

# Distance Vector Multicast Routing Protokoll

Seminararbeit  
von

Michael Schröder

aus  
Dresden

vorgelegt am  
Lehrstuhl für Praktische Informatik IV  
Prof. Dr. Effelsberg  
Fakultät für Mathematik und Informatik  
Universität Mannheim

November 1999

Seite 1

## **1 Einleitung**

Neue Medien, wie Teleteaching Vorlesungen, Videokonferenzen, Radio - und Fernsehsendungen, die über das Internet übertragen werden, sind auf sehr hohe Bandbreiten angewiesen. Das Problem dieser Art von Benutzung ist, daß sie vom Sender vielfach zu jedem einzelnen "Client" dieselbe Information schicken müssen. Multicasting bringt hier einen Lösungsansatz, der vor allem für das Versenden sehr ressourcenschonend arbeitet.

Das traditionelle Internet ist auf Punkt-zu-Punkt oder Unicast-Verbindungen ausgelegt. Die Versendung von Informationen an eine bestimmte Gruppe soll durch das Multicast möglichst effektiv erledigt werden. Der grundlegende Unterschied zwischen Multicast und Broadcast ist, daß sich die Gruppenmitglieder im ersten Fall explizit anmelden müssen. Man will nicht durch n-fache Unicast-Verbindungen ein größeres Publikum mit derselben Information erreichen.

Für Videokonferenzen ist es aber auch nötig, daß verschiedene Gruppenmitglieder untereinander interagieren können. Dies wird durch die Sendemöglichkeit aller Gruppenmitglieder ermöglicht. Es gibt nicht, wie beim Fernsehen, nur einen Sender, der an alle Gruppenmitglieder sendet.

Das Problem der Anmeldung zu den Gruppen soll in dieser Seminararbeit nicht erörtert werden. Vielmehr sollen bestehende Routingprotokolle beschrieben werden. Eines der ersten Protokolle, das für den Multicast implementiert wurde, ist das Distance Vector Multicast Routing Protokoll (DVMRP).

<b>1</b>	<b>EINLEITUNG .....</b>	<b>2</b>
<b>2</b>	<b>TABELLENVERZEICHNIS.....</b>	<b>4</b>
<b>3</b>	<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>4</b>
<b>4</b>	<b>HISTORIE DES MULTICAST .....</b>	<b>5</b>
<b>5</b>	<b>DER ADREßBEREICH.....</b>	<b>6</b>
5.1	Routing.....	7
5.2	Anforderungen an einen Routing-Algorithmus.....	7
5.3	Kategorien von Routing-Verfahren.....	7
<b>6</b>	<b>ENTWICKLUNG DER ROUTINGALGORITHMEN FÜR DEN MBONE ...</b>	<b>8</b>
6.1	Flooding .....	8
6.2	Reverse-Path Forwarding.....	8
<b>7</b>	<b>DAS DISTANCE VECTOR MULTICAST ROUTING PROTOCOL .....</b>	<b>10</b>
7.1	Das Tunneling.....	10
7.2	Allgemeine Übersicht über DVMRP.....	10
7.3	Aufbauen eines DVMRP Netzes.....	11
<b>8</b>	<b>BEISPIEL.....</b>	<b>13</b>
<b>9</b>	<b>RESÜMEE .....</b>	<b>25</b>
<b>10</b>	<b>LITERATURVERZEICHNIS.....</b>	<b>26</b>

## 2 Tabellenverzeichnis

TABELLE 1 : WELL-KNOWN CLASS D ADDRESSES	6
TABELLE 2 : BEISPIEL - ROUTINGTABELLEN FÜR 2 ROUTER, 1 VERBINDUNG	13
TABELLE 3 : BEISPIEL - ROUTINGTABELLEN FÜR 3 ROUTER, 2 VERBINDUNGEN	14
TABELLE 4 : BEISPIEL - ROUTINGTABELLEN FÜR 4 ROUTER, 3 VERBINDUNGEN	15
TABELLE 5 : BEISPIEL - ROUTINGTABELLEN FÜR 4 ROUTER, 4 VERBINDUNGEN	16
TABELLE 6 : BEISPIEL - ROUTINGTABELLEN FÜR 5 ROUTER, 5 VERBINDUNGEN	19
TABELLE 7 : BEISPIEL - ROUTINGTABELLEN FÜR 5 ROUTER, 6 VERBINDUNGEN	22

## 3 Abbildungsverzeichnis

ABBILDUNG 1 : FUNKTIONSWEISE VON UNICAST/MULTICAST	5
ABBILDUNG 2 : FLOODING, A IST SENDER	8
ABBILDUNG 3 : RPF, A IST SENDER	9
ABBILDUNG 4 : BEISPIEL - 2 ROUTER, 1 VERBINDUNG	13
ABBILDUNG 5 : BEISPIEL - 3 ROUTER, 2 VERBINDUNGEN	14
ABBILDUNG 6 : BEISPIEL - 4 ROUTER, 3 VERBINDUNGEN	15
ABBILDUNG 7 : BEISPIEL - 4 ROUTER, 4 VERBINDUNGEN	16
ABBILDUNG 8 : ROUTER 0 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	17
ABBILDUNG 9 : ROUTER 1 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	17
ABBILDUNG 10 : ROUTER 2 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	17
ABBILDUNG 11 : ROUTER 3 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	18
ABBILDUNG 12 : BEISPIEL - 5 ROUTER, 5 VERBINDUNGEN	19
ABBILDUNG 13 : ROUTER 0 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	20
ABBILDUNG 14 : ROUTER 1 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	20
ABBILDUNG 15 : ROUTER 2 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	20
ABBILDUNG 16 : ROUTER 3 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	21
ABBILDUNG 17 : ROUTER 4 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	21
ABBILDUNG 18 : BEISPIEL - 5 ROUTER, 6 VERBINDUNGEN	22
ABBILDUNG 19 : ROUTER 0 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	23
ABBILDUNG 20 : ROUTER 1 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	23
ABBILDUNG 21 : ROUTER 2 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	23
ABBILDUNG 22 : ROUTER 3 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	23
ABBILDUNG 23 : ROUTER 4 SENDET AN ALLE ANGESCHLOSSENEN ROUTER	24

## 4 Historie des Multicast

Die Technologie des Multicastings wurde 1992 zum ersten Mal zur Übertragung der IETF Konferenz (in Bild und Ton) in größerem Maße genutzt. Der dazu genutzte Mbone (Multicast Backbone) ist ein Overlay-Netzwerk, welches auf der existierenden Internet-Infrastruktur aufsetzt. Es hat einen eigenen IP-Adressbereich zugewiesen bekommen, der zwischen 224.0.0.0 und 239.255.255.255 liegt (Mbone.com). Nicht jeder Internet-Router besitzt die Möglichkeit Multicast-Datagramme weiterzuleiten.

Im Frühstadium des Mbone wurden relativ wenige Router, die meist auf manuell konfigurierten UNIX-Rechnern liefen, über sogenannte Tunnels verbunden. Diese Technik des Multicastings, auch über konventionelle Rechnernetze zu verbreiten, wird noch in nachfolgenden Kapiteln behandelt werden.

In den Anfängen des Mbone war die Komplexität des Netzes recht gering. Es gab eine relativ kleine Zahl von Routern, die alle über das gleiche Routingprotokoll die Wegberechnung der Datagramme vornahmen. Daher konnte man den Mbone im Anfangsstadium als ein autonomes Netzwerk ansehen.

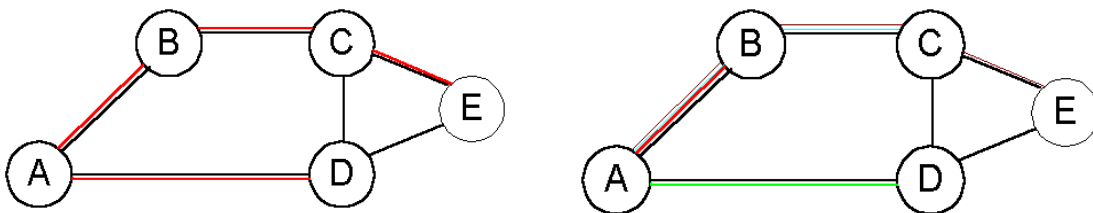


Abbildung 1 : Funktionsweise von Unicast/Multicast

Im Gegensatz zu den Unicastadressen bezeichnet eine Multicastadresse keinen einzelnen Internetknoten oder gar eine Person, sondern eine Gruppe von Rechnern. Ein Sender schickt dabei nicht an jeden Teilnehmer eine Kopie seiner Daten, sondern sendet nur einmal an die Multicastadresse. Was z.B. in einem Ethernet-LAN, bei dem alle Gruppenteilnehmer an demselben Übertragungsmedium angeschlossen sind, funktioniert, wird zum Problem, wenn Interessenten außerhalb des LANs an solch einer Session teilnehmen möchten.

Deshalb wurden Routingverfahren entwickelt, mit deren Hilfe es möglich ist, voneinander entfernte multicastfähige LANs zu koppeln. Was zunächst sehr experimentell anfang, entwickelte sich rasch zu einem weltweit umspannenden multicastfähigen Netz, dem Mbone (Multicast Backbone).

## 5 Der Adreßbereich

Im Gegensatz zu den IP Adressen der Klassen A, B und C haben die letzten 28 bits einer Klasse D Adresse keine Unterstruktur. Die Multicast Gruppenadresse ist die Kombination der oberen 4 bits 1110 und der Multicast Gruppen ID. Ihr Bereich geht von 224.0.0.0 bis 239.255.255.255.

Die Hosts, die sich zu einer bestimmten IP Multicast Adresse angemeldet haben, heißen Host Group. Eine Host Group kann aus Rechnern unterschiedlicher Netzwerke stammen. Eine Quelle kann also seine Datagramme an eine IP Multicast Adresse versenden. Die Datagramme erreichen alle Hosts eine Host Group, auch wenn der Sender die logischen IP Adressen der einzelnen Hosts nicht kennt. Das "Internet Group Management Protocol" (IGMP) kümmert sich um die Definition der Host Groups [DEE99].

Einige Multicast Gruppen Adressen sind als "well-known addresses" von der Internet Assigned Numbers Authority (IANA) festgelegt worden. Diese Multicast Gruppen Adressen werden permanente Host Groups genannt. Sie werden im Prinzip wie die well-known TCP und UDP Port Nummern verwendet.

Gruppen Adressen im Bereich 224.0.0.xxx dürfen nur mit einer TTL von 1 verschickt werden, und Gruppenadressen im Bereich 224.0.1.xxx sind für Operationen der Protokolle reserviert und werden mit normalen TTLs verschickt [CIS99].

Class D Address	Purpose
224.0.0.1	All hosts on a subnet
224.0.0.2	All routers on a subnet
224.0.0.4	All DVMRP routers
224.0.0.5	All MOSPF routers
224.0.0.9	Routing Information Protocol (RIP)- Version 2
224.0.1.1	Network Time Protocol (NTP)
224.0.1.2	SGI Dogfight
224.0.1.7	Audio news
224.0.1.11	IETF audio
224.0.1.12	IETF video
224.0.0.13	Protocol Independent Multicasting

**Tabelle 1 : Well-Known Class D Addresses**

## 5.1 Routing

Das Routing ist verantwortlich für die Leitwegbestimmung für Pakete durch das Netzwerk vom Quellsystem zum Zielsystem.

Der Leitwegbestimmungsalgorithmus eines Vermittlungsrechners entscheidet, auf welcher Ausgangsleitung ein eingegangenes Paket weitergeleitet wird. Das Multicast ist Datagramm-orientiert, weshalb eine Entscheidung für jedes einzelne Paket berechnet wird. [VORL99]

Für das Routing ist nach dem OSI Referenzmodell die dritte Schicht verantwortlich, die Vermittlungsschicht. [TAN97]

## 5.2 Anforderungen an einen Routing-Algorithmus

Routing-Algorithmen sollen folgenden klassischen Kriterien genügen:

- Korrektheit
- Einfachheit
- Robustheit gegen Rechner-oder Leitungsausfälle
- Stabilität (gleichmäßige Ergebnisse)
- Fairness
- Optimalität

Algorithmen müssen für eine lange Netz-Lebensdauer ausgerichtet sein. Sie sollen nicht unbedingt auf ein statisches Netz zugeschnitten sein, das keine Hardwareveränderungen verkraftet, wie das Ausfallen oder Hinzufügen eines Hosts, oder das Kappen oder Hinzufügen einer Verbindung.

## 5.3 Kategorien von Routing-Verfahren

Es gibt zwei grundsätzlich unterschiedliche Routing-Verfahren:

- statische, nicht adaptive Verfahren berücksichtigen nicht den Netzzustand und ändern die Leitwege während des Betriebs nicht. Die Leitwege werden statisch vor Inbetriebnahme eines Netzwerkes festgelegt.
- Adaptive Verfahren fällen Leitwegentscheidungen auf Basis verschiedener Parameter. Hier kann man noch drei Untertypen erkennen:
  - zentralisierte Verfahren
  - isolierte Verfahren
  - verteilte Verfahren

Als letztes Einordnungselement von Algorithmen soll die Differenzierung zwischen Interior- und Exterior Gateway Protokollen (IGP/EGP) angegeben werden.

Zwei Gateways, die Routinginformationen austauschen heißen externe Nachbarn, wenn sie zu zwei verschiedenen autonomen Systemen gehören. Das Protokoll eines externen Gateways wird dazu benutzt, interne Rechner dem Außensystem bekannt zu machen.

Das IGP wird dazu benutzt, die Erreichbarkeit der Rechner innerhalb eines Systems auszutauschen. Ein vorbestimmter Rechner, der alle Rechner des autonomen Netzes und deren Zustand kennt, kann über das EGP Rechner für andere autonome Netze bekannt machen [CIS99].

## 6 Entwicklung der Routingalgorithmen für den Mbone

Die Entwicklung neuer Algorithmen für Multicast-Übertragungen ist immer aktuell. Das große Wachstum des Internets im allgemeinen und des Mbone im speziellen fordert immer neue, den wachsenden Anforderungen gewappnete Algorithmen. Der einfachste Algorithmus folgt dem Prinzip des Flooding. Eine Weiterentwicklung des Flooding ist das Reverse-Path Forwarding. Auf diesem Algorithmus aufbauend wurde das wichtigste Multicast Protokoll entwickelt, das DVMRP.

### 6.1 Flooding

Das Prinzip des Flooding-Algorithmus ist, daß zunächst einmal ein Netzwerkknoten ein Datagramm gesendet bekommt, das an eine Multicast Adresse geschickt wurde. Dieser Netzknoten verifiziert, ob er schon einmal dasselbe Datagramm empfangen hat. Wenn dies der Fall ist, so wird es verworfen. Wenn nicht, so wird dieses Datagramm an alle Verbindungen weitergeschickt, außer an die, von der das Datagramm gekommen ist. Damit wird sichergestellt, daß ein Datagramm auf allen möglichen Wegen alle Netzknoten erreicht. Einer dieser Wege ist mit Sicherheit dann auch der schnellste. Die Router kommen mit diesem Algorithmus ohne Routingtabellen aus. Dies ist ein sehr einfacher Algorithmus, der aber nur in kleineren Netzen sinnvoll ist. Diese Netze dürfen aber auch keine Kapazitätsprobleme haben.

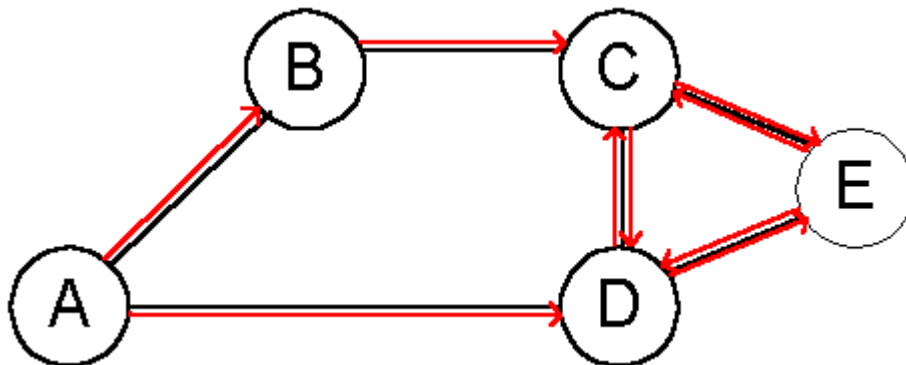


Abbildung 2 : Flooding, A ist Sender

### 6.2 Reverse-Path Forwarding

Effizienter als der Flooding-Algorithmus ist der Reverse Path Forwarding-Algorithmus (RPF). Er nutzt die Tatsache aus, daß jeder Knoten seinen kürzesten Pfad zum Sender aus der Routing Tabelle kennt! Man bezeichnet diese Pfade als Reverse Paths. Ein Router, der den RPF in der einfachsten Form implementiert, muß beim Ankommen eines Paketes nur drei Schritte beachten [TAN97]:



1. Wenn ein Multicast Datagramm ankommt, so notiere Herkunftsknoten und die Ankunftsverbindung, über die das Datagramm angekommen ist.
2. Wenn die Ankunftsverbindung auf dem kürzesten Weg zum Senderknoten liegt, so schicke das Datagramm an alle weiteren Verbindungen.
3. Wenn die Bedingung im 2. Schritt nicht wahr ist, so verwirfe das Datagramm.

Tanenbaum rät jedoch, um eine optimale Route zu erhalten, die Unicast-Routingtabellen nicht zu benutzen, sondern vielmehr für jeden Knoten den schnellsten Weg vom Sendeknoten zu sich selbst zu berechnen, jedoch nicht den Weg in der umgekehrten Richtung zu berechnen, also den Weg von sich zum Sendeknoten.

Das Problem des RPF ist jedoch, daß es nicht die Gruppenzugehörigkeit von Knoten bzw. Routern beachtet und somit gesendete Pakete entweder über das ganze Netz oder bis zum Ablauf des TTL versendet.

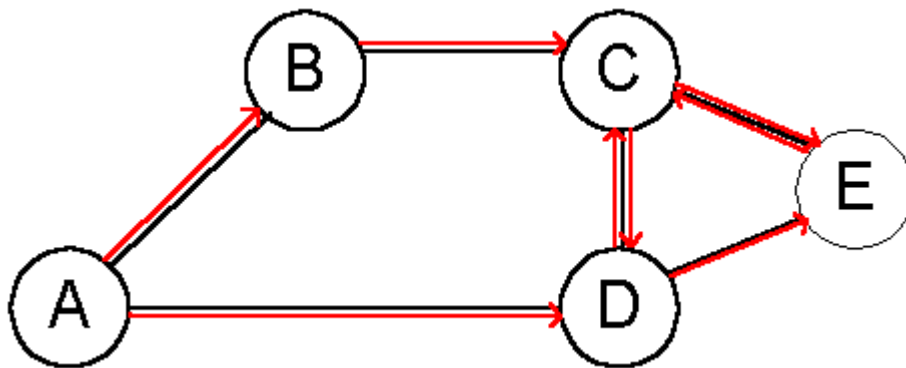


Abbildung 3 : RPF, A ist Sender

## **7 Das Distance Vector Multicast Routing Protocol**

Aus den in den in den vorhergehenden Abschnitten besprochenen Rahmenbedingungen des Mbone hat sich das wichtigste Multicast Protokoll entwickelt, das DVMRP.

Da man den Mbone anfangs als ein selbständiges System ansehen konnte, so war es nur nötig, ein IGP zu schreiben. Multicastanwendungen, die zwischen zwei autonomen Systemen arbeiten wollen, müssen auf ein übergeordnetes Protokoll über einen Gatewayrechner zurückgreifen.

Da das Mbone von Anfang an schon ein weltumspannendes Netz war, auch wenn nur mit relativ wenigen angehängten Knoten, so kam eine zentralisierte Implementation des Routing-Algorithmus nicht in Frage. Man entschied sich für eine verteilte Implementierung. In diesem Falle heißt dies, daß jeder Router in seiner Routingtabelle Informationen über jeden anderen Router beherbergt. Er errechnet sich selbst seine Position im Netz.

Das DVMRP wurde ursprünglich nur zu Forschungszwecken geschrieben. Als Basis hat man das Routing Information Protocol (RIP, RFC 1058) genommen. Dieses Protokoll ist unter anderem in dem Programm „mrouted“ implementiert, welches den Standard für den Austausch von Routing-Informationen zwischen Gateways und Hosts darstellt.

Ein Problem des DVMRP ist, daß es nicht von den Standard-Routern implementiert ist. Für einen weltweiten Betrieb ist es jedoch nötig das schon vorhandene Netz mit seinen Routern zu benutzen. Um dies zu ermöglichen, behilft man sich der Technik des Tunneling.

### **7.1 Das Tunneling**

Das Problem des DVMRP ist es, eine Multicast-Datagramme von einem Multicast-Router zu einem anderen Multicast-Router über ein inkompatibles Netz zu schicken. Beim Tunneling wird manuell eine virtuelle Multicast-Verbindung definiert. Ein Tunnel wird manuell von einem Netzwerkadministrator definiert. Er gibt die Start- und Ziel - IP-Adresse und eine Metrik an. Der Administrator des Zielrouters muß die umgekehrte Richtung des Tunnels definieren. Sollen Multicast-Datagramme an eine Adresse jenseits des Tunnels geschickt werden, so werden sie am Eingang des Tunnels in ein Unicast-Datagramm eingepackt, und als Unicast-Datagramm bis zum Ausgang im "normalen" Internet geroutet. Am Ausgang werden sie dann vom Zielrouter wieder ausgepackt und als normale Multicast-Datagramme weiterbehandelt.

### **7.2 Allgemeine Übersicht über DVMRP**

[PU99] gibt als mögliche reduzierte Sichtweise für das DVMRP die Abkürzung „broadcast & prune“ Protokoll. Es werden für jede einzelne Quelle Broadcast Bäume durch Routing-Informationsaustausch aufgebaut.

Für jede Host Group werden dann sogenannte delivery-trees durch das Abschneiden (pruning) des ursprünglichen Broadcast Baumes gebildet. Über den RPF-Algorithmus wird entschieden, ob Multicast Datagramme den Baum herunter weitergeleitet werden sollen oder nicht.

So kann von jeder Quelle ein separater Shortest-Path Baum gebildet werden.

### 7.3 *Aufbauen eines DVMRP Netzes*

#### 1. **Ausfindigmachen der Nachbarn**

Wenn ein neuer Router im Netz angeschlossen wird, oder wenn er nach einer Down-phase angeschaltet wird oder eine neue Verbindung im Netz aktiviert wird, so versucht er seine Nachbarn zu finden. Er sendet über alle von ihm ausgehenden Verbindungen (Interfaces) an die All-DVMRP-Routers IP Multicast Gruppen Adresse (224.0.0.4) jeweils eine Neighbour Probe Message mit einem TTL von 1. Damit soll sichergestellt werden, daß nur die Nachbarn erreicht werden. Diese antworten auf die Nachricht und der Sender kann die ermittelten Nachbarn in seine Routing Tabelle aufnehmen.

Sicherheitshalber werden Nachbarn in periodischen Zeitabständen gesucht, um auf alle Fälle einen aktuellen Stand des den Router umgebenden Netzes zu haben. Eine adäquate Frequenz ist 35 Sekunden. Die Variable TTL wird jedoch nur für die Probe Messages verwendet. Für die spätere Wegbestimmung der Pakete werden Metriken verwendet, die für jedes Interface und für jeden Tunnel bestimmt werden.

#### 2. **Integrieren der Routingtabellen der Nachbarn**

Nach dem Kennenlernen der Nachbarn fordert ein Router die Routingtabellen der Nachbarn an. Die in einem Baum höhergelegene Router versuchen herauszufinden, ob niedriger gelegene Router zur Datagrammweiterleitung von ihnen abhängig sind oder nicht. Wenn ein niedriger gelegener Router einen höher gelegenen Router als besten Sprung zu einem bestimmten Quell-Netzwerk ansieht, so wird dem höher gelegenen Router die originale Metrik plus unendlich angegeben. Wenn der höhergelegene Router eine Metrik zwischen unendlich und zweimal unendlich sieht, so kann er den niedrigen Router als abhängigen Router für diese Quelle in seine Tabelle eintragen [DEN98].

Eine Routingtabelle beinhaltet im wesentlichen die folgenden Spalten:

- Adresse des im Baum höhergelegenen Routers und seine Metrik
- Liste der im Baum niedrigergelegenen Router

#### 3. **Der normale Betrieb:**

Wenn ein Datagramm bei einem Router ankommt, so vergleicht dieser die angegebene Quelle und die Eingangsverbindung mit den beiden korrespondierenden Parametern in der Routingtabelle. Es wird geschaut, ob der Router auf dem kürzesten Weg zwischen Quelle und Ziel(en) ist. Stimmen die beiden Parameter des ankommenden Datagrammes mit denen der in der Routingtabelle eingetragenen überein, so wird das Datagramm an die in der Routingtabelle angegebene(n) Adresse(n) weitergeschickt.

#### 4. **Pruning**

Es kann sein, daß sich unterhalb eines Routers keine Rechner mehr befinden, die zu

einer Host-Gruppe gehören. In diesem Falle kann der Router seinen übergeordneten Routern mit einer Prune-Message mitteilen, daß er für eine bestimmte Gruppe keine Datagramme mehr erhalten will. Er meldet sich sozusagen aus dem Baum ab. Um einen Router nicht für immer abzuschneiden, gibt er eine Periode mit an, in der das Prune wirksam sein soll. Danach wird er automatisch wieder in den Baum miteinbezogen.

#### 5. **Grafting**

Wenn ein Teilbaum, bzw. ein Router sich durch eine Pruning-Message abgemeldet hat, so kann er sich wieder in dem Baum mit einer Grafting-Message anmelden. Dies kann der Fall sein, wenn sich jemand in eine Host Group dynamisch anmeldet. Für jedes Quell-Netzwerk, aus dem man sich abgemeldet hat, muß eine einzelne Grafting-Message gesendet werden. Die Grafting-Messages werden nur an die übergeordneten Router gesendet. Diese Message wird mit Graft-Ack bestätigt.

## 8 Beispiel

Im folgenden Beispiel soll die Entwicklung der Routingtabellen einzelner Router dargestellt werden. Nach und nach werden Verbindungen zwischen den einzelnen Routern aktiviert.

1. Schritt: Verbindung von Router 0 mit Router 1 über Verbindung V0

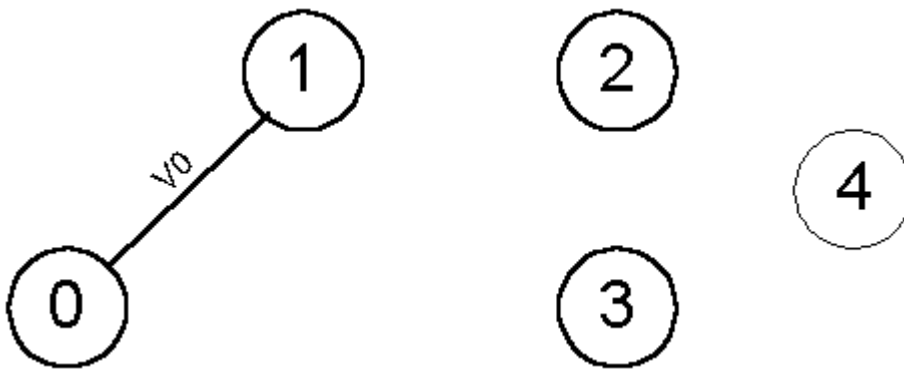


Abbildung 4 : Beispiel - 2 Router, 1 Verbindung

Router	Übergeordnetes Interface	Metrik	Untergeordnetes Interface
Router 0			
1	0	32	
Router 1			
0	0	32	

Tabelle 2 : Beispiel - Routingtabellen für 2 Router, 1 Verbindung

Um die Routingtabellen zu aktualisieren, wird ein Datenaustausch mit anderen Routern vorgenommen. Dafür schicken die beiden Router zuerst „Neighbour probes“, um direkte Nachbarn zu finden. Mit der Antwort des Nachbarn wird seine Adresse in die Routingtabelle des Senders aufgenommen.

Danach werden Routing-Requests verschickt, um von den Nachbarn die Routingtabellen zu erhalten. Da die Router nur sich selbst darin finden können, werden die eigenen Einträge nicht verändert.

Am Ende dieses Schrittes weiß Router 0, daß er Router 1 über das Interface 0, bzw. die Verbindung 0 erreicht. Router 1 weiß, daß er Router 0 ebenfalls über das Interface 0, bzw. die Verbindung 0 erreicht. Router 0 weiß auch, daß er Nachrichten von Router 1 nur dann akzeptiert, wenn sie über Verbindung 0 eintreffen. Router 1 akzeptiert Nachrichten von Router 0 nur dann, wenn sie über Interface 0 eintreffen.

2. Schritt: Verbindung von Router 1 mit Router 2 über Verbindung V1

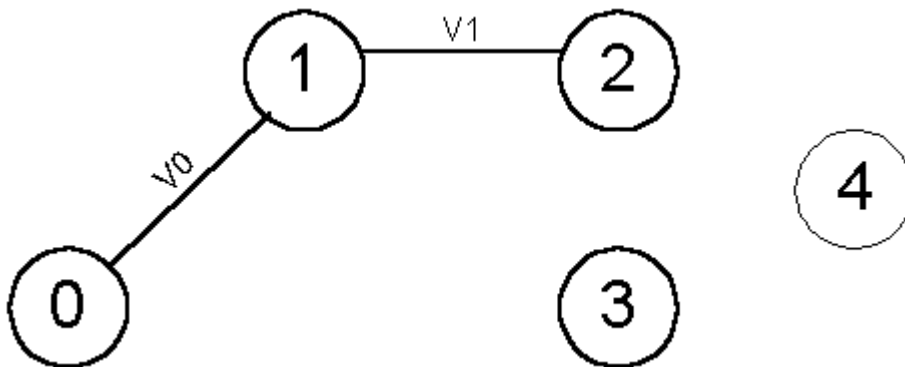


Abbildung 5 : Beispiel - 3 Router, 2 Verbindungen

Router	Übergeordnetes Interface	Metrik	Untergeordnetes Interface
Router 0			
1	0	32	
2	0	64	
Router 1			
0	0	32	V1
2	1	32	V0
Router 2			
1	1	32	
0	1	64	

Tabelle 3 : Beispiel - Routingtabellen für 3 Router, 2 Verbindungen

Router 1 und 2 verschicken "Neighbour Probes", um ihre Nachbarn zu finden. Router 2 nimmt Router 1 in seine Routingtabelle als Nachbarn auf. Router 1 nimmt Router 2 in seine Routingtabelle als Nachbarn auf. Danach verschicken sie ihr Routing-table update. Router 0 empfängt ein Routing-update und nimmt Router 2 auf.

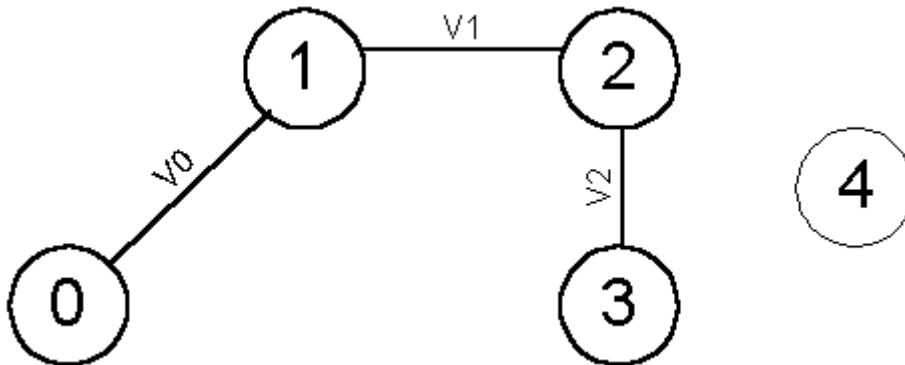
Router 1 empfängt ein "Routing Update" Paket mit einer Route zu Router 0 über Verbindung 1

→ er erkennt, daß das Interface V1 in der Baumhierarchie unter ihm liegt, wenn er Datagramme zu Router 2 erhält.

Er liegt auf dem kürzesten Weg zu Router 0 für die Datagramme, die über V1 weggeschickt werden.

3. Schritt: Verbindung von Router 2 mit Router 3 über Verbindung V2

**Abbildung 6 : Beispiel - 4 Router, 3 Verbindungen**



Router	Übergeordnetes Interface	Metrik	Untergeordnetes Interface
Router 0			
1	0	32	
2	0	64	
3	0	96	
Router 1			
0	0	32	V1
2	1	32	V0
3	1	64	V0
Router 2			
1	1	32	V2
0	1	64	V2
3	2	32	V1
Router 3			
2	1	32	
1	2	64	
0	2	96	

**Tabelle 4 : Beispiel - Routingtabellen für 4 Router, 3 Verbindungen**

In diesem Schaubild muß man sich vor Augen halten, daß die Aktualisierung der Routingtabellen sich nicht in einem Schritt vollzieht. Die oben dargestellten Routingtabellen sind das Resultat mehrerer Kommunikationen unter den Routern.

Ein Router erfährt immer nur von seinem Nachbar, was sich verändert hat. Router 2 gibt die Veränderungen an seine Nachbarn (Router 3, Router1) bekannt. Danach gibt Router 1 die Veränderungen an seine Nachbarn (Router 0, Router 2). Router 0 gibt die Veränderungen schließlich wieder an seine Nachbarn (Router 1) weiter. Die Verbreitung bzw. Aktualisierung der Routingtabellen ist erst nach mehreren Schritten vollendet.

4. Schritt: Verbindung von Router 0 mit Router 3 über Verbindung V3

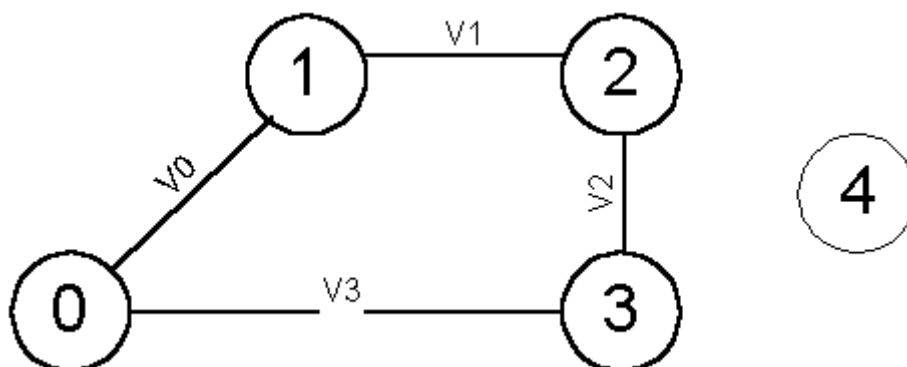


Abbildung 7 : Beispiel - 4 Router, 4 Verbindungen

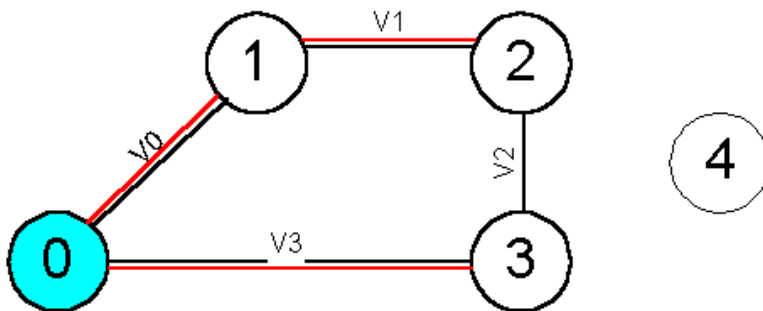
Router	Übergeordnetes Interface	Metrik	Untergeordnetes Interface
<b>Router 0</b>			
1	0	32	V3
2	0	64	
3	0	<b>32</b>	V0
<b>Router 1</b>			
0	0	32	V1
2	1	32	V0
3	1	64	
<b>Router 2</b>			
1	1	32	
0	1	64	
3	2	32	
<b>Router 3</b>			
2	1	32	
1	2	64	
0	2	32	

Tabelle 5 : Beispiel - Routingtabellen für 4 Router, 4 Verbindungen

Mit dem Hinzufügen der Verbindung V3 wird ein zyklischer Aufbau des Netzes hergestellt. Die Routingtabellen werden jedoch mit den schon angewandten Poison-Reverse-Verfahren so aufgebaut, daß Datagramme jeden Knoten nur einmal erreichen. Anhand der obigen Routingtabellen sieht man auch, daß nur Router 0 und 1 ankommende Pakete weiterleiten. Router 2 und 3 leiten keine ankommenden Pakete weiter. Somit kann man auf den ersten Blick sehen, daß wenigstens Verbindung 2 keine zyklische Weiterleitung der Datagramme (von beiden Endpunkten aus, d.h. von Router 2 und 3) erlaubt.



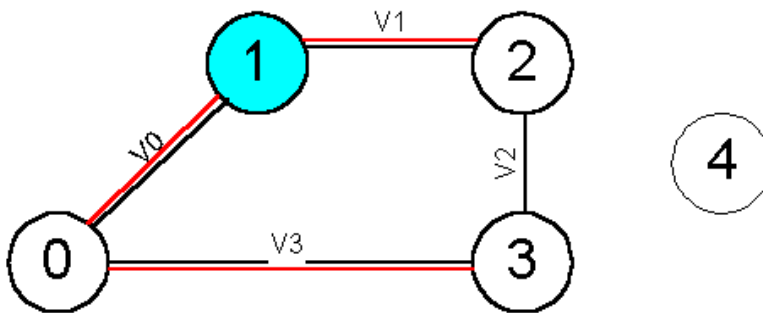
Aus den Routing-Tabellen kann man jetzt die Sendewege herleiten.



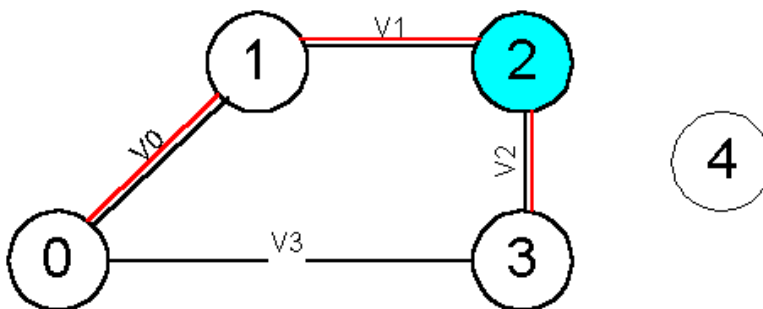
**Abbildung 8 : Router 0 sendet an alle angeschlossenen Router**

In obiger Abbildung kann man sehen, wie Die Nachrichten von Router 0 versendet werden. Man kann eine Asymmetrie beobachten. Router 2 ist sowohl über Router 1 als auch über Router 3 gleichweit entfernt. Jedoch hat es sich im Aufbau der Routing-Tabellen so ergeben, daß Router 0 den Weg über Router 1 wählt. Dies liegt daran, daß Router 0 Router 1 und 2 vor Router 3 kannte. Der einmal gewählte Weg zu Router 2 wird nicht mehr geändert, es sei denn, es fände sich ein kürzerer. In der Routingtabelle von Router 1 sieht man auch, daß von Router 0 kommende Datagramme an Verbindung 1 weitergeleitet werden.

Die folgenden drei Abbildungen zeigen die Broadcast Bäume mit den verschiedenen Routern als Sender:



**Abbildung 9 : Router 1 sendet an alle angeschlossenen Router**



**Abbildung 10 : Router 2 sendet an alle angeschlossenen Router**

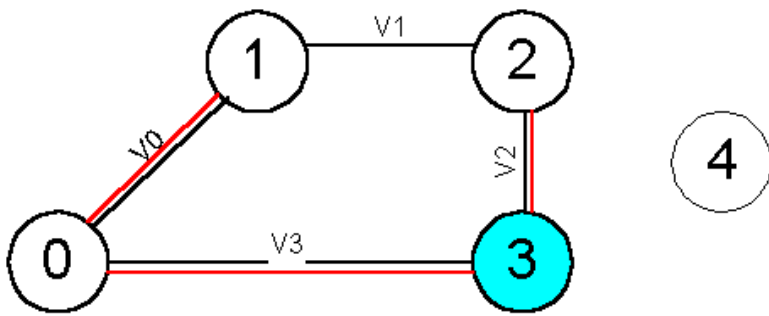


Abbildung 11 : Router 3 sendet an alle angeschlossenen Router

4. Schritt: Verbindung von Router 2 mit Router 4 über Verbindung V4

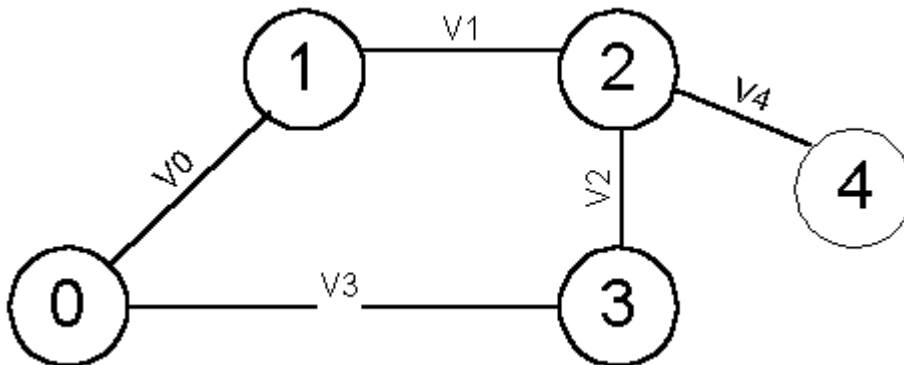


Abbildung 12 : Beispiel - 5 Router, 5 Verbindungen

Router	Übergeordnetes Interface	Metrik	Untergeordnetes Interface
<b>Router 0</b>			
1	0	32	V3
2	0	64	
3	0	32	V0
4	0	96	
<b>Router 1</b>			
0	0	32	V1
2	1	32	V0
3	1	64	
4	1	64	V0
<b>Router 2</b>			
1	1	32	V4
0	1	64	V4
3	2	32	V4
4	4	32	V2,V1
<b>Router 3</b>			
2	1	32	
1	2	64	
0	2	32	
4	2	64	V3
<b>Router 4</b>			
2	4	32	
1	4	64	
0	4	96	
3	4	64	

Tabelle 6 : Beispiel - Routingtabellen für 5 Router, 5 Verbindungen

Nach dem Hinzufügen der Verbindung V4 sieht man eine neue Besonderheit. Bis jetzt gab es in den Broadcast Bäumen nur eine Verzweigung in der Wurzel, d.h. der Sender hat über zwei Verbindungen seine Datagramme dupliziert und versendet. Mit der neuen Verbindung sieht man, daß die Routingtabelle des Routers 2 für von Router 4 kommende Datagramme über V1 und V2 weitersendet. hier findet also eine Duplikation eines Datagrammes in Router 2 statt.

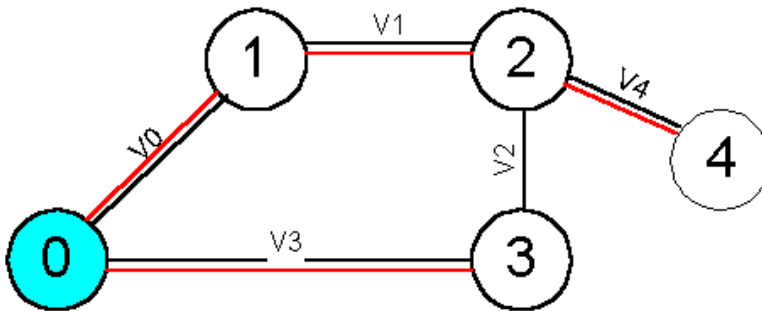


Abbildung 13 : Router 0 sendet an alle angeschlossenen Router

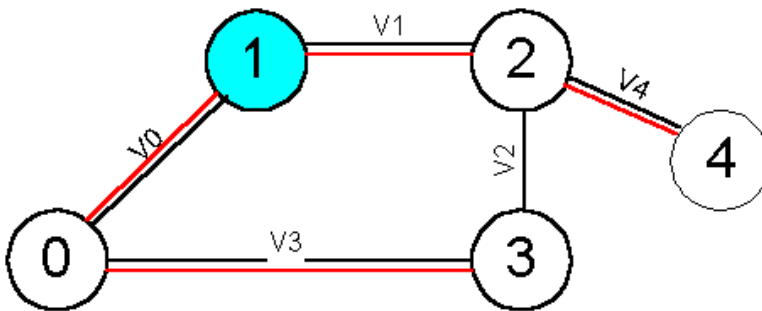


Abbildung 14 : Router 1 sendet an alle angeschlossenen Router

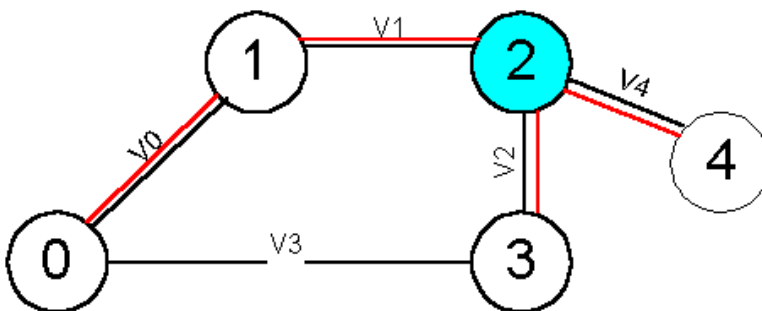
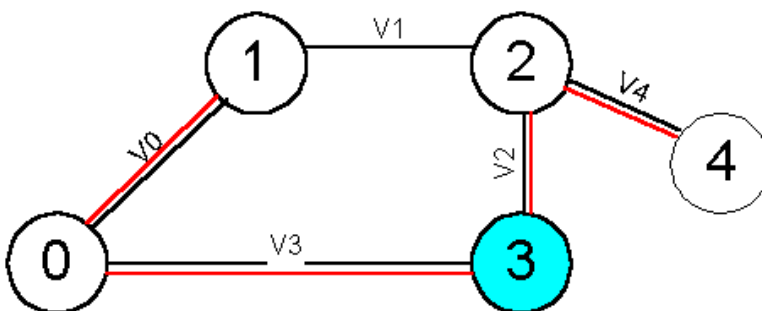
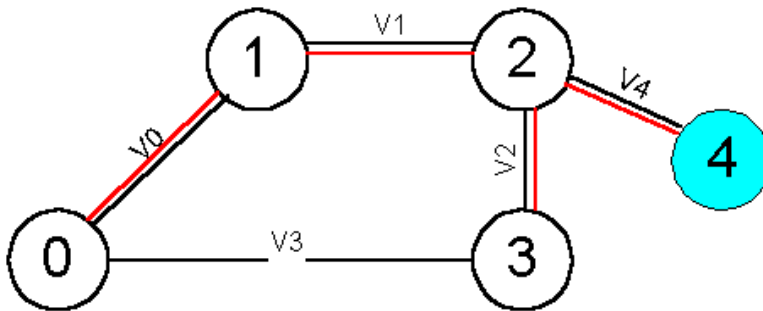


Abbildung 15 : Router 2 sendet an alle angeschlossenen Router



**Abbildung 16 : Router 3 sendet an alle angeschlossenen Router**



**Abbildung 17 : Router 4 sendet an alle angeschlossenen Router**

5. Schritt: Verbindung von Router 3 mit Router 4 über Verbindung V5

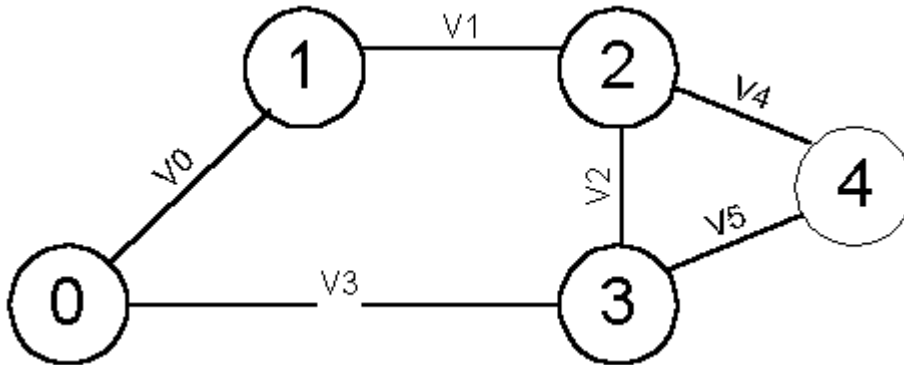


Abbildung 18 : Beispiel - 5 Router, 6 Verbindungen

Router	Übergeordnetes Interface	Metrik	Untergeordnetes Interface
Router 0			
1	0	32	V3
2	0	64	
3	3	32	V0
4	3	64	
Router 1			
0	0	32	V1
2	1	32	V0
3	0	64	
4	1	64	V0
Router 2			
1	1	32	V4
0	1	64	V4
3	2	32	V4
4	4	32	V2,V1
Router 3			
2	1	32	
1	3	64	
0	3	32	V5
4	2	32	V3
Router 4			
2	4	32	
1	4	64	
0	5	64	
3	5	32	

Tabelle 7 : Beispiel - Routingtabellen für 5 Router, 6 Verbindungen

Mit dem Hinzufügen einer fünften Verbindung soll die Versuchsanordnung etwas komplexer gestaltet werden, um das Pruning im folgenden Abschnitt hieran besser erläutern zu können.

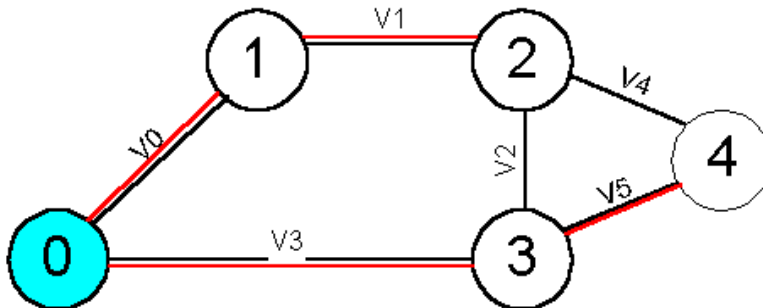


Abbildung 19 : Router 0 sendet an alle angeschlossenen Router

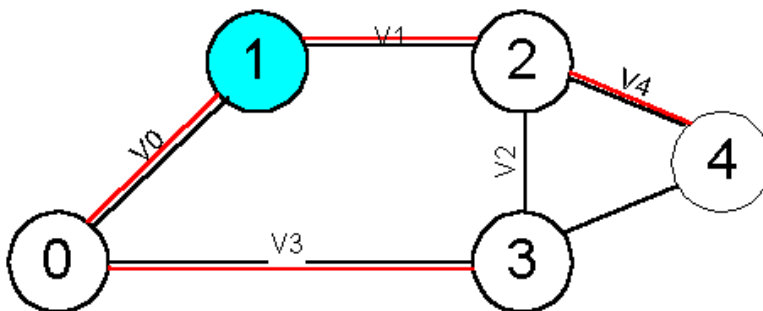


Abbildung 20 : Router 1 sendet an alle angeschlossenen Router

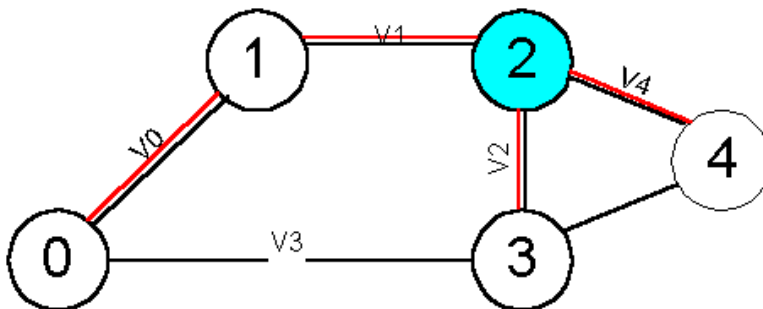


Abbildung 21 : Router 2 sendet an alle angeschlossenen Router

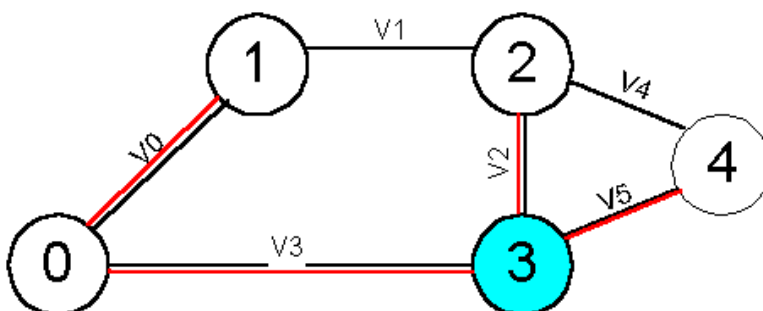
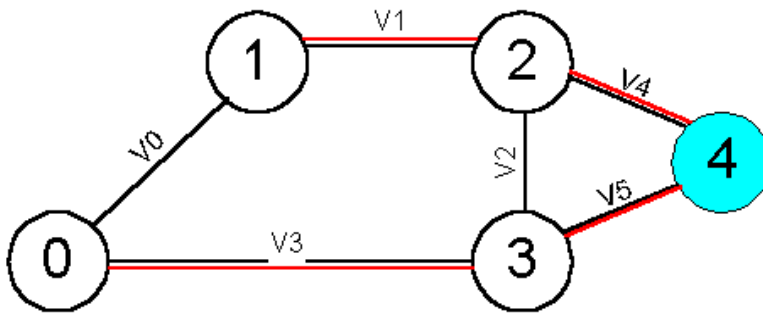


Abbildung 22 : Router 3 sendet an alle angeschlossenen Router



**Abbildung 23 : Router 4 sendet an alle angeschlossenen Router**

### **Anwendungsbeispiel des Pruningmechanismus:**

Zur Erinnerung: Beim Verschicken von pruning-Messages werden in den Routing-Tabellen selbst keine Veränderungen gemacht.

Pruning-messages werden von einem Router geschickt.

Angenommen Router 1 sendet (siehe Abb. 20). Router 2 ist nicht an diesen Nachrichten interessiert. Er kann jedoch keine Pruning-messages geben, da er kein Blatt in diesem von Router 1 aufgespanntem Baum ist. Die Nachricht muß auch Router 4 erreichen. Er erkennt dies in der Routing-Tabelle daran, daß er eine untergeordnete Verbindung V4 für von Router 1 kommende Datagramme hat.

Angenommen Router 3 sendet zusätzlich, so kann Router 2 jetzt nur Pruning-messages an Router 3 senden, da Router 2 für den Sender Router 3 ein Blatt ist.

Eine Pruning-messages wird nur an den übergeordneten Zweig gesendet.



## 9 Resümee

Bei der Implementation des DVMRP muß jeder Router das gesamte Netz kennen. Jeder einzelne Router muß autonom entscheiden können, wohin er ein Datagramm leiten muß, wenn sie von einer bestimmten Verbindung her kommen. Um dies zu gewährleisten werden die Routingtabellen in gewissen Zeitabständen vollkommen aktualisiert. Dies hat ein regelmäßiges Flooding des Netzes zur Folge.

Für überschaubare Netzgrößen ist der Aufwand vertretbar. Beim Übergang von der Nutzung eines Protokolles in einer experimentellen Umgebung zur freien Verfügbarkeit, wie es beim DVMRP der Fall ist, wird das regelmäßige Flooding zu aufwendig. In neueren Multicast-Protokollen wird dieses Manko verbessert.

Da das DVMRP ursprünglich für das Experimentieren mit dem Multicast für ein überschaubares, homogenes Netz geschrieben wurde, hat man nur es nur als IGP konzipiert. Mit dem Anwachsen der Knotenzahl ist man nun gezwungen Teilnetze über Gateways anzusprechen. Informationen müssen zwischen IGP und EGP ausgetauscht werden und man ist auf ein einheitliches Teilnetz angewiesen.

Von künftigen Protokollen wird eine bessere Skalierbarkeit, mehr Flexibilität bzw. Kompatibilität gefordert. Seitdem das Multicast immer intensiver genutzt wird, wird der Aspekt der Congestion-Control auch immer wichtiger.

Das DVMRP ist also nicht mehr den Anforderungen des Multicast-Betriebes gewachsen.

## 10 Literaturverzeichnis

- [VORL99] Vorlesung Rechnernetze Prof. Dr. Effelsberg, WS 98/99
- [TAN97] „Computernetzwerke“, Andrew S.Tanenbaum, 1997
- [DEN98] „Simulation von Multicast Routing Algorithmen in Java“, Robert R. Denda, 1998
- [DEE99] „Multicast Routing in a Datagram Internetwork“, Setphen Edward Deering, 1999
- [HUI95] „Routing in the Internet“, Christian Huitema, 1995
- [CIS99] „IP Multicast“, Cisco Systems Inc., 1999
- [RF1075] „Distance Vector Multicast Routing Protocol“, Waitzman, Partridge & Deering, 1988
- [PU99] „Distance Vector Multicast Routing Protocol“,Internet Draft DVMRP – v03-08, T.Pusateri, 1999