

# Routing

Michael Dienert

25. Oktober 2017

## Inhaltsverzeichnis

<b>1</b>	<b>Routing und Router</b>	<b>2</b>
1.1	Router . . . . .	2
1.1.1	Routersymbol . . . . .	2
1.2	Routing . . . . .	3
<b>2</b>	<b>Routingtabelle</b>	<b>4</b>
2.1	Beispiele für Routingtabellen . . . . .	4
2.1.1	Routngtabelle eines Rechners mit Linux-Kernel . . . . .	4
2.1.2	Routingtabelle eines Cisco-Routers . . . . .	4
2.2	Routingprinzipien . . . . .	5
2.3	Default route . . . . .	5
2.4	Routingablauf . . . . .	6
<b>3</b>	<b>Statische Routen</b>	<b>6</b>
<b>4</b>	<b>Zusammenfassung von Routen</b>	<b>7</b>
<b>5</b>	<b>Dynamisches Routing</b>	<b>7</b>
5.1	Routing-Protokolle . . . . .	7
5.2	Wichtige Routing Protokolle . . . . .	8
5.3	Einteilung der Routing-Algorithmen . . . . .	8
5.4	Ein paar Begriffe . . . . .	8
5.5	Routing-Algorithmen . . . . .	9
<b>6</b>	<b>Routing Information Protocol</b>	<b>9</b>
6.1	Funktionsweise von RIP . . . . .	10
6.2	Schwächen von RIP . . . . .	10
6.2.1	Count to Infinity . . . . .	10
6.2.2	Routing Loops . . . . .	11
<b>7</b>	<b>Open Shortest Path First</b>	<b>12</b>
7.1	Der Link-State-Prozess . . . . .	12

7.2	Funktion von OSPF . . . . .	13
7.3	Ein Beispiel . . . . .	14
7.4	OSPF in Multiaccess-Netzen . . . . .	15
7.4.1	DROthers . . . . .	16
7.4.2	Nachbarschaftsarten . . . . .	16
7.4.3	Austauschen der Link-State-Advertisements . . . . .	16
7.4.4	Skalierbarkeit . . . . .	16
7.5	Wichtige Parameter bei der Konfiguration von OSPF . . . . .	16
7.5.1	Intervalle . . . . .	16
7.5.2	Cost . . . . .	17
7.5.3	Adjacencies . . . . .	17
7.5.4	network-Kommando . . . . .	17
7.5.5	Auswahl der Router-ID: . . . . .	17
7.5.6	Designated Router und BDR . . . . .	18
<b>A</b>	<b>Aufgaben</b>	<b>18</b>

# 1 Routing und Router

## 1.1 Router

Router sind *Computer*, die IP-Pakete anhand der Layer3-Adresse (IP-Adresse) zwischen verschiedenen IP-Netzen vermitteln (Packet-Forwarding).

Der Router selbst kann ein ganz gewöhnlicher PC mit NT- oder Linux-Kernel oder eine spezielle Maschine mit einem für Routingaufgaben optimierten Betriebssystem sein. Auf jeden Fall enthält ein Router immer folgende Komponenten:

- CPU
- RAM
- ROM/Flash-ROM
- Betriebssystem

Der weltweit erste Router für das Advanced Research Projects Agency Network (ARPANET) war ein Honeywell 316 Minicomputer. Mit diesem Router wurde das ARPANET am 30. August 1969 ins Leben gerufen.

Obwohl der Honeywell 316 Minicomputer heisst, war er für heutige Verhältnisse riesig, etwa so gross wie eineinhalb 19 Zoll Schränke.

### 1.1.1 Routersymbol

Um Netzwerkläne mit Routern zeichnen zu können, hat Cisco ein Symbol für Router und Switches entworfen. Die Abb. 1 ist ein Beispiel für den Einsatz eines Routers zwischen zwei verschiedenen Netzen.

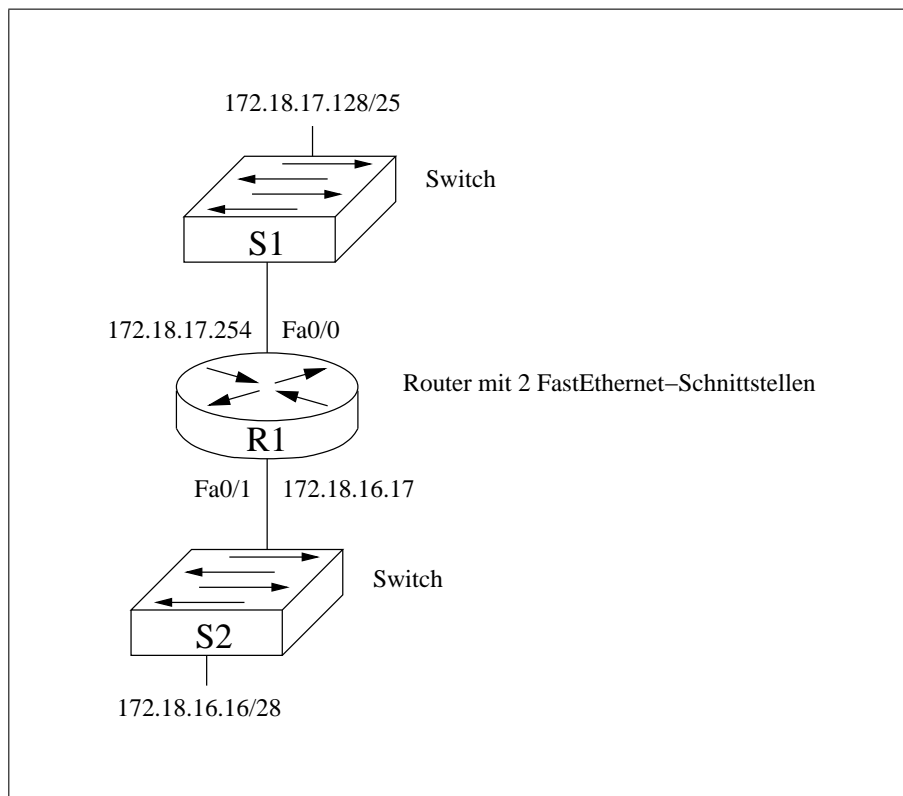


Abbildung 1: Ein Router verbindet zwei verschiedene Netze

## 1.2 Routing

Durch Routing werden verschiedene Netze verbunden. Das Routing findet dabei **ausschliesslich** auf Layer 3 statt:

Der Router entscheidet, in welches Netz er ankommende Daten weiterleiten soll anhand der Layer 3 IP-Zieladresse der ankommenden IP-Pakete.

Je nach physikalischer Schnittstelle (z.B. Ethernet oder seriell) treffen verschiedene Layer-2-**Frames** am Router ein. Der Router entnimmt den Frames die darin enthaltenen Layer-3-**IP-Pakete** und trifft dann die Routingentscheidung.

Bei seinem Weg durch das Internet wird ein Layer-3-Paket niemals verändert. Die Frames dagegen, die das Paket transportieren werden bei jedem Routingvorgang neu gebildet.

Natürlich hat jede Regel Ausnahmen: arbeitet ein Router mit *Net-Address-Translation* (NAT), werden auch die IP-Adressen des Layer3-Pakets verändert.

## 2 Routingtabelle

Die Routingtabelle ist eine Tabelle, die im RAM des Routers steht. Sie enthält Routingeinträge, meistens einfach **Routen** genannt.

Routen gibt es in drei Varianten:

- Route = Zielnetz + Next Hop
- Route = Zielnetz + Exit Interface
- Route = eigenes Netz

Hierbei gilt:

**Next Hop** entspricht dem sog. *Gateway*, also der IP-Adresse des **Nachbar**-Routers, der auf dem Weg zum Zielnetz liegt.

**Die IP-Adresse des Gateways muss im selben Netz wie der Router sein in dessen Routingtabelle das Gateway eingetragen wird., Kurz: das Gateway muss im eigenen Netz sein!**

**Exit Interface** anstelle eines Gateways kann man auf einem Router auch die Netzwerkschnittstelle angeben, über die man den *Next Hop* erreicht. Dabei muss aber sichergestellt sein, dass ein Paket vom *Exit Interface* eindeutig zum *Next Hop* geschickt wird. Gibt es vom *Exit Interface* aus gesehen mehrere nicht eindeutige Ziele, kann man einen solchen Routeneintrag nicht verwenden.

**eigenes Netz** die Routen in die eigenen Netze, also die Netze, in denen der Router selbst IP-Adressen hat werden nicht von Hand in die Routingtabelle eingetragen, sondern tauchen dort automatisch auf, sobald die IP-Adresse aus dem eigenen Netz auf eines der Interfaces des Routers konfiguriert wurde.

### 2.1 Beispiele für Routingtabellen

#### 2.1.1 Routngtabelle eines Rechners mit Linux-Kernel

```
root@xubbie:~# route
Kernel IP Routentabelle
Ziel          Router          Genmask         Iface
172.16.1.0    *              255.255.255.0   eth0
172.16.2.0    *              255.255.255.0   ppp0
172.16.3.0    *              255.255.255.0   ppp0
192.168.1.0   *              255.255.255.0   ppp1
192.168.2.0   192.168.1.1    255.255.255.0
```

#### 2.1.2 Routingtabelle eines Cisco-Routers

Folgende Tabelle ist ein Beispiel für die Routingtabellen auf Cisco-Routern. Der Router R1 hat IP-Adressen in 3 verschiedenen Netzen. Folglich gibt es in der Routingtabelle 3 Einträge für direkt verbundene Netze (am Zeilenanfang mit C gekennzeichnet).

Die beiden statischen Routen wurden einmal mit einem *Exit Interface* (Serial 0/0/0) und einmal mit der Adresse des *Next Hop* eingetragen.

Etwas verwirrend ist die Darstellung der Route mit dem Exit Interface: das Zielnetz wird auch als *directly connected* angegeben, obwohl es einen Hop entfernt ist.

Da sich Cisco noch nicht so recht von den alten Netzklassen A,B,C trennen kann, wird noch die Classful-Netzadresse der Zielnetze angegeben. In diesem Fall liegen 2 der eigenen (direkt verbundenen) Netze und ein Zielnetz im übergeordneten Klasse-B-Netz 172.16.0.0/16.

Die Angabe 172.16.0.0/24 soll angeben, dass die 3 Zielnetze die gemeinsame Maske /24 haben.

```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 3 subnets
C       172.16.1.0 is directly connected, FastEthernet0/0
C       172.16.2.0 is directly connected, Serial0/0/0
S       172.16.3.0 is directly connected, Serial0/0/0
C       192.168.1.0/24 is directly connected, Serial0/0/1
S       192.168.2.0/24 [1/0] via 192.168.1.1
R2#
```

## 2.2 Routingprinzipien

Beim Routing gelten folgende Prinzipien:

- Jeder Router trifft die Routingentscheidung autonom mit Hilfe einer Routingtabelle.
- Die Routingtabellen verschiedener Router in einem Netz müssen nicht die gleichen Einträge besitzen.
- Die Route von A nach B beinhaltet **nicht** automatisch die Route von B nach A.

## 2.3 Default route

Eine Default-Route ist ein Eintrag mit der Zielnetzadresse 0.0.0.0 und der zugehörigen Netzmaske 0.0.0.0.

Die Adresse 0.0.0.0 stellt sozusagen ein Jokerzeichen (Wildcard) für IP-Adressen dar.

## 2.4 Routingablauf

Folgende Schritte werden beim Routing durchgeführt <sup>1</sup> :

1. Empfängt ein Router ein IP-Paket, entnimmt er diesem als erstes die **IP-Zieladresse**.
2. die Zieladresse wird nun **bitweise** und *Schritt für Schritt* mit den einzelnen Zielnetzen der Routingtabelle verglichen. Eine **Übereinstimmung (Match)** besteht, wenn alle Netz-ID-Bits von Zielnetz und Ziel-IP identisch sind. Die Maske des Zielnetzes bestimmt dabei die Netz-ID-Bits.
3. Gibt es mehrere Übereinstimmungen, sendet der Router das Paket an das Zielnetz, bei dem von links betrachtet am meisten aufeinanderfolgende Bits übereinstimmen.
4. Gibt es zwei gleich lange Übereinstimmungen, wird die Route mit der besseren *Metrik* gewählt.
5. Eine Default-Route erzeugt **immer** eine Übereinstimmung, da die Maske der Default-Route den Wert 0 hat: 0 Bits müssen übereinstimmen.

## 3 Statische Routen

Statische Routen werden mit der Hand am Arm in die Routingtabelle eingetragen.

Vorteile statischer Routen:

- sicher, da fest eingetragen
- Routen sind vorhersagbar und konstant
- erzeugen nur geringe CPU-Last auf Router
- sind einfach zu verstehen

Nachteile statischer Routen:

- fehleranfällig beim Anlegen
- man muss die gesamte Topologie des Netzes kennen um die Route zu ermitteln
- zeitaufwändige Pflege: bei einer Topologieänderung des Netzes müssen alle statischen Routen neu eingegeben werden

Wann wendet man statische Routen an:

- Eintrag einer Default-Route
- Route in ein sog. *Stub*-Netz (Stub = Stummel), also ein kleines Netz, zu dem es nur einen Pfad gibt.
- innerhalb kleiner Netze, in denen sich die Einrichtung eines dynamischen Routingprotokolls nicht lohnt.

---

<sup>1</sup>Der Routingablauf ist hier etwas vereinfacht dargestellt. Bei Cisco-Routingtabellen wird noch mit einer Hierarchie innerhalb der Routingtabelle gearbeitet. Hierbei wird zwischen Level 1 und Level 2 Routes z.B. Parent- und Child-Routes unterschieden.

## 4 Zusammenfassung von Routen

Folgende drei Routen haben alle ein gemeinsames Exit-Interface **und** die Zielnetze liegen alle im IP-Bereich 172.16.1.0 bis 172.16.3.255:

```
S      172.16.1.0/24 is directly connected, Serial0/0/1
S      172.16.2.0/24 is directly connected, Serial0/0/1
S      172.16.3.0/24 is directly connected, Serial0/0/1
```

Erweitert man diesen Bereich auf 172.16.0.0 bis 172.16.3.255, kann man den Bereich auch als Netz so angeben:

```
172.16.0.0/22
```

Entsprechend dem Begriff **Subnet** (bei der Aufteilung in kleinere Netze) nennt man das übergeordnete Netz auch **Supernet**.

Anstelle nun 3 einzelne, statische Routen zu konfigurieren, kann man nun alle Routen *zusammenfassen*:

```
S      172.16.0.0/22 is directly connected, Serial0/0/1
```

Beim Zusammenfassen von Routen (route summarization) sucht man immer das kleinstmögliche übergeordnete Netz, das alle Zielnetze, die zusammengefasst werden sollen enthält.

Dabei darf aber kein Subnetz des übergeordneten Netzes als Zielnetz in einer anderen Route auftauchen. Im obigen Beispiel darf es also keine Route mit dem Zielnetz 172.16.0.0/24 geben, das über ein anderes Interface erreicht wird.

Eine andere Möglichkeit die Netze zusammenzufassen wäre diese:

```
S      172.16.1.0/24 is directly connected, Serial0/0/1
S      172.16.2.0/23 is directly connected, Serial0/0/1
```

Hier werden nicht alle Subnetze in einem Supernetz zusammengefasst, dafür wird aber kein Bereich unnötig belegt.

## 5 Dynamisches Routing

Beim *Dynamischen Routing* werden die Einträge in den Routingtabellen von einem Routingprotokoll automatisch aktualisiert und gepflegt.

Dabei ist es völlig in Ordnung, wenn eine Routingtabelle gleichzeitig statische und dynamische Routen enthält!

### 5.1 Routing-Protokolle

Routing Protokolle ermöglichen es den Routern Informationen über *entfernte Zielnetze* auszutauschen. Dabei sind entfernte Zielnetze Netze, in denen der Router keine Netzwerkschnittstelle hat, die also nicht *Directly Connected* sind, sondern über mindestens einen weiteren Router erreicht werden.

Aufgaben der Routing Protokolle:

- Ermittlung des *besten Pfads* ins Zielnetz

- Anpassung der Routingtabellen bei *Topologieänderung*
- Entdeckung entfernter Netze

Allgemein gesagt, besteht ein Routing Protokoll aus folgenden Teilen:

**Datenstrukturen:** das sind Tabellen, also Datenbanken im RAM des Routers. Mit Tabellen sind hier *nicht* die Routingtabelle gemeint.

**Algorithmen:** zur Berechnung des besten Pfads, der Netztopologie und zum Aufbau der Datenstrukturen

**Protokoll-Mitteilungen (Messages):** zur Kommunikation der Router untereinander

## 5.2 Wichtige Routing Protokolle

Wichtige Routing Protokolle sind:

RIP - Routing Information Protocol; mehrere Versionen RIPv1, RIPv2, RIPng; RIP ist ein sog. Distanz-Vektor-Protokoll und wurde 1969 (!) für das ARPANET entwickelt; RIP eignet sich nur für kleine Netze.

(E)IGRP - (Enhanced) Interior Gateway Routing Protocol; Cisco - Eigenentwicklung.

OSPF - Open Shortest Path First; ein freies Link-State-Routingprotokoll; OSPF hat RIP abgelöst und ist heute Standard in LANs.

IS-IS - Intermediate System-to-Intermediate System; ähnlich OSPF

BGP - Border Gateway Protocol; einziges Exterior Gateway Protocol: Routing zwischen Autonomen Systemen (s.u.).

## 5.3 Einteilung der Routing-Algorithmen

### 5.4 Ein paar Begriffe

**Autonomous System** Ein Autonomes System (AS) ist eine Ansammlung von IP-Netzen, deren *internes Routing* von einer Institution (z.B. einer Firma, einer Universität, einem Service Provider) verwaltet wird. Jedes AS erhält auf Antrag von der Internet Assigned Numbers Authority (IANA) bzw. deren regionale Unterorganisation eine Schlüsselnummer.

Der Betreiber eines AS ist also für das *Intra-AS-Routing* zuständig.

**Konvergenz** Konvergenz in einem Netz ist erreicht, wenn nach einer Änderung alle Routing-Tabellen in einem festen, aktuellen Zustand sind. Je schneller dieser Zustand erreicht ist, desto besser ist das eingesetzte Routing Protokoll.



## 5.5 Routing-Algorithmen

Die Routing-Algorithmen lassen sich folgendermassen einteilen:

**Intra-AS-Routing** - Routing *innerhalb* eines Autonomen Systems. Alle diese Routing-Algorithmen gehören zu den **Interior-Gateway-Protocols, IGP**.

Die IGP werden wiederum in zwei Hauptgruppen unterteilt:

**Link-State** bei den Link-State-Protokollen kennt jeder Router die **Topologie** des Netzes. Diese wird durch Weitergabe der Verbindungsdaten eines Routers zu seinen Nachbarn (Link-States) an alle Router im Netz, erlernt.

Die CPU des Routers errechnet aus der Netztopologie die jeweils beste Route ins Ziel.

Wichtige Vertreter der Link-State-Protokolle:

- OSPF - Open Shortest Path First
- IS-IS - Intermediate-System to Intermediate-System

Link-State-Protokolle konvergieren schnell.

**Distanzvektor** Bei den Distanzvektor-Protokollen kennt ein Router immer nur seine direkten Nachbarn, aber nicht die Netztopologie. Der Name Distanzvektor hat folgenden Ursprung:

**Distanz** - wie weit ist das Zielnetz entfernt? Bei RIP (s.u.) wird die Entfernung ins Zielnetz mit der Anzahl der Hops angegeben.

**Vektor** - in welcher Richtung liegt das Zielnetz, d.h. über welches physikalische Interface (z.B. Ethernetschnittstelle) führt ein Weg dorthin.

Wichtige Vertreter der Distanzvektor-Protokolle:

- RIPv1, RIPv2, RIPnG - RIP = Routing Information Protocol
- IGRP / EIGRP (Enhanced) Interior Gateway Routing Protocol

Die RIP-Varianten konvergieren relativ langsam. EIGRP nimmt eine Sonderstellung ein, da es schnell konvergiert, obwohl es ein Distanzvektor-Protokoll ist.

**Inter-AS-Routing** - Routing *zwischen* Autonomen Systemen. Diese Algorithmen gehören zu den **Exterior Gateway Protocols, EGP**

Der einzige Vertreter der EGPs ist das **Border Gateway Protocol, BGP**. Das BGP arbeitet mit einer allgemeinen Form von Distanzvektoren. Der Algorithmus wird **Pfadvektor-Algorithmus** genannt.

## 6 Routing Information Protocol

Das Routing Information Protocol ist das älteste Routingprotokoll überhaupt und seine Anfänge gehen zurück bis ins Jahr 1969!

Eine Implementierung von RIP auf BSD Maschinen gab es ab 1982, das Programm dazu heisst routed (route-dee). Mit routed wurde RIP IP-tauglich.

1988 wurde RIP in der RFC 1058 standardisiert.

## 6.1 Funktionsweise von RIP

- jeder beteiligte Router besitzt eine Routing-Tabelle, in der **jedes** mögliche Zielnetz innerhalb des AS enthalten ist.

Dabei enthält jeder Tabelleneintrag die Adresse des Zielnetzes, die Anzahl der Hops bis ins Ziel (die sog. Metrik) und ein Exit-Interface oder die IP-Adresse des nächsten Routers Richtung Ziel (Gateway).

- jeder Router versendet periodisch (alle 30s) **Updates** an seine *direkten Nachbarn*. Die Updates werden bei RIPv1 als UDP-Broadcast an Port 520 verschickt, bei RIPv2 und RIPnG als Multicast an die Adresse 224.0.0.9, ebenfalls als UDP-Datagramm.

Jeder Update enthält die Zielnetzadressen aus der Routingtabelle und die Metrik.

RIPv1 Updates enthalten **keine** Netzmasken! Es können somit nur Classful-Netze geroutet werden. RIPv2 Updates enthalten die Masken der Ziele, somit können auch Netze mit variabler Maske geroutet werden. RIPnG ist geeignet für IPv6-Netze

Jeder Update enthält die Zielnetzadressen aus der Routingtabelle und die Metrik.

- jeder Nachbarrouter verarbeitet die eintreffenden Updates nach folgender Regel:
  - enthält der Update eine **bessere** Route in ein bestimmtes Ziel, als in der eigenen Routingtabelle enthalten ist, trage die neue, bessere Route in die Routingtabelle ein.
  - ist die im Update enthaltene Route **schlechter** als die in der Routingtabelle, trage sie nur ein, wenn das Update über das Interface hereinkommt, über das auch die ursprüngliche Route gelernt wurde.

## 6.2 Schwächen von RIP

### 6.2.1 Count to Infinity

Bei einer Netzwerkunterbrechung kann es zum Count to Infinity Problem kommen: wenn bei Router R2 die Leitung ins Zielnetz ausfällt, wird das Interface fa0/1 heruntergefahren. Damit verschwindet auch die Route ins direkt verbundene Netz Z aus der Routingtabelle.

Beim nächsten RIP-Update lernt R2 die Route nach Z von R1 - aber mit dem falschen Interface und der Metrik 2 (2 Hops).

Beim übernächsten Update lernt R1 von R2 die Route nach Z und *überschreibt die ursprüngliche* Route, da diese *ebenfalls* von R1 gelernt wurde.

Dieses gegenseitige Überschreiben der Routeneinträge würde sich ohne Gegenmassnahmen bis in alle Ewigkeit fortsetzen.

Um das Problem zu lösen wurde festgelegt:

Sobald die Metrik den Wert 16 erreicht, gilt ein Netz als <b>nicht erreichbar</b> .
-------------------------------------------------------------------------------------

Da die Routingupdates alle 30s versandt werden, ist das Netzwerk nach etwa 4 min wieder konvergiert (8 Updatezyklen).

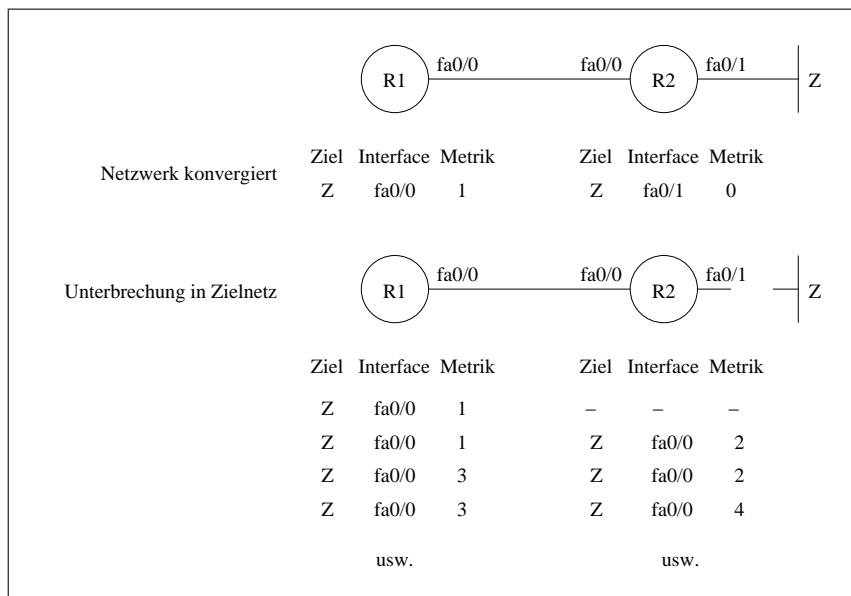


Abbildung 2: Count-to-Infinity

## 6.2.2 Routing Loops

**Split Horizon** - ein Router darf eine Route nicht "zurück"-verbreiten, d.h. er darf sie nicht über das Interface propagieren, über das er sie gelernt hat.

**Route Poisoning** - wenn ein Pfad unerreichbar wird, wird der entsprechende Eintrag in der Routing-Tabelle nicht sofort gelöscht, sondern als unerreichbar markiert, indem seine Metrik auf 16 gesetzt wird.

In der Routing-Datenbank, die ein Router evtl. noch zusätzlich zur Routingtabelle verwaltet, bleibt aber ein Eintrag mit der ursprünglichen Metrik bestehen, um die Route schnell wieder herstellen zu können.

**Split-Horizon-With-Poison-Reverse** - entgegen der Split-Horizon-Regel wird die Route mit der Metrik 16 wieder zurückpropagiert, also zu dem Interface rausgeschickt, von dem sie gelernt wurde. Damit wird die Rückroute, die zu einer Schleife führen würde blockiert (blockierte Rückroute).

**Triggered Update** : Triggered Update wird sofort versendet, wenn:

- ein I/F auf dem Router wechselt von up nach down oder umgekehrt
- eine Route wird als unerreichbar markiert oder von unerreichbar wieder nach erreichbar gesetzt
- eine (statische oder dynamisch erlernte) Route wird in der Routing-Tabelle eingetragen

Im Prinzip lösen Triggered Updates das Routing-Loop-Problem vollständig, sie können aber verloren gehen und verbreiten sich nicht schnell genug in einem grossen Netz. Ausserdem bleibt das Problem mit dem Wackelkontakt am Netzwerkstecker.

**Update Timer** : RIP arbeitet mit **periodischen Updates**: alle **30s** versendet ein RIP-Router die gesamte Routingtabelle an seine Nachbarn.

**Invalid Timer** : **180s**, trifft für eine Route innerhalb dieser Zeit kein Update ein, wird sie durch *Setzen der Metrik auf 16* als *unerreichbar markiert* und nach Ablauf des Flush-Timers (also 60s später) komplett gelöscht.

**Flush Timer** : **240s**, Invalid und Flush Timer werden gleichzeitig gestartet.

**Holddown Timer** : **240s**; Ablauf:

- Router empfängt Triggered Update über ausgefallene Route
- Router startet Holddown Timer und merkt sich die Route als *eventuell unerreikbaar, belässt sie aber in der Routingtabelle* (mit ihrer ursprünglichen Metrik) und stellt auch Pakete dorthin zu
- während der Holddown-Zeit **ignoriert** der Router alle Updates die eine **gleiche oder schlechtere** Metrik wie die markierte Route haben. Kommt ein Update mit **besserer** Metrik herein, wird die markierte Route wieder aus dem Holddown-Zustand entfernt und auf den aktuellen Stand gebracht.

## 7 Open Shortest Path First

OSPF ist der wichtigste Vertreter der Link-State-Protokolle. Es ist mit der RFC 2328 spezifiziert und ist zum Standard-Routing-Protokoll innerhalb autonomer Systeme (AS) geworden <sup>2</sup> .

Die wichtigsten Merkmale von OSPF sind:

**Link-State-Protokoll** - OSPF ist ein Link-State-Protokoll. Jeder Router kennt die Topologie des Netzes, die mit Link-State-Advertisements (LSA, s.u.) verbreitet wird.

**OSPF kennt keine Schleifen** - bei OSPF gibt es keine Routing-Loops, der verwendete Algorithmus (Shortest Path Algorithmus) berechnet jede Route in ein Zielnetz schleifenfrei

**Link-State-Datenbank** - bei OSPF besitzt jeder Router eine Karte des Netzes. Diese Karte ist in einer Datenbank abgelegt, die genau so viele Einträge enthält, wie es Verbindungen zwischen zwei Routern (Links) im Netzwerk gibt. Da jeder Router diese DB besitzt, kann er unabhängig von den anderen Routern mit dem Shortest-Path-First-Algorithmus <sup>3</sup> zu jedem Ziel im Netz die optimale Route berechnen.

### 7.1 Der Link-State-Prozess

Aus der Sicht eines Routers läuft die Bildung der Link-State-Datenbank so ab:

<sup>2</sup>OSPF gibt es seit 1989 und die zugehörige RFC 1131 ist inzwischen überholt.

<sup>3</sup>Der Shortest-Path-First-Algorithmus wurde von Edsger W. Dijkstra 1959 erfunden. Dijkstra war holländischer Informatikprofessor und lebte von 1930 bis 2002. Er erhielt 1972 den Turing-Award und ihm haben wir neben OSPF zu verdanken, dass gute, moderne Programmiersprachen keine GOTO-Befehl mehr kennen.

**Lerne deine eigenen Anbindungen** - das heisst:

- welche Interfaces sind Up-and-Up?
- was sind meine Links zu den direkten Nachbarn?

**Welchen Status haben die Verbindungen zu den direkten Nachbarn** - Tausche **Hello-Packets** mit deinen direkten Nachbarn aus

- lerne deine Link-States

**Baue ein Link-State-Paket für jeden Link** - jedes Link-State-Paket enthält:

- Die Nachbar-Router-ID = Neighbor-ID
- IP-Adresse und Maske des Links
- Bandbreite = Bandwidth des Links und die "Kosten" des Links
- Layer-2-Technologie (z.B. Ethernet, Frame-Relay oder PPP)
- Link-Typ (z.B. Punkt-zu-Punkt-Verbindung zu anderem Router, Anschluss in ein Subnetz (Stub-Network), Verbindung über Multi-Access-Network)

**Flute das Netz mit deinen Link-State-Packets** - sende die Link-State-Packets an alle direkten Nachbarn. Die hierzu verwendeten Nachrichten nennt man **Link State Advertisement LSA**.

Die Empfänger der **LSAs** speichern eine Kopie davon und senden sie dann weiter an die übernächsten Nachbarn. Dieses Fluten des Netzes setzt sich so fort, bis jeder Router im Netz meine Link-State-Packets erhalten hat.

**Link-State-Datenbank** - Da alle Router ihre LSAs versenden, erhalten nach einer Weile **alle** Router Pakete über **alle Links im Netz** und bauen daraus eine **Link-State-Datenbank** auf.

**Shortest-Path-Algorithmus** - aus den Informationen in der Link-State-Datenbank berechnet jeder Router selbstständig

- die Topologie des Netzes
- den besten Pfad für **jedes** Zielnetz

## 7.2 Funktion von OSPF

- jeder OSPF-Router erzeugt eine **Nachbar-Tabelle**, in der alle seine direkten Nachbarn eingetragen sind.
- allen Nachbarn werden alle Verbindungen (Links-States) zu den anderen Nachbarn mitgeteilt (mit **Link-State-Packages, LSP**, werden in den **Link-State-Datenbanken** auf jedem Router gespeichert und an die übernächsten Nachbarn weitergeschickt. Diese speichern wiederum in der LS-DB und schicken das LSP weiter, .... Nach einer Weile haben alle Router eine identische Link-State-Datenbank, die die **Topologie des Netzes** beschreibt.
- Aus der Link-State-DB wird mit dem SPF-Algorithmus auf jedem Router zu jedem Ziel im Netz die beste Route berechnet. Dabei werden die Pfadkosten summiert und der Pfad mit den geringsten Kosten als Route gewählt.

- Multicast-Adressen:

**Hello-Packets** : 224.0.0.5

**Designated Router sendet LSA an DROthers** : 224.0.0.5 . Die genaue Beschreibung von Designated Router und Backup Designated Routers folgt in Abschnitt 7.4.

**DROthers senden LSAs an DR und BDR** : 224.0.0.6

### 7.3 Ein Beispiel

An einem Beispiel soll der Routingalgorithmus nachvollzogen werden. Die Beispieltopologie ist in Abb. 3 dargestellt.

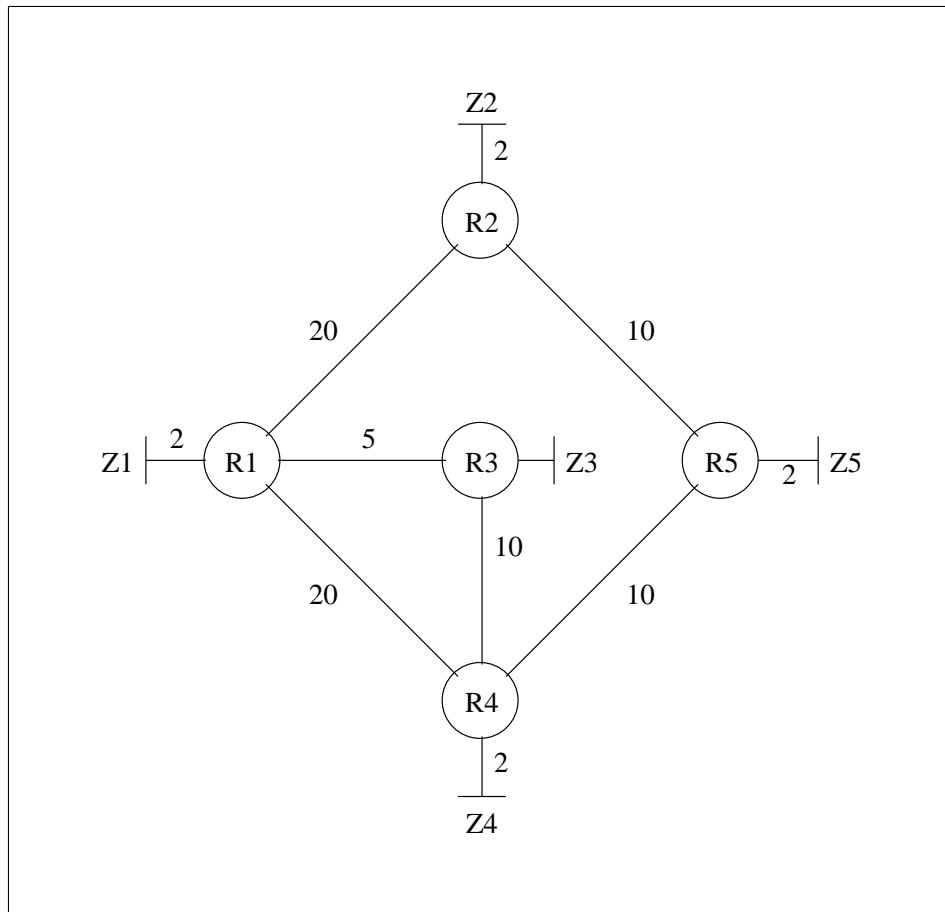


Abbildung 3: Eine Beispiel-Netztopologie

Nachdem alle Router ihre LSAs ausgetauscht haben, besitzt jeder Router folgende Link-State-Datenbank:

Nun soll der beste Weg von Z1 nach Z5 ermittelt werden:

Von	Nach	Kosten
R1	Z1	2
R1	R2	20
R1	R4	20
R1	R3	5
R2	R5	10
R2	Z2	2
R3	R4	10
R3	Z3	2
R4	R5	10
R4	Z4	2
R5	Z5	2

Tabelle 1: Eine Link-State-Datenbank

Der Algorithmus berechnet von R1 ausgehend alle Pfade ins Ziel und summiert dabei die Pfadkosten auf. Treffen sich zwei Pfade an einem Router, wird der schlechtere Pfad verworfen. Pfade:

```
Z1 - R1 - R2 - R5 - Z5
    |       |       |       |
    2       22      32      34
```

```
Z1 - R1 - R3 - R4 - R5 - Z5 bester Pfad gefunden
    |       |       |       |       |
    2       7       17      27      29
```

```
Z1 - R1 - R4 Pfad verwerfen
    |       |
    2       22
```

## 7.4 OSPF in Multiaccess-Netzen

Ein Multiaccess-Netz ist ein Netz, in dem mehr als zwei Teilnehmer über ein gemeinsames Medium, das alle verwenden dürfen, verbunden sind. Ursprünglich war dieses Medium der Ethernet-Bus mit dem CSMA/CD-Verfahren<sup>4</sup>, das sichergestellt hat, dass jeder mit jedem beliebig kommunizieren kann. Inzwischen wurde der Bus durch Switches abgelöst, der *Multiple Access* blieb dabei natürlich erhalten.

Sind nun mehrere Router über einen Bus, einen Hub oder einen Switch miteinander verbunden, entsteht folgendes Problem: alle Router am Bus/Switch sind *Nachbarn zueinander!* Somit entstehen sehr viele, z.T. für das Routingprotokoll unnötige, redundante Nachbarschaften (Adjacencies): jeder mit jedem.

Tauschen nun alle Router mit allen Nachbarn LSAs (Link State Advertisements) aus (fluten des Netzes mit LSAs), entsteht eine unnötig hohe Netzlast und ein Grossteil der ausgetauschten Informationen kommt *mehrfach in verschiedenen LSAs* vor.

Um diese Redundanz zu vermeiden, wird ein sog. **Designated Router** (DR) gewählt. Die Wahl erfolgt beim Austausch der Hello-Packets: die Hello-Packets enthalten die

<sup>4</sup>(CSMA = Carrier Sense Multiple Access with Collision Detection)

ID jedes Routers (RID). Der Router mit der höchsten Router-ID wird nun zum DR gewählt, der mit der zweithöchsten zum Backup-DR (BDR).

Die Regeln für die Bestimmung der RID stehen in Kap. 7.5.5.

Eine solche Wahl findet nur statt, wenn es im Netzwerk weder DR noch BDR gibt. Gibt es bereits einen DR und BDR und wird ein neuer Router hinzugefügt, der eine noch höhere RID hat, findet keine neue Wahl statt. Gewählt wird erst wieder, wenn der DR ausfällt: dann übernimmt der BDR dessen Funktion und der BDR wird neu gewählt.

#### 7.4.1 DROthers

Alle anderen Router, die weder DR noch BDR sind nennt man DROthers.

#### 7.4.2 Nachbarschaftsarten

Nach der Wahl von DR und BDR hat man folgende Nachbarschaftsbeziehungen:

**Full Adjacency** - zwischen DROTHERS ↔ DR, BDR und DR ↔ BDR besteht eine **Full-Adjacency**, eine vollwertige Nachbarschaft

**2-Way Adjacency** - zwischen zwei DROthers besteht nur eine 2-Way-Adjacency.

#### 7.4.3 Austauschen der Link-State-Advertisements

Sind DR und BDR gewählt, senden die DROthers ihre LSAs nur noch an den DR und den BDR, aber nicht mehr an die anderen Router. Hierzu verwenden sie die Multicast-Adresse 224.0.0.6.

Der DR verteilt daraufhin die LSAs an alle anderen Router und verwendet dazu die Multicast-Adresse 224.0.0.5.

Das heißt, das Fluten des Netzes mit LSAs wird nun nur noch von einem Router durchgeführt.

Die DROthers tauschen untereinander weiterhin Hello-Packets über die Multicast-Adresse 224.0.0.5 aus.

#### 7.4.4 Skalierbarkeit

Durch die Wahl eines DR skaliert ein OSPF-Netzwerk wesentlich besser. Hier ein Beispiel:

Zu einem Netzwerk mit 15 Routern ohne DR kommt ein 16-ter Router hinzu: es entstehen 16 neue Nachbarschaften die jeweils zum Fluten des Netzes mit den neuen LSAs führen.

Ist ein DR im Netz entsteht beim Hinzufügen eines weiteren Routers nur eine neue Nachbarschaft und es kommt nur zu einem Flutungsvorgang, der vom DR ausgeht.

### 7.5 Wichtige Parameter bei der Konfiguration von OSPF

#### 7.5.1 Intervalle

**Hello-Intervall** : Takt, in dem Hello-Pakete gesendet werden; 10s bei Ethernet + Serial; 30s bei FrameRelay



**Dead-Intervall** : Bleiben Hello-Pakete länger als diese Zeit aus, wird der Nachbar als 'down' betrachtet; bei Cisco gilt:  $DI = 4 \cdot HI$  ; also 40s bzw. 120s bei FrameRelay

### 7.5.2 Cost

Berechnung der Link-Kosten aus der Bandbreite in kBit/s (auf Cisco-Routern):

$$cost = 10^8 / BW$$

z.B.:  $cost = 10^8 / 64000 = 1562$

Referenzwert ist hier 100Mbit/s =  $10^8$ ; dieser kann verändert werden (siehe unten). Die Link-Kosten können auch direkt gesetzt werden.

### 7.5.3 Adjacencies

Damit zwei Router Nachbarn werden können, muss gelten:

- die Subnetzmasken + Netzadressen der jeweiligen Verbindungen zu den Nachbarn müssen gleich sein
- die Hello-Intervalle müssen gleich sein
- die Dead-Intervalle müssen gleich sein
- die Network-Types müssen gleich sein (Broadcast Multi-Access/NBMA)
- die Area-IDs müssen gleich sein

### 7.5.4 network-Kommando

- Beim ersten network-Kommando wird Router-ID festgelegt.
- Jedes Interface, das zur IP im network-Kommando passt, wird freigeschaltet, OSPF-Pakete zu senden und zu empfangen.
- das mit dem network-Kommando angegebene Netz ist in den OSPF-Routing-Updates enthalten.

### 7.5.5 Auswahl der Router-ID:

1. wird mit `router-id <a.b.c.d>` explizit gesetzt
2. wenn das Kommando `router-id` nicht abgesetzt wird, wählt der Router die **höchste IP** eines seiner Loopback-Interfaces
3. gibt es keine Loopback-Interfaces, wählt der Router die höchste IP seiner konfigurierten Interfaces. Diese müssen up and up sein, aber nicht am OSPF-Routing teilnehmen.

Die Router-ID wird beim Absetzen des ersten `network`-Kommandos festgelegt. Um Änderungen anschliessend wirksam werden zu lassen, muss man `clear ip ospf process` ausführen.

Gibt es doppelt vorkommende Router-IDs, funktioniert OSPF nicht!

### 7.5.6 Designated Router und BDR

Wahl des DR und BDR:

1. DR wird der Router mit der höchsten Interface-Priority
2. BDR wird der Router mit der zweithöchsten I/F-Priority
3. sind die I/F-Priorities gleich, werden die Router-IDs verglichen

Der default-Wert der Interface-Priority ist 1. Wird der Wert auf 0 gesetzt, kann der Router weder zum DR noch BDR gewählt werden.

## A Aufgaben

Abb. 4 zeigt einen Gebäudeplan mit 6 Routern.

1. Der Netzbereich 172.18.16.0/23 soll in geeignete Subnetze unterteilt werden. Gib für dieses Netz an:
  - den Bereich gültiger Hostadressen
  - Netz- und Broadcastadresse
  - die Netzmaske
2. der Bereich soll in 6 Subnetze für die Gebäude aufgeteilt werden. Dabei soll gelten:
  - pro Gebäude ein Teilnetz verwenden
  - die Teilnetze der Grösse nach ordnen und dabei mit dem kleinsten Netz beginnen
  - in jedem Teilnetz soll es ein Gateway geben, das die höchstmögliche Adresse bekommt
  - gib pro Teilnetz die Netzadresse in CIDR-Schreibweise, die Broadcastadresse und die Netzmaske an
3. Gib für die Router in Gebäude 4 und 6 die Routingtabellen an.

Für Router 4 gilt: Das Gebäude 4 soll vollen Zugriff auf die 4 Server in Gebäude 6 haben. Ausserdem sollen Rechner in Gebäude 3 direkt Daten mit den Rechnern in Gebäude 4 austauschen können.

Für Router 6 gilt: die Rechner in allen Gebäuden sollen Zugriff auf die Server haben.

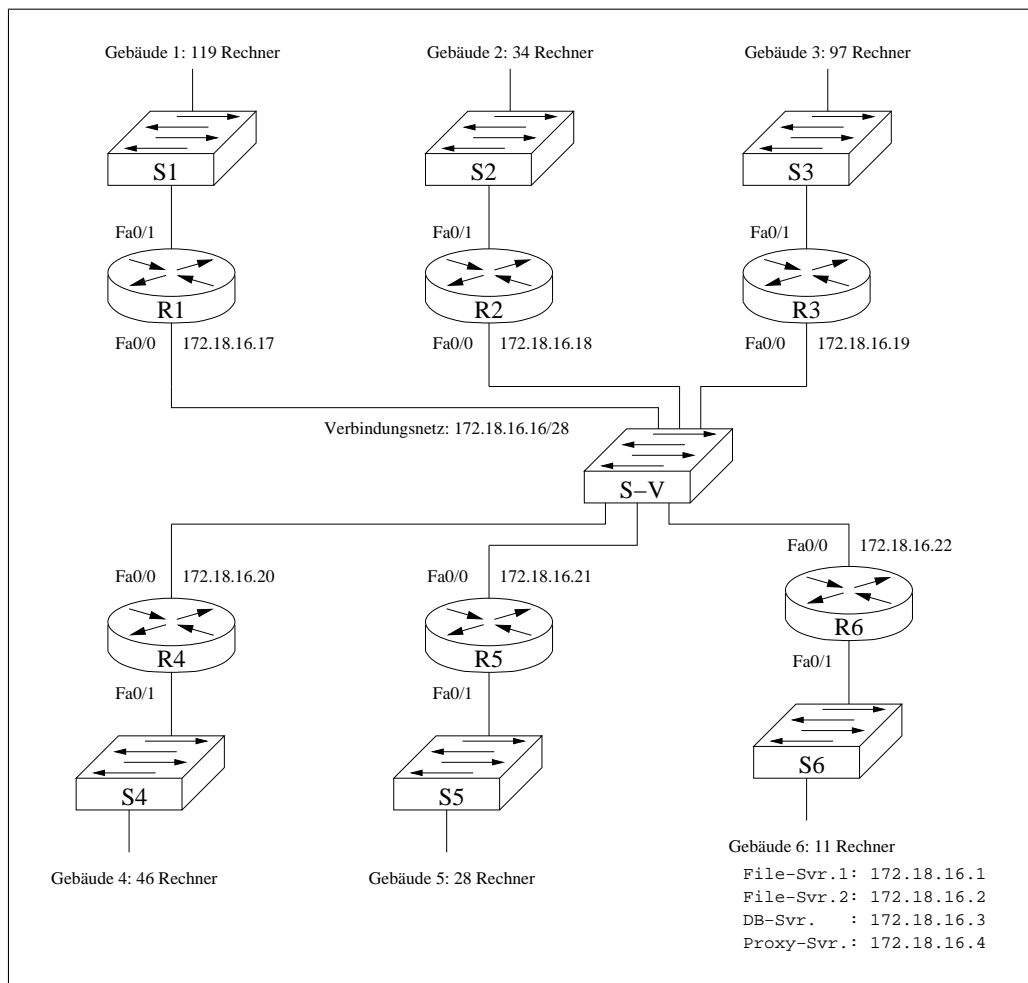


Abbildung 4: Routing zwischen 6 Gebäuden