

# Internet Routing

1. Grundprinzipien
2. Dynamisches Routing
3. Intra-Domain Routingprotokolle
4. Autonome Systeme
5. Inter-Domain Routingprotokolle



# 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 kennen gelernt werden.

Das zugehörige Kapitel im Tanenbaum ist 5.6, eine gründlichere Behandlung des Routings findet sich in Christian Huitema: *Routing on the Internet*, 2<sup>nd</sup> Ed. Pearson, 2000



# 1. Routing im Internet

Routing bezeichnet die Wegefindung der Pakete im Internet

Wichtigste Festlegungen:

- Die Routing-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 des Routings:
  - **Direktes Routing:** Der Zielrechner ist im gleichen Netz, d.h. direkt erreichbar
  - **Indirektes Routing:** Der Zielrechner ist nur über ein Gateway/Router erreichbar, an welchen das Paket zur Weiterleitung geschickt wird (z.B. **Defaultgateway**)



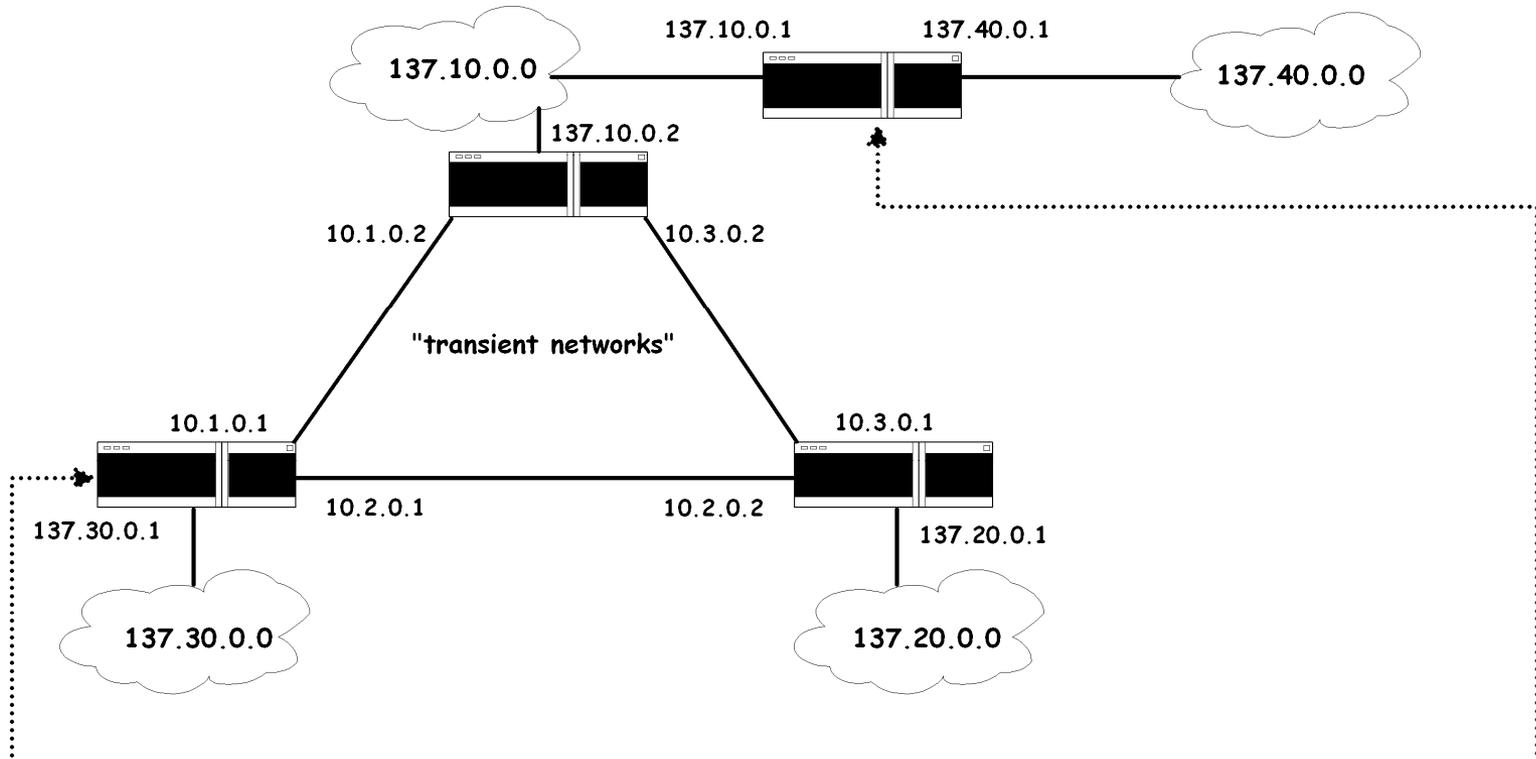
# 1. Routing-Tabellen

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 aber notwendig  $\Rightarrow$  Routing-Protokolle
- ▶ Informationen über Routingtabellen erhält man unter UNIX und Windows mit **netstat -r(n)**



# 1. Routing Beispiel



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

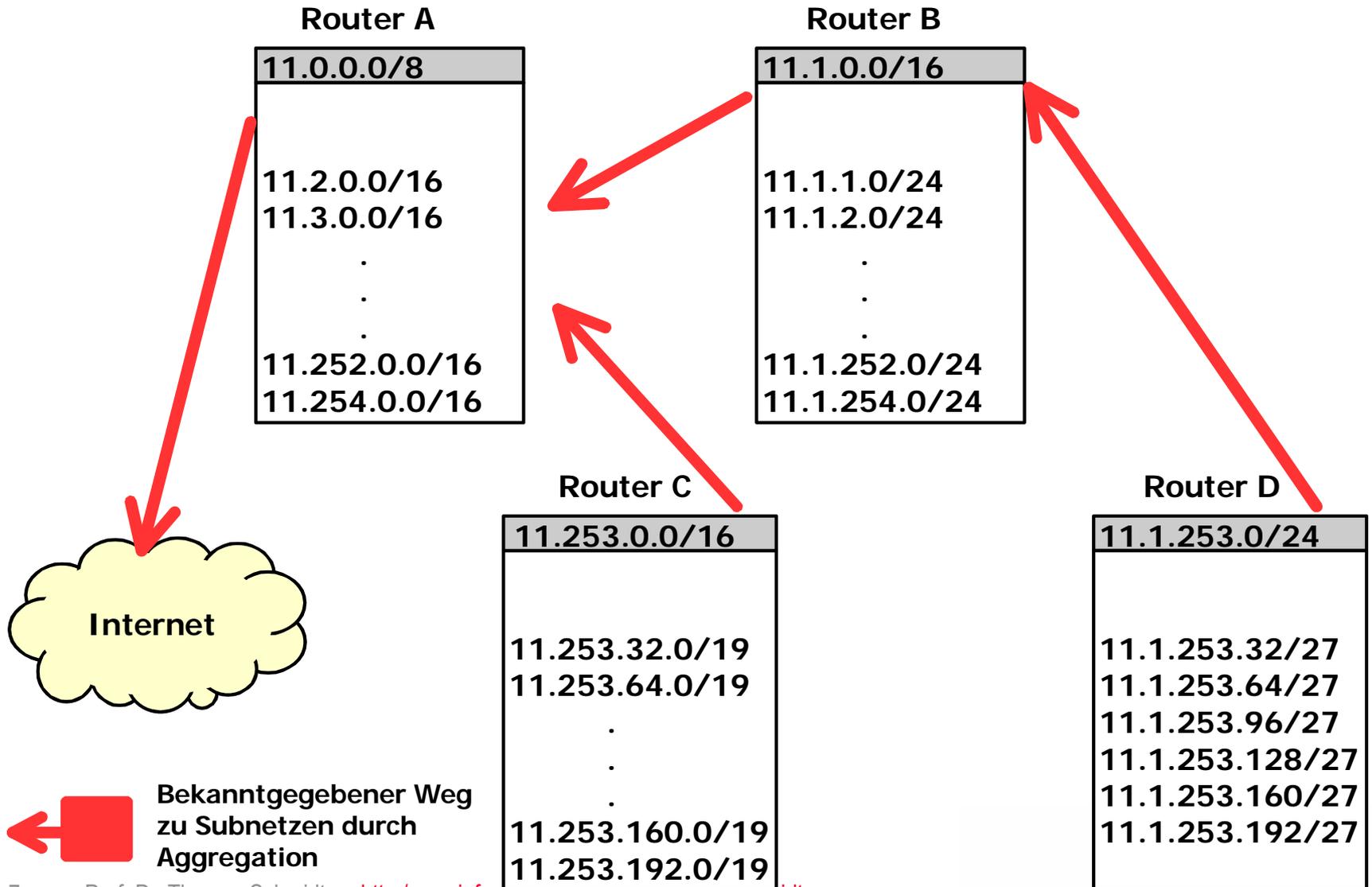
137.40.0.0/16	direct /137.40.0.1
137.10.0.0/16	direct /137.10.0.1
137.30.0.0/16	137.10.0.2
137.20.0.0/16	137.10.0.2

# 1. 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. Routenverdichtung durch VLSM



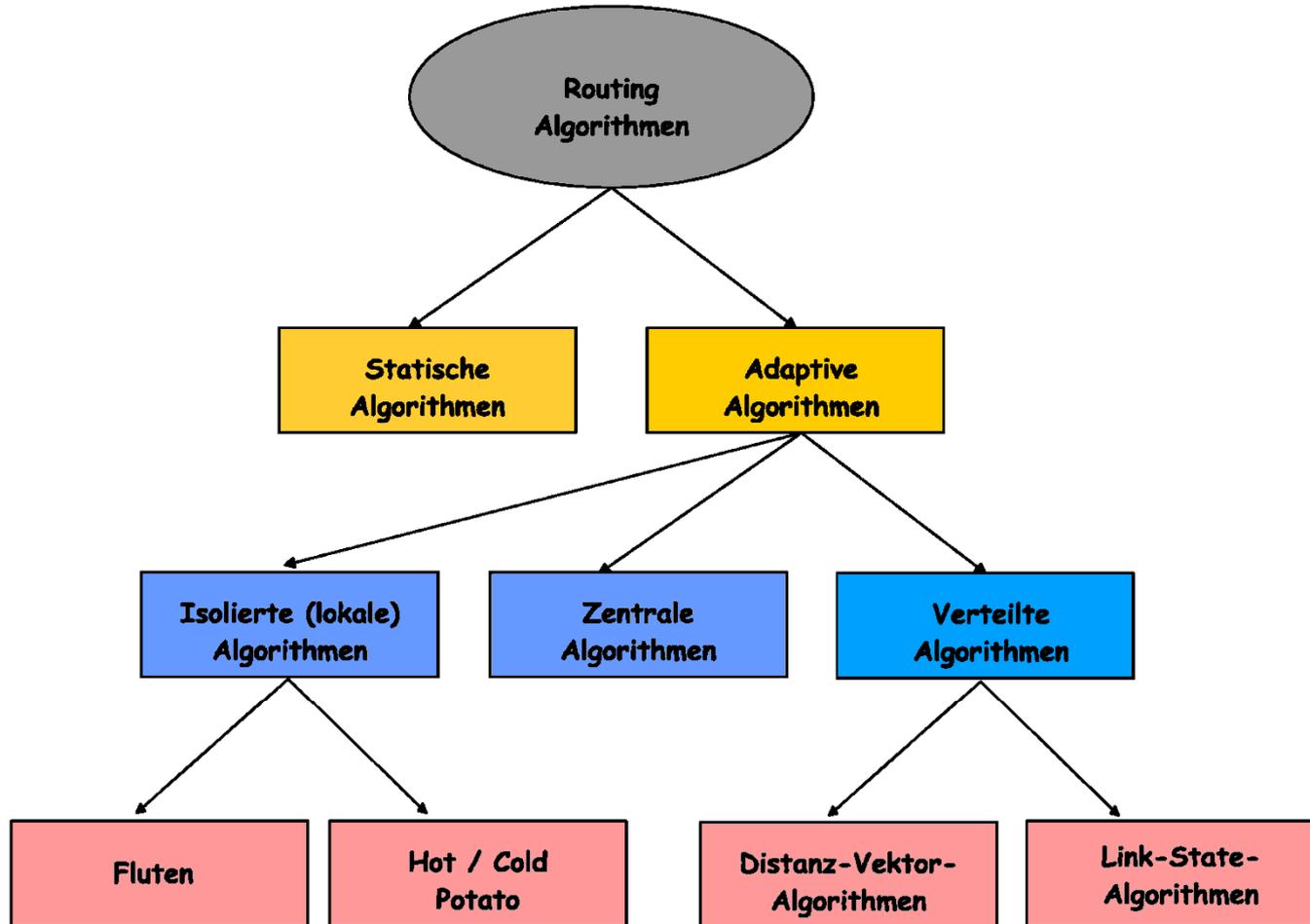
## 2. Dynamisches Routing

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 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.

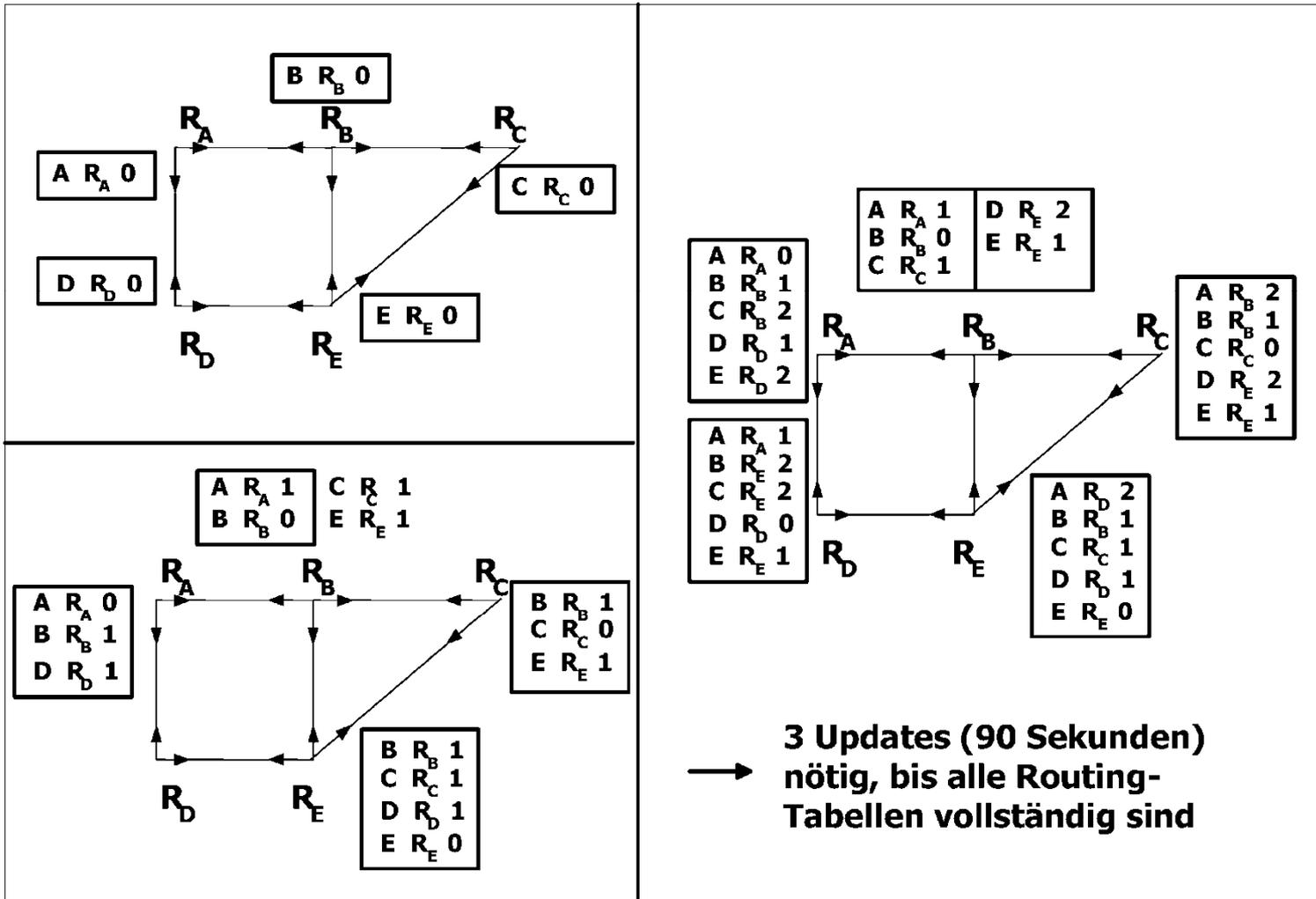


# 3. Beispiel: Routing Information Protocol (RIP)

- ▶ 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 ( $\infty := 16$ )

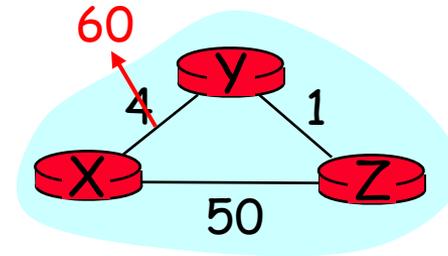


# 3. RIP Initialisierung

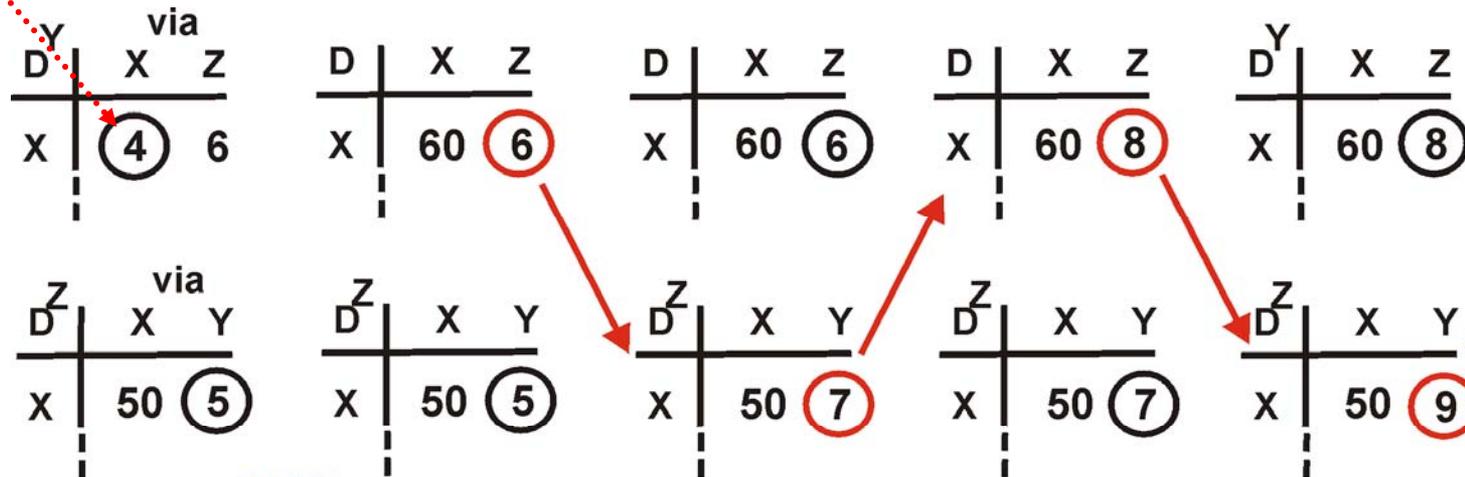


# 3. Link-Kosten Änderung

- good news travels fast
- bad news travels slow - "count to infinity" Problem!



Distanz von Y via X nach X

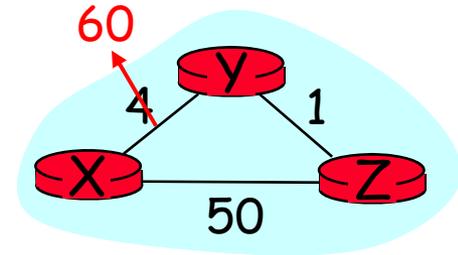


Algorithmus setzt sich fort

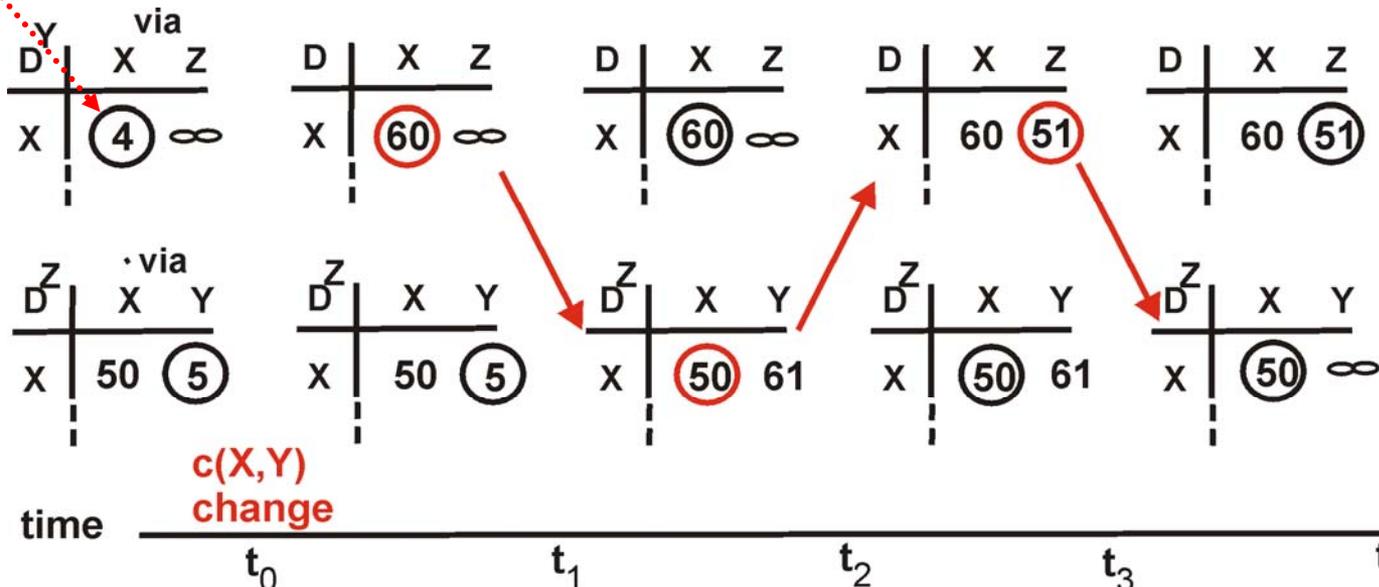
time  $c(X,Y)$  change  
 $t_0$   $t_1$   $t_2$   $t_3$   $t_4$

# 3. Optimierung: Vergiftete Rückwege (Poisonous Reverse)

- Z informiert Y, das seine (Z's) Entfernung zu X unendlich ist (um den Rückweg auszuschliessen)
- Eliminiert 2-Hop Loops



Distanz von Y via X nach X



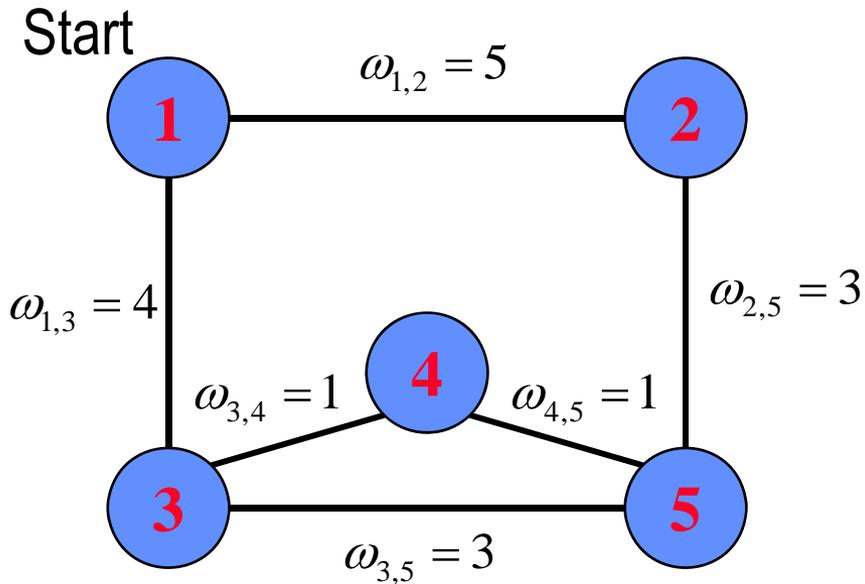
Algorithmus terminiert

# 3. 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  $i \notin E$ :

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

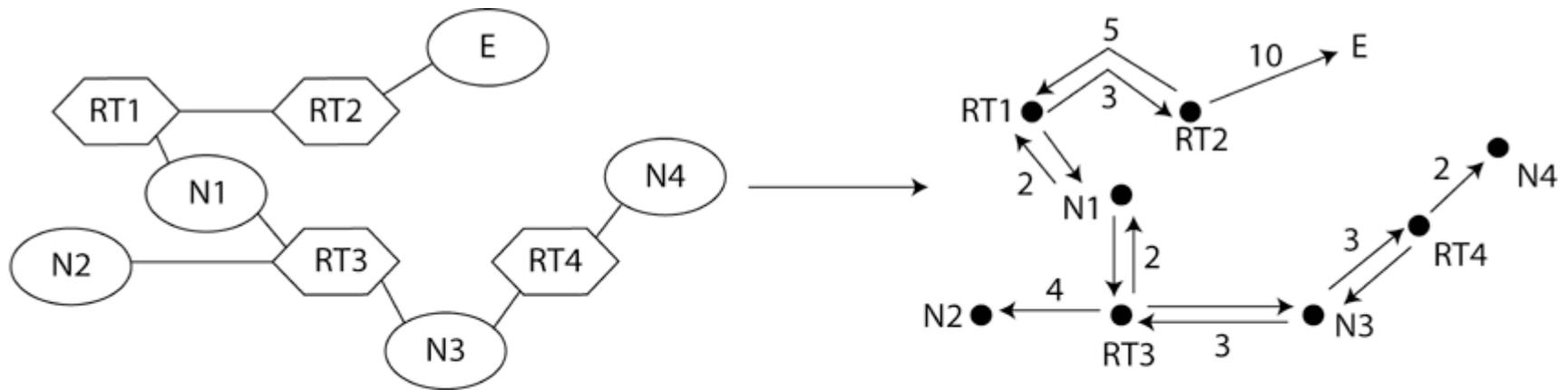


# 3. Beispiel: Open Shortest Path First (OSPF)

- ▶ 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.



# 3. OSPF Topologie

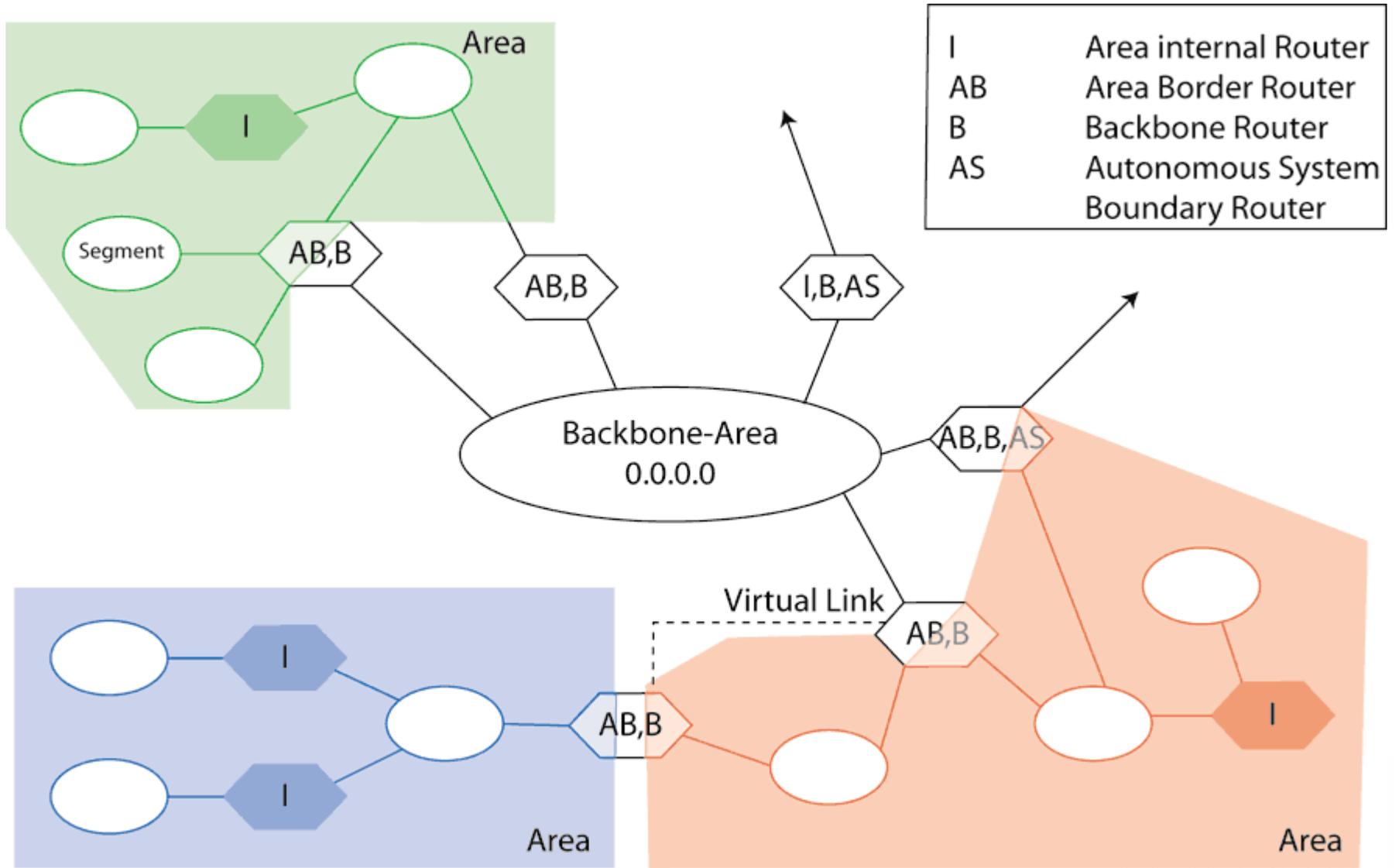


Routing Table at RT3:

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



# 3. OSPF Areas



# 4. 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).



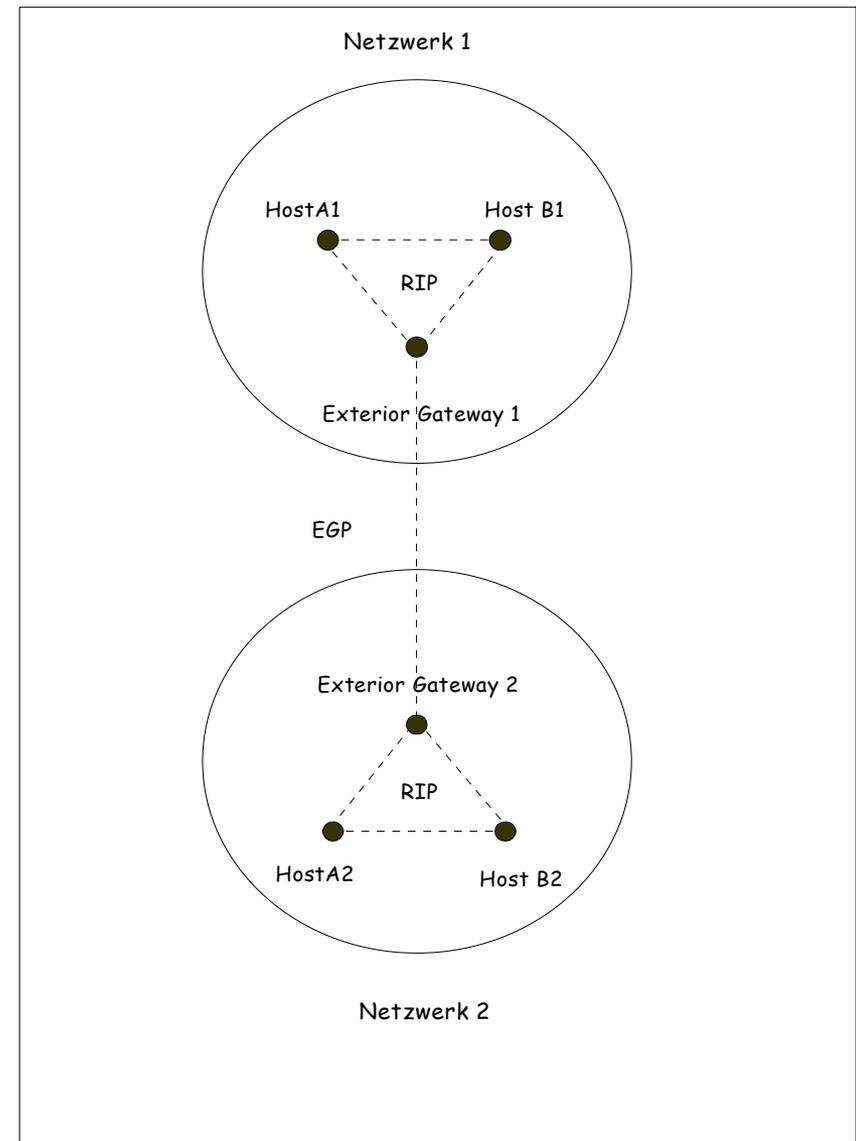
# 4. Exterior / Interior Gateway Protokolle

Exterior:

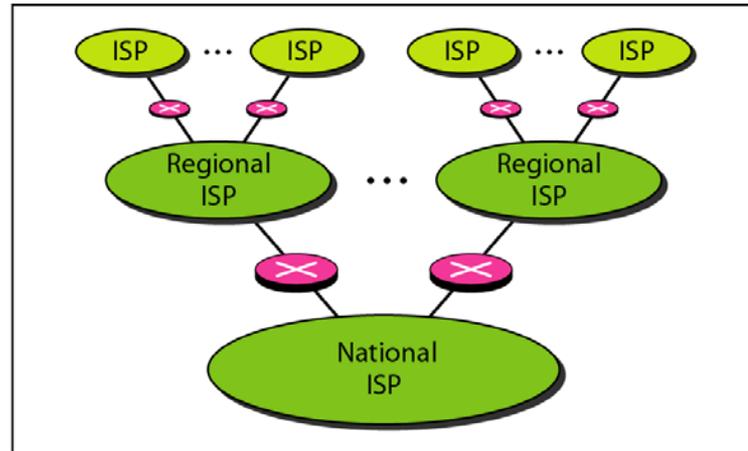
- EGP
- BGP

Interior:

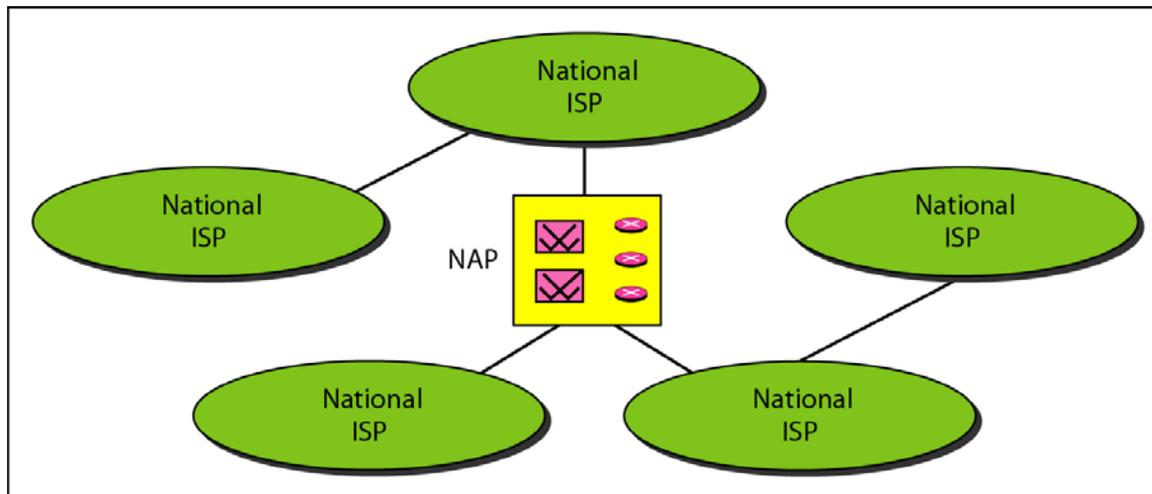
- RIP
- RIP-V2
- OSPF



# 4. Topologie-Hierarchien im Internet



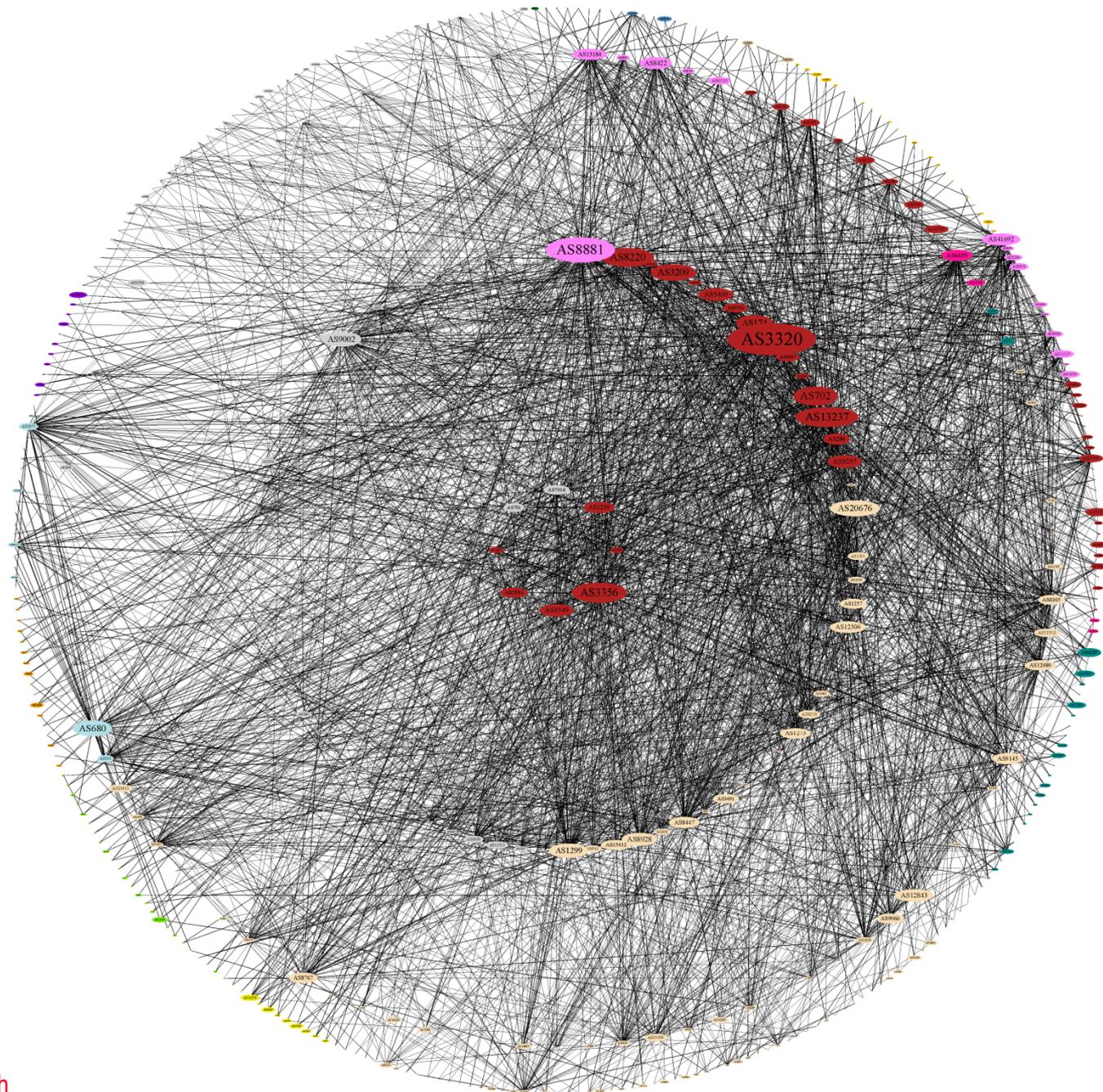
a. Structure of a national ISP



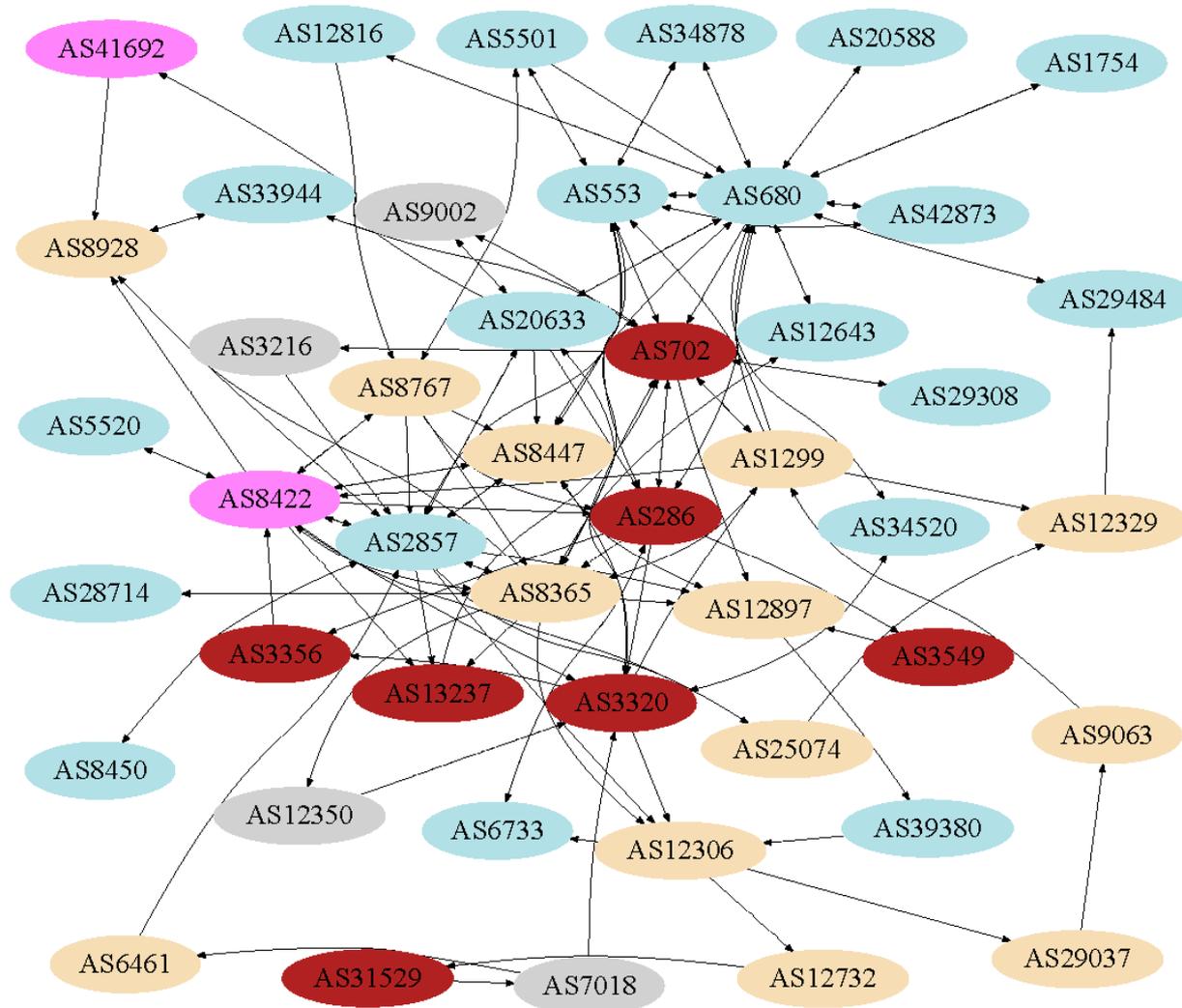
b. Interconnection of national ISPs



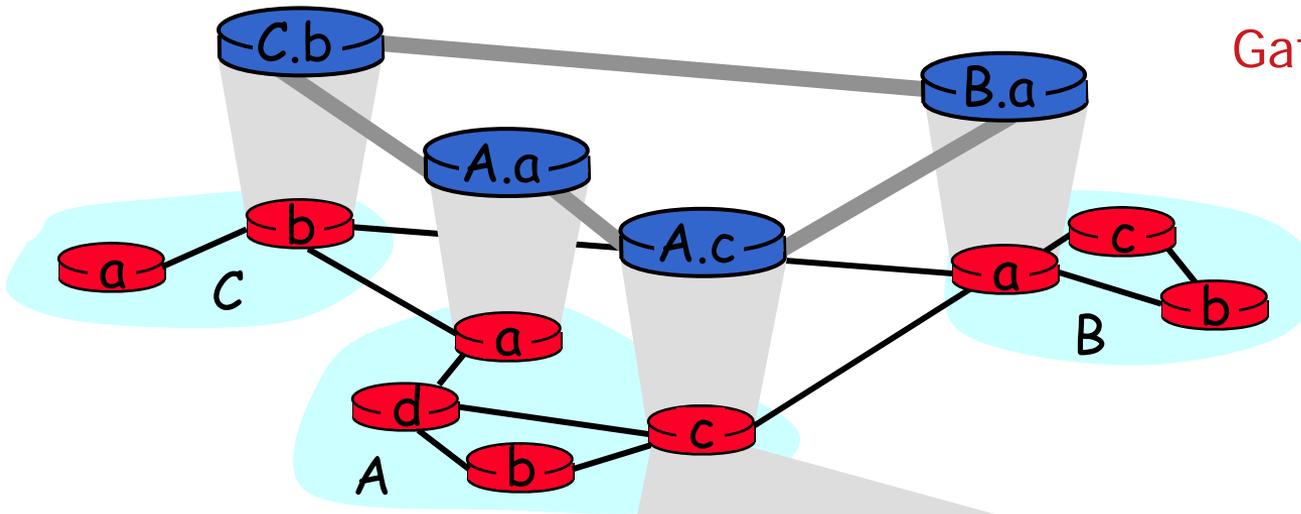
# Backbone Struktur des „Deutschen Internets“



# Branchenausschnitt



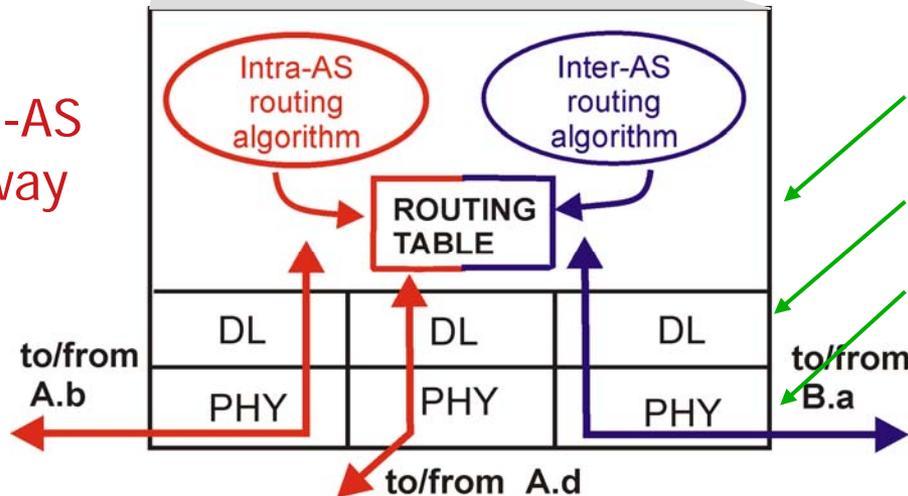
# 4. Intra-AS und Inter-AS Routing



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. 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“.



# 5. 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
- ▶ Alle teilnehmenden BGP-Router müssen eine vollständige (ortsabhängig verschiedene) RIB vorhalten



# 5. 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 Gruppen  
gruppiert Propagationsart



# 5. BGP4 Routing: Auswahlentscheidungen

BGP4-Router müssen über Pfadauswahl und Weiterleitung entscheiden:

## Phase I: Präferenz-Zuweisung

- Aufgrund lokaler Policies und Attributen 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 dann nach Eindeutigkeitsregeln.
- 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.



# 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. Worin liegt das Skalierungsproblem von BGP4?

