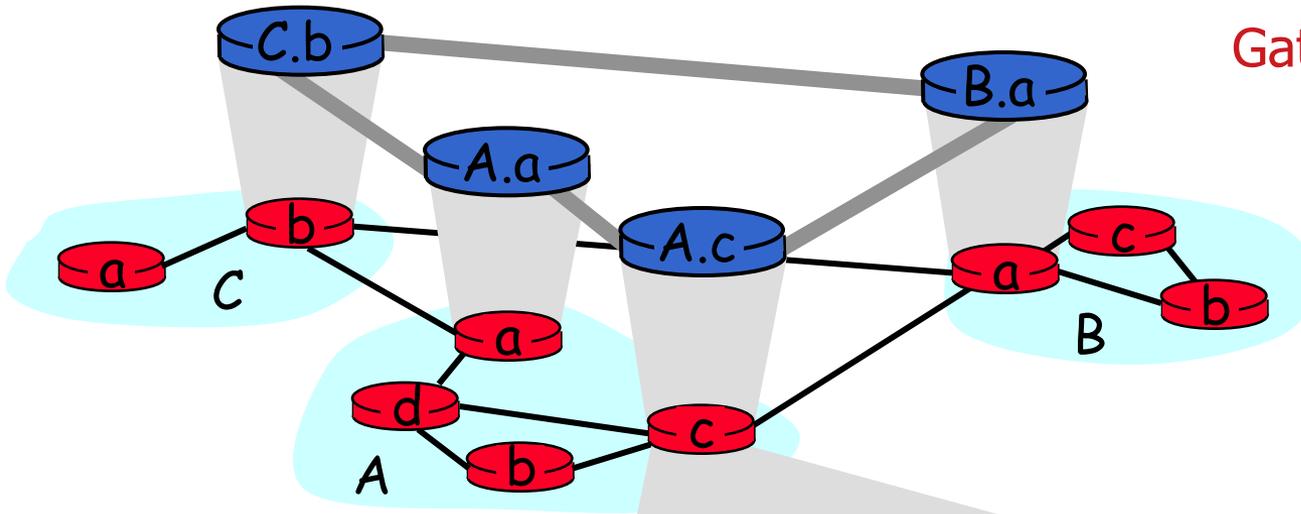
Quelle: G. Labovitz, et al.: Internet Inter-Domain Traffic, SIGCOMM 2010



Branchenausschnitt



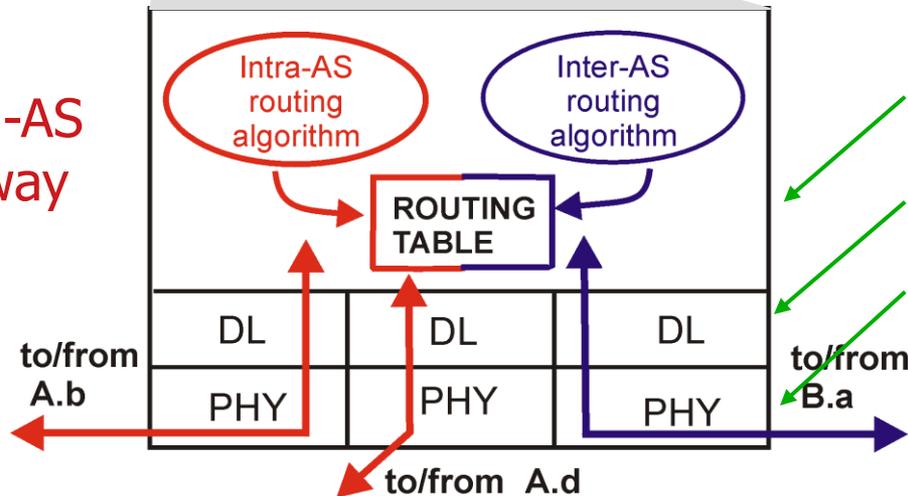
5. Intra-AS und Inter-AS Routing



Gateways:

- inter-AS routing untereinander
- intra-AS routing mit anderen Routern im selben AS

Inter-AS und intra-AS Routing im Gateway Router A.c



Netzwerkschicht
Sicherungsschicht
Physikal. Schicht



5. Ziele und Herausforderungen des Interdomain Routings

Ziel: Austausch von Routing-Informationen zwischen ASes

Herausforderungen:

- ▶ Skalierbarkeit: Aktuell >900k IP-Präfixe, >60k ASes, >1M Router
- ▶ Privatsphäre: Netzbetreiber möchten wenig über interne Struktur preisgeben
- ▶ Provider-spezifische Regeln („Policies“): Jeder Netzbetreiber hat ein eigenes Verständnis davon, was ein guter Weg ist, motiviert von ökonomischen, politische u.a. Argumenten



6. Beispiel: Border Gateway Protocol (BGP4)

- ▶ BGP (RFC 1771-73) gehört zur Familie der **Path Vector** Protokolle.
- ▶ Regelt Routing zwischen BGP-„Sprechern“ der Autonomen Systeme.
- ▶ Typischer Einsatz: ISP Peering
- ▶ BGP operiert auf Path-Vektoren (Liste der ASNs auf einem Weg).
- ▶ BGP Peers empfangen Path-Vektoren von direkten Nachbarn.
- ▶ BGP Peers akzeptieren/verwerfen angebotene Pfade: Offen für Policies, z.B. shortest Path, bevorzugte Nachbarn, Hot Potato oder Cold Potato
...
- ▶ BGP Router entscheiden via Policy über eigene „Advertisements“.

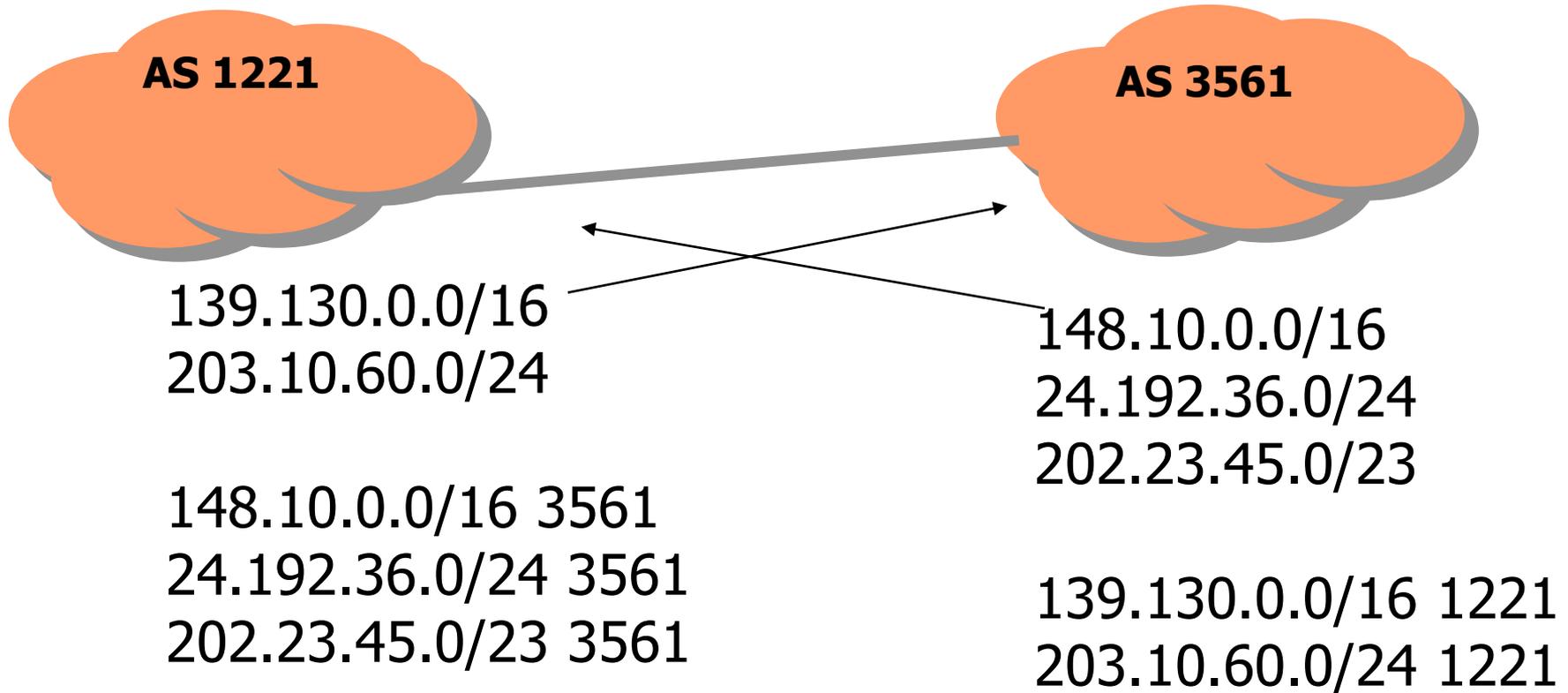


6. BGP4 Routing: Lernen der Wege

- ▶ BGP4-Router erfahren Wege (AS Pfade) zu **Präfixen** von ihren Nachbarn – **den BGP Peers**
- ▶ Ein BGP4-Router sammelt die (redundanten) Pfade in einer Tabelle: die **BGP-RIB** (Routing Information Base)
 - ▶ Diese schließt ein Mapping von IP-Adressen zu AS# ein
- ▶ BGP definiert die **„Default-freie Zone“**: Alle teilnehmenden BGP-Router darin müssen eine vollständige (ortsabhängig verschiedene) RIB vorhalten



6. Beispiel: BGP Pfad-Austausch



6. Beispiel: BGP RIB Eintrag

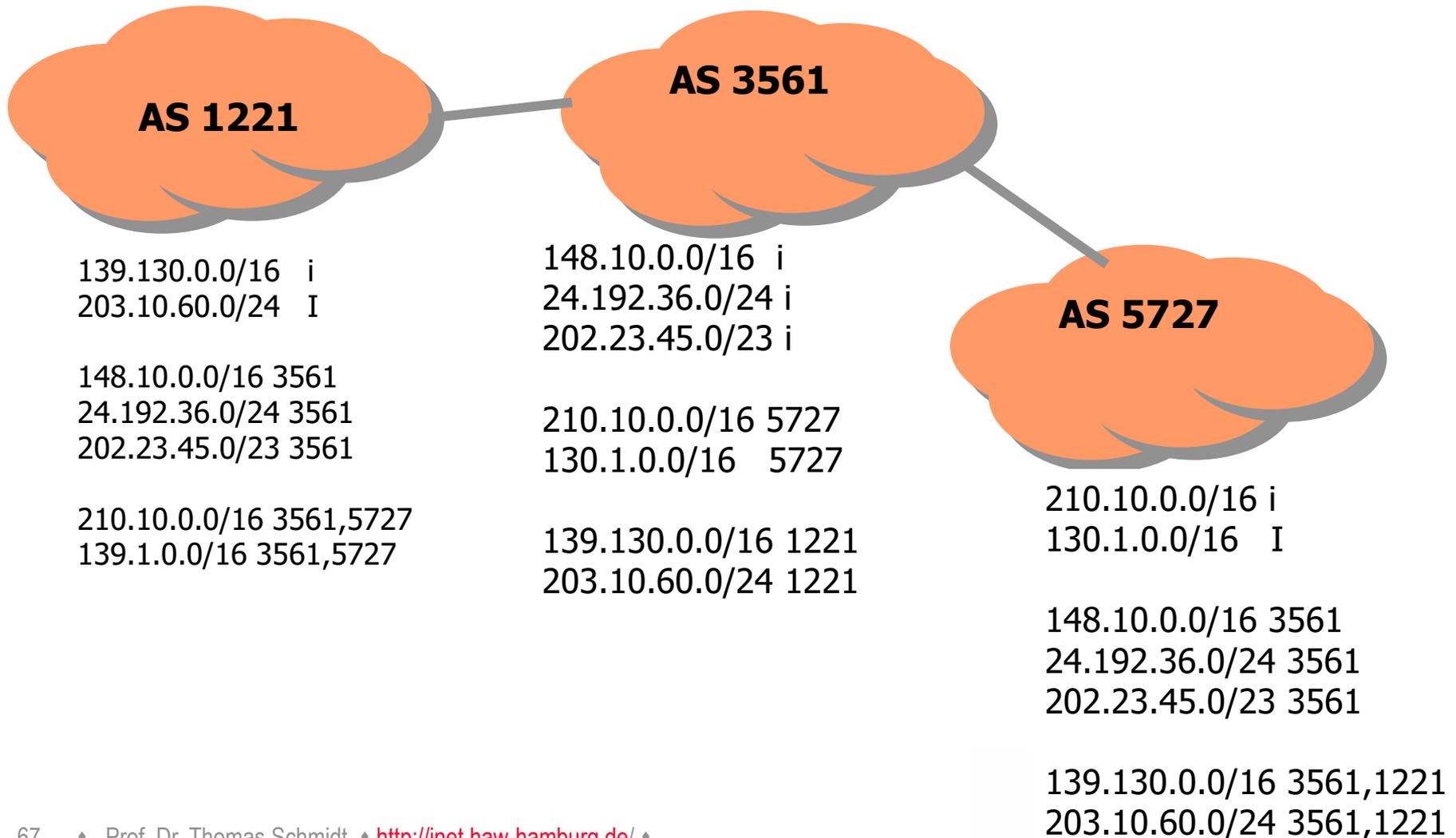
```
TIME: 2008-7-1 02:36:49
TYPE: MSG_TABLE_DUMP/AFI_IP6
VIEW: 0 SEQUENCE: 2702
PREFIX: 2001:0638::/32
ORIGINATED: Mon Jun 30 10:29:18 2008
FROM: 2001:0418:0000:1000:0000:0000:0000:f000 AS2914
AS_PATH: 2914 3549 680
MULTI_EXIT_DISC: 1
COMMUNITIES: 2914:420 2914:2000 2914:3000
```

Inter-AS Link Metrik
Priorisierung für redundante
Interprovider Peerings

Routing Policy Gruppe
gruppiert Propagationsart



6. Beispiel: BGP TRANSIT



6. BGP4 Routing: Auswahlentscheidungen

BGP4-Router müssen über Pfadauswahl und Weiterleitung entscheiden:

Phase I: Präferenz-Zuweisung

- Aufgrund lokaler Policies und Attribute erhalten alle RIB-Einträge eine Präferenz.

Phase II: Routen-Selektion

- Für **jedes Präfix** werden alle Routen höchster Präferenz ausgewählt, hiernach die kürzesten Pfade, dann die Multi-Exit-Discriminators und schließlich weitere Eindeutigkeitsregeln angewandt.
- Ausgewählte Routen wandern in die **FIB (Forwarding Information Base)**.

Phase III: Routen-Weiterleitung

- FIB-Einträge werden vor der Weiterleitung (Routen-Propagation) erneut gem. Policies gefiltert.





6. BGP Routing in der Wildnis

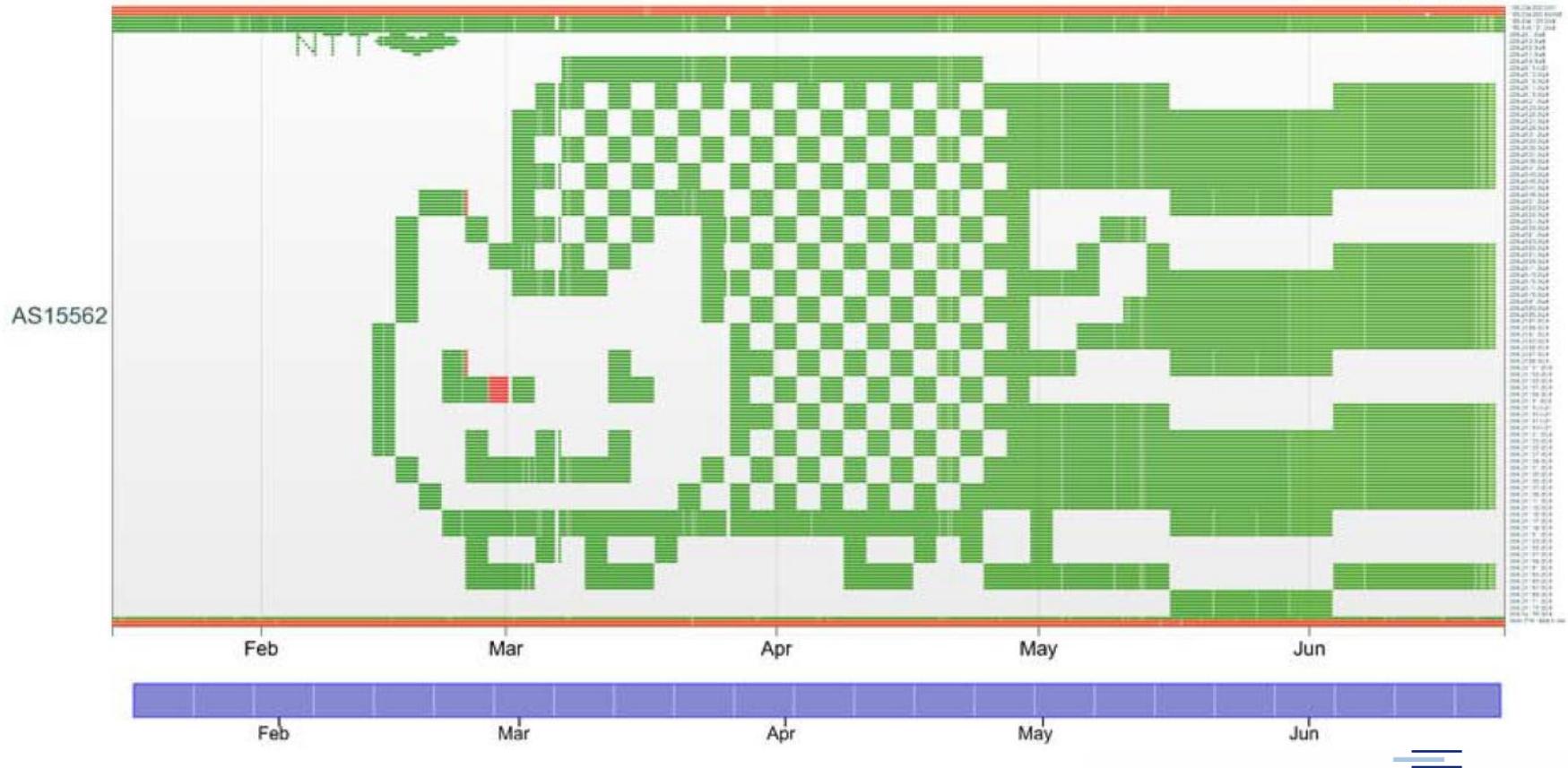
[Switch to Table V](#)

Show of **186** rows Sort by: 123 ↻ ↓ Condensed view

Filters (0): No large prefixes No short timespans No low visibility

Data resolution: 8 h

[Reset Zoom](#)



Selbsteinschätzungsfragen

1. Welche einzelnen Schritte muss ein Router bei dem Forwarding von IP Paketen vornehmen?
2. Es gibt mindestens 3 essentielle Vorteile von CIDR – welche?
3. Warum würde RIP schneller konvergieren, wenn es sich die Quelle seiner Routingeinträge merken würde?
4. Wie unterstützen OSPF und BGP4 asymmetrisches Routing?
5. Was ist die Default-freie Zone und warum gibt es sie?
6. Worin liegt das Skalierungsproblem von BGP4? 