Link-State-Routing
mit

OSPF

Open Shortest Path First

Dipl. Ing. (FH) Gregor Domhan
Web: www.domhan.de
eMail: gregor@domhan.de

Stand: 05.2002

Inhalt:

1. Link-State-Protokoll

Link-State-Protokolle zeichnen sich dadurch aus, daß eder Knoten eine Kopie der topologischen Netzwerkkarte besitzt. Die Berechnung der Routen wird nicht zentral, sondern dezentral durchgeführt.
Darum werden Link-State-Protokolle auch als Konzept der „dezetralen Karte“ bezeichnet.

1.1 Link-State-Datenbank

Die Kopie der topologischen Netzwerkkarte eines jeden Knoten wird in einer Datenbank verwaltet. In dieser Datenbank existieren für jeden genau so viele Einträge, wie unmittelba nachfolgende Knoten existieren. Zusätzlich wird auch eine Nummer der Verbindung und die Entfernung gespeichert.

Beispiel:

Ein Netzwerk mit Routern. Die Zahlen neben den Routern sind Verbindungsnummern, die in der Datenbank benötigt werden. (Abblidung 1)

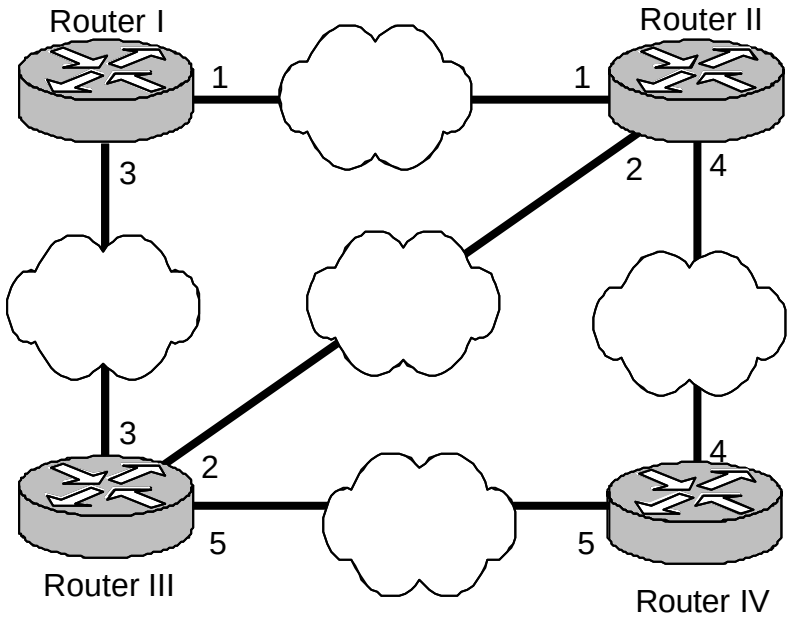


Abbildung 1:

Nachfolgend die Routing-Datenbank:

Von	Nach	Verbindung	Entfernung
I	II	1	1
I	III	3	1
II	I	1	1
II	III	2	1
II	IV	4	1
III	I	3	1
III	II	2	1
III	IV	5	1
IV	II	4	1
IV	III	5	1

Anhand der Datenbank kann nun jeder Knoten den absolut kürzesten Pfad zu jedem anderen Knoten berechnen.
Zusätzlich verhindert das dezentrale Routing eine Schleifenbildung, da jeder Knoten die gleichen Informationen über die Topologie besitzt.

1.2 Flooding Protokoll

Das Link-State-Protokoll hat die Notwendigkeit, seine netzwerkkarte auf einem aktuellen Stand zu halten.
Für diesen Zweck ist das Flooding-Protokoll zuständig.
Das Protokoll vesendet bei einer Veränderung einer Verbindung Nachrichten an die anderen Netzknoten. Diese aktualisieren dann den entsprechenden Eintrag in ihrer Routing-Datenbank.

Beispiel:

Entfällt z.B. eine Verbindung, so werden Nachrichten versandt.

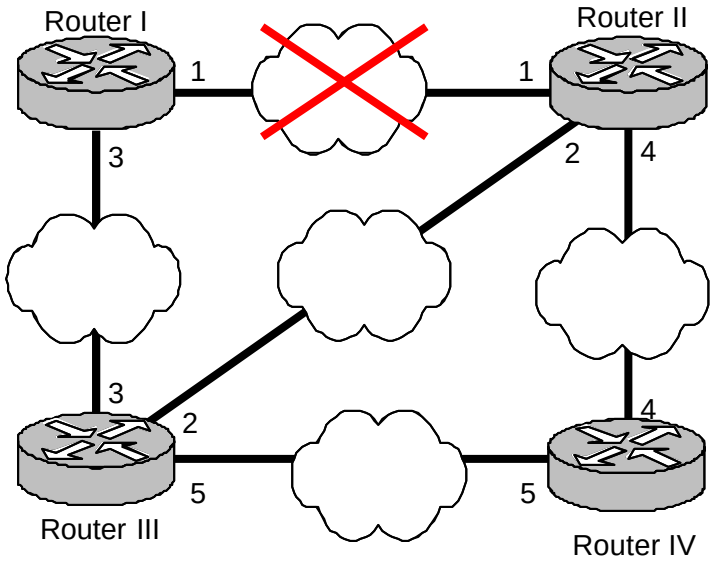


Abbildung 2:

Wenn nun Router I bemerkt, daß die Verbindung 1 zu Router II unterbrochen ist, dann sendet er folgende Nachricht an Router III:

```
Von I nach II Verbindung 1, Entfernung = unendlich (inf)
```

Router II wird unverzüglich diese Nachricht an Router II und Router IV weitersenden. Somit sind alle über die unterbrochene Verbindung 1 informiert und könne ihre Routing-Datenbank dementsprechend anpassen.

Diese Verfahren ist jedoch nicht unkritisch. Beispielsweise kann eine „alte“ Nachricht die Routing-Datenbank verfälschen. Darum geht der Flooding-Algorithmus woe folgt vor:

1. Untersuchung der Datenbankeintrages anhand der empfangenen Nachricht,

2. Falls die Nachricht noch nicht in der Datenbank vorhanden ist, dann eintragen. Dann übertrage die Nachricht an die nächsten Knoten außer an den Knoten, der die Nachricht verschickt hat.

3. Wenn die Nummer in der Datenbank kleiner als die Nummer der Nachricht ist, ersetze den Datensatz durch den neuen Wert und gib die anchricht weiter, mit Ausnahme des Knotens, von dem die Nachricht stammt.

4. Ist die Nummer in der Datenbank hher als die der Nachricht, übertrag eine neue Nachricht mit dem Inhalt der Datensatzes an den Knoten, der die Nachricht gesendet hat.

5. Sind beide Nummern gleich, tue nichts.

Nach dem Ausfall der Verbindung 1 sieht die Routing-Datenbank wie folgt aus:

Von	Nach	Verbindung	Entfernung	Nummer
I	II	inf	1	2
I	III	3	1	1
II	I	inf	1	2
II	III	2	1	1
II	IV	4	1	1
III	I	3	1	1
III	II	2	1	1
III	IV	5	1	1
IV	II	4	1	1
IV	III	5	1	1

Anhand der nun aktuellen Routing-Datenbank kann jeder Knoten den kürzesten Weg zu einem Ziel berechnen.

1. 3 Nachbarschaften

Ein weiteres Problem besteht bei den Link-State-Protokollen. Fallen z.B. zwei Verbindgungen aus, dann kann über diesen Ausfall niemand mehr eine Aktualisierung verschicken.

Beispiel:

Ausfall der Verbidung 1 und der Verbindung 3:

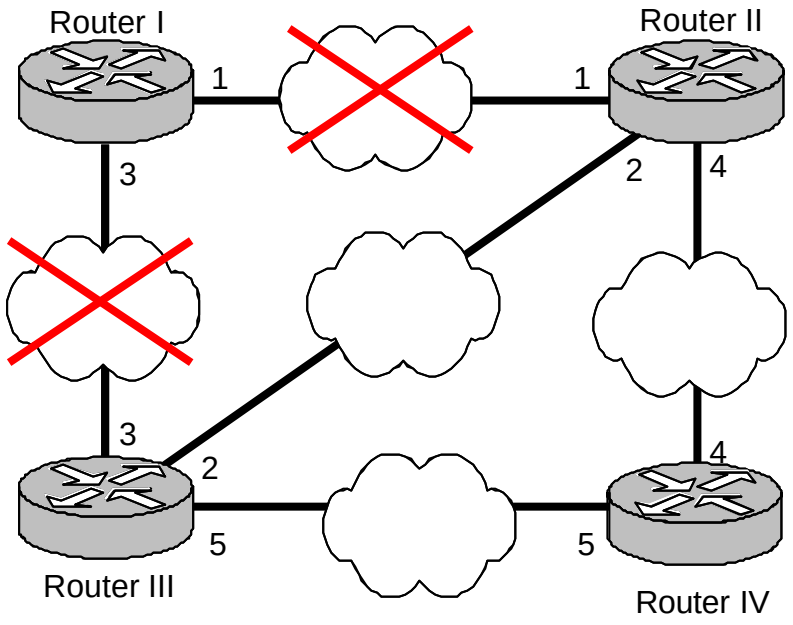


Abbildung 3:

Router I ist jetzt komplett vom Netzwerk abgetrennt. Roter I kann die anderen Knoten jedoch über diese Abtrennung nicht mehr informieren.

Diese Problem wird mit Hilfe von Nachbarschaften gelöst.

Jeder Knoten speichert einen Eintrag über seine Nachbarn. Diese Nachbarn werden in regelmäßigen Intervallen angefragt, ob diese noch leben. Erhält ein Knoten auf diese Nachfrage keine Antwort, so erklärt er die Verbidung zu diesem Nachbarn für gestört und publiziert dies an die noch lebenden Nachbarn.

1.4 Namensgebung

OSPF hat seinen Namen aus folgendem Grund erhalten.
E.W. Dijkstra hat 1959 einen Algorithmus entwickelt und vorgeschlagen, der sehr effizient den kürzesten Weg von einem Knoten zu einem anderen knoten berechnet. Diesen Algorithmus nannte er Shortest-Pat-First (SPF).

1.5 Algorithmus von Dijkstra: Shortest-Path-First

Erläuterung nach A.S. Tanenbaum:

Prizipiell werden zwei Arten von Knoten unterschieden. Ein Knoten kann permanent oder vorläufig sein. Ein permanenter Knoten wird duch einen Stern vor dem Knoten gekennzeichnet. Alle anderen Knoten sind vorläufig. Alle Knoten außer dem Quellknoten erhalten eine Beschriftung, welche die kürzeste Distanz zum Quellknoten und den Vorgängerknoten enthält.

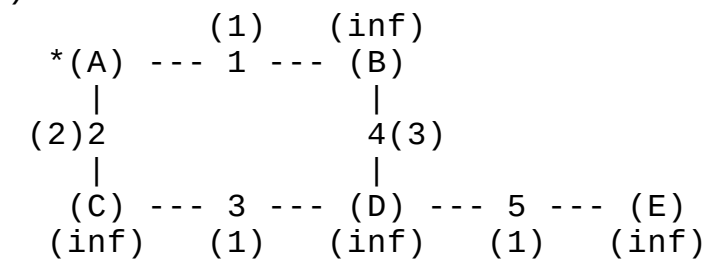
1. Alle Knoten außer dem Quellknoten werden am Anfang mit der Distanz inf (unendlich) und dem Vorängerknoten (keinen) initialisiert. Dann wird der Quellknoten A als permanent gekennzeichnet. Außerdem ist dieser Knoten der Arbeitsknote zu merken.

2. Wähle nacheinander alle nicht permanent markiertenKnoten aus, welche von dem Arbeitsknoten erreichbar sind. Falls die Distanz des ausgewählten Knotens größe ist, als die Sumer der Distanzen des Arbeitsknotens und der Distaz zum ausgewählten Knoten, so ändere die Distanz und den Vorgängerknoten entsprechend ab.

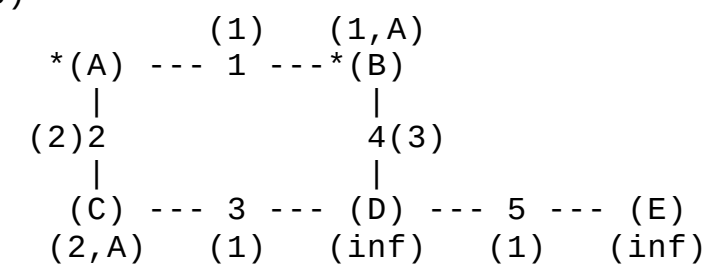
3. Betrachte nun das ganze Netzwerk und wähle dort den Knoten mit der kürzesten Entfernung zum Quellknoten aus, welcher noch nicht als permanent markiert wurde. Markiere diesen Knoten als permanet und merke in als Arbeitsknoten. Wenn der Zielknoten permanent gemacht wird, terminiert der Algorithmus. Ansonsten gehe zu Schritt 2 über.

Nachfolgend ein Beispiel:

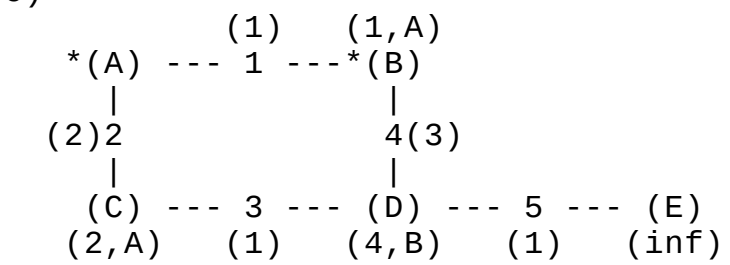
a)



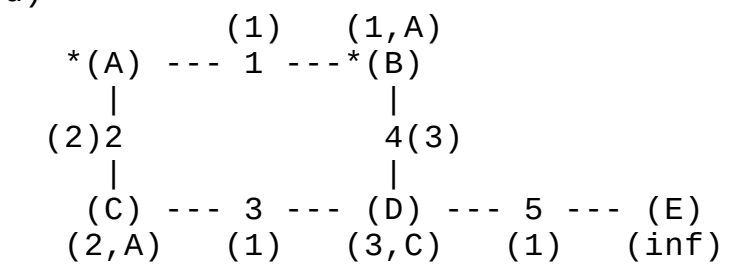
b)



c)



d)



e)

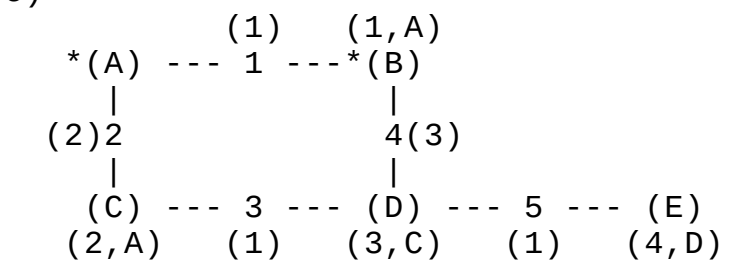


Abbildung 4:

2. Open-Shortest-Path-First (OSPF)

2.1 Router-Klassen

OSPF unterscheidet vier Router-Klassen:

I) Interne Router, die gänzlich zu einem Bereich gehören (**IA**),
II) Router an Bereichsgrenzen, die zwei oder mehr Bereiche verbinden (**ABR**),
III) Backbone-Router, die sich am Backbone befinden (**IA in LSA 0**),
IV) Grenz-Router, die zwischen mehreren autonomen Systemen vermitteln (**ASBR**),

Beispiel / Namensgebung:

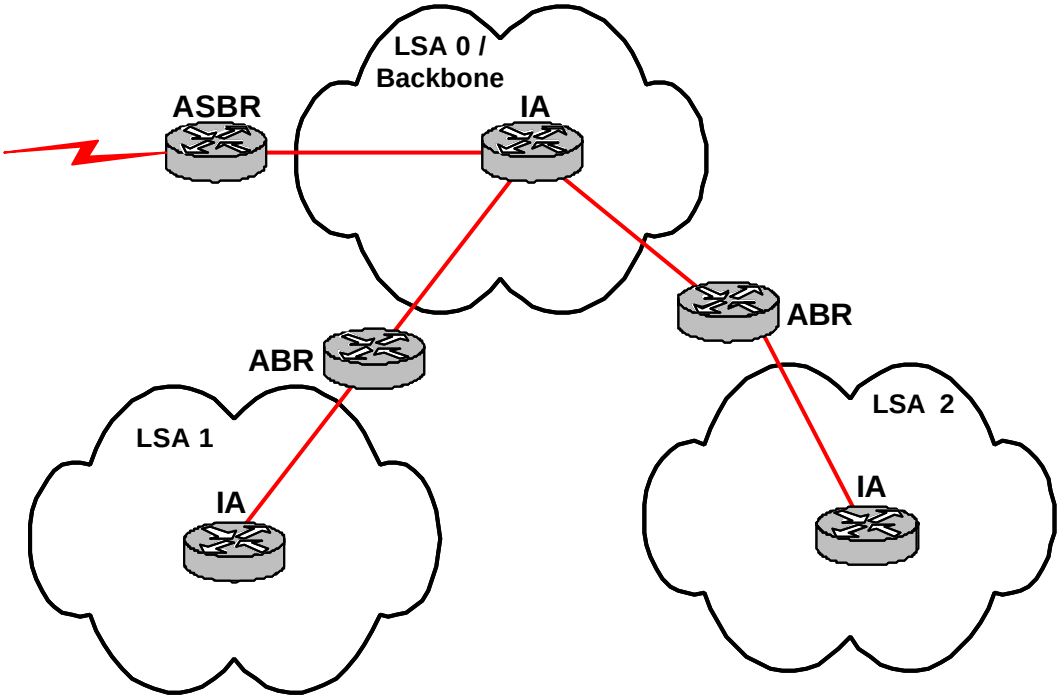


Abbildung 5:

ASBR Autonomous System Border Router,
ABR Area Border Router,
IA Intra Area Router,
LSA Link-State-Area,
LSDB Link-State Data-Base,

2.2 Regeln

Folgende Eigenschaften muß ein per OSPFgeroutetes Netzwerk besitzen:

- es muß immer eine Area 0 existieren (LSA 0),
- alle anderen Router müssen direkt mit der LSA 0 verbunden sein,

Die nachfolgenden Eigenschaften sind für ein OSPF geroutetes Netzwerk tödlich:

- direkte Verbindungen zwischen IAs,
- mehrere LSA 0,
- geteilte Areas,

2.3 OSPF Hello-Protokoll

Das OSPF-Hello-Protokoll hat die Aufgabe, Nachbarn zu finden, Keepalives zu verschicken und die Auswahl eines Designates Routers (**DR**) und eines Backup DR (**BDR**).

Ein Hello-Packet beinhaltet normalerweise folgendes:

- originating Router ID,
- Area ID,
- Subnetzmaske,
- Authentifikation,
- Timer
  - Hello Interval = 10s
  - Dead Interval = 40s
- Router Priority,
- DR, BDR,
- alle Nachbarn

Hier ein typischer Trace eines OSPF-Hello-Paketes:

```
OSPF: ----- OSPF Header ----
OSPF: Version = 2, Type = 1 (Hello),  Length=52
OSPF: Router ID              = [172.17.1.10]
OSPF: Area ID                = [0.0.0.0]
OSPF: Header checksum        = C964 (correct)
OSPF: Authentication: Type   = 0 (no Authentication),
OSPF:
OSPF: Network mask           = [255.255.255.0]
OSPF: Hello interval         = 10 (seconds)
OSPF: Optional capabilities  = 02
OSPF:              .0.. ....= Opaque-LSAs not forward
OSPF:              ..0. ....= Demand Circuit bit
OSPF:              ...0 .... = External Attributes bit
OSPF:              .... 0...= no NSSA capability
OSPF:              .... .0.. = no multicast capability
OSPF:              .... ..1.= external routing capability
OSPF:              .... ...0= no Type of Service routing
OSPF: Router priority        = 1
OSPF: Router dead interval   = 40 (seconds)
OSPF: Designates router      = [172.17.1.1]
OSPF: Backup DR              = [172.17.1.2]
OSPF: Neighbour (1)          = [192.168.1.1]
```

Bei den “neighbours” gibt es verschiedene Stati:

- `down`,
  der Anfangszustand. Es zeigt an, daß der Nachbar noch nicht gesehen wure.

- `attempt`,
  der Nachbar wurde bereits gesehen und ist „alive“.

- `init`,
  der Nachbar wurde gesehen, jedoch wurde keine bi-direktionale Kommunikation festgestellt, d.h. es gab noch keine Antwort auf das Hello.

- `2-way`,
  es gibt für diese Verbindung sowohl einen DR als auch einen BDR.

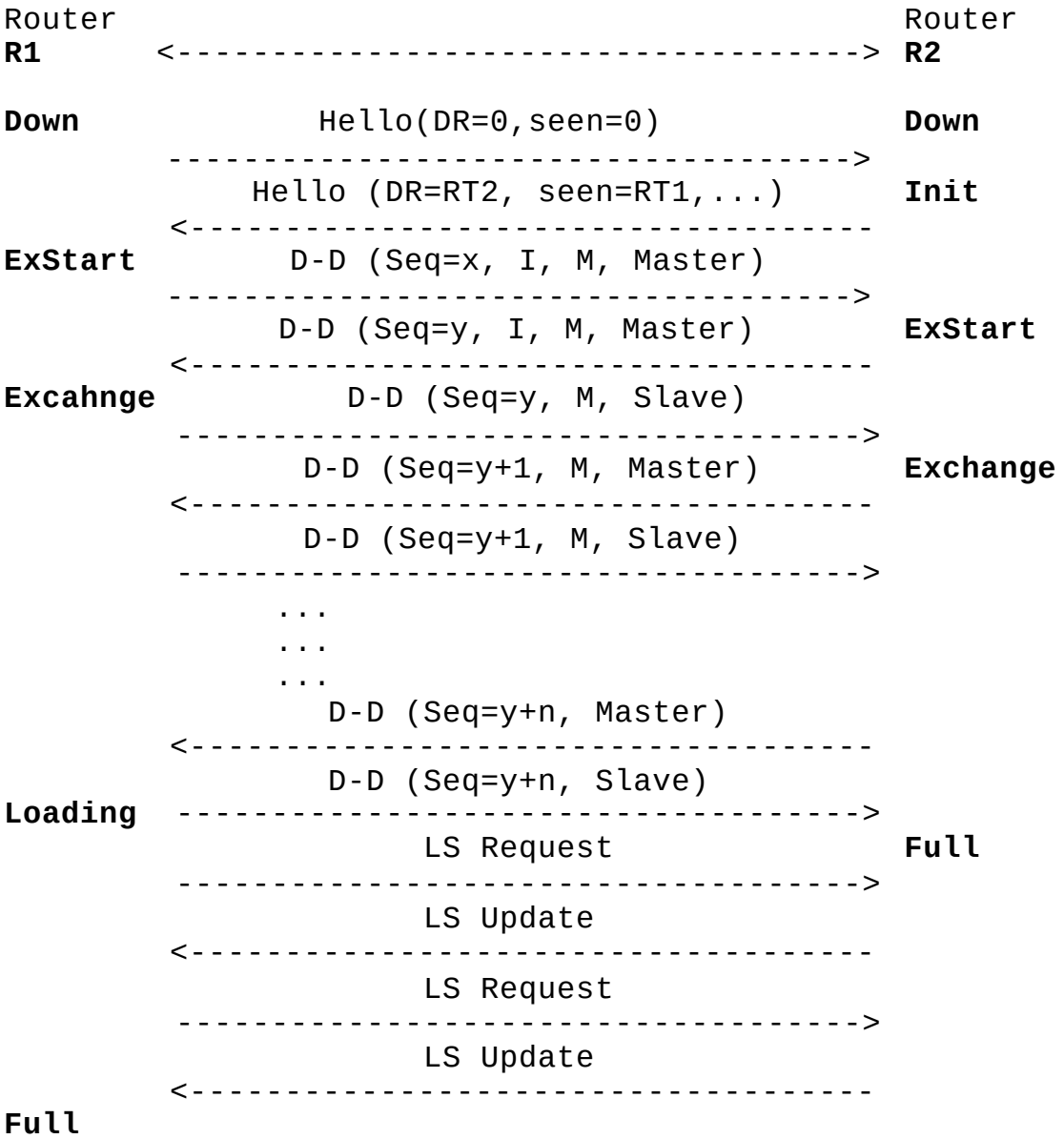
Versand der Hello-Nachrichten sieht wie folgt aus:

DR others fluten nach AI ID Router über → 224.0.0.5
DR fluten zu DR others über → 224.0.0.6

Die Neighbour-Einträge in der Datenbank sehen z.B. wie folgt aus:

```
Neighbour ID  PRI     State     Dead Time  Address       Interface
172.17.1.1    1       FULL/BDR  00:00:36   172.17.1.1    eth0
172.17.1.2    100     FULL/DR   00:00:31   172.17.1.2    eth1
192.168.1.1   1       FULL/ -   00:00:04   192.168.1.1   eth2
```

Ablauf einer hello Konversation:



Erläuterung:

Beim Start des Interfaces am Router 1 beginnt diese Hello-Pakete zu senden. Der Router R1 weiß jedoch nichts über einen Designated Router oder einen Nachbarn.

Router R2 hört diese Hello und sendet ein Paket, das ihn als Designated-Router auszeichnet. Weiterhin zeigt er an, daß er Hello-Pakete empfanen hat.

Daraufhin geht Router R1 in den ExStart Status.

Router R1 versucht nun der Master zu sein. Es sei denn, er erkennt an der höheren Router ID, daß Router R2 bereits Master ist. In diesem Fall „beantragt“ Router R1 den Status Slave.

Wird dieser Antrag befürwortet, dann aktualisieren beide Router ihre Neighbour-Database.

2.4 OSPF over IP

OSPF wird direkt über IP verschickt. Es benutzt das IP-Protokoll 89. Weiterhin sind folgende Eigenschaften im IP-Header aktiviert:

- no fragmentation required,
- TOS = internet control,

Weiterhin gibt es folgende Arten von OSPF-Paketen :

Type	Packet name	Protocol function
1	Hello	Discover/maintain neighbours
2	Database Description	Summarize database contents
3	Link State Request	Database download
4	Link State Update	Database update
5	Link State Ack	Flooding acknowledgment

2.5 Prizipieller Ablauf

Nachfolgend wir der Prizipielle Ablauf bei der Initialisierung eines Routers und die Eintragung/Löschung einer neuen Route aufgezeigt:

```
        ----Hello ----
OSPF:   Rcv hello from 172.17.1.1 area 1 from eth1
OSPF:   End of hello processing
OSPF:   Rcv hello from 172.17.1.2 area 0 from eth0
OSPF:   End of hellp processing

        ---- Abgleich der DB ----
OSPF:   Send DBD to 172.17.1.1 on eth1 seq opt 0x2 flag 0x2
OSPF:   Rcv DBD from 172.17.1.1 on eth1
OSPF:   Databse request to 172.17.1.1
OSPF:   sent LS Req packet to 172.17.1.1
OSPF:   Rcv DBD from 172.17.1.1 on eth 1
OSPF:   Exchange done with 172.17.1.1 on eth1

        ---- Finden der Area 0 ----
OSPF:   Send DBD to 172.17.1.2 on eth0 seq opt 0x2 flag 0x2
OSPF:   Rcv DBD from 172.17.1.2 on eth0
OSPF:   Build network LSA for eth0 router ID 192.168.1.1
OSPF:   running SPF for area 0
```

```
OSPF:   Initializing to run spf
        ---- Wegfall eines Routers ----
OSPF:   Send hello to 172.17.1.1 area 1 t eth1
OSPF:   delete lsa ID 172.17.1.1, type 1

         ---- Eintragung einer neuen Route ----
OSPF:   Rcv Route to 192.168.1.1 via 172.17.1.2, Mask
        255.255.255.0, Metric 1456
OSPF:   Send LSA Update

         --- Löschen einer Route ----
OSPF:   Rcv Route delete to 192.168.1.1 via 172.17.1.2, Mask
        255.255.255.0, Metric 1456
OSPF:   Send LSA Update
```