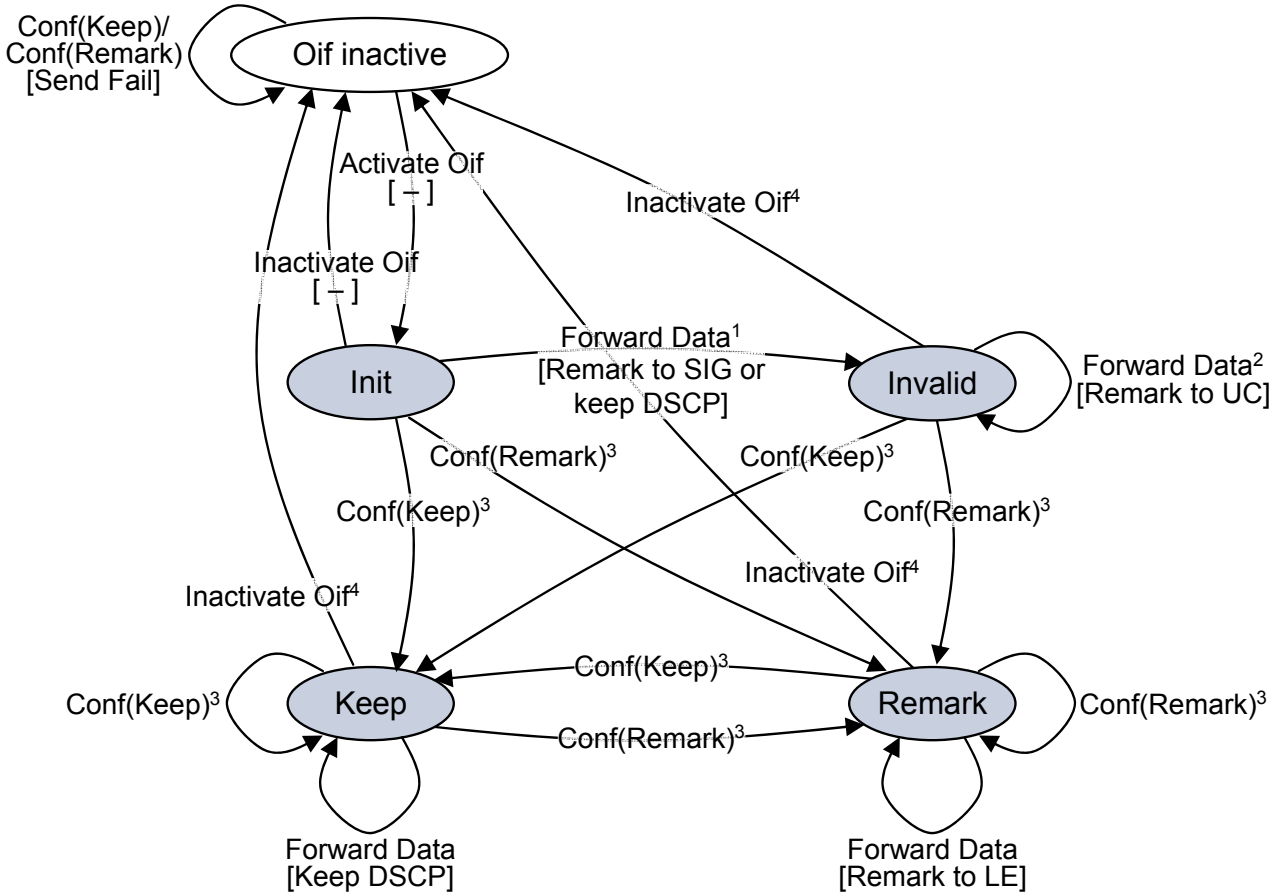
Management qualitätsbasierter Gruppenkommunikation im Internet

Mark Doll



DER ANDERE VERLAG

Mark Doll

Management qualitätsbasierter Gruppenkommunikation im Internet

DER ANDERE VERLAG 2008

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Gedruckt auf säurefreiem, holzfreiem, chlorfrei (TCF) hergestelltem, unbegrenzt alterungsbeständigem Papier nach ANSI-Z 3948 und DIN/ISO 9706 entsprechend der Forderung des Deutschen Bibliotheksinstituts.

Lektorat: Der Andere Verlag, Bergweg 1, 25832 Tönning
Tel. (04861) 610 514, Fax (04861) 610 859
E-Mail: talkto@der-andere-verlag.de
Internet: http://www.der-andere-verlag.de
ISBN: 978-3-89959-791-2
urn:nbn:de:swb:90-93108

Management qualitätsbasierter Gruppenkommunikation im Internet

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

Dissertation

von

Mark Doll

aus Cuxhaven

Tag der mündlichen Prüfung: 14. Dezember 2007

Erster Gutachter:	Prof. Dr. Martina Zitterbart
Zweiter Gutachter:	Prof. Dr. Torsten Braun, Universität Bern

Vorwort

Die vorliegende Arbeit entstand in den letzten zwei Jahren meiner Tätigkeit am Institut für Telematik der Universität Karlsruhe (TH). Frau Prof. Dr. Martina Zitterbart, die mir als Nichtinformatiker die Promotion im reizvollen Bereich der Rechnerkommunikation ermöglichte, gilt mein besonderer Dank. Mein Dank gilt auch Herrn Prof. Dr. Torsten Braun von der Universität Bern für die Übernahme des Zweitreferats.

Eine wissenschaftliche Arbeit, auch wenn sie so praktisch ausgefallen ist wie die vorliegende, ist ohne fachliche Diskussionen unmöglich. Herrn Dr. Roland Bless, dessen Arbeiten den Ausgangspunkt für die vorliegende Arbeit bildeten, und von dessen Wissen ich so viel profitieren konnte, sei an dieser Stelle ganz besonders gedankt. Auch Herrn Dr. Christian Vogt gilt mein Dank für seine hartnäckigen Verständnisfragen, die zur Einführung des vierten NRS-Zustands Init geführt haben. Stellvertretend für die vielen anderen Kollegen am Institut für Telematik, die für das angenehme Arbeitsklima gesorgt haben, danke ich dem harten Kern unseres Wochenausklangs im Vogelbräu, Herrn Detlev Meier und Herrn Dr. Oliver Stanze, die beiden letzten Wegbegleiter aus alten Braunschweiger Tagen, und Lars Völker, der immer allwissend – auf jeden Fall im Bereich Netzsicherheit und beim Mountainbiking – nie darum verlegen war, jedes Aufkommen von Stille im Keim zu ersticken. Neben Dr. Uwe Walters fachlichem Rat werde ich sicher seine unglaublichen „Dr. in Spe“-Präsentationen und Internetfundstücke vermissen.

Dass Promovieren eigentlich sich durchbeißen heißt, glaubt man erst hinterher, trotzdem vielen Dank an Herrn Dr. Uwe Kersting für diese nette, wenn auch ungehört verhallte Vorwarnung. Dass ein Durchbeißen auch ganz persönlicher Unterstützung bedarf, die ich nicht nur von meinem guten Freund Dr. Mark Levermann, sondern vor allem von meiner ganzen Familie, Ruth, Bert und Anja, erfahren habe, ist klar, aber nicht selbstverständlich. Vielen, vielen Dank Euch allen!

Inhaltsverzeichnis

Vorwort **v**

1 Einleitung **1**

2 Grundlagen **3**

2.1 Struktur und Routing des Internets 3

2.1.1 BGP – Border Gateway Protocol 10

2.1.2 Distance Vector Routing 14

2.1.2.1 RIP – Routing Information Protocol 14

2.1.3 Link State Routing 15

2.1.3.1 OSPF – Open Shortest Path First 15

2.1.3.2 IS-IS – Intermediate System to Intermediate System . . . 17

2.1.4 Equal Cost Multipath Routing 18

2.1.5 Netzwerkmanagement mit SNMP 19

2.2 Gruppenkommunikation 20

2.2.1 Systematik der Kommunikationsformen 20

2.2.2 Prinzipien des Multicastroutings 21

2.2.3 Alternative Multicastverfahren 25

2.2.4 Multicastadressen 26

2.2.5 Gruppenmanagement mit IGMP und MLD 29

2.2.6 IGMP/MLD-Proxying 33

2.3 Multicastrouting 35

2.3.1 MFC – Multicast Forwarding Cache 36

2.3.2 PIM – Protocol Independent Multicast 39

2.3.2.1 PIM-SM – PIM Sparse Mode 42
2.3.2.2 PIM-DM – PIM Dense Mode 48
2.3.2.3 BIDIR-PIM – Bi-directional PIM 49
2.3.2.4 BSR – Bootstrap Router Mechanism 50
2.3.3 Historische Multicastroutingprotokolle 51
2.3.3.1 DVMRP – Distance Vector Multicast Routing Protocol . . 51
2.3.3.2 MOSPF – Multicast Open Shortest Path First 53
2.3.3.3 CBT – Core Base Trees 54
2.3.3.4 BGMP – Border Gateway Multicast Protocol 55
2.4 Dienstgüte 55
2.4.1 Diffserv – Differentiated Services 57
2.4.1.1 Class Selector PHB Group 60
2.4.1.2 Expedited Forwarding PHB 60
2.4.1.3 Assured Forwarding PHB Group 61
2.4.1.4 Lower Effort PDB 62
2.4.2 Dienstgütemanagement für Diffserv 62
2.4.2.1 Zentrales Dienstgütemanagement 63
2.4.2.2 Lastverteilung mit Kontingenten 64
2.4.2.3 Messungsbasierte Zugangskontrolle 65

3 Dienstgüteunterstützter Multicast im Internet 67
3.1 Multicast in Diffserv-Netzen – NRS-Problem 67
3.2 DAM – Diffserv-Aware Multicasting 69
3.3 QUASIMODO – EMBAC für PIM-SM 71
3.4 Analyse 74

4 DSMC – Diffserv Multicast 79
4.1 Überblick und Architektur 79
4.2 Basisverfahren 89
4.2.1 Regeln zur Lösung des NRS-Problems 89
4.2.2 Regeln zur Konfiguration des NRS-Zustands 92
4.2.3 NRS-Zustände Init und Invalid 96
4.2.4 DRbit & DSMC-Grenzschnittstelle 96

4.2.5 Triggernachrichten 97
4.2.6 Konfigurationsnachrichten 99
4.2.7 Nachrichtenverlust 100
4.2.8 Nachrichtenvertauschung 102
4.2.9 DRM-seitiges Timing 104
4.2.10 Routerseitiges Timing 106
4.2.11 Beispiele 110
4.2.11.1 PIM-SSM 110
4.2.11.2 PIM-DM 113
4.2.11.3 BIDIR-PIM 114
4.2.12 Weitere Routingverfahren und Multicastroutingprotokolle 117
4.2.12.1 Equal Cost Multipath Routing 117
4.2.12.2 IGMP/MLD-Proxying 118
4.2.12.3 DVMRP 118
4.2.12.4 MOSPF 119
4.2.12.5 CBT 119
4.2.13 Designalternativen 120
4.2.13.1 NRS-Zustand vom MFC-Eintrag trennen 120
4.2.13.2 NRS-Zustand separat für jeden DSCP 120
4.2.13.3 Separate Triggertypen je Multicastroutingprotokoll 121
4.2.13.4 TriggerStop von jedem Router 121
4.2.13.5 Trigger von jedem Router 123
4.2.14 SimpleDSMC 123
4.2.15 Reduzierung getriggerter Conf-Nachrichten für IPv6 124
4.3 Erweiterung für PIM-SM 125
4.3.1 Ergänzende Regeln für PIM-SM 126
4.3.2 Trigger und TriggerRpt 127
4.3.3 Sbit 127
4.3.3.1 TriggerSwitched & TriggerSwitchedAck 129
4.3.3.2 Sofortiger SPT Switchover 131
4.3.4 Beispiel 132
4.3.5 Designalternativen 136

4.3.5.1 Trigger vom Designated Router beim SPT Switchover . . 136
4.3.5.2 TriggerSwitched und TriggerSwitchedAck mittels SNMP . 137
4.4 Ressourcenmanager . . . 137
4.4.1 Voraussetzungen . . . 137
4.4.2 Nachrichtenverarbeitung . . . 139
4.4.3 Multicastpfadbestimmung . . . 146
4.4.4 Behandlung von Inkonsistenzen . . . 147
4.4.4.1 TriggerUC . . . 147
4.4.4.2 Verlust von TriggerStop . . . 148
4.4.4.3 Unerkannte Empfänger . . . 148
4.4.5 Routingänderung . . . 149
4.4.6 PIM-SM . . . 149
4.4.7 Inkrementeller Einsatz . . . 150
4.4.8 DSMC Interdomain . . . 151
4.5 Zusammenfassung . . . 152

5 Implementierung und Evaluierung 155

5.1 FreeBSD . . . 155
5.1.1 Multicastweiterleitung . . . 155
5.1.2 Erweiterte Multicastrouting-API . . . 158
5.1.3 Optionale API-Fähigkeiten . . . 160
5.2 XORP . . . 161
5.2.1 Überblick . . . 162
5.2.2 DSMC-Komponente . . . 167
5.2.3 Modifizierte PIM-SM-Implementierung . . . 169
5.3 Ressourcenmanager . . . 171
5.4 Vergleich mit SimpleDSMC . . . 172
5.5 Zusammenfassung . . . 177

6 Zusammenfassung 179

A Alternative Weiterleitungsarchitekturen 183

B DSMC-Implementierung unter FreeBSD **187**

B.1 Multicastrouting-API 187

B.2 API-Erweiterung – ip_mroute.h 188

B.3 Veränderte Multicastweiterleitung – ip_mroute.c 192

Literaturverzeichnis **199**

Abkürzungsverzeichnis **215**

Index **221**

1 Einleitung

Das heutige Internet arbeitet wie bereits zu seinen Anfängen in den 70er Jahren nach dem so genannten Best-Effort-Servicemodell. Bei ihm werden alle Pakete gleichberechtigt und entsprechend der augenblicklich verfügbaren Ressourcen wie Bandbreite und Pufferspeicher bestmöglich, d. h. mit möglichst geringer Verzögerung und mit möglichst geringem Paketverlust, zum angegebenen Ziel weitergeleitet. Kombiniert mit der Staukontrolle von TCP (Transmission Control Protocol) sorgt Best Effort ansatzweise für eine faire Aufteilung von knappen Ressourcen unter den konkurrierenden Datenströmen.

Dienstgüte (Quality of Service, QoS) verfolgt ein dazu orthogonales Ziel. Im Internet wird mit Hilfe der Differentiated Services (Diffserv) Dienstgüte durch kontrollierte Ungleichbehandlung von Datenströmen bereitgestellt. Die Grundannahme ist, dass gewissen Datenströmen bzw. den dahinter stehenden Anwendungen ein fairer Anteil der knappen Ressourcen nicht genügt. Im Umkehrschluss bedeutet dies, dass einerseits bei fehlenden Ressourcen Datenströme abgewiesen werden müssen und andererseits nicht jeder Datenstrom einen besseren Dienst nutzen darf, sondern nur jene, die ihn tatsächlich erfordern. Anders als Best Effort, wo jeder jederzeit kommunizieren darf, ist zur Bereitstellung von Dienstgüte eine Zugangskontrolle zwingend nötig. Damit ein Dienst garantiert werden kann, muss Verkehr auch nach seiner Zulassung im Rahmen einer Nutzungskontrolle überwacht werden. Beides erfordert die genaue Kenntnis und Verwaltung der verfügbaren Ressourcen. Zugangs- und Nutzungskontrolle auf Basis der Ressourcenverwaltung sind die wesentlichen Aufgaben des Diestgütemanagements.

Kommunikationsbeziehungen im Internet lassen sich anhand ihrer Anzahl an Teilnehmern klassifizieren. Kommunikationsbeziehungen mit einer Quelle und einem Empfänger werden als Unicast bezeichnet, mit zwei und mehr gleichzeitigen Empfängern als Multicast, man spricht von Gruppenkommunikation. Mitte der 1980er Jahre wurde von Deering und Cheriton das Multicastservicemodell für das Internet vorgeschlagen. Die dazu nötige Vervielfachung und Weiterleitung von Paketen nutzt auch heute noch das zuvor Ende der 1970er Jahre von Dalal und Metcalf entwickelte Reverse Path Forwarding (RPF), bei dem die Weiterleitungsentscheidung für ein Paket nicht wie bei Unicast anhand des Ziels, sondern anhand seiner Herkunft getroffen wird. Multicastrouting bestimmt aus dem durch RPF definierten Spannbaum anschließend einen Teilbaum, der alle Empfänger umfasst.

Kann Zugangs- und Nutzungskontrolle bei Unicast allein durch die Überwachung des Verkehrs am Netzeingang unter Berücksichtigung der im Paket vermerkten Zieladresse durchgeführt werden, muss bei Multicast zusätzlich die Vervielfältigung von Paketen im Netzinneren mit einbezogen werden. Diese ist jedoch abhängig von Lage und Anzahl der Empfänger und kann sich fortlaufend ändern. In dieser Arbeit wird daher das Verfahren DSMC (Diffserv Multicast) entworfen, das eine geeignete Erweiterung der Multicastweiterleitungsfunktion in den netzinternen Knoten (Routern) sowie eine zugehörige Signalisierung zwischen Routern und Dienstgütemanagement spezifiziert. Mit DSMC kann das Dienstgütemanagement die Ressourcennutzung von Multicast im Netzinneren kontrollieren und Dienstgüte für Gruppenkommunikation bereitstellen.

Gliederung der Arbeit

Kapitel 2 führt in die Struktur und das Routing im Internet mit Schwerpunkt auf der Gruppenkommunikation ein. Es werden wesentliche Aspekte der aktuellen Dienstgütearchitektur, den Differentiated Services, und dafür vorgeschlagene Ansätze für das Dienstgütemanagement erläutert. Anschließend stellt Kapitel 3 bisherige Ansätze für dienstgüteunterstützten Multicast im Internet vor, identifiziert deren Schwächen und formuliert Anforderungen an ein geeignetes Dienstgütemanagement für Multicast. Kapitel 4 erläutert den in dieser Arbeit entworfenen neuartigen Ansatz DSMC (Diffserv Multicast) zum Dienstgütemanagement von Multicast im Internet. Der Ansatz wurde prototypisch implementiert und evaluiert, die Ergebnisse stellt Kapitel 5 vor. Abschließend werden die wesentlichen Aspekte von DSMC in Kapitel 6 noch einmal zusammengefasst und ein Ausblick auf mögliche zukünftige Weiterentwicklungen gegeben.

2 Grundlagen

Im Folgenden wird ein Überblick über die im Internet standardisierten Architekturen und Protokolle zum Routing, für die Gruppenkommunikation und zur Dienstgüteunterstützung gegeben. Die im weiteren Verlauf verwendeten Begriffe werden eingeführt und ausgewählte Aspekte einiger Protokolle und Verfahren werden mit Hinweisen auf weiterführende Quellen kurz dargestellt.

2.1 Struktur und Routing des Internets

TCP/IP-Schichtenmodell

Das Internet ermöglicht die Kommunikation zwischen weltweit verteilten Computersystemen, im Folgenden Knoten (Node) genannt. Diese recht komplexe Aufgabe wird durch eine Untergliederung der zu bewältigenden Teilaufgaben in Schichten handhabbar gemacht. Mitte der 1980er Jahre wurde das ISO/OSI-Referenzmodell entwickelt [102]. Es gliedert die Teilaufgaben in sieben Schichten, vier niedere transportorientierte und drei höhere anwendungsorientierte Schichten. Die mit ihm eingeführte Nummerierung und Benennung der Schichten ist heute allgemein gebräuchlich. Das etwa zehn Jahre früher aus praktischen Erwägungen heraus entstandene TCP/IP-Schichtenmodell des Internets definiert dagegen nur vier Schichten (s. Abbildung Abbildung 2.1). Den Kern des TCP/IP-Schichtenmodells bilden die Transportschicht und die Vermittlungsschicht. Sie entsprechen der 4. und der 3. Schicht des ISO/OSI-Referenzmodells. Die drei anwendungsorientierten Schichten 5–7 des ISO/OSI-Referenzmodells werden im TCP/IP-Schichtenmodell zu einer, der Anwendungsschicht, zusammengefasst. Die untersten zwei Schichten 1 und 2, die physikalische Schicht und die Sicherungsschicht bilden den Netzzugang. Auf die aufgeführten Beispielprotokolle wird später noch etwas näher eingegangen. Ein Verzeichnis der Abkürzungen findet sich auf Seite 215.

Jede Schicht stellt der darüber liegenden Schicht einen Dienst bereit und nutzt hierzu ihrerseits den ihr von der darunter liegenden Schicht zur Verfügung gestellten Dienst. Die Sicherungsschicht ermöglicht die Kommunikation zweier direkt miteinander verbundener Knoten. Die Vermittlungsschicht erweitert dies zur Kommunikation über ein oder mehrere

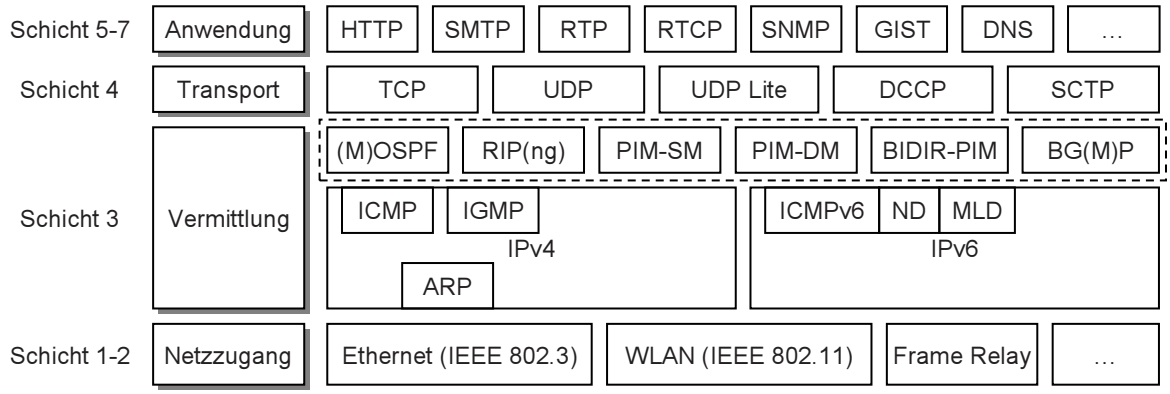


Abbildung 2.1 TCP/IP-Schichtenmodell mit Beispielprotokollen aus jeder Schicht. Die entsprechenden Schichten des ISO/OSI-Referenzmodells sind links angegeben. Routingprotokolle sind gestrichelt umrandet. Transportschichtprotokolle setzen direkt auf IPv4/IPv6 auf.

Knoten hinweg. Sie übernimmt daher Wegewahl (Routing) und Weiterleitung (Forwarding) des Verkehrs vom sendenden Knoten, der Quelle, über ein oder mehrere Zwischenknoten, den Routern, zu einem, oder im Falle der Gruppenkommunikation zu mehreren, Zielknoten bzw. Empfänger(n). Knoten werden entsprechend in Zwischenknoten, die Router, und Endknoten, die Hosts, unterschieden, je nachdem, ob sie Verkehr weiterleiten oder nicht. Ein Router hat immer mehrere Netzwerkschnittstellen, ein Host typischerweise nur eine. Hosts mit mehr als einer Verbindung zu anderen Knoten werden als multihomed bezeichnet. Die Transportschicht schließlich stellt Verbindungsendpunkte für die Anwendungsschicht bereit. Je nach Transportschichtprotokoll erlaubt sie eine zuverlässige oder unzuverlässige, verbindungslose oder verbindungsorientierte, datenstrom- oder datagrammorientierte Datenübertragung zwischen Anwendungen auf verschiedenen Knoten im Internet. Die Transportschicht ist die höchste der transportorientierten Schichten und findet sich wie die Anwendungsschicht nur auf den Endsystemen, den Hosts.

Eine Verbindung zwischen zwei benachbarten Knoten wird im Folgenden als Link bezeichnet. Im Englischsprachigen ist daneben der Begriff „Network“ gebräuchlich. Netz(werk) soll nachfolgend jedoch nur in seiner anderen Bedeutung als eine Menge von über Links miteinander verbunden Knoten verwendet werden. Der Anschluss eines Knotens an einen Link wird Netzwerkschnittstelle oder kurz Schnittstelle genannt. Jeder Knoten besitzt neben diesen physikalischen Schnittstellen zusätzlich mindestens eine Schnittstelle ohne physikalische Entsprechung, die Loopbackschnittstelle. Alles über sie Gesendete wird vom Knoten selbst empfangen und erlaubt Anwendungen auf demselben Knoten, miteinander über Netzwerksockets (s. u.) zu kommunizieren. Ein Link kann sowohl einem physikalischen Medium (Kupferkabel, Glasfaser, etc.) als auch mehreren über Geräte auf Schicht 1 (Repeater) oder Schicht 2 (Bridge, Switch) miteinander verbundenen Medien entsprechen. Umgekehrt können sich mittels Multiplexen auf Schicht 1, z. B. Wellenlängenmultiplex (DWDM), oder auf Schicht 2, z. B. VLAN-Tagging (IEEE 802.1Q [94]), mehrere Links ein und dasselbe physikalische Medium teilen. Aus Sicht eines Knotens ist dies unerheblich und wird im Folgenden nicht mehr unterschieden. Sind genau zwei Knoten miteinander verbunden, spricht man von einer Punkt-zu-Punkt-Verbindung (Point-to-Point Network), bei mehr als zwei Knoten wird je nach Rundruffähigkeit des Mediums bzw. der Siche-

rungsschicht in rundruffähige (Broadcast Capable Network, z. B. Ethernet [95], WLAN[1] [93])) und nicht rundruffähige Links (Nonbroadcast Multiple Access (NBMA) Network), z. B. Frame Relay) unterschieden. Die für das Internet bzw. dessen Vermittlungsschicht standardisierten Verfahren setzen im Allgemeinen eine rundruffähige Sicherungsschicht voraus. Eine der wenigen Ausnahmen bildet OSPF (Abschnitt 2.1.3.1), dass auch NBMA-Links unterstützt. Die Rundruffähigkeit dient einerseits der Auflösung von Vermittlungs- in Sicherungsschichtadressen (s. folgender Abschnitt), und wird andererseits intensiv bei der Gruppenkommunikation (s. Abschnitt 2.2) eingesetzt. Gruppenkommunikation im Internet ist aber auch über NBMA-Links möglich, indem diese z. B. als mehrere (statisch konfigurierte) Punkt-zu-Punkt-Verbindungen aufgefasst werden.

Das Internetprotokoll

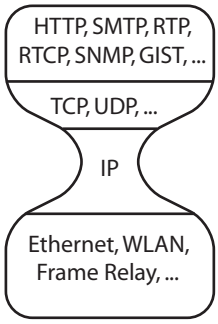


Abbildung 2.2 Anschauliches Sanduhrmodell zur Verdeutlichung der zentralen Rolle von IP als alleiniges Vermittlungsschichtprotokoll im Internet.

Das Internetprotokoll (Internet Protocol, IP) in Version 4 (IPv4) [139] und Version 6 (IPv6) [48] ist das zentrale Protokoll der Vermittlungsschicht. IPv6 wurde Anfang der 1990er Jahre entwickelt und soll in der nahen Zukunft IPv4 ablösen. Derzeit werden beide Versionen parallel eingesetzt. Das Internetprotokoll ermöglicht die Kommunikation über eine Vielzahl unterschiedlicher Schicht-2-Technologien hinweg. Auch oberhalb von IP findet eine Anpassung an die unterschiedlichen Anforderungen, hier der Anwendungen, durch unterschiedliche Protokolle in der Transport- und Anwendungsschicht statt. Man spricht anschaulich vom Sanduhrmodell (Hourglass Model) mit IP als alleinigem alles verbindende Protokoll in der Taille der Sanduhr (Abbildung 2.2). Routingprotokolle, in Abbildung 2.1 gestrichelt umrandet, werden von ihrer Funktion zur Vermittlungsschicht gerechnet, nutzen ihrerseits aber auch IP zur Übertragung.

Das Internetprotokoll arbeitet nach dem Prinzip der Paketvermittlung und stellt der darüber liegenden Transportschicht einen verbindungslosen, unzuverlässigen Datagrammdienst bereit. Die Kommunikation der Protokolle oberhalb der Vermittlungsschicht findet direkt zwischen Endsystemen statt, ihre Daten werden von Zwischenknoten nicht interpretiert und unverändert weitergeleitet, die Zwischenknoten sind sozusagen „dumm“. Jeder über diesen unzuverlässigen, verbindungslosen Datagrammdienst der Vermittlungsschicht hinausgehende Dienst wird allein in den Endsystemen umgesetzt (Ende-zu-Ende-Prinzip). Zwischenknoten sind in anderer Funktion meist gleichzeitig auch Endknoten, z. B. im Rahmen des Netzwerkmanagements (s. Abschnitt 2.1.5).

[1] im Ad-Hoc-Modus innerhalb eines IBSS; Broadcasts im Infrastrukturmodus erfolgen meist indirekt über den Access Point des BSS.

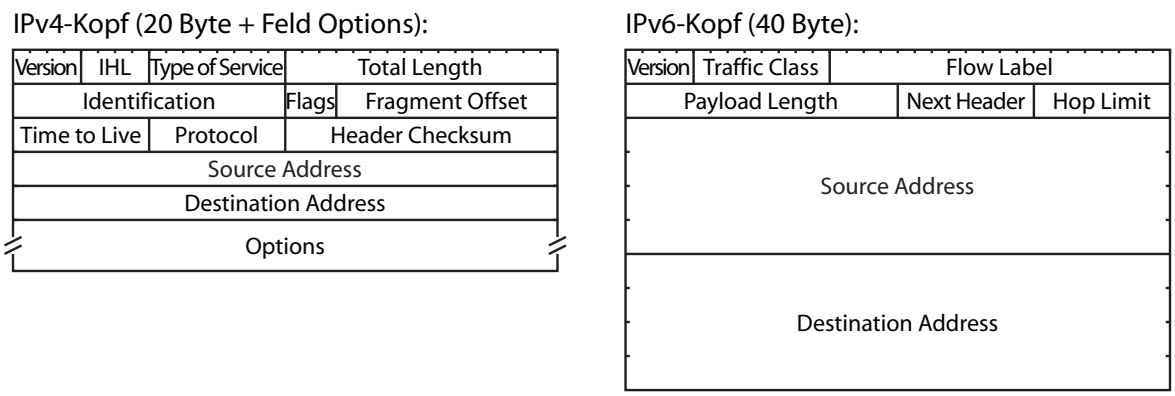


Abbildung 2.3 Aufbau des IP-Kopfs, links Version 4 (nach [139]), rechts Version 6 (nach [48]).

Die Übertragung der Daten erfolgt in IP-Paketen (IP Datagram), im Folgenden auch kurz Paket genannt. Ein IP-Paket besteht aus den von der Transportschicht erhaltenen Daten, denen ein meist (ohne das selten genutzte Optionenfeld) 20 Byte langer IPv4-Kopf vorangestellt ist. Der IPv6-Kopf besitzt kein Optionenfeld und ist fix 40 Byte lang. Stattdessen wird der IPv6-Kopf bei Bedarf, z. B. für Fragmentierung oder Verschlüsselung, durch IPv6-Erweiterungsköpfe ergänzt. Abschnitt 4.2.15 skizziert einen solchen IPv6-Erweiterungskopf zum Zwecke der Konfiguration der Paketweiterleitung für eine dienstgüteunterstützte Gruppenkommunikation. Jeder IP-Kopf enthält u. a. die 32 bit (IPv4) bzw. 128 bit (IPv6) langen IP-Adressen von Quell- und Zielknoten. Um kommunizieren zu können, benötigt daher jeder Knoten im Internet eine innerhalb ihres Gültigkeitsbereichs (z. B. weltweit) eindeutige IP-Adresse, im Folgenden kurz Adresse genannt.[2] Adressen sind immer an eine Netzwerkschnittstelle gebunden, man spricht üblicherweise aber vereinfachend von der Adresse eines Knotens. Eine Schnittstelle kann beliebig viele Adressen tragen. Eine Adresse gliedert sich in Präfix und Schnittstellenidentifikation. Die Adressen aller an einem Link angeschlossenen Knoten tragen dasselbe Präfix, das (Sub-)Netzwerkpräfix. Es stellt quasi die Adresse eines Links dar. Der Aufbau von Multicastadressen, mit denen Gruppen von Empfängerknoten adressiert werden, wird in Abschnitt 2.2.4 besprochen.

Abbildung 2.3 zeigt den IP-Kopf in Version 4 und Version 6. Der IP-Kopf enthält Felder mit der Version (4 oder 6), der Länge des IP-Pakets inklusive Kopf (IPv4, Total Length) bzw. exklusive Kopf (IPv6, Payload Length) und der Quell- und der Zieladresse (Source Address, Destination Address). Das Feld Time to Live bzw. Hop Limit begrenzt die aufgrund von fehlerhaftem Routing entstehende Last durch endlos kreisende Pakete. Der Wert des Felds wird bei jeder Weiterleitung durch einen Router um eins dekrementiert und das Paket bei Erreichen von Null verworfen, so dass die maximale Anzahl an Weiterleitungen bzw. Hops begrenzt bleibt. Es wird auch bei Gruppenkommunikation zur Reichweitenbegrenzung verwendet (s. Abschnitt 2.2.4). Das Feld Protocol bzw. Next Header enthält die das Transportschichtprotokoll identifizierende Protokollnummer[3]. Sind bei IPv6 Erweiterungsköpfe

[2] Ein Knoten, der noch keine Adresse hat, aber zur Beschaffung einer Adresse kommunizieren muss, kann als Quelladresse die unspezifizierte Adresse (Unspecified Address, IPv4 0.0.0.0 bzw. IPv6 ::) verwenden. Sie kann nicht als Zieladresse verwendet werden und eignet sich nur zur Kommunikation innerhalb eines Links.

[3] verwaltet von der Internet Assigned Numbers Authority (IANA), aktuelle Liste unter [92]

eingefügt, nennt das Feld die Protokollnummer des nachfolgenden Erweiterungskopfs. Dieser wiederum enthält ebenfalls ein Feld Next Header, das den Typ des nächsten Erweiterungskopfs anzeigt usw. Die Erweiterungsköpfe bilden dadurch eine Art einfach verkettete Liste mit dem Feld Next Header als Zeiger. Die Protokollnummer des Transportschichtprotokolls befindet sich dann im Feld Next Header des letzten Erweiterungskopfs. Die Erweiterungsköpfe von IPv6 ersetzen und erweitern die Funktion des Optionenfelds von IPv4. Durch das kurze Feld IHL (Internet Header Length) auf nur 40 Byte begrenzt, war das Optionenfeld von IPv4 in seinen Nutzungsmöglichkeiten ohnehin deutlich einschränkt. IPv4-Optionen werden daher heutzutage praktisch nicht mehr genutzt. Die Felder Identification, Flags und Fragment Offset dienen bei IPv4 der Fragmentierung, wenn die von der Transportschicht gelieferte zu übertragene Dateneinheit zuzüglich IP-Kopf größer ist als die von der Sicherungsschicht maximal übertragbare Dateneinheit. Die Fragmentierung ist bei IPv6 in einen separaten Erweiterungskopf ausgegliedert, da sie nur sehr selten verwendet wird. Die Prüfsumme (Feld Header Checksum) ist bei IPv6 entfallen, da die Prüfsumme der Transportschichtprotokolle bereits die wichtigsten Felder des IP-Kopfs mit einbezieht (Protokollnummer, Adressen und Länge der Transportschichtdateneinheit). Nur IPv6 kennt das Feld Flow Label, das weiter unten erläutert wird. Auf das Feld Type of Service (IPv4) bzw. Traffic Class (IPv6) wird bei der Besprechung von Dienstgüte in Abschnitt 2.4 noch näher eingegangen.

Die zur Kommunikation innerhalb eines Links zwischen benachbarten Knoten nötige Auflösung von IP-Adressen in die der Sicherungsschicht erfolgt im Falle von rundruffähigen Links bei IPv4 mittels ARP (Address Resolution Protocol) [137], bei IPv6 mittels Nachbarerkennung (Neighbor Discovery) [125], einer Teilfunktionalität von ICMPv6 (Internet Control Message Protocol) [44]. Eine ARP-Anfrage geht per Broadcast an alle an einen Link angeschlossene Knoten. Jener Knoten, der die in der ARP-Anfrage genannte IP-Adresse besitzt, antwortet mit seiner Sicherungsschichtadresse. Die Nachbarerkennung von IPv6 nutzt dagegen eine feste Abbildung[4] von Gruppenadressen der Vermittlungsschicht auf die Gruppenadressen der Sicherungsschicht. Jeder IPv6-Knoten hört dazu auf einer bestimmten IPv6-Gruppenadresse, der Solicited Node Multicast Address, die sich nach einem festen Schema[5] aus der IPv6-Adresse eines Knotens ableitet. Eine Anfrage zur Nachbarerkennung wird dann an diese Gruppe von Knoten und nicht mehr per Broadcast an alle gesendet. Grund für die Verwendung von Multicast statt Broadcast ist, dass bei Einsatz von Switches, die den ICMPv6-Verkehr „mitschnüffeln“ (Snooping Switches), die Nachbaranfrage im Regelfall tatsächlich nur den einen Knoten erreicht, dessen Adresse aufgelöst werden soll. Die Broadcasts von ARP müssen dagegen von jedem Knoten am Link empfangen und ausgewertet werden. Bei nicht rundruffähigen Links gibt es kein einheitliches Verfahren zur Adressauflösung. Die Abbildung von IP- auf Sicherungsschichtadressen muss hier entweder mit einem sicherungsschichtspezifischen Protokoll erfolgen oder wird statisch konfiguriert.

IP-Adressen

In Abbildung 2.4 ist der grundsätzliche Aufbau einer IP-Adresse wiedergegeben, genauer, der Aufbau einer einen einzelnen Knoten identifizierenden *Unicastadresse*. Sie besteht aus

[4] MAC48-Gruppenadresse (z. B. Ethernet): 16 bit „Präfix“ 33-33-00-00-00-00 hexadezimal plus die letzten 32 bit der IPv6-Gruppenadresse

[5] Präfix ff02::1:ff00:0:0/104 plus die letzten 32 bit der IPv6-Unicastadresse

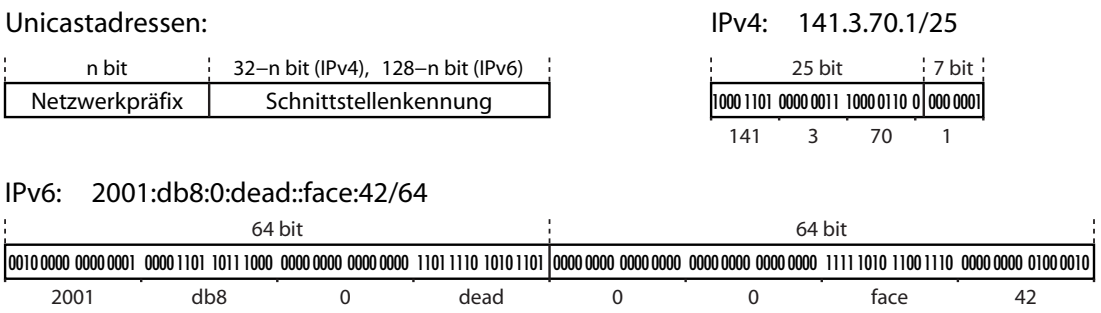


Abbildung 2.4 Aufbau von IP-Unicastadressen (nach [81]) mit je einem Beispiel für IPv4 und IPv6.

einem variabel langen Netzwerkpräfix und einer entsprechend langen Schnittstellenidentifikation, so dass die Gesamtlänge aus Netzwerkpräfix und Schnittstellenidentifikation 32 bit (IPv4) bzw. 128 bit (IPv6) beträgt. IPv4-Adressen werden als vier durch Punkte getrennte Dezimalzahlen geschrieben, IPv6-Adressen als acht durch Doppelpunkte getrennte Hexadezimalzahlen, wobei einmal eine beliebig lange Folge von Nullen durch zwei Doppelpunkte abgekürzt werden kann. Im Beispiel können daher nicht gleichzeitig die Nullen zwischen „db8“ und „dead“ sowie „dead“ und „face“ durch „::“ ersetzt werden, da die Verteilung der Nullen auf die beiden Bereiche nicht mehr eindeutig wäre. Die Präfixlänge wird durch einen Schrägstrich getrennt hinter die Adresse notiert.

Aufgrund der Adressknappheit bei IPv4 finden sich dort in der Praxis alle Präfixlängen ab 30 bit, um den knappen Adressraum effizient zu nutzen. Bei IPv6 sind dagegen von 64 bit abweichende Längen für ein Netzwerkpräfix unüblich. Grund ist, dass eine 64 bit lange Schnittstellenidentifikation bei IPv6 nicht nur händisch oder zustandsbehaftet per DHCPv6 (Dynamic Host Configuration Protocol) [57] sondern auch perzustandsloser Autokonfiguration (Stateless Address Autoconfiguration) generiert werden kann [178]. Hierbei bestimmt ein Knoten autonom eine auf dem Link eindeutige Schnittstellenidentifikation. Ursprünglich wurde diese aus der (weltweit eindeutigen) Sicherungsschichtadresse abgeleitet [57], mittlerweile sind zur Wahrung der Privatsphäre auch Zufallszahlen als Schnittstellenidentifikation möglich [124]. Oder es wird der Hashwert eines öffentlichen Schlüssels in die Schnittstellenidentifikationen integriert (Cryptographically Generated Addresses, CGA [10]), wodurch sich z. B. die Nachbarerkennung (s. u.) absichern (Secure Neighbor Discovery, SEND [6]) oder die Unterstützung für mobile Knoten optimieren lässt [185, 7].

IP-Adressen als Identifikator – Sockets

IP-Adressen haben eine Doppelfunktion als Lokator und Identifikator. Als Identifikator kennzeichnen sie einen Knoten im Internet eindeutig. Zusammen mit den 16 bit langen Portnummern oder kurz Ports, den „Adressen“ der Transportschichtprotokolle[6] TCP (Transmission Control Protocol) [140] und UDP (User Datagram Protocol) [138] bilden sie den Kommunikationsendpunkt für die Anwendungsschicht. Er wird als Netzwerksocket

[6]Neben den beiden ursprünglich einzigen Transportschichtprotokollen TCP und UDP wurde Ende der 1990er Jahre SCTP (Stream Control Transmission Protocol) [165] für die Signalisierung von Sprachverkehr (PSTN) über das Internet entwickelt. Die erst vor wenigen Jahren entstandenen UDP-Varianten UDP-Lite [106] und DCCP (Datagram Congestion Control Protocol) [101] werden dagegen noch kaum eingesetzt.

oder kurz Socket bezeichnet. Das 5-Tupel[7] aus Quell- und Ziel-IP-Adresse, Transportschichtprotokoll und dessen Quell- und Zielport identifiziert einen Socket bzw. dessen Datenstrom (Flow) eindeutig. Da für Router der Zugriff auf die Ports bei eingefügten IPv6-Erweiterungsköpfen das aufwendige Verfolgen aller Next-Header-Felder bis zum Kopf des Transportschichtprotokolls erfordert, wurde im IPv6-Kopf das Feld Flow Label eingeführt. Er erlaubt dem Router, einen Datenstrom anhand des Tripels aus Quelladresse, Zieladresse und Flow Label eindeutig zu identifizieren [148]. Die Eindeutigkeit des Flow Labels muss die Quelle sicherstellen. Die eindeutige Zuordnung von Paketen zu einem Datenstrom wird beispielsweise bei der Bereitstellung von Dienstgüte benötigt (s. Abschnitt 2.4).

In der Anwendungsschicht werden primär Domänennamen, auch DNS-Namen genannt, als Identifikator für Knoten (z. B. www.uni-karlsruhe.de) und Anwendungen bzw. Dienste (z. B. _https._tcp.www.uni-karlsruhe.de) verwendet. Sie werden über das Domain Name System (DNS) [120], einer hierarchisch gegliederten internetweiten Datenbank, auf IP-Adressen, Protokoll- und Portnummern abgebildet. Während die Nutzung des DNS zur Abbildung von Domänennamen auf IP-Adressen[8] allgemein gebräuchlich ist, wird die Abbildung auf Protokoll- und Portnummern[9]nur von wenigen Anwendungen eingesetzt.

IP-Adressen als Lokator – Routing

Als Lokator haben IP-Adressen Einfluss auf die Wegewahl. Adressen werden aus Gründen der Skalierbarkeit der Wegewahl mit zunehmendem Abstand vom Ziel zu Adresspräfixen aggregiert, dargestellt als Tupel aus Präfixadresse und Präfixlänge. Ein Adresspräfix umfasst alle aufeinanderfolgenden Adressen, die in der ersten durch die Präfixlänge angegebenen Anzahl an Bits gleich der Präfixadresse sind. Unicastroutingprotokolle ermitteln bezüglich einer gegebenen Metrik den besten Pfad zu einem Adresspräfix als Ziel. Das Resultat wird als Route bezeichnet. Eine Route ordnet einem Zielpräfix einen Next Hop, den aus Sicht eines Routers nächsten Router auf dem Pfad Richtung Ziel zu. Ist das Zielpräfix gleich dem Netzwerkpräfix eines Links, gibt es keinen nächsten Router und der Next Hop ist die Schnittstelle des Routers zu diesem Link. Eine Route, deren Präfix gleich der Länge der IP-Adressen ist (32 bit bzw. 128 bit), wird als Hostroute bezeichnet. Umgekehrt nennt man die Route mit dem einen Zielpräfix der Länge Null Defaultroute (geschrieben 0/0 (IPv4) bzw. ::/0 (IPv6)). Sie passt auf alle Zieladressen. Routen mit einem längeren und damit spezifischeren Präfix haben unabhängig von ihrer Metrik immer Vorrang vor weniger spezifischen Routen, was als Longest Prefix Match bezeichnet wird. Hostrouten finden entsprechend immer zuerst Anwendung. Die Defaultroute kommt erst zum Tragen, wenn keine passende spezifischere Route existiert.

Routing im Internet – Autonome Systeme

Das heutige Internet besteht aus einem Zusammenschluss von Netzen von einer Vielzahl rechtlich wie wirtschaftlich unabhängiger Netzbetreiber. Routing und Weiterleitung von Verkehr zwischen diesen Netzen erfolgt auf Grundlage zwischen den Betreibern geschlossener bilateraler Verträge. Die politisch wie wirtschaftlich autonome Verwaltung jedes Netzes legt nahe, auch in technischer Hinsicht das Routing in autonome Teilbereiche zu gliedern, die

[7] mehrere Quell- und Zieladressen und -ports bei SCTP

[8] mittels Address Resource Records A (IPv4) [121] und AAAA (IPv6) [177]

[9] mittels Service Resource Records (SRV) [71], ggf. nach vorheriger Delegation über Naming Authority Pointer Resource Records (NAPTR) [114]

sogenannten Autonomen Systeme. Ein Autonomes System, kurz AS, ist definiert als eine Menge von miteinander verbundenen Routern unter gemeinsamer administrativer Kontrolle, die sich nach außen hin gegenüber anderen Autonomen Systemen als eine Einheit mit konsistenter Sicht auf die über sie erreichbaren Ziele darstellt [150]. Die an diese Menge von Routern angeschlossenen Hosts sind Teil des Autonomen Systems. Das Routing zwischen Autonomen Systemen wird als Interdomänenrouting bezeichnet und erfolgt mit einem EGP (Exterior Gateway Protocol), wobei von der IETF (Internet Engineering Task Force), dem Standardisierungsgremium des Internets, derzeit BGP (Border Gateway Protocol) in Version 4 als alleiniges EGP standardisiert wurde [150].

Das Routing innerhalb eines Autonomen Systems wird als Intradomänenrouting bezeichnet und erfolgt mit einem oder auch mehreren IGPs (Interior Gateway Protocol). Eine Routingdomäne oder kurz Domäne ist eine zusammenhängende Menge von Routern, zwischen denen das Routing mit demselben IGP erfolgt. Üblicherweise besitzt ein AS nur eine Routingdomäne und beide Begriffe AS und Domäne bezeichnen dieselbe Menge von Knoten. Als IGP standardisiert sind das Distance-Vector-Routingprotokoll RIP (Routing Information Protocol) in Version 2 [108] für IPv4 und RIPng (RIP Next Generation) in Version 1 [109] für IPv6 sowie die Link-State-Routingprotokolle OSPF (Open Shortest Path First) in Version 2 [123] für IPv4 und Version 3 [43] für IPv6 und das aus der ISO/OSI-Welt stammende IS-IS (Intermediate System to Intermediate System) [96], das gleichermaßen für IPv4 wie IPv6 verwendet werden kann. Die folgenden Abschnitte stellen grundlegende Aspekte der genannten Routingprotokolle näher vor.

2.1.1 BGP – Border Gateway Protocol

Interdomänenrouting mit BGP Version 4 [150] ist von zwei wesentlichen Aspekten geprägt: Pfadattribute und Policies (Richtlinien). Eine Route in BGP fasst die Menge aller Ziele mit dem Pfad, über den diese Ziele erreicht werden, zusammen. Ziele werden in Form eines IP-Adresspräfixes angegeben, bei BGP bezeichnet als Network Layer Reachability Information (NLRI). Ein Pfad wird durch die Menge aller Pfadattribute repräsentiert. Eine Route in BGP erweitert damit die klassische aus Zielpräfix und Next Hop bestehende Route durch weitere Pfadattribute. Die Route stellt die zentrale Informationseinheit dar, die zwischen benachbarten BGP-Routern ausgetauscht wird.

Neben dem Pfadattribut Next Hop (NEXT_HOP) ist eines der Wichtigsten der für die Pfadattribute namensgebende Pfadvektor (AS_PATH). Er listet die auf dem Weg zum Ziel passierten Autonomen Systeme der Reihe nach auf, bei aggregierten Routen z. T. auch ungeordnet als Menge. Hierfür besitzt jedes AS eine eindeutige 16 bit lange AS-Nummer, mit der jeder BGP-Router im AS konfiguriert ist. Die Umstellung auf 32 bit lange AS-Nummern ist vorgesehen [186]. Gibt ein Router einem benachbarten Router mit einer anderen AS-Nummer eine Route bekannt, fügt er seine eigene AS-Nummer vorne an den Pfadvektor an. Der Pfadvektor erlaubt so die Erkennung von Schleifen. Findet ein Router seine eigene AS-Nummer im Pfadvektor einer empfangenen Route, ignoriert er diese. Zudem dient die Länge des Pfadvektors, also die Anzahl seiner Listenelemente[10], als Routingmetrik. Die Routingmetrik entspricht damit dem üblichen Hop Count, d. h. die Anzahl an zu passierenden Knoten bis zum Ziel, allerdings auf Ebene der Autonomen Systeme anstatt auf Ebene der Router.

[10] Ein Element ist entweder eine AS-Nummer oder eine Menge von AS-Nummern.

Das Routing zwischen Autonomen Systemen wird darüber hinaus ganz wesentlich von Policies beeinflusst. Gibt ein Router aus einem AS seinem Nachbarrouter in einem anderen AS eine Route bekannt, so impliziert dies, dass sein AS bereit ist, den Verkehr für die bekanntgegebenen Zielpräfixe weiterzuleiten. Ob eine Weiterleitung überhaupt zulässig oder gewünscht ist, hängt von den mit Betreibern benachbarter Autonomer Systeme geschlossenen Verträgen („Peering Agreement“) wie auch von politischen Entscheidungen des Betreibers selbst ab. Diese Bedingungen werden durch Policies repräsentiert und haben Einfluss auf alle wesentlichen Aspekte des BGP-Routings. Sie bestimmen mit der lokalen Präferenz (Pfadattribut LOCAL_PREF) die wichtigste Metrik für eine Route. Policies bestimmen, welche Pfadattribute beim Weiterleiten an einen Nachbarrouter gesetzt, entfernt oder modifiziert werden und nehmen damit über den Nachbarrouter hinaus Einfluss auf die weitere Verbreitung einer Route. Oder sie unterbinden die weitere Verbreitung einer Route gänzlich. Die Länge des Pfadvektors kommt erst danach als erster von mehreren Tie-Breakern zum Tragen, sollten mehrere Routen dieselbe lokale Präferenz aufweisen.

BGP nutzt TCP für den zuverlässigen Austausch seiner Nachrichten. Der Begriff Nachricht soll im Folgenden für eine im Rahmen des Routings oder Netzwerkmanagements ausgetauschte Dateneinheit stehen. Jede Nachbarschaftsbeziehung bzw. Sitzung zwischen zwei BGP-fähigen Routern, auch BGP-Sprecher genannt, wird fest vom Betreiber vorkonfiguriert, eine selbständige Erkennung von Nachbarroutern (Peer) ist bei BGP nicht vorgesehen. Dies entspricht dem Vorgehen beim Interdomänenrouting, das für die Etablierung einer BGP-Sitzung einen zwischen den Betreibern geschlossenen Vertrag voraussetzt.

Internal BGP

BGP unterscheidet zwei Arten von Verbindungen zwischen Nachbarroutern. Internal BGP (IBGP) bezeichnet die Verbindung zwischen zwei Routern desselben AS, External BGP (EBGP) die zwischen zwei Routern aus unterschiedlichen Autonomen Systemen. External BGP ist das eigentliche Exterior Gateway Protocol, während Internal BGP nur dazu dient, dass jeder Router seine via EBGP gelernten Routen allen anderen Routern in seinem AS bekanntgeben kann. Dadurch wird erreicht, dass alle Router in einem AS eine konsistente Sicht auf alle Routen zu externen Zielen außerhalb ihres eigenen AS besitzen, so wie es per Definition für ein AS verlangt wird. Für eine konsistente Sicht auf die AS-internen Routen sorgt das Intradomänenrouting.

IBGP unterscheidet sich in einigen Punkten von EBGP, u. a. wird der Pfadvektor wird nicht modifiziert und steht somit nicht zur Erkennung von Schleifen (innerhalb des AS) zur Verfügung. Um trotzdem Schleifenfreiheit sicherzustellen, dürfen von einem IBGP-Nachbarn gelernte Routen nicht an einen anderen IBGP-Nachbarn weitergegeben werden (sondern nur an einen EBGP-Nachbarn). Als Konsequenz daraus müssen die Router eines AS per IBGP vollvermascht sein. Da dies mit zunehmender Anzahl an Routern nicht mehr skaliert, wurden mit Routen-Reflexion [17] und AS-Konföderationen [180] zwei Verfahren standardisiert, um auch große und sehr große Autonome Systeme mit vielen hundert Routern und darüber hinaus zu ermöglichen. Beide gestatten es, ohne auf IBGP-Ebene tatsächlich vollvermascht zu sein, dass jeder Router von jedem anderen in seinem AS alle externen Routen erhält.

Da BGP auf TCP aufsetzt, müssen in BGP benachbarte Router nicht notwendigerweise auch physikalisch benachbart sein. Während zwei EBGP-Nachbarn dies meist sind und mit EBGP gleichzeitig die Verbindung hinsichtlich Ausfall des sie verbindenden physikalischen

Links überwacht wird, ist dies für in IBGP benachbarte Router wegen der Vollvermaschung nicht praktikabel und auch nicht nötig. AS-interne Linkausfälle erkennt das dort eingesetzte IGP und sorgt wenn möglich für ein Rerouting. Die IBGP-Sitzungen werden typischerweise zwischen Adressen aufgebaut, die der Loopbackschnittstelle des Routers zugeordnet sind. Diese Adressen werden ebenfalls vom IGP geroutet. Würden die IBGP-Sitzungen eine Adresse einer physikalischen Schnittstelle verwenden, würden mit dem Ausfall dieser Schnittstelle auch alle IBGP-Sitzungen beendet, da die Adresse nicht mehr erreichbar wäre. Durch Verwendung der Loopbackschnittstelle bleiben IBGP-Sitzungen unabhängig von der zugrundeliegenden physikalischen Netzwerktopologie und den Ausfällen darin.

Normalerweise nehmen alle Router eines Autonomen Systems am BGP-Routing teil, da das IGP normalerweise nur Routen zu AS-internen Zielen kennt und routet. Ein Router, der nur am IGP und nicht auch am IBGP teilnimmt, kennt keine Zieladressen außerhalb des eigenen Autonomen Systems und kann Pakete mit solchen Adressen nicht mehr korrekt weiterleiten. Will man solchen Nicht-BGP-Routern trotzdem das Weiterleiten ermöglichen, müssen alle aktiven (d. h. besten) externen, also über EBGP gelernten, Routen in das IGP eingespeist werden. IGPs sind allerdings mit den derzeit mehreren hunderttausend [87] aktiven Interdomänenrouten überfordert [166]. Das Einspeisen in das IGP kann aber dort sinnvoll sein, wo nur wenige externe Routen benötigt werden oder gar Defaultrouten ausreichend sind. Dies gilt typischerweise für ein Stub-AS, also ein AS ohne Transitverkehr. Welche Routen ins IGP einzuspeisen sind, wird über geeignete Policies gesteuert.

Als Endergebnis des BGP-Routings bestimmt jeder Router gemäß eines festgelegten und von den Policies beeinflussten Entscheidungsprozesses die jeweils beste Route zu einem Zielpräfix und legt diese in der lokalen Routing Information Base (RIB) ab. Diese über BGP gelernten Routen fließen zusammen mit Routen aus anderen Quellen (IGP, statisch konfigurierte Routen, direkt angeschlossene Links) in eine routingprotokollübergreifende RIB oder Routingtabelle ein. Hieraus wird schließlich die Forwarding Information Base (FIB) generiert, anhand der im Router die eigentliche Paketweiterleitung stattfindet. Die FIB enthält zu jedem Zielpräfix den Next Hop, an den der Verkehr für dieses Ziel weitergeleitet werden muss, idealerweise schon aufgelöst in Ausgangsschnittstelle und Sicherungsschichtadresse des Next Hop.

Abbildung 2.5 zeigt als Beispiel ein Autonomes System A mit $N = 4$ Routern R1 bis R4. Sie sind über $N \cdot (N-1)/2 = 6$ IBGP-Sitzungen miteinander vollvermascht (gepunktete Linien). Über Links physikalisch vollvermascht (durchgezogene Linien) sind dagegen nur die drei Router R1 bis R3, Router R4 ist lediglich mit Router R3 verbunden. Die beiden IBGP-Sitzungen von R4 werden daher vom Intradomänenrouting über Router R3 geleitet. EBGP-Sitzungen zu den benachbarten Autonomen Systemen AS B und AS C unterhalten die Grenzrouter R1, R2 und R4 (gestrichpunktete Linien), der innere Router R3 besitzt nur IBGP-Sitzungen. Jeder Link von AS A besitzt ein eigenes Netzwerkpräfix, in der Abbildung als n1, n2, n4, n12, n13, n23 und n34 bezeichnet. Sie werden wie die der Loopbackschnittstelle der Router R1 bis R4 zugeordneten Adressen a1 bis a4 durch ein (beliebiges) Intradomänenroutingprotokoll geroutet. Wie bereits erläutert, nehmen normalerweise alle Router eines Autonomen Systems am BGP-Routing teil. Würde Router R3 nicht am IBGP teilnehmen, wüsste R3 nicht, wohin z. B. Pakete mit Zieladressen aus dem AS C, beispielsweise die des Netzwerkpräfixes c24 zwischen Router C2 und C4, weiterzuleiten wären. Schließlich kennt das Intradomänenrouting nur die Adressen des eigenen AS A. Eine Ausnahme bildet ein Stub-AS. Wäre AS A ein solches (multihomed)

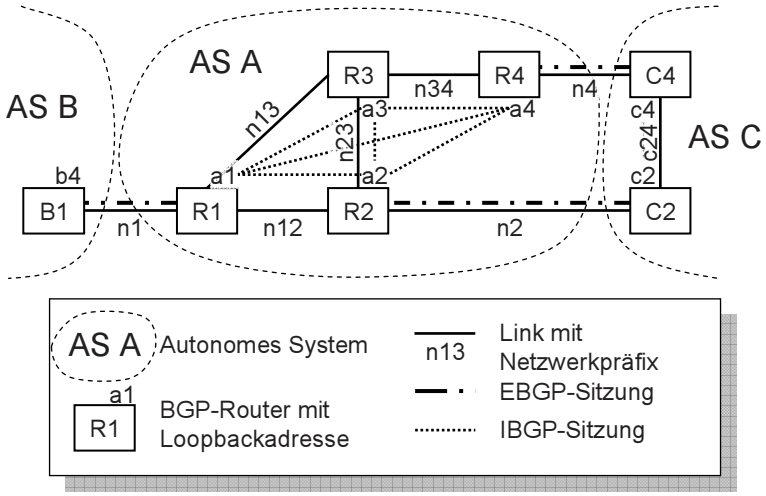


Abbildung 2.5 Beispieltopologie zur Veranschaulichung des Zusammenwirkens von EBGP, IBGP und IGP.

Stub-AS, könnte z. B. eine lokale Policy auf den Routern R2 und R4 besagen, dass eine Defaultroute ins Intradomänenrouting einzuspeisen wäre. R1 würde nur Routen einspeisen, die zu Zielen in AS B führen. R3 leitet Verkehr für Ziele außerhalb von AS A und nicht aus AS B an Router R2 oder R4 weiter, je nachdem, welche der beiden Defaultrouten die niedrigeren Kosten im Intradomänenrouting aufweist. Ziele in AS B werden an Router R1 weitergeleitet, da gemäß Longest Prefix Match spezifischere Routen Vorrang vor Defaultrouten haben.

Erweiterungen für IPv6 und Multicast

BGP wurde ursprünglich nur für IPv4 entworfen. BGP verwendet jedoch das flexible Typ-Länge-Wert-Kodierungsschema, das ein unkompliziertes Erweitern von BGP erlaubt, z. B. beim geplanten Umstieg auf 32 bit lange AS-Nummern. Eine der wichtigsten BGP-Erweiterungen ist die Multiprotokollerweiterung [16]. Mit dieser Erweiterung ist es möglich, Routen für beliebige Adressfamilien[11] und damit auch für IPv6 auszutauschen. Ein IPv4 und IPv6 routendes BGP Version 4 wird häufig als BGP4+ bezeichnet. Mit der Multiprotokollerweiterung können des Weiteren Routen für Unicast und Multicast getrennt angegeben werden. Dies erlaubt zum einen, Multicast- und Unicastverkehr zu separieren, zum anderen können Netzbereiche, die kein Multicastrouting (s. Abschnitt 2.3) beherrschen, vom Multicastverkehr ausgenommen werden. Weil für Multicast separate Erreichbarkeitsinformationen ausgetauscht werden, existiert für Multicast auch eine separate Multicast Routing Information Base (MRIB). Die bisher für das Internet standardisierten IGPs, ausgenommen OSPF mit seiner Erweiterung MOSPF (s. Abschnitt 2.1.3.1 bzw. Abschnitt 2.3.3.2), unterscheiden nicht zwischen Routen für Unicast und Multicast, so dass die von ihnen stammenden Routen in der MRIB mit denen in der RIB identisch sind. Auf Basis der MRIB findet später das Multicastrouting statt, dessen Ergebnis der Multicast Forwarding Cache (MFC) ist. Er kann als Äquivalent zur Forwarding Information Base von Unicast gesehen werden. Der MFC wird in Abschnitt 2.3.1 im Rahmen des Multicastroutings noch ausführlich besprochen.

[11] verwaltet von der IANA, aktuelle Liste unter [88]

2.1.2 Distance Vector Routing

Das Distance-Vector-Routing ist das erste Routingverfahren, das im Internet bzw. in dessen Vorläufer ARPANET (Advanced Research Projects Agency Network) Anwendung fand. Beim Distance-Vector-Routing erfolgt die Berechnung der kürzesten Pfade mit einem verteilt arbeitenden Bellman-Ford-Algorithmus. Der aktuell einzige für das Internet standardisierte Vertreter dieses Prinzips ist das Routing Information Protocol (RIP). Die erste Version von RIP entstand Ende der 1980er Jahre, aktuell ist die im Folgenden beschriebene Version 2 [108].

2.1.2.1 RIP – Routing Information Protocol

RIP nutzt UDP zur Übertragung seiner Nachrichten. Als Metrik wird der Hop Count verwendet. Der Hop Count 16 ist als unendlich definiert und kennzeichnet nicht (mehr) erreichbare Ziele. Dadurch bleibt der Durchmesser eines mit RIP gerouteten Netzes auf maximal 15 Hops beschränkt. Der Wert stellt einen Kompromiss zwischen schneller Konvergenz (s. Count to Infinity unten) und maximaler Netzgröße dar. Bei RIP senden Router allen ihren Nachbarn periodisch eine komplette Routingtabelle mit allen ihnen jeweils bekannten besten Routen. Router kennen nur den jeweils nächsten Hop zum Ziel – der Router, von dem sie die Route empfangen haben. Sie kennen also nicht den weiteren Verlauf des Pfads zum Ziel, wie dies beim Link-State-Routing (zumindest innerhalb eines Gebiets) möglich ist.

Um Schleifen nach einem Linkausfall zu vermeiden, werden die Routen gemäß *Split Horizon with Poisoned Reverse*[12] derart modifiziert, dass von einem Nachbarn gelernte Routen an diesen mit der Metrik unendlich („Poisoned“) zurückgesendet werden. Routingschleifen über drei oder mehr Router werden durch Zählen bis unendlich, *Count to Infinity*, behandelt und entfernt. Um hierbei die Konvergenz zu beschleunigen, werden Änderungen per *Triggered Update* sofort und nicht erst mit der nächsten periodischen Wiederholung gesendet. RIP arbeitet nach dem *Soft-State-Prinzip*, d. h. der durch empfangene Nachrichten etablierte Zustand bleibt nur solange erhalten, wie er durch Wiederholungen der Nachrichten aufgefrischt wird. Erhält ein Router für eine gewisse Zeit (drei Minuten) für ein Ziel keine Route mehr, wird die Route nicht mehr verwendet. Sie wird dann mit einer Metrik von unendlich noch einige Zeit lang (zwei Minuten) allen Nachbarn bekanntgegeben und anschließend gelöscht.

Man hat für IPv6 mit RIPng Version 1 [109] ein neues Protokoll mit geändertem UDP-Port standardisiert, anstatt RIP auf IPv6 zu erweitern, obwohl RIP genau wie die Multiprotokollerweiterungen für BGP ein Feld für die verwendete Adressfamilie besitzt. RIPng Version 1 ist bis auf die Kodierung der Routen – ohne Adressfamilie und somit ausschließlich für IPv6 geeignet – und die fehlende, da IPsec (IP Security [100]) überlassene, Authentisierung mit RIP Version 2 identisch.

RIP ist ein sehr einfaches Routingprotokoll. Durch die Beschränkung auf 15 Hops Durchmesser und den periodischen Austausch vollständiger Routingtabellen ist es nur für kleinere Netze geeignet. RIP konvergiert nach einer Routingänderung trotz Triggered Updates langsamer als die Link-State-Routingprotokolle OSPF und IS-IS. Durch die Beschränkung auf den Hop Count als Metrik ist eine Verkehrslenkung, um beispielsweise einen schmalbandigen und/oder teuren Backuplink erst bei Ausfall eines Hauptlinks zu nutzen, nicht möglich.

[12] oft auch einfach „Poison Reverse“

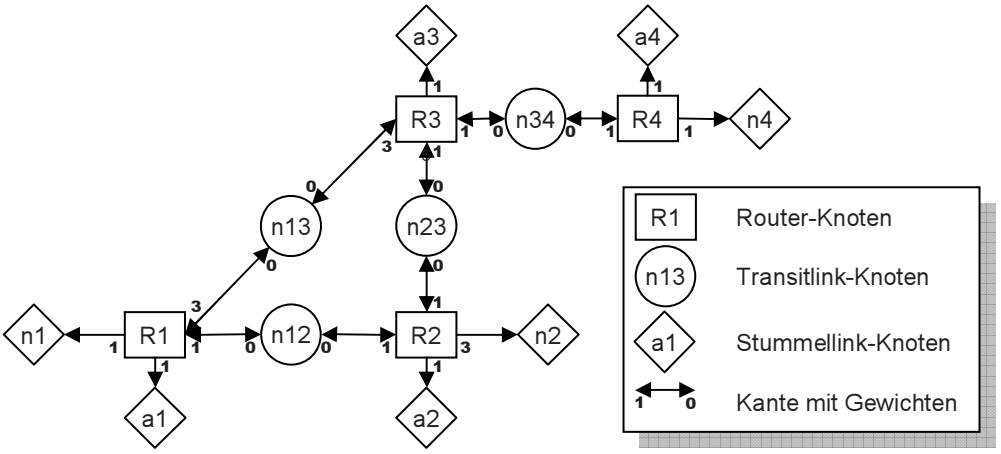


Abbildung 2.6 Resultierender OSPF-Graph (Version 2) zur Beispieltopologie aus Abbildung 2.5.

Wie später noch gezeigt wird, sind die langsamere Konvergenz und der einem Router nicht bekannte vollständige Pfad zum Ziel von Nachteil für das Dienstgütemanagement von mit RIP gerouteten Netzen, machen es jedoch nicht grundsätzlich unmöglich.

2.1.3 Link State Routing

2.1.3.1 OSPF – Open Shortest Path First

Das etwa zeitgleich zu RIP entstandene Link-State-Routingprotokoll OSPF besitzt die genannten Nachteile von RIP nicht, ist dafür jedoch deutlich komplexer. Aktuell sind OSPF Version 2 [123] für IPv4 und Version 3 [43] für IPv6 standardisiert. OSPF-Nachrichten werden direkt über IP übertragen. Allen Schnittstellen eines Routers wird ein dimensionsloses positives ganzzahliges Gewicht, die Kosten (Cost), zugeordnet. Als Routingmetrik dient die Distanz, d. h. die Summe aller Gewichte aller passierten Schnittstellen auf dem Pfad bis zum Ziel, wobei die beste Route jene mit der kürzesten Distanz zum Ziel ist. Die kürzesten Distanzen zu allen Zielen werden beim Link-State-Routing durch den Dijkstra-Algorithmus berechnet. Zur Berechnung unterhält jeder OSPF-Router einen gerichteten Graphen, in dem jeder Router und jeder Link des Netzes durch einen Knoten im Graphen repräsentiert ist. Punkt-zu-Punkt-Verbindungen zwischen zwei Routern können statt entweder als sogenannter Transitlink oder als Punkt-zu-Punkt-Link konfiguriert werden. In letzterem Fall sind die Knoten der beiden Router im Graphen direkt ohne einen dazwischenliegenden Link-Knoten miteinander verbunden. Kanten zwischen den Knoten im Graphen tragen die Gewichte, wie sie den Schnittstellen der Router zugeordnet wurden. Kanten von einem Link-Knoten zu einem Router-Knoten haben immer das Gewicht null.

Als Beispiel zeigt Abbildung 2.6 den Graphen, wie er bei Verwendung von OSPF Version 2 für das in Abbildung 2.5 dargestellte Autonome System A resultieren würde. Die Gewichte sind weitgehend gleich eins gewählt, was dem Hop Count als Metrik entspricht. Die Schnittstellen zu den Links n13 und n2 sind mit Gewicht 3 konfiguriert, so dass sie nur als Backuplink genutzt werden. Die „Schnittstellen“ der Link-Knoten tragen immer das Gewicht null. Auf einen Link-Knoten folgt immer ein Router-Knoten. Im Beispiel gilt dies auch umgekehrt, da keiner der Links als Punkt-zu-Punkt-Link sondern jeder als

Transitlink konfiguriert ist. Die Adressen a1 bis a4 werden durch Stummellinks (s. u.) repräsentiert, ebenso die drei Links n1, n2 und n3 zu den benachbarten Autonomen Systemen. Schließlich nehmen die Router der benachbarten Autonomen Systeme nicht am Intradomänenrouting des Autonomen Systems A teil. Stummellink-Knoten besitzen anders als Transitnetzwerk-Knoten keine ausgehende Kante im Graphen. Bei OSPF Version 3 sähe der Graph ähnlich aus, jedoch kennt Version 3 keine Stummellinks. OSPFv3 verwendet sogenannte Intragebietspräfixe. Sie können sowohl Router- als auch Link-Knoten zugeordnet werden. Jedes Präfix besitzt sein eigenes Gewicht.

Damit alle Router diesen Graphen aufbauen können, generiert jeder Router sogenannte LSAs (Link State Advertisement). Sie werden durch zuverlässiges Fluten allen anderen Routern bekannt gemacht. Das zuverlässige Fluten ist eine wichtige Voraussetzung für die korrekte Funktion des Link-State-Routings und wesentliche Ursache für dessen Komplexität. Jeder Router generiert für sich ein Router-LSA, das alle seine Schnittstellen mit ihrem Typ aufführt. Mögliche Schnittstellentypen sind u. a. Punkt-zu-Punkt-, Transit- und Stummellinks. Wie oben im Beispiel Abbildung 2.6 gezeigt, repräsentierten Stummellinks sowohl Links mit nur einem (teilnehmenden) Router (Netzwerkpräfixe n1, n2 und n4) wie auch Hostrouten (Adressen a1 bis a4). Die Link-LSAs (Network-LSA) können nicht von einer (in der Vermittlungsschicht) passiven Komponente wie einem Link selbst generiert werden. Diese Aufgabe übernimmt der für einen Link gewählte Designated Router. Die „Schnittstellen“ der Link-LSAs tragen keine Gewichte, die Kanten von Link- zu Routerknoten tragen daher immer das Gewicht null. Für Stummellinks gibt es kein entsprechendes LSA, ihre Knoten haben keine ausgehenden Kanten im Graphen.

Alle Router senden periodisch Hello-Nachrichten auf allen ihren Schnittstellen. Dies dient mehreren Zwecken: Router können so selbständig ihre Nachbarrouter erkennen und Nachbarschaftsbeziehungen, Adjazenzen genannt, mit ihnen etablieren. Über die Adjazenzen erfolgt das zuverlässige Fluten der LSAs. Über die Hello-Nachrichten wird zudem der Designated Router gewählt, der die Link-LSAs generiert. Schließlich wird mit ihnen der Ausfall von Routern und/oder Links erkannt. Ein Ausfall wird durch entsprechend aktualisierte und erneut geflutete LSAs allen Routern bekannt gemacht. Router berechnen daraufhin die besten Pfade neu und aktualisieren ihre Routing Information Base. Das Hello-Protokoll arbeitet soft state, während das Fluten der LSAs nur bei Änderungen erfolgt und mittels Bestätigungen abgesichert ist. Allerdings tragen LSAs ein Alter. Erreicht es ein festgelegtes Maximum (1 Stunde), wird das alte gelöscht (durch erneutes Fluten mit maximalem Alter) und ein neues LSA mit neuer Sequenznummer aber sonst gleichem Inhalt generiert.[13]

OSPF-Gebiete – zweistufige Hierarchie

Da der Aufwand für das Fluten der LSAs und die Neuberechnung der kürzesten Pfade mit der Netzwerkgröße überproportional[14] ansteigt, erlaubt OSPF die Untergliederung der gesamten Routingdomäne – OSPF spricht von einem Autonomen System – in mehrere Gebiete (Area). Gebiete sind über Gebietsgrenzrouter (Area Border Router) miteinander

[13] Bei z. B. einhunderttausend aktiven Routen bzw. LSAs bedeutet dies das Fluten eines LSAs alle 36 ms. Mit ein Grund, warum Interdomänenrouten nicht in IGPs wie OSPF eingespeist werden.

[14] je nach Implementierung der Randknotenmenge im Dijkstra-Algorithmus durch Markierung im Graphen, als Heap oder als Fibonacci-Heap mit $O(n^2)$, $O(m \log n)$ oder $O(n \log n + m)$ bei n Knoten und m Kanten im Graphen [132]

verbunden. Grenzrouter sind alle Router mit Schnittstellen in mindestens zwei unterschiedlichen Gebieten. Alle Grenzrouter grenzen per Definition an ein ausgezeichnetes Hauptgebiet (Backbone Area) an, so dass sich bei OSPF eine zweistufige Hierarchie ergibt. In jedem Gebiet läuft eine eigenständige Instanz des Algorithmus ab, das Fluten der LSAs bleibt auf das Innere eines Gebiets beschränkt. Ein Grenzrouter speist die aggregierte Routinginformation eines Gebiets als Zusammenfassungs-LSA (Summary-LSA) in alle anderen an sie angrenzenden Gebiete ein. Die dortigen Router lernen hierdurch die Routen zu Zielen außerhalb ihres eigenen Gebiets. Externe Routen zu Zielen außerhalb der gesamten Routingdomäne werden durch AS-Grenzrouter (AS Boundary Router) in Form von AS-externen-LSAs (AS-external-LSA) allen anderen Routern der Routingdomäne bekannt gemacht.

OSPF für IPv6

OSPF Version 3 für IPv6 unterscheidet sich in einigen Punkten von der bisher beschriebenen Arbeitsweise der Version 2. In Version 2 besitzen die IPv4-Adressen eine Doppelfunktion als eindeutige Identifikation eines Knotens im Graphen und als Adresse eines Netzwerkpräfixes, für das eine Route zu generieren ist. In Version 3 sind diese beiden Aufgaben getrennt worden. Die Präfixe werden stattdessen mit den neu eingeführten Intragebietspräfix-LSAs (Intra-Area-Prefix-LSA) separat bekanntgegeben und geflutet. Sie tragen die ID des Knotens im Graphen, dem die Präfixe zugeordnet sind, und ordnen jedem Präfix ein eigenes Gewicht zu. Jedem Knoten – Router wie Link – kann kein, ein oder können mehrere Präfixe zugeordnet sein. Sie ersetzen damit gleichzeitig die Stummellink-LSAs.

Neben den bisher genannten LSA-Typen stellen die opaken LSAs (Opaque LSA) [18] den Mechanismus des zuverlässigen Flutens versionierter Informationen auch für beliebige andere nicht notwendigerweise routingspezifische Zwecke zur Verfügung. Ein Beispiel für die Nutzung von opaken LSAs sind die Erweiterungen von OSPF zum Traffic Engineering OSPF-TE [98]. Die dadurch innerhalb eines Gebiets verbreiteten Zusatzinformationen wie die Bandbreite von Links ist z. B. auch im Rahmen des Dienstgütemanagements verwendbar und erspart eine Abfrage dieser Informationen über das Netzwerkmanagement.

2.1.3.2 IS-IS – Intermediate System to Intermediate System

IS-IS [96] entstand Ende der 1980er Jahre als Routingprotokoll für die OSI=Protokollfamilie, wurde jedoch bald auch für das Internet adaptiert. OSPF und IS-IS besitzen abgesehen vom zugrundeliegenden Prinzip des Link-State-Routings viele Gemeinsamkeiten wie z. B. die zweistufige Hierarchie zur Unterstützung sehr großer Netze. Einer der Unterschiede ist, dass IS-IS aufgrund seiner Herkunft nicht IP zur Übertragung seiner Nachrichten verwendet, sondern direkt auf der Sicherungsschicht aufsetzt. Mit IS-IS lässt sich zudem sowohl IPv4 [36] wie IPv6 [86] routen, während OSPF hierfür zwei unterschiedliche Protokolle OSPFv2 und OSPFv3 einsetzt. Allerdings befinden sich IS-IS und OSPFv3 in aktiver Weiterentwicklung durch die IETF. Insbesondere ist es mit der neuen Multitopologie-Erweiterung (M-ISIS, MT-OSPFv3) [143, 144] möglich, verschiedene Topologien z. B. für unterschiedliche Dienstgüteanforderungen, mit einer Instanz eines Routingprotokolls zu routen. Da jede Topologie eine andere Adressfamilie nutzen kann, erhält OSPFv3 hierdurch zugleich die Fähigkeit, wie IS-IS beides, IPv4 und IPv6, zu routen. Auch andere Erweiterungen erfolgen parallel an beiden Protokollen. IS-IS und OSPFv3 sind von ihrem Funktionsumfang und Einsatzspektrum daher als äquivalent anzusehen. Allerdings befindet sich IS-IS eher bei großen und OSPF eher bei kleineren Netzbetreibern im Einsatz. Auch bei großen Netzen bleibt die zweistufige Hierarchie in der Praxis ungenutzt [99].

2.1.4 Equal Cost Multipath Routing

Häufig existieren mehrere beste Pfade für ein Zielpräfix. Anstatt den Verkehr immer über denselben Pfad zu leiten, ist es zur besseren Ausnutzung der verfügbaren Ressourcen oft wünschenswert, alle zur Verfügung stehenden besten Pfade zu nutzen. Man spricht von Equal Cost Multipath Routing (ECMP). Unterschiedliche Pfade besitzen allerdings unterschiedliche Paketlaufzeiten. Verteilt man den Verkehr naiv paketweise im Round-Robin-Verfahren auf die verfügbaren Pfade, kommt es zu Paketvertauschungen innerhalb der einzelnen Datenströme des aufgeteilten Verkehrs. Diese Vertauschungen müssen wegen ihrer negativen Auswirkungen[15] auf die höheren Schichten jedoch grundsätzlich vermieden werden. Bei ECMP werden daher alle Pakete eines Datenstroms über denselben Pfad geroutet, der Verkehr wird also nicht mehr paketweise sondern datenstromweise auf die verfügbaren Pfade aufgeteilt. Bei ECMP wird hierzu ein Hashwert über die einen Datenstrom identifizierenden Felder Quell- und Zieladresse, bei IPv4 ggf. zusätzlich das Transportschichtprotokoll und dessen Ports, bei IPv6 das Flow Label, berechnet. Je mehr Felder berücksichtigt werden, desto feingranularer kann der Verkehr aufgeteilt werden. Ein bestimmter Wertebereich des Hashwerts wird schließlich einem der verfügbaren Next Hops zugeordnet.

Das Problem, dass das Hinzufügen oder Entfernen eines Next Hops, beispielsweise aufgrund eines Linksausfalls, den Next Hop für einen erheblichen Teil aller Datenströme ändert, kann durch Verwendung des HRW-Verfahrens (Highest Random Weight) [176] minimiert werden. Bei n Next Hops reduziert sich der Anteil der betroffenen Datenströme von 25 % bis 50 % auf $1/n$ [173]. Beim HRW-Verfahren wird der Next Hop in die Hashwertberechnung mit einbezogen und als Next Hop jener gewählt, der den größten Hashwert liefert [85]. Als Nachteil vergrößert sich durch die n-fache Hashwertberechnung der Aufwand für die Weiterleitung eines Pakets von $O(1)$ auf $O(n)$. Wird jedoch Zustand pro Datenstrom gehalten, was, wie später in Abschnitt 2.3.1 noch ersichtlich wird, bei Multicast meist ohnehin erforderlich ist, kann die Hashwertberechnung nur einmal ausgeführt und das Ergebnis zwischengespeichert werden. Somit stellt der Einsatz des HRW-Verfahrens zumindest bei Multicast keinen Performancenachteil dar.

Relaxed ECMP

Es erweist sich als praktisch äußert schwierig, beim Link-State-Routing die Gewichte so zu wählen, dass sich brauchbare Pfade mit gleichen Kosten ergeben, wie es für striktes ECMP nötig ist [183]. Man kann jedoch ohne Gefahr von Routingschleifen auch Pfade mit etwas schlechteren Kosten in das ECMP-Routing mit einbeziehen, sofern die Kosten vom alternativen Next Hop zum Ziel echt kleiner sind als die Kosten vom aktuellen Router aus [183, 184]. Anschaulich gesprochen muss der Verkehr mit jedem Hop dem Ziel echt näher kommen. Relaxed ECMP erfordert allerdings zusätzlich die Berechnung der kürzesten Pfade aus Sicht jedes benachbarten Routers. Da eine Neuberechnung nur nach einer Änderung in der Topologie nötig ist, stellt Relaxed ECMP bei stabilem Routing nur einen geringen Mehraufwand dar.

Insgesamt bleibt festzustellen, dass bei Einsatz von (Relaxed) ECMP das Routing deterministisch bleibt, da sich der Einsatz statistischer Verfahren wie paketweises Round-Robin

[15] Die Staukontrolle von TCP wertet jedes verspätete, d. h. von drei früheren Paketen überholte, Paket wie ein Paketverlust und halbiert bei Empfang des verspäteten Pakets sein Staukontrollfenster und damit die Senderate. Mit jedem weiteren verspäteten Paket halbieren sich Staukontrollfenster und Senderate erneut, bis nur noch ein Paket, das Minimum, pro Paketumlaufzeit (RTT) übertragen wird.

aufgrund der entstehenden Paketvertauschungen verbietet. Allerdings wird der Pfad durch mehr als die Felder des IP-Pakets bestimmt, insbesondere durch das verwendete Hashverfahren, welches implementierungsspezifisch ist. Die Kenntnis des exakten Pfads ist wichtig für die Bereitstellung von Dienstgüte, wie in Abschnitt 2.4.2 noch erläutert wird.

2.1.5 Netzwerkmanagement mit SNMP

Unter dem Begriff Netzwerkmanagement versteht man die Verwaltung und Überwachung eines Netzes. Das Netzwerkmanagement erfüllt eine Reihe von Aufgaben. Es erlaubt das Konfigurieren der Knoten im Netz. Grundsätzliche Betriebsparameter des Netzes werden festgelegt. Geeignete Einstellungen der Knoten am Netzrand können für eine Zugangskontrolle sorgen, beispielsweise, um die Betriebssicherheit des Netzes zu gewährleisten. Die Überwachung des Netzes erlaubt die Erkennung von Fehlverhalten oder Ausfällen und die Erfassung des Verkehrs im Netz. Ziel ist, die Leistungsfähigkeit eines Netzes dauerhaft sicherzustellen. Zusätzlich ist die Überwachung Voraussetzung für eine spätere Abrechnung des Verkehrs.

Das Rahmenwerk für das Netzwerkmanagement im Internet ist nach dem von ihm verwendeten Simple Network Management Protocol (SNMP) benannt. Aktuell ist die dritte Version des Rahmenwerks SNMPv3 [38], dass den Vorgänger SNMPv2 um ein starkes Sicherheitsmodell und ein Zugriffskontrollmodell ergänzt. Einen Überblick über die SNMP-Standards gibt [142]. Das SNMP-Rahmenwerk unterscheidet verschiedene Typen von Knoten, die beiden wichtigsten werden als Manager und Agent bezeichnet. Die Agenten laufen auf den zu verwaltenden Knoten und werden vom Manager aus kontrolliert. Manager und Agenten tauschen über das Protokoll SNMP Managementinformationen aus. Typischerweise existiert nur ein Manager (oder einige wenige Manager) und eine Vielzahl von Agenten. SNMP ist ein asymmetrisches Protokoll, bei dem nur Manager mit Agenten kommunizieren und weder Manager noch Agenten unter sich. SNMP verwendet UDP als Transportschichtprotokoll zum Austausch seiner Nachrichten, den SNMP-PDUs (Protocol Data Unit). Die SNMP-Einheit auf einem Knoten besteht aus der SNMP-Maschine, die das eigentliche Senden und Empfangen der SNMP-PDUs abwickelt und die Zugriffskontrolle durchführt, und mehreren Anwendungen. Die Anwendungen auf dem Agenten haben Zugriff auf den verwalteten Zustand, der in seiner Gesamtheit als Management Information Base (MIB) bezeichnet wird und dessen Struktur in sogenannten MIB-Modulen beschrieben ist. Die Beschreibung erfolgt mit Hilfe der Structure of Management Information (SMI), aktuell SMIv2 [113], einer angepassten Untermenge der Abstract Syntax Notation One (ASN.1) [97] aus der OSI-Welt. Es existiert für fast jedes Protokoll im Internet ein entsprechendes MIB-Modul. Die Anwendungen auf dem Agenten erlauben dem Manager einerseits lesenden und schreiben Zugriff auf die MIB. Andererseits kann sich der Manager informieren lassen, wenn ein gewisser Zustand in der MIB besteht oder sich eingestellt hat. Dies kann mit einer vom Manager unbestätigten SNMPv2-Trap-PDU oder mit einer bestätigten InformRequest-PDU geschehen. Welche Zustände hierbei aus der MIB mitgesendet werden, ist implementierungsspezifisch.

SNMPv2 hat sich wegen seines schwachen Sicherheitsmodells auf Basis von Communities (quasi Klartextpasswörter) nicht für die Konfiguration von Routern durchgesetzt. SNMPv3 hat diesen Mangel zwar beseitigt, jedoch hat sich mittlerweile die (proprietäre) Kommandozeilenschnittstelle (Command Line Interface, CLI) über Telnet [141] oder SSH (Secure Shell) [195] als alternative Managementschnittstelle von Routern etabliert und wird SNMP häufig vorgezogen, weil sie u. a. in der Regel mehr Funktionalität bietet [157].

2.2 Gruppenkommunikation

2.2.1 Systematik der Kommunikationsformen

Quellen : Empfänger	Form	Routing	Bemerkung
1 : 1	Unicast	Unicast	
m : 1	Concast	Unicast	m-facher Unicast
1 : 1 aus n	Anycast	Unicast	1 (bester Pfad) aus n möglichen
1 : n	Multicast	Multicast	Source Specific Multicast
1 : alle n	Broadcast	kein Routing	maximal Link Scope
m : n	Multipeer	Multicast	Any Source Multicast

Tabelle 2.1 Systematik der Kommunikationsformen und ihre Umsetzung durch das Routing im Internet.

Kommunikationsbeziehungen lassen sich anhand der Anzahl beteiligter Kommunikationspartner in sechs Formen gliedern (Tabelle 2.1). *Unicast* bezeichnet die Kommunikation zwischen genau einer Quelle und einem Empfänger. Bei *Concast* senden mehrere Quellen an genau einen Empfänger.[16] Umgekehrt sendet bei *Multicast* genau eine Quelle an mehrere Empfänger. *Multipeer* ist die Verallgemeinerung von Multicast auf mehrere Quellen, bei der jede der Quellen an dieselbe Gruppe von Empfängern sendet.

Bis auf Unicast stellen alle eine Form von Gruppenkommunikation dar. Allerdings werden im Internet auch Concast und Anycast durch das Unicastrouting umgesetzt. Concast entspricht m Unicastpfaden mit demselben Zielknoten. Bei Anycast wird einer Gruppe von Knoten dieselbe Anycastadresse zugeordnet. Sie wird vom Unicastrouting wie eine Unicastadresse behandelt. Das Unicastrouting wählt somit den Empfänger mit der besten Route, also den mit der kürzesten Distanz zur Quelle, aus. Obwohl die Anycastadresse semantisch eine Gruppenadresse darstellt, ist sie syntaktisch eine Unicastadresse und von dieser nicht zu unterscheiden. Broadcast wird im Internet nicht geroutet, sondern findet ausschließlich innerhalb eines Links Anwendung, z. B. für ARP. Für größere Reichweiten kann er aber durch Multicast nachgebildet werden. Einzig die beiden Formen Multicast und Multipeer nutzen ein spezielles Multicastrouting. Da Multicast einen Spezialfall von Multipeer mit exakt einer Quelle darstellt, erfolgt das Routing in beiden Fällen mit denselben Multicastroutingprotokollen. Der Begriff Multicast steht im Internet daher sowohl für Multicast- wie für Multipeerkommunikation. Zur genaueren Unterscheidung wird im Internet Multicast mit genau einer Quelle als *Source Specific Multicast*, kurz SSM, und Multicast mit mehreren Quellen, also Multipeer, als *Any Source Multicast*, kurz ASM, bezeichnet.

Gruppenkommunikationsbeziehungen lassen sich weiter klassifizieren nach *Offenheit* (offen vs. geschlossen), *Dynamik* (dynamisch vs. statisch), *Lebensdauer* (transient vs. permanent), *Bekanntheit* (anonym vs. bekannt) und *Heterogenität* (heterogen vs. homogen) [189]. Offen bedeutet, dass die Quelle selbst nicht auch Empfänger sein muss. Die Zusammensetzung einer dynamischen Gruppe kann sich ändern, es können also neue Gruppenmitglieder

[16] Teilweise wird unter Concast ein „inverser Multicast“ verstanden [37]. Allerdings ist im Gegensatz zum Duplizieren von Paketen das Zusammenfassen von Verkehr durch Knoten längs des Pfads nicht mehr eindeutig festgelegt sondern auf verschiedene Arten möglich.

hinzukommen oder im Laufe der Kommunikation wegfallen. Bei anonymen Gruppen sind der Quelle die Empfänger nicht (alle) bekannt. Heterogene Gruppen enthalten Empfänger mit unterschiedlichen Eigenschaften, wobei hier nur Eigenschaften von Interesse sind, die Auswirkungen auf die Kommunikation haben. Also z. B. die noch zur Verfügung stehende Bandbreite ihrer Anbindung oder die Fähigkeit, gewisse Datenströme überhaupt handhaben/darstellen zu können.

Gruppenkommunikation, wie sie von der Vermittlungsschicht des Internets bzw. dessen Multicastrouting bereitgestellt wird, ermöglicht ausschließlich offene, dynamische, transiente und anonyme Gruppen. Heterogene Gruppen sind möglich, jedoch unterstützt die Vermittlungsschicht die höheren Schichten nicht bei der Handhabung der Heterogenität. Es wird daher meist durch Mechanismen auf Transport- oder Anwendungsschicht für eine homogene Gruppe gesorgt. Permanente Gruppen werden im Internet als transiente Gruppen mit permanenter Gruppenkennung umgesetzt und damit genau wie transiente Gruppen geroutet. Die permanenten Gruppenkennungen sind bei der IANA (Internet Assigned Numbers Authority) abrufbar [90, 91]. Abweichende Gruppencharakteristika können darauf aufbauend mit Protokollen auf höheren Schichten bereitgestellt werden. Beispielsweise ermöglicht das Real-time Control Protocol (RTCP) [158] u. a. bekannte Gruppen. Heterogene Gruppen lassen sich durch mehrere homogene Multicastgruppen [112, 156] oder auch mittels aktiver und programmierbarer Netze umsetzen [117]. Nachfolgend sollen zuerst grundlegende Prinzipien des Multicastroutings erläutert und alternative Ansätze zur Gruppenkommunikation im Internet kurz vorgestellt werden. Danach wird auf Adressierung und Gruppenmanagement näher eingegangen, bevor abschließend die verschiedenen Multicastroutingprotokolle des Internets im Einzelnen vorgestellt werden.

2.2.2 Prinzipien des Multicastroutings

RPF – Reverse Path Forwarding

Mitte der 1980er Jahre wurde von Deering und Cheriton das Multicastservicemodell für das Internet vorgeschlagen [41, 50] (mittlerweile ersetzt durch [49, 35]). Das klassische von ihnen vorgeschlagene Servicemodell ist Any Source Multicast, damals noch als Host Groups bezeichnet. Bei ihm wird eine Gruppe allein durch die Zieladresse im IP-Paket, die Gruppenadresse, gekennzeichnet. Gruppenadressen unterscheiden sich syntaktisch von Unicast- bzw. Anycastadressen (s. Abschnitt 2.2.4). Gruppenkommunikation bedarf eines speziellen Multicastroutings. Es löst das Problem, dass aus der Gruppenadresse nicht direkt abgeleitet werden kann, wo sich die Empfänger befinden. Das von Deering vorgeschlagene Multicastrouting [52, 53] nutzt als Weiterleitungsprinzip *Reverse Path Forwarding* (RPF). RPF geht auf Dalal und Metcalf zurück [46] und wird auch heute noch von allen[17] für das Internet standardisierten Multicastroutingprotokollen verwendet. Das RPF-Prinzip besagt, dass Pakete nur dann weitergeleitet werden, wenn sie auf der Schnittstelle empfangen wurden, über die *umgekehrt* (Unicast-)Pakete mit der Quelle als Ziel weitergeleitet werden würden. Die Eingangsschnittstelle liegt also auf dem besten/kürzesten Pfad in umgekehrter Richtung zur Quelle.

Diese ausgezeichnete Schnittstelle wird als *Upstreamschnittstelle* bezeichnet in Bezug auf die Richtung stromaufwärts Richtung Quelle und entgegen der Flussrichtung der von ihr gesendeten Multicastpakete. Alle übrigen Schnittstellen sind Downstreamschnittstellen.

[17] bei MOSPF nur bei gebietsübergreifendem Multicast

Wie später noch dargestellt wird, kann die Upstreamschnittstelle auch in Richtung eines anderen ausgezeichneten Knotens zeigen (Rendezvousstelle) oder statisch konfiguriert sein (IGMP/MLD-Proxying, s. Abschnitt 2.2.6). Im Falle bidirektionaler Bäume (BIDIR-PIM, CBT, s. Abschnitt 2.3.2.3 bzw. Abschnitt 2.3.3.3) fließt Verkehr nicht nur stromabwärts sondern auch stromaufwärts. Im Allgemeinen kennzeichnet stromaufwärts daher die Richtung zur Wurzel des Verteilbaums, wie er durch die Wahl einer Upstreamschnittstelle auf allen Routern entsteht, und ist nicht notwendigerweise mit der tatsächlichen Flussrichtung der Pakete korreliert.

RPF vermeidet Weiterleitungsschleifen anders als das simple Fluten (Flooding). Beim Fluten[18] werden Pakete von *jeder* Eingangsschnittstelle akzeptiert und an alle anderen Schnittstellen weitergeleitet. Die Last durch kreisende Pakete wird beim Fluten dadurch begrenzt, dass Pakete mit Erreichen ihres maximalen Hop Counts verworfen werden (Feld Time to Live/Hop Count, s. Abbildung 2.3). RPF vermeidet allerdings nicht die Entstehung von Duplikaten. Hierzu ist die Wahl eines ausgezeichneten Routers je Link nötig, dem (designierten) Weiterleiter oder (Designated) Forwarder. Als Forwarder wird bei Verwendung von RPF jener Router gewählt, der vom Link aus gesehen auf dem kürzesten Pfad zur Quelle liegt. Bei mehreren gleich guten Pfaden dient die Routeradresse als Tie-Breaker. Wer Forwarder ist, folgt damit direkt aus dem Unicastrouting und ist jedem Router am Link bekannt. Die Protokolle der PIM-Familie (s. Abschnitt 2.3.2) sind unabhängig vom Typ des eingesetzten Unicastroutingprotokolls. Sie verlassen sich nicht ausschließlich auf das von einem anderen (Unicast-)Routingprotokoll berechnete Routing, sondern verwenden mit den Asserts einen reaktiven Mechanismus, mit dem ein Router beim ersten Auftreten von Duplikaten seinen Anspruch als alleiniger Forwarder durchsetzt.

Vom Broadcast zum Multicast – Gruppenmanagement und Pruning

RPF ist ein Broadcastverfahren. Deering hat dafür die Begriffe Reverse Path Flooding (RPF)[19] bzw. Reverse Path Broadcasting (RPB) geprägt, je nachdem, ob Duplikate durch die Wahl eines Forwarders verhindert werden (RPB) oder nicht (RPF). Wird darüber hinaus die Gruppenmitgliedschaft von Hosts mit einbezogen, spricht man von Multicast, da der Verteilbaum nun nicht mehr alle Knoten eines Netzes erreicht. Je nachdem, ob nur Links ohne Gruppenmitglieder oder ganze Teilbäume bis zu diesen Links ohne Gruppenmitglieder aus dem Verteilbaum entfernt werden, spricht Deering von Truncated Reverse Path Broadcasting (TRPB) oder Reverse Path Multicasting (RPM). Kenntnis über lokale an einem der eigenen Links angeschlossene Gruppenmitglieder erhalten die Router über ein Gruppenmanagementprotokoll (s. Abschnitt 2.2.5). Hosts geben hierüber den Routern ihre Gruppenmitgliedschaft bekannt, also ihren Wunsch, den Verkehr für bestimmte Gruppenadressen zu empfangen. Multicast erfordert von Routern darüber hinaus die Kenntnis, ob über einen Link abhängige Kindrouter im Verteilbaum versorgt werden. Betrachtet man einen Link, ist jeder Router, dessen Schnittstelle zu diesem Link seine Upstreamschnittstelle ist, ein abhängiger Router des Links. Gibt es abhängige Router, darf ein Forwarder das Weiterleiten auf den Link auch dann nicht einstellen, wenn der Link sonst keine Gruppenmitglieder hat. Wird Distance-Vector-Routing mit Split Horizon with Poisoned Reverse verwendet, kann der Forwarder die von ihm abhängigen Kindrouter an ihrem Poisoned Reverse erkennen [52, 53].

[18] von Dalal und Metcalf Hot Potato Forwarding genannt

[19] Reverse Path Flooding ist mit Reverse Path Forwarding identisch.

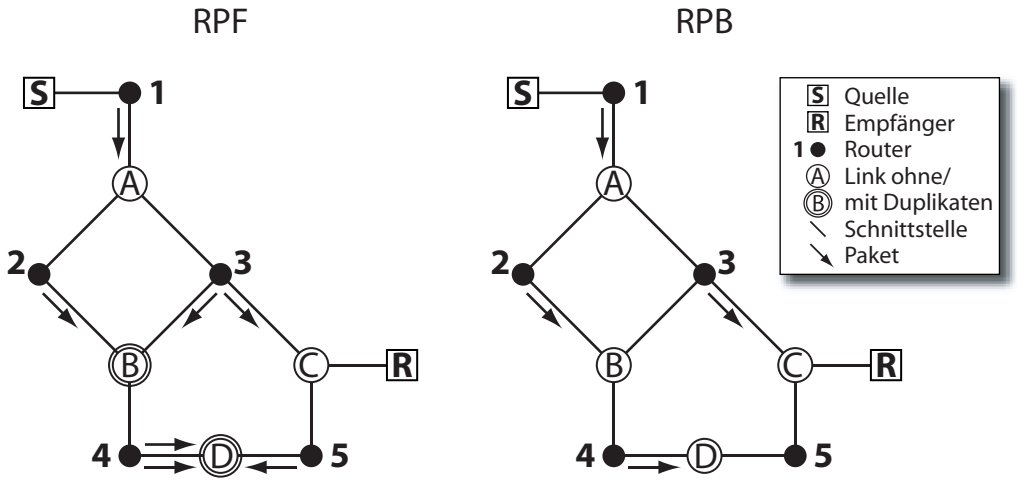


Abbildung 2.7 Die Broadcastverfahren RPF und RPB im Vergleich. RPF vermeidet Schleifen, RPB zusätzlich Duplikate.

Beim Truncated Reverse Path Broadcasting sind immer alle Router Teil des Verteilbaums, unabhängig davon, ob über sie noch Gruppenmitglieder erreicht werden oder nicht. Beim Reverse Path Multicasting wird darüber hinaus der Verkehr auch nicht in Netz*bereiche* ohne Empfänger weitergeleitet. Hierzu werden aus dem Verteilbaum ganze Teilbäume ohne Empfänger herausgeschnitten (Pruning). Beim Pruning informiert ein Router seinen Vaterrouter, dass er keinen Verkehr mehr von der genannten Quelle und Gruppe erhalten möchte. Erhält der Vaterrouter von allen seinen Kindroutern ein solches Prune und hat er auch sonst auf keinem seiner Links Gruppenmitglieder, kann er seinerseits eine Prune-Nachricht zu seinem Vaterrouter schicken usw. Umgekehrt kann sich ein Router wieder an den Verteilbaum „anpfropfen“ (Grafting), wenn er über das Gruppenmanagementprotokoll von neuen Empfängern für die Gruppe und Quelle erfährt oder seinerseits von einem Kindrouter eine Graft-Nachricht erhalten hat. Der durch das Pruning etablierte Zustand ist typischerweise zeitlich nur begrenzt gültig. Mit Ablauf der Gültigkeit wird der Verkehr erneut gebroadcastet, woraufhin erneut Pruning durchgeführt werden kann. Das RPM zugrunde liegende Multicastroutingprinzip wird entsprechend als Flood-and-Prune oder präziser Broadcast-and-Prune bezeichnet. Nach dem RPM-Verfahren arbeitete das erste für das Internet realisierte Multicastroutingprotokoll, das Distance Vector Multicast Routing Protocol (DVMRP, s. Abschnitt 2.3.3.1).

Abbildung 2.7 veranschaulicht die Broadcastverfahren RPF und RPB. Das dargestellte Netz besteht aus fünf Routern 1 bis 5, die über vier Links A, B, C und D miteinander verbunden sind. Die Quelle S ist direkt mit Router 1 verbunden, der einzige Empfänger R befindet sich auf Link C. Leitet ein Router auf einem seiner Downstreamschnittstellen ein Paket der Quelle weiter, ist dies mit einem Pfeil gekennzeichnet. Als Routingmetrik sei der Hop Count angenommen. Gemäß RPF-Prinzip leiten die beiden Router 2 und 3 auf Link B weiter, und Router 4 und 5 auf Link D. Bei RPF kommt es so zu Duplikaten auf diesen Links. Da Router 4 beide Pakete von Link B weiterleitet, kommt es auf Link D sogar zu insgesamt drei identischen Paketen. RPB wählt zur Vermeidung der Duplikate einen Forwarder pro Link. Der Forwarder für Link B sei Router 2, weil die Adresse der Schnittstelle von Router 2 zu Link B numerisch größer als die von Router 3 sei und die größere Adresse als Tie-Breaker gewinne. Entsprechendes gelte für Router 4 und 5 an Link

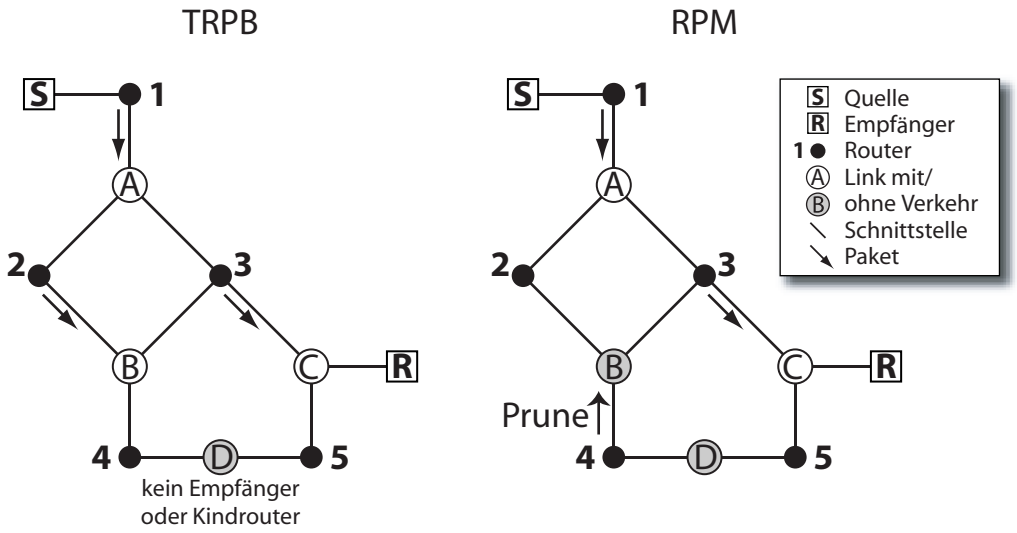


Abbildung 2.8 Die Multicastverfahren TRPB und RPM im Vergleich. TRPB leitet nicht auf Links ohne Empfänger oder abhängige Kindrouter weiter, RPM mittels Pruning auch nicht auf den Pfad zu solchen Links.

D, wo Router 4 zum Forwarder bestimmt wird. Als Konsequenz erhält bei RPB jeder Link im Netz jedes Paket der Quelle genau einmal.

Anders als RPF und RPB leiten die Multicastverfahren TRPB und RPM den Verkehr nicht mehr auf alle Links weiter, wie Abbildung 2.8 verdeutlicht. TRPB unterscheidet sich von RPB dadurch, dass die Pakete der Quelle den Link D nicht mehr erreichen. Auf diesem Link befindet sich weder ein Empfänger noch wird über den Link ein abhängiger Kindrouter erreicht. Schließlich ist sowohl die Schnittstelle von Router 4 als auch die von Router 5 zu Link D gemäß RPF-Prinzip eine Downstreamschnittstelle. RPM vermeidet darüber hinaus Verkehr auf dem Pfad zu Link D. Router 4 als Forwarder des Links D entfernt sich dazu mittels einer stromaufwärts in Richtung der Quelle S gesendeten Pruning-Nachricht selbst vom Verteilbaum. Da er der einzige Kindrouter an Link B ist und sich dort auch sonst kein Empfänger befindet, kann der Forwarder von Link B Router 2 das Weiterleiten auf den Link einstellen. Bei RPM wird das Weiterleiten somit auf die zum Erreichen des Empfängers R tatsächlich benötigten Links A und C eingeschränkt.

Rendezvousstellen – explizites Join

Beim Broadcast-and-Prune sorgt der periodische Broadcast für das Bekanntmachen einer Quelle im gesamten Netz. Wird der Verkehr nicht benötigt, wird er gepruned. Den umgekehrten Weg geht das einige Jahre später entwickelte Rendezvousstellen-basierte Multicastrouting [13, 51]. Hier sendet eine Quelle ihren Verkehr an einen ausgezeichneten Router, die *Rendezvousstelle*. Sie ist allen Routern im Netz bekannt. Gibt es Mitglieder für eine Gruppe, angezeigt durch das Gruppenmanagementprotokoll, sendet ein Router eine *Join*-Nachricht stromaufwärts, jedoch nicht in Richtung der Quelle – die ist einem Router schließlich nicht bekannt – sondern in Richtung der Rendezvousstelle. Der Forwarder des Upstreamlinks sendet seinerseits, ausgelöst durch die empfangene Join-Nachricht, eine neue Join-Nachricht weiter stromaufwärts in Richtung Rendezvousstelle. Dies setzt sich Hop-by-Hop von Forwarder zu Forwarder entlang der Links bis zur Rendezvousstelle fort. Jede

Join-Nachricht aktiviert im Forwarder die Weiterleitung des Verkehrs für die in der Join-Nachricht genannte Gruppe, so dass der Verkehr aller Quellen von der Rendezvousstelle über die Forwarder bis zu allen Empfängern fließt. Rendezvousstellen-basiertes Multicastrouting etabliert einen Verteilbaum mit der Rendezvousstelle als Wurzel anstatt einen Verteilbaum je Quelle. Man spricht vom *geteilten* Rendezvousstellenbaum, da mehrere Quellen denselben einen Verteilbaum der Rendezvousstelle gemeinsam nutzen. Tritt ein Empfänger aus der Gruppe aus, wird der Empfang per Prune-Nachricht wieder beendet.

Das Senden der Quelle an die Rendezvousstelle kann per Unicast oder alternativ per Multicast erfolgen. Bei Unicast werden die Multicastpakete der Quelle vom ersten Router nach der Quelle zur Rendezvousstelle getunnelt, d. h. in Unicastpakete gekapselt gesendet. Von der Rendezvousstelle werden sie wieder entkapselt und per Multicast über den Rendezvousstellenbaum zu den Empfängern weitergeleitet. Alternativ kann der etablierte Rendezvousstellenbaum nicht stromabwärts sondern auch in entgegengesetzter Richtung stromaufwärts zur Multicastweiterleitung verwendet werden. Der Rendezvousstellenbaum wird zum *bidirektionalen* Verteilbaum. Dies ist jedoch nur für Quellen auf dem Verteilbaum möglich, also nur für Quellen, die sich auf Links mit Gruppenmitgliedern befinden. Pakete von Quellen in anderen Netzbereichen werden entweder per Unicast zur Rendezvousstelle gesendet. Oder es wird ein das gesamte Netz umfassender Baum, ein *Spannbaum*, etabliert, der den Verkehr beliebiger Quellen bis zur Rendezvousstelle weiterleitet. Jeder Verteilbaum für eine Gruppe ist dann ein Teilbaum dieses Spannbaums. Es wäre als dritte Alternative denkbar, den Verteilbaum bzw. neue Pfade im Verteilbaum nicht nur durch den Beitritt von Gruppenmitgliedern sondern auch beim beginnenden Senden einer Quellen zu etablieren. Ein entsprechendes Multicastroutingprotokoll wurde jedoch nie entwickelt.

Das mit dem Rendezvousstellen-basierten Multicastrouting eingeführte explizite Join eignet sich ebenfalls zum Routing von Source Specific Multicast. Anders als bei Any Source Multicast ist die Quelle bei SSM prinzipiell bekannt, eine Rendezvousstelle zum Zusammenfinden von Quellen und Empfängern wird nicht benötigt. Die Join-Nachrichten werden bei SSM stattdessen einfach in Richtung der bereits bekannten Quelladresse gesendet. Je Quelle wird ein eigener *quellenspezifischer* Verteilbaum etabliert. Er gleicht dem durch das Broadcast-and-Prune-Prinzip aufgebauten.

2.2.3 Alternative Multicastverfahren

Die beschriebenen Verfahren zum Routing von IP-Multicast sind zwar skalierbar im Sinne, dass sie sehr große Gruppen mit sehr vielen Empfängern unterstützen, sind jedoch bei der Skalierbarkeit im Sinne von sehr vielen gleichzeitig aktiven Gruppen beschränkt. Es wurden daher alternative Multicastverfahren wie Xcast [30] oder MSC [60] vorgeschlagen, die unter dem Begriff Explicit Multicast zusammengefasst werden (einen Überblick gibt [62]). Sie gehen auf eine Idee von Aguilar [4] zurück, die Unicastadressen sämtlicher Gruppenmitglieder im Paket mitzuführen. Source Routed Multicast [55] führt darüber hinaus noch alle Verzweigungsrouter im Multicastpaket mit. Der sonst beim Aufbau von Verteilbäumen in den Routern etablierte Zustand wird bei diesen Verfahren ersetzt durch im Paket selbst mitgeführten Zustand. Größter Nachteil von Explicit und Source Routed Multicast ist die weniger effiziente Ausnutzung der verfügbaren Übertragungsbandbreite durch den im IP-Paket mitgeführten Overhead. Explicit Multicast hat darüber hinaus einen erhöhten Weiterleitungsaufwand, der proportional zur Anzahl an mitgeführten Empfängeradressen wächst.

Diese Nachteile besitzt Application Layer Multicast wie z. B. NARADA [42] oder NICE [14] nicht. Hier wird das Multicastrouting auf die Anwendungsschicht in die Endsysteme verlagert. Die Kommunikation zwischen Endsystemen erfolgt ausschließlich über Unicast. Application Layer Multicast verlagert den zum Multicastrouting nötigen Zustand also aus dem Netz aus den Routern in die Endsysteme. Die Bandbreiteneffizienz ist besser als bei Explicit Multicast, jedoch immer noch schlechter als bei nativem IP-Multicast, da die Pfade zwischen Quelle und Empfängern insgesamt länger sind. Zum Teil werden Pakete über gewisse Links mehrfach übertragen. Problematischer als die Effizienz ist die Gruppendynamik. Sie sorgt für ein häufiges Umstrukturierungen des Verteilbaums und Umleiten des Verkehrs weg von nicht mehr verfügbaren Endsystemen bzw. Gruppenmitgliedern. Die Folge sind temporäre Ausfälle in der Kommunikation und schwankende Ende-zu-Ende-Verzögerungen. Mit Overlay Multicast wie OMNI [15] oder TOMA [105] wurde daher vorgeschlagen, neben Endsystemen zusätzlich Proxys im Netzinneren einzusetzen. Diese dauerhaft verfügbaren Knoten verbessern nicht nur die Stabilität des Overlays bzw. Verteilbaums, sondern auch die Effizienz aufgrund ihrer günstigen Lage im Netzinneren. Erkauft wird dies durch den mit den Proxys erneut im Netzinneren etablierten Zustand und die Notwendigkeit, überhaupt geeignet platzierte Proxys zur Verfügung stellen zu müssen. Auf die alternativen Multicastverfahren wird im Rest des Kapitels nicht weiter eingegangen.

2.2.4 Multicastadressen

Eine Gruppe von Empfängern wird durch ihre *Gruppenadresse* oder auch *Multicastadresse* identifiziert. Multicastadressen werden aus den Präfixen 224/4 (IPv4) bzw. ff00::/8 (IPv6) vergeben, um sie von Unicastadressen syntaktisch zu unterscheiden. Um die nur für ASM notwendige internetweite Eindeutigkeit einer Multicastadresse sicherzustellen, wurden im Rahmen der Internet Multicast Address Allocation Architecture [172] mit Multicast Address-Set Claim (MASC) [103, 147] ein Protokoll zur domänenübergreifenden Adressallokation entwickelt. Multicastadressen aus dem so allozierten Bereich werden innerhalb einer Domäne mit dem Multicast Address Dynamic Client Allocation Protocol (MADCAP) [77] den Hosts auf deren Anfrage hin zugewiesen. MASC arbeitet wie domänenübergreifend üblich über TCP, während MADCAP ein UDP-basiertes Client-Server-Protokoll ist, dass sich aus dem Dynamic Host Configuration Protocol (DHCPv4) [56] heraus entwickelt hat. Allerdings haben sich diese Protokolle aufgrund ihrer Komplexität nicht durchsetzen können [119].

Eine Alternative für die internetweite Allokation von IPv4-Multicastadressen stellt die GLOP-Adressierung dar, bei der das Präfix 233/8 mit der 16 bit langen AS-Nummer zu einem global (internetweit) eindeutigen /24-Präfix ergänzt wird [119]. Einem Autonomen System stehen damit 256 Gruppenkennungen (die frei wählbaren letzten 8 bit) bzw. 256 Multicastadressen exklusiv zur Verfügung. Da AS-Nummern jedoch auf 32 bit erweitert werden sollen, wurde vorgeschlagen, die unicastpräfixbasierten Multicastadressen, wie sie für IPv6 schon standardisiert sind, auch für IPv4 zu übernehmen [174]. Einem Inhaber des Unicastadresspräfixes der Länge n stehen damit 2^{32-8-n} Multicastadressen zur Verfügung. Inhaber von /24-Präfixen erhalten nach diesem Schema genau eine Multicastadresse, Inhaber längerer Präfixe keine mehr. Bei den 128 bit langen IPv6-Adressen tritt dieses Problem nicht auf. Hier können Netzwerkpräfixe bis zu einer Länge von 64 bit eingebettet werden [73].

Die einem Netzbetreiber exklusiv zur Verfügung stehenden Multicastadressen müssen in einem zweiten Schritt den Quellen für die Verwendung zugewiesen werden, um eine

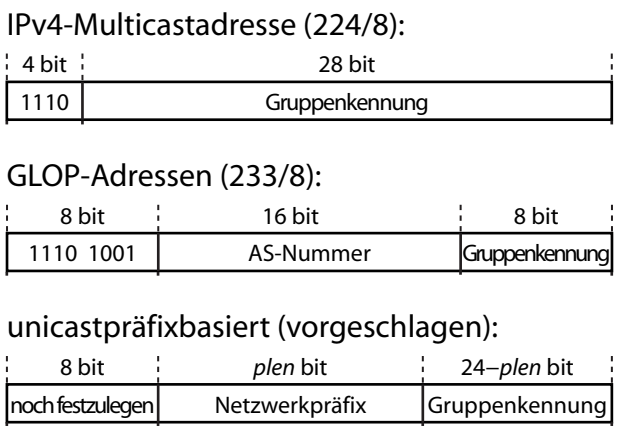


Abbildung 2.9 IPv4-Multicastadressen: oben allgemein [49], Mitte die auf AS-Nummern basierenden GLOP-Adressen [119], unten die vorgeschlagenen unicastpräfixbasierten Multicastadressen [174].

unbeabsichtigte Mehrfachnutzung einer Multicastadresse auszuschließen. Hierfür könnte zwar MADCAP verwendet werden, es sind jedoch kaum Implementierungen bekannt, so dass es praktisch nicht eingesetzt wird [155]. Erweiterungen für DHCPv6 und die zustandslose Autokonfiguration von IPv6 wurden vorgeschlagen [58, 59], aber nicht weiter verfolgt. Die Zuweisung von Multicastadressen erfolgt somit in der Regel händisch. Einzig für linklokale IPv6-Multicastadressen kann statt des Netzwerkpräfixes die Schnittstellenidentifikation eingebettet werden [133]. Da deren Eindeutigkeit auf dem Link bereits bei der zustandslosen Autokonfiguration sichergestellt wurde, ist kein weiteres dynamisches Verfahren nötig, um die doppelte Verwendung dieser Multicastadressen zu verhindern.

Source Specific Multicast baut auf der in den 1990er Jahren von verschiedenen Autoren [84, 135] vorgeschlagenen Idee auf, die Multicastgruppen quellenspezifisch zu machen. Bei SSM kennzeichnet daher erst das als *Kanal* (Channel) bezeichnete Tupel aus Quelladresse und Multicastadresse eine Gruppe von Empfängern eindeutig. Nur Verkehr dieser einen Quelle wird über den für den Multicastkanal aufgebauten Verteilbaum weitergeleitet. Zur Differenzierung zwischen ASM und SSM sind für SSM in IPv4 das Präfix 232/8 und in IPv6 die Präfixe ff3x::/32 (x = Multicast Scope) reserviert worden [82]. Da bei SSM die internetweit eindeutige Adresse der Quelle mit in die Multicastkanalkennung eingeht, muss lediglich die Quelle sicherstellen, keine Gruppenkennung doppelt zu verwenden.

Ein Beschränken der Reichweite bzw. des Gültigkeitsbereichs (Scope) von Multicastpaketen fand ursprünglich über auf den Grenzroutern eines Bereichs konfigurierte Schwellenwerte für die im IP-Paket enthaltene TTL (Time to Live) statt [41]. Zur Auswahl der gewünschten Reichweite versah die Quelle ihre IP-Pakete mit einem hinreichend weit oberhalb dieser Schwelle gelegenen initialen Wert für die TTL. Dieses TTL-Scoping wurde später durch die Festlegung von bestimmten Adresspräfixen im Bereich 239/8 abgelöst, die die Reichweiten Site-lokal und organisationslokal kodieren [118].[20] Andere Reichweiten sind nicht definiert, jedoch werden im Präfix 224.0.0/24 ausschließlich Multicastadressen für Kontrollverkehr mit linklokaler Reichweite (z. B. für Routing oder Gruppenmanagement) vergeben. Alle

[20] 239.255/16 für Site-lokal und 239.192/14 (IANA nennt abweichend 239.192.0.0–239.251.255.255) für organisationslokal

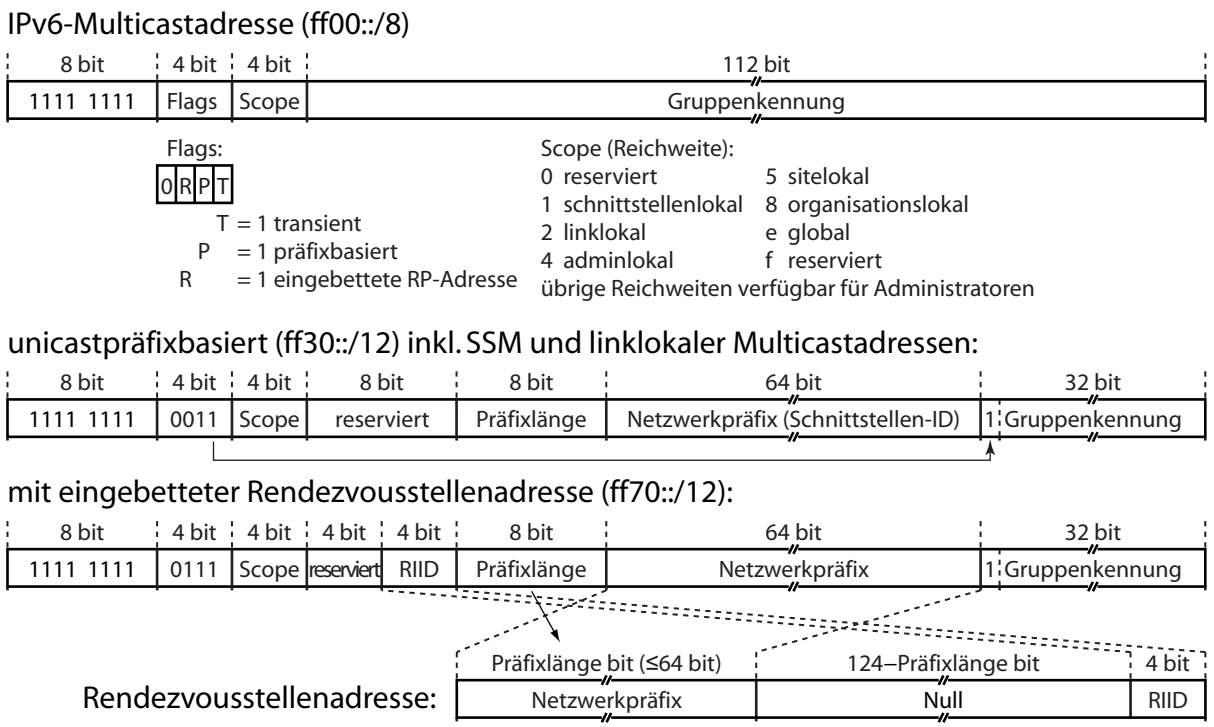


Abbildung 2.10 IPv6-Multicastadressen (nach [81]).

übrigen IPv4-Präfixe haben globale Reichweite, insbesondere auch die SSM-Adressen. Bei IPv6 wurde die Reichweite bereits beim Design berücksichtigt und für sie ein vier Bit langes Feld in die IPv6-Multicastadresse integriert [81]. Jede Reichweite besitzt damit ihr eigenes Adresspräfix, was erlaubt, anders als bei IPv4, auch Reichweiten für SSM anzugeben.

Abbildung 2.9 zeigt den Aufbau der IPv4-Multicastadressen noch einmal grafisch. GLOP-Adressen stellen einem Netzbetreiber über die frei wählbare 8 bit lange Gruppenkennung 256 Multicastadressen, unicastpräfixbasierte Multicastadressen je nach Länge *plen* des eingebetteten Netzwerkpräfixes mehr oder weniger Multicastadressen zur Verfügung. Abbildung 2.10 zeigt den Aufbau von IPv6-Multicastadressen. Während bei IPv4 die Art und Reichweite einer Adresse allein über das Präfix festgelegt wird, besitzt IPv6 hierfür die jeweils 4 bit langen Felder Flags und Scope. Von den 16 möglichen Reichweiten sind derzeit sechs vergeben, der kleinste und größte Wert ist reserviert. Die übrigen stehen einem Administrator bei Bedarf zur Definition weiterer Reichweiten mittels entsprechender Routerkonfiguration zur Verfügung. Von den vier Flags sind bisher drei Flags definiert, das vierte wird immer Null gesetzt. Das T-Flag bestimmt, ob es sich um eine transiente oder permanente Multicastadresse handelt. Ein gesetztes P-Flag zeigt an, dass die Eindeutigkeit der Multicastadresse durch ein eingebettetes (eindeutiges) Netzwerkpräfix hergestellt wurde [73]. Die Reichweite der Multicastadresse darf dabei nicht größer sein als der Gültigkeitsbereich des eingebetteten Netzwerkpräfixes, damit die Eindeutigkeit der Multicastadresse innerhalb ihrer Reichweite sichergestellt bleibt. unicastpräfixbasierte Multicastadressen sind immer transient. Das T-Flag ist bei gesetztem P-Flag daher immer gesetzt, so dass sich als Präfix ff30::/12 ergibt. Die 32 bit lange Gruppenkennung kann

frei gewählt werden, wobei allerdings das höchstwertige Bit dem Wert des T-Flags entsprechen muss [72]. Für linklokale Multicastadressen kann statt eines Netzwerkpräfixes die Schnittstellenidentifikation eingebettet werden, was durch die spezielle Präfixlänge 255 angezeigt wird [133]. Da die reservierten 8 bit in der Multicastadresse Null gesetzt werden, folgt daraus das Präfix ff32:ff::/32. Für SSM-Adressen wurde festgelegt, dass das Netzwerkpräfix ::/0 eingebettet wird. Für SSM sind entsprechend die 16 Präfixe ff3x::/32 mit x=0. . .f reserviert [82].

Das R-Flag schließlich zeigt an, dass aus dem eingebetteten Präfix und der 4 bit langen Schnittstellenidentifikation der Rendezvousstelle (Rendezvous Point Interface Identifier, RIID) zugleich die für das Routing der Multicastadresse benötigte Rendezvousstellenadresse abgeleitet werden kann [154]. Diese so genannten Embedded-RP-Adressen lösen das Problem der internetweit skalierbaren Abbildung von Multicastadressen auf Rendezvousstellenadressen. Ein gesetztes R-Flag impliziert ein gesetztes P-Flag und damit ein gesetztes T-Flag, das Präfix lautet entsprechend ff70::/12. Die eingebettete RP-Adresse wird dann wie folgt aus der Multicastadresse abgeleitet: Das Netzwerkpräfix bildet die ersten 64 bit. Wenn plen weniger als 64 relevante Bit angibt, wird der Rest mit Nullen auf 64 bit aufgefüllt. Die RIID wird in die niedrigstwertigen 4 bit der letzten 64 bit geschrieben und die übrigen 60 bit werden null gesetzt. Die RIID Null darf normalerweise nicht verwendet werden, da eine Schnittstellenidentifikation nur aus Nullen die sogenannte Subnet-Router-Anycastadresse kennzeichnet [81]. Mit ihr werden alle Router auf einem Link konfiguriert. Die eingebettete RP-Adresse würde dann nicht mehr einen sondern mehrere RPs adressieren. Das einzige Rendezvousstellen-basierte Multicastroutingprotokoll, das derzeit Embedded-RP-Adressen, verwendet, ist das in Abschnitt 2.3.2.1 beschriebene PIM-SM.

2.2.5 Gruppenmanagement mit IGMP und MLD

Router müssen wissen, welche Hosts in welcher Gruppe Mitglied sind, um für sie einen geeigneten Weiterleitungspfad einrichten zu können. Diesem Zweck dient das Gruppenmanagementprotokoll. Es erlaubt den Routern, Gruppenmitglieder auf ihren direkt angeschlossenen Links zu erkennen. Auf einem Router läuft je Link eine eigenständige Instanz des Gruppenmanagementprotokolls. Für das Internet standardisiert sind für IPv4 das Internet Group Management Protocol (IGMP), aktuell in Version 3 [35], und für IPv6 die Multicast Listener Discovery (MLD), aktuell Version 2 [181].

IGMP ist wie ARP ein eigenständiges Protokoll der Vermittlungsschicht, während MLD genau wie die Nachbarerkennung von IPv6 ein Unterprotokoll von ICMPv6 ist. Sowohl IGMP wie MLD arbeiten nach dem Soft-State-Prinzip und asymmetrisch. Router senden ausschließlich – außer sie sind gleichzeitig auch Gruppenmitglied – *Anfragen* (Query), Hosts senden ausschließlich *Berichte* (Report). Jeder Router auf einem Link bestimmt aus den empfangenen Berichten der Hosts den auf diesen Link weiterzuleitenden Multicastverkehr. Sieht ein Router eine Anfrage eines anderen Routers mit einer niedrigeren Adresse, unterdrückt er seine eigenen Anfragen. Hierdurch wird der ausgezeichnete Router mit der niedrigsten Schnittstellenadresse zum alleinigen Anfrager, dem *Querier*, für den Link. Der Querier muss nicht mit dem Forwarder für eine Gruppe und Quelle identisch sein. Vielmehr existiert genau ein Querier je Link, während der Forwarder typischerweise der gemäß RPF-Prinzip auf dem kürzesten Pfad zur Quelle oder Rendezvousstelle der Gruppe liegende Router ist, welcher je nach Gruppe und Quelle variiert. Alle Router, auch

Nicht-Querier, müssen daher die Berichte der Hosts auswerten. Hosts werten Berichte von anderen Hosts nicht aus. Im Falle von Snooping Switches in der Sicherungsschicht empfangen Hosts von anderen Hosts gesendeten Berichte auch gar nicht, da Berichte per linklokalem Multicast[21] nur an die Router eines Links gesendet werden [5].

Filter

Ein *Filter* bestimmt, von welcher Quelle in einer (ASM-)Gruppe der Empfang gewünscht ist und von welcher nicht. Je Gruppe existiert genau ein Filter. Im Filtermodus *Einbinden* wird der Empfang auf bestimmte Quellen eingeschränkt, im Filtermodus *Ausschließen* werden alle Quellen einer Gruppe empfangen und nur bestimmte Quellen ausgeschlossen. Der Filter auf einem Host fasst dabei alle Filter der einzelnen auf ihm laufenden Multicastanwendungen zusammen. Anwendungen geben ihren Filter über die Multicast-Socket-API [171] bekannt. Nach dem Zusammenfassen teilen Hosts den resultierenden Filter über das Gruppenmanagementprotokoll per Bericht den Routern auf ihrem Link mit. Diese wiederum fassen ihrerseits alle empfangenen Filter zu einem Filter für den Link (und die Gruppe) zusammen. Zusammenfassen bedeutet, dass aus allen Einbinden-Filtern ein einzelner Einbinden-Filter mit der Vereinigungsmenge aller Mengen von Quellen und aus allen Ausschließen-Filtern ein einzelner Ausschließen-Filter mit der Schnittmenge aller Mengen von Quellen gebildet wird. Die Menge des Einbinden-Filters wird anschließend von der des Ausschließen-Filters abgezogen. Das Ergebnis bildet die Menge des Ausschließen-Filters für den gesamten Link. Nur wenn kein Host bzw. keine Anwendung auf einem Host den Filter Ausschließen meldet, ist der Filter für den gesamten Link für diese Gruppe Einbinden. Die Menge der Quellen ist in diesem Fall die Vereinigung aller Mengen von Quellen aller (Einbinden-)Filter. Anschaulich gesprochen wird Verkehr jeder Quelle auf den Link weitergeleitet, deren Empfang zumindest von einem Host verlangt wird. Der Einbinden-Filter hat eine Doppelfunktion für ASM und SSM, da ein Tupel aus Quelladresse und Gruppenadresse auch einen Multicastkanal von SSM spezifizieren kann [83].

Ändern sich die Gruppenmitgliedschaft eines Hosts und/oder die von ihm gewünschten Quellen, sendet er seinen *Änderungsbericht*. Er sendet diesen in kurzen zufälligen Abständen mehrfach, um einem möglichen Verlust seiner Berichte zu begegnen. Um auch bei Verlust aller Änderungsberichte die Gruppenmitgliedschaft zuverlässig zu erkennen, fordert der Querier periodisch, typisch alle paar Minuten, die Hosts auf, ihre augenblickliche Gruppenmitgliedschaft mit einem *Istzustandsbericht* bekanntzugeben. Dies sorgt gleichzeitig dafür, dass der Ausfall eines Gruppenmitglieds zu keiner dauerhaften Weiterleitung unnötigen Multicastverkehrs auf den Link führt, weil dieser keinen abschließenden Änderungsbericht zum Verlassen der Gruppe gesendet hat.

Der Querier

Nachdem ein Host per Änderungsbericht Desinteresse für eine Gruppe oder für bestimmte Quellen in einer Gruppe bekundet hat, ist der Querier dafür zuständig, sofort nachzufragen, ob kein anderer Host auf dem Link Interesse an der Gruppe bzw. den genannten Quellen hat. Diese Anfragen geben eine kurze maximale Antwortzeit von typisch einer Sekunde vor und werden vom Querier genau wie Änderungsberichte zwecks Robustheit gegen Nachrichtenverlust in kurzem zeitlichen Abstand wiederholt. Mit der kurzen maximalen

[21] Multicastadresse „All IGMPv3/MLDv2-capable Multicast Routers“ 224.0.0.22 bzw. ff02::16

Antwortzeit wird erreicht, dass der von keinem Host mehr benötigte Verkehr möglichst bald nicht mehr auf den Link weitergeleitet wird (Fast Leave).

Es werden drei verschiedene Anfragen unterschieden. Generelle Anfragen nach beliebigen Gruppenmitgliedschaften werden per linklokalem Broadcast[22] an alle Knoten gerichtet. Gruppenspezifische Anfragen, die nach Mitgliedern für eine bestimmte Gruppe fragen, werden per Multicast an die Gruppenadresse gestellt genau wie die Gruppen-und-Quellen-spezifischen Anfragen. Mit ihnen ermittelt der Querier, ob in einer Gruppe noch Interesse für bestimmte Quellen besteht. Die generellen Anfragen werden vom Querier wie bereits erwähnt periodisch generiert, während die beiden anderen Arten ausschließlich aufgrund eines erhaltenen Änderungsberichts gesendet werden. Der Querier gibt bei einer Anfrage die maximale Antwortzeit vor, in der er die Istzustandsberichte von den Hosts zurück erwartet und innerhalb der die Hosts ihre Berichte zufällig verzögern. Die zufällige Verzögerung verhindert, dass es durch zu viele gleichzeitige Berichte zu Kollisionen auf dem Link und dadurch zu verlorenen Berichten kommt.

Streng genommen sind nur die generellen Anfragen nötig, da es in IGMPv3 und MLDv2 im Gegensatz zu älteren Versionen der Protokolle keine Unterdrückung von Berichten (Host Suppression) mehr gibt. Host Suppression bedeutet, dass ein Host seinen eigenen Bericht unterdrückt, wenn bereits ein entsprechender Bericht von einem anderen Host gesendet wurde. Ohne Host Suppression ist ein Router in der Lage, jederzeit jeden einzelnen Empfänger zu kennen. Ohne die zusätzlichen spezifischen Anfragen wäre ein Router allerdings auch dazu gezwungen, jeden Empfänger zu kennen, um beim Austritt des letzten Empfängers aus einer Gruppe die Weiterleitung des Multicastverkehrs einzustellen. Wenn die genaue Kenntnis jedes Empfängers nicht benötigt wird, erlauben die zusätzlichen spezifischen Anfragen eine ressourcensparende Implementierung ohne eine genaue Buchführung über jeden einzelnen Empfänger.

Zusammenfassend bilden die bei einer Änderung der Gruppenmitgliedschaft und zwecks Robustheit mehrfach von den Hosts gesendeten Änderungsberichte und die spezifischen Anfragen des Queriers den Mechanismus, mit dem Router schnell und weitgehend zuverlässig neue oder den Wegfall alter Empfänger für bestimmte Gruppen und Quellen erkennen. Die in großem zeitlichen Abstand periodisch gesendeten generellen Anfragen überprüfen lediglich den bereits etablierten Zustand und bilden den nach dem Soft-State-Prinzip arbeitenden absichernden Mechanismus, sollte der eigentliche Mechanismus durch Ausfall eines Hosts oder Verlust aller Änderungsnachrichten versagen.

Beispiel

Anhand des Weg-Zeit-Diagramms in Abbildung 2.11 soll der Signalisierungsablauf der Gruppenmanagementprotokolle beispielhaft erläutert werden. Er ist bei IGMPv3 und MLDv2 identisch. Auf dem Link befinden sich zwei Router Rt1 und Rt2 und zwei Hosts H1 und H2. Rt1 sei Forwarder für alle Quellen ausgenommen für die Quelle S, für die Router Rt2 der Forwarder ist. Nachrichten sollen, anders als üblich, nicht wiederholt werden.[23] Jeder Router ist zu Beginn ein Querier. Router Rt2 sendet daher nach dem Starten seine generelle Anfrage „Query(*)“ per Broadcast an alle Knoten auf dem Link. Da es noch keine Multicastempfänger gibt, erhält er keinen Bericht zurück.

[22] Multicastadresse „All Systems on this Subnet“ 224.0.0.1 bzw. „Link-Local Scope All Nodes“ ff02::1

[23] entspräche dem Setzen der Robustheitsvariable von IGMP/MLD auf 1; empfohlener Wert ist 2

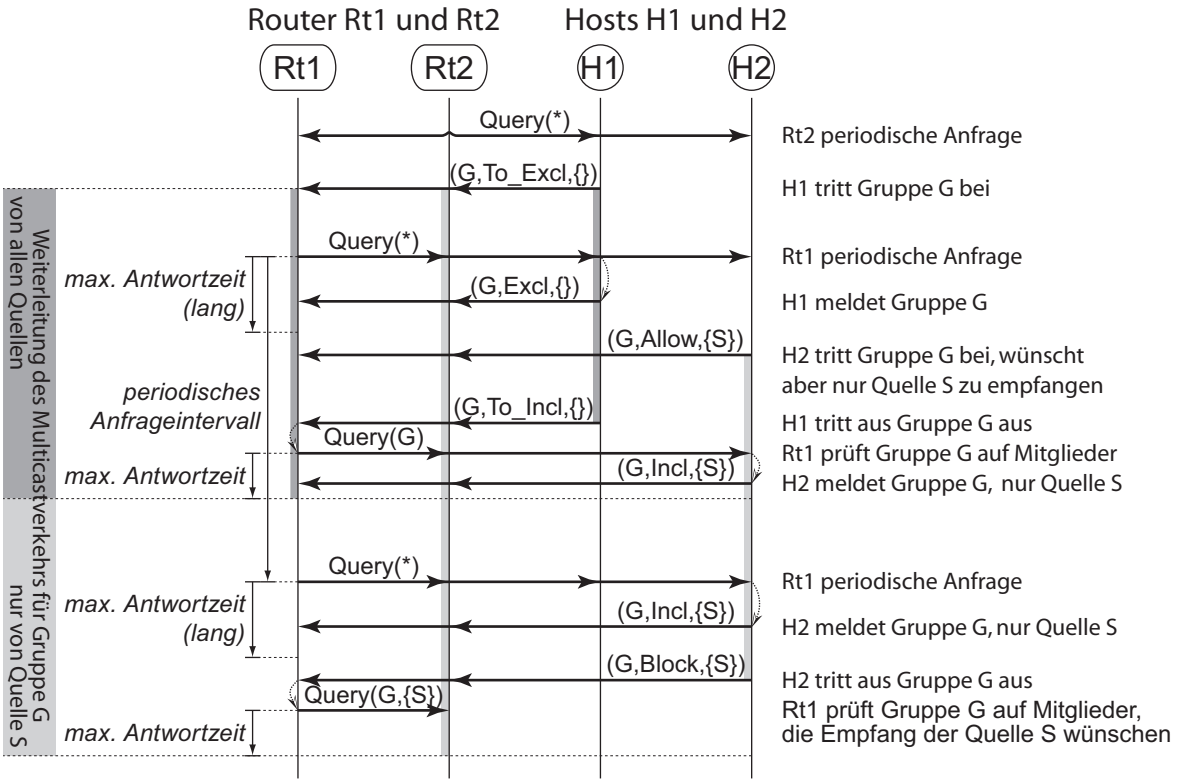


Abbildung 2.11 Beispiel zur Verdeutlichung der Arbeitsweise der Gruppenmanagementprotokolle IGMPv3 und MLDv2.

Host H1 tritt der Gruppe G bei und sendet dazu einen Änderungsbericht per linklokalem Multicast an die Router. Änderungsberichte zeigen je Gruppe entweder einen Wechsel des Filtermodus nach Einbinden (To_Incl) oder Ausschließen (To_Excl) an zusammen mit der Menge an zu filternden Quellen. Oder der Filtermodus bleibt unverändert und es werden nur die zusätzlich zu empfangenen (Allow) oder die nicht mehr zu empfangenen (Block) Quellen benannt. Initial befindet sich der Filter jeder Gruppe im Modus Einbinden mit einer leeren Menge an zu empfangenen Quellen. H1 sendet daher für Gruppe G ein To_Excl mit einer leeren Menge {} an zu filternden Quellen, da H1 alle Quellen empfangen will, die an die Gruppe G senden. Die Router leiten bei Empfang dieses Berichts den Multicastverkehr auf den Link weiter, Rt2 den von Quelle S, Rt1 den aller übrigen Quellen. Das Weiterleiten ist durch einen dunkelgrauen Balken für alle Quellen (außer S) bei Rt1 und durch einen hellgrauen Balken für Verkehr von Quelle S bei Rt2 angedeutet. Der dunkelgraue Balken bei H1 zeigt die Zeit an, in der H1 Multicastverkehr der Gruppe G empfängt (inklusive Quelle S).

Später schickt Rt1 seine periodische generelle Anfrage „Query(*)“, die auch Rt2 empfängt. Rt2 habe eine größere Schnittstellenadresse als Router Rt1, so dass Rt2 zum Nicht-Querier wird. Host H1 antwortet – angedeutet durch den Pfeilbogen – innerhalb der in der Anfrage angegebenen (langen) maximalen Antwortzeit mit einem Istzustandsbericht „(G,Excl,{})“. Dieser nennt alle Gruppen, in H1 Mitglied ist, hier nur Gruppe „G“, und für jede der genannten Gruppen den jeweils aktuellen Filtermodus, hier Ausschließen „Excl“, mit den ausgeschlossenen Quellen, hier keine „{}“. Nun Tritt H2 der Gruppe G bei, wünscht aber nur den Empfang der Quelle S. Er behält entsprechend seinen Filtermodus Einbinden bei und fügt lediglich die Quelle S zu den zu empfangenen Quellen hinzu „(G,Allow,{S})“.

H1 tritt wieder aus der Gruppe aus, indem er den Filtermodus zurück nach Einbinden wechselt und mit einer leeren Menge von Quellen den Empfang allen Verkehrs für die Gruppe unterbindet „(G,To_Incl,{})“. Da H1 der letzte Empfänger der Gruppe G mit Filtermodus Ausschließen war, fragt der Querier Rt1 per gruppenspezifischer Anfrage nach, ob tatsächlich kein weiterer Empfänger für die Gruppe existiert. Nur Host H2 antwortet mit dem Istzustandsbericht „(G,Incl,{S})“. Nach Ablauf der maximalen Antwortzeit kann auch routerseitig der Filtermodus nach Einbinden gewechselt werden mit S als einziger Quelle. Da Rt1 jedoch Verkehr von S nicht weiterleitet, hat Rt1 hiermit keinen Verkehr der Gruppe mehr weiterzuleiten. Lediglich Router Rt2 leitet unverändert Verkehr von S an G auf den Link weiter, da der Wechsel von „(G,Excl,{})“ nach „(G,Incl,{S})“ bei ihm keine Auswirkungen auf die Weiterleitung hat. Es sei angemerkt, dass die an die Gruppenadresse gesendete gruppenspezifische Anfrage den Host H1 gar nicht mehr erreicht, da er bereits aus der Gruppe ausgetreten ist (dunkelgrauer Balken endete), was durch die fehlende Pfeilspitze bei H1 angedeutet ist.

Auch wenn Router Rt1 momentan keinen Multicastverkehr mehr weiterleitet, bleibt er weiterhin als Querier zuständig für das Senden von Anfragen. Nach Ablauf des periodischen Anfrageintervalls sendet Rt1 erneut eine generelle Anfrage „Query(*)“, auf die diesmal Host H2 nach einer zufälligen Verzögerung mit seinem Istzustandsbericht „(G,Incl,{S})“ antwortet. Beendet schließlich auch Host H2 den Multicastempfang und sendet den Änderungsbericht „(G,Block,{S})“, fragt der Querier Rt1 mit der Quellen-und-Gruppen-spezifischen Anfrage „Query(G,{S})“ nach, ob H2 tatsächlich der letzte Empfänger dieser Gruppe für diese Quelle war. Da innerhalb der maximalen Antwortzeit kein Bericht beim Forwarder von S, Router Rt2, eintrifft, beendet dieser die Weiterleitung auf den Link. Wie lange Rt2 warten muss, entnimmt er der in der Anfrage durch Rt1 enthaltenen maximalen Antwortzeit.

2.2.6 IGMP/MLD-Proxying

Die Gruppenmanagementprotokolle lassen sich auch zum Multicastrouting zweckentfremden. Mit dem IGMP/MLD-Proxying [67] wurde ein solches Verfahren standardisiert, das sich als Alternative zu den deutlich komplexeren Multicastroutingprotokollen insbesondere für einfache und kleine Topologien am Netzrand anbietet. Seine Einfachheit gewinnt IGMP/MLD-Proxying vor allem durch die händische Konfiguration der Upstreamschnittstelle. Dies ist zugleich ein wesentlicher Nachteil, da es Weiterleitungsschleifen aufgrund von händischer Fehlkonfiguration nicht verhindern kann. Des Weiteren teilen sich alle Multicastgruppen denselben bidirektionalen Verteilbaum. Die Verkehrskonzentration an dessen Wurzel ist entsprechend hoch.

Ein IGMP/MLD-Proxy, kurz Proxy, übernimmt auf einer ausgezeichneten, stromaufwärts zeigenden Schnittstelle den Hostpart im asymmetrischen Gruppenmanagementprotokoll. Auf seinen übrigen stromabwärts zeigenden Schnittstellen übernimmt er den Routerpart. Alle dort gelernten Gruppenmitgliedschaften fasst er zusammen (vgl. das Zusammenfassen von Filtern im vorigen Abschnitt 2.2.5) und gibt das Ergebnis auf seiner Upstreamschnittstelle bekannt. Ein Proxy ist in der Lage, zwischen seinen Schnittstellen Multicastverkehr in *beiden* Richtungen weiterzuleiten. IGMP/MLD-Proxying etabliert somit einen bidirektionalen Verteilbaum vergleichbar dem des Multicastroutingprotokolls BIDIR-PIM (s. Abschnitt 2.3.2.3).

Ein als Proxy konfigurierter Router leitet den Multicastverkehr von einer oder auf eine bestimmte Schnittstelle nur dann weiter, wenn er der Querier auf dem zugehörigen Link

ist oder es sich um die Upstreamschnittstelle handelt. Die Wahl des Forwarders für einen Downstreamlink erfolgt beim IGMP/MLD-Proxying somit über die Wahl des Queriers. Das Weiterleiten auf einen Downstreamlink erfolgt zudem nur dann, wenn die dort angezeigte Gruppenmitgliedschaft ein Weiterleiten auf den Link tatsächlich erfordert. In umgekehrter Richtung auf die Upstreamschnittstelle wird dagegen aller Verkehr unabhängig von der Gruppenmitgliedschaft weitergeleitet (sofern die Weiterleitung überhaupt zulässig ist, der Proxy also Querier auf der Eingangsschnittstelle downstream ist[24]).

Welche Schnittstelle stromaufwärts zeigt, wird beim IGMP/MLD-Proxying händisch konfiguriert. Es wurde allerdings vorgeschlagen, dies zu automatisieren [182]. Bei der händischen Konfiguration muss darauf geachtet werden, dass es nicht zu Schleifen kommt: Folgt man ausgehend von einem beliebigen Proxy den Pfad stromaufwärts, d. h. über dessen Upstreamschnittstelle und den darüber erreichten Link über den Forwarder des Links weiter über dessen Upstreamschnittstelle usw., darf keine andere stromabwärts liegende Schnittstelle des Ausgangsproxys erreicht werden, auf dem der Ausgangsproxy Querier/Forwarder ist. Alle Pfade müssen stromaufwärts verfolgt am selben Router, der Wurzel, enden.

Die Wurzel ist ein Router, der nicht als Proxy konfiguriert ist und daher keine als stromaufwärts konfigurierte Schnittstelle besitzt. Dieser Router übernimmt auf seinen Links mit Proxys den Routerpart im Gruppenmanagementprotokoll und ist Forwarder für diese Links. Mit seinen übrigen Schnittstellen (zu Links ohne Proxys) nimmt er am Multicastrouting außerhalb des Netzbereichs des IGMP/MLD-Proxyings teil und stellt so die Multicastkonnektivität zum übrigen Netz bzw. dem Internet her. Aus Sicht des Multicastroutings auf der Wurzel bilden alle Links im Netzbereich des IGMP/MLD-Proxyings zusammen einen direkt angeschlossenen Link.

Auf stromabwärts liegende Schnittstellen ohne Empfänger, also ohne Hosts oder Proxys, die per Gruppenmanagementprotokoll eine Gruppenmitgliedschaft angezeigt haben, leitet ein Querier/Forwarder nicht weiter. Es handelt sich bei IGMP/MLD-Proxying somit um ein Multicast- und kein Broadcastverfahren. Man beachte, dass in entgegengesetzter Richtung stromaufwärts immer eine Weiterleitung durch die Forwarder stattfindet, also auch aus Netzbereichen ohne Empfänger nur mit Quellen. Entsprechend nimmt die Verkehrskonzentration in Richtung der Wurzel immer weiter zu, bis an der Wurzel schließlich der Verkehr aller Quellen zusammenfließt.

Abbildung 2.12 links gibt ein Beispiel, wie mittels IGMP/MLD-Proxying in der bereits aus Abbildung 2.7 und Abbildung 2.8 bekannten Topologie Multicastverkehr geroutet werden könnte. Die Upstreamschnittstelle eines Proxys ist mit „Up“ gekennzeichnet. Die Schnittstelle(n), auf denen ein Proxy Querier/Forwarder ist, tragen einen Pfeil. Router 1 nimmt am (internetweiten) Multicastrouting teil. Er ist nicht als Proxy konfiguriert, weist also keine Upstreamschnittstelle auf. Der Querier auf Link B sei Router 2, weil er die kleinste Schnittstellenadresse zu Link B besitze und daher die Wahl zum Querier gegen Router 3 gewinnt. Die Schnittstelle von Router 4 zu Link B ist als Upstreamschnittstelle konfiguriert. Er nimmt daher als Host und nicht als Router am Gruppenmanagementprotokoll auf Link B teil und kann kein Querier werden. Die Quelle S befindet sich diesmal am Link D. Ihr Verkehr wird von den Routern 4 und 2 bis zur Wurzel weitergeleitet. Die vom Empfänger R angezeigte Gruppenmitgliedschaft wird von Router 3, dem Querier und Forwarder von

[24] Die Spezifikation [67] ist diesbezüglich ungenau/fehlerhaft [107].

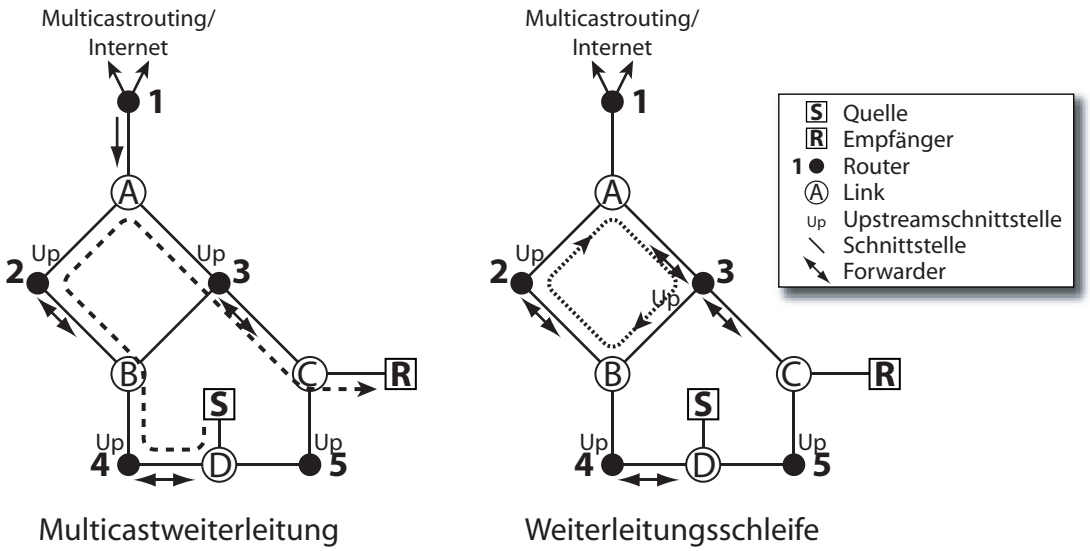


Abbildung 2.12 Beispiele zu IGMP/MLD-Proxying: Weiterleitung über den bidirektionalen Verteilbaum (links), Weiterleitungsschleife aufgrund einer Fehlkonfiguration (rechts).

Link C, an Link B weitergegeben und umgekehrt der zugehörige Verkehr, hier der von Quelle S, auf Link C weitergeleitet. Zwar ist der Verkehr von S bereits auf dem ebenfalls an Router B angeschlossenen Link B verfügbar, jedoch ist B hier weder Querier noch ist Link B an seiner Upstreamschnittstelle angeschlossen. Router B leitet daher von Link B eintreffenden Verkehr nicht weiter. Gleiches gilt für Router 5, der Verkehr von Link D nicht auf Link C weiterleitet.

Abbildung 2.12 rechts zeigt, wie durch die Fehlkonfiguration von Router 3 eine Weiterleitungsschleife entstehen kann. Als Upstreamschnittstelle ist diesmal die zu Link B führende Schnittstelle konfiguriert. Zudem habe Router 3 die kleinste Schnittstellenadresse an Link A, so dass er zum Querier/Forwarder von Link A bestimmt wird. Hierdurch wird jedes Multicastpaket auf Link A durch Router 3 auf Link B weitergeleitet und von Router 2 zurück auf Link A. Dies gilt unabhängig von etwaigen Gruppenmitgliedschaften, da beim IGMP/MLD-Proxying die Weiterleitung stromaufwärts unabhängig hiervon erfolgt.

2.3 Multicastrouting

Die Aufgabe des Multicastroutings unterscheidet sich wesentlich von der des Unicastroutings. Anders als die Unicastadresse ist die Gruppenadresse lediglich Identifikator und nicht zugleich auch Lokator. Anders als bei Unicast lässt sich daher aus der Zieladresse eines Multicastpakets, der Gruppenadresse, nicht auf die Lage der Empfänger, der Gruppenmitglieder, im Netz schließen. Die fehlende Funktion des Lokators wird vom Multicastrouting übernommen. Es ist die primäre Aufgabe des Multicastroutings, Quellen und Gruppenmitglieder im Netz aufzufinden und einen Weiterleitungspfad zwischen ihnen zu etablieren. Hosts machen sich den Routern durch das Senden von Multicastpaketen als Quelle und durch das Senden eines Berichts im Gruppenmanagementprotokoll als Gruppenmitglied bekannt. Die Prinzipien, nach denen beim Multicastrouting Pfade bzw. Verteilbäume etabliert werden können, wurden bereits in Abschnitt 2.2.2 erläutert. Im

Folgenden werden die konkreten Multicastroutingprotokolle, wie sie derzeit im Internet eingesetzt werden oder für das Internet vorgeschlagen wurden, im Einzelnen vorgestellt.

Eine Reihe von Multicastroutingprotokollen sind von der IETF standardisiert worden, zuletzt die der PIM-Familie: PIM-SM (Protocol Independent Multicast – Sparse Mode) [51, 66], PIM-DM (PIM Dense Mode) [2, 54] und BIDIR-PIM (Bi-directional PIM) [75]. Sie benötigen ein beliebiges – daher die Bezeichnung als protokollunabhängig (Protocol Independent) – Unicastroutingprotokolle zur Bereitstellung der MRIB, also den Routen zu allen (Unicast-)Adressen. Das erste protokollunabhängige Multicastroutingprotokoll war CBT (Core Based Trees) [13, 12, 11], dessen Konzepte des Rendezvousstellen-basierten Routings und der bidirektionalen Verteilbäume in den Protokollen PIM-SM und BIDIR-PIM aufgegangen und weiterentwickelt worden sind. Die vor den Protokollen der PIM-Familie entstandenen Multicastroutingprotokolle DVMRP (Distance Vector Multicast Routing Protocol) [187, 145, 146] und MOSPF (Multicast Open Shortest Path First) [122] sind in diesem Sinne nicht protokollunabhängig, sondern bestimmen die MRIB selbst. DVMRP dient ausschließlich dem Routing von Multicastverkehr. MOSPF dagegen ist eine abwärtskompatible Erweiterung von OSPF, so dass ein MOSPF-Router sowohl Unicast- wie Multicastrouting übernehmen kann. CBT, DVMRP und MOSPF wurden nur für IPv4 standardisiert. MOSPF ist des weiteren nicht für Source Specific Multicast geeignet, da seinem Nachrichtenformat ein Feld für die Quelladresse fehlt. Die Protokolle der PIM-Familie sollen zukünftig die einzigen Multicastroutingprotokolle im Internet sein, die älteren Protokolle MOSPF und CBT wie auch BGMP (s. u.) wurden daher als historisch reklassifiziert [153]. MOSPF ist jedoch noch in vielen Routerimplementierungen verfügbar und kann wie DVMRP für reinen Intradomänenmulticast eingesetzt werden. CBT war für die Entwicklung des Rendezvousstellen-basierten Multicastroutings wegweisend, das Routingprotokoll CBT selbst hat jedoch nie praktische Bedeutung erlangt.

Für das Interdomänenmulticastrouting wurde BGMP (Border Gateway Multicast Protocol) [103, 170] vorgeschlagen, das mit einem beliebigen der zuvor genannten Multicastroutingprotokolle im Inneren einer Domäne kombiniert werden kann. Es hat sich wie frühere Vorschläge für eine zwei- oder mehrstufige Hierarchie im Multicastrouting, u. a. HPIM [76], OCBT [162] und dessen Weiterentwicklung HIP [163, 161], im Internet nicht durchsetzen können. Stattdessen erfolgt das Interdomänenmulticastrouting derzeit durchgängig Ende zu Ende mit PIM-SM unter Zuhilfenahme von MSDP [68] (IPv4) bzw. Gruppenadressen mit eingebetteter Rendezvousstellenadresse [154] (IPv6). Einen Überblick über weitere Vorschläge zum Multicastrouting, z. B. mittels Steinerbäumen, gibt [45]. Sie werden im Folgenden nicht weiter betrachtet. Bevor näher auf die einzelnen Multicastroutingprotokolle, zuerst die drei der PIM-Familie, danach auf die historischen Protokolle, eingegangen wird, soll zunächst die Umsetzung der Multicastweiterleitung in den Routern, der Multicast Forwarding Cache, erläutert werden.

2.3.1 MFC – Multicast Forwarding Cache

Multicastrouting speichert die ermittelten Informationen über Quellen, Empfänger, Nachbarrouter usw. in Form von Multicastroutingtabellen (Multicast Routing Table, MRT). Jedes Multicastroutingprotokoll verwendet eigene MRTs. Die MRTs enthalten mehr Informationen als für die eigentliche Paketweiterleitung benötigt wird. So ist es beispielsweise für die Weiterleitung unerheblich, ob sie ein direkt angeschlossener Empfänger oder ein Nachbarrouter wünscht. Aus den MRTs aller Multicastroutingprotokolle wird daher der

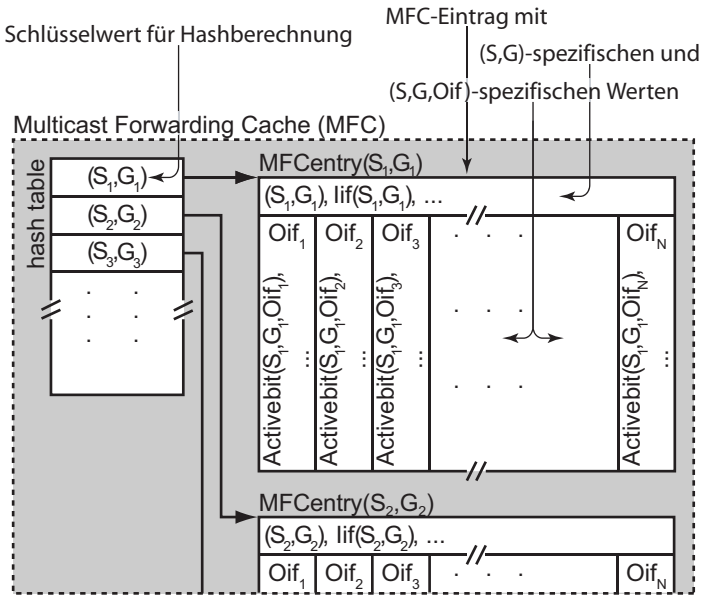


Abbildung 2.13 Aufbau des Multicast Forwarding Caches (MFC) im Unix-Modell.

sogenannte Multicast Forwarding Cache (MFC) erzeugt. Der MFC bestimmt, wie eintreffende Multicastpakete in Abhängigkeit ihrer Gruppenadresse, ggf. auch ihrer Quelladresse und/oder Eingangsschnittstelle, auf die verschiedenen Ausgangsschnittstellen zu duplizieren sind. Die Ableitung des MFCs aus den MRTs entspricht dem Vorgehen bei Unicast, wo aus der Routing Information Base die Forwarding Information Base gebildet wird. Für den MFC findet sich daher manchmal auch der Begriff MFIB (Multicast FIB). Die Kommunikation zwischen Routingebene (Ort der MRTs) und Weiterleitungsebene (Ort des MFCs) ist vergleichsweise aufwendig. Bei einer Implementierung des MFCs in Software sind es Systemaufrufe mit dem Kontextwechsel zwischen User und Kernel Space, bei Hardwareroutern mit verteilter Switcharchitektur (s. u.) ist es die Kommunikation zwischen der Routingprozessorkarte und den Schnittstellenkarten. In der Routingebene befindet sich daher eine Kopie des MFCs, so dass sie ihre Zugriffe auf den eigentlichen MFC in der Weiterleitungsebene auf das minimal Nötige, den rein schreibenden Zugriff, beschränken kann.

Das Unix-Modell

Der MFC (s. Abbildung 2.13) wird in Software gewöhnlich mittels Hashtabelle, in Hardware z. B. mittels Assoziativspeicher (Content Addressable Memory, CAM), implementiert. Dabei dient das Tupel aus Quell- (Source, S) und Gruppenadresse (Group, G) als Schlüssel. Bei Hashtabellen kann es zu Kollisionen kommen. Unterschiedliche Tupel (S,G) liefern dann den gleichen Hashwert. Solche MFC-Einträge mit dem gleichen Hashwert sind üblicherweise in Form einer Überlaufliste einfach verkettet (in Abbildung 2.13 nicht dargestellt). Die in jedem MFC-Eintrag enthaltene Quell- und Gruppenadresse erlaubt die eindeutige Identifikation bei der anschließenden linearen Suche in der Überlaufliste. Dies entfällt bei Ausführung in Hardware mit CAMs, da hier Kollisionen prinzipbedingt nicht auftreten können.

Jeder MFC-Eintrag *MFCentry(S,G)* ist spezifisch für eine bestimmte Gruppe G und Quelle S. Er enthält zum einen eine Reihe von schnittstellenunabhängigen, also nur vom Tupel

(S,G) abhängigen Werten. Wichtig für die Weiterleitung ist die Eingangsschnittstelle *Iif*, auf dem der Router die Multicastpakete der Quelle S für Gruppe G erwartet, geschrieben *Iif(S,G)*. Weitere (S,G)-spezifische Werte sind z. B. die Paketzähler für Statistikzwecke. Zum anderen enthält ein MFC-Eintrag schnittstellenabhängige Werte. In Abbildung 2.13 sind diese für jede (Ausgangs-)Schnittstelle Oif_i, $i = 1 \ldots N$, jeweils in einem Rechteck zusammengefasst. So existiert für jede der N Schnittstellen des Routers ein Flag, das im Folgenden als *Activebit* bezeichnet werden soll. Es zeigt an, ob die jeweilige Ausgangsschnittstelle Oif aktiv ist, ob also auf diese Schnittstelle Multicastpakete der Quelle S für Gruppe G weitergeleitet werden sollen oder nicht. Ein einzelnes Activebit im gesamten MFC wird durch das Tripel aus Quelle S, Gruppe G und Ausgangsschnittstelle Oif eindeutig identifiziert und soll entsprechend als *Activebit(S,G,Oif)* geschrieben werden. Ein Activebit(S,G,Oif) ist die endgültig aufgelöste Weiterleitungsinformation und das Äquivalent zum Next Hop bei Unicast. Eine Auflösung in die Sicherungsschichtadresse wie in der Unicast Forwarding Information Base ist nicht nötig, da eine feste Abbildung von IP-Multicastadressen auf Sicherungsschicht-Multicastadressen besteht.[25] Im Rahmen des in Kapitel 4 entworfenen Verfahrens zum Management der Dienstgüteunterstützung von Multicast werden sowohl weitere schnittstellenabhängige wie -unabhängige Werte spezifiziert.

Ein MFC-Eintrag wird typischerweise verkehrsgesteuert (Data Driven) angelegt, also erst, wenn ein Router ein Multicastpaket von der Quelle S an die Gruppe G empfangen und weitergeleitet hat. Grund ist zum Einen, dass bei ASM die Adresse einer Quelle einem Router im Vorhinein gar nicht bekannt ist. Zum anderen lässt sich damit ungenutzter, im MFC unnötig Speicherplatz belegender Weiterleitungszustand vermeiden. Ein MFC-Eintrag wird wieder gelöscht, wenn der Verkehr von der Quelle für einige Zeit ausgeblieben ist. Die Bezeichnung des MFCs als Cache statt als Information Base spiegelt dieses Verhalten wider.

Die bisher beschriebene Form des MFCs wird als *Unix-Modell* bezeichnet. Sie wurde mit dem Unix 4.4BSD-Lite bzw. dessen Netzwerkstack Net/3 1993 eingeführt [190]. Bei verteilten Hardwarearchitekturen wird der MFC dagegen auch verteilt implementiert. Sind die Schnittstellenkarten über einen Bus miteinander verbunden, bietet sich aufgrund der Busarchitektur ein interner Broadcast an, bei dem ein Multicastpaket von jedem Eingang an alle Ausgänge gesendet wird und erst dort mittels Filter je Tupel (S,G) ggf. verworfen wird. Ein solcher Filter bzw. MFC-Eintrag am Ausgang enthält nur noch ein einziges Activebit(S,G,Oif), jedoch existieren N MFCentry(S,G), einer je Ausgang Oif. Neuere Switcharchitekturen verwenden auch intern Multicast und filtern am Eingang. Jeder der N Eingänge besitzt einen eigenen MFC mit MFC-Einträgen ohne den hier bereits implizit bekannten Wert der Eingangsschnittstelle Iif. Die verteilt gespeicherten Activebits besitzen jedoch dieselben Werte wie im zentralen MFC des Unix-Modells. Die Berechnung der Werte im MFC aus denen der MRTs bleibt unverändert.

Dies gilt nicht mehr für kombinierte Eingangs- und Ausgangsfilter, da sie nicht mehr nur für einzelne Gruppen sondern jeweils für einen ganzen Bereich von Gruppenadressen gelten [175]. Die kombinierten Ein-/Ausgangsfilter bringen jedoch nur unter speziellen Randbedingungen – zufällig verteilte Gruppenadressen, deutlich unterschiedliche Anzahl von aktiven Ausgangsschnittstellen und Routerschnittstellen, aktive Ausgangsschnittstellen

[25] auf MAC48-Adressen (z. B. Ethernet): 01-00-5e-00-00-00 plus die letzten (niedrigstwertigen) 23 bit einer IPv4-Gruppenadresse bzw. 33-33-00-00-00-00 plus die letzten 32 bit einer IPv6-Gruppenadresse

zufällig verteilt – eine wesentliche Ersparnis an Weiterleitungszustand. Ihr praktischer Einsatz ist nicht bekannt.

Quellenunspezifische MFC-Einträge

Das klassische Multicastrouting nach dem RPM-Verfahren, wie es DVMRP und auch PIM-DM nutzen, bestimmt einen quellenspezifischen Verteilbaum. Es benötigt entsprechend quellenspezifischen Weiterleitungszustand wie im Unix-Modell. Die Rendezvousstellen-basierten Multicastroutingprotokolle CBT, PIM-SM und BIDIR-PIM führen zusätzlich quellenunabhängigen Routingzustand ein. Es bietet sich daher an, hierfür entsprechend auch quellenunabhängigen, also nur noch von der Gruppenadresse abhängigen Weiterleitungszustand *MFCentry(*,G)* zu verwenden. Ein solcher MFC-Eintrag gilt für beliebige Quellen, dargestellt durch den Stern „*“. In diesem Fall sind bei Ausführung in Software zwei Hashwertberechnungen je weitergeleitetem Multicastpaket nötig, wobei der quellenspezifische Zustand, wenn vorhanden, Vorrang hat. Bei Ausführung in Hardware kann dies über eine Priorisierungslogik gelöst werden, so dass weiterhin nur ein Lookup nötig ist (Single-Cycle Multiple Logical CAM) [111].

Noch weiter reduziert wird der Weiterleitungszustand durch bidirektionale quellen- und gruppenunabhängige Spannbäume, wie sie BIDIR-PIM nutzt. Im Ergebnis benötigen Router in Netzbereichen ohne Empfänger auch keinen gruppenspezifischen Zustand mehr. Es genügt allein rendezvousstellenspezifischer Zustand. Ähnlich dem Unicastrouting teilen sich ganze Bereiche von Gruppenadressen – Gruppenadresspräfixe – eine Rendezvousstelle und damit eine Zieladresse für die Weiterleitung stromaufwärts. Weiterleitungszustand kann hier wie bei Unicast unabhängig von einer etwaigen späteren Nutzung bereits etabliert werden, wenn die zuständige Rendezvousstelle bekannt wird. Der Lookup erfolgt vergleichbar dem Longest Prefix Match bei Unicast durch das Matchen von Präfixen anstatt von vollständigen Gruppenadressen wie im Unix-Modell.

2.3.2 PIM – Protocol Independent Multicast

Die drei Multicastroutingprotokolle der PIM-Familie PIM-SM, PIM-DM und BIDIR-PIM nutzen nicht nur dasselbe Nachrichtenformat, sondern teilen sich eine Reihe von Protokollmechanismen bzw. nutzen diese auf sehr ähnliche Weise. Diese Gemeinsamkeiten werden im Folgenden erläutert und auf Unterschiede, wo vorhanden, hingewiesen. Danach wird auf die genaue Arbeitsweise jedes der drei Protokolle in einem eigenen Abschnitt eingegangen. Der ebenfalls das PIM-Nachrichtenformat nutzende Bootstrap-Router-Mechanismus wird im Anschluss daran vorgestellt. Er ist selbst kein Multicastroutingprotokoll, mit ihm werden jedoch die von PIM-SM und BIDIR-PIM benötigten Zuordnungen von Gruppen- zu Rendezvousstellenadressen allen Routern intradomain bekanntgegeben.

PIM-Nachrichten

Alle PIM-Protokolle arbeiten nach dem Soft-State-Prinzip. Ohne Wiederholung der PIM-Nachrichten wird durch sie in den Nachbarroutern etablierter Zustand nach Ablauf von dessen Gültigkeit wieder gelöscht. Die meisten zwischen Nachbarroutern auf einem Link ausgetauschten PIM-Nachrichten werden per linklokalem Multicast an die Gruppe „All PIM Routers“ (224.0.0.13, ff02::d) mit TTL/Hop Count 1 gesendet. Innerhalb eines Links per Unicast ausgetauscht werden nur die von PIM-DM genutzten Nachrichten Graft

und Graft-Ack. Ausschließlich per Unicast gesendet werden alle Link-übergreifenden Nachrichten. Dies sind die von PIM-SM genutzten Nachrichten Register und Register-Stop, mit denen die Weiterleitung der Pakete einer Quelle zur Rendezvousstelle erfolgt, und im Rahmen des Bootstraprouter-Mechanismus die Candidate-RP-Advertisement-Nachrichten der Rendezvousstellenkandidaten an den Bootstraprouter.

Hello-Mechanismus

Mit dem Hello-Mechanismus erkennen PIM-Router ihre Nachbarn auf einem Link. Sie werden ab dem Start eines Routers periodisch alle halbe Minute gesendet. Ohne zuvor ein Hello von einem Nachbarn empfangen zu haben, akzeptieren Router von diesem keine anderen PIM-Nachrichten. Über die in den Hello-Nachrichten enthaltenen Hello-Optionen lernen Nachbarrouter ihre jeweiligen Fähigkeiten und Konfigurationen gegenseitig kennen.

Die Hello-Nachrichten nennen wie beim Soft-State-Prinzip üblich eine Gültigkeitsdauer. Sie gibt an, nach welcher Zeit ohne empfangene Hello-Nachrichten die anderen Router am Link einen Router für ausgefallen erklären und den ihn betreffenden Zustand löschen. Mit einer Gültigkeitsdauer von null können Router kurz vor Deaktivieren einer Schnittstelle oder Ändern ihrer Adresse das Löschen allen mit ihnen in den Nachbarroutern assoziierten Zustands sorgen. Fällt ein Router unvorhergesehen aus, kann er keine Gültigkeitsdauer von null mehr senden. Um nach dem Neustart trotzdem schnell das Routing wiederherzustellen, wird über den Hello-Mechanismus eine Generation ID bekannt gegeben, die bei jedem Neustart zufällig neu gewählt wird. Erkennen Nachbarrouter eine geänderte Generation ID, löschen sie ebenfalls allen alten mit dem neu gestarteten Router assoziierten Zustand. Zusätzlich senden sie ihrerseits sofort eine Hello-Nachricht, damit der neu gestartete Nachbar schnell wieder seine Nachbarn kennenlernt und schnell wieder PIM-Nachrichten ausgetauscht werden können.

Join/Prune-Nachricht und kodierte Adressen

Alle Adressen in PIM-Nachrichten sind grundsätzlich kodiert, um Adressfamilie, Adressmaske und einige Flags übermitteln zu können. Durch die Kodierung der Adressen sind die PIM-Protokolle zudem unabhängig von der Adressfamilie und für IPv4, IPv6 und andere einsetzbar. Allerdings dürfen unterschiedliche Adressfamilien in einer Nachricht nicht gemischt werden. Mit der Adressmaske lässt sich ein Bereich von Gruppenadressen zusammenfassen. An Flags sind für Gruppenadressen u. a. das Bi-directional PIM (B) (gesetzt, wenn Gruppe durch BIDIR-PIM geroutet wird) und für Quelladressen die drei Flags Sparse (S), Wild Card (W) und Rendezvousstellenbaum (R) definiert. Letztere werden jedoch nur von PIM-SM verwendet. Andere PIM-Protokolle setzen sie null. PIM-SM sendet somit seine Gruppen- und Quelladressen mit B=0 und S=1, PIM-DM mit B=0 und S=0 und BIDIR-PIM mit B=1 und S=0, wodurch sie eindeutig einem der drei Protokolle zugeordnet werden können.

Joins und Prunes werden mit derselben PIM-Nachricht gesendet, der *Join/Prune-Nachricht*. Sie ist neben der Hello-Nachricht die zweite von allen Multicastroutingprotokollen der PIM-Familie genutzte Nachricht. Die Join/Prune-Nachricht enthält die Adresse jenes Upstreamrouters, an den die Nachricht gerichtet ist, d. h. von dem der Downstreamrouter die Multicastpakete auf den Link weitergeleitet (oder nicht mehr weitergeleitet) haben möchte. Die Nachricht selbst wird jedoch per Multicast gesendet, damit sie auch von den übrigen Router am Link empfangen werden kann. Dies ist u. a. für das Überschreiben von Prunes

wichtig (s. u.). Joins und Prunes werden in der Join/Prune-Nachricht durch eine Liste von Gruppenadressen dargestellt, wobei jede Gruppenadresse wiederum mit einer Liste von zu joinenden und einer Liste von zu prunenden Quelladressen versehen ist. Mit einer Join/Prune-Nachricht kann so eine große Anzahl an unterschiedlichen Joins und Prunes gleichzeitig gesendet werden, im Wesentlichen nur begrenzt durch die maximal unterstützte Paketgröße (implementierungsabhängig). Dies erlaubt im Rahmen der aufgrund des Soft-State-Prinzips nötigen periodischen Wiederholungen eine Aggregation der Joins und Prunes in wenige Nachrichten.

Join/Prune-Nachrichten werden im Abstand von einer Minute periodisch wiederholt, um den jeweiligen Zustand im Upstreamrouter aufrechtzuerhalten. Die Gültigkeitsdauer der Join/Prune-Nachricht ist wie beim Hello explizit in der Nachricht selbst enthalten. Hierdurch ist es möglich, den zeitlichen Abstand dynamisch der Anzahl an Gruppen und damit an zu sendenden Nachrichten anzupassen [160]. Standardimplementierungen nutzen dies jedoch nicht, sondern verwenden fest konfigurierte Werte. Die Gültigkeitsdauer wird aus Robustheitsgründen so gewählt, dass sie dem 3,5-fachen der aktuellen Periodendauer entspricht. Mit dem derzeitigen PIM-Nachrichtenformat ist so eine maximale Periodendauer von etwas über 5 Stunden möglich. Die Wahl der Periodendauer hat keinen Einfluss auf Geschwindigkeit des Auf- und Abbaus von Routingzustand. Dieser erfolgt nicht durch periodische sondern durch getriggerte Join/Prune-Nachrichten. Sie werden u. a. ausgelöst durch einen neuen lokalen Empfänger (angezeigt vom Gruppenmanagementprotokoll), den Empfang einer Join/Prune-PIM-Nachricht von einem Router stromabwärts, eine geänderte Route zu Quelle oder Rendezvousstelle, eine geänderte Rendezvousstelle oder das Ausbleiben der Hellos eines Nachbarrouters.

Existieren mehrere Downstreamrouter an einem Link, die dieselbe Gruppe (und Quelle) gejoint haben, genügt es, wenn nur einer dieser Router das periodische Senden der entsprechenden Joins übernimmt. Die übrigen Router setzen dazu ihr eigenes Join entsprechend der im gesendeten Join genannten Gültigkeitsdauer aus. Dies wird als Join Suppression bezeichnet und lässt sich über die entsprechende Hello-Option deaktivieren. Man beachte, das getriggerte Join-Nachrichten nie unterdrückt werden. Dies würde erfordern, dass ein Router bereits dann auf Joins anderer Nachbarrouter hört, wenn er selbst noch gar keinen Zustand für die betreffende Gruppe (und Quelle) besitzt.

Überschreiben von Prunes

Möchte ein Downstreamrouter den Empfang von Multicastpaketen beenden, sendet er eine Prune-Nachricht an den Forwarder des Upstreamlinks. Bevor dieser tatsächlich die Weiterleitung auf den Link einstellt, gibt er den übrigen Routern am Link die Möglichkeit, das Prune mit einem eigenen Join zu überschreiben. Schließlich ist ihm aufgrund der Join Suppression nicht bekannt, ob noch weitere Downstreamrouter Interesse an dem zu prunenden Multicastverkehr haben.

Über eine Hello-Option kann die Zeit J/P_Override_Interval und damit Geschwindigkeit, mit der ein Prune ausgeführt wird, vorgegeben werden. Der Forwarder wartet nach einem Prune J/P_Override_Interval lang auf ein Join, dass das Prune überschreibt. Die Downstreamrouter verzögern ihrerseits das zum Überschreiben des Prune von ihnen vorbereitete Join zufällig bis zu einem Wert etwas kleiner als J/P_Override_Interval. Dies vermeidet eine Flut gleichzeitiger Join/Prune-Nachrichten an den Forwarder. Zudem senden sie ihr eigenes Join überhaupt nicht mehr, wenn sie vorher ein entsprechendes

Join eines anderen Routers empfangen. Somit erfolgt das Überschreiben meist mit nur einer Nachricht. Wenn bis zum Ablauf von J/P_Override_Interval der Forwarder kein Join empfangen hat, führt er das Prune aus und stoppt die Multicastweiterleitung für die genannte Gruppe und ggf. Quelle. Als zusätzliche Absicherung sendet der Forwarder beim Ausführen des Prunes ein Prune-Echo – eine an sich selbst gerichtete Join/Prune-Nachricht. Falls ein Downstreamrouter das ursprüngliche Prune nicht empfangen haben sollte, erhält er hierdurch erneut die Möglichkeit, das Prune zu überschreiben und die Multicastweiterleitung wiederherzustellen.

Bei nur zwei Routern an einem Link, wie z. B. bei einem Punkt-zu-Punkt-Link, ist das Überschreiben von Prunes überflüssig, da es nur genau einen Downstreamrouter gibt. Das Prune wird hier vom Forwarder grundsätzlich sofort ausgeführt.

Assert-Mechanismus

Asserts werden nur von PIM-SM und PIM-DM genutzt und auch nur dann, wenn mehr als ein Upstreamrouter Pakete auf einen Link weiterleitet bzw. weiterleiten könnte, d. h. entsprechenden Routingzustand besitzt. Ein Assert erfolgt sofort mit dem ersten Paket, das ein zweiter Router auf den Link weiterleitet. Dann wird mit Hilfe der Asserts ein Forwarder für die betreffende Quelle und Gruppe bestimmt. PIM-SM unterstützt zusätzlich einen quellenunabhängigen Assert. Der Gewinner des Asserts ist der neue Forwarder. Alle übrigen Router am Link werden zu Assert-Verlierern. Ein Assert gilt nur für eine begrenzte Zeit (drei Minuten). Kurz vor Ablauf wiederholt der Gewinner daher seine Assert-Nachricht. Wenn der Assert-Gewinner seinen Weiterleitungszustand löscht, sendet er ein sogenanntes AssertCancel, d. h. eine Assert-Nachricht mit der schlechtestmöglichen Assert-Metrik, woraufhin alle Router in den Grundzustand wechseln, d. h. keiner ist mehr Gewinner oder Verlierer. Die Assert-Metrik wird im Wesentlichen von der Distanz zur Quelle bzw. im Falle von quellenunabhängigen Asserts zur Rendezvousstelle bestimmt. Die kürzeste Distanz liefert die beste Metrik. Gibt es nach Wegfall des alten Assert-Gewinners und Fordwarders immer noch mehr als einen Router mit Weiterleitungszustand, wird es zu einem neuen Assert kommen, der einen neuen Forwarder bestimmt.

2.3.2.1 PIM-SM – Protocol Independent Multicast – Sparse Mode

Die drei Protokolle der PIM-Familie repräsentieren drei grundsätzlich verschiedene Verfahren des Multicastroutings. PIM-SM [51, 66] ist das universellste, älteste und daher am meisten implementierte und eingesetzte Multicastroutingprotokoll. Es verwendet eine Rendezvousstelle, im Folgenden kurz RP (Rendezvous Point), zur Lokalisation von Quellen und unidirektionale Bäume zum Verteilen des Multicastverkehrs. Pakete der Quellen werden mit sogenannten Register-Tunneln von den Routern bei den Quellen, den Designated Routern, zum RP gesendet. Der Gruppenbeitritt von Empfängern führt zu expliziten Join-Nachrichten Richtung RP und Aufbau eines Verteilbaums mit dem RP als Wurzel, der als Rendezvousstellenbaum (Rendezvous Point Tree, RPT) bezeichnet wird. Der RP entkapselt die über die Register-Tunnel eintreffenden Pakete aller Quellen einer Gruppe und sendet sie über den RPT. Der RPT wird damit von allen Quellen einer Gruppe gemeinsam genutzt.

PIM-SM ist das einzige Protokoll der PIM-Familie, das sowohl quellenunspezifischen (*,G)-Routingzustand als auch quellenspezifischen (S,G)-Routingzustand nutzt. PIM-DM nutzt nur quellenspezifischen (S,G)-Prune-Zustand, BIDIR-PIM nur quellenunspezifischen

(*,G)-Joinzustand. Zum einen wird quellenspezifischer Routingzustand etabliert, wenn ein Empfänger explizit den Empfang nur einer Quelle einer ASM-Gruppe bzw. den Empfang eines SSM-Kanals anfordert. Dies führt zum Aufbau des quellenspezifischen kürzesten Verteilbaums (Shortest Path Tree, SPT) mit der Quelle als Wurzel. Zum anderen kann, wenn gemäß Router-lokaler Policy der zusätzliche quellenspezifische Zustand z. B. aufgrund der hohen Datenrate der Quelle gerechtfertigt erscheint, für eine Quelle aus einer ASM-Gruppe separat ein SPT etabliert werden. Um den doppelten Empfang über SPT und RPT zu vermeiden, wird nach Aufbau des SPTs der Empfang der Quelle über den RPT gestoppt (Prune(S,G,rpt), s. u.).

Designated Router – Register-Tunnel

Mit Hilfe des Hello-Mechanismus wird unter allen PIM-SM-Routern an einem Link einer zum Designated Router (DR) gewählt. Seine Aufgabe ist es, Pakete einer Quelle in *Register-Nachrichten* gekapselt an die für die Gruppe zuständige Rendezvousstelle zu tunneln. Man spricht vom *Register-Tunnel*. An welchen RP die Register-Nachrichten gerichtet werden müssen, ist einem Router durch Teilnahme am BSR-Mechanismus bekannt, kann im Falle von Embedded-RP-Adressen aus der Gruppenadresse selbst abgelesen werden oder ist anderweitig, z. B. statisch, konfiguriert. Der DR ist nicht mit dem Forwarder zu verwechseln. Dieser wird falls nötig per Assert ermittelt und ist gruppen- und ggf. quellenspezifisch, während der DR für alle Gruppen eines RPs derselbe ist.

Das Register-Tunneln von Paketen ist recht aufwendig, insbesondere bei hohen Paketraten, und zudem unnötig, wenn es noch keine Empfänger für die Pakete gibt. Der RP kann daher mittels Register-Stop-Nachricht an den DR einer Quelle das weitere Senden von Register-Nachrichten für eine gewisse Zeit einstellen lassen. Kurz vor deren Ablauf fragt der DR mit einem Null-Register, d. h. einer Register-Nachricht ohne gekapseltes Datenpaket einer Quelle, erneut beim RP an, ob die Unterdrückung aufrecht erhalten werden soll. Wenn ja, antwortet der RP mit einem erneuten Register-Stop. Andernfalls wird die Unterdrückung beendet und Pakete der Quelle fließen in Register-Nachrichten getunnelt erneut zum RP.

Es gibt keine Möglichkeit für den RP, vor dem regulären Ablauf der Unterdrückungszeit den Register-Tunnel zu reaktivieren, beispielsweise, wenn ein Empfänger der Gruppe beitritt. Die einzige Möglichkeit, nach einem Register-Stop den Wiederempfang der Pakete der Quelle zu erreichen, ist, den SPT aufzubauen. Infolgedessen wird ein RP, der noch nicht bereit ist, einen SPT aufzubauen, auch kein Register-Stop senden. Solange werden die Pakete der Quelle noch zum RP getunnelt, egal, ob es Empfänger für die Gruppe bzw. Quelle gibt oder nicht. Ob ein RP bereit ist oder nicht, bestimmt die Policyfunktion *SwitchToSptDesired(S,G)*.

SPT Switchover

Das Umschalten auf den SPT können nur der RP und die empfängerseitigen Forwarder, d. h. Router, die den Multicastverkehr an lokale Empfänger und nicht nur an weitere Downstreamrouter weiterleiten, initiieren. Der Zeitpunkt des Umschaltens wird gesteuert durch die Policyfunktion SwitchToSptDesired(S,G). Wird sie wahr, beginnt der Aufbau des SPTs. Die Policyfunktion wird typischerweise erst bei einer gewissen minimalen Datenrate der Quelle S wahr (implementierungsspezifisch). Die Idee dahinter ist, dass sich ab einer gewissen Senderate der Quelle der Mehraufwand von quellenspezifischem Zustand eines

SPTs lohnt, auch wenn kein Empfänger explizit den quellenspezifischen SPT angefordert hat.

Damit der quellenspezifische Routingzustand aufrecht erhalten wird, wird der sogenannte *KeepaliveTimer(S,G)* gestartet. Er wird beim Empfang von Paketen der Quelle S Pakete an die Gruppe G immer wieder neu gestartet. Bei einer Sendepause hält er den quellenspezifischen Zustand auch ohne Pakete der Quelle und in Abwesenheit von quellenspezifischen Join-Nachrichten oder Gruppenbeitritten für eine gewisse Zeit (dreieinhalb Minuten) aufrecht. Läuft er ab, wird der SPT wieder abgebaut und der Empfang der Quelle über den RPT reaktiviert.

Zu Beginn der Umschaltung auf den SPT, wenn bereits quellenspezifischer Routingzustand existiert, sollen die über den RPT eintreffenden Pakete der Quelle solange noch weitergeleitet werden, bis der Aufbau der SPTs abgeschlossen ist. Den Abschluss der Umschaltung wird vom sogenannten *SPTbit(S,G)* angezeigt.

Das SPTbit wird in der Funktion *Update_SPTbit(S,G,iif)* gesetzt. (Ausschnitt aus [66]):

```
void
Update_SPTbit(S,G,iif) {
  if ( iif == RPF_interface(S)
        AND JoinDesired(S,G) == TRUE                        # Fall:
        AND ( DirectlyConnected(S) == TRUE                  # (v)
              OR RPF_interface(S) != RPF_interface(RP(G)) # (i)
              OR inherited_olist(S,G,rpt) == NULL           # (iii)
              OR ( ( RPF'(S,G) == RPF'(*,G) ) AND           # (iv)
                   ( RPF'(S,G) != NULL ) )                  #
              OR ( I_Am_Assert_Loser(S,G,iif) ) {           # (ii)
     Set SPTbit(S,G) to TRUE
  }
}
```

Es lassen sich fünf Fälle unterscheiden. Sie sind rechts mit Fall (i) bis (v) gekennzeichnet.[26] Für das Setzen des SPTbits ist in allen Fällen Voraussetzung, dass überhaupt quellenspezifischer Routingzustand existiert (JoinDesired(S,G) == TRUE) und dass die Pakete über die richte Schnittstelle eintreffen (iif = RPF_interface(S)). (i) Unterscheidet sich die Schnittstelle von der vorher genutzten Schnittstelle in Richtung RP (RPF_interface(S) != RPF_interface(RP(G))), erfolgt das Setzen mit dem ersten über den SPT eintreffenden Paket. (ii) Wenn es dieselbe Schnittstelle ist, sich die Upstreamnachbarn aber unterscheiden, wird das erste über den SPT eintreffende Paket zu einem Assert zwischen dem Upstreamrouter auf dem SPT und dem auf dem RPT führen, der Downstreamrouter also selbst den Assert verlieren (I_Am_Assert_Loser(S,G,iif)) und zu diesem Zeitpunkt auf den SPT umschalten. (iii) Allerdings kann es sein, dass es zu gar keinem Assert kommt, weil keiner mehr Pakete der Quelle über den RPT empfangen möchte, sondern nur noch über den SPT (inherited_olist(S,G,rpt) == NULL). Dann wird das SPTbit sofort gesetzt. (iv) Ebenso kann der Router das Ende der SPT-Aufbaus nicht erkennen, wenn der Upstreamnachbar für SPT und RPT derselbe ist (RPF'(S,G) == RPF'(*,G)), und setzt beim Aufbau des SPTs das SPTbit sofort. (v) Befindet sich die Quelle direkt auf

[26] Kennzeichnung nur hier zur Erläuterung. Sie ist so nicht im Standard [66] vorhanden.

dem Upstreamlink (DirectlyConnected(S) == TRUE), ist der Aufbau des SPTs natürlich sofort abgeschlossen und das SPTbit kann ebenfalls sofort gesetzt werden.

Die Funktion Update_SPTbit(S,G,iif) wird zwar konzeptionell bei jedem empfangenen Paket aufgerufen. Reale Implementierungen werden dies natürlich einschränken auf jene Fälle, in denen der Paketempfang überhaupt zum Setzen des SPTbit(S,G) führen kann, d. h. auf den einen Fall (i), in dem sich die Eingangsschnittstelle ändert. In allen anderen Fällen wird das Setzen durch Änderungen des Routingzustands und nicht durch den Empfang von Paketen ausgelöst. Allerdings wird zumindest Fall (ii) von Asserts und damit zumindest indirekt vom Empfang von Paketen über den neu etablierten SPT ausgelöst. Da dies der zweite Fall ist, in dem das Setzen des SPTbit(S,G) direkt zu weiteren Aktionen des Routers führt (Senden eines Prune(S,G,rpt), s. u.), ist das Setzen des SPTbit(S,G) in Fall (ii), also beim Übergang des (S,G)-Assert-Zustandsautomaten vom Grundzustand „NoInfo“ nach „I am Assert Loser“, in der Spezifikation [66] explizit aufgeführt (Abschnitt 4.6.1., Aktion A6):

```
If (I is RPF_interface(S)) AND (UpstreamJPState(S,G) == Joined[27])
set SPTbit(S,G) to TRUE
```

I ist die Schnittstelle, auf dem die Assert-Nachricht empfangen wurde. Da „UpstreamJPState(S,G) == Joined“ äquivalent zu dem in Update_SPTbit(S,G,iif) verwendeten „JoinDesired(S,G) == TRUE“ ist, entspricht dies dem Aufruf von Update_SPTbit(S,G,iif) mit iif = I. Dies wird bei der in Kapitel 5 vorgestellten prototypischen Implementierung ausgenutzt.

Die zwei Fälle geänderte Schnittstelle (i) und geänderter Upstreamnachbar (ii) treten normalerweise nur bei genau einem Router auf, der, bei dem SPT und RPT von Quelle bzw. RP aus gesehen aufeinandertreffen. Dieser *Verzweigungsrouter* sorgt zum Zeitpunkt des Setzens des SPTbit(S,G) dafür, dass keine Pakete der Quelle mehr über den RPT eintreffen, indem er ein *Prune(S,G,rpt)* Richtung RP sendet. Das „rpt“ deutet dabei an, dass das Prune nicht wie ein Join(S,G) zum Upstreamnachbarn Richtung Quelle sondern zum Upstreamnachbar Richtung RP gesendet wird. Downstream dieses Routers können sich auch Router befinden, die keinen SPT angefordert haben. In Folge des Prune(S,G,rpt) können sie jedoch die Quelle S nicht mehr über den RPT empfangen. Der Router speist daher den Verkehr aus dem quellenspezifischen SPT in den quellenunspezifischen RPT ein.

PIM-SSM

Sollen nur SSM-Kanäle oder soll ausschließlich mit quellenspezifischen Bäumen vergleichbar PIM-DM geroutet werden, kann in PIM-SM jeglicher quellenunabhängige (*,G)-Routingzustand, Register-Tunnel (DR-Wahl bleibt aber erhalten), KeepaliveTimer und SPTbit wie auch der BSR-Mechanismus entfallen. Ein derart abgespecktes PIM-SM wird als PIM-SSM bezeichnet.

Anycast-RP

Der RP stellt einen Single Point of Failure dar. Zwar kann man mehrere RPs bereitstellen, so dass nur ein Teil der Gruppen von einem Ausfall betroffen ist, jedoch ist die Zeit, bis diese

[27] in [66] steht statt „Joined“ hier fälschlich „true“

Gruppen einen anderen RP nutzen, u. U. beträchtlich. Wird z. B. der BSR-Mechanismus verwendet, wird ein RP erst mit Ablauf der Gültigkeit seiner von ihm dem Bootstraprouter bekannt gegebenen Zuordnungen aus der Kandidatenliste entfernt und die neue Liste in der Domäne geflutet. Bei standardmäßig minütlicher Wiederholung der Nachrichten dürfte die Gültigkeit im Bereich von Minuten liegen (nicht standardisiert). Solange fällt für betroffene Gruppen die Multicastkommunikation aus.

Anycast-RP [65] macht sich dagegen die schnelle Konvergenzzeit des IGPs zu nutze, indem eine Menge von RPs mit derselben Anycastadresse konfiguriert werden (Anycast-RP Set). Die RPs speisen diese ins IGP ein und machen sie per BSR-Mechanismus allen Routern in der Domäne bekannt. Jeder dieser RPs kennt die Unicastadressen der anderen RPs mit derselben Anycastadresse. Anderen Routern außer den RPs des eigenen Anycast-RP Sets sind die Unicastadressen dagegen unbekannt.

In der vom IGP aufgebauten MRIB ist üblicherweise die Route zum nächstgelegenen RP aus dem Anycast-RP Set enthalten. Je nach seiner Lage in der Domäne verwendet ein Router einen anderen RP aus dem Anycast-RP Set. Damit eine Quelle, deren Pakete per Register-Tunnel einen RP erreichen, auch an jene Empfänger gehen, deren Join an einen anderen RP gerichtet sind, sendet ein RP Kopien der Register-Nachrichten an alle anderen RPs seines Anycast-RP Sets, gerichtet an ihre Unicastadressen. Er tauscht dabei die Adresse des DRs gegen seine eigene Unicastadresse als neue Quelladresse der Register-Nachricht aus. Die anderen RPs leiten das Register nicht mehr weiter, weil sie es von einer Adresse empfangen haben, die nach ihrer Konfiguration einem RP aus dem eigenen Anycast-RP Set gehört. Trotzdem sollte beim Kopieren der TTL/Hop Count von der empfangenen Register-Nachricht übernommen und dekrementiert werden, um die Auswirkung einer durch Fehlkonfiguration entstandenen Weiterleitungsschleife zu minimieren.

Es werden von einem RP nur Register-Nachrichten kopiert und nicht neu generiert. Register-Stop-Nachrichten zwischen den RPs sind entsprechend wirkungslos[28] und können gesendet werden oder auch nicht (implementierungsspezifisch). Register-Nachrichten dienen bei Anycast-RP damit primär zur Lokalisation von Quellen und weniger der Weiterleitung von Datenpaketen im Register-Tunnel. Denn sendet der kopierende RP ein Register-Stop zum DR, beispielsweise weil er quellenspezifischen Routingzustand durch entsprechende Empfänger hat und daher den SPT aufbauen muss, erhält auch kein anderer RP seines Anycast-RP Sets Register-Nachrichten mehr. Die anderen RPs müssen daher zwingend sofort den SPT etablieren (SwitchToSptDesired(S,G) ist immer wahr). Auch ohne Empfänger etabliert ein RP quellenspezifischen Zustand (startet den KeepaliveTimer(S,G)), um bei einem späteren Beitritt eines Empfängers den SPT aufbauen zu können. Dies führt zum Problem, dass im Falle, wenn alle vor dem Register-Stop des ersten RPs zu einem anderen RP kopierten Register-Nachrichten verloren gehen, dieser erst kurz vor Ablauf der Unterdrückungszeit der Register-Nachrichten mit dem Null-Register vom DR eine neue Möglichkeit erhält, von der Existenz der Quelle zu erfahren.

Durch das Weiterleiten des Register-Nachrichten besitzen alle RPs Kenntnis von der Quelle. Fällt ein RP aus, erreichen die Joins einen anderen RP, wobei ein Join bereits innerhalb des J/P_Override_Interval wiederholt wird, wenn sich der Upstreamnachbar geändert hat. Damit kennt nach kurzer Zeit der andere RP die Empfänger des ausgefallenen RPs

[28] Eine Implementierung könnte sie als Bestätigung für den erfolgreichen Empfang der kopierten Register-Nachrichten werten und solange (falls möglich) das Register-Stop zum DR verzögern. Diesbezüglich wurde jedoch nichts standardisiert.

und kann, falls nicht schon aufgrund anderer Empfänger geschehen, die SPTs zu den gewünschten Quellen aufbauen. Die Multicastkommunikation ist wiederhergestellt.

Eingebettete RP-Adressen

Der BSR-Mechanismus erlaubt zwar, innerhalb einer Domäne die von PIM-SM benötigten Zuordnungen von Gruppenadresse auf Rendezvousstellenadresse zu verteilen, für den Interdomäneneinsatz ist der BSR-Mechanismus jedoch ungeeignet. Eine andere Möglichkeit ist, ähnlich Anycast-RP die Quellen in einer Domäne den RPs in anderen Domänen bekannt zu machen, wofür bei IPv4 MSDP (s. folgender Abschnitt) eingesetzt wird. Wegen der Nachteile von MSDP hat man für IPv6 mit den Embedded-RP-Adressen (s. Abschnitt 2.2.4 Abbildung 2.9 auf Seite 27) eine andere Lösung gefunden, die sich für IPv4 wegen der zu kurzen Adressen nicht verwenden lässt. Empfängt ein Router eine Join/Prune-Nachricht für die genannte Gruppe, verwendet er nicht die sonstigen z. B. per BSR-Mechanismus gelernten Zuordnungen von Gruppen- zu RP-Adressen, um den für die Gruppe zuständigen RP zu bestimmen. Stattdessen leitet er die RP-Adresse direkt aus der Gruppenadresse ab. Vorteil neben der internetweiten Skalierbarkeit ist die dadurch inhärent über alle Router konsistente Zuordnung von Gruppen- zu RP-Adresse. Beim BSR-Mechanismus kommt es dagegen während der Änderung der Zuordnungen zu zeitweisen Inkonsistenzen. Nachteilig ist, dass es sehr viele RPs geben kann, was z. B. bei Implementierung des Register-Tunnels als Pseudoschnittstelle zu Skalierbarkeitsproblemen führt, da die maximale Anzahl an Pseudoschnittstellen auf einem Router typischerweise limitiert ist.

MSDP – Multicast Source Discovery Protocol

MSDP ist eine nur für IPv4 spezifizierte Ad-hoc-Lösung für das Problem von PIM-SM, dass der BSR-Mechanismus nicht domänenübergreifend eingesetzt wird. Jede Domäne hat daher ihre eigenen RPs und kann in anderen Domänen liegende Quellen nicht erkennen. Da MSDP nicht als dauerhafte Lösung des Problems angesehen wird, ist es am Ende seiner über dreijährigen versuchten Standardisierung – es gab 20 Draft-Revisionen – mit [68] wieder auf den wesentlichen Mechanismus nur mit Source-Active- und Keepalive-Nachrichten reduziert worden, der hier kurz besprochen werden soll. Bei IPv6 wurde MSDP durch eingebettete RP-Adressen abgelöst.

Für MSDP werden RPs über händisch konfigurierte TCP-Verbindungen beliebig untereinander verbunden, so wie es auch bei BGP üblich ist. Der RP mit der niedrigeren IP-Adresse führt den aktiven TCP-Verbindungsaufbau aus, der mit der höheren wartet passiv auf den eintreffenden Verbindungsaufbau des anderen. Über die TCP-Verbindung werden anschließend die eigentlichen MSDP-Nachrichten ausgetauscht. Gut eine Minute ohne eine MSDP-Nachricht wird als Fehler gewertet, die TCP-Verbindung abgebaut und danach alle halbe Minute ein neuer TCP-Verbindungsaufbau versucht. Um zu verhindern, dass bei fehlenden sonstigen MSDP-Nachrichten eine MSDP-Sitzung versehentlich abgebaut wird, sendet jeder MSDP-Peer nach einer Minute ohne sonstige MSDP-Nachrichten eine *Keepalive-Nachricht*.

Erfährt ein RP von einer neuen Quelle, typischerweise durch Eintreffen einer Register-Nachricht vom DR der Quelle, informiert er die anderen RPs mittels sogenannter *Source-Active-Nachrichten* (SA) über die neue Quelle. Die SA-Nachrichten werden nach dem RPF-Prinzip über die TCP-Verbindungen im gesamten MSDP-Netz geflutet. Um den dazu nötigen RPF-Check durchführen zu können, vermerkt der RP, der die SA-Nachricht ins

MSDP-Netz einspeist, seine eigene IP-Adresse in der Nachricht. Es folgt eine Liste aller aktiven Quellen des betreffenden RPs, jeweils zusammen mit der Gruppenadresse, an die eine Quelle sendet. Alle RPs cachen die empfangenen Informationen. Zum einen kann so ein neuer Nachbar direkt nach Etablierung der TCP-Verbindung mit allen bekannten Quellen versorgt werden. Zum anderen ist jeder Quelle ein Timer zugeordnet, so dass die Anzahl an SA-Nachrichten je Quelle beim Fluten ratenlimitiert werden kann. SA-Nachrichten für eine Quelle werden vom einspeisenden RP periodisch jede Minute erneut geflutet, solange der einspeisende RP quellenspezifischen Zustand für diese Quelle besitzt. MSDP erlaubt vergleichbar IBGP sogenannte Mesh Groups. Diese untereinander über MSDP vollvermaschten RPs leiten wie bei IBGP von anderen RPs der eigenen Mesh Group empfangene MSDP-Nachrichten nur an externe RPs außerhalb der eigenen Mesh Group weiter.

Erhält ein RP ein Join(*,G) für eine Gruppe, so sucht er in seinem MSDP-Cache nach allen in der Gruppe aktiven Quellen und baut zu ihnen einen SPT auf bzw. baut bei Eintreffen einer neuen SA-Nachricht für eine bereits bestehende Gruppe einen SPT für die neue Quelle auf. Zusätzlich bietet die SA-Nachricht für die letzte genannte Quelle die Möglichkeit, ein Paket in der MSDP-Nachricht zu tunneln, das dann von anderen RPs an ggf. vorhandene Empfänger weitergeleitet wird.

MSDP wurde ursprünglich auch innerhalb einer Domäne genutzt, um mehrere RPs zu einem virtuellen Anycast-RP zusammenzuschließen, was aber mittlerweile durch das oben unter Anycast-RP beschriebene viel einfachere Kopieren von PIM-Register-Nachrichten ersetzt wurde.

2.3.2.2 PIM-DM – Protocol Independent Multicast – Dense Mode

PIM-DM [2, 54] ist die moderne, protokollunabhängige Variante von DVMRP. Anders als dieses ist es aber nur für den Intradomäneneinsatz vorgesehen. PIM-DM nutzt wie DVMRP das eingangs erläuterte erste Multicastroutingprinzip Broadcast-and-Prune zum Auffinden der Quellen einer Gruppe, allerdings mit einigen Modifikationen. Im Folgenden steht DVMRP als Stellvertreter für das ursprüngliche Multicastverfahren, wenn auf die Unterschiede hingewiesen wird.

Gemäß Broadcast-and-Prune wird der Verkehr initial gebroadcastet, so dass jeder Router die Quelle kennt. Anders als PIM-SM und BIDIR-PIM ist der Ausgangszustand also weiterleiten. Wie bei Broadcast-and-Prune üblich, kann sich ein abhängiger Router nach einem Prune per Graft wieder an den Verteilbaum hängen. Die *Graft-Nachricht* wird per Unicast direkt an den Upstreamrouter, also der Router auf dem gemäß MRIB kürzesten Pfad Richtung Quelle gesendet. Der Upstreamrouter bestätigt den Empfang mit der ebenfalls per Unicast gesendeten *Graft-Ack-Nachricht* . Graft- und Graft-Ack-Nachricht sind bis auf den anderen Nachrichtentyp mit einer Join/Prune-Nachricht identisch. Die Liste an gepruneten Gruppen/Quellen muss allerdings leer bleiben. Die in der Nachricht vermerkte Gültigkeitsdauer wird nicht benötigt und ignoriert.

Wie bei allen PIM-Protokollen wird die Prune-Nachricht per linklokalem Multicast gesendet und andere Router können diese mit einer Join-Nachricht überschreiben. Dieser bei DVMRP nicht vorhandene Mechanismus zum Überschreiben eines Prunes ist bei PIM-DM nötig, weil ein Upstreamrouter die von ihm abhängigen Router nicht wie bei DVMRP im Rahmen der Topologieerkennung (dort mittels Poisoned Reverse) bestimmen kann. Jeder abhängige Router muss daher selbst dafür sorgen, dass er ein Prune eines anderen Nachbarn ggf. durch

ein eigenes Join überschreibt. Um Stürme von abwechselnden Prune- und Join-Nachrichten zu vermeiden, wird die Rate der Prune-Nachrichten auf einem Router auf typisch eine Prune-Nachricht alle dreieinhalb Minuten begrenzt.

Ebenso fehlt die Möglichkeit, den auf einen Link weiterleitenden Router im Rahmen der Topologieerkennung zu bestimmen. Der Forwarder wird daher mittels Asserts erst in dem Moment bestimmt, in dem mehrere Router Pakete einer Quelle auf den Link weiterleiten. Anders als PIM-SM kennt PIM-DM nur quellenspezifischen Routingzustand und benötigt daher auch nur quellenspezifische Asserts. Ebenso anders als bei PIM-SM tritt ein Assert regelmäßig auf und nicht nur bei inkonsistenter MRIB der Downstreamrouter, da der Ausgangszustand Weiterleiten und nicht wie bei PIM-SM Nichtweiterleiten ist. Jeder Router mit einem Pfad zur Quelle leitet anfänglich auf den Link weiter. Bei zwei oder mehr weiterleitenden Routern kommt es zum Assert. Bei PIM-SM dagegen leitet nur derjenige Upstreamrouter weiter, der die Join-Nachricht von einem Downstream-Router empfangen hat.

Neu gegenüber DVMRP sind die *State-Refresh-Nachrichten*. Statt nach einiger Zeit den Prune-Zustand zu löschen und erneut zu broadcasten, kann mit State-Refresh-Nachrichten der Zustand regelmäßig aufgefrischt und so ein erneutes Broadcasten dauerhaft vermieden werden. Der erste Router hinter der Quelle generiert periodisch jede Minute State-Refresh-Nachrichten, solange auch die Quelle Pakete sendet, und noch einige Zeit (dreieinhalb Minuten) darüber hinaus, um eventuelle Sendepausen der Quelle zu überbrücken. Diese Überbrückung entspricht vom Konzept her dem KeepaliveTimer von PIM-SM. Empfängt ein Router eine State-Refresh-Nachricht vom richtigen Upstreamrouter, erneuert er seinen Prune-Zustand. Zusätzlich sendet er seinerseits eine State-Refresh-Nachricht auf allen anderen Schnittstellen stromabwärts, auf denen er Forwarder ist.

Eine Quelle ohne einen Empfänger in der Domäne führt bei Verwendung von State Refreshes nur zu genau einer State-Refresh-Nachricht je Link je Refresh-Interval (ausgenommen natürlich der Link der Quelle). State Refreshes reduzieren die Signalisierung somit auf die periodische Signalisierung eines gemäß Soft-State-Prinzip arbeitenden Protokolls, vergleichbar der von PIM-SM. Als Nachteil gegenüber PIM-SM bleibt jedoch, dass wegen des Weiterleitens im Grundzustand jeder Router der Domäne Zustand für jede aktive Quelle halten und periodisch signalisieren muss. Bei PIM-SM bleibt er auf die Router auf dem Verteilbaum, also längs der Pfade von der Quelle bzw. der Rendezvousstelle zu den Empfängern, beschränkt. Durch diesen Nachteil ist PIM-DM nicht als Interdomänenroutingprotokoll für das internetweite Multicastrouting geeignet.

2.3.2.3 BIDIR-PIM – Bi-directional Protocol Independent Multicast

Bei BIDIR-PIM [75] ersetzt der Rendezvousstellenlink (Rendezvous Point Link, RPL) die Aufgabe der Rendezvousstelle von PIM-SM. Der RPL wird wie der RP von PIM-SM durch seine Rendezvousstellenadresse (Rendezvous Point Address, RPA) identifiziert. RPL ist derjenige Link, aus dessen Subnetzpräfix die RPA stammt. Die RPA ist also nicht notwendigerweise einem Router zugeordnet. Die Weiterleitung erfolgt im Unterschied zu den beiden anderen Protokollen der PIM-Familie über bidirektionale Bäume. Pakete werden somit über den Baum nicht nur vom RPL zu den Gruppenmitgliedern sondern auch in umgekehrter Richtung von den Quellen zum RPL weitergeleitet. BIDIR-PIM kennt keinen quellenspezifischen Routingzustand und unterstützt somit nur ASM und kein SSM. Derzeit ist BIDIR-PIM ausschließlich für den Intradomäneneinsatz spezifiziert.

BIDIR-PIM bestimmt pro RPA einen bidirektionalen Spannbaum über alle Router in der Domäne, indem für jeden Link der Domäne ein Router als Forwarder gewählt wird, bei BIDIR-PIM Designated Forwarder (DF) genannt. Anders als bei PIM-SM und PIM-DM erfolgt die Bestimmung des DF nicht über Asserts und erst bei Bedarf, sondern mit einem anderen nur von BIDIR-PIM verwendeten Verfahren über *DF-Election-Nachrichten*. Die Wahl erfolgt völlig unabhängig von einer etwaigen Multicastkommunikation nur direkt nach einer Änderung der Route zur RPA oder direkt nach Bekanntwerden einer neuen RPA, z. B. durch den Empfang einer Bootstrap-Nachricht. Allein die Änderung der Zuordnungen von Gruppenadresspräfixen zu bereits bekannten RPAs führt dagegen zu keiner DF-Neuwahl (wohl aber zur Aktualisierung des gruppenunabhängigen Weiterleitungszustands).

Über diese per DF-Wahl etablierten Spannbäume wird der Verkehr einer Quelle je nach Gruppenadresse zu der ihr zugeordneten RPA bzw. deren RPL geleitet. Tritt ein Empfänger der Gruppe bei, sendet der DF auf dessen Link ein Join Richtung RPL, also zum Designated Forwarder des Upstreamlinks (für die der Gruppenadresse zugeordneten RPA). Dieses Joinen setzt sich bis zum Erreichen des RPLs fort. Auf dem RPL selbst gibt es keinen DF mehr. Durch das Join wird entlang des Pfads vom Empfänger zum RPL gruppenspezifischer (*,G)-Weiterleitungszustand etabliert, der allen Verkehr der Gruppe vom RPL zu den Empfängern leitet. Der durch Joins etablierte gruppenspezifische Verteilbaum stellt einen Teilbaum des Spannbaums der für die Gruppe zuständigen RPLs dar. Sobald der Verkehr einer Quelle auf einen Link auf dem Verteilbaum trifft, wird der Verkehr nicht nur Richtung RPL sondern auch in entgegengesetzter Richtung zu den Empfängern weitergeleitet.

Dem Vorteil, in Netzbereichen ohne Empfänger das Weiterleiten Richtung RPL ohne gruppenspezifischen nur mit rendezvousstellenspezifischen Zustand durchführen zu können, steht als Nachteil gegenüber, dass der Verkehr von Quellen unabhängig davon, ob es für ihre Gruppen Empfänger gibt oder nicht, immer bis zum RPL weitergeleitet wird. Es sei angemerkt, dass dieses Weiterleiten auch bei Unicast üblich ist und sich dort wie bei BIDIR-PIM einfach über ein Transport- oder Anwendungsschichtprotokoll vermeiden lässt, das der Quelle Rückmeldungen von den Empfängern gibt, beispielsweise RTCP [158]. Die Quelle kann dann bei leeren Gruppen das Senden einstellen.

2.3.2.4 BSR – Bootstrap Router Mechanism

Der Bootstrap-Router-Mechanismus [20] dient der Herstellung einer domänenweit einheitlichen Zuordnung von Gruppenadresse zu Rendezvousstellenadresse. Der Bootstrap-Router-Mechanismus wurde zusammen mit PIM-SM eingeführt [61]. Mit dem Bootstrap-Router-Mechanismus kann somit zugleich bestimmt werden, welche Gruppenadressen mit welchem der drei Multicastroutingprotokolle der PIM-Familie geroutet werden sollen.

Beim Bootstrap-Router-Mechanismus werden eine Reihe von Routern als Kandidaten für den *Bootstraprouter* konfiguriert. Welcher die Wahl zum Bootstraprouter gewinnt, richtet sich nach der Priorität, die den Kandidaten per Konfiguration zugeordnet wurde. Des Weiteren werden einige Router als Kandidaten für Rendezvousstellen konfiguriert, genauer, als Kandidat für bestimmte Gruppenadresspräfixe, d. h. Bereiche von Gruppenadressen. Es wird nach Präfixen für PIM-SM und BIDIR-PIM unterschieden. Rendezvousstellenkandidaten senden jede Minute ihre konfigurierten Gruppenadresspräfixe, jeweils mit einer ebenfalls konfigurierten Priorität und Gültigkeitsdauer, in einer *Candidate-RP-Advertisement-Nachricht* per Unicast an den gewählten Bootstraprouter.

Der Bootstraprouter sammelt die erhaltenen Zuordnungen und fasst diese zusammen. Dabei ggf. auftretende Widersprüche, wenn eine Gruppe mal als bidirektional und mal

als nicht bekanntgegeben wurde, löst er entsprechend einer lokalen Policy auf. Diese begrenzt ebenso die Anzahl an Zuordnungen je Gruppenadresspräfix anhand ihrer Priorität auf ein sinnvolles Maß (implementierungsspezifisch). Der Bootstraprouter sendet die so gebildete Liste von Zuordnungen periodisch jede Minute in Form einer *Bootstrap-Nachricht*. Sie wird von jedem Router in der gesamten Domäne weitergeleitet, so dass alle Router die Bootstrap-Nachricht empfangen. Die Bootstrap-Nachricht enthält die Zuordnungen sind in Form einer Liste von Gruppenadresspräfixen, denen jeweils eine Liste von RPs, wiederum jeweils mit Gültigkeitsdauer und Priorität aus der entsprechenden Nachricht des Kandidaten, zugeordnet ist.

Jeder Router konfiguriert sich mit den in diesen Bootstrap-Nachrichten enthaltenen Zuordnungen. Die Priorität einer Zuordnung ist dabei ohne Belang. Die für ein Gruppenadresspräfix angegebenen RPs können danach innerhalb ihrer jeweiligen Gültigkeitsdauer als RP für die Gruppenadressen aus diesem Präfix genutzt werden. Zusätzlich gibt der Bootstraprouter mit der ebenfalls in jeder Bootstrap-Nachricht enthaltenen Hashmaskenlänge vor, wie viele führende Bits von einer Gruppenadresse in die Hashwertberechnung zur Auswahl des RPs aus den für eine Gruppe zur Auswahl stehenden RPs eingehen. Dadurch wird eine gewisse Anzahl von Gruppen, üblich sind vier Gruppen (Hashmaskenlänge 30 bit (IPv4) bzw. 126 bit (IPv6)), auf denselben RP gehasht. Hierdurch werden Anwendungen wie Layered Multicast [112], d. h. das Aufteilen eines Datenstroms auf mehrere Gruppen für die Unterstützung einer inhomogenen Gruppe von Empfängern, ermöglicht. Die Auswahl erfolgt wie bei ECMP mittels HRW (s. Abschnitt 2.1.4), wodurch sich das Hinzufügen oder Entfernen eines RPs nur auf 1/N aller aktiven Gruppen auswirkt und damit die Anzahl betroffener Gruppen minimiert.

2.3.3 Historische Multicastroutingprotokolle

2.3.3.1 DVMRP – Distance Vector Multicast Routing Protocol

Das von Deering entwickelte eingangs vorgestellte RPM-Verfahren bildete in Form des Protokolls DVMRP [187] das erste Multicastroutingprotokoll im Internet. Beim Einsatz im internetweiten Forschungsnetz MBone (Multicast Backbone) stellte sich schnell die begrenzte Skalierbarkeit von DVMRP heraus. Es wurde daher mit hierarchischem DVMRP (Hierarchical DVMRP, HDVMRP) [179] eine zweistufige Hierarchie ähnlich der von (M)OSPF vorgeschlagen, wobei HDVMRP auf der oberen Ebene zwischen den Grenzroutern routet und so die (disjunkten) DVMRP-Regionen über die Grenzrouter miteinander koppelt (in den Regionen könnte theoretisch auch ein anderes Protokoll wie z. B. MOSPF laufen). Eine andere vorgeschlagene Alternative ist BGMP, das weiter unten noch erläutert wird. Mit der Entwicklung von PIM-SM wurde jedoch DVMRP und der MBone durch ein auch internetweit skalierendes Multicastroutingprotokoll abgelöst. Die Weiterentwicklung, die DVMRP im Rahmen des MBone erfahren hat, werden derzeit als DVMRP Version 3 in [145, 146] dokumentiert und liegen der folgenden Beschreibung zu Grunde.

Broadcast-and-Prune

DVMRP ist ähnlich wie PIM-DM ein Broadcast-and-Prune-Protokoll. Die Pakete einer Quelle werden initial im gesamten Netz gebroadcastet, damit jeder Router Kenntnis von der Quelle erlangt. Anders als PIM-DM werden jedoch Duplikate auf einem Link durch die automatische Wahl eines Forwarders von vornherein vermieden. Der Router mit der besten Metrik zur Quelle (bei gleichen Metriken gewinnt der Router mit der

niedrigeren Schnittstellenadresse) wird Forwarder. Andere Router am Link, die vom Forwarder abhängen, d. h. deren beste Route zur Quelle über den Forwarder führen, zeigen dies mittels Poisoned Reverse an. Bei DVMRP wird er durch Addition von Unendlich zur vom Forwarder für die Quelle empfangenen Metrik dargestellt. Auf Links ohne Empfänger und ohne abhängige Router oder nur mit abhängigen Routern, die aber alle eine Prune-Nachricht für die Quelle gesendet haben, werden keine Pakete weitergeleitet. Gilt dies für alle Downstream-Schnittstellen eines Routers, sendet er seinerseits seinem Forwarder auf der Upstream-Schnittstelle eine Prune-Nachricht, was den weiteren Empfang von Paketen der Quelle für die in der Prune-Nachricht genannte Dauer unterdrückt, sofern kein anderer abhängiger Router auf dem Upstream-Link existiert, der nicht gepruned hat. Die zu setzende Gültigkeitsdauer berechnet sich als das Minimum der Restgültigkeitsdauer aller von den eigenen abhängigen Routern empfangenen Prune-Nachrichten und dem Standardwert von zwei Stunden (der zur Vermeidung von Synchronisation etwas randomisiert wird). Nach Ablauf der Gültigkeit treffen wieder Pakete der Quelle ein, woraufhin aber erneut gepruned werden kann. Muss vor Ablauf der Gültigkeit wieder auf eine Downstream-Schnittstelle weitergeleitet werden, weil ein Empfänger sich per IGMP oder ein abhängiger Router sich per Poisoned Reverse gemeldet hat, wird mittels Graft-Nachricht an den Forwarder auf der Upstream-Schnittstelle die Unterdrückung vorzeitig aufgehoben. Der Empfang wird mit einer Graft-Ack-Nachricht vom Forwarder bestätigt.

Distance-Vector-Routing

DVMRP baut seine eigene MRIB nach dem Prinzip des Distance-Vektor-Routings auf. Es arbeitet auch hier nach dem Soft-State-Prinzip. Die Report-Nachrichten, mit denen die Routen bekanntgegeben werden, werden periodisch jede Minute wiederholt. Werden innerhalb der Gültigkeitsdauer einer Route von gut zwei Minuten keine weiteren Report-Nachrichten mit dieser Route empfangen, gilt danach das zugehörige Adresspräfix als unerreichbar.

Verglichen mit RIP existieren aber auch eine Reihe von Unterschieden. DVMRP nutzt IGMP statt UDP für seine Routingnachrichten. DVMRP wurde mit dem MBone auch interdomain eingesetzt und unterstützt daher Tunnel, um nicht-DVMRP-fähige Router überbrücken zu können. Damit Routen über Tunnel entsprechend der überbrückten Distanz eine schlechtere Metrik erhalten als direkte physikalische Verbindungen, ordnet DVMRP jeder Schnittstelle genau wie OSPF ein Gewicht zu. Physikalische Schnittstellen tragen typischerweise eins als Gewicht, was der Hop-Count-Metrik von RIP entspricht. Tunnelschnittstellen Werte besitzen Werte größer eins. Der doppelt so große Wert 32 für Unendlich spiegelt ebenfalls den Interdomäneneinsatz im MBone wieder. Der Wert Unendlich selbst gilt nicht als Poisoned Reverse (sondern: Unendlich $<$ Poisoned Reverse $<$ 2·Unendlich), zeigt somit keinen abhängen Router an. Unendlich wird neben seiner Hauptfunktion, ausgefallene Netze bekanntzugeben, daher auch genutzt, um vor dem Herunterfahren eines Routers alle bekanntgegebenen Abhängigkeiten sowie Routen zurückzuziehen, ohne dass Nachbarn bis zum Ende der durch das Soft-State-Prinzip vorgegebenen Gültigkeit warten müssen (Graceful Shutdown).

DVMRP unterstützt Route Hold-down, was in Kombination mit den Flash Updates, der Entsprechung der Triggered Updates von RIP, das lange Count to Infinity in komplexen Topologien vermeidet. Route Hold-down besagt, dass nach Empfang einer Report-Nachricht, die ein Adresspräfix als nicht mehr gültig (Metrik gleich unendlich) markiert, noch für

eine gewisse Zeit (zweifaches Route Report Interval) ein Router seinerseits das Präfix als unerreichbar bekanntgeben muss. Sollte zwischenzeitlich von einem anderen Nachbarn (vom selben Nachbar mit derselben Metrik ausgenommen) das Präfix wieder als erreichbar angezeigt werden – wahrscheinlich, weil dieser noch nicht von der Nichterreichbarkeit erfahren hat –, wird diese Route ignoriert. Durch Route Hold-down propagiert die Nichterreichbarkeit per Flash Update sicher über alle Router auch eines großen Zyklus in der Netztopologie, bevor die Route erneut akzeptiert wird.

DVMRP verwendet mit den Neighbor-Probe-Nachrichten ein dem Hello-Protokoll von OSPF vergleichbares Verfahren. Jeder Router sendet seine Neighbor-Probe-Nachrichten genau wie die Report-Nachrichten periodisch an die linklokale Multicastadresse All-DVMRP-Routers (224.0.0.4). Router an einem Link lernen sich so kennen und stellen sicher, dass bidirektionale Konnektivität zwischen ihnen besteht. Erst wenn ein Router seine eigene Adresse im Neighbor Probe eines Nachbarn sieht, beginnt er mit diesem Multicastroutingnachrichten (Prune, Graft, Graft Ack) auszutauschen. Zudem enthalten Neighbor-Probe-Nachrichten eine Generation ID, die ein Router bei jedem Neustart erhöht. Empfängt ein Router eine größere Generation ID von einem Nachbarn, löscht er allen mit diesem Nachbarn verbundenen Zustand. Ohne Generation ID würde ein Router, der nach Senden einer Prune-Nachricht (ohne Graceful Shutdown) neu startet, bis zum Ende der Gültigkeit des Prunes keine Pakete der Quelle mehr empfangen (bis zu zwei Stunden) und damit auch alle von ihm abhängigen Router (Blackholing). Allerdings erfordert das Inkrementieren, dass die letzte Generation ID permanent auch über einen Reboot gespeichert werden kann, ein Nachteil, den PIM-Protokolle durch Verwenden einer Zufallszahl als Generation ID vermeiden.

2.3.3.2 MOSPF – Multicast Open Shortest Path First

MOSPF [122] nutzt das zuverlässige Fluten von LSAs, um Gruppenmitglieder im gesamten Netz bekanntzugeben. Im einzelnen definiert MOSPF das neue Group-Membership-LSA, mit dem ein Knoten (Link oder Router) im Graphen als Mitglied einer Gruppe bekannt gegeben wird. Auf dem entsprechenden Transitlink befinden sich also Empfänger für die Gruppe bzw. der Router selbst ist Empfänger oder hat auf einem angeschlossenen Stummellink Empfänger. Das Group-Membership-LSA wird innerhalb eines Gebiets geflutet. Ein Gebietsgrenzrouter generiert seinerseits Group-Membership-LSAs in das Hauptgebiet mit sich selbst als Empfänger, wenn sich mindestens in einem der anderen an ihn angrenzenden Gebiete ein Empfänger befindet. Dagegen werden umgekehrt für aus dem Hauptgebiet empfangene Group-Membership-LSAs in den anderen Gebieten keine entsprechenden Group-Membership-LSAs generiert. Stattdessen erhält ein Gebietsgrenzrouter als sogenannter Inter-area Multicast Forwarder den gesamten Multicastverkehr aller Quellen (Wildcard Receiver) aus den Gebieten, in denen er als Inter-area Multicast Forwarder konfiguriert ist (standardmäßig in jedem Gebiet, in dem er Grenzrouter ist). Diesen Verkehr leitet er dann ggf. in das Hauptgebiet weiter, sofern er von dort ein Group-Membership-LSAs für die betreffende Gruppe empfangen hat.

Ein MOSPF-Router berechnet den Pfad für Quellen innerhalb des eigenen Gebiets anders als alle anderen in dieser Arbeit vorgestellten Multicastroutingprotokolle nicht nach dem RPF-Prinzip sondern wie bei Unicast ausgehend von der Quelle den kürzesten Pfad zu den in den Group-Membership-LSAs genannten Knoten mit Gruppenmitgliedern. Befindet sich die Quelle dagegen außerhalb des eigenen Gebiets, wird wie bei den anderen Multicastroutingprotokollen auch in umgekehrter Richtung der kürzeste Pfad von den

Gruppenmitgliedern zur Quelle berechnet. Hier kennt ein MOSPF-Router schließlich nicht mehr die gesamte Topologie, sondern lediglich die Gesamtkosten zu der Quelle, genau wie bei anderen Multicastroutingprotokollen. Die Gesamtkosten sind von den Summary-LSAs bekannt, die die Grenzrouter für alle externen, aber noch AS-internen, Adresspräfixe in ein Gebiet senden. Die Berechnung des letzten Pfadabschnitts vom Grenzrouter, der das Summary-LSA gesendet hat, zu den Knoten mit Empfängern erfolgt ebenfalls mit den Gewichten in umgekehrter Richtung vom Empfänger Richtung Quelle bzw. Grenzrouter. Ziele außerhalb des AS werden mit Summary-LSAs bekanntgegeben, die einen Inter-AS Multicast Forwarder in alle an ihn angeschlossenen Gebiete sendet. Es wird angenommen, dass ein solcher Router auch am Interdomänenmulticastrouting teilnimmt. MOSPF ist wie OSPF aufgrund des Flutens von LSAs nicht für das Interdomänenrouting geeignet.

Die Pfadberechnung mittels Dijkstra-Algorithmus ist relativ aufwendig, so dass sie nur bei Bedarf durchgeführt wird, und zwar beim Eintreffen des ersten Pakets einer Quelle. Dies hat zudem den Vorteil, dass ein Router, der nicht auf dem Verteilbaum liegt, nie die Berechnung ausführen wird, an deren Ende er ohnehin nur feststellen würde, dass er nicht auf dem Verteilbaum liegt und keinen Weiterleitungszustand einrichten muss. Das Ergebnis der Berechnung wird gecacht und bleibt im Prinzip bis zu einer Topologieänderung gültig.

MOSPF ist abwärtskompatibel zu OSPF, d. h. MOSPF und OSPF-Router können in einem AS gemischt werden. MOSPF-Router erkennen herkömmliche OSPF-Router und ziehen dediziert nur MOSPF-Knoten in die Pfadberechnung mit ein. MOSPF routet als Erweiterung zu OSPF Version 2 nur IPv4-Multicast, eine entsprechende Multicasterweiterung von OSPF Version 3 für IPv6 wurde nicht spezifiziert. Des Weiteren stammt MOSPF aus der Zeit vor Entwicklung von SSM. Entsprechend enthält das Group-Membership-LSA nur die Gruppenadresse und keine Quelladresse, so dass mit MOSPF kein Source Specific Multicast möglich ist.

2.3.3.3 CBT – Core Base Trees

Mit CBT [13] wurden eine Reihe von innovativen Konzepten eingeführt: Zum einen das des Kerns (Core), bei dem sich Quellen und Empfänger anmelden und dass sich bei PIM-SM und BIDIR-PIM in Form der Rendezvousstelle bzw. des Rendezvousstellenlinks wiederfindet. Zum anderen, dass die MRIB nicht selbst bestimmt sondern durch ein dediziertes (beliebiges) Unicastroutingprotokoll aufgebaut wird. Ein Konzept, dass allen nachfolgenden Protokollen der PIM-Familie zugrunde liegt und für diese Protokollfamilie namensgebend war. Und schließlich die bidirektionalen Bäume, die für Gruppen mit vielen Quellen erheblich besser skalieren als die quellenspezifischen Bäume von DVMRP, MOSPF, PIM-SM und PIM-DM [23]. Das Konzept der bidirektionalen Bäume ist später in BIDIR-PIM aufgegangen. CBT war für die Entwicklung des Multicastroutings sehr wichtig, das Protokoll [12, 11] selbst hat aber nie Verbreitung gefunden, da die Ideen schnell in PIM-SM und später BIDIR-PIM aufgegangen sind und diese anders als das CBT-Protokoll auch IPv6 unterstützen. Im Unterschied zu CBT verzichtet BIDIR-PIM durch seinen neben den Verteilbäumen etablierten Spannbaum auf die separate Behandlung von Quellen, die sich nicht auf einem Link mit Gruppenmitgliedern befinden. Bei CBT werden die Pakete einer nicht auf dem Baum liegenden Quelle noch wie bei PIM-SM vom First Hop Router, der Entsprechung zum Designated Router von PIM-SM, per IP-in-IP-Tunnel, statt Register-Tunnel bei PIM-SM, zum Kern getunnelt.

2.3.3.4 BGMP – Border Gateway Multicast Protocol

BGMP [170] ist eine Erweiterung von BGP für das Multicastrouting. BGMP basiert auf der Idee, die Trennung in Exterior und Interior Gateway Protocol beim Unicastrouting auch auf das Multicastrouting zu übertragen. Die bisher vorgestellten Multicastroutingprotokolle würden danach als Multicast-IGP (MIGP) lediglich zum Intradomänenmulticastrouting eingesetzt werden. BGMP als einziges Interdomänenmulticastroutingprotokoll verbindet die durch die MIGPs aufgebauten Verteilbäume dann interdomain. Die Kopplung zwischen BGMP und MIGP findet auf den Grenzroutern des AS statt, die sowohl an BGMP als auch am jeweiligen MIGP ihres Autonomen Systems teilnehmen. Wie die Kopplung erfolgt, ist in den Interoperabilitätsregeln [169] spezifiziert. Die Interoperabilitätsregeln beschreiben eine bijektive Abbildung zwischen dem Routingzustand eines Protokolls und einem abstrakten Modell. Der Zustand von einem Protokoll wird dann über dieses abstrakte Modell und weiter in einen gleichwertigen Zustand eines anderen Protokolls überführt. Hierdurch ist nur ein Regelsatz je Routingprotokoll nötig statt $n(n-1)/2$ bei direktem Überführen zwischen n Routingprotokollen.

BGMP betrachtet jedes AS als einen Knoten. Es baut für jede Gruppe einen bidirektionalen Baum mit dem AS als Wurzel auf, dem die Gruppenadresse gehört. BGMP setzt daher eine bekannte Abbildung Gruppenadresse auf AS voraus, wie sie bei GLOP- oder unicastpräfixbasierten Gruppenadressen gegeben ist. Ein AS, das nur Quellen aber keine Empfänger für die Gruppe hat, ist nicht Teil des Baums. Pakete der Quelle in einem solchen AS werden einfach Richtung des für die Gruppe zuständigen Wurzel-AS weitergeleitet. Treffen sie auf ein Grenzrouter eines AS im Baum, werden die Pakete dann über diesen weiterverteilt.

Viele MIGPs verwenden quellenspezifische Bäume. Diese Protokolle erwarten nach dem RPF-Prinzip die Pakete einer Quelle aus Richtung des kürzesten Pfads zur Quelle. Entsprechend muss genau dieser Grenzrouter die Pakete einer AS-externen Quelle in das MIGP einspeisen. BGMP verwendet jedoch bidirektionale Bäume, so dass die Pakete der Quelle über einen anderen Grenzrouter eintreffen können. Die Pakete werden dann von BGMP zum erwarteten Grenzrouter getunnelt. Um den durch das Tunneln entstehenden Overhead einzusparen, erlaubt BGMP in diesen Fall auch einen quellenspezifischen BGMP-Baum aufzubauen.

BGMP und die Interoperabilitätsregeln sind nie über das Konzeptstadium hinausgekommen, es sind keine Implementierungen bekannt.

2.4 Dienstgüte

Das heutige Internet arbeitet wie bereits zu seinen Anfängen in den 1970er Jahren nach dem so genannten Best-Effort-Servicemodell. Dies bedeutet, dass alle Pakete entsprechend der augenblicklich verfügbaren Ressourcen (Bandbreite, Pufferspeicher) bestmöglich, also mit möglichst geringer Verzögerung und mit möglichst geringem Paketverlust, weitergeleitet werden. Wenn der Bedarf größer als die verfügbaren Ressourcen ist, sorgt die Staukontrolle von TCP dabei ansatzweise für eine faire Aufteilung der Ressourcen unter den konkurrierenden Datenströmen. Dienstgüte (Quality of Service, QoS) im Internet wird dagegen durch die kontrollierte Ungleichbehandlung von Datenströmen erreicht.

Zum Zweck der Ungleichbehandlung enthält der Kopf des IP-Pakets das Feld Type of Service (TOS, vgl. auch Abbildung 2.3 auf Seite 6). Ursprünglich wurde mit ihm das

IP-Paket einer von acht Prioritätsstufen (Precedence) zugeordnet und die gewünschte Dienstgüte durch Setzen eines von drei Flags, geringe Verzögerung, hoher Durchsatz und hohe Zuverlässigkeit, spezifiziert [139]. Weitverbreitete Verwendung haben jedoch nur die Prioritätsstufen und von diesen auch nur die beiden höchsten gefunden. Sie wurden zur Absicherung des Verkehrs von Routingprotokollen gegen den übrigen Best-Effort-Verkehr (niedrigste Prioritätsstufe) eingesetzt [126], denn Pakete mit höherer Priorität werden in Stausituationen mit geringerer Wahrscheinlichkeit verworfen.

Ende der 1990er wurden mit den Differentiated Services (Diffserv) [25] die Prioritätsstufen und Flags des TOS-Felds durch den Diffserv Codepoint ersetzt [126], dem anders als beim ursprünglichen TOS keine bestimmten Dienstgüteparameter zugeordnet sind, sondern der lediglich ein konfigurierbares Weiterleitungsverhalten selektiert. Diffserv ist nicht der direkte Nachfolger des ursprünglichen Type of Service, sondern löst die Mitte der 1990er Jahre entwickelten Integrated Services (Intserv) [32] mit ihrem Reservierungsprotokoll RSVP [33] ab. Intserv wurde wegen der aufwendigen per-flow Behandlung in der Weiterleitungsebene als nicht hinreichend skalierbar für das Internet angesehen. Abschnitt 2.4.1 erläutert Diffserv im Detail.

Signalisierung

Diffserv betrachtet allein die Weiterleitungsebene. Es fehlt damit ein standardisiertes Protokoll zur Dienstgütesignalisierung, wie es Intserv mit dem Protokoll RSVP besitzt. Der Einsatz von Diffserv beschränkt sich infolgedessen noch weitgehend auf den Intradomänenfall und auf statische Konfigurationen zwischen benachbarten Diffservdomänen. Die im Internet interdomain bereitgestellte und ubiquitär Ende zu Ende verfügbare Dienstgüte ist dagegen weiterhin lediglich Best Effort, was sich ohne ein für Diffserv standardisiertes Signalisierungsprotokoll nicht ändern wird.

Im Rahmen der IETF-Arbeitsgruppe Next Steps in Signaling (NSIS) wird daher eine flexible Signalisierungsprotokollsuite standardisiert [74]. Zum jetzigen Zeitpunkt sind aber noch keine Protokolle verabschiedet worden. Vorgesehen ist eine dreischichtige Architektur mit einer Signalisierungstransportschicht in Form des Protokolls GIST (General Internet Signaling Transport) [159], das seinerseits auf die verfügbaren Transportschichtprotokolle aufsetzt. Derzeit sieht GIST die Verwendung von TCP, UDP und SCTP vor. Auf GIST setzen wiederum die eigentlichen Signalisierungsprotokolle auf. Neben einem Signalisierungsprotokoll zum Firewall/NAT-Traversal (Network Address Translation) [167] ist zur Dienstgütesignalisierung das Protokoll QoS-NSLP (NSIS Signaling Layer Protocol) [9] in der Entwicklung. Verkehrs- und Dienstgüteparameter werden bei QoS-NSLP in der sogenannten QSPEC (QoS Specification) gekapselt, wodurch QoS-NSLP unabhängig vom zugrundeliegenden Dienstgütemodell wie z. B. Diffserv oder Intserv wird. Was die QSPEC enthalten muss, muss für jedes Dienstgütemodell separat standardisiert werden. Um die Interoperabilität verschiedener Dienstgütemodelle zu ermöglichen, wird mit [9] ein einheitliches Format der QSPEC und der Repräsentation gebräuchlicher Verkehrs- und Dienstgüteparameter angestrebt (Rate, Verzögerung, Verlustwahrscheinlichkeit, etc.). Damit die neue Signalisierungsprotokollsuite nicht in ähnliche Skalierbarkeitsprobleme wie RSVP läuft, wird die Aggregation von Reservierungen von Anfang an mit berücksichtigt. Hierdurch lässt sich Per-Flow-Zustand im Signalisierungsprotokoll zumindest in einem Teil der Router vermeiden.

2.4.1 Diffserv – Differentiated Services

Skalierbarkeit durch Aggregation

Die Differentiated-Services-Architektur [25, 29] ist die aktuelle Dienstgütearchitektur im Internet. Primäres Ziel beim Design von Diffserv war es, die Paketweiterleitung so einfach und damit skalierbar wie möglich zu halten [128]. Diffserv erreicht seine Skalierbarkeit dadurch, dass alle Pakete mit demselben Diffserv Codepoint (DSCP) aggregiert behandelt werden. Alle auf einem Link in dieselbe Richtung fließende Pakete mit demselben DSCP bilden hierdurch ein sogenanntes Verhaltensaggregat (Behavior Aggregate). Sie erfahren bzgl. Diffserv dieselbe Behandlung und damit dieselbe Dienstgüte. Diffserv benötigt somit nur noch Weiterleitungszustand je Aggregat anstatt pro Datenstrom.

Ein Router ordnet Pakete anhand ihres DSCPs einem bestimmten Weiterleitungsverhalten, dem Per-Hop Behavior (PHB), zu. Unterschiedliche DSCPs können dabei dasselbe PHB selektieren. Ein PHB beschreibt das von außen sichtbare Weiterleitungsverhalten eines Routers hinsichtlich der Dienstgüteparameter wie maximale Verzögerung (Delay), Verzögerungsschwankung (Jitter), Paketverlustwahrscheinlichkeit (Loss) etc. Es bestimmt, welche Art von Dienstgüte Pakete eines gewissen Verhaltensaggregats erfahren und damit letztlich welcher Dienst für die enthaltenen Datenströme erbracht wird. Die differenzierte Weiterleitung gemäß DSCP ist die grundlegende Funktion, die jeder Diffserv-fähige Router durchführen muss.

Für Router im Inneren eines Netzes bzw. einer Diffservdomäne (s. u.) ist dies sogar die einzige durchzuführende Funktion. Denn Diffserv verlagert Komplexität so weit wie möglich an den Rand eines Netzes. So muss im Rahmen der Diffservarchitektur lediglich der erste Router nach der Quelle, der First-Hop-Router (FHR), für jeden einzelnen Datenstrom Zustand halten (zusätzlich zum Zustand pro Verhaltensaggregat). Da über diesen Router jedoch nur vergleichsweise wenige Datenströme fließen, stellt dies kein Skalierbarkeitsproblem dar. Bei Intserv muss dagegen jeder Router entlang des Pfads Per-Flow-Zustand halten, auch stark belastete Router im Kernnetz. Für Router im Inneren einer Diffservdomäne ist der Zustand durch maximal 64 Verhaltensaggregate je Schnittstelle je Richtung begrenzt. Typischerweise sind es deutlich weniger, da nicht alle 64 möglichen DSCPs genutzt werden und häufig nur an den Ausgangsschnittstellen differenziert behandelt wird.

Des Weiteren findet auf Routern im Inneren kein Policing (s. u.) statt. Dies ist ausschließlich Aufgabe der Grenzrouter (Border Router, BR) am Rand einer Diffservdomäne. Das Policing stellt die Einhaltung der vertraglichen Vereinbarungen mit der Nachbardomäne sicher. Aber auch auf den Grenzroutern findet die Überprüfung nicht per-flow statt wie beim First-Hop-Router, sondern lediglich pro Verhaltensaggregat. Diffserv verlagert damit das aufwendige Policing an den weniger belasteten Rand einer Domäne und den die Skalierbarkeit begrenzenden Per-Flow-Zustand auf den First-Hop-Router. Dieses Prinzip „Komplexität an den Rand“ wird manchmal auch als Diffserv-Prinzip bezeichnet. Es sei angemerkt, dass die Diffservarchitektur zwar grundsätzlich die Per-Flow-Behandlung auch durch innere Router und Grenzrouter zulässt. Das Resultat wäre jedoch dieselbe schlechte Skalierbarkeit wie die von Intserv, ohne aber dessen exakt definierte Ende-zu-Ende-Dienstgüte zu erreichen, die Intserv mit Hilfe von RSVP und durchgängiger Per-Flow-Behandlung ermöglicht.

Markierung, Klassifizierung, Policing

Der DSCP ist in den ersten (höchstwertigen) sechs Bits des TOS-Felds im IP-Kopf abgelegt [126].[29] Die 64 möglichen DSCPs werden aus drei Pools vergeben, von denen der zweite mit 16 DSCPs für experimentelle Zwecke bzw. domänenlokale Verwendung vorgesehen ist. Das in Kapitel 4 entworfene Verfahren nutzt zwei DSCPs aus diesem zweiten Pool. Derzeit sind 21 der 32 DSCPs aus dem für standardisierte DCSPs vorgesehenen Pool 1 vergeben. Pool 3 hält 16 weitere DSCP als Reserve bereit, sollte Pool 1 einmal ausgehen [89].

Pakete können bereits von der Quelle mit einem DSCP markiert werden oder werden erst durch den First-Hop-Router anhand eines konfigurierten Profils initial mit einem DSCP markiert. Auf jeden Fall überprüft der FHR den eingehenden Verkehr einer Quelle auf seine Zulässigkeit und Konformität hinsichtlich der konfigurierten Profile (Policing). So stellt der FHR sicher, dass nur berechtigte Quellen den mit einem DSCP implizierten Dienst nutzen können und dies auch nur, wenn für ihren Verkehr ausreichend freie Ressourcen entlang des weiteren Pfads zur Verfügung stehen (Zugangskontrolle). Zum anderen stellt der FHR für zugelassenen Verkehr sicher, dass der von der Quelle gesendete Datenstrom den vereinbarten Verkehrsparametern entspricht (Verkehrskontrolle). Die Profile werden entweder statisch konfiguriert oder dynamisch durch das Dienstgütemanagement (s. Abschnitt 2.4.2) verwaltet. Zwecks Zuordnung des Verkehrs zu den Profilen klassifiziert der FHR eingehende Pakete mit einem Mehrfeldklassifizierer [19] anhand verschiedener Felder der Pakets. Typische Felder sind die Quell- und Zieladresse, das IPv6 Flow Label, soweit möglich (z. B. nicht verschlüsselt) Transportschichtprotokoll, Ports etc. Das Policing durch den FHR erfolgt also bis hinunter auf einzelne Datenströme. Der so klassifizierte Verkehr wird gemessen und mit den im Profil hinterlegten Vorgaben verglichen. Je nachdem wird der Verkehr vom FHR (um)markiert, geformt und/oder verworfen, so dass der den FHR verlassende Verkehr konform zu den konfigurierten Profilen ist. Dies wird als Verkehrskonditionierung (Traffic Conditioning) bezeichnet. Die Per-Flow-Behandlung verhindert, dass der Verkehr von einzelnen sich fehlverhaltenden Quellen negativen Einfluss auf den Verkehr der übrigen sich konform verhaltenden Quellen hat. Verkehrskonditionierung erfolgt typischerweise mit Token und Leaky Buckets, deren Parameter vom Profil vorgegeben werden. Verschiedene gebräuchliche Konstellationen von Token Buckets finden sich in [79, 80, 64, 31, 1].

Muss Verkehr aufgrund des Routings getunnelt werden, d. h. ein Paket wird in ein anderes Paket eingekapselt, sollte sich durch den Tunnel bei einer vollständig Diffserv-konformen Implementierung hinsichtlich der erfahrenen Dienstgüte dasselbe Verhalten ergeben, als wären die Pakete nicht getunnelt worden. Dies lässt sich wie folgt erreichen: Am Ingressrouter des Tunnels wird der DSCP der zu tunnelnden Pakete für die (äußeren) Tunnelpakete übernommen. Die Tunnelpakete werden danach wie normale Pakete anhand ihres (äußeren) DSCPs einem Verhaltensaggregat zugeordnet und entsprechend behandelt. Am Egressrouter des Tunnels wird ein ggf. durch Ummarkieren, z. B. an Domänengrenzen, veränderter DSCP des Tunnelpakets vor dem Weiterleiten in das entkapselte getunnelte Paket zurückgeschrieben [24].

Diffservdomänen

Diffserv unterteilt das Netz in Diffservdomänen oder kurz Domänen, was die Gliederung des Internets in Autonome Systeme widerspiegelt. Eine Diffservdomäne ist eine zusammen-

[29] Die letzten beiden Bits des TOS-Felds dienen der expliziten Anzeige von Stausituationen (Explicit Congestion Notification, ECN) [149] und werden von TCP und DCCP genutzt.

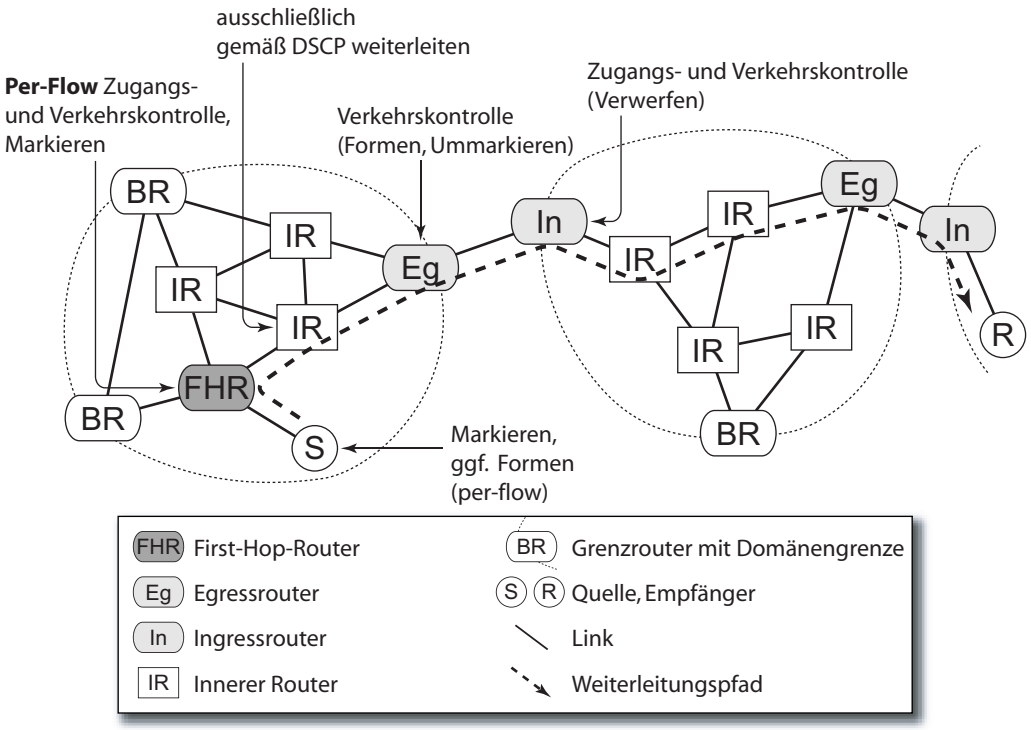


Abbildung 2.14 In der Diffserv-Architektur übernehmen Router unterschiedliche Rollen mit jeweils anderen Aufgaben.

hänge Menge von Diffserv-fähigen Routern mit einem einheitlichen Satz an Richtlinien zur Diensterbringung (Service Provisioning Policy) und PHBs. An den Domänengrenzen überprüfen die Grenzrouter die Verhaltensaggregate hinsichtlich der Verkehrsbeeinflussungsvereinbarung (Traffic Conditioning Agreement, TCA). Sie ist Teil des zwischen zwei benachbarten Domänen geschlossenen Dienstvertrags (Service Level Agreement, SLA). Genauer gesagt überprüfen die Grenzrouter den Verkehr hinsichtlich des technischen Teils des TCAs, der als Verkehrsbeeinflussungsspezifikation (Traffic Conditioning Specification, TCS) bezeichnet wird. Dieser wiederum ist integraler Bestandteil der Dienstspezifikation (Service Level Specification, SLS), den technischen Aspekten des SLAs [70]. Die TCS bestimmen damit letztlich jene Profile, mit denen die Grenzrouter statisch oder vom Dienstgütemanagement konfiguriert werden und die das Policing vorgeben. Am Egressrouter, dem letzten Router vor der stromabwärts liegenden Domäne, muss durch Formen, Ummarkieren und notfalls auch durch Verwerfen der Verkehr konform zur TCS gemacht werden. Ist dies einem Egressrouter ohne Verwerfen nicht möglich, bedeutet dies, dass die Zugangskontrolle der eigenen Domäne zu viel Verkehr in die Domäne lässt und die Profile auf den Ingress- und First-Hop-Routern korrigiert werden müssen. Erreicht nicht zur TCS konformer Verkehr den Ingressrouter der Domäne stromabwärts, kann er ihn einfach verwerfen – und sollte dies auch, um seine eigene Domäne vor Überlast zu schützen. Zusätzlich führen Ingressrouter eine Zugangskontrolle durch und überprüfen, ob die Nachbardomäne stromaufwärts zur Nutzung der DSCPs und zugehörigen Dienste berechtigt ist.

Das von außen sichtbare Verhalten einer Domäne vom Ingress- bis zum Egressrouter wird als Per-Domain Behavior (PDB) [127] bezeichnet und folgt aus dem zu seiner Umsetzung verwendeten PHB und den Verkehrsbeeinflussungsmaßnahmen an den Domänengrenzen

bzw. am FHR. Beispiele für PDBs sind das Virtual Wire PDB [129] zur Emulation einer virtuellen Leitung, das Quick Forwarding PDB [104] zur Minimierung der Latenz von stark burstartigem transaktionsorientiertem Verkehr (Datenbanktransaktionen, Remote Procedure Calls, DNS-Abfragen etc.) oder das zur Separierung von Hintergrundverkehr (nicht interaktiver Verkehr z. B. von Backups, Softwareverteilung oder Tauschbörsen) gedachte Lower Effort PDB [26].

Abbildung 2.14 veranschaulicht noch einmal die verschiedenen Rollen, die Router im Rahmen der Diffserv-Architektur übernehmen sowie die typischerweise in jeder Rolle übernommenen Aufgaben. Der Weiterleitungspfad von der Quelle S bis zum Empfänger R verläuft über drei Domänen hinweg (gestrichelt umrandete Zonen, dritte Domäne nur ausschnittsweise dargestellt). Grenzrouter (BR) sind meist beides, Ingress- (In) wie Egressrouter (Eg). Bezogen auf ein bestimmtes Verhaltensaggregat sind sie aber nur eines von beiden, je nachdem, ob der Verkehr des betrachteten Verhaltensaggregats in ihre Domäne hinein oder aus ihr heraus fließt. In beiden Rollen, als Ingress- wie Egressrouter, überprüfen Grenzrouter den Verkehr mit denselben Profilen, unterscheiden sich aber bzgl. der durchgeführten Aktionen. So versucht nur ein Egressrouter, durch das vergleichsweise aufwendige Formen Verkehr konform zur Verkehrskonditionierungsspezifikation zu machen. Alle Router leiten Pakete gemäß ihres DSCPs weiter. Für die Router im Inneren einer Domäne (IR) ist dies die einzige Aufgabe. Ingress- und First-Hop-Router (FHR) führen eine Zugangskontrolle durch, ersterer pro Verhaltensaggregat, letzterer per-flow. Sowohl die Quelle als auch der First-Hop-Router kann Pakete initial markieren. Das Markieren durch die Quelle ist z. B. nötig, wenn Pakete desselben Datenstroms je nach Inhalt verschiedene DSCPs nutzen sollen. In jedem Fall überprüft der First-Hop-Router die Markierung der Quelle im Rahmen seiner Zugangs- und Verkehrskontrolle.

Abschließend sollen alle bisher standardisierten PHBs kurz vorgestellt werden, bevor im nächsten Abschnitt auf mögliche Formen eines Dienstgütemanagements für Diffserv eingegangen wird.

2.4.1.1 Class Selector PHB Group

Die acht Class Selector Codepoints (CS) [126] wurden zum Zwecke der Kompatibilität mit dem ursprünglichen Type-of-Service spezifiziert. Sie entsprechen vom numerischen Wert her den acht Prioritätsstufen (Precedence) vom TOS-Feld, wenn keines der Flags Verzögerung, hoher Durchsatz und hohe Zuverlässigkeit gesetzt ist. Es wird gefordert, dass ein höherer CS nicht zu einer geringeren Wahrscheinlichkeit für eine zügige Weiterleitung führen sollte (nicht muss!). Zusätzlich werden zwingend mindestens zwei Weiterleitungsklassen gefordert, wobei die beiden höchsten CS7 und CS6 (entspricht bei TOS den Prioritätsstufen Network Control bzw. Internetwork Control) eine bevorzugte Behandlung erfahren müssen verglichen mit CS0. Durch CS0 wird typischerweise das Best Effort (BE) „PHB“ bzw. das Default PHB in einer Diffservdomäne selektiert. Hierdurch wird dem bei TOS üblichen Gebrauch entsprochen, der genau diese – und in der Praxis nur diese – Unterscheidung machte.

2.4.1.2 Expedited Forwarding PHB

Das Expedited Forwarding (EF) PHB [47, 39, 8] bestimmt, dass eingehende EF-Pakete mindestens mit der für das EF-Verhaltensaggregat konfigurierten Rate bedient werden. Dazu definiert EF den idealen Absendezeitpunkt als den Zeitpunkt, an dem das letzte Bit

eines Pakets die Ausgangsschnittstelle des Routers verlassen würde, wäre die physikalische Ausgangsschnittstellengeschwindigkeit gleich der konfigurierten Rate des Aggregats. Ein Fehlerterm beschreibt die durch Scheduling, routerinterne Weiterleitung und Pufferung entstehende maximale Abweichung vom Ideal und muss vom Routerhersteller angegeben werden.

Am einfachsten implementieren lässt sich EF mit einem Prioritätsscheduler, der die Warteschlange für EF-Pakete mit höchster Priorität bedient. Dies führt zudem zu den geringstmöglichen Weiterleitungsverzögerungen, hat jedoch den Nachteil, dass sich durch Aggregation von Verkehr mehrerer Eingänge auf einen Ausgang (kleine) Bursts bilden können. Ebenso möglich ist Weighted Fair Queueing (WFQ), bei dem der Anteil für EF mindestens dem Anteil der für EF konfigurierten Rate an der Schnittstellenrate des Ausgangs entsprechen muss.

Entscheidend für die Qualität von EF ist striktes Policing am FHR bzw. Ingressrouter, so dass die Rate des eingehenden Verkehrs nie größer als die konfigurierte Rate wird. Nur so wird sichergestellt, dass die Warteschlangen kurz und damit die Weiterleitungsverzögerung klein bleibt. Dies bedeutet aber auch, dass Verkehr einer Quelle auch bei kurzzeitigem Überschreiten der zugesicherten Rate vom FHR sofort verworfen werden muss. Oder der Verkehr muss geglättet werden – mit der daraus resultierenden größeren Verzögerung. EF ist daher nicht für Verkehr mit variabler Rate gedacht. Dieser kann EF nur unter Inkaufnahme schlechter Ressourcenausnutzung durch massives Overprovisioning nutzen. Hierbei wird EF mit der maximal auftretenden Rate der Verkehrs (Spitzenrate, Peak Rate) konfiguriert.

2.4.1.3 Assured Forwarding PHB Group

Die Assured Forwarding (AF) PHB Group [78] definiert mehrere Klassen (Class) AFx, x=1...n, wobei jede Klasse jeweils mehrere DSCPs mit unterschiedlichen Verwurfsprioritäten (Drop Precedence) AFxy, y=1...m, zusammenfasst. Standardisiert wurden DSCPs für n=4 Klassen mit jeweils m=3 Prioritäten. AF fordert, dass kurzzeitige Überlast durch Zwischenspeichern von Paketen gehandhabt und langfristige Überlast durch Verwerfen begegnet werden muss. Pakete mit größerer Priorität müssen dabei zwingend mit der gleichen oder einer höheren Wahrscheinlichkeit verworfen werden als Pakete mit niedrigerer Priorität. Werden den Prioritäten einer Klasse nur zwei verschiedene Verwurfswahrscheinlichkeiten zugeordnet, muss die kleinste Priorität 1 eine andere (geringere) Verwurfswahrscheinlichkeit ergeben als die übrigen Prioritäten. Zwischen unterschiedlichen Klassen besteht keine Beziehung hinsichtlich der Verwurfswahrscheinlichkeit. Pakete unterschiedlicher Klassen werden bei der Weiterleitung völlig unabhängig voneinander behandelt.

Implementieren lassen sich unterschiedliche Verwurfswahrscheinlichkeiten z. B. durch Random Early Detect (RED) mit verschiedenen Schwellenwerten für unterschiedliche Prioritäten. Pakete einer Klasse dürfen nicht verschiedenen Warteschlangen zugeordnet werden, da so Paketvertauschungen entstehen können, die grundsätzlich vermieden werden sollten (vgl. Fußnote 15 zu ECMP auf Seite 18). Unterschiedliche Klassen werden dagegen sehr wohl unterschiedlichen Warteschlangen zugeordnet, so dass Pakete aus ein- und demselben Datenstrom keine DSCPs unterschiedlicher Klassen nutzen sollten.

AF eignet sich gut für Verkehr mit variabler Datenrate, da kurzzeitige Überlastsituationen anders als bei EF toleriert werden. Durch die Pufferung entstehen jedoch schwankende und im Mittel deutlich größere Verzögerungen als bei EF. Beschränkt man den Zugang

zu einer AF-Klasse auf die ihr im Mittel zugeteilten Ressourcen, lässt sich damit der von Intserv her bekannte Dienst Controlled Load [191] bereitstellen. Controlled Load besagt, dass die erfahrene Dienstgüte hinsichtlich Paketverlust und Verzögerung der entspricht, die ein bezüglich seiner Verkehrsparameter vergleichbarer Datenstrom bei Weiterleitung als Best Effort in einem nicht überlasteten Netz erfahren würde.

2.4.1.4 Lower Effort PDB

Das Lower Effort (LE) PDB [26] wurde ursprünglich als PHB, das Limited Effort PHB, entwickelt, weshalb es hier mit aufgeführt werden soll. Das LE-Verhaltensaggregat nutzt neben einem zugesicherten Minimum an Bandbreite übrige Bandbreite nur dann, wenn diese von keinem anderen Verhaltensaggregat benötigt wird. Wegen der geforderten minimalen Bandbreite lässt sich Prioritätsscheduling nur dann für LE verwenden, wenn alle Verhaltensaggregate mit höherer Priorität (inklusive Best Effort!) durch Policing in ihrer Bandbreite auf eine Summe kleiner der physikalisch verfügbaren Bandbreite beschränkt sind. Eine Umsetzung mittels Class Based Queueing (CBQ) mit der Erlaubnis zum Borgen freier Bandbreite oder WFQ ist daher sinnvoller.

2.4.2 Dienstgütemanagement für Diffserv

Aufgabe des Dienstgütemanagements ist es, die Zugangs- und Verkehrskontrolle so zu steuern, dass die geforderte Dienstgüte erbracht wird. Hierzu verwaltet es die verfügbaren Ressourcen und konfiguriert die Profile für das Policing. Die geforderte Dienstgüte erfährt das Dienstgütemanagement über ein Protokoll zur Dienstgütesignalisierung.

Dienstgütemanagement lässt sich verteilt oder zentral bewerkstelligen. Im Falle verteilten Dienstgütemanagements nimmt jeder Router die Verwaltung der eigenen Ressourcen bzw. der an ihn angrenzenden (Ausgangs-)Links selbst vor. Beim zentralen Dienstgütemanagement wird einem ausgezeichneten Knoten diese Aufgabe übertragen. Die Ressourcenpartitionierung, also die Verteilung der Router- und Linkressourcen auf die in einer Diffservdomäne implementierten PHBs bzw. Verhaltensaggregate, ist typischerweise statisch konfiguriert. Da Router im Inneren einer Diffservdomäne zudem kein Policing durchführen, ist bei Ihnen keinerlei dynamische Konfiguration nötig. Das Dienstgütemanagement beschränkt sich bei inneren Routern auf die Erfassung der genutzten Ressourcen. Sie ist die Basis für zukünftige Zugangskontrollentscheidungen. Ein verteiltes Dienstgütemanagement mit einer Managementinstanz auf jedem Router, wie es Intserv vorsieht, ist für Diffserv daher unnötig aufwendig. Tatsächlich wurde bei der Entwicklung der Diffservarchitektur ein zentrales Dienstgütemanagement vorgeschlagen [128]. Letztlich wurde aber die Frage des Dienstgütemanagements bei der Standardisierung von Diffserv als zu komplex ausgeklammert, so dass es jedem Netzbetreiber selbst überlassen bleibt, wie er sein Dienstgütemanagement umsetzt.

Nachfolgend soll das Dienstgütemanagement für Diffserv etwas genauer erläutert werden. Es werden zwei Varianten vorgestellt, das rein zentrale Dienstgütemanagement mit einem einzigen Managementknoten – der zur Lastverteilung oder Vermeidung des Single Point of Failure auf mehrere physikalische Knoten verteilt sein kann – und eine Möglichkeit zur Lastverteilung durch einen Managementknoten je Ingressrouter. Die als drittes vorgestellte messungsbasierte Zugangskontrolle ist kein Dienstgütemanagement in diesem Sinne, sondern lediglich eine einfache Möglichkeit, ohne Signalisierung die verfügbaren Ressourcen kooperativ zu nutzen und ansatzweise vor Überlast zu schützen. Eines der in

Kapitel 3 betrachteten alternativen Verfahren zum Dienstgütemanagement von Multicast baut hierauf auf.

2.4.2.1 Zentrales Dienstgütemanagement

Der Managementknoten im zentralen Dienstgütemanagement, im Folgenden als Domänenressourcenmanager (Domain Ressource Manager, DRM) oder kurz Ressourcenmanager bezeichnet, kontrolliert die Ressourcennutzung in der gesamten Diffservdomäne. Sein Ziel ist es, eine Überlastung von Routern oder Links in seiner Domäne auszuschließen und so die angeforderte Dienstgüte dauerhaft garantieren zu können.

Die Anforderung von Dienstgüte erfolgt über ein Protokoll zur Dienstgütesignalisierung. Quelle und/oder Empfänger spezifizieren hierüber die von ihnen gewünschten bzw. benötigten Verkehrs- und Dienstgüteparameter. Der DRM antwortet mit der von ihm getroffenen Zugangskontrollentscheidung. Sie enthält ggf. abgeänderte Verkehrsparameter, denen der neue Datenstrom entsprechen muss, und abgeänderte Dienstgüteparameter, welche der DRM bei Einhaltung der abgeänderten Verkehrsparameter garantiert. Das Ergebnis einer positiven Zugangskontrollentscheidung wird als Reservierung bezeichnet. Neue Datenströme werden vom DRM aber nur dann akzeptiert, wenn noch hinreichend freie Ressourcen entlang des voraussichtlichen Pfads durch die Domäne vorhanden sind, ansonsten werden sie abgewiesen. Erzwungen wird die Entscheidung des DRMs durch das Policing allen in die Domäne hineinfließenden Verkehrs. Der DRM steuert das Policing durch Konfiguration der zugehörigen Profile in den Grenz- und First-Hop-Routern. Die Konfiguration der Profile erfolgt dabei über das Netzwerkmanagement. Im Interdomänenfall findet zusätzlich eine Dienstgütesignalisierung zwischen den Ressourcenmanagern benachbarter Domänen statt, da die Ressourcenmanager aller Domänen entlang des Pfads zwischen Quelle und Empfänger eine Zugangskontrollentscheidung durchführen müssen. Bless hat in [28] eine solche Signalisierung mit der Möglichkeit zur Aggregation von Reservierungen vorgeschlagen. Einige der von ihm entwickelten Konzepte fließen derzeit in QoS-NSLP ein.

Während beim verteilten Dienstgütemanagement ein Router die eigenen maximal verfügbaren Ressourcen (Schnittstellengeschwindigkeit, Pufferplatz etc.) ohne weiteres ermitteln kann, muss beim zentralen Dienstgütemanagement der DRM diese erst über das Netzwerkmanagement bei jedem Router abfragen. Zusätzlich muss der DRM den Pfad kennen, den der Datenstrom zwischen einer Quelle und einem Empfänger in seiner Domäne nehmen wird, um Links und Router entlang des Pfads auf Ressourcenverfügbarkeit hin prüfen und eine Zugangskontrollentscheidung fällen zu können. Die Pfadbestimmung erfordert Kenntnis über die Topologie der Domäne und des aktuellen Routings. Wird Link-State-Routing eingesetzt, kann der DRM durch (passive) Teilnahme daran beides ermitteln. Allerdings sollte das Link-State-Routing mit nur einem Gebiet erfolgen. Andernfalls muss der DRM am Routing in allen Gebieten teilnehmen, wozu er jedoch mit jedem der Gebiete physikalisch verbunden sein muss. Wird Distance-Vector-Routing verwendet, genügt die Teilnahme daran nicht, da Distance-Vector-Routing dem DRM nur einen nächsten Hop und keinen vollständigen Pfad liefern kann. Hier müssen Topologie und Routingzustand vom DRM über das Netzwerkmanagement abgefragt werden. Um Routingänderungen schnell zu erfassen, muss sich der DRM darüber hinaus von den Routern aktiv über Linkausfälle informieren lassen, z. B. per SNMP Trap oder besser per bestätigtem InformRequest. Der DRM berechnet nach einer Routingänderung alle Pfade und die Ressourcenbelegung entlang aller geänderten Pfade neu. Sollten die Ressourcen nicht mehr für alle Reservierungen ausreichen, müssen einige hiervon abgebrochen werden. Quelle und/oder Empfänger und

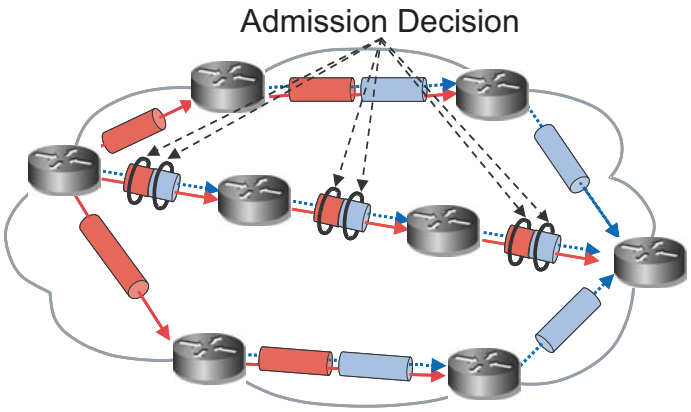


Abbildung 2.15 Wichtig für die effiziente Ressourcennutzung ist die Berücksichtigung des exakten Pfads bei der Zugangskontrolle (aus [116]).

ggf. benachbarte Ressourcenmanager werden dann über die Dienstgütesignalisierung vom Abbruch ihrer Reservierung informiert.

2.4.2.2 Lastverteilung mit Kontingenten

Anstatt den Ressourcenmanager zwecks Lastverteilung mehrfach auszuführen, kann die Zugangskontrollentscheidung von den Grenzroutern einer Domäne übernommen werden. Jeder Ingressrouter ist dabei ausschließlich für die über ihn fließenden Datenströme zuständig. Fällt der Ingressrouter aus, fällt auch der über ihn fließende Verkehr aus und wird über neue Pfade und neue Ingressrouter geroutet. Eine Redundanz gegen Ausfall ist daher nicht mehr nötig. Da der von den Ingressroutern zugelassene Verkehr die Ressourcen der inneren Router gemeinsam nutzt, müssen sich die Ingressrouter untereinander koordinieren. Hierzu erhält jeder Ingressrouter einen gewissen Anteil, ein Kontingent (Budget), an den verfügbaren Ressourcen. Bis sein Kontingent ausgeschöpft ist, kann ein Ingressrouter autonom die Zugangskontrolle abwickeln, ohne sich erneut mit den übrigen Ingressroutern abstimmen zu müssen. Wichtig für eine effiziente Ressourcenausnutzung ist die Einbeziehung des exakten Pfads bei der Kontingentverteilung. Die ressourceneffizienteste aber aufwendigste Möglichkeit ist, jedem Grenzrouter ein Kontingent von jedem Link innerhalb der Domäne zuzuweisen [116]. Wie in Abbildung 2.15 veranschaulicht, wird auf jedem Link entlang des zukünftigen Pfads die Ressourcenbelegung im eigenen Kontingent geprüft und ein Datenstrom nur dann zugelassen, wenn es noch genügend freie Kapazitäten aufweist. Gibt es in einer Domäne ausschließlich Transitverkehr, also keine Quellen oder Empfänger innerhalb der Domäne, kann stattdessen jedem Ingressrouter ein Kontingent je Egressrouter zugewiesen werden. Werden über einen Egressrouter mehrere Ingressrouter in Nachbardomänen erreicht, sollte statt eines Kontingents für den Egressrouter ein Kontingent je nachfolgendem Ingressrouter zugewiesen werden, so dass die Kontingentierung auch die unterschiedlichen Transitlinks vom Egress- zu den Ingressroutern berücksichtigen kann.

Das Berechnen der Kontingente kann durch einen zentralen Knoten [40] oder verteilt geschehen. Bei der zentralen Berechnung erhält der zentrale Knoten (Network Control Server, NCS) regelmäßig oder auf Anfrage von allen Ingressroutern die aktuellen Kontingentauslastungen, berechnet neue Kontingente teilt sie den Ingressroutern mit. Nachteil der zentralen Berechnung ist der Single Point of Failure, dem ggf. mit redundant ausgelegtem

NCS begegnet werden muss. Allerdings führt ein Ausfall des NCS nicht zum Ausfall der Dienstgüte, da bereits in den Grenzroutern etablierte Profile beim Ausfall des NCS erhalten bleiben. Es können sogar im Rahmen der an die Ingressrouter verteilten Kontingente von diesen neue Reservierungen zugelassen und Profile eingerichtet werden. Durch den Ausfall des NCS kommt es lediglich zu keiner weiteren Anpassung der Kontingente. Eine dadurch verursachte ungünstige Kontingentverteilung führt schlimmstenfalls dazu, dass ein Ingressrouter trotz eigentlich noch genügend freier Ressourcen in der Domäne neue Datenströme ablehnen muss, weil das eigene Kontingent bereits ausgeschöpft ist.

Bei verteilter Berechnung teilen die Ingressrouter ihre aktuellen Kontingentauslastungen periodisch oder kurz vor dem vollständigen Ausschöpfen des eigenen Kontingents den anderen Ingressroutern mit. Daraufhin berechnet jeder Ingressrouter auf Basis der Auslastungen nach einem auf allen Routern gleichen Algorithmus, z. B. Optimierung hinsichtlich maximalem minimal noch ungenutztem Kontingent (Maximum Residual Bandwidth), das Kontingent für sich selbst neu. Vorteil gegenüber der zentralen Berechnung durch den NCS ist die inhärente Ausfallsicherheit. Allerdings muss die Konsistenz der Daten, auf deren Basis jeder Ingressrouter die Neuberechnung der Kontingente durchführt, zu jedem Zeitpunkt sichergestellt sein. Der zur Konsistenzerhaltung nötige Signalisierungsaufwand begrenzt den Umfang der Datenbasis und die Häufigkeit, mit der sie aktualisiert werden kann. Dies limitiert u. U. erweiterte Kontingentoptimierungen, die z. B. eine größere Historie der Kontingent- und Verkehrsentwicklung mit einbeziehen.

2.4.2.3 Messungsbasierte Zugangskontrolle

Beim bisher beschriebenen parameterbasierten Ansatz des Dienstgütemanagements wird die Zugangskontrollentscheidung auf Basis der durch Quelle und/oder Empfänger signalisierten Verkehrsparameter und den bereits für andere reservierten Ressourcen getroffen. Ein Ansatz ganz ohne Dienstgütesignalisierung ist die messungsbasierte Zugangskontrolle durch Endsysteme [34]. Sie wird als End-to-End Measurement Based Admission Control (EMBAC) oder Endsystem Admission Control (EAC) bezeichnet. Bei EAC versucht ein Endsystem, die Quelle, mit einer der geplanten Kommunikation vorausgehenden Messung (Probing) anhand des dabei auftretenden Paketverlusts zu ermitteln, ob die verfügbaren Ressourcen entlang des Pfads bis zum Empfänger die geplante Kommunikation noch gestatten. Zur Bestimmung des Paketverlusts ist sie dabei auf Rückmeldungen vom Empfänger angewiesen. Nutzt die Kommunikation Assured Forwarding, genügt für die Messung ein einzelnes Paket, sofern dieses mit einer höheren Verwurfspriorität als die Pakete der späteren Kommunikation gesendet wird [22].

Motiviert wurde EAC durch die Skalierbarkeitsprobleme, die Intserv mit RSVP aufwarf. Die fehlende Dienstgütesignalisierung und Zustandshaltung in den Routern macht EAC zwar skalierbar, jedoch kann kein Policing durchgeführt werden, da den Routern die Verkehrsparameter unbekannt sind. Die Anwendbarkeit von EAC beschränkt sich daher auf Dienste wie Controlled Load, die keine harten Garantien erfordern. EAC setzt zudem voraus, dass sich die Endsysteme kooperativ verhalten, also bei negativem Ausgang des Probings auch tatsächlich keine Kommunikation beginnen.

3 Dienstgüteunterstützter Multicast im Internet

In diesem Kapitel werden Ansätze vorgestellt, die zur Bereitstellung von Dienstgüte für die Multicastkommunikation im Internet vorgeschlagen wurden. Die wesentlichen Eigenschaften der Ansätze werden herausgearbeitet, bewertet und denen des in dieser Arbeit entworfenen Verfahrens gegenübergestellt.

3.1 Multicast in Diffserv-Netzen – NRS-Problem

Bless und Wehrle definieren in [27] das Neglegted Reservation Subtree Problem, kurz NRS-Problem. Es sagt aus, dass die im Rahmen der Multicastweiterleitung stattfindende Paketduplizierung zu unerwünschter Ressourcennutzung und Verdrängung anderer Datenströme führen kann, falls die Duplizierung von hochpriorem Verkehr nicht durch das Dienstgütemanagement kontrolliert wird.

Das NRS-Problem tritt auf, weil bei Diffserv aus Skalierbarkeitsgründen das Policing nur an Domänengrenzen erfolgt. Durch Multicast entsteht jedoch auch im Inneren einer Domäne durch Paketduplizierung zusätzlicher Verkehr in einem Maße, das dem Ingress- oder First-Hop-Router ohne weiteres nicht bekannt ist und daher beim Policing nicht beachtet werden kann. Das NRS-Problem ist das zentrale Problem, das bei dienstgüteunterstütztem Multicast auf Basis von Diffserv gelöst werden muss. Gelingt es, die Duplizierung dienstgütebehafteter Pakete bzw. deren DSCPs im Inneren einer Diffservdomäne zu kontrollieren, ohne auf jedem inneren Router Policing wie am Domänenrand durchzuführen, ist ein zuverlässiger Schutz der Netzes vor Überlast möglich bei gleichzeitig erhaltener Skalierbarkeit.

Als mögliche Lösung schlagen Bless und Wehrle vor, einen MFC-Eintrag um weitere Felder zu ergänzen. Sie geben getrennt für jede Ausgangsschnittstelle den DSCP an, mit dem der Router die über diese Ausgangsschnittstelle weitergeleiteten Pakete dieser Gruppe markieren soll (s. Abbildung 3.1). Dies ermöglicht, die Pakete im Falle einer durch Multicast erforderlichen Paketduplizierung auf den DSCP eines schlechteren Dienstes wie dem Lower Effort PDB umzumarkieren und so eine Nutzung der für andere Datenströme reservierten

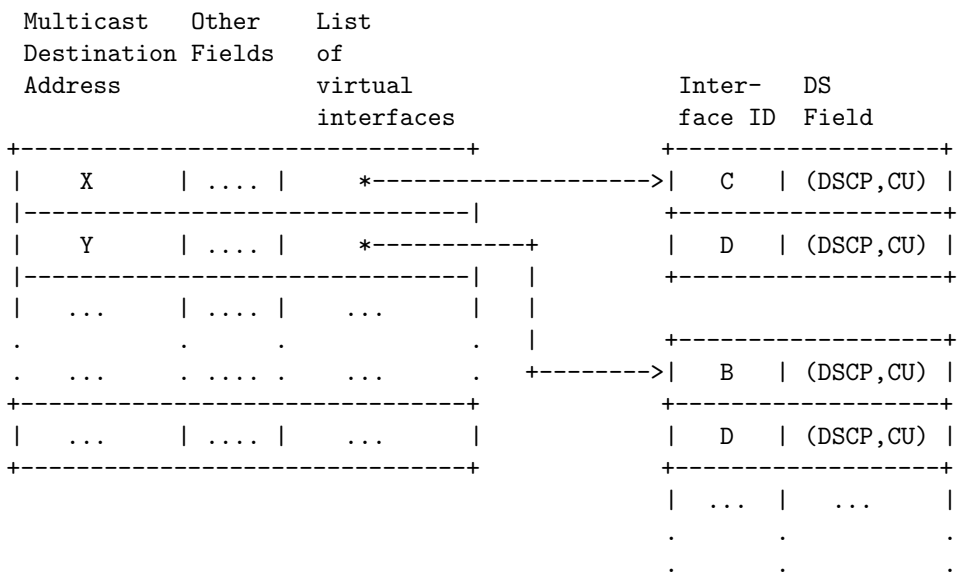


Abbildung 3.1 Bless und Wehrle schlagen zur Lösung des NRS-Problems einen erweiterten MFC mit zusätzlichen Feldern für den DSCP vor (CU=Currently Unused; aus [27]).

Ressourcen auszuschließen. Denn Limited Effort nutzt per Definition nicht die anderen Verhaltensaggregaten zugeteilten Ressourcen, sofern diese sie gerade selbst benötigen.

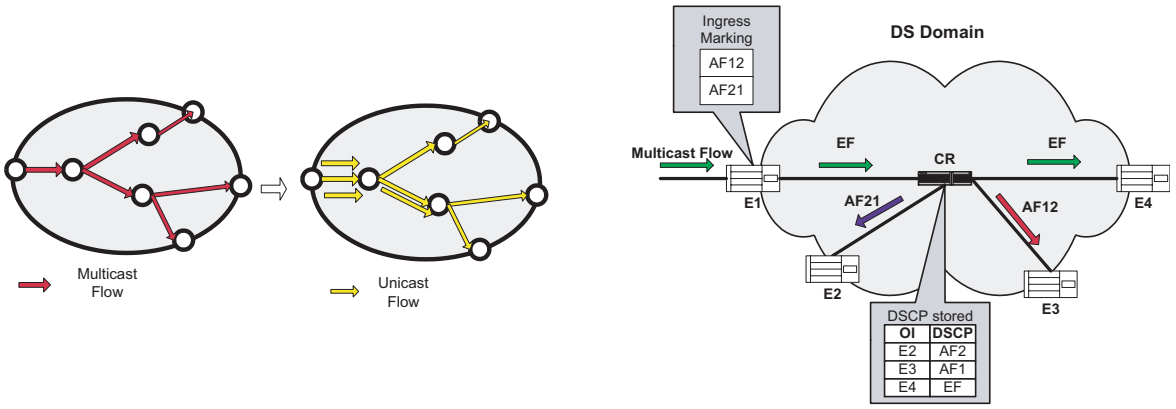
Die Tabelle links in Abbildung 3.1 zeigt pro Zeile einen MFC-Eintrag. Dargestellt sind zwei (quellenunspezifische) MFC-Einträge für die Gruppenadressen X und Y. Jeder MFC-Eintrag besitzt eine Liste mit den aktiven Ausgangsschnittstellen. Der MFC-Eintrag mit Gruppenadresse X hat die aktiven Schnittstellen C und D, Pakete für Gruppe Y werden u. a. auf die Schnittstellen B und D dupliziert. Die von Bless und Wehrle vorgeschlagene Erweiterung ergänzt nun jeden Listeneintrag für eine aktive Schnittstelle mit einem Feld mit dem DSCP, der beim Duplizieren auf die zugehörige Schnittstelle zu setzen ist („CU“ füllt den 6 bit langen DSCP auf 8 bit auf). In der Schreibweise nach Abschnitt 2.3.1 entspricht dies dem neuen schnittstellenspezifischen Wert DSCP(S,G,Oif). Pro Activebit(S,G,Oif) existiert damit ein DSCP(S,G,Oif).

Bewertung

Das vorgeschlagene Ummarkieren vermeidet das NRS-Problem. Allerdings ist die Angabe eines DSCP(S,G,Oif) hinsichtlich der dadurch freien Wahl eines DSCPs unnötig flexibel. Wie von Bless und Wehrle ausgeführt, wird letztlich entweder der DSCP von Limited Effort verwendet oder der ursprüngliche von der Quelle genutzte DSCP beibehalten. Dies ließe sich genauso mit einem Flag erreichen, das das Beibehalten oder Ummarkieren in einen routerglobal zu konfigurierenden DSCP wie Limited Effort anzeigt. Gleichzeitig verhindert ein explizit spezifizierter DSCP die gleichzeitige Nutzung unterschiedlicher DSCPs innerhalb derselben Multicastgruppe. Denn je Ausgang kann nur genau ein DSCP angegeben werden, so dass beim Duplizieren die verschiedenen DSCPs mit diesem einen überschrieben werden. Auch dieser Nachteil ließe sich bei Verwendung eines Flags umgehen, da beliebig viele verschiedene DSCPs entweder unverändert weitergeleitet werden oder bei Ressourcenknappheit alle auf Limited Effort ummarkiert werden. Schließlich wird von

Bless und Wehrle offen gelassen, wie die Konfiguration der im MFC neu eingeführten Felder mit den DSCPs durchgeführt werden soll. Diese Lücke schließen erst die nachfolgend vorgestellten Verfahren.

3.2 DAM – Diffserv-Aware Multicasting



(a) Weighted Traffic Conditioning

(b) Heterogenous DSCP Header Encapsulation

Abbildung 3.2 Zwei Verfahren von Diffserv-Aware Multicasting: (a) WTC betrachtet Multicast wie mehrere Unicastdatenströme, (b) HDE transportiert mehrere DSCPs in einem Paket (aus [194]).

Yang und Mohapatra [194] greifen mit ihrem Diffserv-Aware Multicasting (DAM) die von Bless und Wehrle vorgeschlagene Lösung des NRS-Problems auf und ergänzen sie um ein Verfahren zur Bestimmung und Konfiguration der beim Duplizieren zu verwendenden DSCPs. Ihr Verfahren besteht aus drei Bausteinen: Weighted Traffic Conditioning (WTC),Receiver Initiated Marking (RIM) und Heterogenous DSCP Header Encapsulation (HDE).

WTC besagt, dass im Rahmen des Policings am Ingressrouter ein eingehender Multicastdatenstrom n-fach berücksichtigt wird entsprechend der Anzahl n an vom Multicastrouting stromabwärts in der Domäne noch erzeugten Kopien. Der Multicastdatenstrom wird dadurch beim Policing so behandelt, als würden stattdessen n Unicastdatenströme gesendet werden (Abbildung 3.2(a)). Jeder Ingressrouter unterhält für WTC eine Tabelle, die für jede Multicastgruppe getrennt nach DSCPs die Anzahl n an Kopien vermerkt. Der Wert von n wird im Rahmen von RIM gesetzt und mit jedem Gruppenbeitritt und -austritt eines Empfängers angepasst.

RIM besagt, dass der Empfänger bestimmt, welchen Dienst er erhält. Dazu wird die QoS-Information mit dem vom Empfänger gewünschten Dienst huckepack mit der Join-Nachricht des Multicastroutingprotokolls stromaufwärts gesendet. Es wird vom Empfänger aus gesehen der erste Verzweigungsrouter im Verteilbaum bestimmt, bei dem mindestens die vom Empfänger gewünschte Dienstgüte verfügbar ist. Das heißt, es wird ein Router bestimmt, bei dem der Multicastdatenstrom mit einem DSCP markiert eintrifft, dem ein verglichen mit dem vom Empfänger gewünschten Dienst gleichwertiger oder besserer Dienst zugeordnet ist. Der Verzweigungsrouter informiert daraufhin den Ingressrouter, der

seinerseits beim Dienstgütemanagement anfragt, ob der neue Zweig zugelassen werden soll. Wird der neue Zweig akzeptiert, passt der Ingressrouter sein WTC an sendet eine positive Bestätigung zurück zum Verzweigungsrouter. Dieser ändert daraufhin das Ummarkieren von Best Effort, wie es seit Empfang der Join-Nachricht und Aktivieren der neuen Ausgangsschnittstelle für den Empfänger erfolgt, auf den vom Empfänger gewünschten DSCP. Bei negativer Bestätigung wird weiterhin auf Best Effort ummarkiert. Bei RIM kann ein Empfänger statt eines bestimmten Dienstes auch die jeweils an seinem Ort beste augenblicklich verfügbare Dienstgüte anfordern. Dann wird vom Empfänger aus gesehen am ersten Verzweigungsrouter stromaufwärts der DSCP der dort eintreffenden Pakete einfach für den neuen Empfänger übernommen, nachdem die positive Bestätigung vom Dienstgütemanagement bzw. Ingressrouter eingetroffen ist.

Mittels HDE kann der Ingressrouter jedes Paket mit weiteren DSCPs versehen, auf die es in verschiedenen Zweigen des Verteilbaums stromabwärts noch ummarkiert werden soll. In Abbildung 3.2(b) markiert der Verzweigungsrouter CR (Core Router) auf seinen drei Ausgangsschnittstellen auf Expedited Forwarding (EF) bzw. auf zwei unterschiedliche Assured-Forwarding-Klassen AF1 und AF2 um. Dies steuert der bei ihm für jeden Ausgang vermerkte DSCP (Tabelle „DSCP stored“). Für die Weiterleitung bis zum Verzweigungsrouter CR kann ein Paket allerdings nur mit einem der drei DSCPs markiert werden. Der Ingressrouter E1 wählt daher jenen DSCP, der der höchsten Dienstgüte dieser drei DSCPs entspricht, in der Abbildung ist dies EF. Das vom Ingress Router E1 durchgeführte Policing könnte ergeben, dass das Paket im Rahmen eines der Dienste ummarkiert werden soll. So soll das Paket im Fall der AF-Klasse 1 mit Verwurfspriorität 2 (AF12) und im Fall der AF-Klasse 2 mit Verwurfspriorität 1 (AF21) markiert werden (Tabelle „Ingress Marking“). Mit HDE ist es dem Ingress Router nun möglich, genau dies dem Verzweigungsrouter im Inneren der Domäne mitzuteilen. In der Abbildung fügt der Ingressrouter E1 dem mit EF markierten Paket per HDE die beiden DSCPs AF12 und AF21 hinzu (nicht dargestellt). Welchen der drei DSCPs EF, AF12 und AF21, die das Paket nun trägt, vom Router CR auf welchem seiner Ausgänge verwendet wird, bestimmt die Tabelle von CR mit den DSCPs je Ausgang. Unter HDE bestimmen die bei einem Verzweigungsrouter je Ausgang vermerkten DSCPs somit nur noch die AF-Klasse und nicht mehr die Verwurfspriorität, d. h. nicht mehr den exakten DSCP.

Zusammengefasst bestimmt unter DAM der Empfänger mittels RIM, welchen Dienst und damit AF-Klasse für ihn verwendet wird. Den genauen DSCP bestimmt jedoch der Ingressrouter mittels HDE, wobei das vom Ingressrouter durchgeführte Policing mittels WTC die Anzahl insgesamt erzeugter Duplikate mit berücksichtigt.

Bewertung

DAM behandelt beim Policing jeden Multicaststrom so, als würden stattdessen n Unicastdatenströme gesendet werden. Dies überschätzt die tatsächlich auf Weiterleitungsebene benötigten Ressourcen je nach Anzahl an Empfängern bzw. Paketduplizierungen beträchtlich und führt entsprechend zu einer ineffizienten Ressourcenausnutzung.

DAM benötigt ein Multicastroutingprotokoll mit Join-Nachricht oder einer äquivalenten Nachricht wie dem Graft, das die QoS-Information huckepack transportiert. Damit ist DAM für alle Multicastroutingprotokolle geeignet ausgenommen MOSPF, da MOSPF keine dem Join entsprechende Nachrichten kennt. Auf die konkrete Umsetzung für ein Multicastroutingprotokoll geht DAM allerdings nicht ein.

DAM nimmt an, dass es eine Ordnung unter den verfügbaren DSCPs bzw. Diensten gibt. Diese ist i. A. aber eben nicht gegeben, da unterschiedliche Dienste orthogonale Ziele verfolgen. So bietet EF zwar eine geringe Weiterleitungsverzögerung. Kurzzeitiges Überschreiten der zugesicherten Rate führt aber zu sofortigem Paketverlust. Umgekehrt erlaubt AF zwar Ratenschwankungen, bietet jedoch keine geringe Latenz. EF in AF umzumarkieren, wie in Abbildung 3.2(b) dargestellt, oder umgekehrt AF in EF ist somit wenig sinnvoll.

Selbst wenn man sich beim Ummarkieren auf DSCPs mit einer Ordnung untereinander wie den DSCPs aus einer AF-Klasse beschränken würde, ist es problematisch, die Verringerung der Dienstgüte durch Ummarkieren an Verzweigungsroutern durchzuführen. Denn hier entscheiden die Router entlang des Pfades willkürlich, welche Pakete verworfen werden und welche nicht. Eine Quelle kann dagegen viel gezielter ihre Daten und Rate an die verfügbare Dienstgüte anpassen. Zum Beispiel könnten in einem komprimierten Videostream gezielt nur hochfrequente Bildanteile (Details) verworfen werden. Des Weiteren ist das Potential für ein zufälliges Verwerfen von Paketen je nach Art des Datenstroms u. U. außerordentlich gering. So zeigt komprimiertes Video bereits bei 1 % Paketverlust starke Bildstörungen und ab etwa 2 % bricht die Darstellung völlig ab, wie eigene Versuche mit MPEG-2-komprimiertem Video über RTP und UDP ergeben haben.

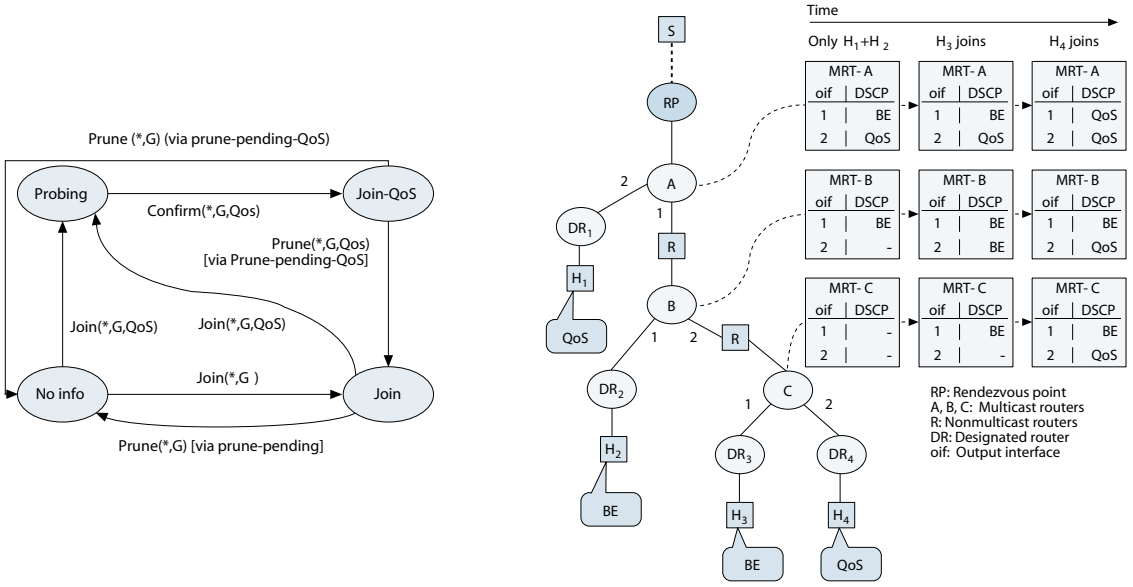
Eine Quelle hat bei DAM keinerlei Einfluss auf die Dienstgüte. Sie muss diese Information daher auf Anwendungsebene von ihren Empfängern erhalten, um beispielsweise die Kodierungsrate für einen Videostream bestimmen und je nach Paketverlustwahrscheinlichkeit Vorkehrungen durch z. B. Vorwärtsfehlerkorrektur oder Interleaving der Daten im Datenstrom treffen zu können. Letztlich könnte der von der Quelle erzeugte Datenstrom sogar einen ganz bestimmten Dienst erfordern, was es sinnlos macht, einen Empfänger die Dienstgüte bestimmen zu lassen.

Zusammenfassend lässt sich festhalten, dass DAM ein das NRS-Problem lösendes Verfahren darstellt. Zudem wird die von Bless und Wehrle offen gelassene Frage der Konfiguration der DSCPs in den Routern beantwortet. DAM skizziert hierzu eine Signalisierung auf Basis von Multicastrouting mit explizitem Join und zentralem Dienstgütemanagement. Die Ressourceneinsparungen durch Multicast werden bei der Zugangs- und Verkehrskontrolle jedoch nicht berücksichtigt, was zu ineffizienter Ressourcenausnutzung führt. Und sowohl die Unterstützung von heterogenen Gruppen durch das Ummarkieren auf beliebige DSCPs als auch der Empfänger-basierte Ansatz zur Auswahl der Dienstgüte erscheinen ohne ein Einbeziehen der Quelle als fragwürdig.

3.3 QUASIMODO – EMBAC für PIM-SM

Das von Bianchi et al. vorgestellte QUASIMODO (QoS-aware Multicasting over Diffserv and Overlay Networks) [21] erweitert das ebenfalls von ihnen für Unicast entwickelte GRIP (Gauge and Gate Reservation with Independent Probing) [22] auf das Multicastroutingprotokoll PIM-SM.

GRIP arbeitet nach dem Prinzip der messungsbasierten Zugangskontrolle durch die Endsysteme. Als Endsysteme fungieren bei GRIP die Grenzrouter einer Diffservdomäne. GRIP benötigt zwei DSCPs einer AF-Klasse AFx1 und AFx2. Der DSCP mit der größeren Verlustwahrscheinlichkeit AFx2 wird ausschließlich für die Probingpakete verwendet. Bei



(a) (*,G)-Zustandsautomat je Downstreamschnittstelle

(b) Beispielszenario

Abbildung 3.3 (a) QUASIMODO erweitert den Zustandsautomat von PIM-SM und ermöglicht so (b) in einer Multicastgruppe gleichzeitig Empfänger ohne und mit – dann allerdings einheitlicher – Dienstgüte (aus [21]).

GRIP sendet der Ingressrouter ein mit AFx2 markiertes Paket zum Egressrouter. Erreicht das Probingpaket den Egressrouter, bestätigt er den Empfang mit einem mit AFx1 markierten Paket. Falls eine bidirektionale Messung gewünscht ist, wird mit einem AFx2 markierten Paket geantwortet. Erreicht die Bestätigung den Ingressrouter nicht innerhalb einer definierten Zeit, gilt der Datenstrom als abgelehnt. Es kann nach einem Backoff ggf. ein erneutes Probing gestartet werden. Andernfalls beginnt mit dem Empfang der Bestätigung die eigentliche Kommunikation. Sie nutzt dabei ausschließlich AFx1.

Es genügt ein Probingpaket um festzustellen, ob die mittlere Last entlang des Pfads einen weiteren Datenstrom zulässt. Grund ist, dass bei Assured Forwarding nur auf langfristige Überlast mit dem Verwerfen von Paketen reagiert werden darf. Kurzfristige Schwankungen müssen dagegen durch Puffern abgefangen werden. Der über den zweiten DSCP AFx2 realisierte Signalisierungskanal wird durch die aktuelle Auslastung der Router mit AFx1-Verkehr also derart beeinflusst, dass ein über ihn gesendetes Probingpaket den 1-Bit-Wert „Stau“ ja/nein von jedem Router entlang des Pfads ausliest und zu den Grenzroutern transportiert. Das Verwerfen des Probingpakets durch einen überlasteten inneren Router realisiert die nötige logische Oder-Verknüpfung aller ausgelesenen Werte.

QUASIMODO überträgt GRIP auf PIM-SM und erweitert dafür PIM-SM um drei neue Nachrichten Join-QoS, Confirm-QoS und Prune-QoS. Die Downstream Join State Machine von PIM-SM wird entsprechend um drei weitere Zustände Probing, Join-QoS und Prune-Pending-QoS ergänzt (s. Abbildung 3.3(a)). Empfängt ein Router ein Join-QoS (in der Abbildung als Join(*,G,QoS) geschrieben), wechselt er vom Grundzustand „No Info“ in den Zustand Probing. Beim späteren Eintreffen des Confirm-QoS (Confirm(*,G,QoS))

wechselt er weiter in den Zustand Join-QoS. Die Weiterleitung erfolgt ab hier mit Dienstgüte unter Beibehaltung des DSCPs. Ein Prune-QoS (Prune(*,G,QoS)) führt über den Zwischenzustand Prune-Pending-QoS in den Zustand Join und deaktiviert damit die Dienstgüte. Pakete werden im Zustand Join auf Best Effort ummarkiert weitergeleitet. Aus dem Zustand Join heraus kann entweder mittels Join-QoS die Dienstgüte erneut aktiviert oder mittels Prune die Multicastweiterleitung ganz beendet werden. Das direkte Beenden einer dienstgüteunterstützten Multicastweiterleitung aus Join-QoS heraus mittels herkömmlichem Prune über den Zwischenzustand Prune-Pending-QoS ist genauso möglich.

GRIP dient QUASIMODO dazu, einen neu zu etablierenden Multicastpfad von einem Verzweigungsrouter mit bereits etablierter Dienstgüte bis zu einem neuen Empfänger auf Stausituationen hin zu überprüfen. Nur wenn kein Stau vorliegt, können auch auf dem neuen Multicastpfad Pakete unter Beibehaltung ihres DSCPs weitergeleitet und Dienstgüte für den neuen Empfänger bereitgestellt werden. Sendet der Designated Router des neuen Empfängers ein Join-QoS Richtung Rendezvousstelle oder Quelle, etabliert dies in den Routern entlang des neuen Pfads bis zum ersten sich im Zustand Join-QoS befindlichen Verzweigungsrouter den Zustand Probing. Der Verzweigungsrouter sendet daraufhin gemäß GRIP ein mit DSCP AFx2 markiertes Probingpaket an die in der Join-QoS-Nachricht genannte Multicastadresse. Das Probingpaket wird von den weiteren Routern stromabwärts nur auf Schnittstellen im Zustand Probing weitergeleitet. Damit während des Probings nicht schon reguläre Datenpakete auf den neuen Multicastpfad fließen, leiten Router im Zustand Probing ausschließlich Probingpakete weiter, also keine mit AFx1 markierten Datenpakete. Erreicht das Probingpaket den Designated Router, wechselt dieser vom Zustand Probing in den Zustand Join-QoS und antwortet stromaufwärts mit einem Confirm-QoS. Diese Nachricht sorgt in allen Routern entlang des neuen Multicastpfads bis hinauf zum Verzweigungsrouter für den Zustandswechsel von Probing nach Join-QoS. Mit Erreichen des Verzweigungsrouters fließt der Verkehr ohne ummarkiert zu werden bis zum neuen Empfänger. Ab diesem Zeitpunkt steht für den neuen Empfänger die Dienstgüte bereit.

Des Weiteren ermöglicht QUASIMODO, dass Pfade ohne Dienstgüte mit einer normale Join-Nachricht an einen Verteilbaum mit Join-QoS-Zustand angehängt werden können. Eine abgestufte Verringerung der Dienstgüte wie bei DAM sieht QUASIMODO nicht vor. QUASIMODO verwendet hierzu die von Bless und Wehrle vorgeschlagene Erweiterung der MFC-Einträge (s. Abbildung 3.3(b)). Jeder der abgebildeten Router A, B und C besitzt jeweils zwei Ausgangsschnittstellen (oif) 1 und 2. Ihre MFC-Einträge für die betrachtete Gruppe sind mit MRT-A, MRT-B und MRT-C bezeichnet. Zu Beginn sind nur die beiden Hosts H_1 und H_2 der Gruppe beigetreten, wobei H_1 Dienstgüte erfährt und H_2 nicht. Entsprechend befindet sich Schnittstelle 2 von Router A zum Designated Router DR_1 von Host H_1 im Zustand Join-QoS. Im MFC-Eintrag ist für diese Schnittstelle entsprechend der von der Quelle verwendete DSCP „QoS“ vermerkt. Anders Schnittstelle 1 Richtung Host H_2. Sie befindet sich im Zustand Join, was ein Ummarkieren nach Best Effort anzeigt. Entsprechend ist für diese im MFC-Eintrag von Router A Best Effort als DSCP vermerkt.

Tritt Host H_3 der Gruppe bei und wünscht dabei keine Dienstgüte, sendet sein Designated Router DR_3 ein Join in Richtung Rendezvousstelle RP, was die Weiterleitung auf Ausgangsschnittstelle 1 von Router C und Ausgangsschnittstelle 2 von Router B aktiviert. Da der Zustand Join ein Ummarkieren nach Best Effort anzeigt, wird jeweils als DSCP BE im MFC eingetragen. Der vierte Host H_4 wünscht dagegen Dienstgüte. Sein Designated Router DR_4 wird daher ein Join-QoS Richtung RP senden. Nach erfolgreichem

Probing durch den ersten Verzweigungsrouter mit Dienstgüte, hier Router B, sorgt das anschließende Confirm-QoS von Router DR_4 für den Zustandswechsel nach Join-QoS und ein entsprechendes Abändern der Einträge in den MFCs von Router B und C und Setzen des DSCPs „QoS“.

Bewertung

QUASIMODO ergänzt wie DAM die von Bless und Wehrle vorgeschlagene Lösung des NRS-Problems um eine geeignete Signalisierung. QUASIMODO wurde explizit für PIM-SM spezifiziert. Die Signalisierung ließe sich jedoch prinzipiell auch auf andere Protokolle anwenden, sofern sie eine dem Join entsprechende Nachricht besitzen. Wie DAM ist QUASIMODO damit auf alle Multicastroutingprotokolle übertragbar ausgenommen MOSPF. QUASIMODO nutzt mit GRIP eine messungsbasierte Zugangskontrolle, wodurch es sich nur für Dienste mit weichen Garantien wie Controlled Load eignet (vgl. Abschnitt 2.4.2.3). Dienste mit harten Garantien können mit QUASIMODO nicht bereitgestellt werden.

3.4 Analyse

Alle präsentierten Ansätze lösen das von Bless und Wehrle identifizierte NRS-Problem. Dies ist die Grundvoraussetzung, um Dienstgüte für Multicast erbringen zu können. Signalisierung zur Konfiguration des eingeführten Zustands lassen Bless und Wehrle offen. Die Signalisierung von QUASIMODO ist für PIM-SM entwickelt worden, DAM bleibt bezüglich des Multicastroutingprotokolls ungenau. Prinzipiell müsste sich jedoch für DAM und QUASIMODO jedes Multicastroutingprotokoll eignen, das mit einem expliziten Join oder einer äquivalenten Nachricht wie dem Graft arbeitet. Bis auf MOSPF trifft dies auf alle derzeit für das Internet standardisierten Multicastroutingprotokolle zu.

Alle Ansätze basieren auf Bless und Wehrles Lösungsvorschlag zum NRS-Problem. Dieser erlaubt jedoch keine gleichzeitige Nutzung mehrerer DSCPs in einer Multicastgruppe. Dies ist für Anwendungen mit gleichzeitig mehreren unterschiedlichen Dienstgüteanforderungen wie beispielsweise für die computergestützte Zusammenarbeit (CSCW) über das Internet wünschenswert. Hierbei findet typischerweise eine Audio-/Videokommunikation (meist konstante Rate, geringe Verzögerung wünschenswert) sowie die gemeinsame Nutzung einer Arbeitsoberfläche oder eines Whiteboards statt (burstartige Kommunikation, möglichst geringer Paketverlust). Ohne die Möglichkeit für unterschiedliche DSCPs innerhalb einer Multicastgruppe bleibt nur die Möglichkeit, jeweils eine eigene Multicastgruppe pro DSCP zu etablieren. Dies bedeutet mehrfachen Multicastroutingzustand in den Routern, der im Hinblick auf die begrenzte Skalierbarkeit des Multicastroutings (vgl. Abschnitt 2.2.3) besser vermieden werden sollte.

DAM setzt auf zentrales Dienstgütemanagement und Policing durch Ingressrouter, ermöglicht somit garantierte Dienste. QUASIMODO ist dagegen durch seine Wahl der messungsbasierten Zugangskontrolle nicht für garantierte Dienste geeignet. Damit können Anwendungen, die jederzeit, insbesondere auch kurzfristig, definierte Verzögerungsgrenzen bzw. Durchsatzraten benötigen, nicht unterstützt werden. Dies sind alle stark interaktiven und damit äußerst verzögerungssensitiven Anwendungen wie beispielsweise verteiltes Musizieren. Bei ihnen ist ein Puffern zum Ausgleich kurzfristiger Ratenschwankungen nicht möglich.

DAM ist durch die Überabschätzung des Ressourcenbedarfs wenig effizient. Dies ist letztlich ein Kostenfaktor, da so Netzwerkressourcen bereitgestellt werden müssen, die ungenutzt bleiben. Weiterhin sieht DAM vor, dass nur Empfänger, nicht aber die Quellen, Einfluss auf die gewünschte Dienstgüte nehmen können. Dies verschenkt die Möglichkeit, den gesendeten Datenstrom optimal an die verfügbare Dienstgüte anzupassen und führt letztlich zu einer schlechteren resultierenden Qualität auf Anwendungsebene als mit den zur Verfügung stehenden reservierten Ressourcen möglich wäre.

Anforderungen

Zusammenfassend lassen sich aus den genannten Kritikpunkten folgende Anforderungen an ein Verfahren zur Bereitstellung von Dienstgüte für Multicast auf Basis von Diffserv identifizieren:

- Es löst das NRS-Problem,
- ermöglicht mehrere DSCPs je Multicastgruppe,
- unterstützt beliebige Dienste, insbesondere auch garantierte Dienste,
- gestattet eine effiziente Ressourcennutzung und
- ermöglicht sowohl Quellen wie Empfängern eine Einflussnahme auf die gewünschte Dienstgüte.

DSMC – Diffserv Multicast

Das im folgenden Kapitel vorgestellte Verfahren „Diffserv Multicast“, kurz DSMC[1], gestattet es einem zentralen Ressourcenmanager, die genannten Anforderungen zu erfüllen. DSMC besteht im Wesentlichen aus zwei Teilen:

- dem in der Weiterleitungsebene im MFC neu eingeführten Zustand, der das NRS-Problem löst und mehrere DSCPs je Multicastgruppe ermöglicht, und
- einer Signalisierung, mit der ein zentraler Ressourcenmanager effizient und skalierbar diesen Zustand konfigurieren kann.

Der im MFC neu eingeführte Zustand basiert auf der Idee, dass bereits ein Flag je Schnittstelle je MFC-Eintrag genügt, das NRS-Problem zu lösen. Hinzu kommt weiterer Zustand, der im Rahmen der Signalisierung zwischen Routern und dem Ressourcenmanager benötigt wird. Die Signalisierung ist effizient und skalierbar bezüglich der von den Routern an den Ressourcenmanager gesendeten Nachrichten und damit hinsichtlich des Verarbeitungsaufwands beim Ressourcenmanager. Meist wird nur genau eine Nachricht je neuem Empfänger bzw. Multicastpfad benötigt unabhängig von der Länge des neuen Multicastpfads. DSMC wird im folgenden Kapitel detailliert erläutert.

[1] Das in dieser Arbeit entworfene DSMC ist nicht zu verwechseln mit dem ebenfalls DSMC genannten Verfahren von Striegel und Manimaran [168], das das Diffserv-Prinzip auf Multicast*routing* anwendet und als Alternative zu klassischem IP-Multicastrouting konzipiert ist. Deren DSMC betrachtet mit dem Routing ein gänzlich anderes Problemfeld als das hier vorgestellte DSMC.

Die unterstützten Dienste und die Effizienz der Ressourcennutzung hängen von der Art der Ressourcenverwaltung durch den Ressourcenmanager ab. DSMC schließt jedoch anders als QUASIMODO weder garantierte Dienste noch wie DAM eine effiziente Ressourcennutzung von vornherein aus. DSMC nimmt an, dass die Einflussnahme der Quelle über die vorhandene Dienstgütesignalisierung für Unicast praktisch ohne Änderungen erfolgen kann und dass es für die Empfänger genügt, zwischen dem Nichtakzeptieren und – hinreichend verfügbare Ressourcen vorausgesetzt – dem Akzeptieren der von der Quelle bestimmten Dienstgüte wählen zu können. Heterogener Multicast wird nicht unterstützt.

Annahmen

Im einzelnen liegen DSMC die folgenden Annahmen zu Grunde. Sie werden jeweils gleich begründet.

- Das Dienstgütemanagement erfolgt zentral mit einem Ressourcenmanager.

  Bei verteiltem Dienstgütemanagement kann jeder Router selbst den mit DSMC neu eingeführten Zustand verwalten. Eine Signalisierung, wie sie DSMC definiert, ist damit überflüssig. Zumindest der von DSMC im MFC neu eingeführte Zustand kann bei verteiltem Dienstgütemanagement als Alternative zum Eingriff in das Multicastrouting oder dem vollständigen Policing auf domäneninternen Routern eingesetzt werden, welche sonst zur Kontrolle der Paketduplizierung und Lösung des NRS-Problems nötig wären.

- Das Multicastrouting erfolgt in der Vermittlungsschicht unter Nutzung eines MFCs in der Weiterleitungsebene.

  DSMC modifiziert wie alle in diesem Kapitel vorgestellten Verfahren den MFC. Es setzt daher eine auf einem MFC basierende Paketweiterleitung voraus. Auf die Dienstgüteunterstützung für alternative Multicastverfahren wird unten noch eingegangen.

- Der DSCP hat keinen Einfluss auf das Routing.

  Routing abhängig vom DSCP, d. h. QoS-Routing (einen Überblick gibt [188]), wird im Internet derzeit nicht eingesetzt. Selbst ein begrenztes Umverteilen des Verkehrs im Rahmen der von Relaxed ECMP zur Verfügung stehenden Alternativpfade [183], um die maximale Last entlang der Alternativpfade zu minimieren, hat sich letztlich nicht durchsetzen können. DSMC lässt daher ein vom DSCP abhängiges Routing außer Acht und geht davon aus, dass ein Ändern des DSCPs zu keiner Änderung des Pfads führt. Heute werden bei Bedarf mittels Traffic Engineering durch MPLS (Multiprotocol Label Switching [152]) den Anforderungen genügende Pfade zwischen Ingress- und Egressroutern erzwungen, wobei diese ggf. mittels zentraler Vorausberechnung auf Basis der aktuellen Auslastung bestimmt werden [116]. DSMC lässt sich nur dann auch für MPLS nutzen, wenn wie in [131] vorgeschlagen, weiterhin klassisches Multicastrouting mit einem MFC und unabhängig vom DSCP durchgeführt wird. Werden dagegen explizite Punkt-zu-Mehrpunkt-Verbindungen konfiguriert [3], gibt es keinen MFC mehr und DSMC ist nicht anwendbar. Da die Pfade und Paketduplizierungen hier jedoch von vornherein bekannt sind, kommt es ohnehin nicht zum NRS-Problem.

- Die Quelle bestimmt den Dienst und der Ressourcenmanager überprüft die Berechtigung der Quelle zur Nutzung des Dienstes.

  Die Quelle fordert beim Ressourcenmanager Dienstgüteunterstützung an und nennt dabei Multicastadresse und die gewünschten Verkehrs- und Dienstgüteparameter. Der Ressourcenmanager überprüft im Rahmen der Zugangskontrolle zu diesem Zeitpunkt lediglich, ob die Quelle grundsätzlich berechtigt ist, den Dienst zu nutzen. Ggf. lässt er die Nutzung nur mit modifizierten Verkehrs- und Dienstgüteparametern zu, die er der Quelle mitteilt. Der Ressourcenmanager aktualisiert daraufhin die Profile im First-Hop-Router bzw. im Falle von Interdomänenmulticast die Profile im Ingressrouter. Eine Zugangskontrolle auf Basis der verfügbaren Ressourcen (vgl. nächster Punkt) findet zu diesem Zeitpunkt noch nicht statt, da ohne Empfänger noch kein Multicastpfad etabliert ist und noch keine Ressourcennutzung erfolgt.

  Dienstgütesignalisierung und -managementarchitektur für Unicast, wie sie z. B. in [28] entworfen wurden, können somit unverändert für Multicast und DSMC übernommen werden. Es muss lediglich gewährleistet sein, dass bei der Dienstgüteanforderung auch eine Multicastadresse als Zieladresse angegeben kann und vom Ressourcenmanager für eine Zugangskontrolle akzeptiert wird.

- Die Zugangskontrolle auf Basis der verfügbaren Ressourcen erfolgt mit dem Beitritt eines Empfängers.

  Die Ressourcenüberprüfung durch den Ressourcenmanager muss für Multicast erweitert werden. Hierunter fällt die Berechnung des neuen Multicastpfads zum beigetretenen Empfänger und die Verwaltung der belegten Ressourcen, da hier ein Verteilbaum berücksichtigt werden muss. Wie dies erfolgen kann, wird im Zusammenhang mit der Nutzung von DSMC durch den Ressourcenmanager im folgenden Kapitel dargestellt.

- Empfänger wählen nur zwischen Empfang mit den von der Quelle vorgegeben Dienstgüteparametern oder dem Nichtempfang.

  Die Wahl zwischen Empfang und Nichtempfang wird den Empfängern durch das Gruppenmanagementprotokoll bereitgestellt. Alles darüber Hinausgehende wie z. B. die garantierte Verfügbarkeit von Ressourcen für bestimmte Empfänger oder die Aushandlung von Dienstgüteparametern zwischen allen Quellen und Empfängern einer Gruppe erfolgt durch eine geeignete Dienstgütesignalisierung zwischen Quellen, Empfängern und Ressourcenmanager. Hierzu ist die Dienstgütesignalisierung von Unicast um entsprechende Funktionen für Multicast zu erweitern. In der vorliegenden Arbeit wird hierauf nicht weiter eingegangen.

- Es werden nur homogene Multicastgruppen unterstützt.

  IP-Multicast ermöglicht heterogene Gruppen, unterstützt die höheren Schichten jedoch nicht, mit der Heterogenität umzugehen. In heterogenen Gruppen werden von den Routern die Pakete wahlfrei zufällig verworfen. Es wäre vorstellbar, mittels verschiedener Verwurfsprioritäten, einer je in der Gruppe auftretenden Qualitätsstufe von Empfänger, das Verwerfen durch die Router von der Quelle aus zu steuern. Jedoch bieten nicht alle Diffserv PHBs mehrere Verwurfsprioritäten an, von den derzeit standardisierten lediglich Assured Forwarding. Und wenn, ist die Anzahl der durch sie selektierten Verwurfswahrscheinlichkeiten gering. Typischerweise implementieren Router nur zwei oder drei verschiedene Verwurfswahrscheinlichkeiten. Da es sich um

Wahrscheinlichkeiten handelt, beginnen die Router zudem bereits mit dem Verwerfen von Paketen einer niedrigeren Verwurfspriorität, bevor alle Pakete einer höheren Verwurfspriorität verworfen worden sind. Somit gelingt das Steuern durch die Quelle nur ansatzweise.

Eine heterogene Gruppe lässt sich jedoch mittels Layered Multicast [112] auf einfache Weise mit mehreren homogenen IP-Multicastgruppen nachbilden. Die Quelle teilt dazu ihren Datenstrom auf mehrere IP-Multicastgruppen auf. Die Empfänger treten je nach gewünschter Qualitätsstufe einer entsprechenden Anzahl dieser Multicastgruppen bei. Heterogene Gruppen auf Basis von Layered Multicast werden von DSMC ohne Einschränkungen unterstützt. Allerdings benötigt Layered Multicast durch die zusätzlichen Multicastgruppen zusätzlichen Weiterleitungszustand, was hinsichtlich der begrenzten Skalierbarkeit von IP-Multicast von Nachteil ist.

Heterogene Gruppen lassen sich ebenfalls mittels aktiver und programmierbarer Netze umsetzen. Dies wird von DSMC zwar nicht direkt unterstützt. Ein aktiver Knoten, der den Multicastverkehr verändert, kann jedoch genau wie eine Quelle per Dienstgütesignalisierung mit dem Ressourcenmanager geänderte Verkehrs- und Dienstgüteparameter für den Teilbaum aushandeln, in dem er die Wurzel bildet. Der Pfad, den die Multicastpakete nehmen, darf durch die aktiven Knoten allerdings nicht verändert werden. Das NRS-Problem beim IP-Multicast zwischen aktiven Knoten kann mit DSMC behandelt werden. Bei der Ressourcenüberprüfung werden dann nicht die Verkehrs- und Dienstgüteparameter der Quelle sondern die des letzten aktiven Knotens stromaufwärts zugrunde gelegt.

Alternative Multicastverfahren

Explicit und Source Routed Multicast (s. Abschnitt 2.2.3) führen nicht zum NRS-Problem. Bereits am First-Hop-Router sind alle Empfänger aus dem Paket ablesbar und ein entsprechendes Policing kann jene Empfängerliste erzwingen, die vom Ressourcenmanager auf Ressourcenverfügbarkeit längs der Pfade hin überprüft und zugelassen wurde. Allerdings müssen sowohl Dienstgütesignalisierung, Dienstgütemanagement als auch Policing erweitert werden, damit komplette Empfängerlisten signalisiert und am First-Hop-Router überprüft werden können. Hauptnachteil dieser Multicastverfahren bleibt ihr hoher zusätzlicher Overhead. Für ihn müssen entsprechend zusätzliche Ressourcen reserviert werden. Speziell beim Explicit-Multicast-Verfahren MSC sowie bei Source Routed Multicast können Router als Empfänger im Paket eingetragen sein. Die Router leiten das Paket dann per IP-Multicast an die eigentlichen Gruppenmitglieder weiter. Sinnvollerweise ist hier der IP-Multicast in seiner Reichweite beschränkt. Sollte die Reichweite größer als linklokal sein, kann es wiederum zum NRS-Problem kommen. Dies lässt sich dann mittels DSMC behandeln.

Application Layer oder Overlay Multicast verwenden in der Vermittlungsschicht ausschließlich Unicast. Das Dienstgütemanagement für Unicast unterstützt damit automatisch auch Multicast auf Basis dieser Verfahren. Die Endsysteme und Proxys müssen lediglich für jede zwischen ihnen bestehende Unicastverbindung separat die benötigte Dienstgüte anfordern. Heterogene Gruppen sind hierbei problemlos möglich. Allerdings ist das im Vergleich zu IP-Multicast weniger stabile Routing von Application Layer Multicast mit seinen relativ häufigen Routingänderungen gerade vor dem Hintergrund von Dienstgüte bestenfalls problematisch.

4 DSMC – Diffserv Multicast

Ausgehend von den im letzten Kapitel aufgeführten Anforderungen und Annahmen wurde das Verfahren DSMC (Diffserv Multicast) entworfen. Die Signalisierung zur Behandlung des NRS-Problems wird durch ein neuartiges Verfahren skalierbar gehalten, bei dem Multicastpakete, also die von der Quelle gesendeten Datenpakete, die DSMC-Signalisierung auslösen. DSMC greift darüber hinaus nicht wie andere Verfahren in das Multicastrouting ein und ist dadurch für beliebige Multicastroutingprotokolle verwendbar.

In den folgenden Abschnitten wird DSMC im Detail vorgestellt. Nach einem Überblick über DSMC und dessen Architektur in Abschnitt 4.1 wird anschließend mit Abschnitt 4.2 zuerst der zur Lösung des NRS-Problems im MFC eingeführte Zustand beschrieben. Danach wird die zu seiner Konfiguration entworfene Signalisierung zwischen Router und Ressourcenmanager erläutert. Die Anwendung von DSMC auf die standardisierten Multicastroutingprotokolle wird an Beispielen erklärt und es werden Designalternativen diskutiert. Abschnitt 4.3 entwirft eine Erweiterung von DSMC, um die Effizienz der DSMC-Signalisierung auch bei PIM-SM mit seinen zwei Verteilbäumen zu gewährleisten. Abschließend wird in Abschnitt 4.4 die Nutzung von DSMC durch einen Ressourcenmanager erläutert.

4.1 Überblick und Architektur

Wie im letzten Kapitel erläutert wurde, muss bei dienstgüteunterstütztem IP-Multicast auf Basis von Diffserv das NRS-Problem gelöst werden, dass also keine unkontrollierte Ressourcennutzung durch unkontrollierte Paketduplizierung im Inneren einer Diffservdomäne stattfinden kann. Ein bedeutender Aspekt und wesentliches Alleinstellungsmerkmal von DSMC gegenüber den im letzten Kapitel vorgestellten Verfahren ist, dass DSMC nicht wie diese auf Ebene des Multicastroutings, sondern auf tiefer gelegener *Ebene der Multicastweiterleitung* ansetzt, wodurch DSMC unabhängig vom eigentlichen Multicastrouting und den dort eingesetzten Protokollen bleibt. Das NRS-Problem wird so für eine Vielzahl von Multicastroutingprotokollen gelöst, ohne jedes Multicastroutingprotokoll einzeln anpassen zu müssen. DSMC ist damit anders als die im letzten Kapitel vorgestellten Verfahren

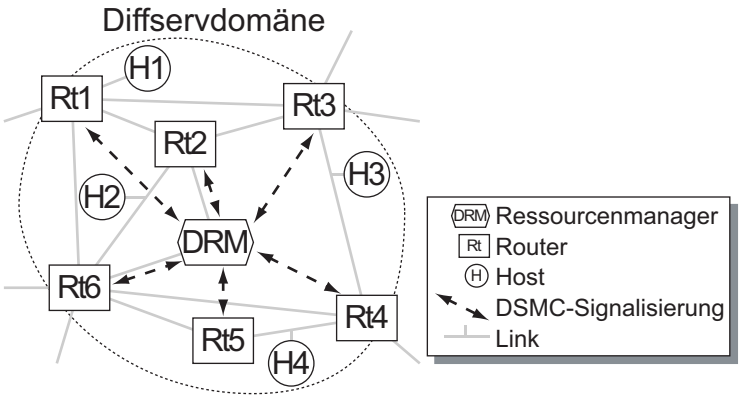


Abbildung 4.1 Jeder DSMC-fähige Router signalisiert direkt zum zentralen Ressourcenmanager seiner Domäne (und umgekehrt).

auch für Multicastroutingprotokolle wie MOSPF ohne eine Join- oder eine vergleichbare Nachricht geeignet.

Der Zustand, den DSMC zur Lösung des NRS-Problems zusätzlich in der Weiterleitungsebene, im MFC, etabliert, wird initial vom Router so gesetzt, dass es nicht zur unkontrollierten Ressourcennutzung kommt. In einem zweiten Schritt wird der Zustand vom Ressourcenmanager der Domäne, im Folgenden kurz DRM genannt, zwecks Dienstgüteerbringung endgültig konfiguriert. Dieser Zustand wird im Folgenden als *NRS-Zustand* bezeichnet. Er gibt an, ob weitergeleitete Multicastpakete ihren DSCP beibehalten dürfen oder ob sie aufgrund von Ressourcenmangel auf dem folgenden Link ummarkiert werden müssen. Der DRM konfiguriert den NRS-Zustand auf einem DSMC-fähigen Router ohne Mitwirkung des Multicastroutings. Der DRM kommuniziert dazu direkt mit jedem DSMC-fähigen Router in seiner Domäne. Es muss aber keine direkte physikalische Verbindung zwischen DRM und jedem Router bestehen, da DSMC-Nachrichten per Unicast über IP gesendet werden. Im Rahmen von DSMC findet keine Kommunikation zwischen DRM und den Endsystemen, den Hosts, statt. Diese erfolgt ausschließlich im Rahmen der Dienstgütesignalisierung, die von DSMC nicht modifiziert wird. Abbildung 4.1 veranschaulicht dies noch einmal graphisch. Dargestellt ist eine Diffservdomäne mit einem DRM, fünf Routern und vier Hosts. Die DSMC-Signalisierung (gestrichelte Pfeile) findet zwischen DRM und Routern statt, nicht zwischen DRM und Hosts. Der DRM ist beliebig in das Netz der Domäne integriert, im Beispiel über Links zu den Routern Rt2 und Rt6. Die DSMC-Signalisierung zu den übrigen Routern wird von diesen dann weitergeleitet.

Abbildung 4.2 zeigt die typischen Komponenten und Kommunikationspfade eines Routers. Für DSMC relevante Kommunikationspfade sind mit dicken Pfeilen dargestellt, neu hinzugefügte Teile sind mit fetter Schrift hervorgehoben. Ein Router gliedert sich in die Weiterleitungs-, Routing- und Managementebene. Die Weiterleitungsebene übernimmt die Paketweiterleitung auf Basis von FIB und MFC. Sie sind das Ergebnis des Routings und werden entsprechend von der Routingebene aus verwaltet. Die Managementebene konfiguriert, steuert und überwacht alle Komponenten eines Routers. Sie stellt über CLI oder SNMP die Schnittstelle für das Netzwerkmanagement bereit. In der Managementebene angesiedelt ist auch die mit DSMC neu eingeführte *DSMC-Komponente*. Sie wickelt die Kommunikation mit dem DRM ab. Der DRM kontrolliert durch sie den im MFC neu eingeführten Zustand (NRS-Zustand, DRbit, Sbit). Die DSMC-Komponente wertet eine vom DRM eintreffende

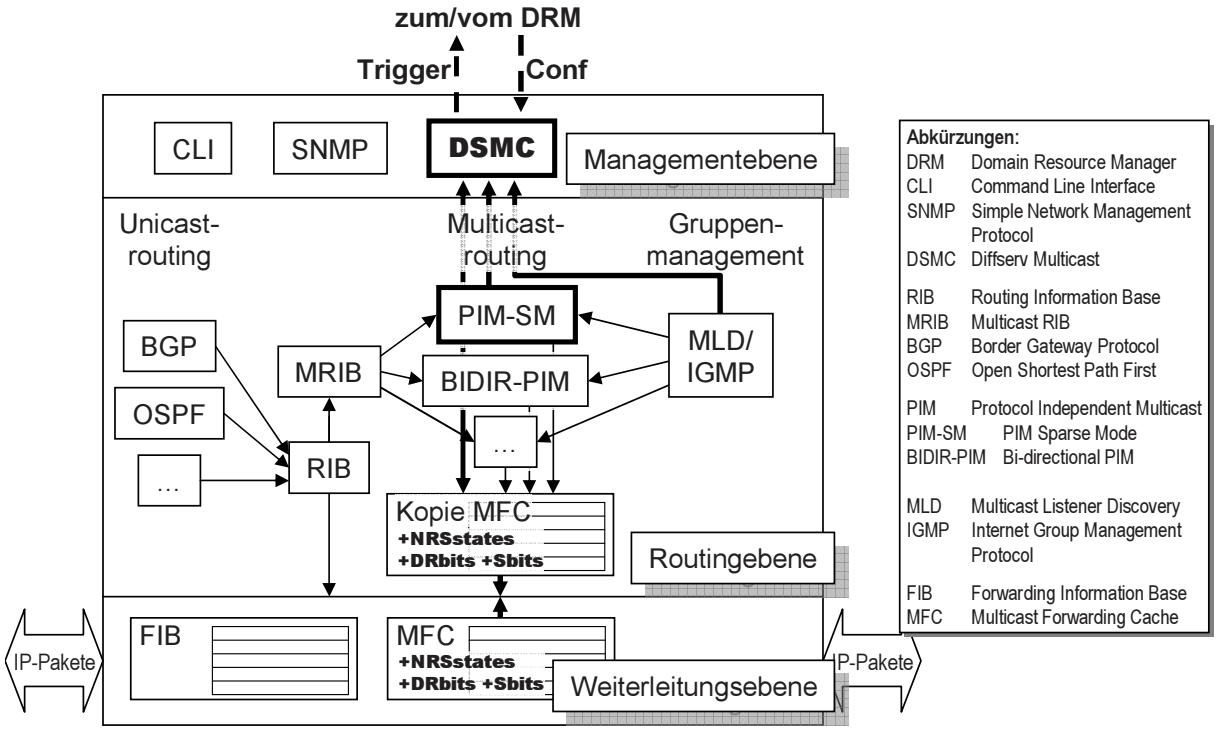


Abbildung 4.2 Vereinfachtes Modell eines Multicastrouters mit DSMC-Fähigkeit. Für DSMC relevante Teile sind fett hervorgehoben.

DSMC-Nachricht (*Conf*) aus und konfiguriert entsprechend den neuen Zustand im MFC. In anderer Richtung wird die DSMC-Komponente bei Weiterleitung von Multicastpaketen unter vom DRM noch nicht konfiguriertem Zustand von der Weiterleitungsebene informiert. Die DSMC-Komponente sendet daraufhin eine DSMC-Nachricht zum DRM (*Trigger*), was die endgültige Konfiguration des Zustands durch den DRM auslöst – daher der Name Trigger. Wann ein Trigger und was für ein Triggertyp gesendet wird, wird von der lokalen Gruppenmitgliedschaft und nicht vom Multicastrouting beeinflusst. Informationen über lokale Gruppenmitglieder erhält die DSMC-Komponente durch Abfrage der MLD/IGMP-Komponente (dicker Pfeil). Speziell für PIM-SM besitzt DSMC eine Erweiterung, die eine minimale Modifikation an der PIM-SM-Komponente – nicht jedoch am PIM-SM-Routing – erfordert. Mit der Erweiterung für PIM-SM wird die DSMC-Komponente zusätzlich noch von dieser bei Auftreten eines bestimmten Routingzustands informiert, was das Senden eines speziellen Triggers auslöst.

DSMC erweitert jeden einzelnen Eintrag im MFC MFCentry(S,G) um N NRS-Zustände *NRSstate(S,G,Oif)* und N Designated-Router-Bits *DRbit(S,G,Oif)*, jeweils einen für jede der N (Ausgangs-)Schnittstellen *Oif* eines Routers (s. Abbildung 4.3). Beide sind genau wie das Activebit(S,G,Oif) auch schnittstellenspezifisch. Mit der Erweiterung für PIM-SM kommt als drittes das nicht schnittstellenspezifische Switchover-Bit *Sbit(S,G)* hinzu. Der NRS-Zustand NRSstate(S,G,Oif) bestimmt, ob Multicastpakete der Quelle S an die Gruppe G, die über die Schnittstelle Oif weitergeleitet werden, ihren DSCP behalten können. Oder ob Pakete, um das NRS-Problem zu lösen, auf einen DSCP ummarkiert werden müssen, der ein niederpriores PHB selektiert. Während die für die jeweilige

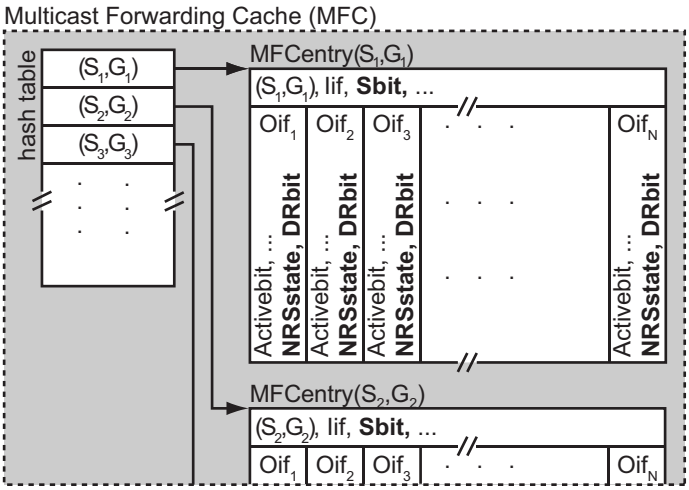


Abbildung 4.3 DSMC erweitert jeden MFC-Eintrag um NRS-Zustand, Designated-Router-Bit und Switchover-Bit.

Multicastgruppe zuständige Multicastroutingkomponente bestimmt, ob eine Schnittstelle aktiv ist, übernimmt die Verwaltung des NRS-Zustands der DRM mit Hilfe der DSMC-Komponente. Das Designated-Router-Bit DRbit(S,G,Oif) vermerkt, ob der über den Ausgang Oif erreichte Link zum Zeitpunkt des Aktivierens (Setzen des Activebit(S,G,Oif)) über lokale Gruppenmitglieder für Gruppe G verfügt, die Verkehr der Quelle S empfangen wollen. Das Designated-Router-Bit wird initial vom Router selbst gesetzt, kann aber später zusammen mit dem NRS-Zustand vom DRM geändert werden. Das Sbit(S,G) wird weiter unten im Zusammenhang mit der Erweiterung für PIM-SM erläutert.

Ein DSMC-fähiger Router informiert seinen DRM mit einem Trigger erst dann, wenn er Multicastpakete unter vom DRM noch nicht konfiguriertem NRS-Zustand – im Folgenden als *ungültiger* NRS-Zustand bezeichnet – weiterleitet. Mit Hilfe der Trigger erfährt der DRM, dass ein neuer Empfänger einer Multicastgruppe beigetreten ist und zu diesem vom Multicastrouting zwar ein neuer Weiterleitungspfad eingerichtet wurde, der entsprechende NRS-Zustand entlang des neuen Pfads jedoch noch ungültig ist. Die Router fordern den NRS-Zustand damit erst bei Weiterleitung von Multicastpaketen an, wenn der NRS-Zustand das erste Mal benötigt wird. Dies hat den Vorteil, dass der Ressourcenmanager erst jetzt die Ressourcenverfügbarkeit prüfen und Ressourcen als belegt vermerken muss. Ressourcen werden folglich erst dann reserviert, wenn sie tatsächlich genutzt werden.

Prinzipieller Ablauf – Basisverfahren

Um das Prinzip von DSMC zu verdeutlichen, soll hier der Ablauf zur Konfiguration des NRS-Zustands in etwas vereinfachter Form anhand einer Topologie aus drei Routern Rt1 bis Rt3, einer Quelle S und den beiden Empfängern R1 und R2 erläutert werden (Abbildung 4.4). DSMC-Signalisierungsnachrichten zwischen Routern und DRM sind gestrichelt dargestellt. Wie der DRM mit der dargestellten Topologie verbunden ist, ist unerheblich. Es wird angenommen, dass über die Dienstgütesignalisierung zwischen Quelle und DRM diesem die für die Quelle zu verwendende Dienstgüte bereits bekannt ist (Schritt 0.). Ebenso ist das konkrete Multicastrouting hier nicht relevant. Es wird angenommen, dass das Multicastrouting nach dem Beitritt des Empfängers R1 (Schritt

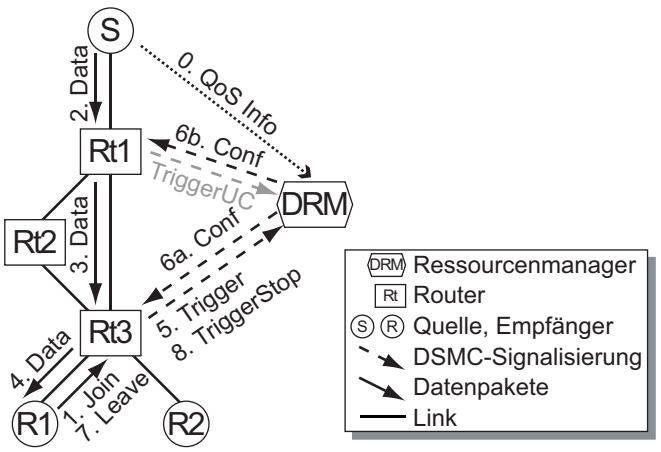


Abbildung 4.4 Prinzipieller Ablauf der Konfiguration des NRS-Zustands.

1.) einen Weiterleitungspfad von der Quelle S bis zum Empfänger R1 hergestellt hat. Die Router Rt1 und Rt3 sollen dabei liegen auf dem Pfad liegen und entsprechend einen Eintrag in ihren Multicastroutingtabellen für die Quelle S und die Multicastgruppe G, an die S ihre Datenpakete („Data“) adressiert, besitzen.

S beginnt zu senden und markiert ihre Datenpakete mit einem DSCP entsprechend der mit dem DRM vereinbarten Dienstgüte. Die Router Rt1 und Rt3 erzeugen beim Eintreffen des ersten Datenpakets aus dem entsprechenden Eintrag ihrer Routingtabelle einen MFC-Eintrag MFCentry(S,G) und leiten die Pakete von S bis zum Empfänger R1 weiter (Schritte 2. bis 4.). Die MFC-Einträge haben zwar das Activebit Activebit(S,G,Oif) gesetzt, jedoch besitzen sie noch keinen gültigen NRS-Zustand NRSstate(S,G,Oif), der daher beim DRM angefordert werden muss (Schritt 5.). Wie gleich noch erläutert wird, genügt es, wenn der letzte Router zum DRM signalisiert. Rt1 sendet daher selbst keinen Trigger, sondern unterdrückt seinen Trigger eine gewisse Zeit lang (genau genommen seinen TriggerUC (s. u.), in der Abbildung grau gezeichnet). Ob ein Router der letzte auf einem neuen Multicastpfad ist, kann mit Hilfe der lokalen Gruppenmitgliedschaft entschieden werden: Nur Rt3 hat einen lokalen Empfänger R1 und daher sendet nur Rt3 (sofort) einen Trigger. Rt3 ist der Router, der die Multicastpakete von S auf den Link mit R1 weitergeleitet. Rt3 ist je nach Multicastroutingprotokoll der Designated Router bzw. im Falle eines Asserts der Assert-Gewinner (PIM-SM, PIM-SSM), Designated Forwarder (BIDIR-PIM, DVMRP) oder Local Forwarder (PIM-DM). Im Folgenden wird dieser letzte Router vor dem Empfänger vereinfachend einheitlich als Designated Router (DR) bezeichnet. Rt3 vermerkt durch Setzen des DRbit(S,G,Oif) an der entsprechenden Ausgangsschnittstelle Oif seines MFC-Eintrags, dass er einen Trigger gesendet hat. Das DRbit wird also nur an der Ausgangsschnittstelle des letzten Routers eines neuen Pfads gesetzt.

Um das NRS-Problem zu lösen, werden die Datenpakete unter ungültigem NRS-Zustand grundsätzlich ummarkiert. Die Datenpakete in Schritt 3. und 4. tragen also vorerst einen DSCP, der ein niederpriores PHB selektiert, wie es z. B. für das Lower Effort PDB verwendet wird. Zur Optimierung der Zuverlässigkeit kann das allererste Datenpaket alternativ auf einen hochprioren DSCP ummarkiert werden. Im Folgenden wird vereinfachend von einem *niederprioren DSCP* gesprochen, wenn der DSCP ein PHB selektiert, das nicht zum NRS-Problem führen kann, anderenfalls von einem *hochprioren DSCP*. Empfänger R1 erhält damit zu Beginn der Kommunikation Datenpakete ohne die von der Quelle

vorgesehene Dienstgüte. Damit trotz Ummarkierens durch Rt1 der nachfolgende Rt3 weiß, dass es sich ursprünglich um einen Datenstrom mit hochpriorem DSCP handelt, nutzt DSMC einen *ausgezeichneten*, auf allen Routern *einheitlich* konfigurierten und nicht anderweitig verwendeten DSCP, im Folgenden als *DSCP_UC* (für „unconfigured“) bezeichnet. Dies ist nötig, da für niederpriore DSCPs, die nicht zum NRS-Problem führen, grundsätzlich kein Trigger gesendet wird. Datenströme mit niederprioren DSCPs werden vom Dienstgütemanagement nicht kontrolliert.

Der Trigger löst die vom DRM mit Conf-Nachrichten durchgeführte Konfiguration des NRS-Zustands aus (Schritte 6a. und 6b.). Der DRM berechnet dazu aus den im Trigger enthaltenen Informationen den neuen Multicastpfad. Ein neuer Multicastpfad beginnt an der Ausgangsschnittstelle des Routers, der bereits auf dem Verteilbaum der Quelle S für Gruppe G liegt und dessen Ausgangsschnittstelle durch den Gruppenbeitritt des neuen Empfängers aktiviert wird. Beim ersten Empfänger beginnt der Pfad bei der Quelle oder, im Falle Rendezvousstellen-basierten Multicastroutings, alternativ bei der Rendezvousstelle. Hier ist R1 der erste Empfänger und der Pfad beginnt bei der Quelle S. Der DRM überprüft daher den Pfad S–Rt1–Rt3–R1 auf Verfügbarkeit genügend freier Ressourcen. Wenn ja, wird der NRS-Zustand der Router so konfiguriert, dass Rt1 und Rt3 zukünftig den DSCP beim Weiterleiten beibehalten und die gewünschte/benötigte Dienstgüte gemäß Schritt 0. wird erbracht. Andernfalls wird der NRS-Zustand so konfiguriert, dass die Datenpakete dauerhaft auf einen zweiten ausgezeichneten DSCP, *DSCP_LE* (für „Lower Effort“), ummarkiert werden, um das NRS-Problem auch weiterhin zu vermeiden. Der neue Empfänger R1 wird damit auch weiterhin keine Dienstgüte erhalten. Der zweite ausgezeichnete DSCP ist notwendig, damit später hinzukommende neue Empfänger und Multicastpfade zu einem Trigger führen. Ein gewöhnlicher niederpriorer DSCP führt, wie bereits weiter oben ausgeführt wurde, zu keinem Trigger. Es ist Sache der DRMs, ob er bei späterem Verfügbarwerden von Ressourcen durch erneute Umkonfiguration des NRS-Zustands Dienstgüte nachträglich bereitstellen will; ebenso, wie es Sache des Empfängers R1 ist, bei fehlender Dienstgüte die Multicastgruppe wieder zu verlassen. In welcher Reihenfolge die Router Rt1 und Rt3 konfiguriert werden, ist von untergeordneter Bedeutung – die Konfiguration aller Router entlang eines neuen Multicastpfads kann parallel erfolgen. Router Rt2 liegt nicht auf dem neuen Pfad, hat keinen MFC-Eintrag und damit auch keinen zu konfigurierenden NRS-Zustand und wird daher in Schritt 6. vom DRM nicht konfiguriert.

Tritt der Empfänger wieder aus der Gruppe aus (Schritt 7.), signalisiert jener Router mit gesetztem Designated-Router-Bit, der auch den Trigger in Schritt 5. gesendet hat, das Zurücksetzen des Activebits mittels einer TriggerStop-Nachricht (Schritt 8.) zum DRM, der daraufhin die reservierten Ressourcen wieder freigibt. Eine erneute Konfiguration der Router wie in Schritt 6. ist nicht erforderlich, da NRS-Zustand und Designated-Router-Bit bei nicht gesetztem Activebit vom Router nicht beachtet bzw. zusammen mit dem MFC-Eintrag gelöscht werden. Kommt es in der Zwischenzeit zu einer Routingänderung, würde z. B. der Verkehr nun über Rt2 laufen, wird der DRM, nachdem ihm die Routingänderung bekannt wird, den NRS-Zustand auf Rt1 und Rt2 mit neuen Conf-Nachrichten konfigurieren, hinreichend Ressourcen entlang des geänderten Pfads S–Rt1–Rt2–Rt3–R1 vorausgesetzt. Der NRS-Zustand auf Rt1 am alten Ausgang Richtung Rt3 wird mit dem Inaktivieren des Ausgangs automatisch verfallen. NRS-Zustand und Designated-Router-Bit auf Rt3 bleiben weiterhin unverändert gültig, da sich der Ausgang nicht geändert hat und der Ausgang auch nicht zwischenzeitlich vorübergehend inaktiv wurde.

Es wäre ineffizient, wenn jeder Router entlang eines neuen Multicastpfads einen Trigger zum DRM sendet. Wie gerade erläutert, kann durch Auswertung der lokalen Gruppenmitgliedschaft die Anzahl der gesendeten Trigger auf eine je neuem Multicastpfad reduziert werden. Einzig der letzte Router vor dem Empfänger, d. h. dessen Designated Router, signalisiert mit einem Trigger zum DRM. Nur er besitzt alle vom DRM für die eindeutige Pfadbestimmung benötigten Informationen: Quell- und Gruppenadresse, der Link, auf dem sich der neue Empfänger befindet, und bei PIM-SM zusätzlich die Art des Multicastbaums – quellenspezifischer kürzester oder Rendezvousstellenbaum –, über den die Pakete zum Empfänger geleitet werden. Er teilt diese Informationen dem DRM mit dem Trigger mit. Der DRM nimmt wie die Router am Link State Unicastrouting (z. B. OSPF) teil, und kann daraus die RIB und die von ihr abgeleitete MRIB berechnen, auf deren Basis vom Multicastrouting die Pfade bestimmt werden. Im Falle von Distance Vector Routing muss das Routing über das Netzwerkmanagement abgefragt und der DRM bei Routingänderungen informiert werden. Der DRM kann so seinerseits aus den im Trigger enthaltenen Informationen den neuen Multicastpfad berechnen und den NRS-Zustand in den Routern entlang dieses Pfads konfigurieren.

Der Trigger zeigt die *vollständig abgeschlossene* Einrichtung des Weiterleitungszustands entlang des *gesamten* neuen Multicastpfads an. Erst ab diesem Zeitpunkt ist die Konfiguration des NRS-Zustands in *allen* Routern entlang des Pfads möglich. Der Trigger hat damit zwei wesentliche Aufgaben: Transport der Topologieinformation zum DRM und Anzeige, dass in jedem Router entlang des Pfads MFC-Eintrag und damit NRS-Zustand zur Konfiguration durch den DRM bereitsteht.

Erweiterung für PIM-SM

Abgesehen von PIM-SM lassen sich mit dem bisher beschriebenen Verfahren alle für DSMC geeigneten Multicastroutingprotokolle unterstützen, d. h. alle Protokolle, die auf Basis der in der MRIB enthaltenen Topologieinformation arbeiten, z. B. die Protokolle der PIM-Familie, oder auch MOSPF, falls OSPF für das Unicastrouting eingesetzt wird. Zwar basiert das Routing von PIM-SM auch auf den Informationen aus der MRIB, jedoch kann PIM-SM als einziges Multicastroutingprotokoll zwei unterschiedliche Pfade verwenden: den Rendezvousstellenbaum und den quellenspezifischen kürzesten Baum. Wann von dem einen auf den anderen Baum umgeschaltet wird, bestimmt bei PIM-SM der Designated Router, der seine Entscheidung von der aktuellen Datenrate der Quelle abhängig macht. Diese Rate entzieht sich jedoch der Kenntnis des DRMs und damit der genaue Umschaltzeitpunkt, so dass er den von PIM-SM zu einem gewissen Zeitpunkt verwendeten Baum nicht vorhersagen kann. Dies ist jedoch zur Pfadbestimmung zwingend notwendig.

Für PIM-SM wird daher zusätzlich das Switchover-Bit, kurz *Sbit*, eingeführt. Das Sbit zeigt beim Umschalten vom Rendezvousstellen- auf den quellenspezifischen kürzesten Baum genau jenen *Verzweigungsrouter* auf beiden Verteilbäumen an, an dem beide Bäume von der Quelle bzw. der Rendezvousstelle aus gesehen aufeinandertreffen, um danach im weiteren Verlauf denselben Verteilbaum zu nutzen. Umgekehrt aus Sicht der Empfänger ist dies jener Router, ab dem beide Bäume beginnen zu divergieren. Dieses Wissen besitzt im Router nur die PIM-SM-Komponente. Sie muss daher so erweitert werden, dass sie das Sbit im entsprechenden Fall setzt. Dies ist die einzige Modifikation, die an der PIM-SM-Komponente vorgenommen werden muss. Insbesondere ändert sich weder der interne Routingzustand von PIM-SM noch das nach außen sichtbare Verhalten, das Protokoll PIM-SM wird nicht

modifiziert. Im Gegensatz zum DRbit ist das Sbit nicht schnittstellenspezifisch, denn es identifiziert nicht wie das DRbit einen Link (jenen mit Empfänger) sondern einen Router, einen Verzweigungsrouter.

Das Sbit beeinflusst genau wie die lokale Gruppenmitgliedschaft, wann eine DSMC-Nachricht an den DRM gesendet wird. Ist es gesetzt, wird mit dem ersten Empfang von Datenpaketen über den quellenspezifischen kürzesten Baum, nachfolgend kurz SPT, der neue nur bei PIM-SM verwendete *TriggerSwitched* gesendet. Wie beim Trigger sendet so nur genau ein Router den TriggerSwitched, nämlich genau der, an dem der SPT vom Rendezvousstellenbaum, kurz RPT, divergiert. Der Empfang des TriggerSwitched wird vom DRM durch ein *TriggerSwitchedAck* bestätigt und das Sbit von der DSMC-Komponente daraufhin zurückgesetzt. Weiterhin muss bei PIM-SM berücksichtigt werden, dass initial nicht nur der Rendezvousstellenbaum, sondern auch direkt ohne vorhergehendes Umschalten der SPT verwendet werden kann. Im Fall des SPT wird wie gewöhnlich ein Trigger an den DRM gesendet, im Fall des RPT der ebenfalls nur für PIM-SM neu eingeführte *TriggerRpt*. Die DSMC-Komponente fragt dazu neben der MLD/IGMP-Komponente zusätzlich die PIM-SM-Komponente ab (Wert des SPTbit(S,G)), um den verwendeten Verteilbaum und damit den richtigen Triggertyp zu bestimmen. Darüber hinaus findet keine Interaktion zwischen der DSMC- und der PIM-SM-Komponente statt.

Behandlung von Inkonsistenzen

Die Sicht des DRMs auf seine Domäne und der tatsächliche Zustand des Routings können sich unterscheiden, z. B. während der Konvergenzphase nach einer Routingänderung durch den Ausfall von Links, Schnittstellen oder ganzen Routern. Infolgedessen können Datenpakete einem anderen Pfad folgen als es der DRM annimmt.

Um solche Inkonsistenzen zu erkennen, definiert DSMC eine weitere Ausprägung des Triggers, *TriggerUC* („unconfigured“) genannt. Er ähnelt dem Trigger, jedoch an Stelle der Bedingung, einen Trigger nur bei Weiterleitung an direkt angeschlossene Empfänger zu senden, tritt beim TriggerUC die Bedingung, dass der Router beim Weiterleiten das Datenpaket auf DSCP_UC *ummarkieren* musste, das eintreffende Paket also noch nicht mit DSCP_UC markiert war. Dieser Fall tritt nur bei genau einem Router auf, dem Router, an dem sich der vom DRM angenommene und mit NRS-Zustand konfigurierte Pfad und der tatsächliche verzweigen. Der erste Router ohne gültigen NRS-Zustand informiert mit dem DSCP_UC alle weiteren Router entlang des abweichenden Pfads darüber, dass die fehlende Konfiguration bzw. Inkonsistenz bekannt ist und der DRM von ihm informiert wird. Mit DSCP_UC markierte werden niemals ummarkiert, so dass ggf. konfigurierter NRS-Zustand auf dem abweichenden Pfad ohne Belang ist. Mit Hilfe des DSCP_UC gelingt es, die Signalisierung von DSMC auch im Falle von Inkonsistenzen von einem Trigger je Router entlang eines von der Inkonsistenz betroffenen Multicastpfads auf einen Trigger je Pfad zu reduzieren.

Solche Inkonsistenzen sind normalerweise transienter Natur. Wenn das Routing erneut konvergiert ist, sind auch die Inkonsistenzen behoben und der DRM kann entlang des geänderten Pfads den NRS-Zustand konfigurieren. Um während solcher kurzzeitiger Inkonsistenzen nicht unnötig TriggerUCs zu generieren, werden die TriggerUCs anfangs für eine konfigurierbar lange Zeitdauer unterdrückt. Diese Dauer sollte idealerweise groß genug sein, das Routing konvergieren zu lassen und dem DRM danach genügend Zeit bleibt, die Router zu konfigurieren. Andererseits sollte sie nicht zu groß gewählt werden, so dass dauerhafte

Inkonsistenzen, wie sie u. a. durch den Verlust von DSMC-Nachrichten entstehen, noch zeitnah erkannt werden können und darauf reagiert werden kann.

Die anfängliche Unterdrückung der TriggerUC erfüllt darüber hinaus noch einen weiteren Zweck. Beginnt eine Quelle mit dem Senden an eine Multicastgruppe, muss zwangsläufig der erste Router hinter der Quelle die Datenpakete auf DSCP_UC ummarkieren, da er mit dem Beginn des Sendens der Quelle für sie einen MFC-Eintrag anlegt und ein neuer MFC-Eintrag initial keinen gültigen NRS-Zustand besitzt. Im Beispiel in Abbildung 4.4 unterdrückt Rt1 daher seinen TriggerUC, wenn er die in Schritt 2. empfangenen Datenpakete der Quelle in Schritt 3. nach DSCP_UC ummarkiert weiterleitet. Kurze Zeit später, wenn die Datenpakete den Designated Router eines Empfängers erreichen, im obigen Beispiel Router Rt3, wird bei diesem das Weiterleiten an den direkt angeschlossenen Empfänger (Schritt 4.) einen Trigger auslösen (Schritt 5.). Die anfängliche Unterdrückung der TriggerUC in Rt1 verhindert so, dass unnötig TriggerUC-Nachrichten generiert werden, obwohl kurze Zeit später, angefordert durch den Trigger von Rt3, der NRS-Zustand auch in Rt1 konfiguriert wird (Schritt 6b.).

Grenzrouter

DSMC modifiziert nicht das Multicastrouting, wodurch es auch inkrementell eingesetzt werden kann. Da Router ohne DSMC („Legacy-Router“) keine Trigger senden, muss dies der letzte DSMC-fähige Router vor einem an einen Legacy-Router angeschlossenen Empfänger übernehmen. Dazu wird die entsprechende Ausgangsschnittstelle des letzten DSMC-fähigen Routers als *DSMC-Grenzschnittstelle* konfiguriert. Ein Router sendet wie solche Schnittstellen grundsätzlich einen Trigger, unabhängig davon, ob es lokale Empfänger gibt oder nicht. Um zu vermeiden, dass es stromabwärts des Grenzrouters zum NRS-Problem kommt, darf an diesem Router nur dann das Ummarkieren auf einen niederprioren DSCP aufgehoben werden, wenn sichergestellt ist, dass auf *allen* möglichen Pfaden stromabwärts hinreichend Ressourcen vorhanden sind. Die Ressourcenüberprüfung durch DRM muss also stromabwärts von einem Broadcast ausgehen.

Ebenfalls als DSMC-Grenzschnittstelle konfiguriert werden die in benachbarte Domänen führenden Schnittstellen der Domänengrenzrouter. Die DSMC-Signalisierung findet ausschließlich zwischen DRM und den Routern seiner eigenen Domäne statt, für einen Empfänger außerhalb der eigenen Domäne erhält der DRM daher keinen Trigger. Stattdessen zeigt der Egressrouter dem DRM per Trigger den innerhalb der eigenen Domäne vollständig etablierten Weiterleitungszustand an, der daraufhin vom DRM konfiguriert wird.

Zusammenfassung

Um einen Multicastrouter DSMC-fähig zu machen, sind lediglich folgende Modifikationen nötig (Basisverfahren, s. Abschnitt 4.2):

- *DSMC-Komponente*: Kommunikation mit dem DRM; Auslesen der lokalen Gruppenmitgliedschaft aus der MLD/IGMP-Komponente; Empfang der Nachrichten vom MFC und Schreiben von NRS-Zustand und Designated-Router-Bit in den MFC. Kein eigener Zustand.

- *MFC*: NRS-Zustand NRSstate(S,G,Oif) und Designated-Router-Bit DRbit(S,G,Oif) je Ausgangsschnittstelle Oif je MFC-Eintrag MFCentry(S,G); Ummarkieren von Paketen in Abhängigkeit des NRS-Zustands; Benachrichtigen der DSMC-Komponente über das Weiterleiten von Paketen unter ungültigem NRS-Zustand in Abhängigkeit des Designated-Router-Bits.

Die Konfiguration als DSMC-Grenzschnittstelle erfolgt im Rahmen des Netzwerkmanagements bei der grundlegenden Schnittstellenkonfiguration eines Routers und mit einer entsprechend erweiterten MIB.

Für PIM-SM sind darüber hinaus folgende zusätzliche Modifikationen nötig, um weiterhin mit nur einem Trigger je neuem Empfänger und einem Trigger je Umschaltung zum quellenspezifischen kürzesten Baum auszukommen (Erweiterung für PIM-SM, s. Abschnitt 4.3):

- *PIM-SM-Komponente*: Setzen des Switchover-Bits Sbit(S,G) (über die DSMC-Komponente) beim Setzen des SPTbits(S,G) von PIM-SM in gewissen Fällen.
- *DSMC-Komponente*: Generieren der TriggerSwitched, Zurücksetzen des Switchover-Bits im MFC beim Empfang eines TriggerSwitchedAck.
- *MFC*: Switchover-Bit Sbit(S,G) je MFC-Eintrag MFCentry(S,G); Benachrichtigen der DSMC-Komponente bei Empfang von Paketen unter gesetztem Switchover-Bit.

Neben diesen Modifikationen und der Definition der Zustände NRSstate, DRbit und SBit umfasst DSMC eine Reihe von Nachrichten zur Konfiguration dieser Zustände durch den DRM. Alle von DSMC definierten Nachrichtentypen sind in Tabelle 4.1 zusammengefasst.

Nachrichtentyp	Richtung	Anmerkung
Trigger	Router → DRM	Basisverfahren
TriggerUC	Router → DRM	
TriggerStop	Router → DRM	
Conf	DRM → Router	
ConfReply	Router → DRM	
TriggerRpt	Router → DRM	PIM-SM
TriggerSwitched	Router → DRM	
TriggerSwitchedAck	DRM → Router	

Tabelle 4.1 Liste aller DSMC-Nachrichten.

Nach diesem Überblick soll DSMC, ausgehend von den Routern, detailliert erläutert werden. Das Basisverfahren von DSMC, das bereits die meisten Multicastroutingprotokolle effizient unterstützt, wird im folgenden Abschnitt 4.2 beschrieben, dessen Anwendung auf die für das Internet standardisierten Multicastroutingprotokolle dargestellt und Designalternativen diskutiert. Abschnitt 4.3 erweitert das Basisverfahren für das im Internet am weitesten verbreitete Multicastroutingprotokoll PIM-SM, um dessen Besonderheit von zwei alternativen Verteilbäumen RPT und SPT effizient zu handhaben. Die Realisierung von DSMC seitens des zentralen Ressourcenmanagers erläutert Abschnitt 4.4, wo neben der DSMC-Nachrichtenverarbeitung, Pfadberechnung und Behandlung von Inkonsistenzen auch erläutert wird, was sich beim Dienstgütemanagement unter Einsatz von DSMC

im Vergleich zu Unicast ändert. Die Möglichkeit, DSMC inkrementell einzusetzen, wird beschrieben und auf domänenübergreifenden bzw. internetweiten Multicast eingegangen. Abschließend fasst Abschnitt 4.5 DSMC noch einmal zusammen.

4.2 Basisverfahren

Das Basisverfahren von DSMC gliedert sich in Regeln zur Lösung des NRS-Problems selbst (Abschnitt 4.2.1) und Regeln zur effizienten Signalisierung zwischen Routern und DRM (Abschnitt 4.2.2), um den in den Routern zwecks Lösung des NRS-Problems neu eingeführten Zustand zu konfigurieren.

4.2.1 Regeln zur Lösung des NRS-Problems

Für die im Rahmen von DSMC verwendeten DSCPs gilt:

Codepointregeln:

DSCP **Zwei ausgezeichnete DSCPs.** Es existieren zwei *ausgezeichnete*, innerhalb einer Diffservdomäne *einheitliche* DSCPs, **DSCP_UC** und **DSCP_LE**, die nicht im Rahmen eines regulären Dienstes verwendet werden.

LE **DSCP_UC und DSCP_LE selektieren niederpriores PHB.** Mit DSCP_UC oder DSCP_LE markierte Pakete selektieren dasselbe PHB. Die Dienstgüte ist nicht besser als die von Best Effort bzw. des Default PHBs der Domäne. Wenn das Lower Effort PDB implementiert ist, sollten mit DSCP_UC und DSCP_LE markierte Pakete wie Pakete des Lower Effort PDBs behandelt werden.

Ein DSMC-fähiger Multicastrouter verhält sich bzgl. Routing wie ein konventioneller multicastfähiger Router, ist bei der Multicastweiterleitung jedoch darüber hinaus in der Lage, Datenpakete beim Weiterleiten umzumarkieren, getrennt konfigurierbar für jeden MFC-Eintrag und jede Ausgangsschnittstelle:

Markierungregeln:

Low **Niederpriore DSCPs niemals ummarkieren.** Pakete die mit einem anderen DSCP als DSCP_UC oder DSCP_LE markiert sind und der ein niederpriores PHB nicht besser als Best Effort bzw. das Default PHB selektiert, also Pakete, deren Weiterleitung nicht zum NRS-Problem führen kann, werden niemals ummarkiert. Für diese Pakete gilt der NRS-Zustand nicht.

UC **DSCP_UC niemals ummarkieren.** Mit DSCP_UC markierte Pakete werden niemals ummarkiert.

Init **Init: DSCP_SIG für erstes Paket.** Beim Weiterleiten auf eine Ausgangsschnittstelle mit NRS-Zustand Init behält ein Paket seinen DSCP bei oder wird auf einen beliebigen DSCP, **DSCP_SIG**, für den der NRS-Zustand gilt, ummarkiert. Die Wahl ist implementierungsspezifisch.

Invd **Invalid: DSCP_UC.** Beim Weiterleiten auf eine Ausgangsschnittstelle mit NRS-Zustand Invalid werden alle Pakete auf DSCP_UC ummarkiert.

Rmrk **Remark: DSCP_LE.** Beim Weiterleiten auf eine Ausgangsschnittstelle mit NRS-Zustand Remark werden alle Pakete auf DSCP_LE ummarkiert.

Keep **Keep: DSCP beibehalten.** Beim Weiterleiten auf eine Ausgangsschnittstelle mit NRS-Zustand Keep behalten alle Pakete ihren DSCP bei.

Zustandsregeln 1:

NRS **MFC-Eintrag erweitert um NRS-Zustand je Schnittstelle.** Jeder MFC-Eintrag MFCentry(S,G) wird um einen NRS-Zustand NRSstate(S,G,Oif) je Ausgangsschnittstelle Oif erweitert. Er beeinflusst, ob und wie Pakete beim Weiterleiten ummarkiert (vgl. Markierungsregeln) und ggf. Nachrichten an den DRM (vgl. Nachrichtenregeln Abschnitt 4.2.2) gesendet werden. Ein NRS-Zustand wird nur beim Weiterleiten und damit nur bei *aktiver* Ausgangsschnittstelle (Activebit(S,G,Oif) = TRUE) ausgewertet und gilt nicht für Pakete, für die Regel **Low** gilt.

Jeder **NRSstate(S,G,Oif)** gilt für genau eine Ausgangsschnittstelle Oif und kann einen der vier Zustände **Init**, **Invalid**, **Remark** und **Keep** annehmen.

Die Zustände Init und Invalid werden vom Router gesetzt: Beim Setzen des Activebit(S,G,Oif) (Activebit(S,G,Oif) = TRUE) wechselt NRSstate(S,G,Oif) in den Zustand Init (Ausgangszustand). Nachdem genau ein Paket unter Zustand Init weitergeleitet wurde, wechselt der Zustand auf Invalid. Wird das Activebit(S,G,Oif) zurückgesetzt (Activebit(S,G,Oif) = FALSE), ist der NRS-Zustand für diesen Ausgang ohne Relevanz, NRSstate(S,G,Oif) ist undefiniert. Die Zustände Remark und Keep werden beim Empfang einer Conf-Nachricht durch den DRM gesetzt (vgl. Nachrichtenregeln).

Durch den Router konfigurierter NRS-Zustand Init und Invalid wird als *ungültig* bezeichnet, durch den DRM konfigurierter NRS-Zustand Remark und Keep als *gültig*.

Abbildung 4.5 fasst dies noch einmal zusammen. Die Zustandsregel definiert die vier Zustände sowie die Übergänge zwischen ihnen, die Markierungsregeln die Aktionen beim Weiterleiten. Gemäß Regel **NRS** führt ein Aktivieren einer Ausgangsschnittstelle in den Grundzustand Init und das erste Weiterleiten weiter in den Zustand Invalid. Die jeweils in eckigen Klammern angegeben Aktionen zeigen die beim Weiterleiten („Forward Data“) gesetzten DSCPs. Ein Inaktivieren führt immer zu einem undefinierten NRS-Zustand. Die erst im folgenden Abschnitt mit den Nachrichtenregeln eingeführten weiteren Zustandsübergänge sind noch grau dargestellt.

Erläuterungen

Um das NRS-Problem zu lösen, muss Verkehr solange auf einen niederprioren DSCP gemäß Regel **LE** ummarkiert werden bis sichergestellt ist, dass die Ressourcen auf dem Pfad hinter dem Ausgang ausreichen und die Ummarkierung aufgehoben werden darf. Zur Lösung des NRS-Problems würden daher allein die beiden Zustände Remark und Keep genügen, d. h. allein die Regeln **Keep**, **Rmrk** und die modifizierten Regeln **Low**

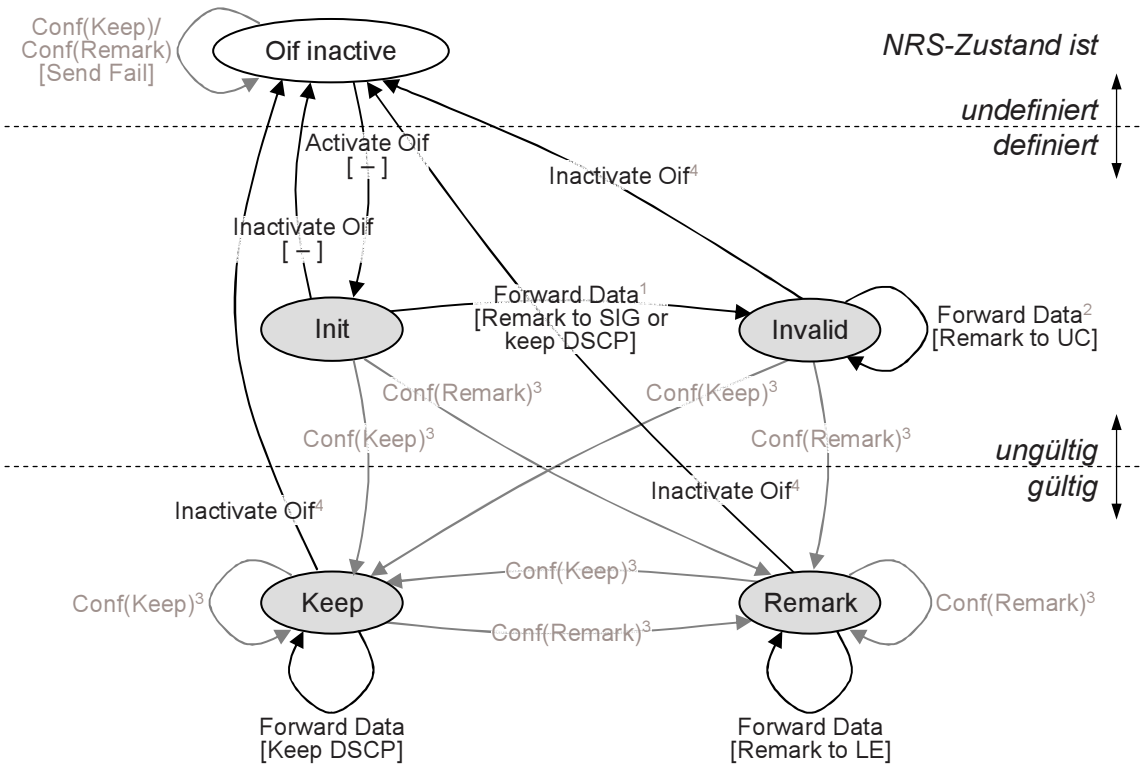


Abbildung 4.5 Zustandsübergangsdiagramm für den NRS-Zustand ohne Signalisierung.

(Ausnahme von DSCP_UC und DSCP_LE entfällt), Regel **NRS** (Zustände Init und Invalid entfallen) und **LE** (DSCP_UC entfällt). Der durch diese fünf Regeln definierte vereinfachte NRS-Zustand ist z. B. im Rahmen von verteiltem Dienstgütemanagement als Alternative zu vollwertigem Diffserv-Policing anwendbar. Man beachte, dass entweder alle hochprioren DSCPs auf LE geändert werden (Regel **Rmrk**) oder keiner geändert wird (Regel **Keep**), denn nach Regel **NRS** differenziert der NRS-Zustand nicht nach DSCPs.

Für die im folgenden Abschnitt 4.2.2 erläuterte DSMC-Signalisierung zur Konfiguration dieser beiden Zustände Keep und Remark werden zwei weitere Zustände Invalid und Init und zwei ausgezeichnete niederpriore DSCPs DSCP_UC („unconfigured“) und DSCP_LE („Limited Effort“) definiert. Neben den hochprioren DSCPs wie DSCP_SIG sind DSCP_UC und DSCP_LE die beiden einzigen niederprioren DSCPs, die die Trigger auslösen, weshalb sie in Regel **Low** ausgenommen sind. Es ist essentiell, dass diese beiden besonderen DSCPs auf allen Routern einer Diffservdomäne *einheitlich* konfiguriert sind, wie von Regel **DSCP** gefordert. Es müssen zudem von keiner Quelle regulär genutzte und von keinem First Hop Router oder Domänengrenzrouter akzeptierte DSCPs sein, damit es umgekehrt nicht versehentlich zum Senden von Triggern kommt. Regel **UC** sorgt dafür, dass mit DSCP_UC markierte Pakete auch dann nicht ummarkiert werden, wenn nach Regel **NRS** der NRS-Zustand ein Ummarkieren nach DSCP_LE anzeigt. Damit ist sichergestellt, dass exakt ein Router auf DSCP_UC ummarkiert und den TriggerUC sendet.

Der DSCP des Default PHBs (Best Effort) und weitere nicht vom Dienstgütemanagement kontrollierte niederpriore DSCPs werden nach Regel **Low** vom NRS-Zustand nicht erfasst und grundsätzlich nicht ummarkiert. Welche der 64 DSCPs von der Ummarkierung ausgenommen sind, kann z. B. in einem Bitfeld global für den Router bzw. alle

MFC-Einträge vermerkt werden. Auch dies muss auf allen Routern einer Diffservdomäne einheitlich konfiguriert sein.

Initial, wenn der NRS-Zustand ungültig (Zustand Init oder Invalid) ist, werden alle Multicastpakete, die zum NRS-Problem führen können, nach Regel **Invd** auf DSCP_UC ummarkiert – ausgenommen das erste Paket. Da es zum Senden des Triggers durch einen Router stromabwärts führen soll, ist es von Vorteil, einen eventuellen Paketverlust aufgrund von knappen Ressourcen so weit wie möglich zu vermeiden. Es besteht daher die Möglichkeit, nach Regel **Init** für das erste Paket einen hochprioren DSCP zu nutzen. Dies kann entweder der originale von der Quelle gesetzte DSCP sein oder ein dedizierter DSCP_SIG, der ein PHB mit geringer Latenz und geringer Verlustwahrscheinlichkeit wie z. B. Class Selector Codepoint 6 (CS6) selektiert und wie er typisch für netzinterne Signalisierungsnachrichten ((Multicast-)Routing etc., vgl. Abschnitt 2.4.1.1) verwendet wird. Da es sich nur um ein einzelnes Paket handelt, kommt es hierdurch nicht zum NRS-Problem. Es kann im Gegenteil davon ausgegangen werden, dass vom DRM ein gewisser Anteil der Ressourcen seiner Domäne fest für Signalisierungszwecke reserviert ist.

Anders als DSCP_UC muss DSCP_SIG kein ausgezeichneter nicht anderweitig verwendeter DSCP sein, da hochpriore DSCPs grundsätzlich vom NRS-Zustand erfasst werden und zum Senden eines Triggers führen, niederpriore DSCPs bis auf die beiden ausgezeichneten DSCPs DSCP_UC und DSCP_LE jedoch nicht. Man beachte, dass netzinterne Signalisierungsnachrichten mit hochpriorem DSCP keinen *gerouteten* Multicast, also keinen Multicast mit einer Reichweite größer als linklokal nutzen dürfen, für den der NRS-Zustand gilt.[1] Andernfalls würde die Signalisierung fälschlich Trigger auslösen. Eventuell für gerouteten Multicast verwendete hochpriore Signalisierungs-DSCPs müssen daher wie niederpriore DSCPs nach Regel **Low** vom NRS-Zustand ausgenommen werden.

Es handelt sich mit DSCP_SIG um eine Optimierung, die ausnutzt, dass das erste Paket das Setzen des DRbits (s. folgender Abschnitt) auslöst und daher ohnehin anders als nachfolgende Pakete behandelt wird. Nimmt man an, dass Datenströme mit hochpriorem DSCP eine hohe Paketrate besitzen, Pakete also schnell aufeinanderfolgen, und ist sichergestellt, dass der Paketverlust von mit DSCP_UC oder DSCP_LE markierten Paketen nicht zu groß wird – so wie es das Lower Effort PDB fordert –, kann DSCP_SIG gleich DSCP_UC oder DSCP_LE gesetzt und wie im Zustand Invalid oder Remark markiert werden.

4.2.2 Regeln zur Konfiguration des NRS-Zustands

Mit den bisher vorgestellten neun Regeln bzw. dem vereinfachten NRS-Zustand mit fünf Regeln lässt sich das NRS-Problem bereits vollständig vermeiden. Wann und wie der NRS-Zustand durch den DRM konfiguriert wird, soll nun erläutert werden. Um die Konfiguration des NRS-Zustands effizient, also mit möglichst wenig Triggernachrichten je neuem Multicastpfad, und ohne Änderungen an den Multicastroutingprotokollen vorzunehmen, werden im Rahmen von DSMC folgende weitere Regeln definiert.

Ein DSMC-fähiger Router informiert den DRM über seinen ungültigen NRS-Zustand mittels sogenannter Trigger-Nachrichten, um von diesem daraufhin mittels Conf-Nachrichten mit gültigem NRS-Zustand konfiguriert zu werden:

[1] Ciscos proprietäres Auto-RP-Protokoll, eine Alternative zum BSR-Mechanismus (s. Abschnitt 2.3.2.4), nutzt z. B. eine Dense-Mode-Multicastgruppe zur Signalisierung.

Nachrichtenregeln:

TrUC **TriggerUC beim Ummarkieren.** Muss ein Paket wegen ungültigen NRS-Zustands beim Weiterleiten *ummarkiert* werden, wird der DRM nach einer Verzögerung, die mit dem Wechsel vom NRS-Zustand Init nach Invalid beginnt, ratenlimitiert mit einem *TriggerUC* über den ungültigen NRSstate(S,G,Oif) informiert, aber nur, wenn für diese Ausgangsschnittstelle das DRbit(S,G,Oif) nicht gesetzt ist.

Tr **Trigger beim Weiterleiten als DR.** Wird ein Paket unter ungültigem NRS-Zustand, also ein Paket, auf das der NRSstate(S,G,Oif) wirkt, an einen *direkt angeschlossenen Empfänger* oder auf einen als *DSMC-Grenzschnittstelle* konfigurierten Ausgang Oif weitergeleitet, oder ist das DRbit(S,G,Oif) gesetzt, wird der DRM ohne Verzögerung ratenlimitiert mit einem *Trigger* über den neuen Empfänger (und den ungültigen NRS-Zustand) informiert.

TrSt **TriggerStop beim Inaktivieren einer Ausgangsschnittstelle.** Wird eine Ausgangsschnittstelle inaktiv, wird der DRM mit einem *TriggerStop* darüber informiert, aber nur, wenn für diese Ausgangsschnittstelle das DRbit(S,G,Oif) gesetzt ist.

Conf **Conf-Nachrichten.** Beim Empfang einer Konfigurationsnachricht *Conf* vom DRM werden der NRSstate(S,G,Oif) und das DRbit(S,G,Oif) gemäß der in der Nachricht enthaltenen Informationen gesetzt, sofern der enthaltene NRS-Zustand gültig[2] ist (Zustand Keep oder Remark).

Rply **ConfReply-Nachrichten.** Fordert die Konfigurationsnachricht durch ein gesetztes ACKbit eine Quittung an und ist die spezifizierte Ausgangsschnittstelle aktiv (Activebit(S,G,Oif) = TRUE), wird durch Senden eines *ConfReply* mit gesetztem ACKbit positiv bestätigt. War es inaktiv (Activebit(S,G,Oif) = FALSE), wird durch ein ConfReply mit zurückgesetztem ACKbit negativ bestätigt, unabhängig davon, ob eine Quittung angefordert wurde oder nicht.

Die Quittung wiederholt bei negativer Quittierung die zu konfigurierenden Werte NRSstate(S,G,Oif) und DRbit(S,G,Oif) aus der Konfigurationsnachricht und bei positiver Quittierung die im MFC aktuell nach dem Conf konfigurierten Werte für NRSstate(S,G,Oif) und DRbit(S,G,Oif). (Im ConfReply wird ACKbit ← Activebit gesetzt und NRSstate und DRbit aus dem Conf übernommen, sofern dort NRSstate gültig war.)

Zustandsregeln 2:

DRbt **MFC-Eintrag erweitert um DRbit je Schnittstelle.**Jeder MFC-Eintrag MFCentry(S,G) wird um ein Designated-Router-Bit DRbit(S,G,Oif) je Ausgangsschnittstelle Oif erweitert. Es beeinflusst, ob und welcher Trigger beim Weiterleiten von Paketen auf den Ausgang und beim Inaktivieren des Ausgangs gesendet werden muss (vgl. Nachrichtenregeln).

[2]Ein Conf mit ungültigem NRSstate wird von DSMC nicht verwendet. Zwecks Debugging können mit ungültigem NRS-Zustand aber die aktuellen Werte von NRSstate und DRbit abgefragt werden, ohne sie zu verändern, vgl. Regel **Rply**.

Nachrichtentyp	(S,G,Oif)	ACKbit	NRSstate	DRbit	Richtung
Trigger	x	-	-	-	Router → DRM
TriggerUC	x	-	-	-	Router → DRM
TriggerStop	x	-	-	-	Router → DRM
Conf	x	x	x	x	DRM → Router
ConfReply	x	x	x	x	Router → DRM

Tabelle 4.2 Alle DSMC-Nachrichtentypen des Basisverfahrens (ohne Erweiterung für PIM-SM).

Nachrichtentyp	ACKbit	NRSstate	DRbit	Notation
Conf	0/1	Keep	0/1	Conf(Keep)
	0/1	Remark	0/1	Conf(Remark)
	1	Keep/Remark	0/1	Conf(Ack)
	0/1	Keep/Remark	1	Conf(DR)
	0/1	Invalid/Init	0/1	nicht verwendet
ConfReply	0	Werte aus Conf		Fail
	1	Werte aus MFC		OK
	1	Keep	0/1	OK_{Keep}
	1	Remark	0/1	OK_{Remark}

Tabelle 4.3 Notation von Conf- und ConfReply-Nachrichten für bestimmte Wertekombinationen.

Jedes Designated-Router-Bit **DRbit(S,G,Oif)** gilt für genau eine Ausgangsschnittstelle Oif. Es ist nur bei *aktiver* Ausgangsschnittstelle (Activebit(S,G,Oif) = TRUE) und NRSstate(S,G,Oif) ≠ Invalid definiert.

Das DRbit(S,G,Oif) wird beim Wechsel des NRSstate(S,G,Oif) von Init nach Invalid gesetzt, wenn nach Regel **Tr** ein Trigger gesendet wird, ansonsten zurückgesetzt. Ein DRbit wird nach Regel **Conf** auch durch ein Conf gesetzt oder zurückgesetzt.

Informationsregeln 1:

TSGO **Jede DSMC-Nachricht enthält Typ & (S,G,Oif).** Neben dem Nachrichtentyp selbst enthalten alle[3] DSMC-Nachrichten das Tripel aus Quelladresse S, Gruppenadresse G und Ausgangsschnittstelle Oif, abgekürzt (S,G,Oif), das NRSstate(S,G,Oif) und DRbit(S,G,Oif) eindeutig identifiziert. Ein quellenunspezifischer MFCentry(*,G) wird durch die unspezifizierte Adresse für S dargestellt.

AND **Conf(Reply) enthält ACKbit, NRSstate und DRbit.** Die DSMC-Nachrichten Conf und ConfReply besitzen das Flag **ACKbit**, mit dem die Quittierung angefordert (Conf) bzw. positiv oder negativ quittiert (ConfReply) wird. Zudem nennen sie die zu konfigurierenden bzw. bei positiver Quittierung die im MFC konfigurierten Werte für NRSstate(S,G,Oif) und DRbit(S,G,Oif).

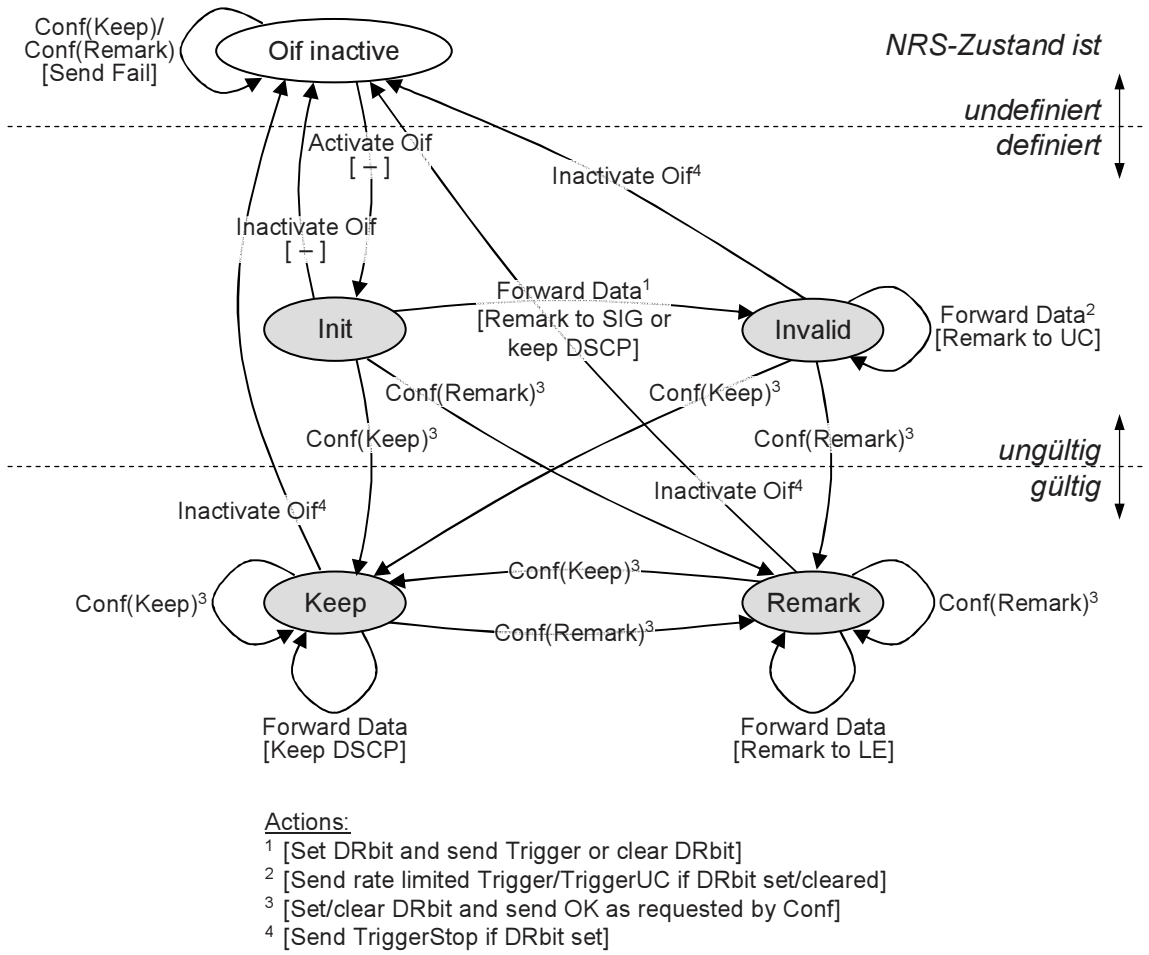


Abbildung 4.6 Zustandsübergangsdiagramm für den NRS-Zustand (Basisverfahren).

Tabelle 4.2 fasst die mit den Nachrichtenregeln definierten DSMC-Nachrichtentypen und die in ihnen gemäß den Informationsregeln minimal enthaltenen Werte noch einmal zusammen. Für Conf- und ConfReply-Nachrichten seien für folgende Typ- und Wertekombinationen die in Tabelle 4.3 aufgeführte Notation vereinbart. Ein hinter dem Conf in Klammern gesetztes Keep, Remark, Ack bzw. DR weist explizit auf den Zustand Keep, Remark, ein gesetztes ACKbit bzw. ein gesetztes DRbit hin. Andernfalls ist der jeweilige Wert aus dem Kontext ersichtlich oder nicht erheblich bzw. das Gesagte gilt für alle Werte. Ein ConfReply wird bei gesetztem ACKbit als OK, bei zurückgesetztem ACKbit oder Fail bezeichnet.

Das von den Regeln geforderte Zusammenwirken von NRS-Zustand und den verschiedenen DSMC-Nachrichten veranschaulicht Abbildung 4.6 graphisch. Die Nachrichtenregeln und die weitere Zustandsregel definieren neue Aktionen, die Regel **Conf** zusätzliche Zustandsübergänge. Im Einzelnen wird beim Übergang von Init nach Invalid bei Vorhandensein eines lokalen Empfängers das DRbit gesetzt und ein Trigger gesendet, ansonsten wird das DRbit zurückgesetzt. Im Zustand Invalid wird dann je nach Wert des DRbits ein Trigger oder TriggerUC ratenlimitiert gesendet. Beim Wechsel nach Keep oder Remark wird das DRbit auf den in der empfangenen Conf-Nachricht enthaltenen Wert gesetzt und das Conf, falls vom Conf anfordert, mit einem ConfReply OK bestätigt. Beim Inaktivieren aus den

[3] TriggerSwitched und TriggerSwitchedAck enthalten kein Oif, vgl. Regel **TSG** Abschnitt 4.3.1.

Zuständen Keep und Remark wird schließlich ein TriggerStop gesendet, sofern das DRbit gesetzt ist. Ein Conf für eine inaktive Ausgangsschnittstelle wird mit einem ConfReply Fail negativ bestätigt.

4.2.3 NRS-Zustände Init und Invalid

Der Zustand Init entkoppelt das Aktivieren und Deaktivieren eines Ausgangs von der Weiterleitung des ersten Pakets. In gewissen Fällen ist dies unnötig. Manche Implementierungen, typischerweise die unixbasierten Softwarerouter wie Linux, FreeBSD, etc., legen einen MFC-Eintrag erst mit dem Empfang des ersten Pakets einer Quelle an. Somit wird ein Ausgang aktiv, der NRS-Zustand befindet sich im Zustand Init, und sofort anschließend wird mit dem Weiterleiten des Pakets in den Zustand Invalid gewechselt. Grund für dieses Verhalten ist, dass ihr MFC keine quellenunspezifischen Einträge MFCentry(*,G) unterstützt. Ein quellenspezifischer Eintrag MFCentry(S,G) kann jedoch nicht sofort mit quellenunspezifischen (*,G)-Routingzustand angelegt werden, da die benötigte Quelladresse S hier nicht bekannt ist. Erst mit dem Eintreffen des ersten Pakets einer Quelle wird sie bekannt und der MFC-Eintrag wird angelegt.

Ist bereits ein Ausgang desselben MFC-Eintrags aktiv und ist dessen NRS-Zustand nicht Init, ist bereits sichergestellt, dass entlang des Pfads bis zum Router Weiterleitungszustand etabliert wurde. Es kann daher beim Aktivieren eines weiteren Ausgangs mit lokalen Empfängern schon mit dem Aktivieren das DRbit gesetzt, ein Trigger gesendet und in den Zustand Invalid gewechselt werden.

In allen anderen Fällen ist der Zustand Init jedoch zwingend erforderlich. wenn also der Weiterleitungszustand zusammen mit dem Routingzustand noch vor Eintreffen des ersten Pakets einer der Quelle angelegt wird, für die der Weiterleitungszustand angelegt wurde. Ohne den Zustand Init, d. h. durch direktes Wechseln in den Zustand Invalid beim Aktivieren eines Ausgangs, würde ein Trigger gesendet werden, bevor der Weiterleitungszustand stromaufwärts sicher etabliert ist. Je nach Routingprotokoll wird der Routingzustand beginnend beim neuen Empfänger in Richtung der Quelle (oder Rendezvousstelle) aufgebaut. Es ist somit nicht mehr sichergestellt, dass der gesamte Pfad etabliert und vom DRM konfiguriert werden kann, wenn er den Trigger erhält.

4.2.4 DRbit & DSMC-Grenzschnittstelle

Von seiner Wirkung auf die DSMC-Signalisierung her ist das DRbit *DRbit(S,G,Oif)* (für Designated Router) ein Schalter. Im NRS-Zustand Invalid und bei einem nicht als DSMC-Grenzschnittstelle konfiguriertem Ausgang Oif entscheidet es, ob bei Weiterleitung zusätzlich nach Regel **TrUC** auf die Bedingung „Ummarkieren“ oder nach Regel **Tr** auf die Bedingung „an einen direkt angeschlossenen Empfänger“ überprüft werden soll. Auch beim späteren Inaktivieren des Ausgangs (oder Löschen des ganzen MFC-Eintrags) wirkt es als Schalter und entscheidet, ob ein TriggerStop nach Regel **TrSt** gesendet werden muss oder nicht.

Von seiner Konzeption her zeigt ein gesetztes DRbit an, dass die Ausgangsschnittstelle Oif *zum Zeitpunkt des Sendens des ersten Triggers* zu einem Link mit lokalen Empfängern führt. Ein gesetztes DRbit ist nicht mit dem Zustand „lokale Empfänger vorhanden“ gleichzusetzen. Besitzt ein Link nicht nur Empfänger sondern auch abhängige Router, bleibt die Ausgangsschnittstelle auch beim Austritt des Empfängers weiterhin aktiviert, da

die anhängigen Router weiterhin mit Multicastverkehr versorgt werden müssen. Solange der Ausgang aktiv ist, bleibt auch das DRbit gesetzt. Empfänger können auf dem Link der Gruppe beitreten und austreten so oft sie wollen. Dies hat keine Auswirkung auf das DRbit, solange sich der Zustand des Activebits nicht ändert. Erst wenn das Activebit zurückgesetzt wird und die Multicastweiterleitung endet, wird ein TriggerStop gesendet. Dieser ist zwar überflüssig, da offensichtlich stromabwärts ein anderer Empfänger existiert, dessen Router ebenfalls einen TriggerStop sendet wird. Schädlich ist er jedoch nicht und bedeutet bei geeigneter Nachrichtenverarbeitung seitens des DRMs auch keinen wesentlichen Mehraufwand (s. 4.4.2)

Es wurde bewusst darauf verzichtet, dass DRbit der lokalen Mitgliedschaft nachzuführen und den DRM beim Setzen des DRbits über die nun vorhandenen lokalen Gruppenmitglieder zu informieren. Dies hätte einerseits Signalisierung bedeutet, die zu jenem Zeitpunkt von Seiten des DRMs gar keiner aktiven Reaktion bedarf, d. h. keiner Umkonfiguration von NRS-Zustand in den Routern. Sollte das DRbit gesetzt und später erneut gelöscht werden, wäre die Signalisierung sogar ganz überflüssig gewesen. Des Weiteren kennt zwar die MLD/IGMP-Komponente die lokale Mitgliedschaft und kann die DSMC-Komponente bei Änderungen informieren. Sie kennt jedoch nicht die in einer Gruppe aktiven Quellen. Ohne Kenntnis der aktiven Quellen können die über das Tupel (S,G) adressierten MFC-Einträge jedoch nicht aufgefunden und deren DRbits aktualisiert werden. Der MFC hätte daher erweitert werden müssen, so dass sich alle MFC-Einträge für eine bestimmte Gruppe auffinden lassen.

Die fehlende Nachführung des DRbits gestattet es dem DRM, den Wert des DRbits nach seinen Wünschen zu steuern und so durch die Wirkung des DRbits als Schalter auf die DSMC-Signalisierung Einfluss zu nehmen. Dies gestattet dem DRM u. a. , bei rendezvousstellenbasierten Multicastroutingprotokollen, die Datenpakete auch ohne Empfänger bis zur Rendezvousstelle fließen lassen, sich vom Designated Router der Quelle per TriggerStop benachrichtigen zu lassen, wenn sie das Senden eingestellt hat. Der DRM kann so frühzeitig belegte Ressourcen wieder freigegeben (s. Abschnitt 4.2.11.3 zu BIDIR-PIM). Das fehlende Nachführen des DRbits resultiert allerdings in einem bestimmten Fall in einem vom DRM nicht erkannten Empfänger. Dieser Fall lässt sich jedoch behandeln, u. a. gerade dadurch, dass das DRbit vom DRM gesteuert werden kann, wie Abschnitt 4.2.13.4 erläutert.

Beim Weiterleiten auf einen als DSMC-Grenzschnittstelle konfigurierten Ausgang mit ungültigem NRS-Zustand werden unabhängig vom DRbit grundsätzlich Trigger gesendet. Die Konfiguration als Grenzschnittstelle erfolgt, wenn mindestens einer der anderen Router am Link entweder nicht DSMC-fähig ist und daher keinen Trigger generieren kann, oder sich ein Router in einer anderen Domäne befindet und von einem anderen Ressourcenmanager kontrolliert wird. Ersteres tritt bei inkrementellem Einsatz von DSMC auf (s. Abschnitt 4.4.7), letzteres bei internetweitem Multicast (s. Abschnitt 4.4.7). Die Konfiguration als DSMC-Grenzschnittstelle wurde nicht in die DSMC-Signalisierung integriert. Sie erfolgt durch ein entsprechend erweitertes MIB-Modul über das Netzwerkmanagement, z. B. durch eine weiteres Bits im Objekt *PimInterfaceEntry* der PIM MIB [164].

4.2.5 Triggernachrichten

Der *Trigger* zeigt die *vollständig abgeschlossene* Einrichtung des Weiterleitungszustands für einen neuen Multicastempfänger an. Ab diesem Zeitpunkt sind entlang des *gesamten* neu eingerichteten Multicastpfads MFC-Einträge angelegt worden und die Konfiguration

des NRS-Zustands kann entlang des gesamten neuen Multicastpfads erfolgen. Es genügt, wenn nur der letzte Router eines neuen Multicastpfads mittels Trigger bzw. TriggerStop zum DRM signalisiert, da hieraus bereits der auf- bzw. abgebaute Pfad ableitbar ist.

Der *TriggerStop* stellt lediglich eine einfache und effiziente Möglichkeit dar, den beginnenden Abbau von Weiterleitungszustand nach Beendigung einer Multicastkommunikation anzuzeigen. Er kann entfallen, sofern das Ende der Gruppenmitgliedschaft anderweitig zur Verfügung steht (vgl. Diskussion von MOSPF in Abschnitt 4.2.12.4) und den Designalternativen in Abschnitt 4.2.13.4). Beim Abbau von Weiterleitungszustand wird der NRS-Zustand automatisch invalidiert bzw. gelöscht, so dass beim Abbau keine weitere Kommunikation zwischen dem DRM und den Routern stattfindet, die wie beim Aufbau durch einen Trigger zeitlich synchronisiert werden müsste.

Ein TriggerStop kann vom Router anders als ein Trigger nicht wiederholt werden, da er das Löschen des Weiterleitungs- und damit des für DSMC im MFC eingeführten Zustands anzeigt. Es existiert ab diesem Zeitpunkt kein Zustand mehr, der für eine Wiederholung der TriggerStop sorgen könnte. Dieser müsste extra für diesen Zweck neu angelegt werden, womit jedoch der Vorteil von DSMC zunichte gemacht würde, dass aller DSMC-spezifische Zustand automatisch mit dem nicht mehr benötigten Weiterleitungszustand gelöscht wird.

TriggerUC – Behandlung von Inkonsistenzen

DSMC setzt voraus, dass sich die Sicht des DRMs auf den Zustand der von ihm verwalteten Domäne vom tatsächlichen Zustand der Domäne nicht unterscheidet. Kommt es hier zu Inkonsistenzen, erlauben *TriggerUC* deren Erkennung. Jener eine Router, ab dem tatsächlicher und vom DRM angenommener Multicastpfad divergieren, wird nach einer gewissen Verzögerung, (s. folgender Abschnitt) einen TriggerUC senden. Das Ummarkieren auf DSCP_UC gemäß Regel **Invd** in Kombination mit der Bedingung des Ummarkierens in Regel **TrUC** verhindert dabei, dass nachfolgende Router ebenfalls einen TriggerUC senden und so möglicherweise eine Flut von gleichzeitigen TriggerUCs generiert wird. Dies verhindert allerdings auch, dass der DRM den vollständigen tatsächlichen Multicastpfad auf einmal erkennen kann, denn dies erfordert einen TriggerUC von jedem betroffenen Router entlang des nicht konfigurierten Pfades.

Eine Bereitstellung von Dienstgüte trotz Inkonsistenz auf Basis des TriggerUC ist trotzdem möglich, allerdings nur nach und nach: Der erste Router fordert mittels TriggerUC seinen NRS-Zustand an, den der DRM konfiguriert. Anschließend muss der nachfolgende Router die Pakete ummarkieren, wird seinerseits einen TriggerUC generieren und vom DRM konfiguriert werden usw., bis tatsächlicher und vom DRM angenommener Multicastpfad wieder zusammenlaufen oder das Ende des Pfads erreicht ist. Die anfängliche Unterdrückung des TriggerUC beginnt bei allen Routern gleichzeitig mit der Weiterleitung des ersten Pakets und dem Wechsel des NRS-Zustands von Init nach Invalid. Nach Ablauf der Unterdrückungszeit sendet jedoch allein der erste Router einen TriggerUC, da nur er ummarkieren muss. Nach seiner Konfiguration muss der nachfolgende Router ummarkieren und sendet sofort einen TriggerUC usw.

Obwohl wie beschrieben die Konfiguration allein auf Basis der TriggerUC möglich ist, ist es sinnvoller, den TriggerUC ausschließlich zur Erkennung von Inkonsistenzen zu verwenden und nach möglichen Ursachen zu differenzieren. Im Falle von verlorenen DSMC-Nachrichten, meist der nur einmal gesendeten TriggerStop, kann die Inkonsistenz vom DRM durch

Anpassung seiner Sicht sofort behoben werden, ggf. durch Senden weiterer Conf-Nachrichten. Bei einer Routingänderung muss er das Ende der Inkonsistenz abwarten und nach erneuter Konvergenz des Routings die neuen Pfade mit NRS-Zustand konfigurieren. In allen anderen Fällen muss von einer vom DRM nicht selbst behebbaren Inkonsistenz ausgegangen werden, z. B. durch fehlerhafte/widersprüchliche statische Konfiguration von Knoten im Netz. Der DRM wird dies gegenüber dem Administrator als Fehler loggen, um die händische Behebung der Ursache zu ermöglichen. In Abschnitt 4.4.4 wird auf Inkonsistenzen noch einmal näher eingegangen. Nur im letzten Fall einer nicht behebbaren Inkonsistenz mag es noch sinnvoll sein, auf Basis der TriggerUC eine Bereitstellung von Dienstgüte zu versuchen. Hierbei muss allerdings beachtet werden, dass bei der Konfiguration mittels Conf(Keep,DR) auch das DRbit gesetzt und so ein TriggerStop beim Abbau des Weiterleitungszustands angefordert wird. Andernfalls wird der Abbau des Pfads erst im Rahmen der periodischen Überprüfung nach T_{check} erkannt und die reservierten Ressourcen werden erst dann wieder freigegeben.

Bei rendezvousstellenbasierten Multicastroutingprotokollen kommt es regelmäßig auch ohne Inkonsistenzen zu einem TriggerUC. Diese Multicastroutingprotokolle leiten in gewissen Fällen auch ohne Empfänger dauerhaft Datenpakete bis zur Rendezvousstelle weiter: PIM-SM, sofern die Rendezvousstelle keine Umschaltung auf den quellenspezifischen kürzesten Baum initiieren will und daher kein Register-Stop sendet, BIDIR-PIM, sofern MFC-Einträge zur Weiterleitung Richtung Rendezvousstelle angelegt werden, und CBT in jedem Fall. Um nach Ablauf der anfänglichen Unterdrückungszeit weitere TriggerUC zu unterbinden, wird der DRM im Regelfall den Pfad von dem den TriggerUC sendenden ersten Router nach der Quelle bis zur Rendezvousstelle mit NRS-Zustand konfigurieren. Bei PIM-SM und CBT erfordert dies nur die Konfiguration auf des ersten Routers selbst, da er die Pakete per Unicast zur Rendezvousstelle tunnelt und Unicastpakete keine Trigger auslösen.

4.2.6 Konfigurationsnachrichten

Trigger lösen Conf-Nachrichten aus, mit denen der DRM den NRS-Zustand entlang eines neuen Multicastpfads konfiguriert. Wie der DRM aus einem empfangenen Trigger und dem nach Regel **TSGO** enthaltenen Tripel (S,G,Oif) den Multicastpfad ableiten kann, erläutert Abschnitt 4.4.3. Was für Conf-Nachrichten an die so bestimmten Router entlang eines neuen Multicastpfads gesendet werden, kann in drei typische Fälle untergliedert werden:

- Bei genügend verfügbaren Ressourcen entlang eines neuen Multicastpfads wird auf allen Routern mittels *Conf(Keep)* der NRS-Zustand auf Keep geändert, so dass der DSCP der Quelle beibehalten wird. Die angeforderte Dienstgüte wird erbracht.

- Andernfalls wird der NRS-Zustand aller Router mittels *Conf(Remark)* auf Remark geändert, Die Pakete der Quelle werden dauerhaft ummarkiert, wodurch die „Dienstgüte“ auf Lower (oder Best) Effort reduziert und Datenströmen mit bereits zugesicherter Dienstgüte dauerhaft keine Ressourcen genommen werden.

- Nur der erste Router des neuen Pfads wird mittels Conf(Remark) zum Ummarkieren angewiesen, alle weiteren Router werden erhalten ein Conf(Keep). Dies führt zum selben Ergebnis wie Fall zwei, erlaubt jedoch mit nur einem Conf(Ack,Keep) (s. u.)

an den ersten Router zum ersten Fall wechseln zu können. Die Dienstgüte kann so später mit nur einem *Conf(Ack,Keep)* an den ersten Router aktiviert und mit *Conf(Ack,Remark)* wieder deaktiviert werden.

Die dritte Variante wird dazu verwendet, Datenpakete von Quellen solange umzumarkieren, solange es für sie keine Empfänger gibt. Erst mit dem Beitritt eines Empfängers wird Dienstgüte bereitgestellt, hinreichend verfügbare Ressourcen vorausgesetzt. Diese Situation tritt jedoch nur bei Multicastroutingprotokollen mit Rendezvousstellen auf (s. Beispiel zu BIDIR-PIM in Abschnitt 4.2.11.3 und PIM-SM in Abschnitt 4.3.4). Des Weiteren können so Ausnahmesituationen, wenn die eigentlich zugesicherte Dienstgüte z. B. durch Ausfall von Komponenten im Netz und dadurch geringerer verfügbarer Bandbreite längerfristig nicht mehr erbracht werden kann, mit geringerem Signalisierungsaufwand behandelt werden als mit einer vollständigen Umkonfiguration von Pfaden wie in den ersten zwei Varianten.

4.2.7 Nachrichtenverlust

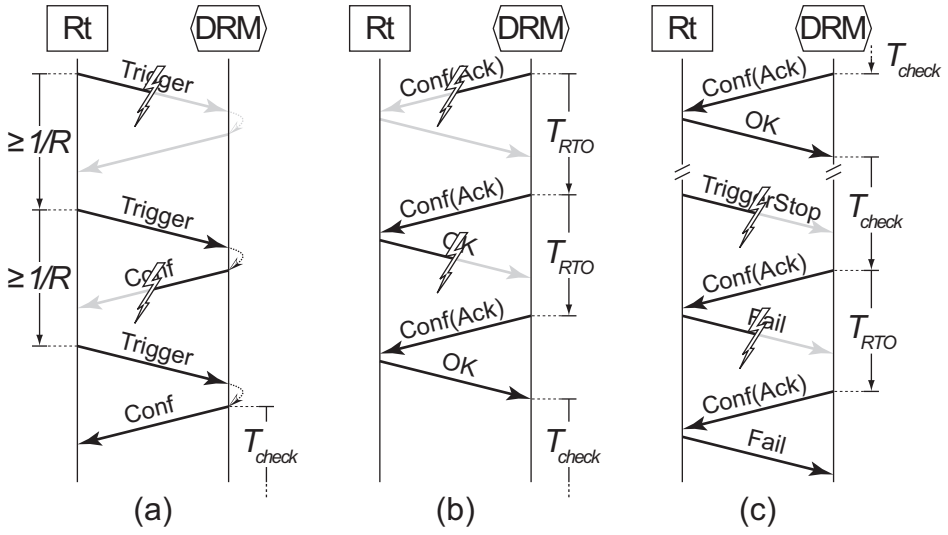


Abbildung 4.7 Behandlung verlorener DSMC-Nachrichten (a) beim ersten Konfigurieren, (b) späterem Ändern und (c) abschließendem Löschen von NRS-Zustand.

Geht eine Trigger- oder Conf-Nachricht verloren, sorgt der weiterhin ungültige NRS-Zustand für das weiterhin periodische Senden eines Triggers, bis der NRS-Zustand vom DRM erfolgreich konfiguriert wurde (s. Abbildung 4.7 (a)). Egal, ob bereits der Trigger verloren geht (oben) oder nur das Conf (Mitte), es wird in beiden Fällen der Trigger weiterhin gesendet. Erst, wenn aufgrund eines Conf der NRS-Zustand gültig wird, stoppt das Senden der Trigger (unten). Die Senderate R der Trigger wird gemäß Gleichung 4.4 auf Seite 106 so begrenzt, dass neben der Zeit zum Übertragen der Nachrichten dem DRM genügend Zeit zum Überprüfen und Belegen der Ressourcen bleibt (in der Abbildung angedeutet durch den gebogenen Pfeil vom Trigger zum Conf). Abschnitt 4.2.10 geht näher auf die geeignete Wahl der maximalen Senderate R ein. T_{check} wird weiter unten zusammen mit Abbildungsteil (c) erläutert.

Trigger werden nicht exakt periodisch mit der Rate R gesendet, sondern mit dem nächsten weitergeleiteten Paket nach Ablauf der Periodendauer $1/R$. In der Abbildung ist dies durch

das Größergleich-Zeichen kenntlich gemacht. Grund ist, dass nach Regel **Tr** Trigger ausschließlich bei der Weiterleitung eines Pakets gesendet werden. Dies gilt ebenfalls für die TriggerUC gemäß Regel **TrUC**. Die Rate der weitergeleiteten Multicastpakete ist verglichen mit der Rate R der Trigger jedoch klein, so dass hier vereinfachend von periodischer Wiederholung gesprochen werden soll. Letztlich ist für die korrekte Funktion von DSMC die konkrete Paketrate unerheblich. Auch wenn gar kein Paket und damit gar kein Trigger mehr gesendet werden würde, stellt die kein Problem dar, da in diesem Fall eine Konfiguration des NRS-Zustands ohnehin überflüssig ist.

Es wurde auf das grundsätzliche Quittieren von Conf-Nachrichten verzichtet. In Abbildungsteil (a) Mitte hätte durch Quittierung zwar das verlorene Conf vom DRM schneller erkannt und behandelt werden können, da $1/R$ größer als die Zeit T_{RTO} (s. u.) ist, nach der ein Conf bei ausbleibender Quittung vom DRM wiederholt wird. Der Verlust einer DSMC-Nachricht stellt bei geeigneter Dienstgüteunterstützung für die Signalisierung jedoch die Ausnahme dar (vgl. auch die Erläuterungen zu Verzögerung und Ratenlimitierung der Trigger unten), so dass der beim ersten und oft einzigen Konfigurieren durch ein Quittieren verursachte zusätzliche Signalisierungsverkehr in den meisten Fällen unnötig ist. Bei einem neuen Pfad mit n zu konfigurierenden Routern bzw. NRS-Zuständen würden statt einem Trigger und n Conf und einem TriggerStop zusätzlich n ConfReply übertragen werden, die Gesamtzahl der DSMC-Nachrichten steigt um 33 % bis 100 % ($2 > (1\,\text{Trigger} + n\,\text{Conf} + n\,\text{ConfReply} + 1\,\text{TriggerStop})/(1\,\text{Trigger} + n\,\text{Conf} + 1\,\text{TriggerStop}) \geq 4/3$, für $n \geq 1$).

Ein Quittieren ist erst dann erforderlich, wenn bereits gültiger NRS-Zustand konfiguriert ist dieser verändert werden soll. Unter konfiguriertem NRS-Zustand wiederholt ein Router keine Trigger mehr und verlorene Nachrichten können nach dem Schema aus Abbildungsteil (a) nicht mehr behandelt werden. Stattdessen lässt sich der DRM den Empfang seines Conf vom Router per *ConfReply* mit gesetztem ACKbit, kurz *OK*, bestätigen (Abbildung 4.7 (b)). Das *Conf(Ack)* hat dazu im Unterschied zum Conf aus Abbildungsteil (a) das ACKbit gesetzt. Empfängt der DRM kein ConfReply innerhalb der erwarteten Zeit T_{RTO} (Retransmit Timeout), wiederholt er das letzte Conf(Ack). Wie T_{RTO} gewählt wird, beschreibt Abschnitt 4.2.9. Auch hier ist es egal, ob bereits das Conf(Ack) (oben) oder erst das OK verloren geht (Mitte). Die periodische Wiederholung der Conf(Ack) endet erst beim Eintreffen eines ConfReply mit denselben Werten für (S,G,Oif), NRS-Zustand und DRbit (und vom korrekten Router). Dies kann auch ein ConfReply mit nicht gesetztem ACKbit, d. h. ein *Fail* sein. Dann wurde zwischenzeitlich der Weiterleitungszustand und damit der NRS-Zustand gelöscht.

Ein ConfReply mit nicht gesetztem ACKbit, *Fail*, tritt genau dann auf, wenn der DRM versucht, den NRS-Zustand von angenommenen Weiterleitungszustand (gesetztes Activebit) mittels Conf(Ack) zu konfigurieren, dieser aber gar nicht mehr existiert. Ein Fail zeigt damit eine Inkonsistenz zwischen der Sicht des DRMs und dem tatsächlichen Weiterleitungszustand bzw. Routing in seiner Domäne an. Eine Ursache für die Inkonsistenz ist der Verlust der nur einmal gesendeten TriggerStop (Abbildung 4.7 (c)). Während beim initialen Einrichten von NRS-Zustand wie in Abbildungsteil (a) der Verlust von DSMC-Nachrichten durch periodisches Wiederholen der Trigger behoben wird, muss beim Invalidieren/Löschen des NRS-Zustands umgekehrt der DRM durch periodische Überprüfung mittels Conf(Ack) einen eventuellen Verlust von TriggerStop beheben.

Die Überprüfung per Conf(Ack) beginnt mit dem ersten Einrichten von NRS-Zustand (Senden des Conf unten in Abbildungsteil (a)) und endet mit dem Empfang eines Trig-

gerStop oder Fail (Abbildungsteil (c) unten) oder wenn ein TriggerStop eines anderen Routers den Abbau anzeigt (letzteres nur bei optimistischer Freigabe belegter Ressourcen, s. „ConfReply – Fail“ in Abschnitt 4.4.2). Hat der DRM für eine gewisse Zeit T_{check} von einem Router für ein Tripel (S,G,Oif) keine Nachricht mehr empfangen, wiederholt der DRM zur Überprüfung seine zuletzt gesendete Konfigurationsnachricht (und mit gesetztem ACKbit). Wird innerhalb T_{RTO} kein ConfReply, OK oder Fail, empfangen, muss entweder das Conf oder das ConfReply verloren gegangen sein. Das Conf(Ack) wird daher solange im Abstand T_{RTO} wiederholt, bis ein ConfReply eingeht (Abbildungsteil (c) Mitte). Das Vorgehen entspricht hier exakt dem aus Abbildungsteil (b) beim Ändern bereits konfigurierten NRS-Zustands. Empfängt der DRM ein OK, war die Überprüfung erfolgreich und der DRM startet die Zeit T_{check} mit dem Empfang des OK neu (Abbildungsteil (b) unten). Wird dagegen ein Fail empfangen (Abbildungsteil (c) unten), existiert offensichtlich der Weiterleitungszustand und damit der NRS-Zustand nicht mehr und der Ressourcenmanager passt seine interne Ressourcenbelegung entsprechend an.

In Abwesenheit von anderweitigen Nachrichten zur Änderung des NRS-Zustands erfolgt die Überprüfung somit im Abstand von T_{check} zuzüglich der Zeit vom Senden des Conf(Ack) bis zum Eintreffen des bestätigenden OK (sofern kein Paketverlust auftritt maximal T_{RTO}). Die Zeitspanne zwischen zwei Überprüfungen richtet sich im Wesentlichen nach den für Überprüfungen noch verfügbaren Leistungsreserven des DRMs. Abschnitt 4.2.9 gibt Empfehlungen für die Wahl von T_{check}.

4.2.8 Nachrichtenvertauschung

Ein empfangenes ConfReply wiederholt die Werte (S,G,Oif), NRS-Zustand und DRbit aus dem Conf, das es beantwortet. Das ConfReply lässt sich hierdurch eindeutig einem an denselben Router gesendeten Conf zuordnen. In kurzen zeitlichen Abständen gesendete Conf enthalten normalerweise dieselben Werte, da sie der im vorigen Abschnitt beschriebenen Behandlung von Nachrichtenverlust dienen. Zwei solche Conf sind tatsächlich vollkommen identisch. Sie tragen insbesondere keine Sequenznummer. Reihenfolgevertauschungen durch die Übertragung zwischen DRM und Router sind daher in diesem Fall völlig unerheblich. Dasselbe gilt für vertauschte ConfReply.

Bei Conf (und ConfReply) mit unterschiedlichen Werte sieht dies anders aus. Typischerweise werden sie zwar in so großem zeitlichem Abstand zueinander gesendet, dass keine Gefahr der Reihenfolgevertauschung besteht. Trotzdem kann es vorkommen, das zwei unterschiedliche Conf kurz hintereinander gesendet werden, z. B. wenn gerade die periodische Überprüfung nach T_{check} erfolgt ist und kurz darauf wegen eines Gruppenbei- oder -austritts der etablierte NRS-Zustand geändert werden soll. Dann muss sichergestellt sein, dass die im zuletzt gesendeten Conf enthaltenen Werte im Router konfiguriert wurden und nicht jene vom zuvor gesendeten Conf. Anhand der Reihenfolge der empfangenen ConfReply kann dies nicht festgestellt werden, da auch sie, oder nur sie, bei der Übertragung vertauscht werden können. Das Problem vertauschter Nachrichten lässt sich jedoch behandeln, indem das letzte Conf zweimal im Abstand von T_{RTO} gesendet wird. Die Wiederholung ist nur dann nötig, wenn der Sendezeitpunkt des letzten zuvor gesendeten Conf mit anderen Werten kürzer als T_{RTO} zurückliegt.

Um dies zu veranschaulichen, sei Abbildung 4.8 betrachtet. In allen Fällen (a) bis (e) folgt ein Conf(Ack,Keep) einem kürzlich gesendeten Conf(Ack,Remark). Eine einfache

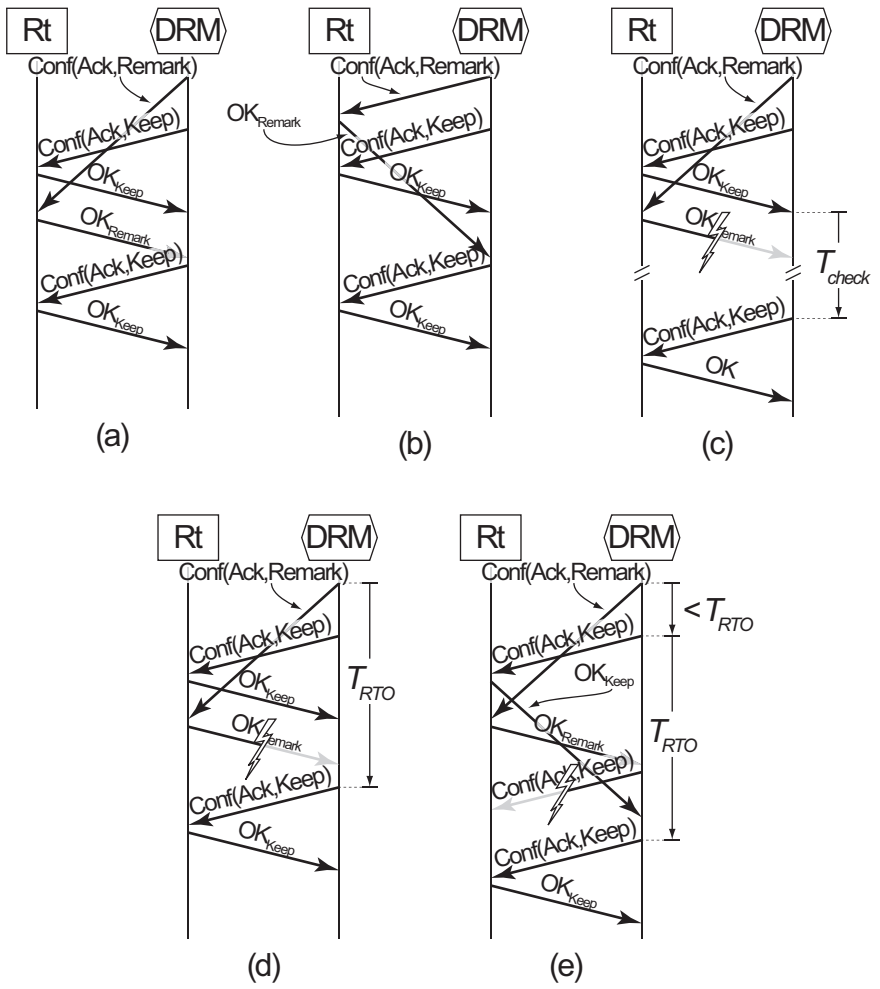


Abbildung 4.8 Verschiedene Fälle von vertauschten DSMC-Nachrichten und Möglichkeiten zu ihrer Behandlung. (e) zeigt das in DSMC verwendete Vorgehen.

Reihenfolgevertauschung (a) oder der Verlust des zweiten Conf(Ack,Keep) (nicht dargestellt) wird durch den Empfang des unerwarteten $\mathrm{OK}_{\mathrm{Reply}}$ erkannt. Zuletzt vom DRM gesendet wurde ein Conf(Ack,Keep), erwartet hätte er demnach ein $\mathrm{OK}_{\mathrm{Keep}}$. Er sendet somit das letzte Conf(Ack,Keep) erneut. Wurden stattdessen (b) nur die OK vertauscht, ist das erneute Senden des Conf(Ack,Keep) zwar unnötig, aber vom DRM nicht von Fall (a) unterscheidbar. Geht dagegen (c) das veraltete $\mathrm{OK}_{\mathrm{Remark}}$ verloren, kommt es ohne weitere Maßnahmen nicht zum Neusenden eines Conf und die falsche Konfiguration bleibt bis zum nächsten periodischen Conf nach T_{check} erhalten. Um dies zu vermeiden, muss (d) ab dem Zeitpunkt, ab dem mit hinreichend hoher Wahrscheinlichkeit ausgeschlossen werden kann, dass kein veraltetes ConfReply mehr eintreffen wird, also T_{RTO} nach Senden des veralteten Conf, nochmals das aktuelle Conf wiederholt werden. Wäre das $\mathrm{OK}_{\mathrm{Remark}}$ nicht verloren gegangen, hätte es als unerwartetes ConfReply wie in (a) oder (b) zwischenzeitlich zum erneuten Senden eines Conf(Ack,Keep) geführt. Im Fall (d) wären dann das Conf(Ack,Keep) zweimal wiederholt worden, obwohl einmal ausgereicht hätte. Im Interesse einer möglichst schnellen Konfiguration des Routers mit den richtigen Werten ist es jedoch sinnvoll, auf während der Wartezeit T_{RTO} unerwartet eintreffende ConfReply trotzdem mit einem erneuten aktuellen Conf zu reagieren. Schließlich stellt das Wiederholen eines Conf für den DRM einen nur geringen Aufwand dar (s. Abschnitt 4.4.2).

Leider weist auch Fall (d) noch ein Problem auf: Ein Verlust des letzten Conf(Ack,Keep) wird u. U. nicht erkannt. Denn das nach dem Ablauf der Wartezeit T_{RTO} gesendete Conf(Ack,Keep) kann durch ein spät eintreffendes $\mathrm{OK}_{\mathrm{Keep}}$ scheinbar bestätigt werden, welches jedoch tatsächlich dem ursprünglichen Conf(Ack,Keep) galt. Dies ist möglich, da seit dem Senden des ersten Conf(Ack,Keep) erst weniger als T_{RTO} vergangen ist. Dieser Fall führt genau dann zum Problem, wenn (e) das erste Conf(Ack,Keep) das veraltete Conf(Ack,Remark) überholt und somit veralteter NRS-Zustand im Router konfiguriert ist. Geht jetzt das neue Conf(Ack,Keep) verloren, bleibt diese Situation erhalten. Der DRM nimmt dagegen an, dass der Router das neue Conf(Ack,Keep) korrekt empfangen hat, da er nach dessen Senden ein $\mathrm{OK}_{\mathrm{Keep}}$ empfängt (das jedoch vom Router für das erste Conf(Ack,Keep) generiert wurde). Diese falsche Zuordnung lässt sich vermeiden, indem grundsätzlich ein Conf nach T_{RTO} wiederholt wird. Ein verspätetes OK ist zu diesem Zeitpunkt und bei korrekter Wahl von T_{RTO} mit großer Wahrscheinlichkeit ausgeschlossen.

Zusammengefasst muss bei DSMC immer dann, wenn vom DRM zwei Conf mit unterschiedlichen Werten von NRSstate und/oder DRbit innerhalb von T_{RTO} gesendet werden, das zweite Conf nach T_{RTO} einmal wiederholt werden und danach im Abstand T_{RTO} – oder mit Exponential Backoff (s. folgender Abschnitt) mit bis T_{RTO} anwachsendem Abstand – solange weiter, bis ein ConfReply mit den aktuellen Werten das aktuelle Conf bestätigt. Darüber hinaus wird bei jedem Eintreffen eines ConfReply mit veralteten Werten das aktuelle Conf sofort wiederholt, um eine möglichst schnelle Konfiguration mit den neuen Werten zu erreichen.

Bei Eintreffen eines ConfReply mit veralteten/falschen Werten sollte grundsätzlich das letzte aktuelle Conf wiederholt werden, auch wenn keine zwei unterschiedlichen Conf kurz innerhalb von T_{RTO} gesendet wurden. Dies sorgt für eine weitere Absicherung, falls T_{RTO} zu klein gewählt wurde oder durch Fehlfunktionen von Routern veraltete Conf-Nachrichten ungewöhnlich spät im Netz erneut auftauchen. Notfalls sorgt das periodische Wiederholen nach T_{check} für einen definierten Zustand in den Routern. Da T_{check} in großen Netzen mit vielen Multicastgruppen u. U. sehr groß gewählt werden muss, um die Belastung des DRM und ggf. den entstehenden Signalisierungsverkehr in Grenzen zu halten, ist es essentiell, dass T_{RTO} nicht zu klein gewählt wird und Gleichung 4.1 erfüllt.

Mit Sequenznummern lassen sich zwar vertauschte und verspätete Nachrichten sehr zuverlässig behandeln. Bei DSMC stehen Nachrichten jedoch vollständig für sich, bilden somit nicht zusammen mit anderen eine Informationseinheit. Nur aus diesem Grund lassen sich wie beschrieben Reihenfolgevertauschungen durch einfaches Wiederholen der jeweils letzten Nachricht beheben. Die DSMC-Komponente der Router kommt so ohne jeden Zustand aus.

4.2.9 DRM-seitiges Timing

Das Retransmission Timeout T_{RTO}, mit dem der DRM seine Conf-Nachrichten wiederholt, wird in Abhängigkeit der Ausdehnung der Domäne und der Bearbeitungszeit durch die Kommunikationspartner gewählt. Die maximale Dauer eines Signalisierungsvorgangs zwischen DRM und einem beliebigen Router seiner Domäne sei durch **MaxDrmRtt** nach oben hin abgeschätzt. MaxDrmRtt ist minimal größer als die Paketlaufzeit einer DSMC-Nachricht vom DRM zu dem von ihm am weitesten entfernt liegenden Router in seiner Domäne und einer zweiten DSMC-Nachricht von diesem Router zurück zum

DRM zuzüglich der routerinternen Dauer der Nachrichtenverarbeitung. Dann sollte T_{RTO} Gleichung

$$T_{RTO} = \text{MaxDrmRtt} \tag{4.1}$$

erfüllen. Ist kein einheitlicher Wert für alle Router sinnvoll, weil Paketlaufzeit oder Verarbeitungszeit stark schwankt, Ist ein Exponential Backoff sinnvoll. Insbesondere wenn mehrere Conf und ConfReply zu einer Nachricht aggregiert werden, was bei der vom DRM vorgenommenen periodischen Überprüfung des in den Routern etablierten Zustands (s.ˑ) sinnvoll ist, wird die Verarbeitungszeit von der Anzahl an aggregierten Conf und ConfReply abhängen. Das initiale Timeout wird beim Exponential Backoff kleiner als $T_{RTO} = \text{MaxDrmRtt}$ gewählt und vor jeder Sendewiederholung verdoppelt. T_{RTO} bildet in diesem Fall die Obergrenze für das durch den Exponential Backoff wachsende Timeout.

Wie bei unterschiedlichen zwei kurz aufeinanderfolgenden Conf-Nachrichten kann es zur Reihenfolgevertauschung einer kurz vor dem TriggerStop gesendeten anderen DSMC-Nachricht mit dem TriggerStop kommen. Während klar ist, dass nach einem TriggerStop ohne vorheriges ein Conf kein ConfReply folgen kann, ist bei einem auf den TriggerStop folgenden Trigger vom DRM nicht feststellbar, ob dieser tatsächlich nach dem TriggerStop oder davor generiert und nur später als der TriggerStop eingetroffen ist. Der DRM muss daher für eine gewisse Zeitspanne nach einem TriggerStop alle empfangenen Trigger ignorieren. Kam der Trigger tatsächlich nach dem TriggerStop, wird er ohnehin vom Router wiederholt werden. Die Zeitspanne darf nicht zu groß gewählt werden, um zügig neue Trigger zu erkennen. Innerhalb von

$$T_{HoldDown} = \frac{1}{2} \cdot \text{MaxDrmRtt} \tag{4.2}$$

hat jeder vor dem TriggerStop gesendete Trigger den DRM erreicht. Ab diesem Zeitpunkt nach einem TriggerStop dürfen somit eintreffende Trigger wieder bearbeitet werden.

Durch das periodische Überprüfen im Abstand T_{check} gleicht der DRM seinen internen Zustand mit dem tatsächlichen vom Multicastrouting in seiner Domäne etablierten Weiterleitungszustand ab. Ein kurzes T_{check} ist zwar für eine zügige Erkennung von Inkonsistenzen wünschenswert, die periodische Überprüfung erzeugt jedoch zusätzliche Last für den DRM. Diese sollte daher grundsätzlich begrenzt werden. Lässt man T_{check} proportional zur Zahl zu überprüfender NRS-Zustände wachsen, bleibt die Gesamtanzahl gesendeter Conf und damit die Last durch daraufhin empfangene Trigger konstant. Des Weiteren muss der augenblicklichen Systemlast Rechnung getragen werden. Bei hoher Last sollte die zusätzliche Last der periodischen Überprüfung durch ein großes T_{check} klein gehalten werden, um die Bearbeitungsgeschwindigkeit der übrigen DSMC- und Dienstgütesignalisierung nicht unnötig negativ zu beeinflussen. Schließlich ist bei viel unbelegten Ressourcen im Netz eine schnelle Erkennung von Inkonsistenzen weniger kritisch, da die als belegt angenommenen, tatsächlich aber bereits wieder freien, Ressourcen nicht zu einer Blockierung neuer Reservierungen führen:

$$T_{check} \text{ sollte monoton wachsen mit } \left\{ \begin{array}{l} \text{der Anzahl an NRS-Zuständen} \\ \text{den noch unbelegten Ressourcen} \\ \text{der Systemlast des DRMs} \end{array} \right\}. \tag{4.3}$$

Da TriggerStop vom Router nicht wiederholt werden, ein Nachrichtenverlust also nur durch ein periodisches Conf(Ack) spätestens nach T_{check} erkannt wird, empfiehlt es sich, T_{check} für

NRS-Zustände mit gesetztem DRbit deutlich kleiner zu wählen als für NRS-Zustände ohne gesetztes DRbit, zumal die Anzahl letzterer deutlich größer ist. T_{check} sollte sich für erstere primär an ihrer Anzahl orientieren und in erster Näherung proportional mit dieser wachsen. NRS-Zustände mit gesetztem DRbit sollten dagegen die noch unbelegten Ressourcen mit berücksichtigen. Die Systemlast begrenzt T_{check} für beide nach unten, d. h. T_{check} wird bei Erreichen einer gewissen maximalen Last nicht mehr weiter verkleinert. Wie T_{check} genau gewählt wird, ist für die korrekte Funktion von DSMC unerheblich und bleibt der Implementierung des Ressourcenmanagers bzw. seiner Nachrichtenverarbeitungsroutine (s. Abschnitt 4.4.2) überlassen. Es werden lediglich im Falle verlorener TriggerStop-Nachrichten Ressourcen vom DRM umso länger als noch belegt angenommen, je größer T_{check} ist.

4.2.10 Routerseitiges Timing

Die Zeitmessung für die Verzögerung der TriggerUC beginnt mit dem beim Wechsel vom NRS-Zustand Init nach Invalid mit dem ersten Datenpaket. Das Inaktivieren eines Ausgangs stoppt eine laufende Zeitmessung. Sie beginnt nach dem erneuten Aktivieren von vorn, d. h. nach einem Inaktivieren/Aktivieren-Zyklus einer Ausgangsschnittstelle sind TriggerUC erneut zu verzögern. Dasselbe gilt für die Ratenlimitierung. Ein Inaktivieren/Aktivieren-Zyklus einer Ausgangsschnittstelle setzt konzeptionell allen DSMC betreffenden Zustand zurück.

Das Timing für Trigger und TriggerUC sollte idealerweise dafür sorgen, dass unter regulären (konsistenten) Bedingungen genau ein Trigger und kein TriggerUC gesendet wird und vom DRM zu bearbeiten ist. Idealerweise ist die Verzögerung der TriggerUC und die Ratenlimitierung der Trigger und TriggerUC daher groß genug, dass dem DRM nach Empfang eines Triggers genügend Zeit zur Konfiguration aller Router entlang des neuen Pfads bleibt.

MaxDrmRtt sei wie oben definiert. **MaxDrmProcessingTime** sei die Zeit, die der DRM maximal zur Ressourcenüberprüfung benötigt (Pfad aus Trigger berechnen, Ressourcen längs des Pfads auf Verfügbarkeit prüfen und ggf. reservieren sowie entsprechende Conf-Nachrichten der Nachrichtenverarbeitungsroutine übergeben). Und **MaxSignalingDelay** sei das Maximum der Dauer der Multicastsignalisierung zwischen jedem möglichen Paar von Routern in der Domäne (z. B. zwischen zwei Grenzroutern an gegenüberliegenden Seiten der Domäne). MaxSignalingDelay umfasst neben der reinen Paketlaufzeit auch die Bearbeitungszeit der Signalisierungsnachricht oder des Datenpakets in den Routern entlang des Pfads zwischen diesen beiden Routern. Ein Datenpaket ist hier das erste Datenpaket einer Quelle, dass durch die Routingebene bearbeitet und weitergeleitet wird und die Etablierung von Weiterleitungszustand auslöst. Dann lässt sich obige Forderung, dass nur genau ein Trigger und kein TriggerUC generiert wird, mit folgenden Werten für Rate R und Unterdrückungszeit T_S (für Suppression) erfüllen:

$$1/R = \text{MaxDrmProcessingTime} + \text{MaxDrmRtt} \tag{4.4}$$

$$T_S = \text{MaxDrmProcessingTime} + \text{MaxDrmRtt} + \text{MaxSignalingDelay} \tag{4.5}$$

Die Zeit für T_S kommt resultiert daher, dass der erste Router hinter der Quelle immer ummarkieren muss und daher einen TriggerUC senden wird. Bis das erste Datenpaket den Designated Router des Empfängers erreicht, der daraufhin den Trigger sendet, vergeht MaxSignalingDelay.

Allerdings muss das Timing auch eine zügige Erkennung und Behandlung verlorener Trigger ermöglichen. Denn je kleiner die Rate R gewählt wird, desto länger dauert es bei einem Verlust von Trigger- oder Conf-Nachrichten, bis der Trigger an den DRM wiederholt wird und dieser sein Conf (erneut) sendet (s. Abbildung 4.7). Wenn die Bearbeitungszeit im DRM stark schwankt und das Maximum deutlich nach oben hin von der üblichen Bearbeitungszeit abweicht, oder wenn die Bearbeitungszeit im DRM ohnehin sehr groß (einige Sekunden) ist, sollte $1/R$ deshalb zu Gunsten der Behandlung von Paketverlust kleiner gewählt werden. Wie klein, ist relativ unkritisch. Während ein unnötig gesendeter TriggerUC vollständig vom DRM bearbeitet werden muss, können Wiederholungen von Triggern vom DRM leicht als solche erkannt werden und verursachen nur eine geringe zusätzliche Last (s. Nachrichtenverarbeitungsroutine Abschnitt 4.4.2). Ein zu kleines T_S führt dagegen zu einem unnötigen TriggerUC, der vollständig inklusive Ressourcenüberprüfung bearbeitet werden muss. Die erhöhte Belastung des DRMs verschlechtert damit die Skalierbarkeit von DSMC. Da die erste Nachricht entscheidend für die Belastung des DRMs ist, ist ein Exponential Backoff anders als bei der Wiederholung der Conf hier nur von geringem Vorteil, zumal er weiteren Zustand im MFC oder in der DSMC-Komponente erfordern würde.

Die zweite Gleichung 4.5 setzt voraus, dass es zu keinem Verlust von Trigger-Nachrichten, Datenpaketen und Signalisierungsnachrichten des Multicastroutings kommt. Soll auch beim Verlust von n aufeinanderfolgenden Triggern des Designated Routers und m aufeinanderfolgenden Datenpaketen kein unnötiger TriggerUC generiert werden, müsste T_S weiter vergrößert werden. Schlimmstenfalls müsste $T_S = (n+1) \cdot ((m+1)/r) + \text{MaxDrmProcessingTime} + \text{MaxDrmRtt} + \text{MaxSignalingDelay}$ gewählt werden, wenn r die minimale Paketrate des Multicastdatenstroms ist, für den neuer NRS-Zustand konfiguriert werden soll. Trigger werden nicht exakt mit der durch Gleichung 4.4 gegebenen Rate gesendet, sondern ein Trigger wird per Definition immer nur durch die Weiterleitung eines Datenpakets ausgelöst. Er kann damit bei einem Verlust um bis zu dem maximalen zeitlichen Abstand zweier aufeinanderfolgender Datenpakete später erfolgen, bei Verlust von m Datenpaketen entsprechend $(m+1)/r$ später. Normalerweise ist r nicht bekannt, müsste also geschätzt werden, was aufgrund der großen Heterogenität der in einer Domäne auftretenden Datenraten schwierig ist. Der Verlust von Signalisierungsnachrichten lässt sich kaum sinnvoll erfassen, da z. B. PIM-DM standardmäßig nur alle 210 s ein erneutes Prune erlaubt, der Verlust einer solchen Nachricht also MaxSignalingDelay um 210 s vergrößern würde.

Anstatt Verluste bei der Unterdrückungszeit T_S der TriggerUC mit zu berücksichtigen, ist es einfacher und sinnvoller, DSMC-Nachrichten und Multicastroutingnachrichten einem PHB mit niedriger Verlustwahrscheinlichkeit zuzuordnen und so die Wahrscheinlichkeit eines Paketverlust gering zu halten. Für das Unicastrouting ist dies gängige Praxis. Zusätzlich sorgt dies für einen zügigeren Aufbau des Multicastpfads und schnellerer Bereitstellung von Dienstgüte, da Paketverluste die Signalisierung seltener verzögern können. Ebenfalls sinnvoll ist die Nutzung eines hochprioren DSCPs für DSCP_SIG nach Regel **Init**. Wegen der geringen Größe von $1/r$ verglichen mit der Periodendauer einzelner Signalisierungsnachrichten im Multicastrouting ist dies allerdings sehr viel weniger kritisch.

Broadcast-and-Prune

Bei Multicastrouting nach dem Broadcast-and-Prune-Verfahren (z. B. PIM-DM) werden Datenpakete initial auch ohne Empfänger weitergeleitet, bis durch Prune-Nachrichten Routingzustand etabliert ist, der die Weiterleitung stoppt. Ein Trigger bleibt ohne Empfänger

jedoch aus. Um in dieser Zeit nicht unnötig TriggerUCs zu generieren, sollte die Zeitdauer ihrer Unterdrückung größer sein als die Summe aus der Zeit, in der das erste Datenpaket einer beliebigen Quelle jeden Link der Domäne erreicht hat, und der maximalen Zeit, die das anschließende Prunen benötigt, um zurück zum ersten Router hinter der Quelle zu gelangen, also mindestens zweimal MaxSignalingDelay[4]. Bei Broadcast-and-Prune-Routing gilt anstatt Gleichung 4.5 daher für T_S:

$$\begin{aligned} T_S &= \max(2 \cdot \text{MaxSignalingDelay}, \\ &\qquad \text{MaxDrmProcessingTime} + \text{MaxDrmRtt} + \text{MaxSignalingDelay}) \end{aligned} \tag{4.6}$$

Router können nicht nur zu zweit Punkt-zu-Punkt sondern es können mehrere Knoten über einen Link miteinander verbunden sein. Zur Unterscheidung werden Links mit drei und mehr Knoten im Folgenden als LAN bezeichnet. Dies mach aus Sicht von PIM-SM solange keinen Unterschied vergleichen mit einem Punkt-zu-Punkt-Link, solange weiterhin nur genau zwei PIM-DM mit diesem LAN verbunden sind. Ab dem dritten Router allerdings kommt es zum Überschreiben von Prune-Nachrichten. Für solche Links muss T_S noch größer gewählt werden, da je solchem LAN mit drei oder mehr Routern entlang des Pfads das J/P_Override_Interval hinzu kommt. Während dieser Zeit wartet der Router auf eine Join-Nachricht eines anderen Nachbarrouters auf dem LAN, die das Prune ggf. wieder überschreibt (s. Überschreiben von Prunes in Abschnitt 2.3.2). Erst nach Ablauf dieser Zeit wird die Schnittstelle als gepruned betrachtet und, sofern es keine weiteren Schnittstellen mehr mit entsprechendem Join-Zustand gibt, selbst ein Prune Upstream Richtung Quelle gesendet. Die Zeitdauer, die maximal vergeht, bis eine Schnittstelle des ersten Routers hinter der Quelle gepruned ist, nimmt entsprechend der Anzahl der LANs auf einem Pfad das Mehrfache von J/P_Override_Interval in Anspruch.

T_S richtet sich nach dem Multicastpfad mit der größten Summe aus allen J/P_Override_Interval(LAN_i) der N längs des Pfads auftretenden LANs LAN_i:

$$\begin{aligned} T_S &= \max(2 \cdot \text{MaxSignalingDelay} + \sum_{i=1}^{N} \text{J/P\_Override\_Interval}(\text{LAN}_i), \\ &\qquad \text{MaxDrmProcessingTime} + \text{MaxDrmRtt} + \text{MaxSignalingDelay}) \end{aligned} \tag{4.7}$$

Typischerweise ist dieser kritische Multicastpfad mit der längsten Signalisierungszeit einer zwischen zwei Grenzroutern an gegenüberliegen Seiten der Domäne, der auf seinem Pfad die meisten solcher LANs beinhaltet. Welcher Pfad genau dies ist, kann der DRM aus seiner Kenntnis der Topologie und damit der Lage der LANs in seiner Domäne sowie den über das Netzwerkmanagement bekannten jeweiligen Werten für J/P_Override_Interval ableiten. Der ermittelte Wert für T_S wird vom DRM dann über das Netzwerkmanagement auf allen Routern konfiguriert. Letztlich genügt auch eine Näherung, die die Summe nach oben hin abschätzt. Eine zu kleine Schätzung führt dagegen bei Quellen, bei denen der durch das RPF-Prinzip entstehende Spannbaum den kritischen Pfad beinhalten, zu einem unnötigen TriggerUC.

Der Standardwert für J/P_Override_Interval ist mit 3 Sekunden sehr groß. Existieren viele solche LANs, führt dies ggf. zu einem sehr großen Wert für T_S, was die Erkennung von

[4]Bei Broadcast-and-Prune endet die Signalisierung bei Links mit mehr als einem Router mit dem Fluten des Links und nicht bereits beim letzten Router vor dem letzten Link, vgl. der Assert im Beispiel zu PIM-DM Abschnitt 4.2.11.2. MaxSignalingDelay ist durch den einen zusätzlichen Hop hier etwas größer.

Inkonsistenzen entsprechend verzögert. J/P_Override_Interval kann aber über die LAN Prune Delay Hello Option mit einer Auflösung von einer Millisekunde verkleinert werden. In Abhängigkeit der Bearbeitungsgeschwindigkeit der Router und der Ausdehnung des jeweiligen LANs sind Werte bis hinunter zu einigen zehn Millisekunden möglich.

Aggregation

Aggregation, z. B. durch Konkatenation mehrerer Trigger und Übertragen in einem IP-Paket, reduziert die Anzahl an DSMC-*Nachrichten*, nicht jedoch die Anzahl an zu bearbeitenden Triggern. Der DRM muss weiterhin für jede Ausgangsschnittstelle separat die Ressourcenverfügbarkeit prüfen. Die Aggregation reduziert somit lediglich den Signalisierungs*verkehr*.

Während eine Aggregation der periodisch im Abstand T_{check} wiederholten Conf- und ConfReply-Nachrichten gut möglich und sinnvoll ist, treten Trigger typisch nur vereinzelt auf. Eine Aggregation ist daher nur dann möglich, wenn mehrere innerhalb der durch Verzögerung und Ratenlimitierung entstehenden Unterdrückungszeit zu sendenden Trigger geeignet verzögert und andere vorgezogen werden. Hierbei sollte sichergestellt sein, dass der erste Trigger weiterhin sofort durch das erste Datenpaket ausgelöst wird und der erste TriggerUC nicht kürzer unterdrückt wird als ohne eine solche Aggregation. Andernfalls wird die Bereitstellung der Dienstgüte unnötig verzögert bzw. es könnte ein zu verfrühter und damit unnötiger TriggerUC erzeugt werden. Deutlich verzögert werden darf ein TriggerUC jedoch nicht, um die Erkennung von Inkonsistenzen nicht unnötig zu verlangsamen. Der Weiteren sollte die Wiederholung eines Triggers eher früher als später erfolgen, da ein verfrüht wiederholter Trigger zwar einen gewissen (geringen) Mehraufwand seitens des DRM bedeutet, umgekehrt das Verzögern des Triggers jedoch im Falle des Verlusts des ersten Triggers zu verzögerter Bereitstellung von Dienstgüte führt. Zusammengefasst kann die Unterdrückungszeit einzelner TriggerUC und die Wiederholrate einzelner Trigger zum Zwecke der Aggregierbarkeit in Maßen vergrößert werden, jedoch sollte weder das eine noch das andere verkleinert werden.

Beispielimplementierung

Eine einfache routerseitige Implementierung, wie sie im folgenden Kapitel vorgestellt wird, differenziert im Timing weder nach Multicastroutingprotokoll noch lässt es Reserven für den Verlust von Nachrichten und aggregiert nicht. Für die Trigger wird die Rate gemäß Gleichung 4.4 gewählt. Schätzt man MaxSignalingDelay mit der Summe aus MaxDrmProcessingTime und MaxDrmRtt nach oben hin ab, sind Gleichung 4.5 und Gleichung 4.6 äquivalent. Da nach Gleichung 4.4 MaxSignalingDelay bzw. diese Summe $1/R$ entspricht, kann die anfängliche Unterdrückung der TriggerUC durch Verwerfen der ersten zwei TriggerUC-Nachrichten erreicht werden. Hierdurch lassen sich bereits bei fast allen Multicastroutingprotokollen unnötige TriggerUC vermeiden. Bei PIM-DM mit LANs mit drei oder mehr Routern und beim ebenfalls vom Basisverfahren unterstützten PIM-SM mit sofortigem SPT Switchover (s. Abschnitt 4.3.3.2) muss die Zahl n verworfener TriggerUC weiter vergrößert werden, bis auch für diese Protokolle $n/R \geq T_S$ gilt mit T_S nach Gleichung 4.7 für PIM-DM bzw. nach Gleichung 4.8 für PIM-SM. Da nicht nach Multicastroutingprotokoll differenziert wird, muss sich bei gleichzeitigem Einsatz mehrerer Multicastroutingprotokolle n nach dem Maximum aller T_S aller Multicastroutingprotokolle orientieren.

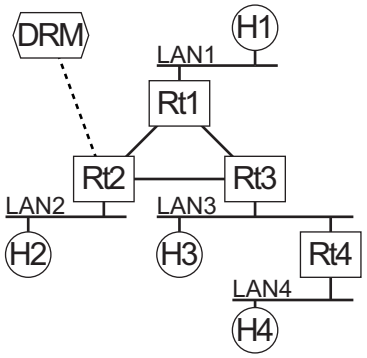


Abbildung 4.9 Topologie zur Erläuterung der DSMC-Signalisierung.

4.2.11 Beispiele

Im Folgenden soll die Anwendung von DSMC auf die Multicastroutingprotokolle der PIM-Familie erläutert werden. Gegeben sei die in Abbildung 4.9 dargestellte Topologie aus vier Routern Rt1 bis Rt4 und vier Hosts H1 bis H4 gegeben. Die Hosts sind über die LANs angeschlossen. Ob die Links zwischen Rt1, Rt2 und Rt3 z. B. bei OSPF als Punkt-zu-Punkt-Link oder normalen Transitlink konfiguriert sind, macht für die Multicastroutingprotokolle der PIM-Familie keinen Unterschied. Entscheidend für das Verhalten der Router im Multicastrouting ist lediglich die Anzahl an Nachbarroutern auf einem Link. Auf allen Links der gezeigten Topologie ist die Zahl der Router eins oder zwei, die Zahl der Nachbarn somit maximal eins.

Dies trifft auch auf LAN3 zu, da Host H3 nicht am Multicastrouting teilnimmt sondern nur am Gruppenmanagementprotokoll. Es wurde auf Links mit drei oder mehr Routern verzichtet, da dies den grundlegenden Ablauf der Signalisierung von DSMC nicht ändert, jedoch in gewissen Fällen zu weiterer Signalisierung im Rahmen des Multicastroutings führt (Asserts und Überschreiben von Prunes). Das Multicastrouting soll hier nicht im Zentrum stehen, sondern wird nur mit dargestellt, um die Nachrichten der DSMC-Signalisierung zeitlich einzuordnen. Der Ressourcenmanager DRM ist beliebig mit dieser Topologie verbunden, beispielsweise mit Rt2.[5] Signalisierungsnachrichten vom und zum DRM nehmen einen Weg, der durch das Unicastrouting bestimmt wird und hier unerheblich ist. Als (Unicast-)Routingmetrik wird der Hopcount angenommen. Jede Schnittstelle der Router hat demnach die Metrik eins. Hier wie in allen folgenden Beispielen wird davon ausgegangen, dass dem Ressourcenmanager die gewünschte Dienstgüte für die betrachtete Multicastkommunikation bereits bekannt ist und die Quelle berechtigt ist, diese Dienstgüte zu nutzen. Sie markiert daher ihre Datenpakete entsprechend mit dem zugehörigen DSCP und der First Hop Router ist vom DRM mit einem Profil konfiguriert, das so markierte Datenpakete passieren lässt.

4.2.11.1 PIM-SSM

PIM-SM als wichtigstes Routingprotokoll der PIM-Familie soll zuerst betrachtet werden, wobei vorerst nur dessen als PIM-SSM bezeichnete Teilfunktionalität für Source Specific

[5] Typischerweise wird der DRM zwecks Ausfallsicherheit mehrfach verbunden sein, seine eigene Adresse durch Teilnahme am Unicastrouting bekanntgeben, durch entsprechende Metriken aber gleichzeitig dafür sorgen, dass er keinen Transitverkehr weiterleiten muss, z. B. bei OSPF mittels Stub Router Advertisement [151].

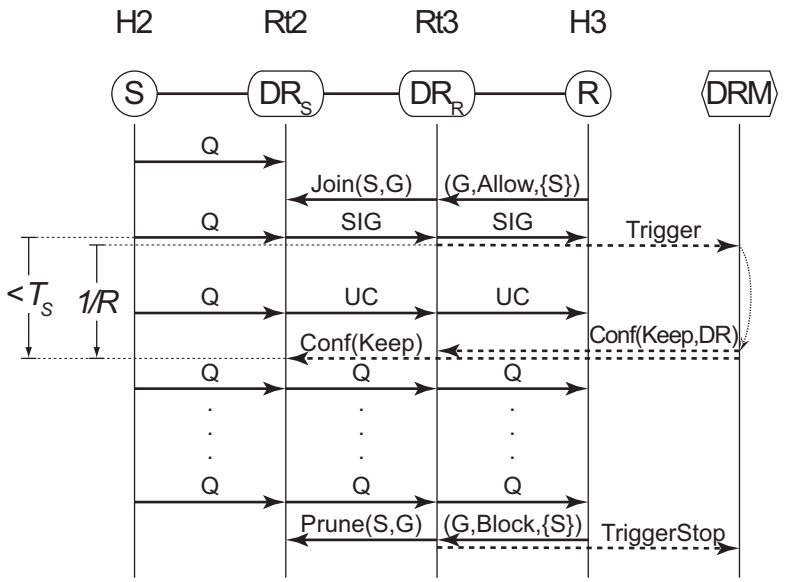


Abbildung 4.10 Weg-Zeit-Diagramm zur Erläuterung des Basisverfahrens von DSMC anhand von PIM-SSM (Q/SIG/UC = hochpriorer DSCP/DSCP_SIG/ DSCP_UC).

Multicast (s. Abschnitt 2.3.2.1 Seite 45) angenommen wird. PIM-SSM verwendet nur einen, den quellenspezifischen kürzesten Baum, und wird so bereits vom Basisverfahren von DSMC unterstützt.

Für die betrachtete Multicastkommunikation sei Host H2 die Quelle S und H3 der einzige Empfänger R. Da in dieser Konstellation nur die Router Rt2 als Designated Router der Quelle S und Rt3 als DR des Empfängers R an der Signalisierung teilnehmen werden, sind neben S und R nur diese zwei Router im Weg-Zeit-Diagramm Abbildung 4.10 als DR_S und DR_R eingetragen. Der relevante Ausschnitt der Topologie ist oben im Weg-Zeit-Diagramm nochmals durch Verbindungslinien zwischen den Knoten angedeutet.

Die Quelle S markiert ihre Datenpakete mit dem gewünschten DSCP. Der genaue DSCP ist für die DSMC-Signalisierung nicht relevant, solange er nicht zu den niederprioren DSCPs gehört, durch die das NRS-Problem nicht ausgelöst werden kann und die, gemäß Regel **Low**, von DSMC nicht berücksichtigt werden. In den Weg-Zeit-Diagrammen werden diese Datenpakete allgemein mit Q („Quality“) bezeichnet.

Die Pakete treffen beim ersten Router DR_S ein und werden noch nicht weitergeleitet, da es noch keinen Empfänger gibt. Sie werden von DR_S verworfen. Nun tritt R durch Senden des Änderungsberichts (G,Allow,{S}) über das Gruppenmanagementprotokoll, entweder IGMP oder MLD, der Multicastgruppe bzw. dem SSM-Kanal (S,G) bei. Der Designated Router des Empfängers DR_R sendet daraufhin einen quellenspezifischen Join(S,G) in Richtung der Quelle. Damit besitzt die Gruppe einen Empfänger und die Pakete werden von DR_S weitergeleitet, dabei jedoch vom ursprünglichen DSCP „Q“ gemäß Markierungsregel **Init** auf DSCP_SIG ummarkiert. DR_R hat keinen Empfänger, setzt das DRbit daher zurück und sendet keinen Trigger. Der TriggerUC wird gemäß Nachrichtenregel **TrUC** vorerst noch unterdrückt. Nachfolgende Pakete werden dann gemäß Regel **TrUC** auf DSCP_UC ummarkiert, um das NRS-Problem zu lösen. Schließlich ist dem DRM der neue

Empfänger und Multicastpfad noch nicht bekannt und der Pfad noch nicht hinsichtlich Ressourcenverfügbarkeit überprüft worden.

Kurz darauf erreicht das ummarkierte Paket den Router DR_R. Er besitzt ebenfalls ungültigen NRS-Zustand, informiert nach Regel **Tr** jedoch unverzüglich den DRM mit einem Trigger über den ungültigen NRS-Zustand. Der DRM kann aus den nach Regel **TSGO** in der Triggernachricht enthaltenen Informationen zusammen mit der Adresse des den Trigger sendenden Routers jenen Link der eigenen Domäne ermitteln, auf dem sich der neue Empfänger befindet. Aus der ebenfalls enthaltenen Adresse der Quelle bestimmt er den Link der Quelle und berechnet daraufhin den Multicastpfad zwischen diesen beiden Links. Wie er zu berechnen ist, hängt vom Multicastroutingprotokoll und der zugrundeliegenden MRIB ab. Ersteres leitet der DRM aus der ihm bekannten Zuordnung der Gruppenadressen zu den in der Domäne eingesetzten Multicastroutingprotokollen ab. Letzteres ist ihm durch Teilnahme am Unicastrouting bekannt. Der neue Multicastpfad muss nicht vom neuen Empfänger bis ganz zur Quelle verlaufen. Sollte bereits ein Empfänger für (S,G) vorhanden sein und der Multicastpfad zum neuen Empfänger auf diesen treffen, muss lediglich der neue Teil von jenem Router, bei dem die Pfade aufeinandertreffen, bis zum Link des Empfängers auf Ressourcenverfügbarkeit hin überprüft und Ressourcen belegt werden.

Im Beispiel wird G per PIM-SSM geroutet, der Multicastpfad also gemäß RPF-Prinzip Richtung Quelle berechnet. R ist der einzige Empfänger und der Pfad ist der gesamte Pfad S–DR_S–DR_R–R beginnend direkt bei der Quelle. Sind hinreichend Ressourcen vorhanden, sendet der DRM allen Routern entlang des neuen Pfads ein Conf(Keep), so dass anschließend alle Router gemäß Regel **Conf** ihren NRS-Zustand auf den im Conf enthaltenen NRS-Zustand Keep gesetzt haben. Im Conf(Keep,DR) an DR_R ist zusätzlich das DRbit gesetzt. Das auf DR_R bereits gesetzte DRbit soll nicht verändert werden und DR_R soll später einen TriggerStop senden. Wären keine Ressourcen mehr vorhanden, würde der DRM den NRS-Zustand trotzdem konfigurieren, um weitere Triggernachrichten zu unterbinden, wegen Regel **Rmrk** jedoch mit der Nachricht Conf(Remark) an DR_S und Conf(Remark,DR) an DR_R, um den NRS-Zustand auf Remark zu setzen und alle Datenpakete auf den niederprioren DSCP_LE umzumarkieren.

Wie links in Abbildung 4.10 eingetragen, war die anfängliche Unterdrückungszeit T_S der TriggerUC groß genug, damit der DRM in dieser Zeit die Konfiguration der Router vornehmen konnte und letztlich kein TriggerUC gesendet werden musste. Ebenso wurde die maximale Rate R, mit der die Trigger wiederholt werden, hinreichend klein gewählt, so dass der DRM innerhalb der Periodendauer $1/R$ der Trigger die Konfiguration vornehmen konnte und kein zweiter unnötiger Trigger generiert wurde. Nach nur einer Triggernachricht und einer Konfigurationsnachricht je Router auf dem neuen Multicastpfad fließen die Datenpakete der Quelle ohne ummarkiert zu werden bis zum neuen Empfänger. Die von der Quelle angeforderte Dienstgüte wird für den neuen Empfänger erbracht.

Verlässt R die Gruppe wieder durch Senden des Änderungsberichts (G,Block,{S}), informiert DR_R den DRM gemäß Regel **TrSt** mittels TriggerStop, dass er keine Empfänger mehr in der Gruppe besitzt und seinen Weiterleitungszustand abgebaut hat. Der DRM kann daraufhin den Pfadabschnitt berechnen, der durch den Austritt von R wegfallen wird und die bei ihm intern als belegt markierten Ressourcen wieder als verfügbar kennzeichnen. Da es im Beispiel keine anderen Empfänger gab, ist dies der komplette Pfad bis zur Quelle. Der NRS-Zustand muss vom DRM nicht gelöscht werden, da er beim Inaktivieren einer Ausgangsschnittstelle im MFC-Eintrag seine Gültigkeit verliert bzw. beim Löschen des

gesamten MFC-Eintrags (nach dem Inaktivieren der letzten Ausgangsschnittstelle im MFC-Eintrag) zusammen mit dem MFC-Eintrag gelöscht wird.

DR_R besitzt nun keine Schnittstelle mehr mit Join-Zustand für (S,G) oder mit lokalen Empfängern für (S,G) und sendet daher ein Prune(S,G) Richtung Quelle, um sich selbst vom Multicastbaum zu entfernen. Das Prune erreicht DR_S, der daraufhin keine Datenpakete mehr an diese Schnittstelle weiterleitet. Die normalerweise nach einem Prune eingefügte Wartezeit J/P_Override_Interval, die anderen Nachbarroutern an dieser Schnittstelle die Möglichkeit gibt, das Prune mit einem eigenen Join zu überschreiben, entfällt hier, da DR_R der einzige Nachbar von DR_S auf dieser Schnittstelle ist. Der MFC-Eintrag auf DR_S bleibt erhalten, solange die Quelle sendet. Der NRS-Zustand ist nach Regel **NRS** jedoch nun undefiniert und wird nicht beachtet, da die Ausgangsschnittstelle, zu dem er gehört, inaktiv geworden ist. DR_S sendet kein TriggerStop, da bei ihm das DRbit(S,G) beim Inaktivieren der Ausgangsschnittstelle nicht gesetzt war.

4.2.11.2 PIM-DM

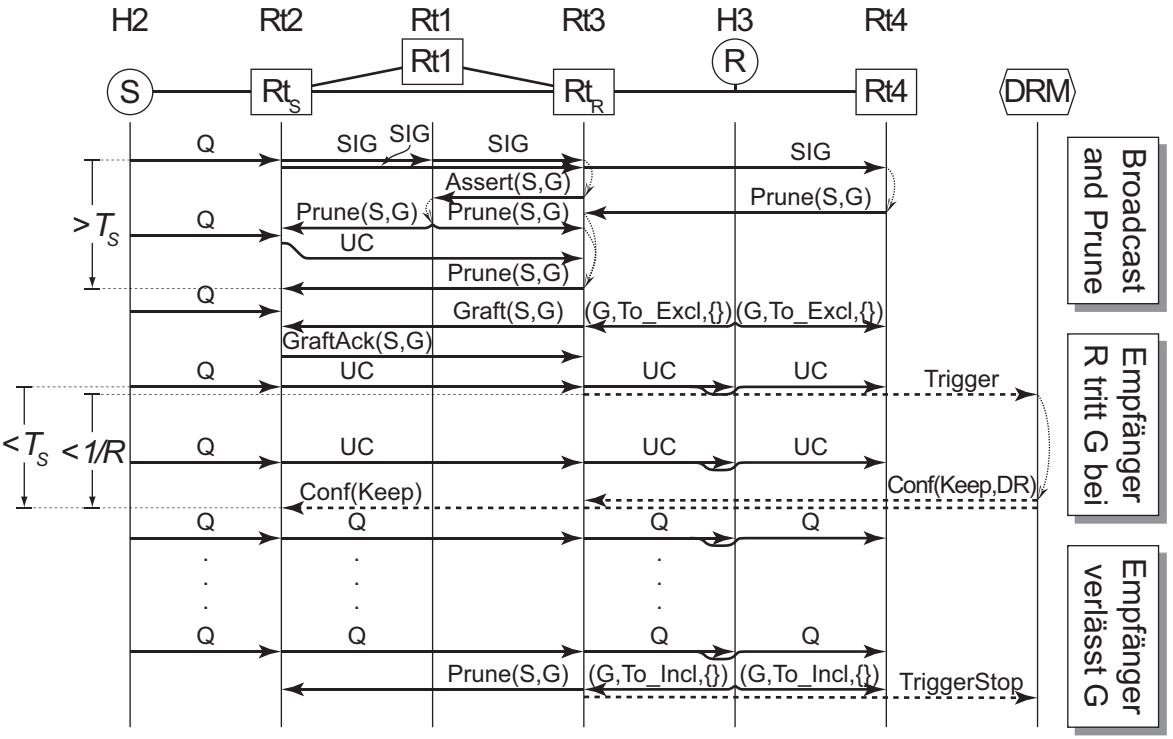


Abbildung 4.11 Ablauf der DSMC-Signalisierung (Basisverfahren) bei PIM Dense Mode.

Bei PIM-DM genügt ebenfalls das Basisverfahren von DSMC. Das Weg-Zeit-Diagramm Abbildung 4.11 zeigt den Ablauf der Signalisierung für PIM-DM. Wie zuvor bei PIM-SSM seien H2 und H3 wieder Quelle S bzw. Empfänger R. Die PIM-DM-Spezifikation kennt keine Designated Router. Die an der DSMC-Signalisierung beteiligten Router sind im Weg-Zeit-Diagramm daher einfach als Rt_S und Rt_R bezeichnet.

PIM-DM arbeitet nach dem Broadcast-and-Prune-Verfahren, bei dem Datenpakete initial im gesamten Netz geflutet werden. Obwohl Empfänger R der Gruppe G noch nicht beigetreten ist, leitet daher Rt_S das Datenpaket von S nach Ummarkierung auf DSCP_UC

weiter stromabwärts auf alle übrigen Schnittstellen mit Nachbarroutern, hier an Rt1 und Rt_R, denn das Paket ist gemäß RPF-Prinzip über die richtige Eingangsschnittstelle auf dem kürzesten Pfad Richtung S eingetroffen. Der TriggerUC wird vorerst unterdrückt. Auch bei Rt1 trifft das Paket über die richtige Eingangsschnittstelle ein und auch er hat mit Rt_R einen Nachbarrouter stromabwärts und leitet daher das Datenpaket ebenfalls weiter. Ebenso Rt_R, der umgekehrt Rt1 und zusätzlich noch Rt4 als Nachbarrouter hat und das Datenpaket an beide weiterleitet. Rt4 schließlich hat keine weiteren Nachbarrouter stromabwärts und dort auch keine lokalen Empfänger und sendet ein Prune(S,G) zu Rt_R, um keine weiteren Datenpakete mehr von der Quelle S zu erhalten. Zwischen Rt1 und Rt_R wird per Assert derjenige Router bestimmt, der Datenpakete zukünftig auf den Link Rt1–Rt_R weiterleitet. Rt_R gewinnt aufgrund seiner besseren Assertmetrik, so dass Rt1 als Verlierer ein Prune(S,G) zum Gewinner Rt_R sendet. Damit sind beide Downstreamschnittstellen von Rt_R gepruned und Rt_R sendet seinerseits ein Prune(S,G) zu seinem Upstreamnachbarn Rt_S. Nachdem das Prune Rt_S erreicht hat, markiert er in seinem MFC-Eintrag für (S,G) die Ausgangsschnittstelle als inaktiv und leitet keine Datenpakete von S mehr weiter. Für inaktive Schnittstellen wird kein NRS-Zustand benötigt und daher kein TriggerUC mehr generiert bzw. unterdrückt. Läuft der Broadcast-and-Prune-Zyklus innerhalb der Unterdrückungszeit T_S der TriggerUC ab, hat Rt_S zwischenzeitlich keine unnötigen TriggerUC erzeugt.

Nach diesem initialen Broadcast-and-Prune-Zyklus haben alle Router (S,G)-Prunezustand. Fordert nun R durch den Änderungsbericht (G,To_Excl,{}) den Empfang von Datenpaketen beliebiger Quellen der Gruppe G an, können die Router aufgrund ihres Wissens über die Existenz der Quelle S mittels Graft(S,G) erneut den Empfang der Datenpakete von S anfordern. So auch Router Rt_R, der den Graft(S,G) zu seinem Upstreamrouter Rt_R sendet und von diesem den Empfang mittels GraftAck(S,G) bestätigt bekommt. Anschließend fließen die Daten, werden von Rt_S aber ummarkiert. Der TriggerUC wird erneut unterdrückt, da zwischenzeitlich die Ausgangsschnittstelle im MFC-Eintrag nicht aktiv war und die anfängliche Verzögerung erneut beginnt. Auch Rt_R besitzt noch keinen NRS-Zustand und fordert als letzter Router vor dem Empfänger per Trigger den NRS-Zustand beim DRM an. Im Beispiel werden beide mit NRS-Zustand Keep konfiguriert, so dass Datenpakete, ohne ummarkiert zu werden, bis zum Empfänger R fließen und die angeforderte Dienstgüte erbracht wird. Abschließend tritt der Empfänger R aus der Multicastgruppe G aus. Das bereits im vorigen Abschnitt zu TriggerStop und Prune(S,G) am Beispiel PIM-SSM Gesagte gilt unverändert auch für PIM-DM.

4.2.11.3 BIDIR-PIM

BIDIR-PIM verwendet, im Gegensatz zu den anderen beiden Protokollen PIM-SSM und PIM-DM, keine quellenspezifischen Multicastbäume, sondern einen Baum für alle Quellen einer Multicastgruppe. Es unterstützt damit kein SSM, sondern nur Any Source Multicast (ASM). BIDIR-PIM verwendet das Konzept der Rendezvousstelle. Datenpakete von Quellen werden von den Designated Forwardern (DF) automatisch bis zum Rendezvousstellenlink weitergeleitet, ebenso die Join-Nachrichten, so dass Quellen und Empfänger zueinanderfinden. Bei BIDIR-PIM leiten nur die Designated Forwarder (DF) Datenpakete weiter. Sie bilden zusammen einen Spannbaum mit dem RPL als Wurzel, der unabhängig von einer späteren Multicastkommunikation etabliert wird. Der Aufbau soll hier nicht weiter betrachtet werden, sondern es wird angenommen, dass die Wahl der Designated Forwarder vor Beginn der Multicastkommunikation bereits abgeschlossen ist.

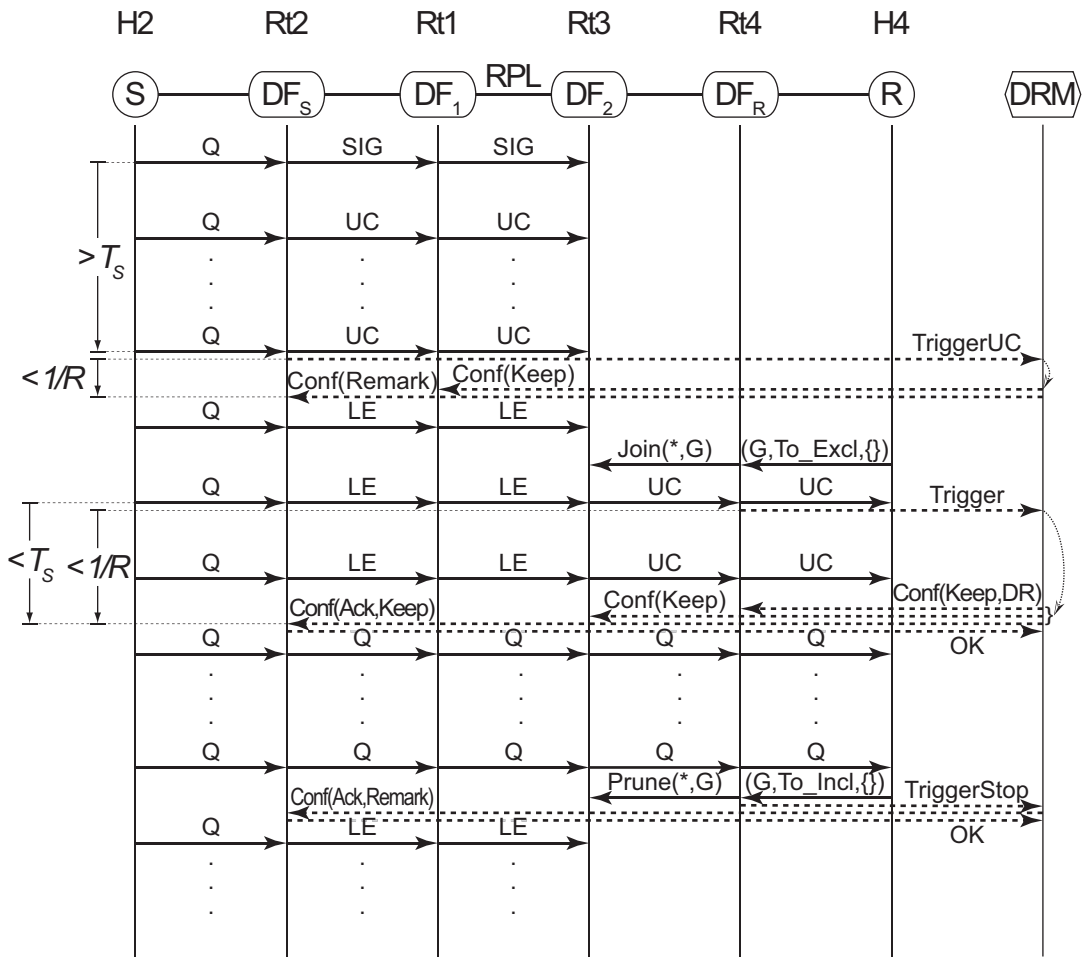


Abbildung 4.12 Ablauf der DSMC-Signalisierung (Basisverfahren) bei Bi-directional PIM.

BIDIR-PIM wird ebenfalls direkt durch das Basisverfahren unterstützt, da es, anders als PIM-SM, nur eine Art von Verteilbaum, den (bidirektionalen) Rendezvousstellenbaum verwendet. Das Weg-Zeit-Diagramm Abbildung 4.11 zeigt den Ablauf der DSMC-Signalisierung. Dargestellt ist der Fall, bei dem in Richtung Rendezvousstellenlink RPL konventionell per MFC weitergeleitet wird. Router mit Implementierungen, die per Longest-Prefix-Match weiterleiten (s. Abschnitt 2.3.1), legen bis zum Beitritt eines Empfängers keinen MFC-Eintrag an und werden daher nicht erfasst. Ihre Pakete fließen, vom Profil auf dem FHR zugelassen, auch ohne Empfänger grundsätzlich mit dem ursprünglichen von der Quelle verwendeten DSCP bis zum RPL. Dies stellt kein Problem dar, da der DRM über die Dienstgütesignalisierung Quelle, Gruppenadresse und damit den zugeordneten RPL und den Pfad kennt, und keine Paketduplizierung stattfindet, es also nicht zum NRS-Problem kommen kann. Der Fall gleicht Unicast, wo ebenfalls Pakete einer Quelle auch ohne Empfänger bis zum angegebenen Ziel weitergeleitet werden. Die DSMC-Signalisierung in Abbildung 4.11 vor dem Join(*,G) entfällt bei einer solchen Implementierung ersatzlos. Ebenfalls implementierungsspezifisch ist, ob beim Empfang des ersten Pakets einer Quelle S auf Basis von (*,G)-Joinzustand ein quellenspezifischer MFCentry(S,G) oder ob sofort mit dem Join(*,G) ein quellenunspezifischer MFCentry(*,G) angelegt wird. Bzgl. DSMC-Signalisierung macht dies keinen Unterschied. Das Dienstgütemanagement muss in

letzterem Fall allerdings ggf. die Ressourcenverfügbarkeit immer für mehrere Quellen der Gruppe G gemeinsam sicherstellen (s. Abschnitt 4.4.3).

Beginnt die Quelle S mit dem Senden, leitet DF_S die Datenpakete, nach Anlegen eines MFC-Eintrags für die Weiterleitung, Richtung RPL nach Ummarkierung erst auf DSCP_SIG, dann auf DSCP_UC an DF_1 weiter, der genauso verfährt und auf den RPL weiterleitet. DF_2 hat auf keiner seiner Schnittstellen Join-Zustand für die Gruppe G und leitet daher die Pakete vom RPL vorerst nicht weiter. Nach Ablauf der Unterdrückungszeit T_S informiert DF_S per TriggerUC den DRM. Es gibt verschiedene Möglichkeiten, wie der Ressourcenmanager auf den TriggerUC reagieren kann: mit Conf(Keep) den Router zur Beibehaltung des DSCPs anweisen oder mit Conf(Remark) eine dauerhafte Ummarkierung aller DSCPs nach DSCP_LE konfigurieren. Ein Conf(Keep) an alle Router auf dem Pfad sorgt für die Beibehaltung des DSCP der Quelle bis zum RPL und unterbindet ein weiteres Senden der TriggerUC. Nachteilig hierbei ist, dass die Pakete unnötig Dienstgüte erfahren, schließlich gibt es noch keine Empfänger, und dadurch anderen Kommunikationsbeziehungen unnötig Ressourcen genommen werden. Es kann daher sinnvoller sein, z. B. wenn angenommen wird, dass dieser empfängerlose Zustand länger erhalten bleibt, stattdessen mittels Conf(Remark) an den ersten Router hinter der Quelle DF_S und Conf(Keep) an die anderen Router, hier DF_1, ebenfalls den NRS-Zustand zu konfigurieren und damit die TriggerUC zu unterbinden. In Abbildung 4.11 ist diese Variante dargestellt und die Pakete werden von DF_S nach Erhalt des Conf(Remark) auf DSCP_LE ummarkiert. Prinzipiell besteht noch eine dritte Möglichkeit, die TriggerUC-Nachrichten zu ignorieren und nichts zu unternehmen. Die TriggerUC-Nachrichten werden weiterhin gesendet, jedoch gemäß Regel **TrUC** in ihrer Rate limitiert, so dass die durch sie erzeugte Last beim DRM begrenzt bleibt (wiederholte Trigger können vom DRM leicht als solche identifiziert werden). Diese Variante bringt bei BIDIR-PIM i. A. aber keinen Vorteil, kann jedoch bei PIM-SM unter gewissen Umständen sinnvoll sein (vgl. Beispiel zu PIM-SM Abschnitt 4.3.4). Treten bei den nicht rendezvousstellenbasierten Multicastroutingprotokollen bei korrekter Wahl der Unterdrückungszeit keine TriggerUC auf, muss hier auch die Rate der TriggerUC limitiert werden und Gleichung 4.4 erfüllen, um dem DRM ein hinreichend großes Zeitfenster zur Durchführung der Konfiguration zu lassen.

Der Zustand des Weiterleitens bis zum RPL nach Ummarkieren auf LE bleibt solange erhalten, bis später Empfänger R der Gruppe beitritt (Änderungsbericht (G,To_Excl,{})). Dies veranlasst DF_R dazu, ein Join(*,G) Richtung RPL zu senden, woraufhin DF_2 einen MFC-Eintrag anlegt und Datenpakete vom RPL nach Ummarkieren auf erst DSCP_SIG, dann DSCP_UC an DF_R weiterleitet. Während DF_2 seinen TriggerUC vorerst unterdrückt, fordert DF_R mit seinem Trigger sofort beim Weiterleiten des ersten Datenpakets NRS-Zustand beim DRM an. Mit dem Beitritt des ersten Empfängers zur Gruppe G ist nicht nur nötig, den per Join zum RPL aufgebauten Multicastpfad mit NRS-Zustand zu versehen, sondern auch die Ummarkierung auf DSCP_LE durch DF_S aufzuheben. Der DRM sendet daher an DF_S ein Conf(Ack,Keep), dessen Empfang dieser mittels ConfReply (OK) bestätigt, sowie ein Conf(Keep) an alle Router vom RPL bis zum Empfänger, im Beispiel DF_2 und DF_R. Die Router zwischen DF_S und RPL, hier nur DF_1, wurden eingangs ja bereits mittels Conf(Keep) konfiguriert und markieren auch den neuen DSCP nicht um. Das gesetzte ACKbit für DF_S ist nötig, um den Verlust der Conf-Nachricht erkennen zu können. Im Gegensatz zu den übrigen Routern besitzt DF_S bereits gültigen NRS-Zustand und wird daher keine Trigger mehr senden, sollte ihn das Conf(Ack,Keep) nicht erreichen, während die übrigen Router mittels Trigger bzw. TriggerUC erneut ihre Konfiguration beim DRM

anfordern werden. Bleibt das OK aus, wiederholt der DRM das Conf(Ack,Keep) an DF_S nach T_{RTO} (s. Abschnitt 4.2.7). Auch bei BIDIR-PIM sollte die Rate R, mit der Trigger wiederholt werden, und die Zeit T_S, in der TriggerUC unterdrückt werden, Gleichung 4.4 bzw. Gleichung 4.5 erfüllen, damit die Konfiguration durch den DRM abgeschlossen ist, bevor erneut ein Trigger bzw. ein TriggerUC gesendet werden würde.

Der abschließende Abbau von Multicastpfad und NRS-Zustand erfolgt analog zu den Protokollen PIM-SSM und PIM-DM, bei BIDIR-PIM jedoch mittels Prune(*,G) statt Prune(S,G). DF_R zeigt den Austritt des letzten Empfängers an der betreffenden Schnittstelle mittels TriggerStop beim DRM an. Stellt der DRM fest, dass dies (wie im Beispiel) der letzte Empfänger in dieser Multicastgruppe überhaupt war, empfiehlt es sich, erneut einen unnötigen Ressourcenverbrauch zu vermeiden und die Ummarkierung der empfängerlosen Datenpakete von Q auf DSCP_LE zu reaktivieren. Der DRM rekonfiguriert daher den NRS-Zustand mit einer Conf(Ack,Remark)-Nachricht an DF_S, der den Empfang durch ein OK positiv bestätigt. Die Datenpakete fließen wieder mit DSCP_LE markiert bis zum RPL, bis erneut ein Empfänger der Gruppe beitritt.

Stellt S das Senden ein, wird der MFC-Eintrag in DF_S und DF_1 gelöscht. Da keiner der Router ein gesetztes DRbit hat, wird kein TriggerStop zum DRM gesendet. Dieser erkennt einerseits bei der nächsten Abfrage nach T_{check} anhand der dann empfangenen Fail von DF_S und DF_1, dass auch der Multicastpfad S–DF_S–DF_1 abgebaut wurde und kann seinen internen Zustand anpassen. Wird wie im Beispiel auf DSCP_LE ummarkiert, ist dies die Variante mit dem geringsten Signalisierungsaufwand. Fließen dagegen Pakete mit dem DSCP „Q“ der Quelle bis zum RPL, stehen solange die reservierten Ressourcen keinen neuen Reservierungen zur Verfügung. Dies lässt sich vermeiden, indem mit einem Conf(Ack,Keep,DR) an DF_S das DRbit gesetzt wird, so dass DF_S mittels TriggerStop den DRM informiert, wenn S das Senden einstellt und DF_S seinen MFC-Eintrag löscht. Es sei angemerkt, dass Ressourcen allerdings auch dann sofort freigegeben werden können, wenn die Quelle über die Dienstgütesignalisierung das Ende ihrer Reservierung angezeigt hat bzw. die Reservierung von ihr nicht mehr aufgefrischt wurde. Die daraufhin erfolgende Löschung des Profils im First Hop Router durch den DRM sorgt dafür, dass die Quelle keine hochprioren DSCPs mehr nutzen kann.

4.2.12 Weitere Routingverfahren und Multicastroutingprotokolle

Neben den aktuellen Multicastroutingprotokollen der PIM-Familie existieren weitere früher für das Internet standardisierte und daher nur IPv4 unterstützende Multicastroutingprotokolle: DVMRP, MOSPF und CBT. Neu standardisiert wurde mit IGMP/MLD-Proxying zudem ein halbautomatisches Verfahren für einfache Netzwerktopologien als Alternative zu vollwertigem Multicastrouting. Equal Cost Multipath Routing ist eine einfache Möglichkeit zur Lastverteilung, falls mehrere gleich gute alternative Pfade zu einer Quelle bzw. Rendezvousstelle zur Verfügung stehen. Im Folgenden wird kurz besprochen, was bei der Anwendung von DSMC auf diese Protokolle und Verfahren zu beachten ist.

4.2.12.1 Equal Cost Multipath Routing

Entscheidend bei ECMP (s. Abschnitt 2.1.4) ist, dass der DRM die von den Routern getroffene Routingentscheidung vorhersagen kann. Dem DRM muss dazu von jedem Router die verwendete Hashfunktion und Hashmaske bekannt sein und ob ggf. Relaxed ECMP

verwendet wird. Andernfalls wird der DRM nach einem Trigger einen abweichenden Pfad zum neuen Empfänger berechnen, der mit TriggerUCs nur nach und nach erkannt werden kann. Die Pfadberechnung durch den DRM ändert sich nicht wesentlich. Die Metrik ist lediglich nicht mehr allein von der Zieladresse (Quelle oder Rendezvousstelle) abhängig, sondern bezieht auch die Gruppenadresse, und bei Relaxed ECMP zusätzlich die Metrik, zum Next Hop mit ein.

4.2.12.2 IGMP/MLD-Proxying

IGMP/MLD-Proxying (s. Abschnitt 2.2.6) verwendet, wie BIDIR-PIM, einen bidirektionalen Baum. Es werden allerdings die Upstreamschnittstellen nicht dynamisch vom Protokoll, sondern statisch (über das Netzwerkmanagement) konfiguriert. Die Rolle des Designated Forwarders übernimmt der Querier im Rahmen des IGMP- bzw. MLD-Protokolls. Dessen Wahl erfolgt wie bei BIDIR-PIM automatisch. Der Wahl der Querier, und damit der Aufbau des Spannbaums, erfolgt unabhängig von der eigentlichen Multicastkommunikation. Das zu BIDIR-PIM gesagte gilt daher unverändert für IGMP/MLD-Proxying. Es sei angemerkt, dass durch quellenspezifische Berichte, anders als bei BIDIR-PIM, quellenspezifischer „Routing"-Zustand angelegt wird. Dessen Behandlung unterscheidet sich jedoch bzgl. DSMC, wie bereits bei BIDIR-PIM erläutert, nicht von quellenunspezifischem Zustand.

4.2.12.3 DVMRP

Das Multicastroutingprotokoll DVMRP bestimmt seine MRIB per Distance Vector Routing selbst. Zentrales Ressourcenmanagement, wie es DSMC voraussetzt, ist jedoch nur bei vollständiger Kenntnis von Topologie und Routing in der Domäne möglich. Der DRM muss daher die MRIB jedes Routers auf anderem Wege, d. h. über das Netzwerkmanagement, ermitteln und sich von Ausfällen bzw. Routingänderungen informieren lassen (s. Abschnitt 4.4.1).

Im Vergleich zu PIM-DM ergibt sich bei DVMRP der Vorteil, dass es kein Überschreiben von Prunes gibt. Der Broadcast-and-Prune-Zyklus läuft somit schneller ab als bei PIM-DM. Während bei PIM-DM die Unterdrückungszeit T_S bei Auftreten von LANs mit drei und mehr PIM-DM-Routern nach Gleichung 4.7 gewählt werden muss, kann sich bei DVMRP die Unterdrückungszeit immer nach Gleichung 4.6 richten, egal ob es LANs mit drei oder mehr DVMRP-Routern gibt oder nicht. Das periodische Broadcasten der Pakete einer Quelle im gesamten Netz stellt kein Problem dar, da durch das darauffolgende Prune die Ausgangsschnittstellen wieder deaktiviert werden. Beim nächsten Broadcast-and-Prune-Zyklus beginnt die Unterdrückung der TriggerUC somit erneut; es kommt nicht zum Senden eines TriggerUC bei jedem neuen Broadcast. Um den Ressourcenverbrauch durch die Broadcast-and-Prune-Zyklen gering zu halten, sollte DSCP_SIG gleich DSCP_UC oder DSCP_LE gesetzt werden. Noch besser wird mit einer großen Prune Holdtime (max. zwei Stunden) der diesbezügliche Ressourcenverbrauch von vornherein gering gehalten.

Sofern die in DVMRP optionalen Flash Updates verwendet werden und somit eine schnelle Konvergenz des Routings nach einem Linkausfall sichergestellt ist, steht dem Einsatz von DVMRP für dienstgüteunterstützten Multicast unter Anwendung des Basisverfahrens von DSMC nichts entgegen. Ohne Flash Updates dürfte der nach einer Routingänderung ggf. entstehende Konnektivitätsverlust zu lang für eine sinnvolle Bereitstellung von Dienstgüte sein.

4.2.12.4 MOSPF

Das nur Any Source Multicast unterstützende MOSPF verwendet als Erweiterung des OSPF-Protokolls die durch OSPF aufgebaute RIB als MRIB. Damit ist sichergestellt, dass der am Unicastrouting, also hier an OSPF, teilnehmende DRM dieselbe Sicht auf das Multicastrouting hat. Es kommt somit zu keinen (dauerhaften) Inkonsistenzen zwischen den tatsächlichen und den durch den DRM angenommenen Multicastpfaden. Implementiert der DRM nicht nur OSPF, sondern MOSPF, kennt er darüber hinaus auch alle Gruppenmitglieder. Die Trigger sind trotzdem notwendig, da sie nicht nur neue Gruppenmitglieder, sondern auch die Einrichtung des Weiterleitungszustands anzeigen, der bei MOSPF erst mit dem ersten Datenpaket einer Quelle angelegt wird. Der TriggerStop bzw. Regel **TrSt** kann dagegen entfallen, da der DRM das Ende einer Gruppenmitgliedschaft bereits durch seine Teilnahme an MOSPF erkennen kann.

Es ist zu beachten, dass bei MOSPF MaxSignalingDelay deutlich größer ausfällt als bei den Protokollen der PIM-Familie. Ein PIM-Router kennt direkt die Liste der aktiven Ausgangsschnittstellen: jene, über die er eine Join-Nachricht empfangen hat. Ein MOSPF-Router muss dagegen nach dem Empfang des ersten Datenpakets einer neuen Quelle zuerst den Shortest-Path-Algorithmus ausführen, um aus seiner Link-State-Datenbank den neuen MFC-Eintrag für diese Quelle zu berechnen. Die anfängliche Unterdrückungszeit der TriggerUC bei MOSPF muss dies berücksichtigen und entsprechend größer gewählt werden.

Alternativ kann TriggerUC bzw. Regel **TrUC** ganz entfallen. Bei MOSPF ist, im Gegensatz zu den PIM-Routingprotokollen, die Multicastroutinginstanz nicht mehr unabhängig von der Unicastroutinginstanz und kann nicht getrennt ausfallen. Der DRM kann den Ausfall eines Routers damit direkt durch seine Teilnahme an (M)OSPF erkennen und muss nicht erst durch TriggerUC und anschließende Abfrage per Netzwerkmanagement (SNMP) den Ausfall der Multicastroutinginstanz feststellen, um seine Sicht dem tatsächlichen Multicastrouting anzupassen. Die Teilnahme des DRMs an MOSPF hat darüber hinaus den Vorteil, dass später hinzukommende Empfänger auf bereits konfigurierten Pfaden mit konfiguriertem NRS-Zustand automatisch erkannt werden und sich eine diesbezügliche Überprüfung erübrigt (s. Abschnitt 4.2.13.4).

DSMC unterstützt somit auch Multicastrouting auf Basis von MOSPF. Sofern der DRM nicht nur am OSPF-, sondern auch am MOSPF-Routing teilnimmt, genügt es zudem, von den drei Triggertypen Trigger, TriggerUC und TriggerStop für MOSPF nur den Trigger nach Regel **Tr** zu implementieren.

4.2.12.5 CBT

CBT verwendet, wie die Protokolle der PIM-Familie, die durch ein separates Unicastroutingprotokoll aufgebaute MRIB als Basis für die Multicastpfadbestimmung. CBT setzt, wie BIDIR-PIM, zur Paketweiterleitung einen bidirektionalen Baum mit dem Core als Wurzel ein. Wie PIM-SM werden für Quellen auf Links ohne Empfänger in der Gruppe Tunnel zum Core eingerichtet. DSMC kann daher auf CBT-basiertes Multicastrouting angewandt werden, wobei das zu BIDIR-PIM, bzw. das zum Register Tunnel bei PIM-SM (s. Abschnitt 4.3.4) Gesagte entsprechend für CBT gilt. Einen SPT Switchover kennt CBT nicht, eine diesbezügliche Erweiterung von DSMC, wie bei PIM-SM, ist daher nicht nötig.

4.2.13 Designalternativen

Bei der Darstellung des Basisverfahrens wurden bereits einige Designvarianten wie Sequenznummern, DRbit nachführen und Exponential Backoff für Trigger diskutiert und erläutert, warum sie im Rahmen von DSMC keine Verwendung finden. Im Folgenden werden weitere naheliegende Designalternativen vorgestellt, die einer ausführlicheren Diskussion bedürfen.

4.2.13.1 NRS-Zustand vom MFC-Eintrag trennen

Es ist denkbar, den Diffserv-spezifischen Zustand zur Lösung des NRS-Problems unabhängig vom MFC, d. h. dem für die eigentliche Paketweiterleitung nötigen Zustand, zu verwalten. Dadurch wird es möglich, den NRS-Zustand vor dem MFC-Eintrag anzulegen, so dass vom ersten Datenpaket an die Dienstgüte erbracht werden kann. Viele Implementierungen legen allerdings erst mit dem ersten Datenpaket einer Quelle und nicht mit dem Aufbau des Routingzustands durch Join-Nachrichten bzw. Gruppenmitgliedschaftsberichte einen neuen MFC-Eintrag an. Die ersten Datenpakete erfahren daher entweder eine zusätzliche Verzögerung, da sie durch die Routingebene des Routers geleitet werden, oder sie werden sogar ganz verworfen, bis ein passender MFC-Eintrag in der Weiterleitungsebene eingerichtet ist. Hier ist eine anfangs verminderte Dienstgüte nicht zu vermeiden, was den Vorteil zum Teil wieder zunichte macht.

Dieser Vorteil wird durch eine komplexere Implementierung und damit i. A. geringere Robustheit und schlechtere Wartbarkeit erkauft. erkauft. Ein separat vom MFC verwalteter NRS-Zustand muss beim Anlegen eines MFC-Eintrags aufgefunden, mit ihm verknüpft und danach konsistent gehalten werden. Integriert man dagegen den NRS-Zustand in den MFC-Eintrag, so wie es in DSMC vorgesehen ist, können beide Zustände leicht konsistent gehalten werden. Das Löschen eines MFC-Eintrags sorgt automatisch für das Löschen des zugehörigen NRS-Zustands. Nicht mehr benötigte NRS-Zustände werden dadurch nicht nur nach dem Austritt von Empfängern oder Quellen, sondern auch nach einer Routingänderung zusammen mit dem MFC-Eintrag automatisch entfernt. Des Weiteren wird der NRS-Zustand erst dann angelegt, wenn dieser wirklich benötigt wird. Ohne den zugehörigen MFC-Eintrag bleibt ein getrennt verwalteter NRS-Zustand ungenutzt und belegt unnötig den Speicher in der Weiterleitungsebene.

4.2.13.2 NRS-Zustand separat für jeden DSCP

Der NRS-Zustand unterscheidet nicht nach DSCPs. Würde der NRS-Zustand nicht für alle DSCPs gemeinsam sondern separat für jeden der 64 möglichen DSCPs gelten, könnte innerhalb einer Multicastgruppe eine abgestufte Dienstgüte je nach Anbindung der Empfänger umgesetzt werden: Eine Quelle markiert ihre Datenpakete mit unterschiedlichen DSCPs und ordnet sie so unterschiedlichen Verhaltensaggregaten zu. Je nach Anbindung der Empfänger sind nur für einige dieser Verhaltensaggregate ausreichend Ressourcen vorhanden, die Pakete der übrigen werden auf DSCP_LE ummarkiert. Dies kann beispielsweise im Rahmen von Layered Multicast genutzt werden.

Eine solche Differenzierung des NRS-Zustands nach DSCPs hat jedoch zwei entscheidende Nachteile: Obwohl für einige Verhaltensaggregate in einigen Teilen des Multicastbaums nicht genügend Ressourcen verfügbar sind, werden diese trotzdem dorthin weitergeleitet, wenn auch nach Ummarkierung auf DSCP_LE. Sinnvoll wäre es dagegen, die Weiterleitung ganz zu unterbinden, da der ummarkierte Verkehr aufgrund der fehlenden Dienstgüte und des daraus resultierenden Paketverlusts sehr wahrscheinlich für die Empfänger nutzlos

ist. Ein solches selektives Prunen von Verhaltensaggregaten ist aber nur dann möglich, wenn sie verschiedene Multicastgruppen nutzen. Des Weiteren haben Empfänger keine Kontrolle darüber, welche der Verhaltensaggregate auf DSCP_LE abgebildet werden und welche nicht. Bauen die Daten in den Verhaltensaggregaten aufeinander auf, kann dies dazu führen, dass auch die Daten, die weiterhin Dienstgüte erfahren, nutzlos werden, wenn sie von anderen Daten abhängen, die auf DSCP_LE ummarkiert wurden.

Diesen Nachteilen einer Multicastgruppe mit nach DSCPs differenzierter Ummarkierung steht als einziger Vorteil, verglichen mit mehreren Multicastgruppen mit einheitlicher Ummarkierung, die Reduzierung von Multicastrouting- und -weiterleitungszustand gegenüber. NRS-Zustand wird nicht eingespart. Der zur Konfiguration des NRS-Zustands nötige Signalisierungsaufwand bleibt entsprechend unverändert. Zusätzlich erhöht sich die Komplexität der Implementierung der Weiterleitungsebene, wenn der NRS-Zustand speicherplatzsparend abgelegt und nicht für 64 verschiedene DSCPs das 64-fache an Speicher belegen werden soll.

4.2.13.3 Separate Triggertypen je Multicastroutingprotokoll

Die von den rendezvousstellenbasierten Protokollen PIM-SM und BIDIR-PIM benötigte Zuordnung von Gruppen- zu Rendezvousstellenadressen kann dynamisch vom Bootstrap-Router-Mechanismus oder einem proprietären Protokoll wie Ciscos Auto-RP-Protokoll verteilt und synchron gehalten werden. Sofern in der Domäne gleichzeitig unterschiedliche Multicastroutingprotokolle eingesetzt werden, lässt sich daraus implizit die Aufteilung der Gruppenadressen auf die verschiedenen Multicastroutingprotokolle ableiten.

Wie bei allen verteilten dynamischen Verfahren kann es hier zu transienten Inkonsistenzen kommen. Der DRM würde dann einen falschen Pfad aus einem Trigger ableiten und könnte den tatsächlichen Multicastpfad erst nach und nach mit dem Eintreffen der TriggerUC erkennen und konfigurieren. Verschiedene Triggertypen für verschiedene Multicastroutingprotokolle würden es dem DRM erlauben, eine solche Inkonsistenz in der Aufteilung der Gruppenadressen auf die Multicastroutingprotokolle schon am Triggertyp zu erkennen.

Allerdings müssten, um trotz einer solchen Inkonsistenz zumindest für die im Trigger genannte Multicastgruppe den korrekten Pfad ermitteln zu können, je nach Multicastroutingprotokoll noch weitere Informationen in der Triggernachricht vermerkt werden. Für die Protokolle PIM-SM und BIDIR-PIM wäre dies die Adresse der Rendezvousstelle bzw. des Rendezvousstellenlinks (sofern keine Adressen mit eingebetteter Rendezvousstellenadresse verwendet werden). Besitzen jedoch auch die Router untereinander inkonsistente Zuordnungen, ist während dieser Zeit das Multicastrouting für die betroffenen Gruppen ohnehin gestört. Bestenfalls für einen Teil der Empfänger und/oder Quellen lassen sich funktionierende Pfade etablieren. Separate Triggertypen je Multicastroutingprotokoll lösen das Problem somit nur ansatzweise. Sinnvoller ist es, das Ende der Inkonsistenz abzuwarten und erst dann den neuen Pfad zu konfigurieren.

4.2.13.4 TriggerStop von jedem Router

Das LAN3 der Topologie in Abbildung 4.9 weist eine Besonderheit auf: An ihm sind gleichzeitig Empfänger und mehr als ein Router angeschlossen. Solche LANs können Gruppenmitglieder besitzen ohne ein Blattknoten[6] im Multicastverteilbaum zu sein. Trigger

[6]Nur Router und LANs (=Transitlinks) werden als Knoten des Routinggraphen betrachtet, nicht aber Hosts (=Stummellinks) und Punkt-zu-Punkt-Links, da an ihnen keine Verzweigung erfolgen kann, s. Abschnitt 2.1.3.1.

(und TriggerStop) werden aber nur von Blattknoten generiert, so dass Empfänger auf Nichtblattknoten dem DRM verborgen bleiben. Hierbei führt nur der Fall zum Problem, bei dem der Empfänger später beitritt, wenn ein Link schon Teil des Verteilbaums ist. Der umgekehrte Fall, bei dem ein Link mit Empfänger ursprünglich Blattknoten und erst später durch Beitritt weiterer Empfänger stromabwärts zu einem inneren Knoten des Verteilbaums wird, führt lediglich zu einem überflüssigen TriggerStop.

Zur Veranschaulichung sei, wie im Beispiel zu BIDIR-PIM Abschnitt 4.2.11.3, der Multicastverteilbaum (H2)LAN2—Rt2—Rt1—Rt3—LAN3—Rt4—LAN4(H4) aufgebaut worden. LAN4 ist Blatt im Multicastverteilbaum, nicht aber LAN3. Würde nun Host H3 ebenfalls der Gruppe beitreten, wird kein Trigger zum DRM gesendet, da sich der NRS-Zustand NRSstate(S,G,LAN3) auf Rt3 bereits im Zustand Keep befindet. Der DRM erfährt daher nicht vom Gruppenbeitritt von H3. Und wie bei den Erläuterungen zum DRbit beschrieben, wird das DRbit, hier auf Rt3, nicht der lokalen Gruppenmitgliedschaft nachgeführt, also nicht nachträglich mit dem Beitritt von H3 gesetzt. Tritt H4 aus der Gruppe aus, sollte der DRM eigentlich nur die nun nicht mehr benötigten Ressourcen auf LAN4 freigeben. Tatsächlich wird der DRM jedoch die Ressourcen entlang des gesamten Pfads wieder für andere Kommunikationsbeziehungen freigeben, da er nichts vom Empfänger H3 weiß. Die Datenpakete von H2 fließen weiterhin entlang des um LAN4 gekürzten Multicastverteilbaums, aber ohne Wissen des DRMs, so dass es hier zum NRS-Problem kommen kann, wenn die genutzten Ressourcen vom DRM neuen Kommunikationsbeziehungen zugesichert werden. Im betrachteten Beispiel konfiguriert zwar der DRM den NRS-Zustand in Rt2 mittels Conf(Ack,Remark) um, so dass es in diesem Fall nicht zum NRS-Problem kommt. Jedoch erfährt H3 dann keine Dienstgüte mehr, obwohl Ressourcen vorhanden sind, was genauso ungünstig ist. Der DRM muss daher mittels Conf(Ack) den etablierten Multicastverteilbaum auf allen in Frage kommenden LANs auf evtl. nachträglich hinzugekommene Empfänger hin überprüfen, bevor er die Ressourcen freigibt oder die Bereitstellung der Dienstgüte mittels Conf(Ack,Remark) stoppt (s. Abschnitt 4.4.4.3).

Statt des Abfragens durch den DRM mittels Conf(Ack) kann alternativ jeder Router grundsätzlich beim Inaktivieren eines Ausgangs den DRM mit einem TriggerStop informieren. Regel **TrSt** ist dazu wie folgt zu modifizieren:

TrSt′ **TriggerStop beim Inaktivieren einer Ausgangsschnittstelle.** Wird eine Ausgangsschnittstelle inaktiv, wird der DRM mit einem *TriggerStop* darüber informiert.

Das DRbit, dass im Wesentlichen dazu dient zu entscheiden, welcher Router einen TriggerStop sendet, ist somit überflüssig: Regel **DRbt** entfällt und in den Regeln **Conf**, **Rply** und **AND** sind die Verweise auf das DRbit zu entfernen. Man beachte, dass sich diese Variante auch mit dem Basisverfahren von DSMC nachbilden lässt. Hierzu muss der DRM bei der Konfiguration von NRS-Zustand lediglich immer auch das DRbit setzen.

Wenn für die Mehrzahl oder alle LANs einer Domäne der beschriebene Fall von LAN3 zutrifft, kann diese Variante Signalisierungsnachrichten einsparen, da eine Nachricht TriggerStop zwei Nachrichten Conf(Ack) und ConfReply ersetzt. Wenn aber bereits die Hälfte aller Links einer Domäne keine solchen Links sowohl mit Empfängern wie mit abhängigen Routern sind, erzielt diese Variante im Mittel keine Einsparung mehr. In Domänen ganz ohne solche Links kommt das Basisverfahren ganz ohne Conf(Ack,Keep)-Abfragen aus und alle in dieser Variante generierten TriggerStop sind nutzlos.

Wenn die Gruppenmitgliedschaft bzw. das Ende einer Gruppenmitgliedschaft anderweitig bekannt ist, z. B. im Rahmen von AAA (Authentication, Authorization, Accounting) oder durch Verwendung von MOSPF, das die Gruppenmitgliedschaft flutet, ist überhaupt kein TriggerStop mehr nötig, selbst der des Basisverfahrens nicht mehr. Statt Regel **TrSt** zu modifizieren, kann diese bzw. Regel **TrSt′** vollständig entfallen. Auch dies lässt sich mit dem Basisverfahren von DSMC nachbilden. Hierzu muss der DRM bei der Konfiguration von NRS-Zustand lediglich immer auch das DRbit zurücksetzen.

Man beachte aber, dass solche externen Methoden zur Erkennung der Gruppenmitgliedschaft nicht den Trigger Regel **Tr** ersetzen können, da ein Trigger nicht nur neue Empfänger, sondern vor allem eine wichtige Aufgabe zur Synchronisation besitzt und die vollständig abgeschlossene Einrichtung des Weiterleitungszustands entlang des gesamten neuen Multicastpfads anzeigt.

4.2.13.5 Trigger von jedem Router

Es ist ausschließlich mittels eines nicht verzögerten Triggers beim Weiterleiten unter ungültigem NRS-Zustand möglich, das NRS-Problem zu vermeiden und alle Router entlang eines neuen Multicastpfads zu erkennen, um anschließend deren NRS-Zustand zu konfigurieren. Regel **TrUC** entfällt und Regel **Tr** ist wie folgt zu modifizieren:

Tr′ **Trigger beim Weiterleiten.** Wird ein Paket unter ungültigem NRS-Zustand, also ein Paket, auf das der NRSstate(S,G,Oif) wirkt, weitergeleitet, wird der DRM ohne Verzögerung ratenlimitiert mit einem *Trigger* über den ungültigen NRS-Zustand informiert.

Nachteil ist ein deutlich erhöhtes Aufkommen an (unterschiedlichen) Triggern und damit einhergehender erhöhter Belastung des DRMs und schlechterer Skalierbarkeit. Statt nur einem Trigger pro neuem Empfänger bzw. Multicastpfad wird ein Trigger je Router entlang des neuen Pfads generiert, der vom DRM verarbeitet werden muss.

4.2.14 SimpleDSMC

Kombiniert man die beiden letzten vorgestellten Designalternativen eines von jedem Router gesendeten Triggers mit der eines von jedem Router gesendeten TriggerStop, sendet also jeder Router sofort beim Auf- wie auch beim Abbau eine Triggernachricht an den DRM, oder kombiniert man die Variante eines von jedem Router gesendeten Triggers mit einer von DSMC unabhängigen externen Methode zur Erkennung (des Endes) der Gruppenmitgliedschaft, vereinfacht sich das gesamte Verfahren weiter.

Neben den neun Regeln zur Lösung des NRS-Problems nach Abschnitt 4.2.1, der unveränderten Informationsregel **TSGO** und den drei modifizierten Informations- und Nachrichtenregeln **AND**, **Conf** und **Rply**, bei denen der Bezug zum entfallenen DRbit gestrichen wurde (s. Abschnitt 4.2.13.4), kommt für die Variante mit DSMC-externer Methode Nachrichtenregel **Tr′**, für die Methode mit TriggerStop zusätzlich Nachrichtenregel **TrSt′** hinzu.

Diese beiden Varianten sollen entsprechend der eingesetzten Methode zur Erkennung des Endes einer Gruppenmitgliedschaft als **SimpleDSMCext** (Erkennung durch DSMC-externes Verfahren) und **SimpleDSMCwithStop** (Erkennung mittels TriggerStop) bezeichnet werden. SimpleDSMC stellt quasi die einfachste Art und Weise dar, wie man die zur Lösung

des NRS-Problems in Abschnitt 4.2.1 vorgestellte Erweiterung der Weiterleitungsebene um eine geeignete Signalisierung zu deren Konfiguration ergänzen kann: Jeder Router fordert unmittelbar für sich den benötigten Zustand beim DRM an. SimpleDSMC dient bei der Untersuchung der Skalierbarkeit von DSMC in Kapitel 5 als Referenz, um die Einsparungen bei der Signalisierung quantifizieren zu können.

4.2.15 Reduzierung getriggerter Conf-Nachrichten für IPv6

DSMC hält die Anzahl an Trigger-Nachrichten an den DRM gering, nicht aber die der Conf-Nachrichten an die Router, da dies nur eine geringe Aufwandsverringerung für den Ressourcenmanager bedeuten würde und somit nur geringen Einfluss auf die Skalierbarkeit von DSMC hat. Möchte man auch Signalisierungsverkehr einsparen, lässt sich zumindest für IPv6 die Anzahl der aufgrund einer Triggernachricht gesendeten Conf-Nachrichten auf eine je neu zu konfigurierenden Pfad reduzieren, indem ein geeigneter neuer IPv6-Erweiterungskopf *DSMC Conf Header* ähnlich dem des (Unicast) Source Routing Header [48] sowie ein neuer Wert für die Router Alert Option [134] definiert wird. Dies soll in Grundzügen hier kurz skizziert werden.

Der DSMC Conf Header würde mindestens die jeweiligen Ausgangsschnittstellen und zu konfigurierenden NRSstates und DRbits in Form einer Liste, einen Zeiger auf das aktuelle Element der Liste, ein ACKbit sowie die Adresse von Quelle und Gruppe des zu konfigurierenden Multicastpfads enthalten. Der DRM sendet die Konfigurationsnachricht mit dem DSMC Conf Header an den Router am Beginn des zu konfigurierenden Multicastpfads, im Folgenden kurz Kopfrouter genannt. Der Router tauscht die Adresse des DRMs (Quelladresse) und seine Adresse (Zieladresse) im IPv6-Kopf gegen die im DSMC Conf Header vermerkte Quell- und Gruppenadresse der zu konfigurierenden Multicastpfads aus, so dass im DSMC Conf Header jetzt die Adresse des DRMs im Feld der Quelladresse und die des Kopfrouters im Feld der Gruppenadresse steht. Die Konfigurationsnachricht wird ab dem Kopfrouter wie ein Multicastpaket weitergeleitet. Der Zeiger identifiziert Ausgangsschnittstelle und zu konfigurierende Werte für NRSstate und DRbit. Nach Konfiguration der Schnittstelle wird der Zeiger auf das nächste Element der Liste gesetzt und das Multicastpaket nicht über alle aktiven Ausgangsschnittstellen, sondern nur über diese gerade konfigurierte Ausgangsschnittstelle weitergeleitet.

Wünscht der DRM eine Bestätigung, was er durch ein gesetztes ACKbit anzeigt, sendet der letzte Router die Konfigurationsnachricht zurück zum DRM, wozu er wieder beide Adressen vom IPv6-Kopf und DSMC Conf Header austauscht, so dass im DSMC Conf Header wieder Quell- und Gruppenadresse stehen. Die DRM-Adresse steht jetzt allerdings im Feld der Zieladresse des IPv6-Kopfs und als Quelladresse im IPv6-Kopf trägt der Router seiner eigene Adresse ein. Praktischerweise hat der DRM die Konfigurationsnachricht mit einem UDP-Paket versehen, so dass die Antwort bei ihm gleich an den UDP-Port hochgereicht wird, den er sonst für ConfReply-Nachrichten verwendet. Das UDP-Paket enthält zumindest noch einmal die Adresse des Kopfrouters, damit die Antwort eindeutig zugeordnet werden kann (sie wurde ja vom letzten Router durch dessen Adresse überschrieben). Ebenso zeigt einer der Router entlang des Pfads einen aufgetretenen Fehler (nicht aktive Ausgangsschnittstelle) durch sofortiges Zurücksenden der Konfigurationsnachricht zum DRM mit zurückgesetztem ACKbit an, denn ab diesem Router ist wegen Inkonsistenzen eine weitere Konfiguration entlang des Pfads ohnehin nicht mehr möglich.

Damit der DSMC Conf Header ausgewertet wird, muss ein neuer Wert für die Router Alert Option definiert werden. Er besagt, dass ein DSMC Conf Header folgt, der im Falle

einer Multicastadresse als Zieladresse im IPv6-Kopf ausgewertet werden muss. Im Falle einer Unicastadresse, also auf dem Unicastpfad vom DRM zum Kopfrouter und später vom letzten Router oder einem Router mit aufgetretenem Fehler zurück zum DRM ist eine Auswertung des DSMC Conf Headers dagegen unnötig.

Ggf. kann es sinnvoll sein, DRM-Adresse und Quelladresse am Kopfrouter nicht auszutauschen, falls nicht sicher ist, dass alle Router den DSMC Conf Header unterstützen und ein ICMP Parameter Problem zurück an den DRM gehen soll. Die Quelladresse zum Auffinden des MFC-Eintrags steht dann statt im IPv6-Kopf weiterhin im DSMC Conf Header. Statt eines neuen Werts für die Router Alert Option wäre eine neue Hop-by-Hop Option *DSMC Conf Option* zu definieren mit den ersten beiden Bits im Optionstyp gleich 10 (wenn Option unbekannt, ICMP-Nachricht zurück an Absender auch bei Multicast). Um ICMP-Nachrichten längs der Unicastpfade von/zum DRM zu vermeiden, könnte der Kopfrouter die Bits von 00 (Option ignorieren, wenn unbekannt) auf 10 und der letzte Router zurück von 10 auf 00 ändern, wozu entsprechend ein zweiter neuer Hop-by-Hop-Optionstyp zu definieren wäre. Ein Ändern von Optionstypen ist im IPv6-Standard allerdings nicht vorgesehen (im Gegensatz zum Ändern des Optionswerts).

4.3 Erweiterung für PIM-SM

Das bisher beschriebene Basisverfahren von DSMC ist prinzipiell für alle Multicastroutingprotokolle geeignet, bei denen der DRM aus Quell- und Gruppenadresse sowie dem Link des Empfängers, also den im Trigger transportierten Informationen, den Multicastpfad ableiten kann. PIM Sparse Mode (PIM-SM) hat im Fall von Any Source Multicast (ASM) jedoch eine besondere Eigenschaft, die eine Erweiterung des Basisverfahrens erfordert: PIM-SM verwendet zwei verschiedene Multicastbäume, den Rendezvousstellenbaum und den quellenspezifischen kürzesten Baum. Dabei wird normalerweise zuerst der RPT aufgebaut und dann von der Rendezvousstelle und den Designated Routern der Empfänger gemäß lokaler Policy, repräsentiert durch die PIM-SM-Funktion SwitchToSptDesired(S,G), auf den SPT umgeschaltet. Mittels TriggerUC-Nachrichten würde diese Umschaltung bzw. Routingänderung als Inkonsistenz zwischen der Sicht des DRMs und des tatsächlichen Routings auch vom Basisverfahren erkannt und gehandhabt werden können. Die Konfiguration würde jedoch nur langsam nach und nach und ineffizient mit vielen TriggerUC-Nachrichten erfolgen (s. Abschnitt 4.2.5).

Das Umschalten auf den SPT stellt nichts weiter als eine Routingänderung dar. DSMC versucht wie bei jeder anderen Routingänderung auch, bestmöglich im Nachhinein darauf zu reagieren. Bestmöglich heißt, abzuwarten, bis die durch die Routingänderung verursachte Pfadänderung vollständig abgeschlossen ist und erst dann den neuen Pfad mit NRS-Zustand zu konfigurieren. „Gewöhnliche“ Routingänderungen, z. B. durch Linkausfall, führen zu Änderungen der dem Multicastrouting zugrundeliegenden MRIB und werden durch Teilnahme am Link State Unicastrouting bzw. bei Distance Vector Routing über das Netzwerkmanagement erkannt. Die Umschaltung auf den SPT stellt dagegen eine Routingänderung im Multicastrouting selbst dar. Bei PIM-SM hat jeder Router nur eine lokale Sicht auf das Multicastrouting, kennt also nur die Multicastpfade, die über ihn geroutet werden. Der DRM kann daher nicht durch Teilnahme an PIM-SM vom Umschalten bei jeder beliebigen Multicastkommunikation in seiner Domäne erfahren. Die Erweiterung führt daher den neuen DSMC-Nachrichtentyp TriggerSwitched ein, der den DRM über das Umschalten genau dann informiert, wenn das Umschalten auf den SPT

abgeschlossen ist. Der DRM bestätigt den Empfang mit dem ebenfalls neuen Nachrichtentyp TriggerSwitchedAck. Ein Umschalten zurück auf den RPT findet bei PIM-SM nicht statt. Der als drittes für PIM-SM neu eingeführte DSMC-Nachrichtentyp TriggerRpt zeigt den Aufbau des Rendezvousstellenbaums an im Unterschied zum quellenspezifischen kürzesten Baum, der wie gewohnt mittels Trigger signalisiert wird.

4.3.1 Ergänzende Regeln für PIM-SM

Um bei PIM-SM pro Umschaltung vom RPT auf den SPT bzw. pro neuem Multicastpfad mit nur einer Triggernachricht auszukommen, ist es notwendig, weitere Bedingungen und Triggernachrichten einzuführen. Das Basisverfahren von DSMC wird für PIM-SM daher wie folgt erweitert:

PIM-SM-Regeln:

TrSM **Trigger beim Weiterleiten als DR vom SPT.** Wird ein Paket unter ungültigem NRS-Zustand, also ein Paket, auf das der NRSstate(S,G,Oif) wirkt, vom *quellenspezifischen kürzesten Baum* (SPTbit(S,G) = TRUE) an einen direkt angeschlossenen Empfänger oder auf einen als DSMC-Grenzschnittstelle konfigurierten Ausgang Oif weitergeleitet, wird der DRM ohne Verzögerung ratenlimitiert mit einem Trigger über den neuen Empfänger informiert.

TrR **TriggerRpt beim Weiterleiten als DR vom RPT.** Wird ein Paket unter ungültigem NRS-Zustand, also ein Paket, auf das der NRSstate(S,G,Oif) wirkt, vom *Rendezvousstellenbaum* (SPTbit(S,G) = FALSE) an einen direkt angeschlossenen Empfänger oder auf einen als DSMC-Grenzschnittstelle konfigurierten Ausgang Oif weitergeleitet, wird der DRM ohne Verzögerung ratenlimitiert mit einem TriggerRpt über den neuen Empfänger informiert.

TrSw **TriggerSwitched beim Weiterleiten unter gesetztem Sbit.** Wird ein Paket für einen MFC-Eintrag MFCentry(S,G) mit gesetztem Sbit(S,G) (Sbit(S,G) = TRUE) empfangen, wird der DRM ohne Verzögerung ratenlimitiert mit einem TriggerSwitched über die abgeschlossene Umschaltung auf den quellenspezifischen kürzesten Baum informiert.

Tsub **Ersetzte Regeln.** Die Regeln **TrSM** und **TrR** ersetzen Regel **Tr**.

Zustandsregeln 3:

Sbit **MFC-Eintrag erweitert um Sbit.** Jeder MFC-Eintrag MFCentry(S,G) wird um das Switchover-Bit **Sbit(S,G)** erweitert. Ein gesetztes Sbit(S,G) zeigt an, dass gemäß Regel **TrSw** bei Empfang von Datenpaketen ein TriggerSwitched gesendet werden muss.

Initial beim Anlegen eines MFCentry(S,G) und wenn alle Ausgänge inaktiv sind, ist das Sbit(S,G) immer zurückgesetzt (Sbit(S,G) = FALSE). Der von Quelle und Rendezvousstelle aus gesehen erste Router, der sowohl auf dem quellenspezifischen kürzesten als auch auf dem Rendezvousstellenbaum liegt, setzt beim Umschalten auf den quellenspezifischen kürzesten Baum (SPTbit(S,G) ← TRUE durch Update_SPTbit(S,G,iif) in Fall (i) und (ii), wenn Fall (iii) nicht zutrifft) immer auch das Sbit(S,G) (Sbit(S,G) ← TRUE). Beim Empfang eines TriggerSwitchedAck wird das Sbit(S,G) zurückgesetzt (Sbit(S,G) ← FALSE).

Informationsregeln 2:

TSG **TriggerSwitched/-Ack enthält (S,G).** Neben dem Nachrichtentyp selbst enthalten TriggerSwitched und TriggerSwitchedAck das Tupel aus Quelladresse S und Gruppenadresse G, das das Sbit(S,G) eindeutig identifiziert. Die Nennung der Ausgangsschnittstelle Oif entfällt, Regel **TSGO** gilt nicht.

Tabelle 4.4 listet alle für PIM-SM neu eingeführten DSMC-Nachrichtentypen und die in ihnen mindestens enthaltenen Informationen auf.

Typ	(S,G,Oif)	(S,G)	ACKbit	NRSstate	DRbit	Richtung
TriggerRpt	x	-	-	-	-	Router → DRM
TriggerSwitched	-	x	-	-	-	Router → DRM
TriggerSwitchedAck	-	x	-	-	-	DRM → Router

Tabelle 4.4 Mit der Erweiterung für PIM-SM zusätzlich eingeführte DSMC-Nachrichtentypen.

4.3.2 Trigger und TriggerRpt

Die Unterscheidung, ob Trigger oder TriggerRpt gesendet werden muss, entscheidet das SPTbit(S,G). Das SPTbit(S,G) ist Teil des Routingzustands von PIM-SM und kann von der DSMC-Komponente durch Abfrage der PIM-SM-Komponente ermittelt werden. Ist es gesetzt, erfolgt die Weiterleitung anhand des quellenspezifischen kürzesten Baums und die DSMC-Komponente sendet wie beim Basisverfahren einen Trigger zum DRM. Andernfalls sendet es den für PIM-SM neu eingeführten TriggerRpt.

Es wäre möglich, mit nur einem Trigger auszukommen und den TriggerSwitched auch beim direkten Aufbau des SPT zu senden (Fall (iii), s. u.). Dies würde jedoch für den wichtigen Anwendungsfall Source Specific Multicast dazu führen, dass immer zwei DSMC-Nachrichten, Trigger und TriggerSwitched, zum DRM gesendet werden. Des Weiteren wäre keine Interoperabilität mit PIM-SSM-Routern möglich, die die hier beschriebene Erweiterung gar nicht implementieren und daher auch gar keinen TriggerSwitched generieren können.

4.3.3 Sbit

Das Sbit(S,G) ist wie das SPTbit(S,G) nicht schnittstellenspezifisch, da bei PIM-SM ein SPT Switchover immer alle Ausgangsschnittstellen bzw. die über sie erreichten Empfänger betrifft. Egal, ob ein Empfänger bzw. dessen Designated Router Datenpakete vom quellenspezifischen kürzesten Baum angefordert hat oder nicht, erhält er automatisch Datenpakete vom SPT, wenn sie verfügbar werden. Dies ist dann der Fall, wenn ein anderer DR den SPT aufgebaut hat und sich der SPT zu diesem mit dem RPT zu ersterem DR überschneiden. Der TriggerSwitched nennt daher keine Ausgangsschnittstelle und wird entsprechend beim Empfang anstatt wie sonst beim Weiterleiten eines Datenpakets ausgelöst. In der Praxis ist beides identisch, da per Definition das Sbit nur gesetzt ist, wenn auch mindestens eine Ausgangsschnittstelle aktiv ist. Da es jedoch irrelevant ist, auf welchen der Ausgänge weitergeleitet wird, wurde der Empfang zur Definition verwendet.

Das Sbit(S,G) wird von der PIM-SM-Komponente (über die DSMC-Komponente) gleichzeitig mit dem SPTbit(S,G) gesetzt, wenn sich der nächste Router Richtung Quelle von dem Richtung Rendezvousstelle unterscheidet und der SPT nicht direkt aufgebaut wurde. Dies ist genau dann der Fall, wenn in der Funktion Update_SPTbit(S,G,iif) (s. Abschnitt 2.3.2.1) entweder Fall (i) oder Fall (ii) zutrifft (unterschiedlicher Upstreamrouter für RPT und SPT) bei gleichzeitig nicht zutreffendem Fall (iii) (keine aktiven Ausgangsschnittstellen für den Rendezvousstellenbaum). Der Fall (v), also eine direkt angeschlossene Quelle, darf dagegen sehr wohl gleichzeitig auftreten. In Fall (iv) ist der Upstreamrouter von SPT und RPT derselbe und tritt daher nie gleichzeitig mit (i) oder (ii) auf. Dieses Setzen des Sbit(S,G) ist die einzige Modifikation, die an einer PIM-SM-Implementierung im Rahmen von DSMC vorzunehmen ist.

Problemfall unerkannter SPT

Es ist möglich, dass auf dem Verzweigungsrouter während des SPT Switchovers die Menge der Ausgangsschnittstellen, die Verkehr vom Rendezvousstellenbaum erhalten, leer wird. Hier wird also während noch der SPT stromaufwärts aufgebaut wird, der RPT stromabwärts vom Verzweigungsrouter schon wieder abgebaut (durch den Austritt aller Empfänger). Dies führt dazu, dass am Verzweigungsrouter obiger Fall (iii) zutrifft und somit auch kein TriggerSwitched gesendet wird. Dies ist nur dann schädlich, wenn ebenfalls kein Trigger gesendet wird. Dies passiert, wenn der Aufbau des SPTs von einem Router längs eines bereits mit NRS-Zustand konfigurierten Pfads des RPTs aus erfolgt, und dieser Pfad kurz danach abgebaut wird, bevor der Verzweigungsrouter den SPT Switchover mit dem Setzen des SPTbit vollendet[7]. Der Abbau des RPT vor der Vollendung des Aufbaus des SPT wird durch die einfache Abfrage von Fall (iii) also nicht als Umschalten erkannt.

Zwar ließe sich theoretisch auch dieser Fall handhaben, dies ist aus praktischen Gründen jedoch nicht sinnvoll: Der Test hinsichtlich Fall (iii) müsste in dem Moment erfolgen, in dem der Routingzustand anzeigt, dass Multicastverkehr weitergeleitet werden soll und nicht erst beim Setzen des SPTbit bzw. Sbit. Zudem müsste sichergestellt sein, dass in einer Join/Prune-Nachricht empfangene Joins vor den Prunes verarbeitet werden, um ein gleichzeitiges Eintreffen korrekt zu handhaben. Des Weiteren müssten kurz hintereinander erfolgende Änderungen im Routingzustand immer sofort zur Aktualisierung des MFC-Eintrags führen und dürften nicht erst alle verarbeitet werden, um anschließend nur einmal den MFC-Eintrag zu aktualisieren. Dies wäre nötig, um den Fall zu erfassen, dass ein Löschen des MFC-Eintrags nach Ablauf des Override_Intervals für den quellenunspezifischen Routingzustand und ein quasi gleichzeitiger quellenspezifischer Beitritt eines Empfängers zu einem Deaktivieren/Aktivieren-Zyklus der Ausgangsschnittstelle im MFC-Eintrag und damit zum Senden eines Triggers führt. Die insgesamt hierfür nötigen Eingriffe in eine PIM-SM-Implementierung stehen jedoch in keinem sinnvollen Verhältnis zum Nutzen, diesen konstruierten seltenen Fall effizient handhaben zu können.

Denn auch der Verlust von Trigger-Nachrichten kann dazu führen, dass ein aufgebauter SPT unerkannt bleibt: Die Abfolge Trigger von Router A, der verloren geht, TriggerRpt von Router B, dessen RPT den SPT von Router A kreuzt und TriggerStop von Router A führt dazu, dass der von Router A aufgebaute SPT von Router B aufrecht erhaltenen wird, ohne das dies über die empfangenen beiden Nachrichten TriggerRpt und TriggerStop

[7] Dies tritt ebenfalls im pathologischen Fall von sich mehrfach kreuzendem RPT und SPT und dadurch verfrühtem Prune(S,G,rpt) auf [136].

eindeutig erkannt werden kann. Der TriggerStop ohne einen vorherigen Trigger ist zwar ein Hinweis, jedoch könnte auch dieser (nur einmal gesendete) TriggerStop verloren gegangen sein.

Zusammenfassend lassen sich alle Probleme des Nichterkennens eines SPTs auf das Design von PIM-SM zurückführen, das nicht strikt zwischen RPT und SPT trennt, sondern SPT-Verkehr für den RPT nutzt und dadurch einen SPT weiter aufrecht erhält, auch wenn der ihn aufbauende Empfänger längst nicht mehr Mitglied der Gruppe ist. DSMC verzichtet zu Gunsten der Einfachheit auf die Behandlung aller möglichen Fälle und verlässt sich auf die TriggerUC, die in jedem Fall später einen ggf. nicht erkannten SPT anzeigen werden.

4.3.3.1 TriggerSwitched & TriggerSwitchedAck

Ist das Sbit gesetzt ist, wird ein TriggerSwitched solange ratenlimitiert wiederholt, bis der DRM den Empfang mittels TriggerSwitchedAck bestätigt hat und das Sbit zurückgesetzt wird. Hierdurch wird ein möglicher Verlust des TriggerSwitched oder TriggerSwitchedAck behandelt. Anders als bei den Triggern kann die Rate höher gewählt werden. Nimmt man an, dass die Dauer der DRM-seitigen Nachrichtenverarbeitung exklusive der Ressourcenüberprüfung nicht länger dauert als die Router-seitige Nachrichtenverarbeitung, kann ein TriggerSwitched bereits nach T_{RTO} gemäß Gleichung 4.1 wiederholt werden. Anders als ein Conf erfordert ein TriggerSwitchedAck keine vorherige Ressourcenüberprüfung durch den DRM, sondern quittiert lediglich den Empfang des TriggerSwitched (vgl. Nachrichtenverarbeitungsroutine Abschnitt 4.4.2).

Sinne der Einfachheit der Implementierung kann es gerechtfertigt sein, für alle Triggertypen incl. TriggerSwitched dieselbe Rate nach Gleichung 4.4 zu verwenden. Im Falle des Verlusts des TriggerSwitched ist die Dauer des Dienstgüteausfalls nach einem SPT Switchover hierdurch entsprechend länger. Schließlich löst erst der TriggerSwitched die Konfiguration des neuen Multicastpfads von der Quelle bis zum Verzweigungsrouter aus. Solange verbleibt der NRS-Zustand entlang dieses Pfads im Zustand Invalid und alle Empfänger stromabwärts vom Verzweigungsrouter erfahren für diese Zeit keine Dienstgüte. Die Wahrscheinlichkeit für den Verlust von DSMC-Nachrichten sollte allerdings bei Verwendung hochpriorer DSCPs für die Signalisierung gering sein.

Befinden sich mehrere Downstreamrouter an einem Link und schalten auf den SPT um, so sendet jeder von ihnen einen TriggerSwitched. Es ist essentiell für den DRM, an einem solchen Link jeden Downstreamrouter zu kennen, der auf den SPT umgeschaltet hat, denn der SPT stromaufwärts von einem solchen Link wird erst dann abgebaut, wenn der letzte dieser Router den SPT nicht mehr nutzt (angezeigt durch TriggerStop von den DRs stromabwärts dieser Router). Router, die von vornherein den SPT verlangt haben, senden zwar keinen TriggerSwitched, der DRM weiß hier aber bereits über den Trigger von einem ihrer DR stromabwärts, dass auch diese Router den SPT nutzen. Man beachte, dass es beim Zurückumschalten zu einer Unterbrechung in der Weiterleitung von Multicastverkehr kommt, da der Abbau des SPTs gleichzeitig mit dem Aufbau des RPTs beginnt und nicht erst danach. Man beachte zudem, dass bei weiteren Routern am Link, bei denen Fall (ii) zutrifft, und der Assert-Gewinner, seitdem diese Router Zustand für die betreffende Gruppe besitzen und daher Asserts für die Gruppe beachten, seinen Assert noch nicht wiederholt hat, erst der nächste periodische Assert (standardmäßig nach spätestens Assert_Time - Assert_Override_Interval = 177 s) die Router zu Assert-Verlierern macht und das Setzen des SPTbits und Sbits und damit das Senden des TriggerSwitched auslöst. Sollten

bis dahin alle anderen Downstreamrouter am Link ihren Weiterleitungszustand abbauen (alle ihre DRs stromabwärts senden einen TriggerStop), kann der DRM nicht unterscheiden, ob die am Link verbliebenen Router keinen TriggerSwitched gesendet haben, weil sie den RPT nutzen wollen oder weil sie noch keinen Assert empfangen haben. Der DRM muss in diesem Fall den Weiterleitungszustand auf dem Upstreamrouter RPF'(S,G) auf dem SPT per Conf(Ack,DR) überprüfen und nur bei Empfang eines Fail die Ressourcen auf dem SPT stromaufwärts freigegeben und stattdessen die Ressourcen auf dem RPT belegen und die dortigen Router mit NRS-Zustand konfigurieren. Das gesetzte DRbit erfasst den Fall, dass die Überprüfung durch das Conf zu früh erfolgt. Dann führt kurze Zeit später das TriggerStop vom RPF'(S,G) zur Umbelegung der Ressourcen. Geht der TriggerStop verloren, wird dies durch einen TriggerUC von einem Router auf dem RPT erkannt. Da bei einer Zurückumschaltung vom SPT auf den RPT kein Trigger den abgeschlossenen Aufbau des RPTs anzeigt, muss der DRM stromaufwärts eines solchen Links mit drei und mehr Routern wie bei einer Routingänderung vorgehen (s. Abschnitt 4.4.5) und die Konfiguration des RPTs ggf. einige Male wiederholen.

Die Kombination aus einem oder mehreren TriggerRpt von den DRs und einem TriggerSwitched vom Verzweigungsrouter definiert jenen Teilbaum mit dem Verzweigungsrouter als Wurzel, der den Verkehr über den SPT erhält. Solange noch ein Empfänger stromabwärts vom Verzweigungsrouter vorhanden ist – es kann auch einer nach dem TriggerSwitched neu hinzugekommener sein –, wird der SPT stromaufwärts vom Verzweigungsrouter nicht abgebaut, irrelevant, ob der Empfänger bzw. sein DR den SPT oder den RPT angefordert hat. Es ist für den DRM unerheblich zu wissen, welche Zweige stromabwärts vom Verzweigungsrouter SPT und welche RPT sind, da hier beide Bäume übereinstimmen. Das Vorgehen des DRMs, wenn dies nach einer (Unicast-)Routingänderung nicht mehr zutreffen sollte, beschreibt Abschnitt 4.4.6.

Ein Abbau des SPTs und Zurückumschalten zum RPT findet ausschließlich dann statt, wenn keiner der Empfänger explizit den SPT wünscht (immediate_olist(S,G) = NULL) und die Quelle längere Zeit nicht mehr gesendet hat, so dass der KeepaliveTimer(S,G) abgelaufen ist. In diesem Fall wird jedoch der ohne aktive Quelle ungenutzte MFC-Eintrag ebenfalls gelöscht und entsprechend ein TriggerStop gesendet, das Zurückumschalten findet also nur im Routingzustand und nicht im Weiterleitungszustand statt, der einfach gelöscht wird. DSMC benötigt daher keinen „TriggerSwitchedStop“ äquivalent dem TriggerStop.

Rendezvousstellen

Eine Rendezvousstelle verhält sich bzgl. TriggerSwitched wie jeder andere (Verzweigungs-) Router auch. Initiiert sie den SPT Switchover, um den Aufwand für das Tunneln in Register-Nachrichten zu vermeiden, erzeugt sie mit dem ersten über den SPT nativ empfangenen Multicastpaket ein TriggerSwitched an den DRM, da Fall (i) erfüllt ist. RPF_interface(RP(G)) ist nach PIM-SM-Spezifikation bei der Rendezvousstelle der Register-Tunnel, der als Pseudoschnittstelle aufgefasst und wie jede andere Schnittstelle behandelt wird. Da beim RP immer Fall (i) auftritt, sendet er anders als ein DR auch immer einen TriggerSwitched. Auch wenn der RP gleichzeitig der DR der Quelle ist (Fall (v)) und der Register-Tunnel nie verwendet wird, trifft auch hier Fall (i) zu und der RP sendet immer ein TriggerSwitched, obwohl dieser eigentlich unnötig ist. Es bleibt einer Implementierung überlassen, ob sie diesen Fall optimieren und den TriggerSwitched einsparen möchte oder nicht.

Bei Einsatz von Anycast-RP (s. Abschnitt 2.3.2.1) müssen alle RPs, die ein Register von einem anderen RP ihres Anycast-RP-Sets erhalten haben, zwingend den SPT Switchover sofort durchführen (SwitchToSptDesired(S,G) ist immer wahr), da sie nicht beeinflussen können, wie lange der DR der Quelle noch Register sendet und diese an sie weitergeleitet werden. Hier wird der TriggerSwitched also immer zusammen mit dem TriggerRpt des ersten DRs auftreten und ist daher prinzipiell überflüssig. Dieser Fall wird bei DSMC aber ebenfalls nicht separat berücksichtigt, weil zum Zeitpunkt des Setzens des SPTbit(S,G) nicht mehr bekannt ist, woher das Register stammte. Hierzu wäre also neben dem Setzen des Sbits eine weitere Modifikation an einer PIM-SM-Implementierung nötig, der durch den geringen Nutzen aber nicht gerechtfertigt erscheint. Vielmehr wird bei Anycast-RP auch der RP, der die Register vom DR erhält, sofort ein Register-Stop senden, da ein fortbestehendes Kopieren der Register-Nachrichten an die übrigen RPs aus dem beschrieben Grund ohnehin nutzlos ist. Der Register-Tunnel dient bei Anycast-RP ausschließlich zur Bekanntgabe von Quellen und nicht wie sonst auch zur dauerhaften Multicastweiterleitung. Die im Beispiel unten erläuterten Varianten für die Konfiguration von NRS-Zustand für den Register-Tunnel betreffen Anycast-RP entsprechend nicht. Schaltet grundsätzlich jeder RP sofort auf den SPT um und ist dieses Verhalten dem DRM bekannt, so dass dieser auch ohne ein TriggerSwitched den SPT von Quelle zu jedem RP annimmt, kann das Setzen des Sbit und Senden der TriggerSwitched durch RPs optional entfallen. Allerdings kann es aus Gründen der Robustheit von Vorteil sein, auch dann weiterhin TriggerSwitched zu nutzen, da es sich bei Anycast-RP um statische händische Konfiguration handelt, die leicht inkonsistent werden kann.

Für MSDP (s. Abschnitt 2.3.2.1) gilt prinzipiell dasselbe wie für Anycast-RP. Der SPT Switchover erfolgt sofort, da auch MSDP nur der Bekanntmachung neuer Quellen und nicht der Multicastweiterleitung von Paketen dieser Quellen dient (zwar kann auch bei MSDP ein Paket in der Source-Active-Nachricht getunnelt werden, jedoch ist dies wie bei Anycast-RP nicht vom empfangenden RP steuerbar). Der TriggerSwitched wird wie erläutert von einem RP immer gesendet, also auch bei MSDP.

4.3.3.2 Sofortiger SPT Switchover

Manche Netzbetreiber konfigurieren ihre PIM-SM-Router so, dass sie immer sofort mit dem ersten eintreffenden Datenpaket einer Quelle den SPT Switchover initiieren. Die Policyfunktion SwitchToSptDesired(S,G) ist dieser Konfiguration immer wahr. Rendezvousstellenbaum und der Register-Tunnel dienen nur noch zum Auffinden von Quellen und nicht mehr einer dauerhaften Multicastweiterleitung. Liegt eine solche Konfiguration des Netzes vor, kann PIM-SM allein mit dem Basisverfahren von DSMC effizient unterstützt werden und die Erweiterung von DSMC für PIM-SM wird nicht benötigt. Der DRM muss lediglich sicherstellen, dass er die Konfiguration des NRS-Zustands frühestens $2 \cdot \text{MaxSignalingDelay}$ nach Eintreffen des Triggers vom Designated Router eines neuen Empfängers beginnt.

Zur Erläuterung sei der in Abbildung 4.13 dargestellte Signalisierungsablauf eines sofortigen SPT Switchovers betrachtet. Der Designated Router des neuen Empfängers DR_idxR baut über Router Rt den RPTs bis zur Rendezvousstelle RP auf (Schritt 1a. und 1b.). Die Rendezvousstelle hat den Register-Tunnel vom Designated Router der Quelle DR_S sofort per Register-Stop beendet (nicht dargestellt), da sie ja ebenfalls mit dem ersten Empfänger sofort den SPT etablieren will. Sie sendet daher zum Reaktivieren des Empfangs der Quelle

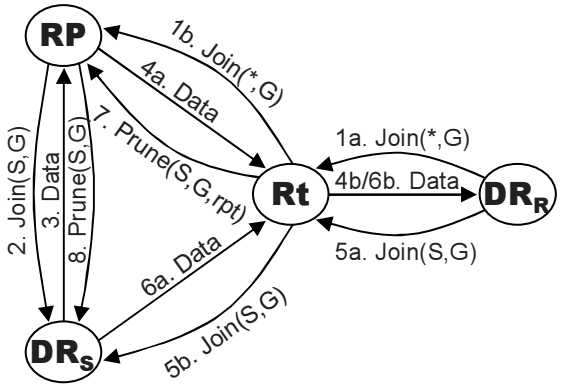


Abbildung 4.13 Bei sofortigem SPT Switchover genügt für PIM-SM das Basisverfahren von DSMC mit T_S nach Gleichung 4.8, so dass DR_S den TriggerUC vom Fließen der ersten Daten in Schritt 3. bis zum Eintreffen des Prune in Schritt 8. unterdrückt.

sofort ein Join zu DR_S (Schritt 2.). Ab hier fließen die ersten Daten der Quelle über RP, Rt bis zu DR_R (Schritte 3., 4a., 4b.). DR_R initiiert daraufhin sofort den Aufbau des SPTs (Schritt 5a.), der über bei Rt vom RPT beginnt zu divergieren. Das von Rt weitergesendete Join etabliert also ab hier einen neuen Pfad zu DR_S (Schritt 5b.). Mit Eintreffen der ersten Daten über diesen neuen Pfad (Schritt 6a.) sorgt Rt als Verzweigungsrouter von RPT und SPT für das Prune der Quelle vom RPT (Schritt 7.). RP hat keine Empfänger mehr und pruned daraufhin den gerade erst aufgebauten SPT wieder (Schritt 8.).

Da dem DRM bekannt ist, dass es zum sofortigen SPT Switchover mit diesem Signalisierungsablauf kommt, kann er die unnötige Konfiguration des nur kurzzeitig genutzten RPTs unterlassen. Der DRM führt dazu nicht sofort mit Empfang einer Triggers von DR_R in Schritt 4b. eine Konfiguration des RPTs durch, sondern wartet ab, bis der SPT mit Schritt 6a. etabliert ist und konfiguriert nur diesen. Da diese Zeit mit $2 \cdot \text{MaxSignalingDelay}$ nach oben hin abgeschätzt werden kann, muss der DRM lediglich sicherstellen, dass er seine Conf-Nachrichten nicht früher als diese Zeit nach Eintreffen der Trigger-Nachricht von DR_R sendet.

Währenddessen eintreffende TriggerUC von DR_S kann der DRM aufgrund seines Wissens vom bevorstehenden SPT Switchover ignorieren. Oder sie werden durch eine geeignete Wahl der initialen Verzögerung T_S von vornherein vermieden:

$$\begin{aligned} T_S &= \max(6 \cdot \text{MaxSignalingDelay}, \\ &\qquad \text{MaxDrmProcessingTime} + \text{MaxDrmRtt} + \text{MaxSignalingDelay}) \end{aligned} \quad (4.8)$$

Der Faktor 6 überbrückt die dargestellten 6 Signalisierungsschritte vom Fließen der ersten Daten am Designated Router der Quelle DR_S in Schritt 3. bis zum dortigen Eintreffen des Prune in Schritt 8. Jeder Schritt kann nach oben hin mit MaxSignalingDelay abgeschätzt werden.

4.3.4 Beispiel

Zur Erläuterung der Erweiterung von DSMC für PIM-SM soll wie bei den Beispielen zum Basisverfahren die Topologie aus Abbildung 4.9 herangezogen werden. Wie üblich ist Host

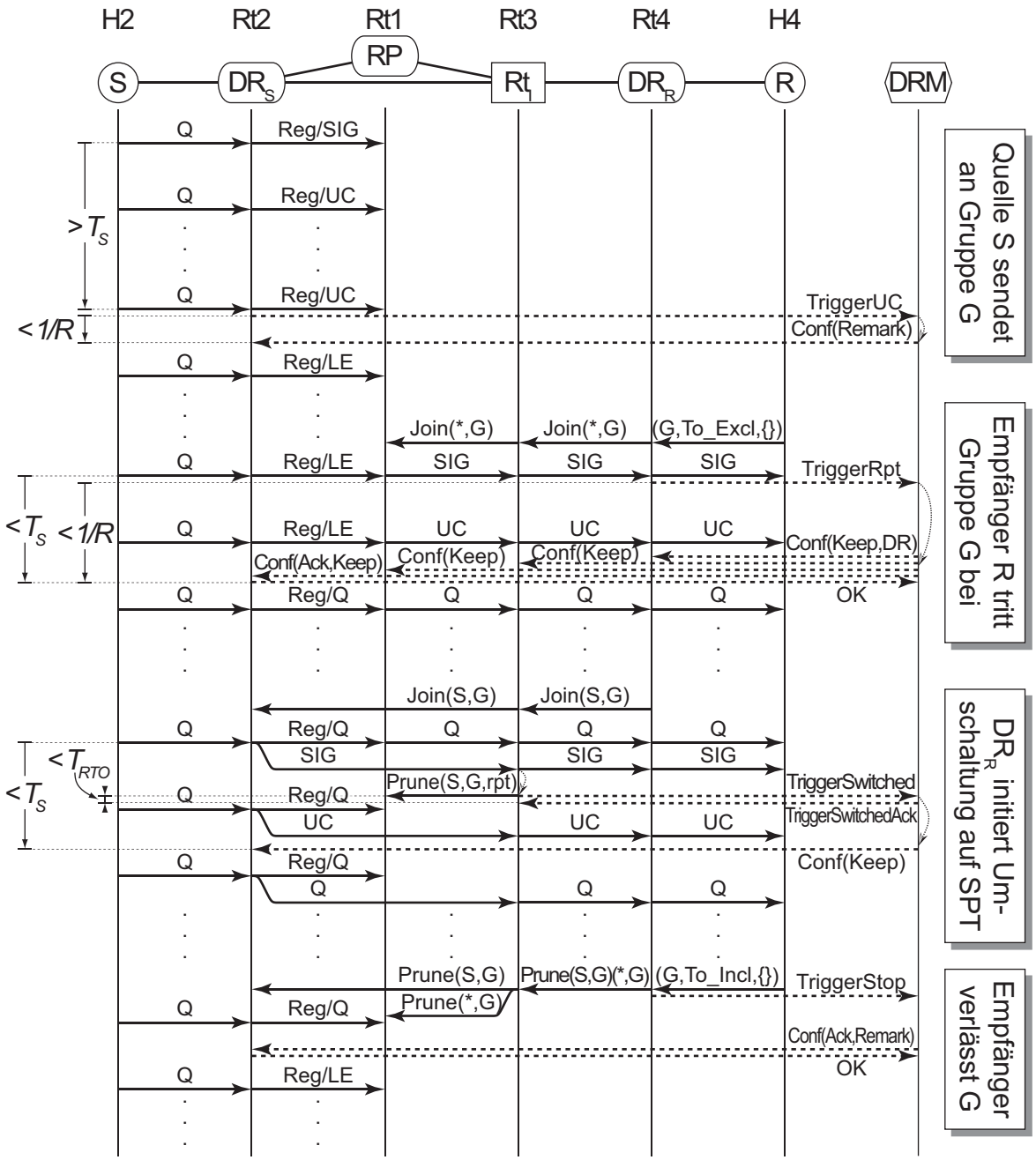


Abbildung 4.14 PIM Sparse Mode erfordert drei weitere DSMC-Nachrichtentypen TriggerRpt, TriggerSwitched und TriggerSwitchedAck.

H2 die Quelle und Host H4 der Empfänger, Router Rt2 und Rt3 sind die Designated Router DR_S und DR_R von Quelle bzw. Empfänger und Rt1 die für die betrachtete Gruppe zuständige Rendezvousstelle RP. Rt3 ist der Router Rt_I im Inneren der Domäne, an dem quellenspezifischer kürzester und Rendezvousstellenbaum divergieren. Der Ablauf der Signalisierung ist im Weg-Zeit-Diagramm Abbildung 4.14 wiedergegeben.

Die Quelle S beginnt mit dem Senden an die Multicastgruppe G. DR_S legt einen MFC-Eintrag für (S,G) an und tunnelt die Pakete in Register-Nachrichten gekapselt per Unicast an die für G zuständige Rendezvousstelle RP. Da noch kein NRS-Zustand für die Register-Tunnelschnittstelle konfiguriert wurde, ist dieser ungültig und die Datenpakete werden

erst auf DSCP_SIG, dann auf DSCP_UC ummarkiert. Die Register-Nachrichten tragen denselben DSCP wie die in ihnen getunnelten Multicastpakete (s. Abschnitt 2.4.1), der beim Entkapseln vom RP in das Multicastpaket zurückgeschrieben wird, was hier keine Auswirkungen hat, da die DSCPs übereinstimmen.

Die Pakete werden von RP nicht weitergeleitet, da es noch keine Empfänger für G gibt. Im Beispiel besage die lokale Policy von RP, dass keine sofortige Umschaltung auf den SPT initiiert wird. RP sendet demnach kein RegisterStop(S,G) zurück an DR_S und die Register-Nachrichten fließen weiterhin von DR_S zu RP. Nach Ablauf der Unterdrückungszeit T_S informiert DR_S den DRM über seinen ungültigen NRS-Zustand. Wie schon bei BIDIR-PIM erläutert, gibt es prinzipiell drei Möglichkeiten, wie der Ressourcenmanager auf den TriggerUC reagieren kann: Conf(Keep), Conf(Remark) oder nichts unternehmen.

Mit Conf(Keep) wird der NRS-Zustand endgültig konfiguriert und ein weiteres Senden der TriggerUC unterbunden. Wie schon bei BIDIR-PIM ausgeführt wurde, ist hieran nachteilig, dass die Pakete unnötig Dienstgüte erfahren, obwohl es noch keine Empfänger gibt, und dadurch unnötig Ressourcen blockieren. Wird erwartet, dass dieser empfängerlose Zustand länger erhalten bleibt und der RP in nächster Zeit keinen SPT Switchover, kann stattdessen mittels Conf(Remark) der NRS-Zustand konfiguriert und ebenfalls weitere TriggerUC unterbunden werden. Im betrachteten Beispiel ist diese Variante dargestellt und Pakete werden von DR_S nach Erhalt des Conf(Remark) auf DSCP_LE ummarkiert, wobei die Ratenlimitierung der TriggerUC nach Gleichung 4.4 dem DRM zur Konfiguration genug Zeit gelassen hat. Man beachte, dass ein Conf(Keep) bzw. Conf(Remark) an den Designated Router der Quelle genügt, während bei BIDIR-PIM ein Conf je Router auf dem Pfad zum Rendezvousstellenlink nötig ist. Denn Register-Nachrichten werden per Unicast zum RP gesendet und werden als solche von DSMC nicht beachtet.

Mit DSCP_UC markierte Pakete erfahren nach Regel **DSCP** dieselbe Dienstgüte wie ummarkierte Pakete nach Regel **Rmrk**. Daher kann als dritte Alternative der DRM die TriggerUC-Nachrichten ignorieren und nichts unternehmen und weiterhin ratenlimitiert eintreffende TriggerUC-Nachrichten in Kauf nehmen, die sich als Duplikate leicht erkennen lassen (s. Abschnitt 4.4.2) und keine weitere Verarbeitung im DRM auslösen, nachdem einmal entschieden wurde, sie zu ignorieren. Diese Variante kann sinnvoll sein, wenn erwartet wird, dass die Rendezvousstelle demnächst auf den SPT umschalten wird und somit der NRS-Zustand für die Register-Tunnelschnittstelle ohnehin gleich wieder gelöscht werden würde.

Nun tritt Empfänger R durch Senden des Änderungsberichts (G,To_Excl,{}) der Gruppe G bei. Da es sich nicht um einen quellenspezifischen Gruppenbeitritt wie im Beispiel zu PIM-SSM Abschnitt 4.2.11.1 handelt, sondern R anzeigt, dass er jede Quelle empfangen möchte, wird von DR_R ein Join(*,G) Richtung der für G zuständigen Rendezvousstelle RP gesendet. RP entkapselt daraufhin die über den Register-Tunnel eintreffenden Pakete, markiert wegen ungültigen NRS-Zustands das erste auf DSCP_SIG, alle folgenden auf DSCP_UC um und leitet sie an Rt_I weiter, wobei RP den TriggerUC vorerst unterdrückt. Rt_I hat auch keinen gültigen NRS-Zustand, auch er markiert das erste Paket mit DSCP_SIG und wechselt dabei den NRS-Zustand von Init nach Invalid. Nachfolgende Pakete tragen bereits DSCP_UC und dürfen nach Regel **UC** nicht ummarkiert werden. Entsprechend wird Rt_I nach Regel **TrUC** auch dann kein TriggerUC generieren, wenn die Unterdrückungszeit bei ihm abläuft. DR_R hat einen direkt angeschlossenen Empfänger R und ist als letzter Router des neuen Multicastpfads dafür verantwortlich, den DRM zu informieren. Da

die Pakete vom geteilten Rendezvousstellenbaum eintreffen, informiert DR_R den DRM nach Regel **TrR** sofort mit einem TriggerRpt, nachdem er das DRbit(S,G,Oif) für die Ausgangsschnittstelle Oif Richtung R gesetzt hat. Der DRM kennt die für die Gruppe zuständige Rendezvousstelle, kann damit den neuen Multicastpfad ableiten und konfiguriert alle Router ausgenommen DR_S bei genügend Ressourcen wie im Beispiel mittels Conf(Keep), sonst mittels Conf(Remark). Der NRS-Zustand von DR_S muss mit einem Conf(Keep,Ack) umkonfiguriert werden, um ein ConfReply anzufordern und so Paketverlust erkennen zu können, da DR_S bereits konfigurierten NRS-Zustand besitzt und bei Nichtempfang der Conf-Nachricht keinen Trigger bzw. TriggerUC mehr senden würde wie die übrigen Router. Die Unterdrückung des TriggerUC durch RP und die Rate des TriggerRpt von DR_R müssen wie üblich Gleichung 4.4 bzw. Gleichung 4.5 erfüllen, um unnötige Triggernachrichten zu vermeiden. Die Daten der Quelle fließen ab jetzt ohne ummarkiert zu werden über den Register-Tunnel DR_S–RP zur Rendezvousstelle und weiter über den Rendezvousstellenbaum RP–Rt_I–DR_R bis zum Empfänger, die Dienstgüte wird erbracht.

Nach einer gewissen Zeit entscheidet sich DR_S für die Umschaltung auf den quellenspezifischen kürzesten Baum (die Policyfunktion SwitchToSptDesired(S,G) wird wahr). Er sendet ein quellenspezifischen Join(S,G) Richtung Quelle, also zu Rt_I, der daraufhin seinerseits ein Join(S,G) zu seinem Nachbarrouter DR_S Richtung Quelle sendet. DR_S aktiviert die Schnittstelle Richtung Rt_I in seinem MFC-Eintrag für (S,G). Der NRS-Zustand ist anfänglich ungültig, so dass er das erste Paket beim Weiterleiten über die neue Ausgangsschnittstelle nach DSCP_SIG, nachfolgende nach DSCP_UC ummarkiert, der TriggerUC wird vorerst unterdrückt. Das erste Paket, das Rt_I direkt von DR_S empfängt, sorgt für das Setzen des SPTbit(S,G). Da sich die Eingangsschnittstelle von der Richtung RP unterscheidet (Fall (i) in Update_SPTbit(S,G,iif)), ist Rt_I der erste Router auf beiden Multicastbäumen und setzt daher nach Regel **Sbit** auch das Sbit(S,G). Das Empfangen (und Weiterleiten) des ersten Datenpakets löst gemäß Regel **TrSw** einen TriggerSwitched aus und zeigt dem DRM an, dass der neue quellenspezifische Multicastbaum vollständig etabliert ist und der NRS-Zustand entlang des neuen Pfads konfiguriert werden kann, hier nur die neu aktivierte Ausgangsschnittstelle bei DR_S. Die Änderung der Eingangsschnittstelle im MFC-Eintrag bei Rt_I verändert bzw. invalidiert den NRS-Zustand dort nicht, da NRS-Zustand an die Ausgangsschnittstellen im MFC-Eintrag gebunden ist, die Ausgangsschnittstelle hat sich bei Rt_I aber nicht verändert. Die Limitierung der Rate der TriggerSwitched auf $< 1/T_{RTO}$ vermeidet unnötige weitere TriggerSwitched, bevor der DRM den Empfang mittels TriggerSwitchedAck bestätigen kann. Der TriggerSwitchedAck löscht das Sbit(S,G) und unterbindet so weitere TriggerSwitched. Nach einer kurzen Unterbrechung von $< 1/R$, in der mit DSCP_UC markierte Datenpakete R erreichen, gelangen wieder mit dem ursprünglichen DSCP „Q“ markierte Pakete zum Empfänger, jetzt über den neuen quellenspezifischen kürzesten Pfad S–DR_S–Rt_I–DR_R–R. Der Empfang des ersten Pakets über der SPT führt dazu, dass Rt_I mittels Prune(S,G,rpt) den Empfang von Datenpaketen über den Rendezvousstellenbaum stoppt, um keine Duplikate zu erhalten. Der Ausgang von RP Richtung Rt_I wird dadurch inaktiv und der zugehörige NRS-Zustand von RP ungültig. RP sendet keinen TriggerStop, da das DRbit(S,G) beim ihm nicht gesetzt ist. Dem DRM ist dieses Verhalten bekannt und kann mit dem Empfang des TriggerSwitched die Ressourcen auf diesem Pfadabschnitt RP–Rt_I bei sich erneut als frei markieren.

Abschließend verlässt R die Gruppe wieder mittels Änderungsbericht (G,To_Incl,{}). DR_R löscht seinen Weiterleitungszustand, was wegen des gesetzten DRbits einen TriggerStop auslöst, und entfernt sich selbst mittels Prune(S,G) vom SPT und mittels Prune(*,G)

vom RPT. Dasselbe tut daraufhin auch Rt_I, bei dem sich das Ziel der beiden Prune-Nachrichten jedoch unterscheidet, denn bei ihm als ersten Router sowohl auf dem SPT wie dem RPT divergieren beide Bäume. Das Prune(S,G) erreicht DR_S, der seinen MFC-Eintrag daraufhin löscht. Das Prune(*,G) sorgt bei der Rendezvousstelle dagegen nur für eine Änderung im Routingzustand. Der einzige Weiterleitungszustand für diese Gruppe war der für die Quelle S und wurde bereits durch das Prune(S,G,rpt) entfernt. Ohne Empfänger fließen nun die Datenpakete von S durch den Register-Tunnel zu RP, ohne dass es Empfänger für die Daten gibt. Um unnötigen Ressourcenverbrauch zu vermeiden, kann mittels Conf(Ack,Remark) DR_S erneut dazu veranlasst werden, auf DSCP_LE umzumarkieren, bis wieder ein Empfänger der Gruppe beitritt. Das OK von DR_S bestätigt dem DRM, dass DR_S sein Conf(Ack,Remark) erfolgreich empfangen und verarbeitet hat. Andernfalls würde DRM sein Conf(Ack,Remark) bis zum Empfang eines OK mit der Rate $1/T_{RTO}$ wiederholen oder beim Empfang eines Fail erkennen, dass die Quelle nicht mehr sendet und der Register-Tunnel mit zugehörigem NRS-Zustand abgebaut wurde. Wie bereits bei BIDIR-PIM erläutert, kann statt Conf(Ack,Remark) ein Conf(Ack,Remark,DR) gesendet werden, um explizit von DR_S mit einem TriggerStop informiert zu werden, wenn die Quelle das Senden eingestellt hat.

4.3.5 Designalternativen

4.3.5.1 Trigger vom Designated Router beim SPT Switchover

Eine naheliegende Variante ist es, jeden Designated Router beim Umschalten auf den SPT einen Trigger senden zu lassen, so dass der DRM zu jeder Zeit exakt weiß, zu welchem DR der SPT und welchem den RPT zu konfigurieren ist. Hierdurch lässt sich allerdings nicht ohne Weiteres der TriggerSwitched ersetzen, da dieser den vollständig abgeschlossenen Aufbau von Weiterleitungszustand auf dem neuen Pfad zwischen Quelle und Verzweigungsrouter anzeigt, die Trigger dagegen bis auf die Fälle (i) und (ii) in Update_SPTbit zu früh kommen. Erst wenn für den Trigger bei den anderen Fällen zusätzlich gefordert wird, dass Pakete mit DSCP_UC oder DSCP_SIG markiert weitergeleitet werden, ist der SPT vollständig eingerichtet, denn offensichtlich haben die Pakete den noch unkonfigurierten neuen Pfadabschnitt des SPTs passiert und sind dort ummarkiert worden. Dieser Test schlägt allerdings fehl, wenn SPT und RPT vollständig identisch sind. Es muss daher spätestens nach einer definierten maximalen Verzögerung mit dem ratenlimitierten Senden des Triggers begonnen werden.

Nachteilig an dieser Lösung ist, dass sich die Anzahl an vom DRM zu bearbeitenden Triggern fast verdoppelt von einem TriggerRpt je DR und einem TriggerSwitched vom Verzweigungsrouter auf einen TriggerRpt und einen Trigger je DR (wenn man annimmt, das alle DR ähnlich konfiguriert sind und ab einer gewissen Senderate der Quelle alle den SPT Switchover initiieren werden). Zudem wird diese Information erst dann für den DRM relevant, wenn durch eine spätere Routingänderung SPT und RPT stromabwärts vom Verzweigungsrouter auch tatsächlich nicht mehr identisch sein sollten (dies muss nicht nach jeder Routingänderung zutreffen). Hier lässt sich jedoch über das Netzwerkmanagement mit demselben Aufwand (je eine Abfrage je DR) der aktuelle Wert des SPTbits auf jedem DR ermitteln, bietet also auch in diesem Fall keinen Vorteil. Es ist daher unnötig und in DSMC nicht umgesetzt, jeden DR den DRM seinen SPT Switchover anzeigen zu lassen, sondern es genügt, dass dies der Verzweigungsrouter durch seinen TriggerSwitched für alle betroffenen DR zusammen tut und sich der DRM nach einer Routingänderung die fehlende Information je nach Bedarf nachträglich beschafft.

4.3.5.2 TriggerSwitched und TriggerSwitchedAck mittels SNMP

Die Funktion Update_SPTbit(S,G,iif) wird nur konzeptionell beim Empfang von Datenpaketen aufgerufen. Reale Implementierungen werden sie immer dann aufrufen, wenn sich eine der in ihr enthaltenen Bedingungen geändert hat. Allerdings tritt Fall (i) direkt beim Empfang bzw. Weiterleiten eines Datenpakets, Fall (ii) indirekt beim Empfang eines von einem Datenpaket ausgelösten Asserts auf. Entsprechend wird das Setzen des Sbits erst durch das Weiterleiten von Datenpaketen ausgelöst, das Setzen der SPTbit in Fall (i) und (ii) impliziert damit den vollständigen Abschluss der Einrichtung von Weiterleitungszustand stromaufwärts des Verzweigungsrouters. Eine Implementierung kann daher alternativ unabhängig vom tatsächlichen Empfang von Datenpaketen das Senden der TriggerSwitched mit dem Setzen des Sbits beginnen und die ratenlimitierte Wiederholung timergesteuert durchführen. Die Wiederholung endet anders als bei anderen Triggern nicht durch die Konfiguration von NRS-Zustand, sondern durch eine reine Empfangsbestätigung mittels TriggerSwitchedAck. Sollte der Weiterleitungszustand vor Eintreffen der Bestätigung wieder abgebaut worden sein, zieht ein TriggerSwitchedAck anders als ein Conf keinen weiteren Aufwand in Form eines vom DRM zu bearbeitenden ConfReply (Fail) nach sich, sondern wird vom Router schlicht ignoriert. Es ist daher unschädlich, das Generieren der Trigger unabhängig von der Weiterleitung von Paketen zu gestalten.

Allerdings bedeutet das timergesteuerte Wiederholen Zustand in der ansonsten zustandslosen DSMC-Komponente. Je nach Fähigkeit der Netzwerkmanagementkomponente eines Routers, SNMPv2-Trap-PDUs oder InformRequest-PDUs bei Auftreten gewisser Werte in der MIB ratenlimitiert zu wiederholen, kann dieser Zustand jedoch dorthin verlagert werden: Der TriggerSwitched wird durch eine SNMPv2-Trap-PDU oder InformRequest-PDU ersetzt, die das gesetzte Sbit dem DRM anzeigt, und der TriggerSwitchedAck wird durch eine SetRequest-PDU ersetzt, die das Sbit zurücksetzt. Die anschließende Bestätigung durch eine Response-PDU ist zwar überflüssig aber unschädlich. Eine derart implementierte Erweiterung von DSMC für PIM-SM kommt mit nur einem zusätzlichen DSMC-Nachrichtentyp TriggerRpt aus, den die DSMC-Komponente statt eines Triggers verwendet, wenn das SPTbit gesetzt ist. Die PIM-SM-Komponente wird wie beschrieben um die Fähigkeit zum Setzen des Sbits durch Update_SPTbit(S,G,iif) in den Fällen (i) und (ii) und nicht zutreffendem Fall (iii) ergänzt, das Sbit befindet sich jedoch nicht mehr in der Weiterleitungsebene im MFC, sondern in der MIB in der Routingebene.

4.4 Ressourcenmanager

DSMC stellt ein Verfahren dar, mit dem ein zentraler Ressourcenmanager das NRS-Problem lösen und Dienstgüte für Multicast bereitstellen kann. Im Folgenden soll erläutert werden, wie das in den bisherigen Abschnitten beschriebene Verfahren DSMC von einem DRM genutzt werden kann und welche Voraussetzung er dabei erfüllen muss. Insbesondere wird skizziert, wie die Verarbeitung von DSMC-Nachrichten im DRM auf einfache Weise erfolgen kann.

4.4.1 Voraussetzungen

Voraussetzung für ein erfolgreiches Ressourcenmanagement ist, dass der DRM in seiner Domäne alle Router mit all ihren Adressen und konfigurierten Subnetzadresspräfixen, die komplette Topologie, also die Vernetzung der Router untereinander, sowie alle für

die Bereitstellung von Dienstgüte wesentlichen physikalischen Parameter der Router und der Links zwischen ihnen exakt kennt. Hier ändert sich gegenüber dem Unicastdienstgütemanagement nichts. Die Werte sind über das Netzwerkmanagement (SNMP oder CLI) ermittelbar. Durch die Teilnahme am Link-State-Routing kann der DRM darüber hinaus einfacher die Metriken bestimmen und von Linkausfällen erfahren als über das Netzwerkmanagement mit SNMP Traps. Bei Distance-Vector-Routing ist er dagegen auf SNMP und Traps bzw. besser auf bestätigte InformRequests angewiesen, um Ausfälle schnell erkennen zu können.

Multicast & DSMC

Im Vergleich zum Unicastdienstgütemanagement muss der DRM zusätzlich folgende den Multicastpfad beeinflussende Daten ermitteln:

- Unterstützte und aktivierte Multicastroutingprotokolle
- Zuordnung der Gruppenadressen zu Multicastroutingprotokollen
- Zuordnung von Gruppen- zu RP-Adressen (nur rendezvousstellenbasierte Protokolle PIM-SM und BIDIR-PIM)
- DR-Priorität je Schnittstelle (nur PIM-SM)

Alle genannten Punkte sind prinzipiell routerspezifisch. Da unterschiedliche Protokolle an einem Link nicht interoperabel sind, ist an einem Link ein Protokoll jedoch immer auf allen Routern aktiviert oder auf keinem der Router. Auch die Zuordnungen der Gruppenadressen zu den Multicastroutingprotokollen und ggf. den RP-Adressen dürfen sich zwischen den Routern einer Routingdomäne nicht unterscheiden, soll Multicastrouting für diese Gruppenadressen funktionieren. Für PIM-SM und BIDIR-PIM sind beide Zuordnungen über die Teilnahme am BSR-Mechanismus ermittelbar. Die DR-Priorität bestimmt bei PIM-SM quellenseitig den DR, der Register-Nachrichten sendet, und empfängerseitig den DR, der Joins für lokale Gruppenmitglieder sendet, solange es zu keinem Assert kommt. Die DR-Priorität ist problemlos über das Netzwerkmanagement bestimmbar.

Soll die Bereitstellung von Dienstgüte mit dem Verfahren DSMC erfolgen, sind zusätzlich noch die

- Zuordnung von Ausgangsschnittstelle zu Link
- Unterstützung von (*,G)- und/oder (S,G)-Weiterleitungszustand
- Links mit möglichen Empfängern

zu ermitteln. Um die Signalisierung von DSMC effizient zu gestalten, werden Ausgangsschnittstellen in DSMC-Nachrichten nicht durch IP-Adressen sondern durch den Schnittstellenindex identifiziert, wie es auch in SNMP üblich ist. Welchen Index eine Schnittstelle hat und welcher Link bzw. welches Subnetz über sie erreicht wird, ist entsprechend bereits vom Netzwerkmanagement her bekannt. Die Art des vom Router unterstützten Weiterleitungszustands ist implementierungsspezifisch und meist nicht über das Netzwerkmanagement ermittelbar. Dies kann jedoch leicht beim ersten Konfigurationsversuch eines Routers mit

NRS-Zustand festgestellt werden, wenn dieser statt des erwarteten MFCentry(S,G) einen MFCentry(*,G) angelegt hat (oder umgekehrt) und den nicht existierenden MFCentry(S,G) (oder MFCentry(*,G)) per ConfReply mit zurückgesetztem ACKbit (Fail) anzeigt. Legt ein BIDIR-PIM-Router auf Pfaden ohne Gruppenmitglieder überhaupt keinen MFC-Eintrag an, kann dies auf dieselbe Weise festgestellt werden. Sind Links bekannt, an denen sich Empfänger befinden können, reduziert dies den Aufwand der nach einem TriggerStop oder einer Routingänderung stattfindenden Überprüfung der Links auf nachträglich hinzugekommene Empfänger (s. Abschnitt 4.4.4.3). Lässt sich beispielsweise auf allen Routern eines Links unabhängig vom Multicastroutingprotokoll IGMP/MLD deaktivieren, können auf einem solchen Link Empfänger sicher ausgeschlossen werden.

Ressourcengraph

Der Ressourcengraph ist die zentrale Datenstruktur des Ressourcenmanagers. In ihm vermerkt der DRM für alle Router und Links die Reservierungen, die über diese laufen. Links werden wie Router als Knoten im Ressourcengraphen dargestellt. Der Graph gleicht damit dem vom Link-State-Routing aufgebauten Routinggraphen, speichert jedoch deutlich mehr Routinginformation. So sind im Ressourcengraphen nicht nur alle Adressen vermerkt, sondern an jedem Knoten sind alle Routen verfügbar, so dass ausgehend von diesem anhand eines gegebenen Zielknotens bzw. dessen Adresse der nächste Hop und damit Hop by Hop der Pfad bestimmt werden kann.

Bei Erweiterung des Ressourcengraphen für Multicast ändert sich hier nichts Prinzipielles. Bezüglich der vermerkten Reservierungen ändert sich nur, dass nicht nur ein Vorgänger- und Nachfolgerknoten, sondern aufgrund der Baumstruktur bei Multicast mehrere Nachfolger existieren können. Die meisten Änderungen betreffen die Routinginformation: Neben einer von Unicast getrennten zweiten Routinginformation – MRIB und RIB können sich unterscheiden – und den oben bei den Voraussetzungen für Multicast aufgeführten Daten sind aus Effizienzgründen ggf. weitere das Multicastrouting beeinflussende Informationen wie z. B. der aktuelle Forwarder für einen Link zu speichern, um diesen nicht bei jeder Multicastpfadbestimmung immer neu ermitteln zu müssen. Für DSMC müssen darüber hinaus die Schnittstellenindizes und die Art des vom Router unterstützten Weiterleitungszustands vermerkt werden. Die Reservierungen enthalten bei DSMC zudem die jeweils aktuell konfigurierten Werte für NRS-Zustand und Designated-Router-Bit.

4.4.2 Nachrichtenverarbeitung

Das Gegenstück zur DSMC-Komponente in den Routern ist die Nachrichtenverarbeitungsroutine im DRM. Der in diesem Abschnitt skizzierte Implementierungsvorschlag für eine Nachrichtenverarbeitungsroutine arbeitet unabhängig vom restlichen DRM. Neu eintreffende Nachrichten werden in einer Liste zur Bearbeitung durch den DRM abgelegt. Die Erkennung wiederholter und Behandlung verlorener DSMC-Nachrichten übernimmt die Routine mit Hilfe einer Hashtabelle selbständig, belastet andere Teile des DRMs damit also nicht weiter. Eine Aggregation von Conf-Nachrichten wird nicht betrachtet.

Die vorgeschlagene Implementierung speichert von den Triggern eines Routers für ein bestimmtes Tupel (S,G,Oif) immer nur den zuletzt empfangenen Typ. Ein dadurch überschriebener vorheriger Trigger anderen Typs ist aus Sicht dieser Implementierung verloren gegangen und wird wie tatsächlich verloren gegangene Trigger letztlich durch spätere TriggerUC oder ConfReply mit zurückgesetztem ACKbit (Fail) behandelt, falls durch

das Überschreiben wesentliche Informationen verloren gegangen sein sollten. Dies betrifft jedoch ausschließlich PIM-SM und ausschließlich den Problemfall eines nicht Erkannten SPTs (s. Abschnitt 4.3.3). Da ein Überschreiben bedeutet, dass der DRM mit der Verarbeitung der Trigger nicht schnell genug nachkommt, ist das Speichern der Historie ohnehin kontraproduktiv. DSMC ist so entworfen worden, dass Nachrichtenverlust, ob tatsächlich oder künstlich wie hier aufgrund vereinfachter Verarbeitung seitens der DRMs, außer zu einer verzögerten Bereitstellung von Dienstgüte oder verspäteten Freigabe nicht mehr genutzter Ressourcen keine weiteren negativen Folgen hat, es insbesondere nicht zum NRS-Problem kommt.

Trigger, TriggerUC & TriggerRpt

Für jeden empfangenen Trigger, TriggerUC und TriggerRpt wird ein Eintrag mit Quelle, Gruppe, Ausgangsschnittstelle, Router (vom dem der Trigger stammt), DSMC-Nachrichtentyp, Zeitstempel, Bearbeitet-Flag, NRSstate, DRbit, Zähler und Timer in einer Hashtabelle angelegt. Das 4-Tupel aus den ersten vier Werten dient zugleich als Schlüssel für die Hashwertberechnung. Das Bearbeitet-Flag ist initial zurückgesetzt, wenn ein Eintrag aufgrund einer eingetroffenen DSMC-Nachricht angelegt wird, andernfalls ist es initial gesetzt. Die übrigen Felder werden weiter unten erläutert. Zusätzlich wird jeder neue Eintrag an eine Liste[8] mit unbearbeiteten Einträgen angehängt, wozu jeder Eintrag entsprechende Zeiger enthält. Bei wiederholt empfangenen DSMC-Nachrichten ist bereits ein Eintrag mit demselben 4-Tupel vorhanden. In diesem Fall wird lediglich der Zeitstempel mit dem neuen Empfangszeitpunkt aktualisiert. Weitere Aktionen sind nicht nötig. Der Verarbeitungsaufwand für wiederholt empfangene Trigger ist damit sehr gering. Hat sich der Triggertyp geändert, wird zusätzlich der vermerkte Typ aktualisiert. Das Verhalten, wenn der Typ des existierenden Eintrags nicht Trigger, TriggerUC oder TriggerRpt sondern ConfReply oder TriggerStop ist, wird weiter unten beschrieben.

Der DRM arbeitet die Einträge mit Hilfe der Liste der unbearbeiteten Einträge der Reihe nach ab, also in der sie initial angelegt und an die Liste angehängt wurden. Wenn ein TriggerRpt-Eintrag bearbeitet wird, sollte vorher die Liste nach evtl. vorhandenen unbearbeiteten Trigger- oder TriggerSwitched-Einträgen (s. u.) abgesucht werden, die dasselbe Quelle-Gruppe-Tupel oder, falls die Quelladresse im TriggerRpt Null (die unspezifizierte Adresse) ist, dieselbe Gruppe aufweisen. Diese Einträge sollten dann zuerst bearbeitet werden. Umgekehrt muss bei einem Trigger jedoch nicht nach passenden TriggerRpt gesucht werden, da RPT-Pfade an SPT-Pfaden enden können, aber nicht umgekehrt (s. Abschnitt 4.4.6). TriggerUC-Einträge sollten bei der Abarbeitung übersprungen werden, bis alle anderen Triggertypen einschließlich TriggerSwitched abgearbeitet sind, da sich durch diese die Bearbeitung eines TriggerUC u. U. erübrigt. Die weiter unten erläuterten letzten zwei Typen TriggerStop und ConfReply (Fail) zeigen gelöschten Weiterleitungszustand an und sollten in zwei Schritten bearbeitet werden: Die Ressourcen beim Router, von dem das betreffende TriggerStop bzw. Fail stammt, können sofort freigegeben werden. Der durch sie angezeigte abgebaute Weiterleitungspfad wird aber ggf. von anderen für die Quelle und Gruppe neu aufgebauten Pfaden begrenzt, die erst nach Bearbeitung der anderen

[8] Es sind noch weitere hier nicht beschriebene Listen sinnvoll, z. B. eine mit allen Einträgen eines bestimmten Routers, um Conf-Nachrichten aggregieren oder bei Routerausfall Einträge schnell löschen zu können. Oder Listen mit allen Einträgen einer Gruppe und eines Quelle-Gruppe-Tupels für die schnelle Behandlung bei Änderungen in der Zuordnung von Gruppen- zu RP-Adresse oder bei Routingänderungen, usw.

Trigger mit demselben Quelle-Gruppe-Tupel oder derselben Gruppe feststehen. Ein danach verbliebener Restpfad kann anschließend freigegeben werden.

Vor Bearbeitung jedes Eintrags wird überprüft, ob der im Zeitstempel vermerkte letzte Empfangszeitpunkt schon länger zurückliegt. Trigger werden ratenlimitiert wiederholt, um Paketverlust zu begegnen. Ab einer gewissen Wartezeit muss aber davon ausgegangen werden, dass nicht Paketverlust Ursache des Nichtempfangs ist, sondern die Trigger nicht mehr gesendet werden. Typischerweise geht man bei Protokollen im Internet von nicht mehr als drei hintereinander verloren gegangenen Nachrichten aus. Liegt der zuletzt empfangene Trigger mehr als das 3-fache, z. B. das 3,5-fache, der Periodendauer nach Gleichung 4.4 zurück, kann davon ausgegangen werden, dass der MFC-Eintrag nicht mehr existiert und nicht mehr konfiguriert werden muss. Der Eintrag kann dann ohne Bearbeitung gelöscht werden. Wird der Faktor zu klein gewählt, führt dies lediglich zu einem Mehraufwand durch versehentliches Löschen und kurz darauf erfolgendes Neuanlegen. Das Faktor sollte aber auch nicht zu groß gewählt werden, damit keine unnötige Bearbeitung durch den DRM erfolgt.

Aus dem 4-Tupel des Trigger-Eintrags wird in Abhängigkeit des Typs der Multicastpfad bestimmt und die Ressourcen entlang des Pfads überprüft (s. Abschnitt 4.4.3). Daraufhin wird für jeden Router entlang des Pfads ein Eintrag mit Typ TriggerUC und gesetztem Bearbeitet-Flag in der Hashtabelle angelegt mit dem 4-Tupel des zu sendenden Conf. Im Eintrag werden zusätzlich NRSstate und DRbit des Conf für eine spätere Wiederholung vermerkt und der Zähler auf Null gesetzt. Existiert bereits ein Eintrag, wird der Eintrag aus der Liste der unbearbeiteten Einträge entfernt und die Werte wie beschrieben gesetzt. Hierdurch wird beispielsweise ein bisher noch nicht bearbeiteter TriggerUC automatisch aus der Liste entfernt und muss nicht mehr bearbeitet werden. Bei gesetztem Bearbeitet-Flag wird in der Nachrichtenverarbeitungsroutine lediglich zwischen Triggern und ConfReply unterschieden. Der genaue Triggertyp ist unerheblich, er kann einfach beibehalten werden. War der existierende Eintrag vom Typ ConfReply (s. u.), muss das Conf mit gesetztem ACKbit gesendet werden, ansonsten (Trigger, TriggerUC, TriggerRpt) mit zurückgesetztem ACKbit. Der Zähler wird um eins auf eins inkrementiert. Der Timer wird nach dem Senden mit zurückgesetztem ACKbit immer auf T_{check} gemäß Gleichung 4.3, beim Senden mit gesetztem ACKbit auf T_{RTO} gemäß Gleichung 4.1 gesetzt. Gemäß Abschnitt 4.2.9 sollte abhängig davon, ob das DRbit gesetzt ist oder nicht, T_{check} auf den kleineren bzw. den größeren von zwei Werten für T_{check} gesetzt werden, um einerseits einen möglichen Verlust des späteren TriggerStop schnell erkennen zu können, um andererseits aber auch den Signalisierungsverkehr durch das periodische Wiederholen gering zu halten. Der Zeitstempel wird auf den Zeitpunkt des Sendens des Conf gesetzt. Bei Trigger-Einträgen mit zurückgesetztem Bearbeitet-Flag (und TriggerStop-Einträgen, s. u.) vermerkt er also den letzten Empfangszeitpunkt, bei konfigurierten und bei ConfReply-Einträgen den letzten Sendezeitpunkt eines Conf.

Trifft ein TriggerUC, Trigger oder TriggerRpt ein, wird das Conf mit den gespeicherten Werten erneut gesendet, da das Conf u. U. verloren gegangen sein könnte. Dies kann ohne eine erneute Ressourcenüberprüfung durch den DRM erfolgen. Das Conf wird hier immer mit zurückgesetztem ACKbit gesendet, da offensichtlich der NRS-Zustand noch unkonfiguriert ist und der Router bei erneutem Verlust des Conf seinen Trigger erneut senden wird. Da das Bearbeitet-Flag gesetzt ist, wird der Typ des Eintrags durch eintreffende Trigger nicht mehr geändert, alle drei Triggertypen werden bei gesetztem Bearbeitet-Flag

gleich behandelt[9]. Mit jedem Senden der Conf wird der Zähler inkrementiert und der Timer mit T_{check} neu gestartet, sofern der Eintrag nicht vom Typ ConfReply ist. Bei ConfReply-Einträgen verändern empfangene Trigger dagegen grundsätzlich weder Timer noch Zähler. Überschreitet der Zähler einen konfigurierten Schwellenwert c_{fail}, sollte dies gegenüber dem Administrator als Fehler geloggt werden, da hier für eine u. U. lange Zeit keine Dienstgüte erbracht wird. Die Conf werden unabhängig davon weiterhin wiederholt (und der Zähler weiter inkrementiert). Erreicht das Conf den Router, werden daraufhin die Trigger ausbleiben, der Zustand ist konfiguriert.

ConfReply – OK

Bei Ablauf des Timers wird das Conf wiederholt. Sollte der Typ des Eintrags nicht ConfReply sein, wird er jetzt auf ConfReply geändert und der Zähler auf Null zurückgesetzt. Grundsätzlich immer, wenn das Senden durch den Timer ausgelöst wird oder der Typ des Eintrags ConfReply ist, ist das ACKbit im Conf zu setzen. Der Timer wird mit T_{RTO} neu gestartet, oder, wenn Exponential Backoff verwendet wird, mit $\min(T_{MAX}, 2^{c-o} \cdot T_{RTO})$, wobei T_{MAX} ein konfiguriertes maximales Timeout ist, c der Zähler und $o \geq 0$ ein Offset, mit dem der Timeout initial kleiner als T_{RTO} gewählt werden kann. Wann ein Exponential Backoff sinnvoll ist, wurde in Abschnitt 4.2.9 erläutert.

Der Zähler wird anschließend um eins inkrementiert. Mit dem Empfang eines entsprechenden ConfReply wird, sofern auch NRSstate und DRbit übereinstimmen und das ACKbit im empfangenen ConfReply gesetzt ist (OK), der laufende Timer gestoppt und mit T_{check} neu gestartet, wobei der Zähler auf Null zurückgesetzt wird. Der ganze Vorgang beginnt von neuem. Da T_{check} gemäß Gleichung 4.3 dynamisch nachgeführt wird, wird sich das neue T_{check} typischerweise vom letzten verwendeten T_{check} unterscheiden. Ggf. kann zur Vorbeugung gegen Synchronisationseffekte T_{check} um einen gewissen zufälligen Betrag variiert werden. Überschreitet der Zähler einen konfigurierten Schwellenwert c_{fail}, sollte dies auch hier gegenüber dem Administrator als Fehler geloggt werden, da dies ein Hinweis auf eine durch den DRM nicht erkannte oder nicht behebbare Inkonsistenz sein könnte und ggf. ein veralteter NRS-Zustand im Router konfiguriert ist. Die Conf werden unabhängig davon weiterhin wiederholt und der Zähler weiter inkrementiert.

Existiert kein passender Eintrag, oder hat der existierende Eintrag den Typ TriggerStop und dessen Holddown-Flag (s. u. bei der Behandlung von Nachrichtenvertauschungen) war zurückgesetzt ($c \geq 0$), sollte dies als Fehler geloggt werden. Um das NRS-Problem zu vermeiden, wird ein neuer ConfReply-Eintrag mit gesetztem Bearbeitet-Flag angelegt bzw. der Typ auf ConfReply geändert. Dieser Eintrag wird trotz gesetztem Bearbeitet-Flag an die Liste der unbearbeiteten Einträge angehängt, damit der DRM vom offensichtlich existierenden MFC-Eintrag mit konfiguriertem NRS-Zustand erfährt. Das DRbit wird aus dem empfangenen ConfReply übernommen und NRSstate auf Remark gesetzt. Auch wenn der NRSstate im empfangenen Conf schon Remark war, wird ein entsprechendes Conf gesendet, um ggf. ein Fail (s. ConfReply – Fail) zur provozieren, sollte das eingetroffene ConfReply nur eine ungewöhnlich lange verzögerte Nachricht sein und der MFC-Eintrag tatsächlich schon nicht mehr existieren.

[9]Eine alternative Implementierung könnte die durch den geänderten Typ erhaltene Zusatzinformation auswerten, indem der empfangene geänderte Triggertyp übernommen und der Eintrag an die Liste der unbearbeiteten Einträge angehängt wird (das Bearbeitet-Flag bleibt dabei gesetzt).

Nachrichtenvertauschungen

Wird vom DRM ein neues Conf mit geänderten Werten generiert, d. h. existiert ein Eintrag mit Typ ConfReply mit demselben 4-Tupel und abweichendem NRSstate oder DRbit, und ist der Zähler im Eintrag nicht Null, wurde also gerade ein Conf mit alten Werten gesendet und wird auf ein passendes ConfReply gewartet, müssen eventuelle Nachrichtenvertauschungen berücksichtigt werden, damit es im Router nicht zum fälschlichen Beibehalten der veralteten Werte kommt (s. Abschnitt 4.2.8). Hierzu werden die neuen Werte für NRSstate und DRbit eingetragen und das neue Conf im Abstand von T_{RTO}, wie in Abbildung 4.8 (e) dargestellt, zweimal gesendet. Ein zwischenzeitlich empfangenes ConfReply setzt also anders als sonst nicht den Timer neu auf T_{check}. Dies wird mit Hilfe des Holddown-Flags erreicht. Ist der Zählerwert negativ ($c < 0$), ist das Holddown-Flag gesetzt. Ist das Holddown-Flag gesetzt, verändern zwischenzeitlich eintreffende ConfReply den Timer und den Zähler nicht (außer das ACKbit ist nicht gesetzt, s. u.). Beim Ablauf des Timers wird wie üblich das Conf erneut gesendet, der Timer auf T_{RTO} gesetzt und der Zähler inkrementiert. Damit das Holddown-Flag nach dem Senden zurückgesetzt ist, muss der Zähler also vom DRM auf $c = -1$ gesetzt werden. Zusätzlich wird mit jedem eintreffenden ConfReply mit gesetztem ACKbit aber falschen bzw. veralteten Werten das Conf sofort erneut gesendet. Ist das Holddown-Flag nicht mehr gesetzt, wird wie üblich verfahren und das Conf bis zum Eintreffen eines aktuellen ConfReply wiederholt, woraufhin der Timer auf T_{check} und der Zähler auf Null zurückgesetzt wird. Eintreffende Trigger sollten nicht auftreten, führen aber wie üblich zum sofortigen Senden mit zurückgesetztem ACKbit und verändern in ConfReply-Einträgen weder Timer noch Zähler.

War der Eintrag noch vom Typ Trigger, TriggerUC oder TriggerRpt, als der DRM ein neues Conf mit geänderten Werten generiert, muss er prüfen, ob der im Zeitstempel vermerkte Sendezeitpunkt des letzten Conf mehr als T_{RTO} zurückliegt. Wenn nein, sind Vertauschungen hinreichend unwahrscheinlich und es genügt einfaches Senden. Es wird dann wie beim Ablauf des Timers vorgegangen (s. ConfReply – OK), also der Typ der Eintrags auf ConfReply geändert, der Zähler auf Null zurückgesetzt, das neue Conf gesendet, der Zähler inkrementiert und der Timer auf T_{RTO} gesetzt.

Bei der Überprüfung auf nicht erkannte nachträgliche Empfänger (s. Abschnitt 4.4.4.3) kann der Holddown-Mechanismus genutzt werden, um durch Setzen des Zählers auf $c = -n$ das Conf im Abstand T_{RTO} n-fach bzw. bis zum ersten eintreffenden Fail zu wiederholen.

ConfReply – Fail

Ist das ACKbit im empfangenen ConfReply nicht gesetzt (Fail), wird falls nötig der Typ auf ConfReply geändert sowie das Bearbeitet-Flag zurückgesetzt und bei Zählerstand Null (s. u.) das ConfReply in die Liste der unbearbeiteten Einträge eingetragen. Existiert kein passender Eintrag (beliebigen Typs), oder war der Eintrag vom Typ TriggerStop und dessen Holddown-Flag war zurückgesetzt (c=0), sollte dies als Fehler geloggt werden. In diesem Fall sind keine weiteren Aktionen erforderlich, es wird insbesondere kein neuer ConfReply-Eintrag angelegt.

Der DRM wird bei Abarbeitung der Liste das ConfReply entfernen und seine Ressourcenbelegung entsprechend aktualisieren, denn offensichtlich existiert der entsprechende MFC-Eintrag im Router nicht oder nicht mehr. Da stromabwärts dieses Routers kein Multicastverkehr mehr fließt, werden auch bei den Routern stromabwärts, bei PIM-SM

ggf. nur bis zu einem möglichen Verzweigungsrouter, an dem der SPT Verkehr in den RPT einspeist, die betroffenen MFC-Einträge nicht mehr existieren. Um nicht zum nächsten Conf nach T_{check} warten zu müssen, kann der DRM entweder konservativ ein erneutes Conf sofort senden, um dort ebenfalls ein Fail zu provozieren und die Ressourcen daraufhin freizugeben, oder optimistisch die Ressourcen sofort freigeben und wie bei einem TriggerStop (s. u.) verfahren und die Einträge in TriggerStop-Einträge umwandeln und deren Löschung initiieren.

Anders als bei einem Trigger ist bei den in Folge einer Routingänderung gesendeten Conf nicht sichergestellt, dass die Router längs des neu zu konfigurierenden Pfads ihrerseits die Routingänderung schon nachvollzogen und einen neuen MFC-Eintrag angelegt haben (s. Abschnitt 4.4.5). Um diesen Fall des noch nicht existierenden MFC-Eintrags effizient zu handhaben, ist es nötig, nicht mit dem ersten empfangenen Fail aufzugeben, sondern einige (wenige) Male das Conf im Abstand T_{RTO} zu wiederholen. Dazu wird das Bearbeitet-Flag vom DRM zurückgesetzt und der Zähler auf z. B. $c_{retry} = 3$ gesetzt. Mit Ablauf des Timers wird das Conf wiederholt, sofern der Zähler größer Null ist. Der Timer wird mit T_{RTO} neu gestartet und der Zähler dekrementiert. Andernfalls bei Erreichen von Zählerstand Null wird ein ConfReply-Eintrag in die Liste der unbearbeiteten Einträge eingetragen. Zwischenzeitlich eintreffende Fail haben keine Auswirkung. Wird jedoch ein OK (ConfReply mit gesetztem ACKbit) empfangen, wird das Bearbeitet-Flag gesetzt und der Timer auf T_{check} und der Zähler auf Null gesetzt, der oben beschriebene Ablauf der Conf beginnt. Das Rückwärtszählen unter zurückgesetztem Bearbeitet-Flag (im Unterschied zum Hochzählen bei gesetztem Bearbeitet-Flag) erlaubt dem DRM, Einfluss auf die Dauer der versuchten Konfiguration zu nehmen und kann so ggf. abhängig von der Art der Routingänderung von ihm unterschiedlich gesetzt werden.

TriggerStop

Beim Eintreffen eines TriggerStop wird je nach Bearbeitet-Flag und Typ des Eintrags der Timer neu gestartet: Bei einem Trigger-, TriggerUC- oder TriggerRpt-Eintrag mit zurückgesetztem Bearbeitet-Flag wird mit $T_{HoldDown}$ nach Gleichung 4.2 neu gestartet, bei gesetztem Bearbeitet-Flag oder bei einem ConfReply-Eintrag wird mit $\max(T_{HoldDown}, \text{TimeStamp} - \text{TimeNow} + T_{RTO})$ neu gestartet, wobei TimeStamp den Zeitstempel aus dem Eintrag und TimeNow den Empfangszeitpunkt des TriggerStop bezeichnen. Mit $\text{TimeStamp} - \text{TimeNow} + T_{RTO}$ wird der voraussichtlich späteste Zeitpunkt erfasst, an dem ein ConfReply auf das zum Zeitpunkt TimeStamp gesendete Conf noch eintreffen kann. Falls der Wert größer als $T_{HoldDown}$ und der Zähler nicht Null ist, wird der laufende Timer genau diesen Wert haben und muss nicht neu gestartet werden. Der Typ des Eintrags wird auf TriggerStop geändert, der Empfangszeitpunkt des TriggerStop im Zeitstempel vermerkt, das Bearbeitet-Flag zurückgesetzt, das Holddown-Flag gesetzt ($c = -1$) und der Eintrag an die Liste unbearbeiteter Einträge angehängt. Das gesetzte Holddown-Flag zeigt an, dass alle eintreffenden DSMC-Nachrichten ignoriert werden sollen (Holddown). Hierdurch wird vermieden, dass durch eine Reihenfolgevertauschung eine ggf. kurz vorher noch gesendete DSMC-Nachricht den TriggerStop überschreibt. Bei Ablauf des Timers wird das Holddown-Flag zurückgesetzt ($c = 0$) und der Eintrag ggf. gelöscht (s. u.). Nach Bearbeitung setzt der DRM das Bearbeitet-Flag und löscht den Eintrag, sofern das Holddown-Flag schon zurückgesetzt ist. Ansonsten trägt er es lediglich aus der Liste aus, es wird dann mit dem Ablaufen des Timers gelöscht.

Wenn vor Bearbeitung durch den DRM und nach Ablauf des Holddown neue Trigger eintreffen, überschreiben sie unter zurückgesetztem Bearbeitet-Flag wie üblich den Typ des Eintrags. Der DRM sieht und bearbeitet dann keinen TriggerStop mehr sondern den entsprechenden Triggertyp. Hier kann nicht unterschieden werden, ob $T_{HoldDown}$ zu kurz war oder der Trigger tatsächlich erst nach dem TriggerStop generiert wurde. Der Empfang eines ConfReply nach dem Holddown wird wie dort bereits beschrieben gehandhabt: Anders als Trigger ist ein ConfReply sicher nicht nach dem TriggerStop generiert worden. Sie werden daher als Fehler geloggt, um auf das offensichtlich zu klein gewählte $T_{HoldDown}$ hinzuweisen. Zusätzlich wird der DRM informiert, um sicherzustellen, dass seine Ressourcenbelegung korrekt ist. Zeigt das ConfReply zudem existierenden Zustand an, wird zur Vermeidung des NRS-Problems dieser vorsorglich auf Remark umkonfiguriert.

Aus dem TriggerStop berechnet der DRM den abgebauten Pfad und verfährt mit allen betroffenen Einträgen ähnlich wie gerade beschrieben. Wurde vor mehr als T_{RTO} das letzte Conf gesendet, ist also TimeNow − TimeStamp $> T_{RTO}$, kann der Eintrag sofort gelöscht werden. Andernfalls wird der Timer mit TimeStamp − TimeNow + T_{RTO} neu gestartet. Wenn der Zähler ungleich null ist, wird der Timer bereits genau diesen Wert besitzen und muss nicht verändert werden. Der Typ wird auf TriggerStop geändert, Bearbeitet- und Holddown-Flag werden beide gesetzt und der Empfangszeitpunkt im Zeitstempel vermerkt (optional). Mit Ablauf des Timers wird der Eintrag gelöscht. Alle bis dahin zwischenzeitlich eintreffenden DSMC-Nachrichten mit dem 4-Tupel werden ignoriert. Neben diesem optimistischen Vorgehen kann der DRM alternativ, wenn nicht sicher ist, ob noch Weiterleitungszustand existiert (s. Abschnitt 4.4.4), konservativ wie oben bei ConfReply (Fail) beschrieben vorgehen. Ein Eintrag wird dann nicht in TriggerStop umgewandelt, sondern lediglich durch erneutes Senden des Conf versucht ein Fail zu provozieren und erst dann die Ressourcen freigegeben und der Eintrag gelöscht.

TriggerSwitched

Beim Empfang eines TriggerSwitched wird ein Eintrag in der Hashtabelle angelegt, wobei für die hier nicht verwendete Ausgangsschnittstelle Null eingetragen wird (oder ein anderer beliebiger aber fester und für eine Schnittstelle nicht gültiger bzw. nicht verwendeter Wert). Die 4-Tupel der TriggerSwitched und TriggerSwitchedAck sind dadurch niemals identisch mit denen der übrigen DSMC-Nachrichten, ein Überschreiben des Typs findet also nie statt. Das Bearbeitet-Flag ist zurückgesetzt und der Zähler wird auf Null gesetzt. Wie alle anderen Triggertypen werden neue Einträge an die Liste der unbearbeiteten Einträge angehängt. Die Nachrichtenverarbeitungsroutine kann aus dem TriggerSwitched ohne weiteres Zutun des DRMs direkt ein passendes TriggerSwitchedAck ableiten, das sofort gesendet wird. Bei erneutem Empfang eines TriggerSwitched wird der Empfangszeitpunkt aktualisiert und das TriggerSwitchedAck erneut gesendet. Mit jedem Senden wird der Zähler inkrementiert. Auch hier sollte bei Erreichen eines bestimmten Werts c_{fail} ein Fehler geloggt werden.

Das Löschen erfolgt beim Abarbeiten der Einträge der Liste durch den DRM. Liegt der letzte Empfangszeitpunkt hinreichend lange zurück (z. B. das 3,5-fache der Periodendauer T_{RTO} der TriggerSwitched, ggf. auch $1/R$ nach Gleichung 4.4, s. Abschnitt 4.3.3.1), kann der Eintrag direkt nach Bearbeitung gelöscht werden, andernfalls wird er in eine zweite Liste mit zu löschenden Einträgen umgetragen. Diese wird von Zeit zu Zeit durchgegangen und Einträge mit hinreichend lange zurückliegendem Empfangszeitpunkt gelöscht. Der

genaue Zeitpunkt des Löschens ist hier unkritisch, da sie nicht die Verarbeitung anderer Nachrichtentypen beeinflussen. Alternativ kann nach erfolgter Bearbeitung der Trigger-Switched-Eintrag aus der Liste der unbearbeiteten Einträge ausgetragen und der Timer mit einem passenden Wert TimeStamp − TimeNow + $3{,}5 \cdot T_{RTO}$ (oder ggf. $1/R$ statt T_{RTO}, s. o.) gestartet werden. Bei Ablauf des Timers wird der TriggerSwitched-Eintrag schließlich gelöscht. Die durch den TriggerSwitched ausgelösten Conf führen zum Anlegen von ConfReply-Einträgen, die Verarbeitung erfolgt wie dort beschrieben.

4.4.3 Multicastpfadbestimmung

Verarbeitet der DRM einen Trigger, wird als erster Schritt anhand der genannten Quell- und Gruppenadresse das entsprechende Reservierungsprofil, ggf. auch die Profile weiterer in der Gruppe aktiver Quellen (nötig für quellenunspezifischen Weiterleitungszustand, s. u.), gesucht und die Verkehrs- und Dienstgüteparameter ausgelesen, anhand derer nachfolgend auf Ressourcenverfügbarkeit hin geprüft wird. Fehlt das Reservierungsprofil, zeigt dies eine Fehlkonfiguration des Policings im First-Hop-Router oder Ingressrouter an, denn dieser hätte den hochprioren Verkehr der Quelle, der zum Trigger geführt hat, gar nicht passieren lassen dürfen.

Auf Basis des der Gruppe zugeordneten Multicastroutingprotokolls und der MRIB ist danach der Pfad vom Link mit dem neuen Empfänger, bestimmt durch den den Trigger sendenden Router und den im Trigger genannten Schnittstellenindex, zur Quelle oder je nach Protokoll und Triggertyp zur Rendezvousstelle als Zieladresse zu bestimmen. Entlang des so bestimmten Pfads wird jeder Link anhand der im ersten Schritt ermittelten Parameter auf hinreichend Ressourcen überprüft und die Ressourcen wenn verfügbar belegt. Wenn der Router quellenunspezifischen (*,G)-Weiterleitungszustand nutzt, und sich stromaufwärts nicht mindestens ein (S,G)-Zustand nutzender Router befindet, müssen die verfügbaren Ressourcen für alle Quellen der Gruppe ausreichen, da hier der NRS-Zustand nicht mehr nach Quellen differenzieren kann.

Bei der Pfadberechnung ist vom DRM für jeden Link der Forwarder zu berechnen und dessen Ausgangsschnittstelle mit NRS-Zustand zu konfigurieren. Dies ist bei BIDIR-PIM der DF, bei PIM-DM der Assert-Gewinner und beim IGMP/MLD-Proxying der Querier. Bei PIM-SM bestimmt, solange es nur einen Downstreamrouter an einem Link gibt, dessen Metrik zur Quelle bzw. Rendezvousstelle den RPF'(S,G) bzw. RPF'(*,G) und damit den Forwarder. Erst mit einem weiteren Downstreamrouter am selben Link, und auch nur dann, wenn dieser einen abweichen RPF' bestimmt und daher sein Join zu diesem abweichenden Upstreamrouter schicken würde, kommt es zu einem Assert und der Forwarder ist wie bei PIM-DM der Assert-Gewinner. Die Basis der Berechnung bildet in allen Fällen die MRIB mit den IP-Adressen der Router als Tie-Breaker. Bei PIM-SM kommt an den Enden eines Pfads die durch die Hello-Option DR Priority beeinflusste DR-Wahl hinzu. Man beachte, dass diese ohne Einfluss bleibt, wenn mindestens ein Router am Link diese nicht unterstützt.

TriggerUC-Nachrichten sollten, solange die Pfadberechnung und erste Konfiguration mit NRS-Zustand nicht abgeschlossen ist, ignoriert werden. TriggerUC-Nachrichten werden mit anderen Worten erst dann bearbeitet, wenn alle anderen Triggertypen abgearbeitet worden sind (s. voriger Abschnitt 4.4.2). Nachfolgende TriggerUC zeigen verlorene Conf-Nachrichten an, wenn der in ihnen genannte NRS-Zustand hätte konfiguriert sein sollen. Dies wird durch die oben beschriebene Nachrichtenverarbeitungsroutine bereits gehandhabt. Ansonsten zeigen sie eine Inkonsistenz an.

4.4.4 Behandlung von Inkonsistenzen

Inkonsistenzen zwischen den tatsächlichen Multicastpfaden und den vom DRM angenommenen und konfigurierten Pfaden sorgen dafür, dass Dienstgüte nicht erbracht wird. Das NRS-Problem wird autonom durch die Router gelöst, da sie solange auf den niederprioren DSCP_UC ummarkieren wie der DRM keine explizite Konfiguration durchführt. Inkonsistenzen entstehen einerseits durch verlorene oder in gewissen Fällen nicht gesendete DSMC-Nachrichten. Verlorene DSMC-Nachrichten stellen i. A. kein Problem dar, da sie periodisch wiederholt werden. Sie führen schlimmstenfalls zum unnötigen Senden von TriggerUCs (Abschnitt 4.4.4.1). Probleme entstehen dort, wo Trigger entweder nur einmal (Abschnitt 4.4.4.2) oder gar nicht (4.4.4.3) gesendet werden.

Andererseits sind Inkonsistenzen Folge von geändertem Uni- und Multicastrouting nach einem Linkausfall, Routerausfall oder -neustart oder geänderter Zuordnung von Gruppenadressen zu Multicastroutingprotokollen und ggf. RP-Adressen. Diese sind normalerweise transienter Natur und mit dem erneuten Konvergieren des Routings behoben. Routingänderungen lösen bei DSMC wie Trigger eine Pfadneuberechnung, Umbelegung von Ressourcen und Umkonfiguration von NRS-Zustand aus. Der folgende Abschnitt 4.4.5 beschreibt das Vorgehen nach einer Routingänderung und Abschnitt 4.4.6, was bei PIM-SM dabei zusätzlich zu beachten ist.

Schließlich entstehen Inkonsistenzen durch z. B. inkonsistente/fehlerhafte Konfiguration von Routern, Fehlfunktionen oder Implementierungsfehler. Solche Inkonsistenzen können nicht behoben sondern lediglich als Fehler gegenüber dem Administrator geloggt werden. In diesem Zusammenhang kann beim Abbau von Weiterleitungszustand zwischen optimistischem und konservativem Vorgehen beim Löschen des zugehörigen NRS-Zustands unterschieden werden. Wenn nach Ansicht des DRMs kein Weiterleitungszustand mehr existieren sollte, wird er bei optimistischem Vorgehen mit dem Löschen der Reservierungen im Ressourcengraphen und Freigabe der Ressourcen auch den vermerkten NRS-Zustand sofort löschen. Wird allerdings durch Fehler (vom DRM falsch berechneter Pfad etc.) der in den Routern etablierte Weiterleitungs- und damit auch der NRS-Zustand gar nicht gelöscht, kann es zum NRS-Problem kommen, sofern der konfigurierte Zustand Keep war. Beim konservativen Vorgehen wird daher der NRS-Zustand zusammen mit den dann nur als gelöscht markierten Reservierungen solange im Ressourcengraphen vermerkt bleiben, bis ein ConfReply mit zurückgesetztem ACKbit (Fail) von jedem Router entlang eines vom DRM als abgebaut angenommenen Pfads empfangen wurde. Vor einer neuen Reservierung kann ggf. mittels Conf(Ack) ein noch nicht erhaltenes ConfReply explizit angefordert werden.

4.4.4.1 TriggerUC

Die TriggerUC treten im regulären Betrieb bei rendezvousstellenbasierten Multicastroutingprotokollen auf (s. Abschnitt 4.2.5), sind hier also nicht als Inkonsistenz zu interpretieren, sondern der NRS-Zustand ist wie dort beschrieben zu konfigurieren.

Ansonsten treten TriggerUC bei verlorenen Triggern auf, wenn durch den Verlust die Unterdrückungszeit abläuft, bevor die verspätete Konfiguration des Pfads erfolgt. TriggerUC dürfen deshalb nicht sofort als eine nicht behebbare Inkonsistenz gegenüber dem Administrator geloggt werden. Erst, wenn aufgrund eines TriggerUC NRS-Zustand konfiguriert und Ressourcen belegt wurden und für diese bis zur späteren Wiederfreigabe im Rahmen

der periodischen Überprüfung nach T_{check}, oder innerhalb einer vom Administrator vorgegebenen Zeit, kein Trigger eintrifft, der ebenfalls zur Konfiguration dieses NRS-Zustands führen würde, liegt eine nicht erkannte/behebbare Inkonsistenz vor.

4.4.4.2 Verlust von TriggerStop

TriggerStop werden anders als alle anderen DSMC-Nachrichten nicht wiederholt. Geht ein TriggerStop verloren, wird dies erst mit dem nächsten periodischen Conf nach T_{check} festgestellt, wenn der DRM ein unerwartetes Fail von einem Router entlang des Pfads empfängt. Ein Fail muss daher vom DRM nicht nur zur Freigabe der Ressourcen auf dem angezeigten Link führen, sondern auch als Hinweis auf einen verloren gegangenen TriggerStop interpretiert werden. Die Ressourcen entlang des gesamten Pfads können daraufhin freigegeben werden. Bei Empfang eines Fail wird demnach so verfahren, als wäre von allen Routern stromabwärts, die zuvor einen Trigger oder TriggerRpt gesendet haben, ein TriggerStop empfangen worden. Dies schließt die Überprüfung auf unerkannte Empfänger längs des Pfads mit ein, wobei hier jedoch davon ausgegangen werden kann, dass stromabwärts des Routers, von dem das Fail stammt, kein Empfänger mehr existiert.

4.4.4.3 Unerkannte Empfänger

Trigger werden in DSMC nur dann gesendet, wenn neuer Weiterleitungszustand etabliert wurde und eine Konfiguration von NRS-Zustand durch den DRM erforderlich ist. Nachträglich längs eines bereits konfigurierten Pfads hinzukommende Empfänger bleiben somit unerkannt, außer sie sind wie z. B. bei MOSPF bereits durch Teilnahme am Multicastrouting bekannt (s. Abschnitt 4.2.13.4). Wird ein Pfad abgebaut, angezeigt durch einen TriggerStop oder eine Routingänderung, muss der DRM alle Links entlang des Pfads auf solche möglicherweise nachträglich hinzugekommenen Empfänger hin überprüfen, wenn er z. B. wegen Ressourcenknappheit nicht die automatische Freigabe durch die nächste periodische Überprüfung nach T_{check} abwarten kann.

Sind dem DRM die Links bekannt, die auch Empfänger besitzen können, muss er nur diese mittels Conf(Ack) an den Forwarder des Links überprüfen. Die Ressourcen stromabwärts eines solchen Links können sofort freigegeben werden, stromaufwärts kann der DRM sie erst nach Empfang eines Fail freigeben bzw. muss sie bei Empfang eines OK beibehalten. Danach sollte der DRM auf dem Forwarder des erkannten neuen Empfängers (der letzte Router entlang des Pfads, der ein OK gesendet hat) das DRbit per Conf(DR,Ack) setzen, so dass dieser Router später ein TriggerStop senden wird. Sind es nur wenige Links, die sowohl Empfänger als auch abhängige Router haben können, kann es sinnvoll sein, dass der DRM grundsätzlich Router an einem solchen Link durch ein Conf(DR) mit gesetztem DRbit konfiguriert. Dies führt zwar ggf. zu mehreren TriggerStop, wenn entlang eines Pfads mehrere solcher Router liegen, erspart aber das Conf(Ack) und ConfReply je Router. Sind die meisten Links mögliche Kandidaten, kann es sinnvoll sein, nicht sofort jeden Router mittels Conf(Ack) parallel zu überprüfen. Denn ist die Wahrscheinlichkeit eines nachträglichen Empfängers gering (geringe Empfängerdichte), sollte zuvor der Pfadanfang überprüft werden. Wenn dort bereits der Weiterleitungszustand abgebaut wurde, ist kein Empfänger unerkannt nachträglich hinzugekommen und die Ressourcen können wie vermutet längs des gesamten Pfads ohne Überprüfung von weiteren Routern freigegeben werden. Nur wenn er dort nicht abgebaut wurde, muss der wie beschrieben per Conf(Ack) an jeden Router der neue Empfänger gefunden werden. Man beachte, dass der TriggerStop anders als ein Trigger, TriggerRpt oder TriggerSwitched nicht den Abschluss des Vorgangs

(hier des Abbaus von Weiterleitungszustand) anzeigt, sondern den Beginn. Bei der beschriebenen Überprüfung sollte daher wie bei einem Trigger nach einer Routingänderung mehrfach und/oder verzögert überprüft werden (vgl. die Erläuterungen zum Holddown unter Nachrichtenvertauschungen in Abschnitt 4.4.2).

4.4.5 Routingänderung

Die TriggerUC werden während der kurzen Konvergenzzeit von Link-State-Routingprotokollen und Distance-Vector-Protokollen mit getriggerten Updates nach einer Routingänderung je nach Dauer der Konvergenz u. U. vollständig unterdrückt. Die Konvergenzzeit ist durch die Ausbreitung der geänderten Routinginformation im Netz, also der Geschwindigkeit der Signalisierung MaxSignalingDelay bestimmt, nach der sich die Unterdrückungszeit der TriggerUC richtet (s. Abschnitt 4.2.10). Alle nicht unterdrückten TriggerUC werden vom DRM ignoriert, bis er seinerseits die Neuberechnung der Pfade und deren Konfiguration mit NRS-Zustand abgeschlossen hat.

Anders als bei einem Trigger ist bei einer Routingänderung als Auslöser der Neuberechnung nicht sichergestellt, dass jeder Router den neuen Weiterleitungszustand schon etabliert hat, wenn die Konfiguration des NRS-Zustands durch den DRM erfolgt. Sollte der DRM die Pfade schneller neuberechnen als ein Router neuen Weiterleitungszustand etabliert, wird ein Router das Conf durch ein entsprechendes ConfReply Fail negativ bestätigen. Dies wird dadurch gehandhabt, dass der DRM den Konfigurationsversuch im Falle einer Routingänderung einige Male wiederholt, so wie es in der oben vorgeschlagenen Nachrichtenverarbeitungsroutine umgesetzt ist. Gelingt trotz mehrerer Versuche keine Konfiguration, muss angenommen werden, dass der geänderte Pfad nicht etabliert wurde. Die Ressourcen müssen dann freigegeben werden. Sollte später doch noch ein neuer Pfad etabliert werden. wird dies ein TriggerUC später anzeigen. Bei PIM-SM ist darüber hinaus das Problem der zwei Verteilbäume SPT und RPT zu beachten, wie nachfolgend erläutert wird.

4.4.6 PIM-SM

Der DRM muss zur Pfadberechnung bei PIM-SM zusätzlich wissen, ob der RP oder ein DR für eine bestimmte Quelle und Gruppe den SPT oder RPT verwendet. DSMC reduziert über den TriggerSwitched die Signalisierung jedoch soweit, dass dieses Wissen nur teilweise vorhanden ist. Dies ist solange unerheblich, solange SPT und RPT bei den betroffenen Routern übereinstimmen. Ändert sich dies durch eine Routingänderung (geänderte MRIB oder Zuordnung von Gruppen- zu Rendezvousstellenadressen), muss der DRM über das Netzwerkmanagement den aktuellen Wert des SPTbits auf allen DRs abfragen, von denen er das Wissen nicht besitzt und bei denen sich nach der Routingänderung RPT und SPT unterscheiden. Im Folgenden wird zusammengefasst, welche Router dies sind und wie nach einer Routingänderung vorgegangen werden kann.

Bei DSMC signalisiert ein DR, der einen TriggerRpt gesendet hat, nicht nochmals beim SPT Switchover zum DRM, sondern nur der Verzweigungsrouter stromaufwärts und dieser auch nur, wenn bei ihm der SPT nicht bereits etabliert ist (durch einen anderen DR). Für alle Router stromabwärts eines Verzweigungsrouters erfährt der DRM nicht mehr von einem eventuellen SPT Switchover. Einzig im Fall, dass sich der Pfad stromabwärts eines einen TriggerSwitched sendenden Routers nicht verzweigt, es also zum Zeitpunkt des TriggerSwitched nur genau einen DR gab (der einen TriggerRpt gesendet hat), und der Verzweigungsrouter nicht der RP ist, kann für diesen DR vermerkt werden, dass er den

SPT nutzt. Der RP sendet anders als ein DR immer einen TriggerSwitched, da er wegen des Register-Tunnels immer auch Verzweigungsrouter ist. Da neben den DRs auch der RP den SPT Switchover initiieren kann, kann ein TriggerSwitched keinem anderen DR stromabwärts zugeordnet werden, auch wenn es nur einen DR gibt. Ist die Quelle direkt am RP angeschlossen, sendet der RP zwar keinen TriggerSwitched, schaltet jedoch wie jeder DR einer Quelle sofort mit dem ersten von einer Quelle empfangenen Datenpaket auf den SPT um.

Zusammengefasst weiß der DRM für jedes Quelle-Gruppe-Tupel, dass der DR und alle Router, die einen Trigger gesendet haben, den SPT nutzen. Ebenso weiß der DRM aus dem TriggerSwitched jederzeit, was der RP und ggf. ein weiterer DR nutzt, wenn es zum Zeitpunkt des TriggerSwitched der einzige DR stromabwärts des den TriggerSwitched sendenden Routers war und der TriggerSwitched nicht vom RP kam. Bei allen übrigen DRs ist dem DRM unbekannt, welchen Baum sie nutzen. Nach einer Routingänderung berechnet der DRM als ersten Schritt den SPT zu jedem DR, von dem er weiß, dass er zuvor den SPT nutzte. Anschließend wird von jedem anderen DR der Pfad von SPT und RPT berechnet, bis der im ersten Schritt berechnete SPT erreicht ist. Gibt es keine Unterschiede zwischen den zwei Pfaden, ist auch nach der Routingänderung der Wert des SPTbits unerheblich. Andernfalls ist dieser per Netzwerkmanagement abzufragen und daraufhin der entsprechende Pfad zu konfigurieren.

4.4.7 Inkrementeller Einsatz

Da DSMC keine Änderungen an den Multicastroutingprotokollen vornimmt, sind die Modifikationen Router-lokal und im Multicastrouting nach außen hin nicht sichtbar, ein inkrementeller Einsatz ist damit möglich. Bei nicht modifizierten Routern („Legacy-Router“) kommt es ohne besondere Behandlung zum NRS-Problem. Hier müssen daher die durch die Multicastkommunikation beanspruchten Ressourcen bei der Ressourcenbelegung im DRM durch einen Broadcast nach oben hin abgeschätzt werden. Es können prinzipiell zwei Fälle unterschieden werden, je nachdem, ob der Netzbereich mit Legacy-Routern stromabwärts oder stromaufwärts des Netzbereichs mit DSMC-fähigen Routern liegt:

- **Stromabwärts:** Der Ausgang eines DSMC-fähigen Routers zu einem Link mit einem oder mehreren Legacy-Routern wird als DSMC-Grenzschnittstelle konfiguriert. Ein von ihm gesendeter Trigger oder TriggerRpt sorgt im DRM für die Ressourcenbelegung auf allen möglichen Pfaden stromabwärts des nachfolgenden Legacy-Routers, als würden die Pakete der Quelle dort gebroadcastet werden. Eine Abfrage aller Legacy-Router durch den DRM, um die tatsächlich angelegten MFC-Einträge bzw. Multicastpfade zu ermitteln und nicht auf allen Links Ressourcen belegen zu müssen, ist nicht möglich, da Legacy-Router hochpriore DSCPs ohne Einschränkung duplizieren und somit nachträglich hinzukommende Empfänger bei einem solchen Vorgehen zum NRS-Problem führen würden.

- **Stromaufwärts:** Mit dem Reservierungswunsch einer Quelle wird die Ressourcenverfügbarkeit auf jedem Link im Netzbereich der Legacy-Router überprüft. Sind dort überall hinreichend Ressourcen verfügbar, werden diese belegt und das Profil im First-Hop-Router angepasst, so dass es hochpriore DSCPs passieren lässt. Andernfalls kann es zum NRS-Problem kommen und der Reservierungswunsch der Quelle muss vom DRM abgelehnt werden.

Wenn Empfänger im Bereich der Legacy-Router ausgeschlossen werden können, z. B. durch Deaktivieren des Gruppenmanagementprotokolls, kann bis auf PIM-SM so vorgegangen werden, als wären die Router DSMC-fähig. Zwar entfällt die Absicherung durch TriggerUC, jedoch genügen ohne Inkonsistenzen allein die Trigger der Designated Router der Empfänger für die Pfadberechnung, die unter dieser Voraussetzung alle DSMC-fähig sind. Bei PIM-SM kann ggf. der Verzweigungsrouter ein Legacy-Router sein. In diesem Fall müssen die Ressourcen sowohl längs des SPT wie des RPT überprüft und belegt werden, da kein TriggerSwitched einen späteren SPT Switchover anzeigen wird.

In Netzbereichen sowohl mit DSMC-fähigen wie mit Legacy-Routern muss jeder Link stromabwärts eines Legacy-Routers bis zum Erreichen eines DSMC-fähigen Routers auf Ressourcenverfügbarkeit überprüft werden, da die Duplizierung hier vom DRM nicht kontrolliert werden kann. DSMC-Grenzschnittstellen müssen nur dann konfiguriert werden, wenn sich Empfänger an einem der Legacy-Router befinden können. Aber auch ohne Konfiguration von DSMC-Grenzschnittstellen kann in jedem Fall der für einen Empfänger an einem Legacy-Router etablierte Multicastpfad über TriggerUC erkannt und mit NRS-Zustand konfiguriert werden, wenn auch etwas verzögert.

4.4.8 DSMC Interdomain

Ein DRM kommuniziert ausschließlich mit den Routern seiner eigenen Domäne sowie mit den DRMs aller benachbarten Domänen bzw. Autonomen Systeme. Die Schnittstellen aller Grenzrouter zu Links, die in benachbarte Domänen führen, werden als DSMC-Grenzschnittstellen konfiguriert. Internetweiter Multicast findet derzeit ausschließlich auf Basis von PIM-SM statt. Befinden sich Quelle und Rendezvousstelle in unterschiedlichen Domänen bzw. Autonomen Systemen – was nur bei IPv6 unter Einsatz der Multicastadressen mit eingebetteten RP-Adressen auftritt –, und sendet der RP keinen Register-Stop, wird der DRM der Quelle einen TriggerUC erhalten und für den Register-Tunnel eine Unicastreservierung bis zum RP etablieren. Der DRM des RPs kennt damit auch die Verkehrs- und Dienstgüteparameter der Quelle. Sendet der RP sofort einen Register-Stop, entfällt dieser Schritt.

Nach Aufbau des Weiterleitungszustands erhält der DRM der Quelle, oder der DRM des RPs im Falle des Register-Tunnels bis zum RP, einen Trigger oder TriggerRpt von einem seiner Egressrouter und überprüft daraufhin die Ressourcenverfügbarkeit anhand der bereits bekannten Verkehrs- und Dienstgüteparameter. Sind hinreichend Ressourcen verfügbar, fragt der DRM seinerseits über die unverändert von Unicast übernommene Dienstgütesignalisierung den DRM der stromabwärts vom Egressrouter liegenden Domäne, ob die Quelle für diese Gruppe berechtigt ist, in der Nachbardomäne stromabwärts Dienstgüte zu nutzen. Durch diese Anfrage erhält auch jener DRM stromabwärts die Verkehrs- und Dienstgüteparameter. Ist die Quelle berechtigt, kann der DRM anschließend den NRS-Zustand in seiner Domäne mit Conf(Keep) konfigurieren, so dass hochpriore Pakete beginnen, durch seine Domäne zu fließen. Sollte die Quelle nicht berechtigt sein, wird hochpriorer Verkehr nach dem eigenen Egressrouter nicht mehr weiterfließen. Der Verkehr sollte in diesem Fall per Conf(Remark) an den eigenen Egressrouter von diesem ummarkiert werden, ggf. auch schon von einem Router weiter stromaufwärts im Inneren der Domäne, um unnötige Ressourcennutzung innerhalb der eigenen Domäne zu vermeiden.

War die Quelle berechtigt, wird auch der benachbarte DRM stromabwärts das Profil in seinem Ingressrouter so anpassen, das fortan dienstgütebehaftete Pakete (Pakete mit

hochpriorem DSCP) der Quelle an die Gruppe passieren können. Infolgedessen wird nun dessen Egressrouter einen Trigger oder TriggerRpt an ihn senden und der DRM wird seinerseits beim DRM seiner stromabwärts liegenden Nachbardomäne hinsichtlich Berechtigung der Quelle zum Senden an die Gruppe anfragen. Dieser Vorgang pflanzt sich fort, bis schließlich dienstgütebehaftete Pakete ganz bis zum Empfänger fließen, die Dienstgüte wird erbracht.

Man beachte, dass die ausgezeichneten DSCPs von jeder Domäne unabhängig von anderen Domänen gewählt werden können, da durch die Grenzrouter eine vollständige Umsetzung von DSCPs entsprechend der mit einer Nachbardomäne vereinbarten SLS durchgeführt wird. DSCP_UC und DSCP_LE werden dabei in einen DSCP eines regulären niederprioren Dienstes wie Best Effort oder Lower Effort umgesetzt.

4.5 Zusammenfassung

Mit DSMC wurde ein Verfahren zur Bereitstellung von Dienstgüte auf Basis von Diffserv für die Multicastkommunikation im Internet entwickelt. Es löst das hierbei auftretende NRS-Problem für beliebige Multicastroutingprotokolle. Es gestattet einem zentralen Ressourcenmanager, mit einer einfachen und skalierbaren Signalisierung, die Ressourcennutzung durch Multicastkommunikation in seiner Domäne zu kontrollieren. Mit Hilfe der DSMC-Grenzrouter-Funktionalität ist sowohl ein inkrementeller Einsatz innerhalb des eigenen Netzes als auch der Einsatz für internetweiten Multicast unabhängig davon möglich, ob benachbarte Autonomen Systeme DSMC nutzen oder nicht.

DSMC setzt auf der Ebene der Multicastweiterleitung im MFC an. Es bleibt dadurch unabhängig von den Multicastroutingprotokollen. DSMC ist universell einsetzbar, solange die Multicastweiterleitung auf Basis eines MFCs erfolgt. DSMC führt im MFC den sogenannten NRS-Zustand mit den vier Zuständen Init, Invalid, Remark und Keep ein. Seine Konfiguration erfolgt zentral durch den DRM mit einer von der Weiterleitung von Multicastpaketen ausgelösten Signalisierung und insgesamt fünf verschiedenen Nachrichten. DSMC wurde für den unzuverlässigen Transport seiner Signalisierungsnachrichten entworfen. So ist auch unter Nachrichtenverlust die zuverlässige, wenn auch verzögerte, Bereitstellung von Dienstgüte möglich.

Das im MFC ebenfalls neu eingeführte Designated-Router-Bit erlaubt es DSMC, mit nur einer Nachricht, dem sogenannten Trigger, die Konfiguration für alle Router eines neu eingerichteten Multicastpfads beim DRM anzufordern. Gegenüber dem einfachen Anfordern einer Konfiguration durch jeden Router eines Pfads reduziert DSMC so die Anzahl an Triggernachrichten deutlich. Im selben Maße sinkt die Belastung des Ressourcenmanagers, was DSMC skalierbar macht. Die Anzahl der Konfigurationsnachrichten an die Router wird dagegen nicht reduziert. Dies würde nur eine geringe Aufwandsverringerung für den Ressourcenmanager bedeuten und hätte somit nur geringen Einfluss auf die Skalierbarkeit von DSMC. Es wurde jedoch skizziert, wie sich mit Hilfe eines neuen IPv6-Erweiterungskopfs, dem „DSMC Conf Header“, zumindest für IPv6 auch die Anzahl an Konfigurationsnachrichten auf eine reduziert werden kann. Je neuem Multicastpfad sind dann nur noch insgesamt zwei DSMC-Nachrichten je neuem Empfänger bzw. Multicastpfad nötig.

DSMC ist für zentrales Ressourcenmanagement entwickelt worden. Der auf die Zustände Keep und Remark vereinfachte NRS-Zustand bietet jedoch auch unter verteiltem

Dienstgütemanagement die Möglichkeit, das NRS-Problem zu lösen, ohne dabei in das Multicastrouting eingreifen oder vollständiges Policing auf domäneninternen Routern durchführen zu müssen. Die mit DSMC eingeführte Signalisierung wird bei verteiltem Dienstgütemanagement nicht benötigt.

Das Basisverfahren von DSMC ist für alle Multicastroutingprotokolle geeignet, wenn der durch ein Multicastroutingprotokoll aufgebaute Pfad allein aus Kenntnis der zugrundeliegenden MRIB berechnet werden kann. Damit ist DSMC für alle für das Internet standardisierten Multicastroutingprotokolle geeignet. Für PIM-SM gilt die Einschränkung, dass zusätzlich bekannt sein muss, ob der Rendezvousstellenbaum oder der quellenspezifische kürzeste Baum verwendet wird bzw. wann zwischen beiden gewechselt wird. DSMC besitzt daher eine Erweiterung für PIM-SM, um mit nur einer Nachricht je Umschaltung vom Rendezvousstellenbaum zum quellenspezifischen kürzesten Baum auszukommen (ein Zurückumschalten zum Rendezvousstellenbaum findet auf Ebene der Multicastweiterleitung nicht statt).

Da DSMC das Multicastrouting nicht verändert, ist ein inkrementeller Einsatz mit Hilfe der DSMC-Grenzrouter-Funktionalität möglich. Sie erlaubt auch, DSMC zur Bereitstellung von Dienstgüte bei domänenübergreifendem Multicastrouting für internetweite Gruppen einzusetzen. Die DSMC-Signalisierung erfolgt nur innerhalb einer Domäne zwischen Ressourcenmanager und Routern und ist unabhängig davon, ob DSMC in Nachbardomänen eingesetzt wird oder nicht. Die Dienstgütesignalisierung wird nicht verändert, sondern es kann die bereits für Unicast vorhandene für DSMC unverändert weitergenutzt werden.

5 Implementierung und Evaluierung

Das Verfahren DSMC wurde unter XORP (Extensible Open Router Platform) [192] Version 1.3 und FreeBSD [69] Version 5.4 prototypisch implementiert, um die praktische Umsetzbarkeit zu demonstrieren. Nachfolgend werden die für DSMC erweiterte Multicastrouting-API von FreeBSD und die entscheidenden zu ihrer Nutzung an XORP durchgeführten Erweiterungen vorgestellt. Weiterhin wurden mit OMNeT++ [130] Version 3.2 simulativ die mittleren Multicastpfadlängen von zufällig generierten Topologien ermittelt, so dass sich die potentiellen Einsparungen an Triggern, die mit dem Basisverfahren von DSMC gegenüber SimpleDSMC (s. Abschnitt 4.2.14) möglich sind, quantitativ einordnen lassen.

5.1 FreeBSD

FreeBSD repräsentiert in der prototypischen Implementierung die Weiterleitungsebene des Routers (unterer Teil mit FIB und MFC in Abbildung 4.2, Seite 81). Die Wahl fiel auf FreeBSD und nicht Linux, da es verglichen mit Linux besser strukturiert und dokumentiert ist. Zudem bietet es gerade im Bereich Multicast einen größeren Funktionsumfang. So unterstützt nur FreeBSD Multicastrouting für IPv6. Die gesamte Implementierung von DSMC in FreeBSD kommt inklusive aller optionalen Fähigkeiten, Nachrichten und Statistikzähler mit etwa 300 Zeilen (exklusive Kommentare) C-Code je Adressfamilie aus. Nach einem Überblick über die Multicastweiterleitung von FreeBSD werden die für DSMC vorgenommenen Erweiterungen an dessen Multicastrouting-API erläutert. Ihre Nutzung zur routerseitigen Umsetzung von DSMC demonstriert die für XORP prototypisch implementierte DSMC-Komponente. Sie wird in Abschnitt 5.2 vorstellt.

5.1.1 Multicastweiterleitung

Abbildung 5.1 zeigt den vereinfachten Multicastweiterleitungspfad von FreeBSD für IPv4, beschränkt auf die Vermittlungsschicht. Die Multicastweiterleitung bei IPv6 erfolgt analog, die entsprechenden Funktionen heißen *ip6_mforward* statt *ip_forward* usw. Ein von der Sicherungsschicht eintreffendes Paket („Incoming Packet“) wird in der Funktion *ip_input* vorverarbeitet. Es werden Gültigkeitschecks durchgeführt (u. a. Prüfsumme), die

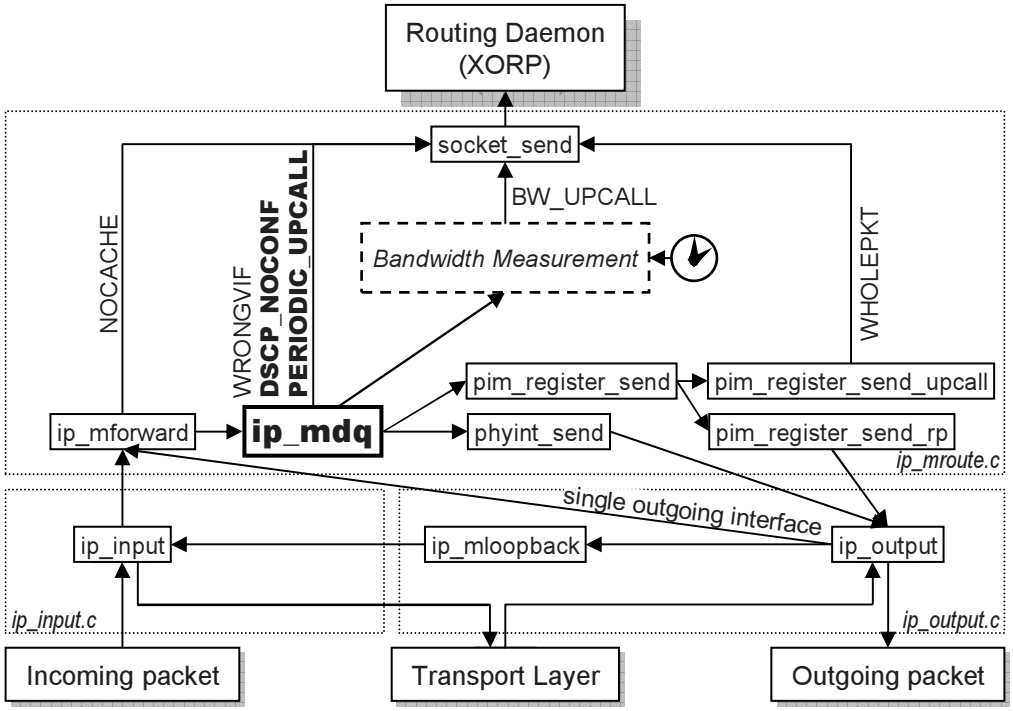


Abbildung 5.1 Vereinfachter Multicastweiterleitungspfad in der Vermittlungsschicht von FreeBSD für IPv4. Die für DSMC erweiterte Funktion *ip_mdq* und die zwei neuen Nachrichten (IGMPMSG_*) sind fett hervorgehoben.

IP-Optionen bearbeitet und, sofern das Paket für den Knoten selbst bestimmt ist, ein ggf. fragmentiertes Paket reassembliert. Es wird dann der Transportschicht übergeben („Transport Layer“). Arbeitet der Knoten als Multicastrouter, d. h. der Mrouter-Socket (s. Abschnitt B.1) wurde von einem Routingdaemon geöffnet, werden zudem alle Multicastpakete an die Funktion *ip_mforward* übergeben. Sie sucht den passenden, d. h. den in Quell- und Gruppenadresse übereinstimmenden, MFC-Eintrag heraus. Findet sie keinen, wird über den Mrouter-Socket der Routingdaemon mit einer NOCACHE-Nachricht vom fehlenden MFC-Eintrag informiert. Er wird dann vom Routingdaemon, ebenfalls über den Mrouter-Socket, neu angelegt. Bis dahin wird ein temporärer MFC-Eintrag angelegt und weitere eintreffende Pakete in einer ihm zugeordneten Warteschlange begrenzt zwischengespeichert. Ansonsten wird das Paket zusammen mit einem Zeiger auf den passenden MFC-Eintrag der Funktion *ip_mdq* übergeben.

Funktion ip_mdq

Die Funktion *ip_mdq* implementiert das Duplizieren an alle aktiven Ausgangsschnittstellen und damit die eigentliche Multicastweiterleitung. Sie ist die einzige der in Abbildung 5.1 wiedergegebenen Funktionen, die im Rahmen von DSMC modifiziert werden muss.

Ist ein Paket über eine andere als die erwartete Eingangsschnittstelle eingetroffen, wird es von der Funktion *ip_mdq* verworfen. Zusätzlich wird ratenlimitiert die Nachricht WRONGVIF[1] generiert. Hierdurch kann der Routingdaemon beispielsweise das Ende eines SPT Switchovers, bei dem sich die Eingangsschnittstelle unterscheidet (Fall (i) in Update_

[1] genau IGMPMSG_WRONGVIF; das Präfix IGMPMSG_ wird im Folgenden bei allen Nachrichtennamen weggelassen

SPTbit(S,G,iif), s. Abschnitt 2.3.2.1), erkennen. Oder die WRONGVIF-Nachricht löst im Routingdaemon einen Assert aus, wenn das Paket über eine aktive Ausgangsschnittstelle eintraf. Die WRONGVIF-Nachricht kann vom Routingdaemon auch deaktiviert werden, separat für jede Schnittstelle. Sollte an einem Link ein anderer Router der Forwarder sein, kann dadurch das ständige Erzeugen unnötiger WRONGVIF-Nachrichten vermieden werden.

ip_mdq ruft für jedes empfangene Paket die Ratenmessung auf. Die zugehörigen Funktionen sind in Abbildung 5.1 als „Bandwidth Measurement“ zusammengefasst. Die Ratenmessung wird vom Routingdaemon konfiguriert und informiert ihn mit einer BW_UPCALL-Nachricht timergesteuert (angedeutete Uhr) bei Über- oder Unterschreiten einer bestimmten Paket- oder Datenrate gemessen über eine konfigurierbare Zeitspanne. Die Ratenmessung wird für das Initiieren eines SPT Switchovers (Policyfunktion SwitchToSptDesired(S,G), s. Abschnitt 2.3.2.1) sowie zum Erkennen nicht mehr aktiver Quellen und damit zum Löschen nicht mehr benötigter MFC-Einträge verwendet.

Neu mit DSMC hinzugekommen ist die Auswertung des NRS-Zustands. *ip_mdq* markiert je nach NRS-Zustand auf einen der in einer routerglobalen (Daten-)Struktur abgelegten DSCPs um. Sie implementiert somit die Codepoint- und Markierungsregeln aus Kapitel 4. Zudem führt sie selbständig den Übergang vom NRS-Zustand Init nach Invalid gemäß Regel **NRS** aus. Zusätzlich unterstützt *ip_mdq* den Routingdaemon in der Umsetzung der Nachrichtenregeln. Sollte der NRS-Zustand ungültig sein, wird der Routingdaemon mit der neuen Nachricht DSCP_NOCONF informiert. Sie löst das Senden einer Triggernachricht durch den Routingdaemon aus. Die zweite neue Nachricht PERIODIC_UPCALL vereinfacht das wiederholte Senden der TriggerSpt-Nachrichten. Sie ist optional, da sie durch eine andere Implementierung der TriggerSpt im Routingdaemon ersetzt werden kann (s. Abschnitt 5.1.3). In der vorliegenden prototypischen Implementierung wird sie jedoch genutzt, was die Implementierung der DSMC-Komponente im Routingdaemon zustandslos hält. Abschnitt B.3 beschreibt die wesentlichen Änderungen an der Funktion *ip_mdq* anhand von Quellcodeausschnitten im Detail.

Je nachdem, ob es sich bei einer Ausgangsschnittstelle um den Register-Tunnel handelt oder nicht, wird das duplizierte Paket der Funktion *pim_register_send* oder der Funktion *phyint_send* übergeben.[2] Hat der Routingdaemon das Einkapseln für den Register-Tunnel dem FreeBSD-Kern übertragen, leitet erstere Funktion das Paket an die vom Routingdaemon konfigurierte Rendezvousstellenadresse weiter (*pim_register_send_rp*). Andernfalls übergibt FreeBSD jedes einzukapselnde Paket per WHOLEPKT-Nachricht dem Routingdaemon zur Weiterleitung an die Rendezvousstelle (*pim_register_send_upcall*). Abschließend erreicht das Paket die Funktion *ip_output*, die u. a. die Prüfsumme einfügt und falls nötig das Paket fragmentiert.[3] Danach ruft sie die zur Schicht-2-Technologie der Ausgangsschnittstelle (Ethernet, GRE-Tunnel, etc.) passende Funktion über den für jede Schnittstelle vermerkten Funktionszeiger *if_output* auf („Outgoing Packet“).

Die Funktion *ip_output* wird auch von der Transportschicht aufgerufen. Multicastpakete werden dann entweder unter Umgehung des MFCs gezielt auf eine vorgegebene Ausgangs-

[2]Die in FreeBSD 5.x und 6.x noch als drittes vorhandene Funktion *encap_send* für einen IPIP-Tunnel, wie ihn CBT nutzt, entfällt mit der kommenden FreeBSD Version 7, da dies flexibler über *if_output* und eine virtuelle Tunnelschnittstelle umgesetzt werden kann.

[3]Der derzeit noch vorgeschaltete multicastspezifische Token Bucket entfällt ebenfalls mit FreeBSD Version 7, da ALTQ zusammen mit IPFILTER dieselbe und mehr Funktionalität flexibler bereitstellt.

Socketoption	Parameter	Bemerkung
MRT_DSCP	mrt_dscp.*	neue Socketoption
	(1) dscp_unavail	DSCP_LE
	(2) dscp_noconf	DSCP_UC
	(3) dscp_signal	DSCP_SIGNAL
	(4) dscpmap_lowprio	niederpriore DSCPs
MRT_DSCP_NOCONF	mrt_dscp_noconf.*	neue Socketoption
	(5) msg_time	$1/R$, Einheit: μs
	(6) msg_suppress	T_S, Einheit: $1/R$
MRT_API_CONFIG	mrt_api_config / MRT_MFC_*	API-Fähigkeit aktiviert:
	FLAGS_KEEP_DSCP	Ummarkieren
	FLAGS_DISABLE_DSCP_NOCONF	Nachricht (A)
	FLAGS_SLOW_DSCP_NOCONF	Unterdrückung T_S
	PERIODIC_UPCALL	Nachricht (B)
MRT_ADD_MFC	mfcctl2.mfc_flags[] / MRT_MFC_FLAGS_*	Kern benötigt Parameter:
	(7) KEEP_DSCP	(1), (4); 0=ummarkieren
	(8) DISABLE_DSCP_NOCONF	(1)–(5); 0=Nachr. senden
	(9) SLOW_DSCP_NOCONF	(1)–(6); 1=verzögern
Nachrichten:	(A) IGMPMSG_DSCP_NOCONF	bei (8)=0; Trigger/-UC/-Rpt
	(B) IGMPMSG_PERIODIC_UPCALL	bei (9)=1; TriggerSwitched

Tabelle 5.1 Für DSMC bei IPv4 neu definierte Socketoptionen und Parameter bzw. Flags/API-Fähigkeiten. (6), (9) und (B) sind optional.

schnittstelle weitergeleitet (Pfeil „single outgoing interface“) oder das Paket wird über die Funktion *ip_mloopback* in die reguläre Multicastweiterleitung eingespeist.

5.1.2 Erweiterte Multicastrouting-API

Die Implementierung des Verfahrens DSMC erweitert die Multicastrouting-API von FreeBSD nur geringfügig. Eine kurze Einführung in die Multicastrouting-API gibt Abschnitt B.1. Die Implementierung der API-Erweiterung wird in Abschnitt B.2 anhand relevanter Quellcodeausschnitte der Headerdatei *ip_mroute.h* ausführlich beschrieben. Im Folgenden sollen nur die neuen API-Fähigkeiten in ihrer Funktion erläutert werden.

Tabelle 5.1 fasst die API-Erweiterung von IPv4, Tabelle 5.2 die von IPv6 zusammen. Es werden je Adressfamilie zwei neue Socketoptionen mit zwei neuen Parameterstrukturen definiert. Für IPv4 werden mit der Socketoption MRT_DSCP routerglobal die drei DSCPs konfiguriert, auf die in den NRS-Zuständen Remark (DSCP_LE), Invalid (DSCP_UC) und Init (DSCP_SIGNAL) ummarkiert werden soll. Verzögerung und Ratenlimitierung der Nachrichten IGMPMSG_DSCP_NOCONF an den Routingdaemon werden mit der zweiten neuen Socketoption MRT_DSCP_NOCONF, ebenfalls routerglobal, gesetzt. Dies erlaubt dem Routingdaemon bzw. seiner DSMC-Komponente auf einfache Weise, die von ihr an den Ressourcenmanager zu sendenden Trigger-Nachrichten in ihrer Rate R gemäß Gleichung 4.4 zu begrenzen und die anfängliche Unterdrückungszeit T_S, je nach Multicastroutingprotokoll gemäß Gleichung 4.5, 4.6 oder 4.7, einzuhalten, ohne dazu selbst Timer und damit eigenen Zustand zu benötigen.

Diese zwei neuen Socketoptionen werden mit der Socketoption MRT_API_CONFIG und folgenden API-Fähigkeiten stückweise aktiviert:

Socketoption	Parameter	Bemerkung
MRT6_DSCP	mrt6_dscp.*	neue Socketoption
	(1) dscp_unavail	DSCP_LE, aktiviert (7)
	(2) dscp_noconf	DSCP_UC, aktiv. (8) & (9)
	(3) dscp_signal	DSCP_SIGNAL
	(4) dscpmap_lowprio	niederpriore DSCPs
MRT6_DSCP_NOCONF	mrt6_dscp_noconf.*	neue Socketoption
	(5) msg_time	$1/R$, Einheit: μs
	(6) msg_suppress	T_S, Einheit: $1/R$
MRT6_ADD_MFC	mf6cctl.*	Kern benötigt Parameter:
	(7) mf6cc_keep_dscp	(1), (4). 0=ummarkieren
	(8) mf6cc_disable_dscp_noconf	(1)–(5). 0=Nachr. senden
	(9) mf6cc_slow_dscp_noconf	(1)–(6). 1=verzögern
Nachrichten:	(A) MRT6MSG_DSCP_NOCONF	bei (8)=0. Trigger/-UC/-Rpt
	(B) MRT6MSG_PERIODIC_UPCALL	bei (9)=1. TriggerSwitched

Tabelle 5.2 Für DSMC bei IPv6 neu definierte Socketoptionen und Parameter bzw. Flags. Ein gültiger *dscp_unavail* aktiviert das Ummarkieren mit vereinfachtem NRS-Zustand (nur Keep und Remark), zusätzlich ein gültiger *dscp_noconf* alle vier NRS-Zustände und Nachrichten. (6), (9) und (B) sind optional.

- **MRT_MFC_FLAGS_KEEP_DSCP** aktiviert den für verteiltes Dienstgütemanagement anwendbaren vereinfachten NRS-Zustand mit nur zwei Zuständen Keep und Remark (s. Abschnitt 4.2.1). Im Falle des Zustands Remark wird auf den mittels MRT_DSCP konfigurierten DSCP *mrt_dscp.dscp_unavail* ummarkiert. Das Bitfeld *mrt_dscp.dscp_lowprio* bestimmt, welche DSCPs vom NRS-Zustand ausgenommen und niemals ummarkiert werden. Diese Fähigkeit aktiviert somit die erste neue Socketoption MRT_DSCP zur Konfiguration der DSCPs. Die Felder *dscp_noconf* und *dscp_signal* in der Struktur *mrt_dscp* werden noch nicht benötigt und von FreeBSD ignoriert.

  Der NRS-Zustand wird mit der Socketoption MRT_ADD_MFC und den dabei in *mfcctl2.mfc_flags* übergebenen Flags konfiguriert. Die API-Fähigkeit MRT_MFC_FLAGS_KEEP_DSCP repräsentiert zugleich eines der je Ausgangsschnittstelle konfigurierbaren Flags. Der NRS-Zustand Keep entspricht einem gesetzten Flag, der DSCP wird beim Weiterleiten beibehalten. Andernfalls ist die Ausgangsschnittstelle im Zustand Remark und es wird auf den routerglobal konfigurierten DSCP ummarkiert. Dies ist zugleich der Ausgangszustand nach dem Aktivieren einer Ausgangsschnittstelle.

- **MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF** aktiviert den vollen NRS-Zustand mit allen vier Zuständen und die Nachrichten IGMPMSG_DSCP_NOCONF. Sie werden gemäß *mrt_dscp_noconf.msg_time* ratenlimitiert. Die zum Setzen des Werts benötigte zweite neue Socketoption MRT_DSCP_NOCONF wird entsprechend mit aktiviert. Das in der Struktur *mrt_dscp_noconf* nicht genutzte Feld *msg_suppress* wird ignoriert. Das Flag repräsentiert zusammen mit dem Flag MRT_MFC_FLAGS_KEEP_DSCP den NRS-Zustand gemäß Tabelle 5.3.

- **MRT_MFC_FLAGS_SLOW_DSCP_NOCONF** aktiviert das anfängliche Unterdrücken der Nachricht IGMPMSG_DSCP_NOCONF im NRS-Zustand Invalid. Ist an

NRS-Zustand	(keep_dscp, disable_dscp_noconf)
Init	(0, 0)
Invalid	(1, 0)
Remark	(0, 1)
Keep	(1, 1)

Tabelle 5.3 Kodierung des NRS-Zustands.

der betreffenden Ausgangsschnittstelle das Flag gesetzt, werden die ersten *mrt_dscp_noconf.msg_suppress* Nachrichten unterdrückt. Befindet sich mehr als eine Ausgangsschnittstelle im Zustand Invalid, kann die Unterdrückungszeit auch größer ausfallen, da mit jeder neu aktivierten Ausgangsschnittstelle das Zählen erneut beginnt (s. Abschnitt B.3).

- **MRT_MFC_PERIODIC_UPCALL** aktiviert die Nachricht IGMPMSG_PERIODIC_UPCALL. Sie wird gesendet, wenn das Flag MRT_MFC_FLAGS_SLOW_DSCP_NOCONF an der *Eingangs*schnittstelle gesetzt ist. Die Rate ist dabei dieselbe wie die der Nachrichten IGMPMSG_DSCP_NOCONF.

Die API-Fähigkeiten bauen in der angegebenen Reihenfolge aufeinander auf. Eine Aktivierung setzt somit die Aktivierung aller jeweils zuvor genannten Fähigkeiten voraus. Die beiden letzten API-Fähigkeiten sind optional und können durch eine geänderte Implementierung des Routingdaemons ersetzt werden.

Die Multicastrouting-API von IPv6 kennt anders als die von IPv4 keine API-Fähigkeiten. Eine Aktivierung der mit DSMC neu eingeführten Funktionen erfolgt daher mit der Socketoption MRT6_SET_DSCP durch Setzen gültiger Werte für *mrt6_dscp.dscp_unavail* und *mrt6_dscp.dscp_noconf*. Ist *mrt6_dscp.dscp_unavail* ein gültiger DSCP (≤ 63), wird der vereinfachte NRS-Zustand mit zwei Zuständen aktiviert. Enthält auch *mrt6_dscp.dscp_noconf* einen gültigen DSCP, ist der volle NRS-Zustand verfügbar und es werden Nachrichten MRT6MSG_DSCP_NOCONF generiert. Eine separate Aktivierung der beiden optionalen Fähigkeiten unabhängig vom vollen NRS-Zustand ist nicht möglich. Ansonsten entspricht die Funktionalität der von IPv4.

5.1.3 Optionale API-Fähigkeiten

Hier soll kurz mit Vor- und Nachteilen erläutert werden, wie die beiden optionalen API-Fähigkeiten durch eine geänderte Implementierung des Routingdaemons ersetzt werden können.

MRT_MFC_FLAGS_SLOW_DSCP_NOCONF

Die anfängliche Unterdrückung der TriggerUC kann auch allein in der DSMC-Komponente im Routingdaemon implementiert werden. Sie muss dazu die ersten empfangenen Nachrichten vom Kern ignorieren und erst nach gewisser Zeit oder nach einer bestimmten Anzahl erhaltener Nachrichten beginnen, für jede Nachricht einen TriggerUC zu generieren. Das Flag MRT_MFC_FLAGS_SLOW_DSCP_NOCONF und die Fähigkeit zur Unterdrückung einiger Nachrichten nach der beim Wechsel von Init nach Invalid immer erzeugten ersten Nachricht kann entfallen. Parameter (6) und Flag (9) entfallen. Nachteilig hieran ist, dass

unnötig Nachrichten vom FreeBSD-Kern generiert werden. Vorteil ist, dass der Routingdaemon das Flag im Kern nicht aktualisieren muss, sich also einen Systemaufruf setsockopt mit der Socketoption MRT_ADD_MFC spart.

MRT_MFC_PERIODIC_UPCALL

Beim SPT Switchover von PIM-SM löst das erste Paket, das bei geänderter Eingangsschnittstelle über das RPF_interface(S) eintrifft, eine Nachricht IGMPMSG_WRONGVIF an den Routingdaemon aus, so dass dieser das SPTbit(S,G) setzen kann. Da dies auch zum Setzen des Sbit(S,G) führt (Fall (i), s. Abschnitt 4.3.3), wird der erste TriggerSwitched durch die Nachricht IGMPMSG_WRONGVIF ausgelöst. Der erste TriggerSwitched in Fall (ii) wird durch einen Assert, d. h. durch eine empfangene PIM-Nachricht ausgelöst. IGMPMSG_PERIODIC_UPCALL dient somit lediglich der ratenlimitierten *Wiederholung* des TriggerSwitched, bis das Sbit zurückgesetzt wird.

Die API-Fähigkeit MRT_MFC_PERIODIC_UPCALL lässt sich einsparen, wenn der Routingdaemon selbst die ratenlimitierte Wiederholung der TriggerSwitched bei gesetztem Sbit übernimmt. Hierzu kann der TriggerSwitched beispielsweise als SNMP Trap ausgeführt werden. Die Wiederholung durch die DSMC-Komponente wäre zwar auch möglich, würde wegen der benötigten Timer allerdings Zustand in die ansonsten zustandslose DSMC-Komponente einführen.

Es sei angemerkt, dass IGMPMSG_WRONGVIF solange ratenlimitiert (alle 3 Sekunden) wiederholt wird, bis das Flag MRT_MFC_FLAGS_DISABLE_WRONGVIF vom Routingdaemon gesetzt wird. Es wäre somit prinzipiell möglich, IGMPMSG_PERIODIC_UPCALL einfach durch IGMPMSG_WRONGVIF zu ersetzen. Der Routingdaemon würde FreeBSD die Nachricht solange weiter wiederholen lassen, bis er den TriggerSwitchedAck empfangen hat. Dies würde jedoch den schreibenden Zugriff der DSMC-Komponenten auf ein von PIM-SM genutztes Flag bedeuten, d. h. das Ändern von Zustand, den auch das PIM-SM-Routing intern nutzt.

5.2 XORP

XORP ist eine in C++ geschriebene unter der GPL (GNU General Public License) stehende Implementierung eines Routingdaemons. Ziel ist es, nicht nur eine Low-End-Routerplattform zu schaffen, sondern vor allem eine Basis für die weitere Forschung bereitzustellen. XORP befindet sich derzeit noch im Aufbau.[4] Während bereits alle wichtigen Unicastroutingprotokolle verfügbar sind, ist noch immer PIM-SM das einzige implementierte Multicastroutingprotokoll. Die Wahl für die Implementierung des DSMC-Prototypen fiel dennoch auf XORP, da es die einzige frei verfügbare PIM-SM-Implementierung nach dem aktuellen Standard [66] ist und XORP gleichermaßen IPv4 wie IPv6 unterstützt. Nach einem Überblick über XORP und den darin für DSMC modifizierten Teilen werden anschließend die Implementierung der DSMC-Komponente und die Erweiterung von PIM-SM zum Setzen des Sbits detailliert beschrieben.

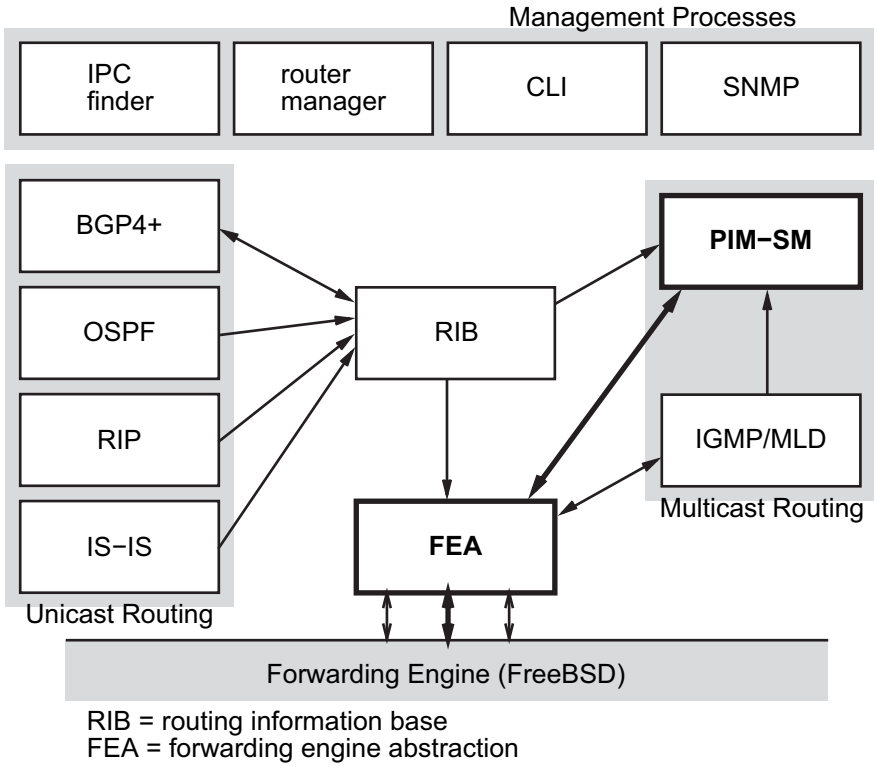


Abbildung 5.2 In XORP implementiert Routingprotokolle, Managementebene und die zentralen Komponenten RIB und FEA als eigenständige in separaten Prozessen laufende XORP-Module. Die zwei für DSMC modifizierten Module sind fett hervorgehoben (nach [193]).

5.2.1 Überblick

In XORP existiert für jedes Routingprotokoll ein eigenes XORP-Modul (s. Abbildung 5.2). Hinzu kommen weitere Module mit XORP-Infrastrukturkomponenten wie dem Router Manager. Die XORP-Module kommunizieren untereinander mit Hilfe der XORP-eigenen Interprozesskommunikation, wodurch jedes Modul in einem eigenen Prozess laufen kann (aber nicht muss). Jedes Modul registriert sich dazu beim Finder, der zusammen mit dem Router Manager in einem Prozess läuft. Der Router Manager ist das erste gestartete Modul. Er startet alle übrigen Module und konfiguriert sie. Zudem bietet er über das Command Line Interface die Möglichkeit zur späteren Umkonfiguration und Statusabfrage. Funktionsaufrufe zwischen XORP-Modulen werden mit so genannten XRLs (XORP Resource Locator) in einer an URLs angelehnten Syntax beschrieben. Der Finder löst den in der XRL genannten Modulnamen in IP-Adresse und UDP-Port des Moduls auf, an den der Funktionsaufruf gesandt werden muss. Ein XORP-Router kann dadurch auch verteilt auf mehreren physikalischen Netzwerkknoten laufen.

XORP abstrahiert von der Weiterleitungsebene („Forwarding Engine“). Alles, was von der jeweiligen Plattform wie beispielsweise FreeBSD abhängt, wird in der FEA (Forwarding Engine Abstraction) gekapselt. Sie enthält auch die MFEA (Multicast FEA), die die

[4]Die Entwicklung an XORP begann Ende 2000, die erste öffentlich verfügbare Alpha-Release erschien zwei Jahre später. Mittlerweile erscheinen in unregelmäßigen Abständen neue Releases, die nicht nur Bugs fixen, sondern vor allem weitere Routingprotokolle, Prozessorarchitekturen und Betriebssysteme unterstützen.

Multicastrouting-API kapselt und den anderen XORP-Modulen gleichwertige Funktionen über ihre XRLs bereitstellt. Das RIB-Modul sammelt alle über die Unicastroutingprotokolle gelernten Routen und generiert daraus die Weiterleitungsinformationen, die es über die FEA in die Weiterleitungsebene (in deren FIB) schreibt. Die MRIB, auf deren Basis die Multicastroutingprotokolle die RPF-Berechnung durchführen, ist Teil des RIB-Moduls. Multicastroutingprotokolle – derzeit nur PIM-SM – registrieren sich bei der MRIB, um bei Routingänderungen von der MRIB informiert zu werden. Sie können so, falls nötig, ihren Multicastroutingzustand entsprechend der Änderungen anpassen. Ebenso registrieren sich Multicastroutingprotokolle bei den Gruppenmanagementprotokollen (IGMP/MLD), damit sie von diesen über lokale Empfänger informiert werden.

Erweiterungen für DSMC

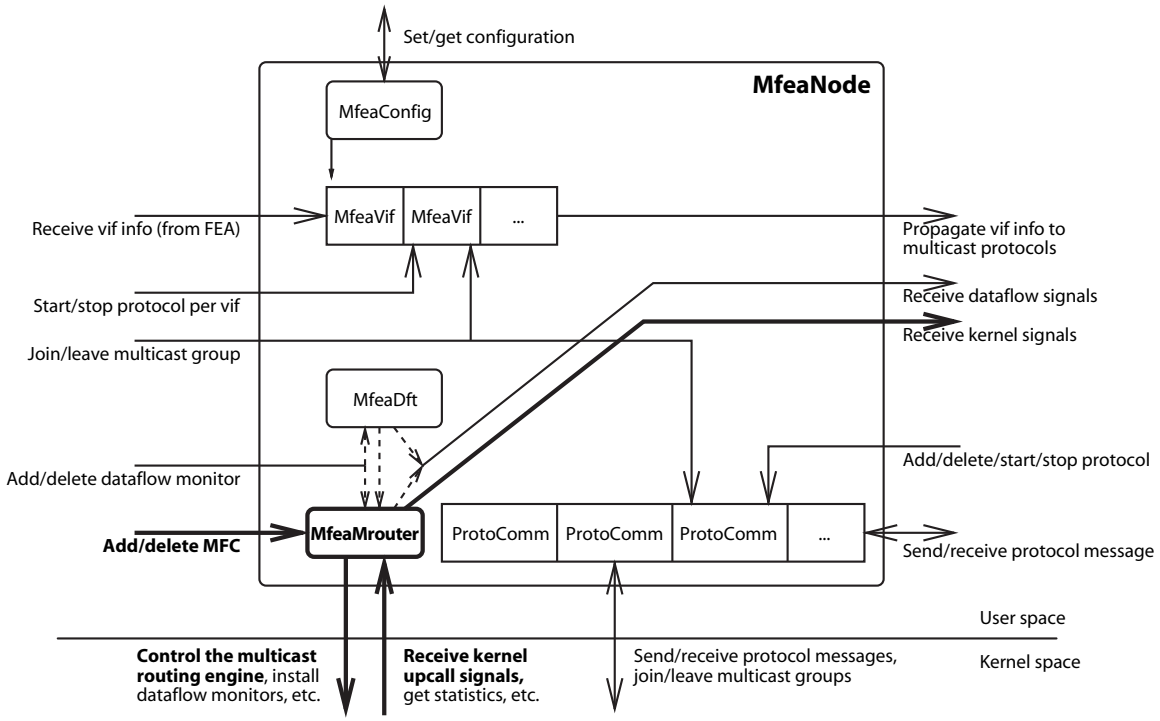


Abbildung 5.3 Die Klasse MfeaNode fasst die Klassen der MFEA (Teil des XORP-Moduls FEA) zusammen. Die Klasse MfeaMrouter kapselt die plattformabhängige Multicastrouting-API der Weiterleitungsebene. Für DSMC relevante Teile sind fett hervorgehoben (nach [193]).

Die DSMC-Komponente wurde nicht als eigenständiges XORP-Modul ausgeführt, sondern in das Modul des derzeit einzigen Multicastroutingprotokolls PIM-SM integriert. Wenn ein weiteres Multicastroutingprotokoll in XORP verfügbar wird, ist das Auslagern in ein eigenes XORP-Modul sinnvoll. Durch die Integration konnte die aufwendige XRL-basierte Interprozesskommunikation reduziert werden.

Die Implementierung von DSMC unter XORP erfordert die Modifikationen von nur zwei XORP-Modulen, PIM-SM und FEA. Bei letzterem beschränken sich die Modifikationen auf die darin enthaltene MFEA. Abbildung 5.3 zeigt den Aufbau der MFEA bzw. der sie

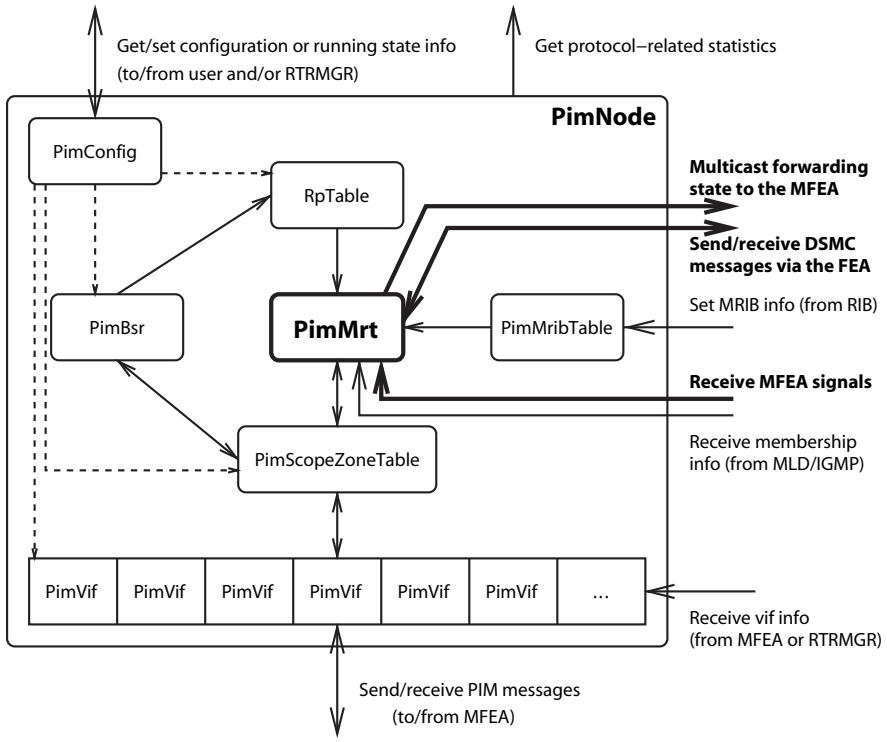


Abbildung 5.4 Die Klasse PimNode fasst die Klassen des Protokolls PIM-SM zusammen. Die PimMrt enthält die Multicastroutingtabellen und eine Kopie des MFCs aus der Weiterleitungsebene. Für DSMC relevante Teile sind fett hervorgehoben (nach [193]).

repräsentierenden Klasse MfeaNode. Die Kapselung der Multicastrouting-API erfolgt in der Klasse MfeaMrouter. Diese muss entsprechend der in Abschnitt 5.1.2 beschriebenen API-Erweiterung von FreeBSD angepasst werden. Die insgesamt nötigen Änderungen sind jedoch minimal. Es müssen die neuen API-Fähigkeiten aktiviert (nur IPv4), das Konfigurieren der routerglobalen DSCPs (SET_DSCP) und des Timings (SET_DSCP_NOCONF) ermöglicht und die neuen Nachrichtentypen (DSCP_NOCONF, PERIODIC_UPCALL) akzeptiert werden können. Hinzu kommt die Bearbeitung der um Parameter für die neuen Flags erweiterten XRL *mfea/0.2/add_mfc* (s. Abbildung 5.6). Die Änderungen an der MFEA werden hier nicht weiter dargestellt. Alle übrigen Teile der MFEA sind für DSMC nicht relevant und bleiben unverändert.

Den Aufbau der Klasse PimNode zeigt Abbildung 5.4. Sie ist wie MfeaNode im Wesentlichen ein Container für eine Reihe anderer Klassen. Diese implementieren das Protokoll PIM-SM (Klassen PimMrt und PimVif), den Bootstrap-Router-Mechanismus (Klasse PimBsr) oder wickeln die Kommunikation mit den anderen relevanten XORP-Modulen IGMP/MLD, RIB, FEA sowie dem Router Manager ab. Die zentrale Klasse ist PimMrt. Sie enthält nicht nur alle Multicastroutingtabellen und eine Kopie des MFCs, sondern implementiert letztlich zusammen mit der eine Netzwerkschnittstelle repräsentierenden Klasse PimVif das eigentliche Multicastroutingprotokoll PIM-SM.

Bei jeder Änderung in der Kopie des MFCs (in PimMrt) wird diese über die MFEA in den eigentlichen MFC in der Weiterleitungsebene (FreeBSD) geschrieben, so dass beide MFCs konsistent bleiben. Hier sind für DSMC Ergänzungen nötig, um die neuen Flags,

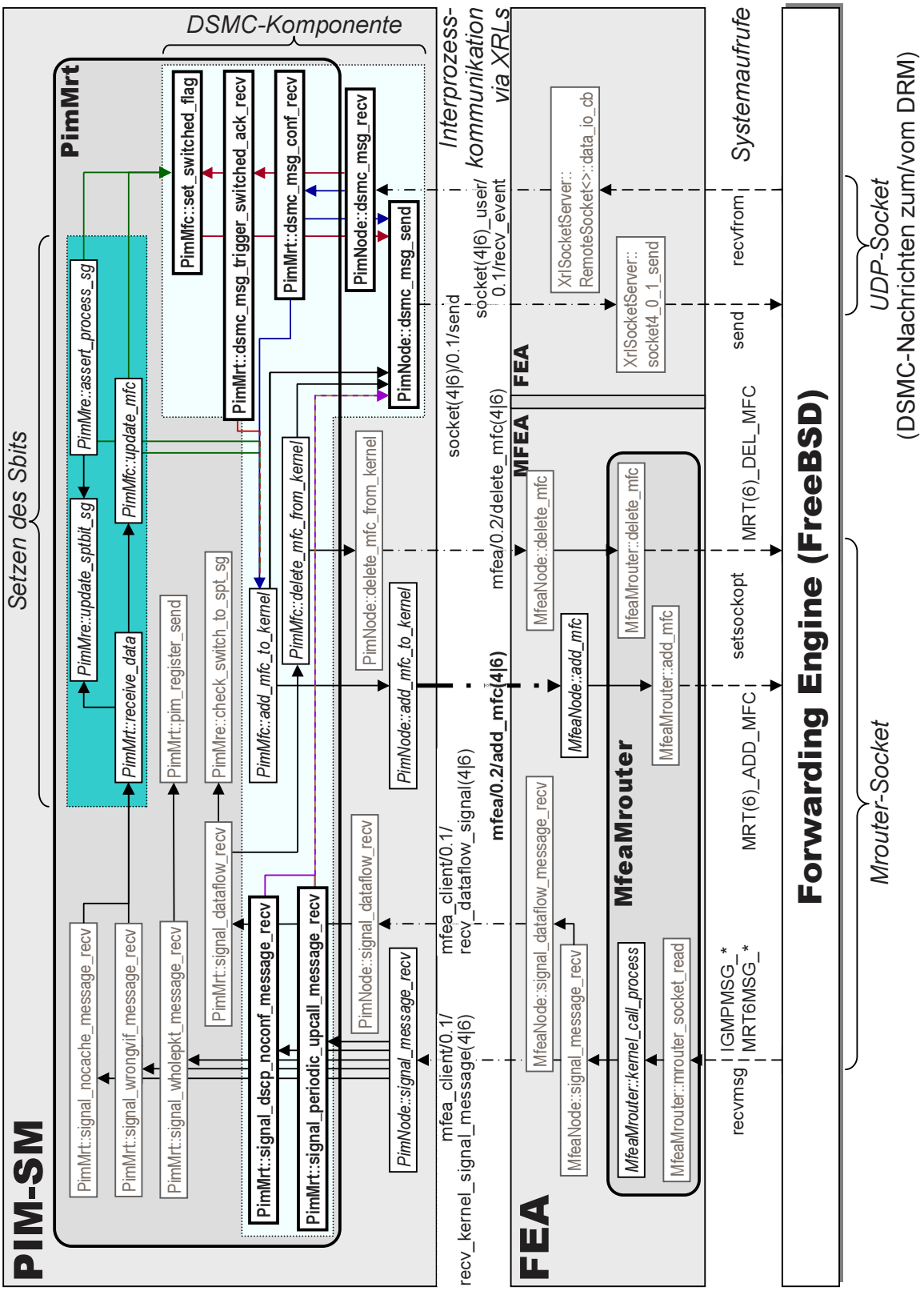


Abbildung 5.5 Für DSMC relevanter Ausschnitt aus den XORP-Modulen FEA und PIM-SM (Erläuterung s. Text). Unveränderte Funktionen in grauer Schrift.

```
  8 interface mfea/0.2 {
142         add_mfc4 ? xrl_sender_name:txt    \
143                         & source_address:ipv4 & group_address:ipv4 \
144                         & iif_vif_index:u32    \
145                         & oiflist:binary    \
146                         & oiflist_disable_wrongvif:binary  \
147                         & oiflist_keep_dscp:binary   \
148                         & oiflist_disable_dscp_noconf:binary  \
149                         & oiflist_slow_dscp_noconf:binary  \
150                         & max_vifs_oiflist:u32    \
151                         & rp_address:ipv4
152         add_mfc6 ? xrl_sender_name:txt    \
153                         & source_address:ipv6 & group_address:ipv6 \
154                         & iif_vif_index:u32    \
155                         & oiflist:binary    \
156                         & oiflist_disable_wrongvif:binary  \
157                         & oiflist_keep_dscp:binary   \
158                         & oiflist_disable_dscp_noconf:binary  \
159                         & oiflist_slow_dscp_noconf:binary    \
160                         & max_vifs_oiflist:u32    \
161                         & rp_address:ipv6
322 }
```

Abbildung 5.6 Erweiterte XRLs *mfea/0.2/add_mfc4* und *mfea/0.2/add_mfc6*, um die mit DSMC neu eingeführten Flags dscp_noconf, disable_dscp_noconf und slow_dscp_noconf im MFC setzen zu können (Ausschnitt aus Datei xrl/interfaces/mfea.xif).

die den NRS-Zustand (keep_dscp[5], disable_dscp_noconf) und das DRbit bzw. Sbit (slow_dscp_noconf) repräsentieren, speichern und in den MFC schreiben zu können. Dies betrifft auch die XRL zum Hinzufügen und Ändern eines MFC-Eintrags *mfea/0.2/add_mfc*, wie bereits oben bei der MFEA beschrieben. Die mit DSMC neu eingeführten Nachrichten DSCP_NOCONF und PERIODIC_UPCALL erfordern dagegen keine Änderung an einer der XRLs, da es sich lediglich um zwei neue Nachrichtentypen handelt und somit keine weiteren Parameter zu einer XRL hinzuzufügen sind.

Neben der Kommunikation zwischen PimMrt und der Weiterleitungsebene ist die Kommunikation mit dem Ressourcenmanager nötig. Hierfür wurde auf bereits existierende Infrastruktur in Form des Socket-Servers zurückgegriffen. Er ist Teil der FEA und kapselt darin die Kommunikation über Netzwerksockets. Als Transportprotokolle stehen TCP und UDP zur Verfügung. Die prototypische Implementierung nutzt UDP.

Nach diesem kurzen Überblick sollen die genannten Änderungen nachfolgend etwas genauer dargestellt werden. Sie zeigen, wie die erweiterte Multicastrouting-API von FreeBSD genutzt wird und verdeutlichen die routerseitige Umsetzung von DSMC.

```
ostringstream message;
message << msg_type << ' '
        << source.str() << ' ' << group.str() << ' ' << oif_index << ' '
        << nrs_state << ' ' << stop_flag << ' ' << ack_flag << '\n';
// stringstreams have no iterators, so first convert into a string
string tmp(message.str());
vector<uint8_t> pdata(tmp.begin(), tmp.end());
```

Abbildung 5.7 DSMC-Nachrichten werden als einzeiliger ASCII-Text übertragen, in dem Leerzeichen die einzelnen Felder trennen (Ausschnitt aus Funktion *PimNode::dsmc_msg_send*).

5.2.2 DSMC-Komponente

Abbildung 5.5 zeigt den für DSMC relevanten Ausschnitt aus den beiden XORP-Modulen PIM-SM und FEA. Die Funktionen aus PIM-SMs zentraler Klasse PimMrt bzw. die in ihr gekapselten Klassen und die Funktionen aus der Klasse MfeaMrouter sind jeweils eingerahmt. Hier befinden sich die wichtigsten Änderungen. Die Funktionen außerhalb repräsentieren lediglich Schnittstellen zu den XRLs. Die Interprozesskommunikation mit XRLs zwischen diesen Schnittstellenfunktionen ist durch Strichpunktpfeile angedeutet. Tatsächlich ist jeder XRL-Aufruf asynchron und läuft über Warteschlangen und separate Schnittstellenklassen, welche zum Teil automatisch mittels Skripten aus einer XRL-Schnittstellenbeschreibung wie in Abbildung 5.6 generiert werden. Zu diesem Zweck bilden die Klassen PimNode, die per Skript generierte Klasse XrlPimTargetBase und einige weitere Klassen die Basisklassen der Klasse XrlPimNode, die das eigentliche XORP-Modul PIM-SM darstellt. Entsprechendes gilt für die MFEA. Diese sind in der Abbildung nicht wiedergegeben.

Abbildung 5.5 gliedert sich grob in vier Bereiche. Auf der linken Seite mit aufwärts gerichteten Pfeilen befinden sich Funktionen, die die von der Weiterleitungsebene generierten Nachrichten bearbeiten (NOCACHE, WRONGVIF, WHOLEPKT, DSCP_NOCONF, PERIODIC_UPCALL, BW_UPCALL). Mittig mit abwärts gerichteten Pfeilen liegen Funktionen zum Anlegen/Ändern (add_mfc) und Löschen (delete_mfc) von MFC-Einträgen. Auf der rechten Seite mit Pfeilen in beiden Richtungen sind Funktionen zur Kommunikation mit dem Ressourcenmanager dargestellt. Die Funktionen oben in der Mitte bilden den Übergang zum Rest der PIM-SM-Implementierung mit dem eigentlichen Routingprotokoll PIM-SM, d. h. mit der Verarbeitung von PIM-Nachrichten. Zum Beispiel wird *PimMre::assert_process_sg* beim Empfang einer quellenspezifischen Assert-Nachricht aufgerufen. Hier befindet sich auch die im folgenden Abschnitt 5.2.3 beschriebene Modifikation von PIM-SM zum Setzen des Sbits. Da DSMC nicht ins Multicastrouting selbst eingreift sondern nur auf Ebene des MFCs arbeitet, sind keine Änderungen am Rest der PIM-SM-Implementierung nötig und dieser in Abbildung 5.5 nicht wiedergegeben.

Für DSMC modifizierte Funktionen sind mit kursiver Schrift, neu eingeführte Funktionen in fetter Schrift und fett umrandet und unveränderte Funktionen in grauer Schrift grau umrandet dargestellt. Die eigentliche DSMC-Komponente ist hell hinterlegt. Ohne die in der Abbildung nicht wiedergegebenen Funktionen zur XRL-basierten Kommunikation und

[5]Zur besseren Lesbarkeit werden statt der vollständigen Bezeichnung MRT_MFC_FLAGS_KEEP_DSCP (IPv4) bzw. mf6cc_keep_dscp (IPv6) im Folgenden gekürzte Bezeichnungen verwendet.

inklusive der Erweiterung für PIM-SM besteht sie aus sieben neuen und zwei modifizierten Funktionen. Fünf neue Funktionen gehören zur Klasse PimMrt: Je eine pro Nachrichtentyp vom Kern (DSCP_NOCONF und PERIODIC_UPCALL) und DRM (Conf und TriggerSwitchedAck) sowie eine zum Setzen des Sbits und Generieren des TriggerSwitched. Hinzu kommen zwei neue Funktionen in der Klasse PimNode, die das Parsen bzw. das Serialisieren der mit dem DRM ausgetauschten DSMC-Nachrichten übernehmen. Die zwei bereits vorhandenen und nur noch für DSMC erweiterten Funktionen gehören zur Klasse PimMfc. Ein Objekt der Klasse PimMfc repräsentiert einen MFC-Eintrag, der sich mit seiner Methode *PimMfc::add_mfc_to_kernel* in den MFC der Weiterleitungsebene schreibt und mittels *PimMfc::delete_mfc_from_kernel* bei seinem Löschen auch wieder selbst entfernt. Erstere der beiden Funktionen ist so modifiziert, dass sie auch die zusätzlichen Flags in den MFC in der Weiterleitungsebene schreibt. Beide generieren zudem beim Inaktivieren einer Ausgangsschnittstelle – das Löschen des gesamten MFC-Eintrags impliziert das Inaktivieren aller Ausgangsschnittstellen – einen TriggerStop.

Die XRL zum Setzen eines MFC-Eintrags *mfea/0.2/add_mfc4* bzw. *mfea/0.2/add_mfc6* (fetter Strichpunktpfeil in Abbildung 5.5) wurde um Felder für die neuen Flags ergänzt (s. Abbildung 5.6, Zeilen 147–149 und 157–159). Entsprechend der erweiterten XRL waren auch die sie bearbeitenden Funktionen *PimNode::add_mfc_to_kernel* und *MfeaNode::add_mfc* aus den Klassen PimNode bzw. MfeaNode anzupassen. Weitere XRLs mussten für die Arbeit der DSMC-Komponente nicht modifiziert werden. Lediglich die Funktion *PimNode::signal_message_recv* muss die neuen mit der XRL *mfea_client/0.1/recv_kernel_signal_message* eintreffenden Nachrichtentypen kennen, damit sie die richtigen Funktionen der DSMC-Komponente aufrufen kann.

Die prototypische Implementierung überträgt eine DSMC-Nachricht in Form eines einzeiligen ASCII-Texts, in dem die Felder mit Leerzeichen voneinander getrennt sind (s. Abbildung 5.7). Jede Nachricht wird in einem eigenen UDP-Paket übertragen. Alle Werte werden als Zahlen dargestellt ausgenommen die Quell- und Gruppenadresse, die in ihrer üblichen Schreibweise mit Punkten (IPv4) bzw. Doppelpunkten (IPv6) dargestellt werden. Das Serialisieren in die über UDP zu übertragende Bytefolge *pdata* erfolgt mittels Stringstreams.

Die DSMC-Komponente mit ihren insgesamt neun Funktionen implementiert die im letzten Kapitel 4 aufgestellten Regeln, ausgenommen die durch die Weiterleitungsebene (FreeBSD) implementierten Codepoint- und Markierungsregeln. Im einzelnen führen die Funktionen der DSMC-Komponente folgende Aktionen durch:

- **PimNode:dsmc_msg_recv** parst die vom DRM eintreffende DSMC-Nachricht und ruft bei einem Conf die Funktion *PimMrt::dsmc_msg_conf_recv*, bei einem TriggerSwitchedAck die Funktion *PimMrt::dsmc_msg_trigger_switched_ack_recv* mit den in der Nachricht enthaltenen Werten auf.
- **PimNode:dsmc_msg_send** erzeugt eine DSMC-Nachricht des angegebenen Typs mit den übergebenen Werten für Quelle, Gruppe, Ausgangsschnittstelle (nicht bei TriggerSwitched) sowie bei Conf zusätzlich ACKbit, NRSstate und DRbit.
- **PimMfc::add_mfc_to_kernel** prüft für alle Ausgänge Oif, ob das Activebit(S,G,Oif) = FALSE und NRSstate(S,G,Oif) $\neq$ Init ist. In diesem Fall wird NRSstate(S,G,Oif) $\leftarrow$ Init gesetzt und ein TriggerStop generiert (Aufruf von *PimNode:dsmc_msg_send*). Sind alle Ausgänge inaktiv, wird zusätzlich das Sbit(S,G) $\leftarrow$ FALSE zurückgesetzt.

- **PimMfc::delete_mfc_from_kernel** geht wie *PimMfc::add_mfc_to_kernel* vor, jedoch entfällt der Test bzgl. zurückgesetztem Activebit sowie die Behandlung des Sbits, da das Löschen eines MFC-Eintrags dem Zurücksetzen aller Activebits sowie des Sbits entspricht.

- **PimMrt::signal_dscp_noconf_message_recv** prüft, ob NRSstate(S,G,Oif) = Init ist. Wenn ja, wird in NRSstate(S,G,Oif) ← Invalid gewechselt. Ist Oif in pim_include(*,G) oder pim_include(S,G) enthalten oder wurde Oif als DSMC-Grenzschnittstelle konfiguriert, wird DRbit(S,G,Oif) ← TRUE, andernfalls DRbit(S,G,Oif) ← FALSE gesetzt. Ist NRSstate(S,G,Oif) ≠ Init, wird der bereits beim Wechsel nach Invalid konfigurierte Wert des DRbit(S,G,Oif) ausgelesen. Ist DRbit(S,G,Oif) = FALSE wird ein TriggerUC generiert, andernfalls bei SPTbit(S,G) = TRUE ein Trigger, sonst ein TriggerRpt (Aufruf von *PimNode:dsmc_msg_send*). Der TriggerUC wird nicht beim ersten Mal generiert, wenn NRSstate(S,G,Oif) = Init war und nach Invalid gewechselt wurde. Hier wird stattdessen das Flag slow_dscp_noconf gesetzt, um Nachrichten vom Kern für einige Zeit zu unterdrücken. Ansonsten wird für jede Nachricht vom Kern ein Trigger generiert, die Ratenlimitierung des Kerns bestimmt die Wiederholungsrate der Trigger.

- **PimMrt::dsmc_msg_conf_recv** setzt im MFCentry(S,G,Oif) NRSstate(S,G,Oif) und DRbit(S,G,Oif) gemäß der Werte aus der Conf-Nachricht und aktualisiert den MFCentry(S,G,Oif) in der Weiterleitungsebene (Aufruf von *PimMfc::add_mfc_to_kernel*). Falls vom Conf angefordert, wird die erfolgte Konfiguration mit ConfReply OK bestätigt (Aufruf von *PimMrt::dsmc_msg_conf_send*). Ist dagegen Activebit(S,G,Oif) = FALSE, wird in jedem Fall ein ConfReply Fail mit den Werten aus dem Conf generiert, um den nicht vorhandenen Weiterleitungszustand anzuzeigen.

- **PimMrt::signal_periodic_upcall_message_recv** generiert für jede Nachricht vom Kern einen TriggerSwitched (Aufruf von *PimNode::dsmc_msg_send*).

- **PimMrt::dsmc_msg_trigger_switched_ack_recv** setzt das Sbit(S,G) ← FALSE zurück (Aufruf von *PimMfc::set_switched_flag*) und beendet den PERIODIC_UPCALL durch Zurücksetzen des entsprechenden Flags im Kern (Aufruf von *PimMfc::add_mfc_to_kernel*).

- **PimMrt::set_switched_flag** ist die Schnittstellenfunktion zum Setzen des Sbit(S,G) durch die modifizierte PIM-SM-Implementierung (s. folgender Abschnitt). Beim Setzen wird zusätzlich der erste TriggerSwitched gesendet (Aufruf von *PimMfc::dsmc_msg_send*).

Die DSMC-Komponente ist bezüglich des Implementierungsumfangs mit den Modifikationen der Multicastweiterleitung von FreeBSD vergleichbar. Hinzu kommen eine Reihe von XRL-Aufrufen und Funktionen zur Konfiguration durch den Router Manager und zur Kommunikation mit dem Ressourcenmanager über den UDP-Socket. Sie machen den Großteil (etwa 90 %) der prototypischen Implementierung unter XORP aus und sind hier nicht dargestellt.

5.2.3 Modifizierte PIM-SM-Implementierung

Die Erweiterung des Basisverfahrens von DSMC für PIM-SM erfordert einerseits Ergänzungen an der DSMC-Komponente, andererseits die Modifikation der PIM-SM-Implemen-

```
57 bool
58 PimMre::update_sptbit_sg(uint32_t iif_vif_index)
59 {
60     PimMre *pim_mre_wc = wc_entry();
61     PimNbr *pim_nbr_rpfp_nbr_wc = NULL;
62
63     if (iif_vif_index == Vif::VIF_INDEX_INVALID)
64         return false;
65
66     if (! is_sg())
67         return false;
68
69     if (mrib_s() == NULL)
70         return false;
71
72     if (pim_mre_wc != NULL)
73         pim_nbr_rpfp_nbr_wc = pim_mre_wc->rpfp_nbr_wc();
74
75     if ((iif_vif_index == rpf_interface_s())
76         && is_join_desired_sg()
77         && (is_directly_connected_s()
78             || (rpf_interface_s() != rpf_interface_rp())
79             || (inherited_olist_sg_rpt().none())
80             || ((rpfp_nbr_sg() == pim_nbr_rpfp_nbr_wc)
81                 && (rpfp_nbr_sg() != NULL))
82             || is_i_am_assert_loser_state(iif_vif_index))) {
83
84         bool old_sptbit = is_spt();
85
86         set_spt(true);
87
88         if (!old_sptbit)
89             return (rpf_interface_s() != rpf_interface_rp() ||
90                     is_i_am_assert_loser_state(iif_vif_index)) &&
91                     !inherited_olist_sg_rpt().none();
92
93     }
94
95     return false;
96 }
```

Abbildung 5.8 Die modifizierte Funktion *PimMre::update_sptbit_sg* liefert wahr zurück, wenn das Sbit gesetzt werden muss (Ausschnitt aus Datei pim/pim_mre_data.cc).

tierung selbst, d. h. der Klasse PimMrt und den darin gekapselten Klassen. Die im vorigen Abschnitt beschriebene DSMC-Komponente enthält bereits die nötigen Ergänzungen in Form der drei Funktionen *PimMrt::signal_periodic_upcall_message_recv*, *PimMrt::dsmc_msg_trigger_switched_ack_recv* und *PimMrt::set_switched_flag*. Sie erledigen das Senden des TriggerSwitched, das Empfangen des TriggerSwitchedAck und das Zurücksetzen des Sbits. Die dritte Funktion entscheidet dabei nicht selbst über das Setzen des Sbits, sondern stellt lediglich der PIM-SM-Implementierung eine Schnittstelle zum Setzen des Sbits bereit.

Die Entscheidung über das Setzen trifft die modifizierte PIM-SM-Implementierung. Sie muss das Sbit(S,G) zusammen mit dem SPTbit(S,G) setzen, jedoch nur in den Fällen (i) und (ii) bei gleichzeitig nicht zutreffendem Fall (iii) in Update_SPTbit(S,G,iif) (s. Abschnitt 4.3.3). Die vier modifizierten PIM-SM-Funktionen, mit denen das Setzen der Sbits in den zwei betreffenden Fällen veranlasst wird, sind in Abbildung 5.5 dunkel hinterlegt. Die Funktion *PimMre::update_sptbit_sg* (die Klasse PimMre repräsentiert einen Eintrag in einer Multicastroutingtabelle) wurde so erweitert, dass sie wahr zurückliefert, wenn das Sbit(S,G) gesetzt werden muss (s. Abbildung 5.8). Nach einigen grundsätzlichen Überprüfungen in den ersten Zeilen findet sich die Definition von Update_SPTbit(S,G,iif) aus Abschnitt 2.3.2.1 in den Zeilen 75–82 wieder. Zum Setzen des Sbits ergänzt wurden die anschließenden Zeilen ab 84, ausgenommen Zeile 86 mit dem Befehl zum Setzen des SPTbit(S,G). Wurde das SPTbit(S,G) gesetzt, d. h. es war vorher nicht gesetzt (Zeile 88), wird genau dann wahr zurückgeliefert, wenn Fall (i) (Zeile 89) oder Fall (ii) (Zeile 90) und nicht Fall (iii) (Zeile 91) zutrifft.

Die Funktion *PimMre::update_sptbit_sg* wird im Rahmen der PIM-SM-Weiterleitungsroutine von *PimMrt::receive_data* (Fall (i)) und im Rahmen der Assert-Verarbeitung von *PimMre::assert_process_sg* (Fall (ii)) aufgerufen. Der Aufruf von *PimMre::update_sptbit_sg* ersetzt in letzterer Funktion das ursprünglich dort durchgeführte direkte Setzen des SPTbit(S,G) gemäß Aktion A6 (s. Abschnitt 2.3.2.1). Wird wahr zurückgeliefert, setzen danach beide Funktionen durch Aufruf von *PimMfc::set_switched_flag* das Sbit(S,G), erstere indirekt über die Hilfsfunktion *PimMfc::update_mfc*. Beim Setzen sorgt *PimMfc::set_switched_flag* zugleich für den ersten TriggerSwitched. Anschließend wird durch Aufruf von *PimMfc::add_mfc_to_kernel* das Flag im MFC der Weiterleitungsebene gesetzt, um den PERIODIC_UPCALL und damit die ratenlimitierte Wiederholung der TriggerSwitched zu starten. Normalerweise greift die Assert-Verarbeitung in *PimMre::assert_process_sg* nur auf den Multicastroutingeintrag und nicht auf den zugehörigen MFC-Eintrag zu. Es muss daher dort zuerst der passende MFC-Eintrag ermittelt werden, bevor *PimMfc::set_switched_flag* und *PimMfc::add_mfc_to_kernel* aufgerufen werden können. Wie bereits im vorigen Abschnitt erwähnt, wird die Assert-Verarbeitung durch den in Abbildung 5.5 nicht dargestellten Empfang von Multicastpaketen bzw. PIM-Nachrichten ausgelöst.

Da die DSMC-Komponente in das XORP-Modul PIM-SM integriert ist, fällt zum Setzen des Switchover-Bits keine XRL-basierte Kommunikation zwischen der modifizierten PIM-SM-Implementierung und der DSMC-Komponente an. Die an der PIM-SM-Implementierung nötigen Modifikationen beschränken sich so auf etwa 20 Zeilen C++-Code. Insgesamt zeigen die in diesem und im letzten Abschnitt präsentierten Kernbestandteile der prototypischen Implementierung von DSMC unter FreeBSD und XORP den geringen Umfang, den die routerseitige Implementierung von DSMC bedeutet – zumindest, wenn der trotz Integration der DSMC-Komponente in das PIM-SM-Modul immer noch erhebliche verbliebene XORP-spezifische Overhead durch XRL-basierte Interprozesskommunikation außer Acht bleibt.

5.3 Ressourcenmanager

Der Ressourcenmanager wurde nicht prototypisch implementiert. Zum Testen der routerseitigen Implementierung anhand von Topologien aus bis zu sechs Konten wurde stattdessen ein Perl-Skript als Ersatz verwendet. Es öffnet einen UDP-Socket und wartet auf eintreffende Triggernachrichten der Router. Es erzeugt daraufhin passende Conf- und TriggerSwitchedAck-Nachrichten gemäß einer vorgegebenen Konfiguration, die es aus einer

Datei ausliest. Eine Pfadberechnung oder Ressourcenüberprüfung führt das Skript nicht durch.

5.4 Vergleich mit SimpleDSMC

Das in Abschnitt 4.2.14 beschriebene SimpleDSMC repräsentiert eine einfache Art und Weise, mit der die Konfiguration des NRS-Zustands beim Ressourcenmanager angefordert werden kann. Bei SimpleDSMC sendet jeder Router sofort, wenn er den NRS-Zustand benötigt, einen Trigger zum Ressourcenmanager. Beim Basisverfahren von DSMC sendet dagegen nur der letzte Router eines neuen Pfads einen Trigger. Um quantitativ einschätzen zu können, wie groß gegenüber SimpleDSMC die Einsparungen im Basisverfahren an vom Ressourcenmanager zu bearbeitenden Triggern sind, müssen daher die Pfadlängen bestimmt werden, wie sie für neue Empfänger bei ihrem Gruppenbeitritt auftreten. Die Pfadlänge in Hops ist dabei genau das gesuchte Verhältnis an gesendeten Triggern.

Um die Pfadlängen zu bestimmen, wurden mit dem Topologiegenerator BRITE [115] in seiner C++-Variante Topologien unterschiedlicher Größe und mit unterschiedlichen Parametern erzeugt. Als Verfahren wurde Waxman gewählt. Die übrigen von BRITE unterstützten Verfahren dienen der Erzeugung einer Topologie vernetzter Autonomer Systeme. Auf Ebene der Autonomen Systeme weist die Vernetzung eine ganz andere Charakteristik auf als die Vernetzung innerhalb eines Autonomen Systems. Die Verteilung der Anzahl an Links eines Knotens (Autonomes System) zu einem anderen folgt einem Potenzgesetz [63]. Ein neuer Link zwischen Autonomen Systemen führt mit höherer Wahrscheinlichkeit zu einem Autonomen System mit vielen Verbindungen als zu einem, das erst über wenige Verbindungen zu anderen Autonomen Systemen verfügt. Entsprechende Netze heißen skalenfrei oder Kleine-Welt-Netze. Innerhalb eines Autonomen Systems spielt dagegen mehr die geographische Lage der Knoten (Router) zueinander eine Rolle. Je näher zwei Router beieinander liegen, desto eher werden sie vernetzt. Letztlich sind längere Links teurer. Waxman modelliert genau diese empirische Beobachtung.

Die Router werden bei Waxman zufällig auf einer rechteckigen Fläche verteilt. Von jedem Router werden m Links zu einem anderen Router gelegt. Der Zielrouter des Links wird über die Wahrscheinlichkeit $P(u, v)$ für einen Link gemäß

$$P(u,v) \quad = \quad \alpha e^{-d/(\beta L)}, \qquad 0 < \alpha,\beta \leq 1 \tag{5.1}$$

bestimmt. d ist die euklidische Distanz zwischen den Routern u und v, L ist die größte Distanz zwischen zwei beliebigen Routern und α und β sind die Waxman-Parameter. Für die Bestimmung der mittleren Pfadlängen wurden Netze mit N=10 bis N=200 Knoten und m=1 und m=2 pro Router gelegtem Link herangezogen. Die Waxman-Parameter wurden mit α=0,15 und β=0,2 auf ihren Standardwerten belassen.

Pfadlängenbestimmung mittels OMNeT++

Die mit Brite erzeugte Topologiebeschreibung wurde in den diskreten Ereignissimulator OMNeT++ [130] eingelesen. Für jeden Knoten wurde ein Simple Module erzeugt, das gemäß der Topologiebeschreibung über seine Gates mit den anderen verbunden wurde. In einem zweiten Schritt wurde jedes dieser einem Router entsprechenden Simple Modules mit genau einem weiteren (anderen) Simple Module verbunden. Hierdurch war es möglich, mit Hilfe der OMNeT++-Klasse cTopology die Pfade von jedem der Stummellinks zu

jedem anderen berechnen zu lassen. Die Klasse cTopology kann zu einem gegebenen Simple oder Compound Module von jedem anderen Module aus mittels Dijkstra-Algorithmus den kürzesten Pfad berechnen. Als Metrik dient der Hop Count.

Auf jedem der Stummellinks wurde ein Empfänger platziert bis auf einen, der stattdessen die Quelle beherbergte. Die Empfänger treten nun in zufälliger Reihenfolge alle derselben Gruppe mit dieser einen Quelle bei. Bei jedem weiteren Beitritt eines Empfängers wurde die Länge des für ihn neu etablierten Pfads bestimmt, d. h. die Anzahl an Hops, bis der Pfad des neuen Empfängers auf einen bereits etablierten Pfad eines anderen Empfängers trifft oder die Quelle erreicht ist. Dies wurde für jede der N möglichen Positionen der Quelle, d. h. für jeden der N Stummellinks, in der Topologie durchgeführt. Für jede der N Positionen der Quelle wurden R verschiedene zufällig gewählte Beitrittsreihenfolgen der $N-1$ Empfänger untersucht. Die Anzahl an Permutationen wurde mit $R = 10^5$ dabei so groß gewählt, dass sich die resultierenden Häufigkeiten weniger als ein Promille von einer Bestimmung über R andere zufällig gewählte Permutationen unterscheiden. Eine exakte Bestimmung der Häufigkeitsverteilungen ist selbst bei einer Beschränkung auf die ersten k Empfänger mit $(N-1) \cdot (N-2) \cdot \ldots \cdot (N-k-1) = (N-1)!/(N-k-1)!$ durchzuführenden Permutation schon bei Topologien mit wenigen zehn Knoten nicht mehr durchführbar. Wie unten noch ersichtlich wird, sind die Häufigkeitsverteilungen zweier Topologien, die mit demselben Parametersatz generiert wurden, stark unterschiedlich. Eine exakte Bestimmung bringt somit keinen Vorteil, wenn wie hier nach den für einen Parametersatz typischen Häufigkeitsverteilungen gesucht wird.

Die Empfänger und die Quelle wurden nicht als solche in OMNeT++ modelliert. Stattdessen wurde der von cTopology berechnete kürzeste Pfad vom Link mit einem gedachten neu beigetretenen Empfänger bis zum Link mit der gedachten Quelle verfolgt. Dabei wurde an jedem passierten Gate jedes Simple Modules (=Schnittstelle einer Routers) ein Nutzungszähler um eins inkrementiert. War er vor der Erhöhung Null, gehörte der Link noch zu keinem anderen bereits etablierten Pfad. Die Anzahl der Links mit Nutzungszählerstand Null auf dem Pfad zum Stummellink mit der Quelle wurden gezählt und für eine spätere statistische Auswertung zusammen mit der augenblicklichen Pfadnummer in der Beitrittsreihenfolge auf Festplatte geschrieben. Nachdem alle Empfänger beigetreten waren, d. h. von jedem Stummellink zum Stummellink mit der Quelle die Pfade verfolgt und die Links mit Nutzungszählerstand Null gezählt wurden, wurden alle Nutzungszähler zurückgesetzt und der Vorgang wiederholt, insgesamt R mal mit jeweils anderer (zufällig gewählter) Beitrittsreihenfolge. Danach wurde die Quelle auf den nächsten der N Stummellinks platziert und erneut R Durchläufe ausgeführt usw., bis die Pfadlängen für alle N möglichen Positionen der Quelle R-fach bestimmt waren. Die auf Festplatte geschriebenen Pfadlängen wurden anschließend mit Hilfe von Perl und dem Perlmodul Statistics::Descriptive statistisch ausgewertet. Für jede Pfadnummer innerhalb der Beitrittsreihenfolge wurde das arithmetische Mittel über alle R Läufe und N Positionen gebildet.

Für jeden Satz von zur Topologieerzeugung verwendeten Parametersatz N, m, α und β wurden mit BRITE mehrere Topologien mit unterschiedlichen Seeds für dessen Zufallszahlengenerator erzeugt. Für jede generierte Topologie wurden mit OMNeT++ wie beschrieben die mittleren Pfadlängen getrennt nach Nummer in der Beitrittsreihenfolge ermittelt. Die Anzahl der je Parametersatz erzeugten unterschiedlichen Topologien kann Tabelle 5.4 entnommen werden. Von den großen Topologien wurden weniger als von kleinen vermessen, da die Pfadlängenbestimmung bei einer 200-Knoten-Topologie bereits 12 Stunden in Anspruch nahm. Aus demselben Grund wurden auch keine Topologien mit 500

oder mehr Knoten vermessen. Bei den großen Topologien konzentrierten sich zudem die Untersuchungen auf m=1, da auch in ihnen bereits Router mit bis zu 10 Links auftraten. Größere Werte für m führen zu zunehmend unrealistisch stark vermaschten Topologien.

Ergebnisse

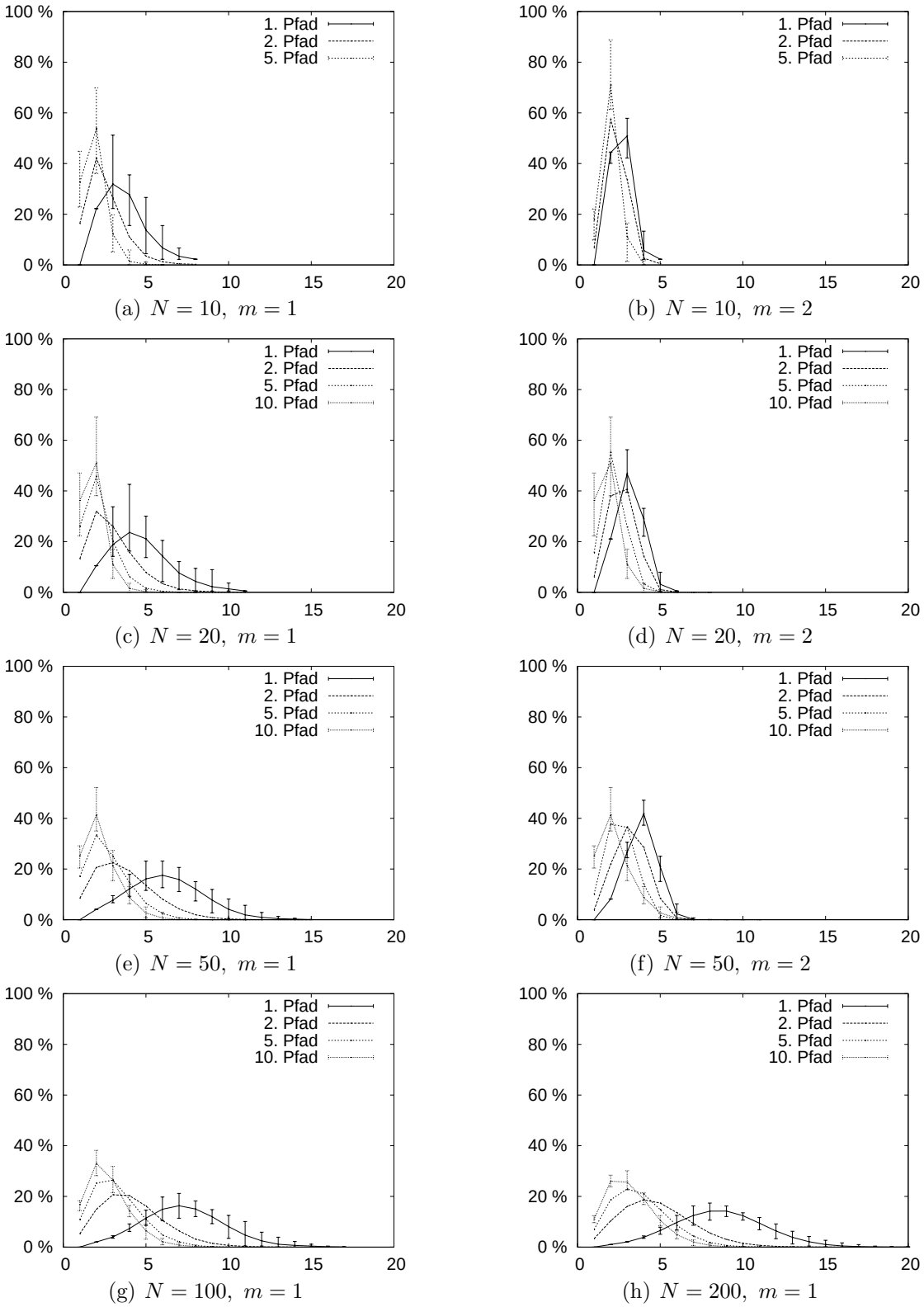
Abbildung 5.9 zeigt für jeden der acht untersuchten Parametersätze die Häufigkeitsverteilungen für einige der ersten beigetretenen Empfänger bzw. Pfade. Die dargestellten Verteilungen sind jeweils durch Mittelung über alle innerhalb eines Parametersatzes vermessenen Topologien entstanden.

Die Verteilungen eines Parametersatzes enden beim Erreichen des maximalen Durchmessers aller innerhalb des Parametersatzes vermessenen Topologien. Dargestellt sind die Verteilungen der Pfadlängen für den 1., 2., 5. und 10. Empfänger. Da es bei den zwei Parametersätzen mit N=10 Routern nur 9 Empfänger gibt, fehlt die Häufigkeitsverteilung für den 10. Empfänger. Die Verteilungen für später beigetretene Empfänger sind nicht mehr dargestellt. Der in der Abbildung erkennbare Trend setzt sich jedoch fort, so dass bei den letzten Empfängern nur noch sehr kurze Pfade mit einem oder zwei Hops auftreten. Schließlich sind durch die vielen schon beigetretenen Empfänger schon fast alle Links Teil des Multicastverteilbaums.

Die mittleren Pfadlängen zweier Waxman-Topologien unterscheiden sich trotz derselben zur Erzeugung verwendeten Parametersätze erheblich. Zur Verdeutlichung sind in Abbildung 5.9 für den 1. und 10. Pfad die innerhalb des jeweiligen Parametersatzes auftretenden minimalen und maximalen mittleren Häufigkeiten durch Balken kenntlich gemacht. So tritt z. B. bei einer Topologie mit N=100 Routern und m=1 gelegtem Link je Router beim 1. Empfänger eine Pfadlänge von 11 Hops mit über 10 % Wahrscheinlichkeit auf, bei einer anderen Topologie mit denselben Parametern dagegen sehr selten. Dies ist auf den stark unterschiedlichen Durchmesser zurückzuführen, der bei der zweiten Topologie nur 11 beträgt. Nur durch die Mittelung über viele Topologien lassen sich daher für einen Parametersatz aussagekräftige Werte erhalten.

In Abbildung 5.10 ist die durchschnittliche Pfadlänge, d. h. der Erwartungswert der Verteilungen aus Abbildung 5.9, über der Empfänger- bzw. Pfadnummer für alle $N-1$ Pfadnummern einer Topologie aufgetragen. Auch hier verdeutlichen Balken die zwischen den einzelnen Topologien mit gleichem Parametersatz deutlich schwankenden Durchschnittswerte. So weisen insbesondere beim 1. und 2. Multicastpfad einige Topologien eine größere durchschnittliche Pfadlänge auf als manche Topologie mit doppelt so vielen Routern.

Wie in Abbildung 5.10 zu sehen, nimmt die Wahrscheinlichkeit für längere Pfade schnell ab. Der 2. Pfad ist bei Topologien mit m=1 bereits um etwa 40 % kürzer, der 10. Pfad meist über 60 %. Topologien mit m=2 sind enger vermascht und der 1. Pfad entsprechend deutlich kürzer. Dafür nimmt die Länge mit der Pfadnummer weniger stark ab. Für Gruppen mit sehr hoher Empfängerdichte, d. h. wenn beim Beitritt eines spät hinzukommenden Empfängers nahezu jeder Link bereits vom Multicastbaum abgedeckt ist, nähert sich die durchschnittliche Pfadlänge 1,5 an. Das bedeutet, es ist zunehmend etwa gleich wahrscheinlich, dass der Router, an dem sich der Empfänger befindet, bereits auf dem



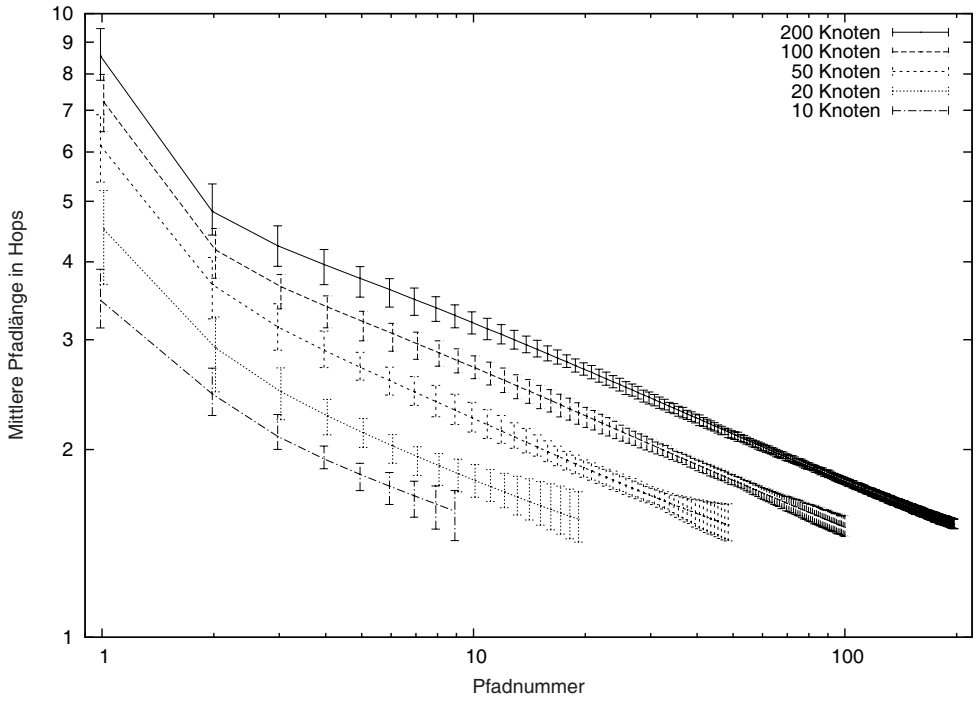
(a) $N = 10,\ m = 1$ (b) $N = 10,\ m = 2$

(c) $N = 20,\ m = 1$ (d) $N = 20,\ m = 2$

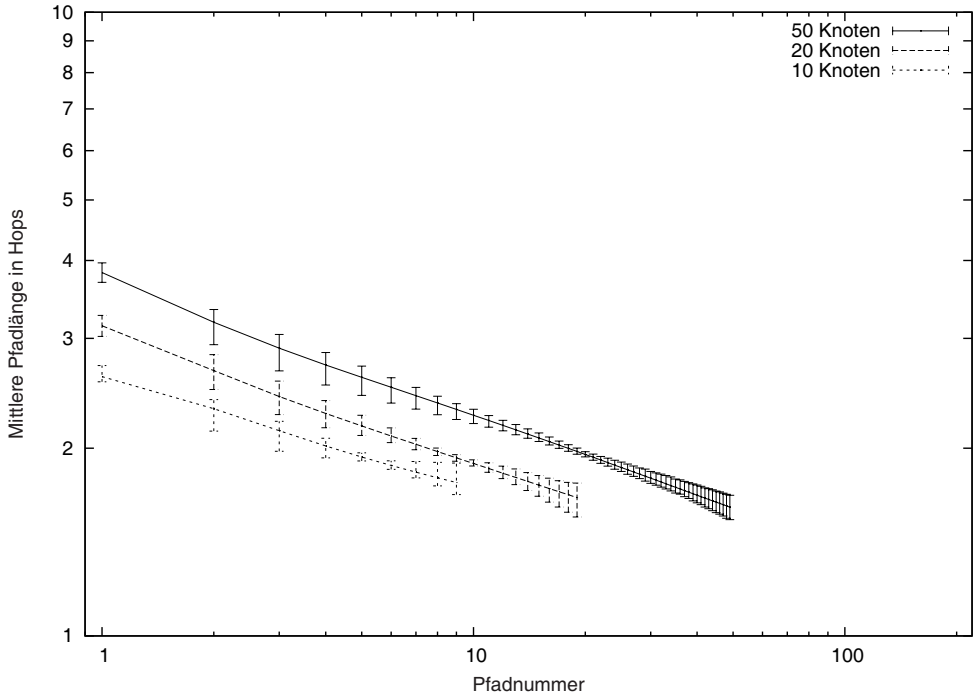
(e) $N = 50,\ m = 1$ (f) $N = 50,\ m = 2$

(g) $N = 100,\ m = 1$ (h) $N = 200,\ m = 1$

Abbildung 5.9 Wahrscheinlichkeit verschiedener Pfadlängen (Abszisse, in Hops), gemittelt über 14–63 (s. Tabelle 5.4) verschiedene Waxman-Topologien. Für den 1. und 10. (5. bei N=10) Pfad sind zusätzlich die Maxima und Minima eingetragen. α=0,15, β=0,2.



(a) $m = 1$



(b) $m = 2$

Abbildung 5.10 Durchschnittliche Pfadlänge (Erwartungswert der Verteilungen aus Abbildung 5.9) für den n-ten Pfad eines Multicastverteilbaums für verschiedene Waxman-Topologien mit N Routern (Knoten), je Parametersatz gemittelt über 14–63 (s. Tabelle 5.4) verschiedene Topologien. Die Balken geben die größte und kleinste durchschnittliche Pfadlänge aller Topologien innerhalb eines Parametersatzes an. (a) m=1, (b) m=2; α=0,15, β=0,2.

N	m	Ø der Pfade		Anzahl
		1–10	1–5	Topologien
10	2	2,0	2,2	63
10	1	2,0	2,4	63
20	2	2,3	2,5	63
20	1	2,4	2,9	63
50	2	2,7	3,0	63
50	1	3,1	3,7	63
100	1	3,6	4,3	24
200	1	4,2	5,1	15

Tabelle 5.4 Durchschnitt der Pfadlängen der ersten 5 bzw. 10 Empfänger (bei N=10 Durchschnitt aller 9 Empfänger), gemittelt über die angegebene Anzahl verschiedener Waxman-Topologien mit demselben Parametersatz N, m, α, β; α=0,15, β=0,2.

Multicastbaum liegt, wie dass einer der von diesem Router aus nächsten Router auf dem Multicastbaum liegt.

Das Ziel des Pfadaufbaus muss nicht notwendigerweise eine Quelle sein. Die Pfadlänge bzw. die Anzahl an durch SimpleDSMC gesendeten Triggern ist dieselbe, wenn an Stelle der Quelle die Rendezvousstelle von PIM-SM oder der Rendezvousstellenlink von BIDIR-PIM tritt und der Aufbau des Rendezvousstellenbaums betrachtet wird. Gleiches gilt, wenn die Schnittstellen zu den Stummellinks nicht die Schnittstellen zu Links mit Quellen oder Empfängern sind sondern DSMC-Grenzschnittstellen. Die Stummellinks entsprechen dann den Links zu benachbarten Autonomen Systemen.

Geht man von Gruppen mit geringer Empfängerdichte aus, deren Empfänger über das gesamte Internet verteilt sind, sind meist nur die ersten Pfadlängen relevant. Bei einer typischen Pfadlänge im Internet von 4 Autonomen Systemen [28] besitzt eine Multicastgruppe bei konstant 5 Pfaden je passiertem Autonomen System 5^4=625, bei 10 Pfaden bereits 10000 Gruppenmitglieder. Die höheren Pfadnummern treten daher nur bei sehr großen internetweit verteilten Gruppen auf. Für Tabelle 5.4 wurden daher Durchschnittswerte nur über die ersten 5 und die ersten 10 Pfade bzw. alle 9 Pfade bei N=10 bestimmt. SimpleDSMC erzeugt demnach im Mittel für den Beitritt eines Empfängers in kleinen Topologien bis zu einigen zehn Routern 2 bis 3 Trigger, für größere mit hundert und mehr Routern bis zu 5 Trigger. Im selben Maße reduziert das Basisverfahren von DSMC den Aufwand für den Ressourcenmanager, da hier je Empfänger nur jeweils ein Trigger erzeugt wird. Bei gleicher Leistungsfähigkeit des Ressourcenmanagers lassen sich somit gegenüber SimpleDSMC je nach Größe eines Autonomen Systems etwa 3 bis 5 mal mehr Multicastgruppen gleichzeitig unterstützen.

5.5 Zusammenfassung

Die routerseitige Implementierung von DSMC wurde prototypisch für FreeBSD und XORP vorgenommen und mit einem Perl-Skript als Ersatz für einen Ressourcenmanager überprüft. Die praktische Umsetzbarkeit des Verfahrens DSMC wurde damit routerseitig nachgewiesen

und gezeigt, wie gering die nötigen Erweiterungen ausfallen. Die prototypische Implementierung von DSMC kommt in der Weiterleitungsebene (FreeBSD) mit etwa 300 Zeilen C-Code aus. Die Implementierung der DSMC-Komponente in der Routingebene (XORP) ist nur wegen der aufwendigen XRL-basierten Interprozesskommunikation von XORP etwa eine Größenordnung umfangreicher. Die Kernfunktion ist jedoch nicht aufwendiger als die der Weiterleitungsebene.

Die Untersuchung der auftretenden Multicastpfadlängen anhand generierter Topologien von Autonomen Systemen legt nahe, dass mit DSMC gegenüber der einfachen direkten Anforderung der Konfiguration durch jeden Router selbst (SimpleDSMC) eine deutliche Einsparung an Triggern und damit eine deutliche Reduzierung der Belastung des Ressourcenmanagers erreicht werden kann. Je nach Größe des Autonomen Systems scheint eine Reduzierung um den Faktor 2 bis 3 für einige zehn Router bis hin zu einem Faktor von 5 für sehr große Autonome Systeme mit einhundert und mehr Routern realistisch.

6 Zusammenfassung

In dieser Arbeit wurde das Verfahren DSMC (Diffserv Multicast) entworfen mit dem Ziel, das von Bless und Wehrle definierte Neglegted Reservation Subtree Problem [27] (vgl. Abschnitt 3.1) zu lösen. Das NRS-Problem besagt, dass die im Rahmen der Multicastweiterleitung stattfindende Paketduplizierung zur unerwünschten Verdrängung von Datenströmen anderer Dienste führen kann, wenn die Duplizierung von hochpriorem Verkehr nicht durch das Dienstgütemanagement kontrolliert wird. Das NRS-Problem tritt nur bei herkömmlichem Multicastrouting auf. Alternative Verfahren zur Gruppenkommunikation im Internet bilden diese entweder mittels Overlaynetzen auf Unicast ab oder die Quelle gibt die Empfänger fest vor, so dass es zu keiner unkontrollierten Duplizierung kommen kann (vgl. alternative Multicastverfahren in Abschnitt 3.4). DSMC erlaubt gegenüber der von Bless und Wehrle vorgeschlagenen Lösung, nicht nur einen Dienst sondern gleichzeitig mehrere Dienste in einer Multicastgruppe zu nutzen. Darüber hinaus umfasst DSMC eine geeignete Signalisierung, mit der die Ressourcennutzung durch einen zentralen Ressourcenmanager kontrolliert werden kann, diese fehlt in [27] völlig. Die wesentlichen Vorteile von DSMC gegenüber vergleichbaren Ansätzen [194, 21] (vgl. Abschnitte 3.2 und 3.3) sind, dass es ohne Änderungen an den Multicastroutingprotokollen auskommt und alle derzeit im Internet eingesetzten Multicastroutingprotokolle unterstützt werden.

DSMC erweitert die von allen Protokollen gemeinsam genutzte Paketweiterleitung der Router um eine Funktion zur Kontrolle der Duplizierung von hochpriorem Verkehr. Wichtiges Designziel war es, eine für Unicast vorhandene Dienstgütesignalisierung und Zugangskontrolle möglichst unverändert weiter nutzen zu können. DSMC erwartet lediglich, dass sich hierbei auch Multicastadressen als Zieladressen angeben lassen und die Zugangskontrolle auf Basis von Multicastadressen erfolgen kann. Prinzipiell nicht vermeiden lässt sich dagegen eine Erweiterung der Ressourcenverwaltung im zentralen Ressourcenmanager, denn sie muss Multicastpfade berechnen und Ressourcen für sie belegen können. DSMC ist nicht für verteiltes Ressourcenmanagement entworfen worden, Teile des NRS-Zustands können jedoch als Alternative zum Eingriff in das Multicastrouting oder dem vollständigen Policing auf domäneninternen Routern verwendet werden, was andernfalls zur Kontrolle der Paketduplizierung bzw. der Ressourcennutzung und damit zur Lösung des NRS-Problems nötig wäre.

DSMC wurde prototypisch unter XORP (Extensible Open Router Platform) [192] und FreeBSD [69] implementiert. Es führt zur Lösung des NRS-Problems den neu entworfenen NRS-Zustand (vgl. Abschnitt 4.2.1) in den MFC (Multicast Forwarding Cache, s. Abschnitt 2.3.1) ein und spezifiziert eine Signalisierung (vgl. Abschnitt 4.2.2), bei der erst mit der Weiterleitung von Paketen die Konfiguration des NRS-Zustands mittels so genannter Trigger-Nachrichten beim Ressourcenmanager angefordert wird, d. h. erst dann, wenn der NRS-Zustand tatsächlich benötigt wird. Das neben dem NRS-Zustand als zweites in den MFC eingeführte DRbit (Designated-Router-Bit) sorgt dafür, dass DSMC mit nur einer Trigger-Nachricht je neuem Empfänger bzw. Multicastpfad unabhängig von der Anzahl an Routern auf dem Pfad auskommt, was die Anzahl der vom Ressourcenmanager zu verarbeitenden Nachrichten gering hält und DSMC skalierbar macht.

DSMC setzt auf der Ebene der Paketweiterleitung an. Damit bleibt es unabhängig vom Multicastroutingprotokoll, solange der Multicastpfad vom zentralen Ressourcenmanager allein durch Kenntnis von Quelle, Empfänger, für die Gruppe verwendetem Multicastroutingprotokoll und zugrunde liegender MRIB (Multicast Routing Information Base, s. Abschnitt 2.1.1) bestimmt werden kann. Dies trifft auf alle Multicastroutingprotokolle im Internet zu, für PIM-SM (Protocol Independent Multicast – Sparse Mode, s. Abschnitt 2.3.2.1) mit der Einschränkung, dass darüber hinaus bekannt sein muss, welchen seiner zwei alternativen Multicastpfade, quellenspezifischer kürzester Baum oder Rendezvousstellenbaum, PIM-SM momentan nutzt. Anstatt jedoch wie andere Ansätze [194, 21] PIM-SM zu modifizieren, greift DSMC nicht in den Protokollablauf von PIM-SM ein, sondern bestimmt lediglich den aktuellen Wert des SPTbit, ohne ihn dabei zu verändern, und sendet entsprechend eine von zwei alternativen Trigger-Nachrichten. Zusätzlich muss beim Wechsel auf den quellenspezifischen kürzesten Baum der zentrale Ressourcenmanager informiert werden, wozu eine PIM-SM-Implementierung, nicht jedoch das Protokoll PIM-SM, für DSMC minimal erweitert wird: In der PIM-SM-Funktion Update_SPTbit (s. Weiterleitungsroutine und SPTbit in Abschnitt 2.3.2.1) wird in zwei von fünf Fällen zusammen mit dem SPTbit auch das für PIM-SM in DSMC neu eingeführte Sbit (Switchover-Bit) gesetzt (vgl. Abschnitt 4.3.1). Hierdurch benötigt DSMC bei PIM-SM zusätzlich nur eine Nachricht je Wechsel des Verteilbaums unabhängig von der Anzahl an Empfängern und bleibt auch bei PIM-SM skalierbar. Darüber hinaus verlangt DSMC keine weiteren Anpassungen.

DSMC reduziert die Anzahl an Trigger-Nachrichten an den Ressourcenmanager verglichen mit einer einfachen Signalisierung, bei der jeder Router seine Konfiguration für sich beim Ressourcenmanager anfordert. Nicht durch DSMC reduziert werden die Konfigurationsnachrichten vom Ressourcenmanager an die Router, da dies nur eine geringe Aufwandsverringerung für den Ressourcenmanager bedeuten würde und somit nur geringen Einfluss auf die Skalierbarkeit von DSMC hat. Es wurde jedoch eine Möglichkeit skizziert, wie sich zumindest für IPv6 mit Hilfe eines neuen IPv6-Erweiterungskopfs die Anzahl an Konfigurationsnachrichten auf eine je neuen Pfad und damit die Anzahl an DSMC-Nachrichten insgesamt auf zwei je neuem Empfänger reduzieren lässt (vgl. Abschnitt 4.2.15).

Durch den Verzicht auf multicastspezifische Erweiterungen in der Dienstgütesignalisierung beschränkt sich bei DSMC die Einflussnahme der Empfänger darauf, die von der Quelle vorgesehene Dienstgüte zu akzeptieren, oder nicht zu akzeptieren und wieder aus der Gruppe auszutreten. Alles darüber Hinausgehende erfordert die Aushandlung von Dienstgüteparametern zwischen allen Sendern und Empfängern einer Gruppe, was entweder durch die Anwendung selbst oder durch eine Multicast unterstützende Dienstgütesignalisierung

umgesetzt werden kann. Eine diesbezügliche Erweiterung des Dienstgütesignalisierungsprotokolls QoS-NSLP [110] (NSIS (Next Steps in Signaling) Signaling Layer Protocol, s. Abschnitt 2.4) wurde nicht betrachtet, da sich QoS-NSLP noch in der Entwicklung befindet, wäre für die Zukunft aber denkbar.

A Alternative Weiterleitungsarchitekturen

Es existieren Architekturen für die Multicastpaketweiterleitung, die nicht in der Lage sind, Pakete bei der routerinternen Paketduplizierung zu modifizieren. Je nach Art der internen Paketduplizierung müssen zwei Varianten unterschieden werden (vgl. Abschnitt 2.3.1): Ausgang filtert (interner Broadcast) oder der Eingang bestimmt die Ausgänge (interner Multicast). Bei ersterer befindet sich der NRSstate(S,G,Oif) zusammen mit dem Activebit(S,G,Oif) am Ausgang Oif. Hier muss je nach NRS-Zustand das Paket vor dem Weiterleiten wie gehabt ummarkiert werden, die Einschränkung stellt hier also kein Problem dar. Bei letzterem muss der Eingang dagegen nacheinander mehrere Kopien des Pakets an unterschiedliche Ausgänge schicken: Je eine mit unverändertem DSCP, DSCP_SIG, DSCP_UC und DSCP_LE an die aktiven Ausgänge im NRS-Zustand Keep, Init, Invalid bzw. Remark, sinnvollerweise in dieser Reihenfolge, um die Verzögerung für Pakete mit hochprioren DSCPs zu minimieren.

Alternativ ist es auch möglich, DSMC ohne jede Modifikation der Ein- und Ausgangsschnittstellen nur durch Hinzufügen von zwei Art Loopbackschnittstellen zu implementieren (Tabelle A.1). Die Loopbackschnittstelle „Invalid“ markiert auf DSCP_UC, die Schnittstelle „Remark“ auf DSCP_LE um. Der NRS-Zustand befindet sich dann nur noch in der Kopie des MFCs in der Routingebene und wird im eigentlichen (verteilten) MFC in der Weiterleitungsebene durch Aktivieren der entsprechenden (Loopback-)Schnittstellen dargestellt:

- Ausgang filtert: Initial unter ungültigem NRS-Zustand erhält jede Ausgangsschnittstelle die Loopbackschnittstelle Invalid als Eingangsschnittstelle und diese die tatsächliche Eingangsschnittstelle als Eingangsschnittstelle. Wechselt der NRS-Zustand nach Remark oder Keep, wird der Ausgangsfilter so umkonfiguriert, dass er Pakete von der Loopbackschnittstelle Remark bzw. von der tatsächlichen Eingangsschnittstelle weiterleitet.
- Eingang bestimmt die Ausgänge: Wird eine Ausgangsschnittstelle Oif aktiv, wird diese statt an der Eingangsschnittstelle Iif an der Loopbackschnittstelle Invalid

NRSstate(S,G)	Routing-ebene	Loopbackschnittstelle Invalid	Loopbackschnittstelle Remark	Oif	Fall
Init	Iif			Routing	DSCP_SIG ≠ DSCP_UC
Init	Iif	Iif			DSCP_SIG = DSCP_UC
Init	Iif			Routing	DSCP beibehalten
Invalid	Iif	Iif		Invalid	
Remark			Iif	Remark	
Keep				Iif	

NRSstate(S,G,Oif)	Iif	Routing-ebene	Loopbackschnittstelle Invalid	Loopbackschnittstelle Remark	Fall
Init	Routing	Oif			DSCP_SIG ≠ DSCP_UC
Init	Ind+Rtg		Oif		DSCP_SIG = DSCP_UC
Init	Routing	Oif			DSCP beibehalten
Invalid	Ind+Rtg		Oif		
Remark	Remark			Oif	
Keep	Oif				

Tabelle A.1 Oben: Ausgang filtert. Schnittstellen Oif, Invalid und Remark leiten abhängig vom NRS-Zustand von Iif, Invalid oder Remark weiter.
Unten: Eingang bestimmt die Ausgänge. Iif, Invalid und Remark leiten je nach NRS-Zustand auf Oif, Invalid oder Remark weiter. Zustand Init s. Text (Rtg/Routing = Routingebene, Ind = anstelle Paket nur Indikation über fehlenden MFC-Eintrag an Routingebene).

aktiviert und diese Loopbackschnittstelle an der Eingangsschnittstelle aktiviert. Bei Konfiguration durch den Ressourcenmanager wird Oif bei der Loopbackschnittstelle Invalid wieder inaktiviert und je nach NRS-Zustand entweder bei der Loopbackschnittstelle Remark aktiviert (und diese bei der eigentlichen Eingangsschnittstelle) oder Oif direkt bei der eigentlichen Eingangsschnittstelle aktiviert.

Eine Loopbackschnittstelle „Init“ macht dagegen keinen Sinn. Je nach Implementierung wird der MFC-Eintrag in den Schnittstellen immer erst mit dem Empfang des ersten Datenpakets angelegt. In diesem Fall wird das erste Paket von der Routingebene empfangen, die daraufhin den MFC-Eintrag anlegt, den Zustand nach Invalid wechselt und die Unterdrückungszeit der TriggerUC startet. Empfängt die Routingebene vollständige Pakete und nicht nur Nachrichten über einen fehlenden MFC-Eintrag, kann sie das Paket auf DSCP_SIG ummarkiert weiterleiten. Andernfalls gehen die ersten Pakete verloren, bis der MFC-Eintrag angelegt ist, unabhängig davon, ob DSMC implementiert ist oder nicht. Legt eine Implementierung MFC-Einträge bereits zusammen mit dem Routingzustand an, ist es am sinnvollsten, DSCP_SIG = DSCP_UC zu wählen und den MFC-Eintrag initial so zu konfigurieren, dass über die Loopbackschnittstelle Invalid weitergeleitet wird. Alternativ kann der Ausgang initial deaktiviert werden und erst nach dem Wechsel in den NRS-Zustand Invalid aktiviert werden. Dies macht allerdings nur dann Sinn, wenn die Routingebene vollständige Pakete empfängt und Pakete ummarkiert weiterleiten kann. Um die Unterdrückungszeit und die durch die Weiterleitung von Paketen ausgelösten Trigger zu implementieren, muss die Routingebene, solange der NRS-Zustand ungültig ist, Pakete empfangen bzw. über ihren Empfang benachrichtigt werden können. Wird beim Wechsel vom Zustand Init nach Invalid das DRbit gesetzt, kann alternativ der Empfang durch die Routingebene beendet und die Trigger timergesteuert wiederholt werden. Wird das DRbit

nicht gesetzt, ist dies so nicht möglich, da festgestellt werden muss, ob Pakete bereits mit DSCP_UC markiert eintreffen oder nicht. Und nur falls nicht dürfen TriggerUC gesendet werden.

B DSMC-Implementierung unter FreeBSD

Um zu verdeutlichen, wie gering die für DSMC nötigen Modifikationen im performancekritischen Teil der Weiterleitungsebene sind, wird nachfolgend am Beispiel von IPv4 die Implementierung von DSMC im FreeBSD-Kern wiedergegeben. Nach einer kurzen Einführung in die Multicastrouting-API werden anhand von Quellcodeausschnitten die wesentlichen Erweiterungen an der Multicastrouting-API sowie der Multicastweiterleitung erläutert. Alle für DSMC nötigen Modifikationen beschränken sich auf die beiden Dateien *ip_mroute.h* und *ip_mroute.c*.

B.1 Multicastrouting-API

Die Schnittstelle zwischen Routingdaemon und Multicastweiterleitung bildet ein Socket vom Typ Raw und je nach Adressfamilie (*domain*) mit Protokoll IGMP bzw. ICMPv6:

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int domain, int type, int protocol);
int getsockopt(int s, int level, int optname, void *optval, socklen_t *optlen);
int setsockopt(int s, int level, int optname, const void *optval, socklen_t optlen);

mrouter_socket4 = socket(AF_INET, SOCK_RAW, IPPROTO_IGMP);
mrouter_socket6 = socket(AF_INET6, SOCK_RAW, IPPROTO_ICMPV6);

#include <netinet/ip_mroute.h>
struct mrt_dscp dscp;

error = getsockopt(mrouter_socket4, IPPROTO_IP, MRT_DSCP, &dscp, sizeof(dscp));

#include <netinet6/ip6_mroute.h>
struct mrt6_dscp dscp6;

error = getsockopt(mrouter_socket6, IPPROTO_IPV6, MRT6_DSCP, &dscp6, sizeof(dscp6));
```

Über diese meist als Mrouter-Socket bezeichnete Schnittstelle steuert der Routingdaemon alle Funktionen des MFCs im FreeBSD-Kern mit Hilfe der beiden Systemfunktionen

setsockopt und *getsockopt*, wobei *level* auf IPPROTO_IP bzw. IPPROTO_IPV6 gesetzt wird. Abhängig von der jeweiligen Socketoption *optname* wird mit *optval* ein Zeiger auf einen Parameter der Länge *optlen* übergeben.

Die zur Steuerung des MFCs benötigten Socketoptionen *optname* sind in der Datei <*netinet/ip_mroute.h*> (IPv4) bzw. <*netinet6/ip6_mroute.h*> (IPv6) definiert[1]. Nachfolgend wird die Implementierung von DSMC für IPv4 erläutert. Wie bereits die Unterschiede zwischen Tabelle 5.1 und Tabelle 5.2 erkennen lassen, unterscheiden sich die Implementierungen der Multicastweiterleitung für IPv4 und IPv6 in vielen Details. So besitzt ein Ausgang im MFC-Eintrag bei IPv6 keine Flags, DSMC verwendet stattdessen Bitfelder (*dscp_keep_dscp* usw.). Da jedoch ansonsten die Umsetzung von DSMC in der Funktion *ip6_mdq* analog zu der in *ip_mdq* (s. Abschnitt B.3) erfolgt, soll hier die Beschreibung von IPv4 genügen.

B.2 API-Erweiterung – ip_mroute.h

Das Verfahren DSMC definiert zwei neue Socketoptionen MRT_DSCP und MRT_DSCP_NOCONF (Zeilen 72 und 73):

netinet/ip_mroute.h

```
59 #define MRT_INIT 100 /* initialize forwarder */
60 #define MRT_DONE 101 /* shut down forwarder */
61 #define MRT_ADD_VIF 102 /* create virtual interface */
62 #define MRT_DEL_VIF 103 /* delete virtual interface */
63 #define MRT_ADD_MFC 104 /* insert forwarding cache entry */
64 #define MRT_DEL_MFC 105 /* delete forwarding cache entry */
65 #define MRT_VERSION 106 /* get kernel version number */
66 #define MRT_ASSERT 107 /* enable assert processing */
67 #define MRT_PIM  MRT_ASSERT /* enable PIM processing */
68 #define MRT_API_SUPPORT 109 /* supported MRT API */
69 #define MRT_API_CONFIG 110 /* config MRT API */
70 #define MRT_ADD_BW_UPCALL 111 /* create bandwidth monitor */
71 #define MRT_DEL_BW_UPCALL 112 /* delete bandwidth monitor */
72 #define MRT_DSCP 113 /* global DSCP values */
73 #define MRT_DSCP_NOCONF 114 /* rate limits for dscp_noconf notif'ns */
```

Mit der ersten Option MRT_DSCP und der Struktur *mrt_dscp* als Parameter werden die drei DSCPs, auf die in den NRS-Zuständen Remark, Invalid und Keep ummarkiert wird, und die niederprioren DSCP, für die der NRS-Zustand nicht gilt, routerglobal konfiguriert und ausgelesen:

```
145 struct mrt_dscp {
146     u_char dscp_unavail;
147     u_char dscp_noconf;
148     u_char dscp_signal;
149 #define DSCP_KEEP (DSCP_MAX+1)
150     uint64_t dscpmap_lowprio;
151 };
152 #define DSCP_UNAVAIL DSCP_EXP(14)
153 #define DSCP_NOCONF DSCP_EXP(15)
154 #define DSCP_SIGNAL DSCP_CS(6)
155 #define DSCPMAP_LOWPRIO DSCPMAP_BIT(DSCP_BE)
```

[1] Linux IPv4: <linux/mroute.h>; IPv6 Multicast Routing unterstützt Linux nicht.

```
156
157 #define DSCPMAP_BIT(n)  ((uint64_t)1 << (n))
158 #define DSCPMAP_SET(m,n) ((m) |= DSCPMAP_BIT(n))
159 #define DSCPMAP_CLEAR(m,n) ((m) &= ~DSCPMAP_BIT(n))
160 #define DSCPMAP_ISSET(m,n) ((m) & DSCPMAP_BIT(n))
```

Für *dscp_signal* kann mit dem speziellen für einen DSCP ansonsten ungültigen Wert DSCP_KEEP das Beibehalten des DSCPs im NRS-Zustand Init konfiguriert werden. Standardmäßig werden als DSCP_LE und DSCP_UC zwei experimentelle DSCPs aus DSCP-Pool 2 und CS6 für DSCP_SIGNAL verwendet. Die für die DSCPs verwendeten Makros mit selbsterklärenden Namen wurden ebenfalls neu eingeführt:

```
110 #define DSCP_AF(c,p) (((c) & 7) << 3 | ((p) & 3) << 1) /* assured forward.*/
111 #define DSCP_CS(n) DSCP_AF(n,0) /* class selector codepoints */
112 #define DSCP_EF  DSCP_AF(5,3) /* expedited forwarding */
113 #define DSCP_BE  DSCP_CS(0) /* best effort */
114 #define DSCP_LE  DSCP_CS(1) /* lower effort */
115 #define DSCP_POOL3(n) (((n) & 15) << 2 | 1) /* exp/lu; MAY become std */
116 #define DSCP_EXP(n) (((n) & 15) << 2 | 3) /* experimental/local use */
117
118 #define DSCP_MAX 63
119 #define DSCP_MASK DSCP_MAX
120 #define DSCP_SHIFT 2 /* right shift to get dscp from tos */
```

Mit der zweiten Socketoption MRT_DSCP_NOCONF und der Struktur *mrt_dscp_noconf* werden die Rate über ihren Kerwert die Periodendauer (in Mikrosekunden) und die anfängliche Unterdrückungszeit (in Vielfachen der Periodendauer) der Nachrichten an den Routingdaemon routerglobal konfiguriert und ausgelesen:

```
173 struct mrt_dscp_noconf {
174     long msg_time;
175     u_char msg_suppress;
176 };
177 #define DSCP_NOCONF_MSG_TIME  1000000  /* one second */
178 #define DSCP_NOCONF_MSG_SUPPRESS 3  /* three seconds */
```

Wenn die DSMC-Komponente im Routingdaemon wie die in Abschnitt 5.2.2 vorgestellte bei jeder Nachricht des MFCs sofort einen Trigger, TriggerRpt, TriggerSwitched oder TriggerUC weiter zum Ressourcenmanager sendet, ist dies gleichzeitig die Rate der Trigger, TriggerRpt und TriggerSwitched und die Verzögerung der TriggerUC.

Der NRS-Zustand, Designated-Router-Bit und Switchover-Bit werden mit Hilfe noch ungenutzter *mfc_flags[]* umgesetzt. Sie werden mit der Socketoption MRT_ADD_MFC (Zeile 63) gesetzt[2], die die Struktur *mfcctl2*[3] als Parameter übergibt. Die Struktur *mfcctl2* selbst bleibt jedoch unverändert:

```
 81 #define MAXVIFS 32
212 struct mfcctl2 {
213         /* the mfcctl fields */
214         struct in_addr mfcc_origin;  /* ip origin of mcasts */
215         struct in_addr mfcc_mcastgrp;  /* multicast group associated*/
216         vifi_t  mfcc_parent;  /* incoming vif  */
```

[2]Ein Auslesen des MFCs ist nicht möglich.

[3]Linux kennt nur die einfachere Struktur *mfcctl*, in der die Flags fehlen.

```
217         u_char  mfcc_ttls[MAXVIFS]; /* forwarding ttls on vifs */
218
219         /* extension fields */
220         uint8_t  mfcc_flags[MAXVIFS]; /* the MRT_MFC_FLAGS_* flags */
221         struct in_addr mfcc_rp;  /* the RP address */
222 };
223 /*
224  * The advanced-API flags.
225  *
226  * The MRT_MFC_FLAGS_XXX API flags are also used as flags
227  * for the mfcc_flags field.
228  */
229 #define MRT_MFC_FLAGS_DISABLE_WRONGVIF (1 << 0) /* disable WRONGVIF
        signals */
230 #define MRT_MFC_FLAGS_BORDER_VIF (1 << 1) /* border vif */
231 #define MRT_MFC_FLAGS_KEEP_DSCP  (1 << 2) /* keep diffserv codepoint */
232 #define MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF (1 << 3)/*disable
        DSCP_NOCONF sig */
233 #define MRT_MFC_FLAGS_SLOW_DSCP_NOCONF (1 << 4) /* delayed DSCP_NOCONF
        */
234 #define MRT_MFC_RP   (1 << 8) /* enable RP address */
235 #define MRT_MFC_BW_UPCALL  (1 << 9) /* enable bw upcalls */
236 #define MRT_MFC_PERIODIC_UPCALL  (1 << 10)/* enable periodic upcall */
237 #define MRT_MFC_FLAGS_ALL  (MRT_MFC_FLAGS_DISABLE_WRONGVIF | \
238                                         MRT_MFC_FLAGS_BORDER_VIF | \
239                                         MRT_MFC_FLAGS_KEEP_DSCP | \
240                                         MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF | \
241                                         MRT_MFC_FLAGS_SLOW_DSCP_NOCONF)
242 #define MRT_API_FLAGS_ALL  (MRT_MFC_FLAGS_ALL |  \
243                                         MRT_MFC_RP |   \
244                                         MRT_MFC_BW_UPCALL |  \
245                                         MRT_MFC_PERIODIC_UPCALL)
```

Die vier NRS-Zustände werden durch die beiden MFC-Flags MRT_MFC_FLAGS_KEEP_DSCP und MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF implementiert (Zeilen 231 und 232). MRT_MFC_FLAGS_SLOW_DSCP_NOCONF (Zeile 243) ist das invertierte DRbit(S,G,Oif) und gleichzeitig das Sbit(S,G). Diese doppelte Nutzung ist möglich, da das DRbit(S,G,Oif) an der Eingangsschnittstelle nicht benötigt wird daher als Sbit(S,G) genutzt werden kann. Die Flags dienen gleichzeitig als API-Fähigkeiten (vgl. MRT_API_FLAGS_ALL, Zeilen 237–245). Die vom FreeBSD-Kern unterstützten Fähigkeiten werden mit der Socketoption MRT_API_SUPPORT (Zeile 68) abgefragt und die gewünschten Fähigkeiten mit der Socketoption MRT_API_CONFIG (Zeile 69) vom Routingdaemon aktiviert bzw. abgefragt, was aktiviert wurde.

Die Fähigkeit MRT_MFC_FLAGS_KEEP_DSCP aktiviert den vereinfachten NRS-Zustand mit zwei Zuständen Keep und Remark, wie sie im Falle von verteiltem Ressourcenmanagement genutzt werden kann. Die Fähigkeit MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF fügt die Zustände Init und Invalid und das ratenlimitierte Senden von Nachrichten IGMPMSG_DSCP_NOCONF bei Weiterleitung unter ungültigem NRS-Zustand hinzu. MRT_MFC_FLAGS_SLOW_DSCP_NOCONF ergänzt die anfängliche Unterdrückung dieser Nachrichten. Neben den drei genannten Flags definiert DSMC noch die vierte Fähigkeit MRT_MFC_PERIODIC_UPCALL, die für den TriggerSwitched genutzt wird. Die beiden letzten Fähigkeiten MRT_MFC_SLOW_DSCP_NOCONF und MRT_MFC_PERIODIC_UPCALL sind optional, sie können alternativ durch eine geänderte Implementierung

der DSMC-Komponente im Routingdaemon ersetzt werden (s. Abschnitt 5.1.3). Die vier Fähigkeiten bauen aufeinander auf, jede kann nur aktiviert werden, wenn auch alle vorangegangenen aktiviert sind. Die letzte Fähigkeit benötigt wegen der doppelten Nutzung des Flags MRT_MFC_FLAGS_SLOW_DSCP_NOCONF kein eigenes Flag, definiert aber mit IGMPMSG_PERIODIC_UPCALL einen weiteren Nachrichtentyp. Nachrichten an den Routingdaemon werden mit einer an ein IP-Paket angelehnten Struktur *igmpmsg* über den Mrouter-Socket zum Routingdaemon gesendet:

```
408 struct igmpmsg {
409     u_long unused1;
410     u_long unused2;
411     u_char im_msgtype;   /* what type of message */
412 #define IGMPMSG_NOCACHE  1 /* no MFC in the kernel  */
413 #define IGMPMSG_WRONGVIF 2 /* packet came from wrong interface */
414 #define IGMPMSG_WHOLEPKT 3 /* PIM pkt for user level encap. */
415 #define IGMPMSG_BW_UPCALL 4 /* BW monitoring upcall  */
416 #define IGMPMSG_DSCP_NOCONF 5 /* flag DISABLE_DSCP_NOCONF not set */
417 #define IGMPMSG_PERIODIC_UPCALL 6 /* flag SLOW_DSCP_NOCONF set on iif */
418     u_char im_mbz;   /* must be zero  */
419     u_char im_vif;   /* vif rec'd on  */
420     u_char unused3;
421     struct in_addr im_src, im_dst;
422 };
```

DSMC nutzt das vorhandene Nachrichtenformat unverändert weiter und definiert lediglich zwei neue Nachrichtentypen IGMPMSG_DSCP_NOCONF und IGMPMSG_PERIODIC_UPCALL (Zeilen 416 und 417), wovon der letztere optional ist.

MFC-Eintrag

Die Struktur *mfc* stellt den MFC-Eintrag dar. Sie gehört eigentlich nicht zur API, ist jedoch ebenfalls in Datei ip_mroute.h definiert:

```
383 struct mfc {
384         struct in_addr mfc_origin;  /* IP origin of mcasts */
385         struct in_addr mfc_mcastgrp;  /* multicast group associated*/
386         vifi_t  mfc_parent;  /* incoming vif */
387         u_char  mfc_ttls[MAXVIFS]; /* forwarding ttls on vifs */
388         u_long  mfc_pkt_cnt;  /* pkt count for src-grp */
389         u_long  mfc_byte_cnt;  /* byte count for src-grp */
390         u_long  mfc_wrong_if;  /* wrong if for src-grp */
391         u_long  mfc_dscp_remarked; /* codepoint changed */
392         u_long  mfc_dscp_noconf; /* forwarded with dscp_noconf*/
393         int  mfc_expire;  /* time to clean entry up */
394         struct timeval mfc_last_assert; /* last time I sent an assert*/
395         struct timeval mfc_last_dscp_noconf; /* last time I notified daemon*/
396         struct rtdetq *mfc_stall;  /* q of packets awaiting mfc */
397         struct mfc *mfc_next;  /* next mfc entry */
398         uint8_t  mfc_flags[MAXVIFS]; /* the MRT_MFC_FLAGS_* flags */
399         struct in_addr mfc_rp;   /* the RP address */
400         struct bw_meter *mfc_bw_meter;  /* list of bandwidth meters */
401         uint8_t  mfc_suppress_dscp_noconf;/*if=0 sent SLOW_DSCP_NOCONF*/
402 };
```

DSMC erweitert den MFC-Eintrag um zwei Felder *last_dscp_noconf* (Zeile 395) und *suppress_dscp_noconf* (Zeile 401), mit denen die Ratenlimitierung und Verzögerung der

Nachrichten an den Routingdaemon und damit der Trigger umgesetzt wird. Ersteres wird nur nach Aktivierung von MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF genutzt, letzteres nur nach Aktivierung von MRT_MFC_FLAGS_SLOW_DSCP_NOCONF.

B.3 Veränderte Multicastweiterleitung – ip_mroute.c

Die für DSMC durchgeführten Veränderungen an der Multicastweiterleitung betreffen im Wesentlichen die Funktion *ip_mdq* (s. Abbildung 5.1). Sie markiert Pakete um und sendet die für DSMC neu definierten Nachrichten an den Routingdaemon. Dabei greift sie auf die zwei neu eingeführten routerglobalen Strukturen *mrt_dscp* und *mrt_dscp_noconf* zu, die die je nach NRS-Zustand zu verwendenden DSCPs, die vom NRS-Zustand auszunehmenden niederprioren DSCPs sowie Rate und Verzögerung der Nachrichten an den Routingdaemon enthalten (*viftable* und *mrt_api_config* werden anschließend im Zusammenhang mit *ip_mdq* erläutert):

netinet/ip_mroute.c

```
113 static struct vif viftable[MAXVIFS];
332 static struct mrt_dscp mrt_dscp = { DSCP_UNAVAIL, DSCP_NOCONF, DSCP_SIGNAL,
333                                     DSCPMAP_LOWPRIO };
338 static struct mrt_dscp_noconf mrt_dscp_noconf = { DSCP_NOCONF_MSG_TIME,
339                                                   DSCP_NOCONF_MSG_SUPPRESS };
347 static uint32_t mrt_api_config = 0;
```

Die Werte können wie im vorigen Abschnitt beschrieben mit den zwei für DSMC neu eingeführten Socketoptionen MRT_SET_DSCP und MRT_SET_DSCP_NOCONF ausgelesen und verändert werden. Neben Veränderungen an der im Folgenden beschriebenen Funktion *ip_mdq* sind lediglich kleinere Ergänzungen zur Initialisierung der neuen Felder im MFC-Eintrag und dieser beiden globalen Strukturen sowie zum Lesen und Schreiben der Strukturen über die zwei neuen Socketoptionen nötig. Sie sorgen dafür, dass die in den Strukturen enthaltenen Werte gültig sind und die Abhängigkeiten zwischen den API-Fähigkeiten eingehalten werden. Auf eine Darstellung wird hier verzichtet.

Funktion ip_mdq

Die Funktion *ip_mdq* wird mit Zeigern auf den das Multicastpaket enthaltenen mbuf[4] *m*, die Verwaltungsstruktur der Eingangsschnittstelle *ifp*, den für die Quell- und Zieladresse des Pakets relevanten MFC-Eintrag *rt* und mit dem Ausgangsschnittstellenindex *xmt_vif* aufgerufen:

```
1799 /*
1800  * Packet forwarding routine once entry in the cache is made
1801  */
1802 static int
1803 ip_mdq(struct mbuf *m, struct ifnet *ifp, struct mfc *rt, vifi_t xmt_vif)
1804 {
1805     struct ip *ip = mtod(m, struct ip *);
1806     vifi_t vifi;
1807     int plen = ip->ip_len;
1808     u_char dscp = ip->ip_tos >> DSCP_SHIFT;
1809     u_char ecn = ip->ip_tos & ((1 << DSCP_SHIFT)-1);
```

[4]FreeBSD verwaltet Pakete im Netzwerkstack mit sogenannten mbufs (Memory Buffer).

Sofern *xmt_vif* einen gültigen Index nennt, wird der MFC umgangen und nur an die angegebene Ausgangsschnittstelle weitergeleitet (Pfeil „single outgoing interface“ in Abbildung 5.1). Zum Zeitpunkt des Aufrufs wurde bereits sichergestellt, dass ein passender MFC-Eintrag existiert bzw. das Anlegen eines MFC-Eintrags wurde mit der Nachricht IGMPMSG_NOCACHE vom Routingdaemon angefordert und von diesem durchgeführt. Solange legt der Kern einen vorläufigen MFC-Eintrag an und speichert zwischenzeitlich eintreffende Pakete in der Warteschlange *mfc_stall* (ip_mroute.h Zeile 396). Aus dem TOS-Feld des IP-Pakets werden der DSCP *dscp* und die beiden ECN-Bits *ecn* (s. Abschnitt 2.4.1) für eine spätere Verwendung extrahiert. Anschließend erfolgt der hier nicht wiedergegebene Test, ob das Paket über die erwartete Schnittstelle eintraf. Falls nicht, wird eine Nachricht IGMPMSG_WRONGVIF an den Routingdaemon erzeugt und das Paket verworfen. Falls ja, werden die Paketzähler der Eingangsschnittstelle für die Statistik aktualisiert (nicht wiedergegeben).

Nun folgt die Überprüfung auf das Sbit(S,G), ob also die zugehörige API-Fähigkeit MRT_MFC_PERIODIC_UPCALL aktiviert (Zeile 1939) und das Flag an der Eingangsschnittstelle gesetzt ist (Zeile 1940):

```
1930    struct timeval now;
1931    int time_checked = 0;
1932    long can_notify = 0;
1933
1934    /*
1935     * Send a message to the routing daemon since we have forwarded a packet
1936     * based on an mfc entry with the SLOW_DSCP_NOCONF flag set on the incoming
1937     * interface.
1938     */
1939    if ((mrt_api_config & MRT_MFC_PERIODIC_UPCALL) &&
1940        rt->mfc_flags[vifi] & MRT_MFC_FLAGS_SLOW_DSCP_NOCONF) {
1941                    /*
1942                     * The rate limit of PERIODIC_UPCALL is shared with
1943                     * DSCP_NOCONF, cf. last 'for' loop below.
1944                     */
1945                    time_checked = 1;
1946                    long delta;
1947                    GET_TIME(now);
1948                    TV_DELTA(now, rt->mfc_last_dscp_noconf, delta);
1949                    can_notify = delta / mrt_dscp_noconf.msg_time;
1950                    if (can_notify < rt->mfc_suppress_dscp_noconf)
1951                        rt->mfc_suppress_dscp_noconf -= can_notify;
1952                    else
1953                        rt->mfc_suppress_dscp_noconf = 0;
1954
1955                    if (can_notify) {
```

Bevor eine Nachricht an den Routingdaemon gesendet wird, muss sichergestellt werden, dass seit dem Senden der letzten Nachricht genug Zeit vergangen ist, *can_notify* wird entsprechend gesetzt (Ratenlimitierung, Zeilen 1947–1949). Das Makro GET_TIME setzt dabei den übergebenen Wert auf die aktuelle Zeit, das Makro TV_DELTA berechnet die Differenz der ersten beiden Parameter und legt die Differenz im dritten ab (beide Makros nicht wiedergegeben). Zusätzlich wird der Zähler *rt->mfc_suppress_dscp_noconf*, der die Unterdrückung der ersten Nachrichten implementiert, dekrementiert (Zeile 1950–1953). Er wird aber erst später in Zeile 2123 verwendet ebenso wie das Flag *time_checked*, das erst in Zeile 2097 ausgewertet wird. Erlaubt *can_notify* das Senden (Zeile 1955), wird eine

Nachricht (Struktur *igmpmsg*) mit Typ IGMPMSG_PERIODIC_UPCALL erzeugt und über den Mrouter-Socket zum Routingdaemon gesendet (nicht wiedergegeben) sowie der letzte Sendezeitpunkt identisch Zeile 2129 (s. u.) aktualisiert. Die Zeilen 1934–1987 können je nach Implementierung des Routingdaemons entfallen (s. Abschnitt 5.1.3).

Nun erfolgt die eigentliche Paketduplizierung. Sind keine der mit DSMC neu eingeführten API-Fähigkeiten aktiviert, oder handelt es sich um einen niederprioren DSCP, für den der NRS-Zustand nicht gilt, wird ohne umzumarkieren weitergeleitet:

```
1985                    }
1986    }
1987
1988    /*
1989     * no dscp processing: MRT_MFC_FLAGS_KEEP_DSCP capability not enabled,
1990     * or remarking enabled but incoming packet carries a low priority dscp.
1991     * IMPORTANT:
1992     * - check api first! dscpmap_lowprio not valid if capability not enabled.
1993     * - DSCP_UNAVAIL may or may not be set in mrt_dscp.dscpmap_lowprio.
1994     * Not setting it increases forwarding costs and changes statistics.
1995     * Externally visible forwarding behavior stays the same, though.
1996     */
1997    if (!(mrt_api_config & MRT_MFC_FLAGS_KEEP_DSCP) ||
1998        DSCPMAP_ISSET(mrt_dscp.dscpmap_lowprio,dscp)) {
1999
2000        /*
2001         * For each vif, decide if a copy of the packet should be forwarded.
2002         * Forward if:
2003         *  - the ttl exceeds the vif's threshold
2004         *  - there are group members downstream on interface
2005         */
2006        for (vifi = 0; vifi < numvifs; vifi++)
2007            if ((rt->mfc_ttls[vifi] > 0) && (ip->ip_ttl > rt->mfc_ttls[vifi])) {
2008                viftable[vifi].v_pkt_out++;
2009                viftable[vifi].v_bytes_out += plen;
2010                MC_SEND(ip, viftable+vifi, m, rt);
2011            }
```

Dies ist die klassische Multicastweiterleitung. Das Paket wird dabei nur dann auf einen Ausgang weitergeleitet, wenn der TTL-Wert im IP-Kopf größer als der konfigurierte TTL-Wert des Ausgangs ist (TTL-Scoping, s. Abschnitt 2.2.4). Ein TTL-Wert von Null kennzeichnet inaktive Ausgänge (Activebit(S,G,Oif) == FALSE). Das Makro MC_SEND (nicht wiedergegeben) ruft je nach Art der Ausgangsschnittstelle die geeignete Funktion zum Senden des Pakets auf: *phyint_send* für direktes Senden über Schicht 2, *pim_register_send* für den PIM-Register-Tunnel, wobei ggf. das Paket in der Nachricht IGMPMSG_WHOLEPKT gekapselt an den Routingdaemon weitergeleitet wird, oder *encap_send* für einen IPIP-Tunnel (s. Abbildung 5.1).

Ist nur der auf die beiden Zustände Remark und Keep vereinfachte NRS-Zustand aktiviert, ist also die API-Fähigkeit MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF nicht aktiviert (Zeile 2018), wird geprüft, ob das Flag MRT_MFC_FLAGS_KEEP_DSCP gesetzt ist (Zeile 2024) und der DSCP beibehalten werden kann (Zeile 2025), oder ob auf den in der globalen Struktur vermerkten DSCP *mrt_dscp.dscp_unavail* ummarkiert werden muss (Zeile 2031):

```
2013    /*
2014     * dscp remarking but no signals: MRT_MFC_FLAGS_KEEP_DSCP capability
```

```
2015        * enabled but MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF capability is not,
2016        * and incoming dscp is not one of the low priority dscps.
2017        */
2018       } else if (!(mrt_api_config & MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF)) {
2019
2020           for (vifi = 0; vifi < numvifs; vifi++)
2021               if ((rt->mfc_ttls[vifi] > 0) && (ip->ip_ttl > rt->mfc_ttls[vifi])) {
2022                   viftable[vifi].v_pkt_out++;
2023                   viftable[vifi].v_bytes_out += plen;
2024                   if ((rt->mfc_flags[vifi] & MRT_MFC_FLAGS_KEEP_DSCP))
2025                       ip->ip_tos = (dscp << DSCP_SHIFT) | ecn;
2026                   else {
2027                       viftable[vifi].v_pkt_remarked++;
2028                       viftable[vifi].v_bytes_remarked += plen;
2029                       ++rt->mfc_dscp_remarked;
2030                       ++mrtstat.mrts_dscp_remarked;
2031                       ip->ip_tos = (mrt_dscp.dscp_unavail << DSCP_SHIFT) | ecn;
2032                   }
2033                   MC_SEND(ip, viftable+vifi, m, rt);
2034               }
```

Andernfalls sind auch die beiden NRS-Zustände Init und Invalid sowie die Benachrichtigung des Routingdaemons bei Weiterleitung unter diesen Zuständen aktiviert. Die Weiterleitung erfolgt wie in den anderen Fällen auch mit der einer Schleife über alle Schnittstellen:

```
2036       /*
2037        * dscp remarking and (rate limited) signals: both capabilites
2038        * MRT_MFC_FLAGS_KEEP_DSCP and
                MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF
2039        * are enabled and incoming dscp is not one of the low priority dscps.
2040        * Both (DISABLE_DSCP_NOCONF, KEEP_DSCP) together represent the nrs state:
2041        * (0,0)=init, (0,1)=invalid, (1,0)=remark, (1,1)=keep
2042        * IMPORTANT:
2043        * - special dscps mrt_dscp.dscp_(unavail|noconf|signal) must not be set
2044        * in mrt_dscp.dscpmap_lowprio! This is ensured by set_dscp().
2045        */
2046       } else {
2047
2048           for (vifi = 0; vifi < numvifs; vifi++)
2049               if ((rt->mfc_ttls[vifi] > 0) && (ip->ip_ttl > rt->mfc_ttls[vifi])) {
2050                   viftable[vifi].v_pkt_out++;
2051                   viftable[vifi].v_bytes_out += plen;
```

Jetzt muss nach NRS-Zustand differenziert werden: Ist er konfiguriert (Zustände Keep oder Remark, Flag MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF ist gesetzt), wird genau wie in den Zeilen 2024–2032 beim vereinfachten NRS-Zustand vorgegangen, wobei allerdings zusätzlich auf DSCP_UC überprüft werden muss (Zeile 2058), da so markierte Pakete nicht mehr ummarkiert werden dürfen:

```
2052                   /*
2053                    * DSCP_NOCONF signal was disabled: mfc already configured,
2054                    * nrs state either keep or remark
2055                    */
2056                   if (rt->mfc_flags[vifi] & MRT_MFC_FLAGS_DISABLE_DSCP_NOCONF)
2057                       if ((rt->mfc_flags[vifi] & MRT_MFC_FLAGS_KEEP_DSCP) ||
2058                           (dscp == mrt_dscp.dscp_noconf))
```

```
2059                    /* nrs state keep */
2060                    ip->ip_tos = (dscp << DSCP_SHIFT) | ecn;
2061                else {
2062                    /* nrs state remark */
2063                    viftable[vifi].v_pkt_remarked++;
2064                    viftable[vifi].v_bytes_remarked += plen;
2065                    ++rt->mfc_dscp_remarked;
2066                    ++mrtstat.mrts_dscp_remarked;
2067                    ip->ip_tos = (mrt_dscp.dscp_unavail << DSCP_SHIFT) | ecn;
2068                }
```

Ist er noch nicht konfiguriert, wird je nach NRS-Zustand (Zeile 2083) auf *mrt_dscp.dscp_noconf* (Invalid, Zeile 2085) oder *mrt_dscp.dscp_signal* (Init, Zeile 2088) ummarkiert, oder der DSCP beibehalten (Init, Zeile 2091), falls letzterer den speziellen Wert DSCP_KEEP enthält:

```
2069            /*
2070             * DSCP_NOCONF signal still enabled: mfc not yet configured,
2071             * nrs state either init or invalid
2072             */
2073            else {
2074                viftable[vifi].v_pkt_remarked++;
2075                viftable[vifi].v_bytes_remarked += plen;
2076                viftable[vifi].v_pkt_noconf++;
2077                viftable[vifi].v_bytes_noconf += plen;
2078                ++rt->mfc_dscp_remarked;
2079                ++mrtstat.mrts_dscp_remarked;
2080                ++rt->mfc_dscp_noconf;
2081                ++mrtstat.mrts_dscp_noconf;
2082
2083                if (rt->mfc_flags[vifi] & MRT_MFC_FLAGS_KEEP_DSCP)
2084                    /* nrs state invalid, remark to dscp_noconf */
2085                    ip->ip_tos = (mrt_dscp.dscp_noconf << DSCP_SHIFT) | ecn;
2086                else if (mrt_dscp.dscp_signal != DSCP_KEEP)
2087                    /* nrs state init, remark to dscp_signal */
2088                    ip->ip_tos = (mrt_dscp.dscp_signal << DSCP_SHIFT) | ecn;
2089                else
2090                    /* nrs state init, keep dscp */
2091                    ip->ip_tos = (dscp << DSCP_SHIFT) | ecn;
```

Vor dem Senden einer Nachricht an den Routingdaemon wird exakt dieselbe Ratenlimitierung wie oben beim Sbit(S,G) durchgeführt:

```
2093            /*
2094             * Check rate only once, so the rate limit
2095             * is per incoming packet, not per outgoing packet.
2096             */
2097            if (!time_checked) {
2098                time_checked = 1;
2099                long delta;
2100                GET_TIME(now);
2101                TV_DELTA(now, rt->mfc_last_dscp_noconf, delta);
2102                can_notify = delta / mrt_dscp_noconf.msg_time;
2103                if (can_notify < rt->mfc_suppress_dscp_noconf)
2104                    rt->mfc_suppress_dscp_noconf -= can_notify;
2105                else
2106                    rt->mfc_suppress_dscp_noconf = 0;
```

```
2107                    }
```

Das Flag *time_checked* verhindert dabei, dass mehrfach geprüft bzw. der letzte Sendezeitpunkt aktualisiert wird, denn die Ratenlimitierung erfolgt mit nur einem Zeitstempel je MFCentry(S,G), soll aber separat je NRSstate(S,G,Oif) bzw. Ausgang Oif wirken. Trotz nur eines Zeitstempels ist sichergestellt, dass die konfigurierte Rate niemals überschritten wird. Wird ein zweiter Ausgang eines MFC-Eintrags aktiviert, bevor der letzte zuvor aktivierte Ausgang konfigurierten NRS-Zustand besitzt, wird die Rate beim zweiten Ausgang allerdings stärker limitiert als es eigentlich nötig wäre. Dies ist ein Kompromiss zu Gunsten des reduzierten Speicherbedarfs und des Effizienzgewinns. Zudem ist der Weiterleitungsaufwand mit max. einer Berechnung der Zeitstempel je eintreffendem Paket konstant anstatt mit bis zu einer Berechnung je weitergeleitetem Duplikat schlimmstenfalls proportional zur Anzahl aktivierter Ausgangsschnittstellen.

```
2109                    /*
2110                     * Send a message to the routing daemon since we have
2111                     * forwarded a packet marked with a high priority dscp or
2112                     * one of the special dscps dscp_(unavail|noconf|signal)
2113                     * under nrs state that is not yet configured.
2114                     *
2115                     * Always send if in init state, otherwise rate limit,
2116                     * eventually suppressing the first few messages if the
2117                     * SLOW_DSCP_NOCONF flag is set and don't send if the
2118                     * incoming packet carries dscp_noconf.
2119                     */
2120                    if ( !(rt->mfc_flags[vifi] & MRT_MFC_FLAGS_KEEP_DSCP) ||
2121                         ( can_notify &&
2122                           ( !(rt->mfc_flags[vifi] &
                                 MRT_MFC_FLAGS_SLOW_DSCP_NOCONF) ||
2123                             ( !rt->mfc_suppress_dscp_noconf &&
2124                               dscp != mrt_dscp.dscp_noconf ) ) ) ) {
```

Die Ratenlimitierung betrifft nur den Zustand Invalid. Im Zustand Init wird die Nachricht immer gesendet (Zeile 2120). Im Zustand Invalid wird bei zurückgesetztem DRbit(S,G.Oif) (gesetztes Flag MRT_MFC_FLAGS_SLOW_DSCP_NOCONF, Zeile 2122) die Nachricht noch weiter unterdrückt, sofern die Unterdrückungszeit noch nicht abgelaufen ist, der Zähler also noch nicht Null erreicht hat (Zeile 2123). Ist das eintreffende Paket bereits mit DSCP_UC markiert, wird ebenfalls keine Nachricht gesendet (Zeile 2124).

```
2139                    rt->mfc_last_dscp_noconf = now;
2140
2141                    if (!(rt->mfc_flags[vifi] &
2142                          MRT_MFC_FLAGS_KEEP_DSCP)) {
2143                        /* transition from nrs state init to invalid */
2144                        rt->mfc_flags[vifi] |= MRT_MFC_FLAGS_KEEP_DSCP;
2145                        /* make shure that we suppress long enough */
2146                        rt->mfc_suppress_dscp_noconf =
2147                            mrt_dscp_noconf.msg_suppress;
2148                    }
```

Wird eine Nachricht an den Routingdaemon gesendet (das Senden selbst ist nicht wiedergegeben), wird der Zeitstempel aktualisiert (Zeile 2139) und, sofern der momentane NRS-Zustand Init ist (Zeilen 2141–2142) in den Zustand Invalid gewechselt (Zeile 2144) sowie die

Unterdrückungszeit für die Nachrichten IGMPMSG_DSCP_NOCONF neu gestartet (Zeilen 2146–2147). Jede Nachricht IGMPMSG_DSCP_NOCONF wird die DSMC-Komponente im Routingdaemon zu einem Trigger, TriggerRpt oder TriggerUC veranlassen. Einzig die erste Nachricht beim Zustandswechsel von Init nach Invalid führt ggf. zu keinem Trigger (s. Abschnitt 5.2.2, Funktion *PimMrt::signal_dscp_noconf_message_recv*).

Nach dieser Verarbeitung wird schließlich das Paket für den Ausgang dupliziert und der nächste Schleifendurchlauf für die nächste Ausgangsschnittstelle gestartet:

```
2165                        }
2166                    }
2167 mcsend:
2168                    MC_SEND(ip, viftable+vifi, m, rt);
2169                }
2170
2171        }
```

Nachdem alle Ausgänge abgearbeitet sind, werden abschließend die Routinen zur Ratenmessung mit den im MFC-Eintrag über *mfc_bw_meter* (ip_mroute.h Zeile 400) verlinkten Strukturen aufgerufen (nicht wiedergegeben), was ggf. weitere Nachrichten IGMPMSG_BW_UPCALL an den Routingdaemon nach sich zieht.

Literaturverzeichnis

[1] Aboul-Magd, O. und S. Rabie: *A Differentiated Service Two-Rate, Three-Color Marker with Efficient Handling of in-Profile Traffic.* RFC 4115 (Informational), Juli 2005.

[2] Adams, A., J. Nicholas und W. Siadak: *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised).* RFC 3973 (Experimental), Januar 2005.

[3] Aggarwal, R., D. Papadimitriou und S. Yasukawa: *Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs).* RFC 4875 (Proposed Standard), Mai 2007.

[4] Aguilar, Lorenzo: *Datagram Routing for Internet Multicasting.* In: *SIGCOMM 84: Proceedings of the ACM SIGCOMM symposium on Communications architectures and protocols: tutorials & symposium*, Seiten 58–63, Montréal, Québec, Kanada, 6.–8. Juni 1984. ACM Press. DOI 10.1145/800056.802060.

[5] Arends, R., R. Austein, M. Larson, D. Massey und S. Rose: *Protocol Modifications for the DNS Security Extensions.* RFC 4035 (Proposed Standard), März 2005. Ersetzt RFCs 2535, 3008, 3090, 3445, 3655, 3658, 3755, 3757, 3845, aktualisiert RFCs 1034, 1035, 2136, 2181, 2308, 3225, 3007, 3597, 3226, aktualisiert durch RFC 4470.

[6] Arkko, J., J. Kempf, B. Zill und P. Nikander: *SEcure Neighbor Discovery (SEND).* RFC 3971 (Proposed Standard), März 2005.

[7] Arkko, J., C. Vogt und W. Haddad: *Enhanced Route Optimization for Mobile IPv6.* RFC 4866 (Proposed Standard), Mai 2007.

[8] Armitage, G., B. Carpenter, A. Casati, J. Crowcroft, J. Halpern, B. Kumar und J. Schnizlein: *A Delay Bound alternative revision of RFC 2598.* RFC 3248 (Informational), März 2002.

[9] As, Gerald, Attila Bader, Cornelia Kappler und David R. Oran: *QoS NSLP QSPEC Template.* Internet Draft, Work in Progress – draft-ietf-nsis-qspec-20.txt, April 2008.

[10] Aura, T.: *Cryptographically Generated Addresses (CGA).* RFC 3972 (Proposed Standard), März 2005. Aktualisiert durch RFCs 4581, 4982.

[11] Ballardie, A.: *Core Based Trees (CBT) Multicast Routing Architecture.* RFC 2201 (Historic), September 1997.

[12] Ballardie, A.: *Core Based Trees (CBT version 2) Multicast Routing – Protocol Specification –.* RFC 2189 (Historic), September 1997.

[13] Ballardie, Tony, Paul Francis und Jon Crowcroft: *Core based trees (CBT).* In: *SIGCOMM 93: Conference proceedings on Communications architectures, protocols and applications*, Seiten 85–95, San Francisco, California, USA, 13.–17. September 1993. ACM Press. DOI 10.1145/166237.166246.

[14] Banerjee, Suman, Bobby Bhattacharjee und Christopher Kommareddy: *Scalable Application Layer Multicast.* In: *SIGCOMM 02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, Seiten 205–217, Pittsburgh, Pennsylvania, USA, 19.–23. August 2002. ACM Press. DOI 10.1145/633025.633045.

[15] Banerjee, Suman, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee und Samir Khuller: *OMNI: An Efficient Overlay Multicast Infrastructure for Real-time Applications.* Computer Networks, 50(6):826–841, April 2006. Special Issue on Overlay Distribution Structures and their Applications. DOI 10.1016/j.comnet.2005.07.023.

[16] Bates, T., R. Chandra, D. Katz und Y. Rekhter: *Multiprotocol Extensions for BGP-4.* RFC 4760 (Draft Standard), Januar 2007. Ersetzt RFC 2858.

[17] Bates, T., E. Chen und R. Chandra: *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP).* RFC 4456 (Draft Standard), April 2006. Ersetzt RFCs 2796, 1966.

[18] Berger, L., I. Bryskin, A. Zinin und R. Coltun: *The OSPF Opaque LSA Option.* RFC 5250 (Proposed Standard), Juli 2008. Ersetzt RFC 2370.

[19] Bernet, Y., S. Blake, D. Grossman und A. Smith: *An Informal Management Model for Diffserv Routers.* RFC 3290 (Informational), Mai 2002.

[20] Bhaskar, N., A. Gall, J. Lingard und S. Venaas: *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM).* RFC 5059 (Proposed Standard), Januar 2008. Ersetzt RFC 2362, aktualisiert RFC 4601.

[21] Bianchi, Giuseppe, Nicola Blefari-Melazzi, Giuliano Bonafede und Emiliano Tintinelli: *QUASIMODO: Quality of Service-Aware Multicasting over DiffServ and Overlay Networks.* IEEE Network, 17(1):28–45, Februar 2003. DOI 10.1109/MNET.2003.1174176.

[22] Bianchi, Giuseppe, Nicola Blefari-Melazzi, M. Femminella und F. Pugini: *Performance Evaluation of a Measurement-Based Algorithm for Distributed Admission Control in a DiffServ Framework.* In: *Global Telecommunications Conference, 2001. GLOBECOM '01*, Band 3, Seiten 1886–1891. IEEE, November 2001. DOI 10.1109/GLOCOM.2001.965902.

[23] Billhartz, T., J. Bibb Cain, Ellen Farrey-Goudreau, Doug Fieg und Stephen Gordon Batsell: *Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols.* IEEE Journal on Selected Areas in Communications, 15(3):304–315, April 1997. DOI 10.1109/49.564130.

[24] Black, D.: *Differentiated Services and Tunnels.* RFC 2983 (Informational), Oktober 2000.

[25] Blake, S., D. Black, M. Carlson, E. Davies, Z. Wang und W. Weiss: *An Architecture for Differentiated Service.* RFC 2475 (Informational), Dezember 1998. Aktualisiert durch RFC 3260.

[26] Bless, R., K. Nichols und K. Wehrle: *A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services.* RFC 3662 (Informational), Dezember 2003.

[27] Bless, R. und K. Wehrle: *IP Multicast in Differentiated Services (DS) Networks.* RFC 3754 (Informational), April 2004.

[28] Bless, Roland: *Integriertes Management qualitätsbasierter Internetkommunikationsdienste.* Doktorarbeit, Universität Karlsruhe (TH), Dezember 2002.

[29] Bless, Roland, Mark Doll, Klaus Wehrle und Martina Zitterbart: *Diffserv-basierte Dienstgüte im Internet der nächsten Generation.* PIK – Praxis in der Informationsverarbeitung und Kommunikation, 25(2):104–111, April–Juni 2002. http://doc.tm.uka.de/2002/zit-pik-2002-02-diffserv.pdf.

[30] Boivie, Rick, Nancy Feldman, Yuji Imai, Wim Livens, Dirk Ooms und Olivier Paridaens: *Explicit Multicast (Xcast) Basic Specification.* Internet Draft, Work in Progress – draft-ooms-xcast-basic-spec-00.txt, Dezember 2000.

[31] Bonaventure, O. und S. De Cnodder: *A Rate Adaptive Shaper for Differentiated Services.* RFC 2963 (Informational), Oktober 2000.

[32] Braden, R., D. Clark und S. Shenker: *Integrated Services in the Internet Architecture: an Overview.* RFC 1633 (Informational), Juni 1994.

[33] Braden, R., L. Zhang, S. Berson, S. Herzog und S. Jamin: *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification.* RFC 2205 (Proposed Standard), September 1997. Aktualisiert durch RFCs 2750, 3936, 4495.

[34] Breslau, Lee, Edward W. Knightly, Scott Shenker, Ion Stoica und Hui Zhang: *Endpoint Admission Control: Architectural Issues and Performance.* In: *SIGCOMM 00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Seiten 57–69, Stockholm, Schweden, 28. August – 1. September 2000. ACM Press. DOI 10.1145/347059.347400.

[35] Cain, B., S. Deering, I. Kouvelas, B. Fenner und A. Thyagarajan: *Internet Group Management Protocol, Version 3.* RFC 3376 (Proposed Standard), Oktober 2002. Ersetzt RFC 2236, aktualisiert durch RFC 4604.

[36] Callon, R.W.: *Use of OSI IS-IS for routing in TCP/IP and dual environments.* RFC 1195 (Proposed Standard), Dezember 1990. Aktualisiert durch RFC 1349.

[37] Calvert, Kenneth L., James Griffioen, Billy Mullins, Amit Sehgal und Su Wen: *Concast: Design and Implementation of an Active Network Service.* IEEE Journal on Selected Area in Communications, 19(3):426–437, März 2001. DOI 10.1109/49.917704.

[38] Case, J., R. Mundy, D. Partain und B. Stewart: *Introduction and Applicability Statements for Internet-Standard Management Framework.* RFC 3410 (Informational), Dezember 2002. Ersetzt RFC 2570.

[39] Charny, A., J. Bennet, K. Benson, J. Boudec, A. Chiu, W. Courtney, S. Davari, V. Firoiu, C. Kalmanek und K. Ramakrishnan: *Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior).* RFC 3247 (Informational), März 2002.

[40] Charzinski, Joachim, Uwe Walter und Martina Zitterbart: *Architecture of a Network Control Server for autonomous and efficient operation of Next Generation Networks.* In: *International Conference on Telecommunication Systems, Modeling and Analysis (ICTSM 2005)*, Seiten 117–134, Dallas, Texas, USA, November 2005. http://doc.tm.uka.de/2005/WaZiCh_NCS_Architecture_ICTSMA_final.pdf.

[41] Cheriton, David R. und Stephen E. Deering: *Host Groups: A Multicast Extension for Datagram Internetworks.* In: *SIGCOMM 85: Proceedings of the ninth symposium on Data communications*, Seiten 172–179, Whistler Moutain, British Columbia, Kanada, 10.–12. September 1985. ACM Press. DOI 10.1145/319056.319039.

[42] Chu, Yang hua, Sanjay G. Rao und Hui Zhang: *A Case for End System Multicast.* In: *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, Seiten 1–12, Santa Clara, California, USA, Juni 2000. ACM Press. DOI 10.1145/339331.339337.

[43] Coltun, R., D. Ferguson, J. Moy und A. Lindem: *OSPF for IPv6.* RFC 5340 (Proposed Standard), Juli 2008. Ersetzt RFC 2740.

[44] Conta, A., S. Deering und M. Gupta: *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* RFC 4443 (Draft Standard), März 2006. Ersetzt RFC 2463, aktualisiert RFC 2780, aktualisiert durch RFC 4884.

[45] Dabbous, Christophe Diot Walid und Jon Crowcroft: *Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms.* IEEE Journal on Selected Areas in Communications, 15(3):277–290, April 1997. DOI 10.1109/49.564128.

[46] Dalal, Yogen K. und Robert M. Metcalfe: *Reverse Path Forwarding of Broadcast Packets.* Communications of the ACM, 21(12):1040–1048, 12 1978. DOI 10.1145/359657.359665.

[47] Davie, B., A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu und D. Stiliadis: *An Expedited Forwarding PHB (Per-Hop Behavior).* RFC 3246 (Proposed Standard), März 2002. Ersetzt RFC 2598.

[48] Deering, S. und R. Hinden: *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460 (Draft Standard), Dezember 1998. Ersetzt RFC 1883, aktualisiert durch RFC 5095.

[49] Deering, S.E.: *Host extensions for IP multicasting*. RFC 1112 (Standard), August 1989. Ersetzt RFCs 988, 1054, aktualisiert durch RFC 2236.

[50] Deering, S.E. und D.R. Cheriton: *Host groups: A multicast extension to the Internet Protocol*. RFC 966, Dezember 1985. Ersetzt durch RFC 988.

[51] Deering, Stephen, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu und Liming Wei: *An Architecture for Wide-Area Multicast Routing*. In: *SIGCOMM 94: Proceedings of the conference on Communications architectures, protocols and applications*, Seiten 126–135, London, Großbritannien, 31. August – 2. September 1994. ACM Press. DOI 10.1145/190314.190326.

[52] Deering, Stephen E.: *Multicast Routing in Internetworks and Extended LANs*. In: *SIGCOMM 88: Symposium proceedings on Communications architectures and protocols*, Seiten 55–64, Stanford, California, USA, 16.–18. August 1988. ACM Press. DOI 10.1145/52324.52331.

[53] Deering, Stephen E. und David R. Cheriton: *Multicast Routing in Datagram Internetworks and Extended LANs*. ACM Transactions on Computer Systems (TOCS), 8(2):85–110, Mai 1990. DOI 10.1145/78952.78953.

[54] Doll, Mark: *Errata to RFC 3973*, Mai 2007.

[55] Doll, Mark: *Source Routed Multicast: Ein Multicast-Routing-Header für IP Version 6*. In: Braun, Torsten, Georg Carle und Burkhard Stiller (Herausgeber): *KiVS 2007 – Kommunikation in Verteilten Systemen*, Seiten 187–192. VDE Verlag, Berlin, Februar 2007. 15. ITG/GI-Fachtagung vom 26. Februar bis 2. März 2007 in Bern, Schweiz. Industriebeiträge, Kurzbeiträge und Workshops, http://doc.tm.uka.de/2007/doll-mcasthdr-kivs2007.pdf.

[56] Droms, R.: *Dynamic Host Configuration Protocol*. RFC 2131 (Draft Standard), März 1997. Ersetzt RFC 1541, aktualisiert durch RFCs 3396, 4361.

[57] Droms, R., J. Bound, B. Volz, T. Lemon, C. Perkins und M. Carney: *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. RFC 3315 (Proposed Standard), Juli 2003. Aktualisiert durch RFC 4361.

[58] Durand, Jerome und Pekka Savola: *IPv6 multicast address assignment with DHCPv6*. Internet Draft, Work in Progress – draft-jdurand-assign-addr-ipv6-multicast-dhcpv6-01.txt, Februar 2005.

[59] Durand, Jerome und Pekka Savola: *Route Advertisement Option for IPv6 Multicast Prefixes*. Internet Draft, Work in Progress – draft-jdurand-ipv6-multicast-ra-00.txt, Februar 2005.

[60] Egger, Stefan und Torsten Braun: *Multicast for Small Conferences: A Scalable Multicast Mechanism Based on IPv6*. Communications Magazine, IEEE, 42(1):121–126, Januar 2004. DOI 10.1109/MCOM.2004.1262171.

[61] Estrin, D., D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma und L. Wei: *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification.* RFC 2117 (Experimental), Juni 1997. Ersetzt durch RFC 2362.

[62] *Explicit Multicast.* http://www.watersprings.org/links/xcast/.

[63] Faloutsos, Michalis, Petros Faloutsos und Christos Faloutsos: *On Power-Law Relationships of the Internet Topology.* In: *SIGCOMM 99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Seiten 251–262, Cambridge, Massachusetts, USA, 30. August – 3. September 1999. ACM Press. DOI 10.1145/316188.316229.

[64] Fang, W., N. Seddigh und B. Nandy: *A Time Sliding Window Three Colour Marker (TSWTCM).* RFC 2859 (Experimental), Juni 2000.

[65] Farinacci, D. und Y. Cai: *Anycast-RP Using Protocol Independent Multicast (PIM).* RFC 4610 (Proposed Standard), August 2006.

[66] Fenner, B., M. Handley, H. Holbrook und I. Kouvelas: *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised).* RFC 4601 (Proposed Standard), August 2006. Ersetzt RFC 2362, aktualisiert durch RFC 5059.

[67] Fenner, B., H. He, B. Haberman und H. Sandick: *Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying").* RFC 4605 (Proposed Standard), August 2006.

[68] Fenner, B. und D. Meyer: *Multicast Source Discovery Protocol (MSDP).* RFC 3618 (Experimental), Oktober 2003.

[69] *FreeBSD.* http://www.freebsd.org/.

[70] Grossman, D.: *New Terminology and Clarifications for Diffserv.* RFC 3260 (Informational), April 2002. Aktualisiert RFCs 2474, 2475, 2597.

[71] Gulbrandsen, A., P. Vixie und L. Esibov: *A DNS RR for specifying the location of services (DNS SRV).* RFC 2782 (Proposed Standard), Februar 2000. Ersetzt RFC 2052.

[72] Haberman, B.: *Allocation Guidelines for IPv6 Multicast Addresses.* RFC 3307 (Proposed Standard), August 2002.

[73] Haberman, B. und D. Thaler: *Unicast-Prefix-based IPv6 Multicast Addresses.* RFC 3306 (Proposed Standard), August 2002. Aktualisiert durch RFCs 3956, 4489.

[74] Hancock, R., G. Karagiannis, J. Loughney und S. Van den Bosch: *Next Steps in Signaling (NSIS): Framework.* RFC 4080 (Informational), Juni 2005.

[75] Handley, M., I. Kouvelas, T. Speakman und L. Vicisano: *Bidirectional Protocol Independent Multicast (BIDIR-PIM).* RFC 5015 (Proposed Standard), Oktober 2007.

[76] Handley, Mark, Jon Crowcroft und Ian Wakeman: *Hierarchical Protocol Independent Multicast (HPIM).* ftp://cs.ucl.ac.uk/darpa/hpim.ps.gz, November 1995.

[77] Hanna, S., B. Patel und M. Shah: *Multicast Address Dynamic Client Allocation Protocol (MADCAP).* RFC 2730 (Proposed Standard), Dezember 1999.

[78] Heinanen, J., F. Baker, W. Weiss und J. Wroclawski: *Assured Forwarding PHB Group.* RFC 2597 (Proposed Standard), Juni 1999. Aktualisiert durch RFC 3260.

[79] Heinanen, J. und R. Guerin: *A Single Rate Three Color Marker.* RFC 2697 (Informational), September 1999.

[80] Heinanen, J. und R. Guerin: *A Two Rate Three Color Marker.* RFC 2698 (Informational), September 1999.

[81] Hinden, R. und S. Deering: *IP Version 6 Addressing Architecture.* RFC 4291 (Draft Standard), Februar 2006. Ersetzt RFC 3513.

[82] Holbrook, H. und B. Cain: *Source-Specific Multicast for IP.* RFC 4607 (Proposed Standard), August 2006.

[83] Holbrook, H., B. Cain und B. Haberman: *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast.* RFC 4604 (Proposed Standard), August 2006. Aktualisiert RFCs 3376, 3810.

[84] Holbrook, Hugh W. und David R. Cheriton: *IP multicast channels: EXPRESS Support for Large-scale Single-source Applications.* In: *SIGCOMM 99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Seiten 65–78, Cambridge, Massachusetts, USA, 30. August – 3. September 1999. ACM Press. DOI 10.1145/316188.316207.

[85] Hopps, C.: *Analysis of an Equal-Cost Multi-Path Algorithm.* RFC 2992 (Informational), November 2000.

[86] Hopps, Christian E.: *Routing IPv6 with IS-IS.* Internet Draft, Work in Progress – draft-ietf-isis-ipv6-07.txt, Oktober 2007.

[87] Huston, Geoff: *BGP Routing Table Analysis Reports.* http://bgp.potaroo.net/.

[88] *Address Family Numbers.* http://www.iana.org/assignments/address-family-numbers. Internet Assigned Numbers Authority.

[89] *Differentiated Services Field Codepoints.* http://www.iana.org/assignments/dscp-registry. Internet Assigned Numbers Authority.

[90] *Internet Multicast Addresses.* http://www.iana.org/assignments/multicast-addresses. Internet Assigned Numbers Authority.

[91] *Internet Protocol Version 6 Multicast Addresses.* http://www.iana.org/assignments/ipv6-multicast-addresses. Internet Assigned Numbers Authority.

[92] *Protocol Numbers*. http://www.iana.org/assignments/protocol-numbers. Internet Assigned Numbers Authority.

[93] *IEEE Standard 802 – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-2007, 2007. DOI 10.1109/IEEESTD.2007.373646.

[94] *IEEE Standard for Local and metropolitan area networks: Virtual Bridged Local Area Networks*. IEEE Std 802.1Q-2005, 2006. http://ieeexplore.ieee.org/servlet/opac?punumber=10905.

[95] *IEEE Standard 802 – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*. IEEE Std 802.3-2005, 2005. http://ieeexplore.ieee.org/servlet/opac?punumber=10531.

[96] *Information technology – Telecommunications and information exchange between systems – Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)*. ISO/IEC 10589:2002, 2002.

[97] *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*. ISO/IEC 8824-1:2002, 2002.

[98] Katz, D., K. Kompella und D. Yeung: *Traffic Engineering (TE) Extensions to OSPF Version 2*. RFC 3630 (Proposed Standard), September 2003. Aktualisiert RFC 2370, aktualisiert durch RFC 4203.

[99] Katz, David: *IS-IS and OSPF: A Comparative Anatomy*. http://www.nanog.org/mtg-0006/ppt/katz/, Juni 2000. Präsentation auf der NANOG19.

[100] Kent, S. und K. Seo: *Security Architecture for the Internet Protocol*. RFC 4301 (Proposed Standard), Dezember 2005. Ersetzt RFC 2401.

[101] Kohler, E., M. Handley und S. Floyd: *Datagram Congestion Control Protocol (DCCP)*. RFC 4340 (Proposed Standard), März 2006.

[102] Krüger, Gerhard und Dietrich Reschke (Herausgeber): *Lehr- und Übungsbuch Telematik*. Fachbuchverlag im Carl-Hanser-Verlag, München, Wien, 2000.

[103] Kumar, Satish, Pavlin Radoslavov, David Thaler, Cengiz Alaettinoğlu, Deborah Estrin und Mark Handley: *The MASC/BGMP Architecture for Interdomain Multicast Routing*. In: *SIGCOMM 98: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, Seiten 93–104, 31. August – 4. September 1998, Vancouver, British Columbia, Kanada. ACM Press. DOI 10.1145/285237.285264.

[104] Küfner, Tobias, Mark Doll, Götz Lichtwald und Martina Zitterbart: *Speeding up Transaction-oriented Communications in the Internet*. In: Dadam, Peter und Manfred Reichert (Herausgeber): *GI-Edition – Lecture Notes in Informatics: Informatik 2004 – Informatik verbindet*, Band 2, Seiten 256–260, September 2004. 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 20.–24. September 2004, Ulm, Germany, http://doc.tm.uka.de/2004/kuefner-qf-pdb-mcmi2004.pdf.

[105] Lao, Li, Jun-Hong Cui und Mario Gerla: *TOMA: A Viable Solution for Large-Scale Multicast Service Support.* In: *4th International IFIP-TC6 Networking Conference. Proceedings*, Band 3462/2005 der Reihe *Lecture Notes in Computer Science*, Seiten 906–917, Waterloo, Canada, Mai 2005. DOI 10.1007/11422778_73.

[106] Larzon, L-A., M. Degermark, S. Pink, L-E. Jonsson und G. Fairhurst: *The Lightweight User Datagram Protocol (UDP-Lite).* RFC 3828 (Proposed Standard), Juli 2004.

[107] *Diskussion auf der Mailingliste der MAGMA-Arbeitsgruppe der IETF bzgl. RFC 4605.* http://www.ietf.org/mail-archive/web/magma/, September 2007.

[108] Malkin, G.: *RIP Version 2.* RFC 2453 (Standard), November 1998. Ersetzt RFC 1723, aktualisiert durch RFC 4822.

[109] Malkin, G. und R. Minnear: *RIPng for IPv6.* RFC 2080 (Proposed Standard), Januar 1997.

[110] Manner, Jukka, Georgios Karagiannis und Andrew McDonald: *NSLP for Quality-of-Service Signaling.* Internet Draft, Work in Progress – draft-ietf-nsis-qos-nslp-16.txt, Februar 2008.

[111] McAuley, Anthony J. und Paul Francis: *Fast routing table lookup using CAMs.* In: *INFOCOM '93. Proceedings.Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future*, Band 3, Seiten 1382–1391, San Francisco, California, USA, April 1993. DOI 10.1109/INFCOM.1993.253403.

[112] McCanne, Steven, Van Jacobson und Martin Vetterli: *Receiver-driven Layered Multicast.* In: *SIGCOMM 96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, Seiten 117–130, Palo Alto, California, USA, 26.–30. August 1996. ACM Press. DOI 10.1145/248156.248168.

[113] McCloghrie, K., D. Perkins und J. Schoenwaelder: *Structure of Management Information Version 2 (SMIv2).* RFC 2578 (Standard), April 1999. Ersetzt RFC 1902.

[114] Mealling, M.: *Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database.* RFC 3403 (Proposed Standard), Oktober 2002. Ersetzt RFCs 2915, 2168.

[115] Medina, Alberto, Anukool Lakhina, Ibrahim Matta und John Byers: *BRITE: an approach to universal topology generation.* In: *Ninth IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'01)*, Seiten 346–353, Cincinnati, Ohio, USA, August 2001. ACM Press. DOI 10.1109/MASCOT.2001.948886.

[116] Menth, Michael: *Efficient Admission Control and Routing for Resilient Communication Networks.* Doktorarbeit, Julius-Maximilians-Universität Würzburg, Juli 2004.

[117] Metzler, Bernard, Till Harbaum, Ralph Wittmann und Martina Zitterbart: *AMnet: Heterogeneous Multicast Services Based on Active Networking.* In: *1999 IEEE Second Conference on Open Architectures and Network Programming Proceedings. OPENARCH '99*, Seiten 98–105, New York, New York, USA, März 1999. IEEE Communications Society. DOI 10.1109/OPNARC.1999.758560.

[118] Meyer, D.: *Administratively Scoped IP Multicast.* RFC 2365 (Best Current Practice), Juli 1998.

[119] Meyer, D. und P. Lothberg: *GLOP Addressing in 233/8.* RFC 3180 (Best Current Practice), September 2001. Ersetzt RFC 2770.

[120] Mockapetris, P.V.: *Domain names - concepts and facilities.* RFC 1034 (Standard), November 1987. Ersetzt RFCs 973, 882, 883, aktualisiert durch RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592.

[121] Mockapetris, P.V.: *Domain names - implementation and specification.* RFC 1035 (Standard), November 1987. Ersetzt RFCs 973, 882, 883, aktualisiert durch RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343.

[122] Moy, J.: *Multicast Extensions to OSPF.* RFC 1584 (Historic), März 1994.

[123] Moy, J.: *OSPF Version 2.* RFC 2328 (Standard), April 1998. Ersetzt RFC 2178.

[124] Narten, T., R. Draves und S. Krishnan: *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* RFC 4941 (Draft Standard), September 2007. Ersetzt RFC 3041.

[125] Narten, T., E. Nordmark, W. Simpson und H. Soliman: *Neighbor Discovery for IP version 6 (IPv6).* RFC 4861 (Draft Standard), September 2007. Ersetzt RFC 2461.

[126] Nichols, K., S. Blake, F. Baker und D. Black: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.* RFC 2474 (Proposed Standard), Dezember 1998. Ersetzt RFCs 1455, 1349, aktualisiert durch RFCs 3168, 3260.

[127] Nichols, K. und B. Carpenter: *Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification.* RFC 3086 (Informational), April 2001.

[128] Nichols, K., V. Jacobson und L. Zhang: *A Two-bit Differentiated Services Architecture for the Internet.* RFC 2638 (Informational), Juli 1999.

[129] Nichols, Kathleen, Van Jacobson und Kedarnath Poduri: *A Per-Domain Behavior for Circuit Emulation in IP Networks.* SIGCOMM Computer Communication Review, 34(2):71–83, 2004. DOI 10.1145/997150.997158.

[130] *OMNeT++.* http://www.omnetpp.org/.

[131] Ooms, D., B. Sales, W. Livens, A. Acharya, F. Griffoul und F. Ansari: *Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment.* RFC 3353 (Informational), August 2002.

[132] Ottmann, Thomas und Peter Widmayer: *Algorithmen und Datenstrukturen*. BI-Wissenschafts-Verlag, Mannheim, Leipzig, Wien, Zürich, 2 Auflage, 1993.

[133] Park, J-S., M-K. Shin und H-J. Kim: *A Method for Generating Link-Scoped IPv6 Multicast Addresses*. RFC 4489 (Proposed Standard), April 2006. Aktualisiert RFC 3306.

[134] Partridge, C. und A. Jackson: *IPv6 Router Alert Option*. RFC 2711 (Proposed Standard), Oktober 1999.

[135] Perlman, Radia, Cheng-Yin Lee, Tony Ballardie, Jon Crowcroft, Zheng Wang, Thomas Maufer, Christophe Diot Joseph Thoo und Mark Green: *Simple Multicast: A Design for Simple, Low-Overhead Multicast*. Internet Draft, Work in Progress – draft-perlman-simple-multicast-03.txt, Oktober 1999.

[136] *Diskussion zwischen Yi Zhao <edrt@citiz.net> und Mark Doll auf der Mailingliste der PIM-Arbeitsgruppe der IETF*. http://www.ietf.org/mail-archive/web/pim/, September 2004.

[137] Plummer, D.: *Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*. RFC 826 (Standard), November 1982. Aktualisiert durch RFC 5227.

[138] Postel, J.: *User Datagram Protocol*. RFC 768 (Standard), August 1980.

[139] Postel, J.: *Internet Protocol*. RFC 791 (Standard), September 1981. Ersetzt RFC 760, aktualisiert durch RFC 1349.

[140] Postel, J.: *Transmission Control Protocol*. RFC 793 (Standard), September 1981. Aktualisiert durch RFC 3168.

[141] Postel, J. und J.K. Reynolds: *Telnet Protocol Specification*. RFC 854 (Standard), Mai 1983. Ersetzt RFC 764, aktualisiert durch RFC 5198.

[142] Pras, Aiko und Jürgen Schönwälder (Herausgeber): *Standards Summary*. The Simple Times. The Quarterly Newsletter of SNMP Technology, Comment, and Events., 10(1), Dezember 2002. http://www.simple-times.org/pub/simple-times/issues/10-1.html#standards.

[143] Przygienda, T., N. Shen und N. Sheth: *M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)*. RFC 5120 (Proposed Standard), Februar 2008.

[144] Psenak, P., S. Mirtorabi, A. Roy, L. Nguyen und P. Pillay-Esnault: *Multi-Topology (MT) Routing in OSPF*. RFC 4915 (Proposed Standard), Juni 2007.

[145] Pusateri, Thomas: *Distance Vector Multicast Routing Protocol*. Internet Draft, Work in Progress – draft-ietf-idmr-dvmrp-v3-11.txt, Oktober 2003.

[146] Pusateri, Thomas: *Distance Vector Multicast Routing Protocol Applicability Statement*. Internet Draft, Work in Progress – draft-ietf-idmr-dvmrp-v3-as-01.txt, Mai 2004.

[147] Radoslavov, P., D. Estrin, R. Govindan, M. Handley, S. Kumar und D. Thaler: *The Multicast Address-Set Claim (MASC) Protocol*. RFC 2909 (Experimental), September 2000.

[148] Rajahalme, J., A. Conta, B. Carpenter und S. Deering: *IPv6 Flow Label Specification*. RFC 3697 (Proposed Standard), März 2004.

[149] Ramakrishnan, K., S. Floyd und D. Black: *The Addition of Explicit Congestion Notification (ECN) to IP*. RFC 3168 (Proposed Standard), September 2001. Ersetzt RFC 2481, aktualisiert RFCs 2474, 2401, 793.

[150] Rekhter, Y., T. Li und S. Hares: *A Border Gateway Protocol 4 (BGP-4)*. RFC 4271 (Draft Standard), Januar 2006. Ersetzt RFC 1771.

[151] Retana, A., L. Nguyen, R. White, A. Zinin und D. McPherson: *OSPF Stub Router Advertisement*. RFC 3137 (Informational), Juni 2001.

[152] Rosen, E., A. Viswanathan und R. Callon: *Multiprotocol Label Switching Architecture*. RFC 3031 (Proposed Standard), Januar 2001.

[153] Savola, P.: *Overview of the Internet Multicast Routing Architecture*. RFC 5110 (Informational), Januar 2008.

[154] Savola, P. und B. Haberman: *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*. RFC 3956 (Proposed Standard), November 2004. Aktualisiert RFC 3306.

[155] Savola, Pekka: *Overview of the Internet Multicast Addressing Architecture*. Internet Draft, Work in Progress – draft-ietf-mboned-addrarch-05.txt, Oktober 2006.

[156] Schmitt, Jens, Frank Zdarsky, Martin Karsten und Ralf Steinmetz: *Heterogeneous Multicast in Heterogeneous QoS Networks*. In: *Proceedings Ninth IEEE International Conference on Networks. ICON 2001*, Seiten 349–354, Bangkok, Thailand, Oktober 2001. IEEE Computer Society. DOI 10.1109/ICON.2001.962366.

[157] Schoenwaelder, J.: *Overview of the 2002 IAB Network Management Workshop*. RFC 3535 (Informational), Mai 2003.

[158] Schulzrinne, H., S. Casner, R. Frederick und V. Jacobson: *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550 (Standard), Juli 2003. Ersetzt RFC 1889.

[159] Schulzrinne, Henning und Robert Hancock: *GIST: General Internet Signalling Transport*. Internet Draft, Work in Progress – draft-ietf-nsis-ntlp-15.txt, Februar 2008.

[160] Sharma, Puneet, Deborah Estrin, Sally Floydy und Van Jacobsony: *Scalable Timers for Soft State Protocols*. In: *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, Band 1, Seiten 222–229, Kobe, Japan, April 1997. DOI 10.1109/INFCOM.1997.635133.

[161] Shields, Clay: *Secure Hierarchical Multicast Routing and Multicast Internet Anonymity*. Doktorarbeit, Computer Engineering, University of California, Santa Cruz, California, USA, Juni 1999. http://www.soe.ucsc.edu/research/ccrg/publications/clay.phd.pdf.

[162] Shields, Clay und J. J. Garcia-Luna-Aceves: *The Ordered Core Based Tree Protocol*. In: *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, Band 2, Seiten 884–891, Kobe, Japan, April 1997. DOI 10.1109/INFCOM.1997.644571.

[163] Shields, Clay und J. J. Garcia-Luna-Aceves: *The HIP protocol for Hierarchical Multicast Routing*. In: *PODC '98: Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, Seiten 257–266, Puerto Vallarta, Mexico, Juni 1998. ACM Press. DOI 10.1145/277697.277744.

[164] Sivaramu, R., J. Lingard, D. McWalter, B. Joshi und A. Kessler: *Protocol Independent Multicast MIB*. RFC 5060 (Proposed Standard), Januar 2008.

[165] Stewart, R.: *Stream Control Transmission Protocol*. RFC 4960 (Proposed Standard), September 2007. Ersetzt RFCs 2960, 3309.

[166] Stewart III, John W.: *BGP4: Inter-Domain Routing in the Internet*. Addison Wesley, 1998.

[167] Stiemerling, Martin, Hannes Tschofenig, Cedric Aoun und Elwyn Davies: *NAT/Firewall NSIS Signaling Layer Protocol (NSLP)*. Internet Draft, Work in Progress – draft-ietf-nsis-nslp-natfw-18.txt, Februar 2008.

[168] Striegel, Aaron und Govindarasu Manimaran: *A Scalable Approach for DiffServ Multicasting*. In: *ICC 2001. The IEEE International Conference on Communications*, Band 8, Seiten 2327–2331, Helsinki, Finland, Juni 2001. DOI 10.1109/ICC.2001.936538.

[169] Thaler, D.: *Interoperability Rules for Multicast Routing Protocols*. RFC 2715 (Informational), Oktober 1999.

[170] Thaler, D.: *Border Gateway Multicast Protocol (BGMP): Protocol Specification*. RFC 3913 (Historic), September 2004.

[171] Thaler, D., B. Fenner und B. Quinn: *Socket Interface Extensions for Multicast Source Filters*. RFC 3678 (Informational), Januar 2004.

[172] Thaler, D., M. Handley und D. Estrin: *The Internet Multicast Address Allocation Architecture*. RFC 2908 (Informational), September 2000.

[173] Thaler, D. und C. Hopps: *Multipath Issues in Unicast and Multicast Next-Hop Selection*. RFC 2991 (Informational), November 2000.

[174] Thaler, Dave: *Unicast-Prefix-based IPv4 Multicast Addresses*. Internet Draft, Work in Progress – draft-ietf-mboned-ipv4-uni-based-mcast-05.txt, Februar 2008.

[175] Thaler, David und Mark Handley: *On the Aggregatability of Multicast Forwarding State*. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, Band 3, Seiten 1654–1663, Tel Aviv, Israel, März 2000. DOI 10.1109/INFCOM.2000.832564.

[176] Thaler, David G. und Chinya V. Ravishankar: *Using Name-Based Mappings to Increase Hit Rates*. IEEE Transactions on Networking, 6(1):1–14, Februar 1998. DOI 10.1109/90.663936.

[177] Thomson, S., C. Huitema, V. Ksinant und M. Souissi: *DNS Extensions to Support IP Version 6*. RFC 3596 (Draft Standard), Oktober 2003. Ersetzt RFCs 3152, 1886.

[178] Thomson, S., T. Narten und T. Jinmei: *IPv6 Stateless Address Autoconfiguration*. RFC 4862 (Draft Standard), September 2007. Ersetzt RFC 2462.

[179] Thyagarajan, Ajit S. und Stephen E. Deering: *Hierarchical Distance-Vector Multicast Routing for the MBone*. In: *SIGCOMM 95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Seiten 60–66, Cambridge, Massachusetts, USA, 28. August – 1. September 1995. ACM Press. DOI 10.1145/217382.217411.

[180] Traina, P., D. McPherson und J. Scudder: *Autonomous System Confederations for BGP*. RFC 5065 (Draft Standard), August 2007. Ersetzt RFC 3065.

[181] Vida, R. und L. Costa: *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*. RFC 3810 (Proposed Standard), Juni 2004. Aktualisiert RFC 2710, aktualisiert durch RFC 4604.

[182] Vijayananda, J., Jayanth Chennamangalam, Krishna Agarwal und Prashant Jhingran: *Internet Group Management Protocol (IGMP)/ Multicast Listener Discovery (MLD) Proxy Upstream Interface Learning*. Internet Draft, Work in Progress – draft-vijay-magma-igmpproxy-upstream-intf-learning-00.txt, März 2007.

[183] Villamizar, Curtis: *OSPF Optimized Multipath (OSPF-OMP)*. Internet Draft, Work in Progress – draft-ietf-ospf-omp-02.txt, Februar 1999.

[184] Villamizar, Curtis und Tony Li: *IS-IS Optimized Multipath (ISIS-OMP)*. Internet Draft, Work in Progress – draft-villamizar-isis-omp-00.txt, Oktober 1998.

[185] Vogt, Christian: *Efficient and Secure End-to-End Mobility Support in IPv6*. Doktorarbeit, Universität Fridericiana zu Karlsruhe (TH), Dezember 2007.

[186] Vohra, Q. und E. Chen: *BGP Support for Four-octet AS Number Space*. RFC 4893 (Proposed Standard), Mai 2007.

[187] Waitzman, D., C. Partridge und S.E. Deering: *Distance Vector Multicast Routing Protocol*. RFC 1075 (Experimental), November 1988.

[188] Wang, Bin und Jennifer C. Hou: *Multicast Routing and Its QoS Extension*. IEEE Network, 14(1):22–36, Januar 2000. DOI 10.1109/65.819168.

[189] Wittmann, Ralph und Martina Zitterbart: *Multicast: Protokolle und Anwendungen*. dpunkt, Heidelberg, 1999.

[190] Wright, Gary W. und Richard W. Stevens: *TCP/IP Illustrated, Volume 2: The Implementation*, Band 2 der Reihe *TCP/IP Illustrated*. Addison Wesley, 1995.

[191] Wroclawski, J.: *Specification of the Controlled-Load Network Element Service*. RFC 2211 (Proposed Standard), September 1997.

[192] *eXtensible Open Router Platform*. http://www.xorp.org/.

[193] *XORP Documentation*. http://www.xorp.org/design_docs.html.

[194] Yang, Baijian und Prasant Mohapatra: *Multicasting in Differentiated Service Domains*. In: *Global Telecommunications Conference, 2002. GLOBECOM '02.*, Band 3, Seiten 2074–2078. IEEE, November 2002. DOI 10.1109/GLOCOM.2002.1188996.

[195] Ylonen, T. und C. Lonvick: *The Secure Shell (SSH) Protocol Architecture*. RFC 4251 (Proposed Standard), Januar 2006.

Abkürzungsverzeichnis

A

A	Address (DNS Resource Record)
AAA	Authentication, Authorization, Accounting
AAAA	„Quadruple-A“ (DNS Resource Record)
AC	Adminssion Control
ACK	Acknowledgement
AF	Assured Forwarding (PHB Group)
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
ASCII	American Standard Code for Information Interchange
ASM	Any Source Multicast
ASN.1	Abstract Syntax Notation One

B

BGMP	Border Gatway Multicast Protocol
BGP	Border Gateway Protocol
BIDIR-PIM	Bi-directional PIM
BR	Border Router
BSR	Bootstrap Router
BSS	Basic Service Set

C

CAM	Content Addressable Memory
CBQ	Class Based Queueing
CBT	Core Based Trees
CGA	Cryptographically Generated Address
CLI	Command Line Interface
CS	Class Selector (PHB Group)

D

DAM	Diffserv-Aware Multicasting
DCCP	Datagram Congestion Control Protocol
DF	Defignated Forwarder
DF	Designated Forwarder
Diffserv	Diffentiated Services
DNS	Domain Name System
DR	Designated Router
DRbit	Designated Router Bit
DRM	Domain Resource Manager
DS	Differentiated Services, Diffserv
DSCP	Diffserv Codepoint
DSMC	Diffserv Multicast
DVMRP	Distance Vector Multicast Routing Protocol
DWDM	Dense Wavelength Devision Multiplex

E

EAC	Endystem Adminssion Control
EBGP	Exterior Border Gateway Protocol
ECMP	Equal Cost Multipath
ECN	Explicit Congestion Notification
EF	Expedited Forwarding (PHB)
EGP	Exterior Gateway Protocol
EMBAC	End-to-End Measurement Based Adminssion Control

F

FEC	Forward Error Correction
FEA	Forwarding Engine Abstraction
FHR	First Hop Router
FIB	Forwarding Information Base

G

G	Group (Address)
GIST	General Internet Signaling Transport
GLOP	*Eigenname*
GMP	Group Management Protocol
GNU	GNU is not Unix
GPL	GNU General Public Licence
GRE	Generic Routing Encapsulation Protocol
GRIP	Gauge and Gate Reservation with Independent Probing

H

HDE	Heterogenous DSCP Header Encapsulation

HDVMRP	Hierarchical DVMRP
HPIM	Hierarchical Protocol Independent Multicast
HRW	Highest Randon Weight
HTTP	Hypertext Transfer Protocol

I J K

IBGP	Interior Border Gateway Protocol
IBSS	Independent Basic Service Set
ICMP	Internet Control Message Protocol
ID	Identification
IEEE	„I-triple-E“, formerly Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
Iif	Incoming Interface
Intserv	Integrated Services
IP	Internet Protocol
IPC	Inter-process Communication
IPsec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization

L

LAN	Local Area Network
LE	Lower Effort (PDB), formerly Limited Effort (PHB)
LSA	Link State Advertisement

M

M-ISIS	Multi Topology Intermediate System to Intermediate System
MADCAP	Multicast Address Dynamic Client Allocation Protocol
MASC	Multicast Address-Set Claim
MBAC	Measurement Based Adminssion Control
MBone	Multicast Backbone
MF	Multifield (Classifier)
MFC	Multicast Forwarding Cache
MFIB	Multicast Forwarding Information Base
MFEA	Multicast Forwarding Engine Abstraction
MIB	Management Information Base
MIGP	Multicast Interior Gateway Protocol
MLD	Multicast Listener Discovery
MOSPF	Multicast Open Shortest Path First
MPEG	Motion Picture Experts Group
MPLS	Multiprotocol Label Switching

MRIB	Multicast Routing Information Base
MSC	Multicast for Small Conferences
MSDP	Multicast Source Discovery Protocol
MT-OSPFv3	Multi Topology Open Shortest Path First Version 3
NACK	Negative Acknowledgement
NAPTR	Naming Authority Pointer (DNS Resource Record)
NARADA	*Eigenname*

N

NICE	NICE is the Internet Cooperative Environment
NCS	Network Control Server
NLRI	Network Layer Reachability Information
NRS	Neglegted Reservation Subtree
NRSstate	Neglegted Reservation Subtree State
NSIS	Next Steps In Signaling
NSLP	NSIS Transport Layer Protocol

O

OCBT	Ordered Core Based Trees
Oif	Outgoing Interface
OMNeT	Objective Modular Network Testbed in C++
OMNI	Overlay Multicast Network Infrastructure
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPF-TE	OSPF Traffic Engineering (Extensions)

P

PDB	Per-Domain Behavior
PDU	Protocol Data Unit
PHB	Per-Hop Behavior
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PIM-SSM	PIM Source Specific Multicast (feature reduced PIM-SM)
PSTN	Public Switched Telephone Network

Q

Q	Quality (high priority DSCP)
QF	Quick Forwarding (PDB)
QoS	Quality of Service
QoS-NSLP	NSLP for Quality of Service Signaling
QSPEC	Quality of Service Specification
QUASIMODO	QoS-aware Multicasting over Diffserv and Overlay Networks

R

R	Receiver (Address or Host)
RED	Randon Early Detect
Reg	Register Tunnel (Packet)
RFC	Request for Comment
RIB	Routing Information Base
RIID	Rendezvous Point Interface Identifier
RIM	Receiver Initiated Marking
RIP	Routing Information Protocol
RIPng	Routing Information Protocol – Next Generation
RP	Rendezvous Point
RPA	Rendezvous Point Address
RPF	Reverse Path Forwarding, Reverse Path Flooding
RPL	Rendezvous Point Link
RPM	Reverse Path Multicasting
RPT	Rendezvous Point Tree
Rt	Router
RTCP	Real-time Transport Control Protocol
RTO	Retransmission Timeout
RTP	Real-time Transport Protocol
RTT	Round Trip Time

S

S	Source (Address or Host)
SA	Source Active
Sbit	Switchover Bit
SCTP	Stream Control Transmission Protocol
SEND	Secure Neighbor Discovery
SIG	Signaling (high priority DSCP)
SLA	Service Level Agreement
SLA	Service Level Specification
SMI	Structure of Management Information
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SPT	Shortest Path Tree
SPTbit	Shortest Path Tree Bit
SRV	Service (DNS Resource Record)
SSH	Secure Shell
SSM	Source Specific Multicast

T

TCAM	Ternary Content Adressable Memory
TCP	Transmission Control Protocol
TCS	Traffic Conditioning Specification
TOMA	Two-tier Overlay Multicast Architecture

TTL	Time To Live

U

UC	Un-configured
UDP	User Datagram Protocol
UDP-Lite	Lightweight User Datagram Protocol
URL	Uniform Resource Locator

V

VLAN	Virtual LAN
VW	Virtual Wire (PDB)

W

WLAN	Wireless LAN
WFQ	Weighted Fair Queueing
WTC	Weighted Traffic Conditioning

X Y Z

XORP	Extensible Open Router Platform
XRL	XORP Resource Locator

Index

A

Abstract Syntax Notation One 19
ACKbit 94
Activebit(S,G,Oif) **38**, 82, 90, 93
Address Resolution Protocol 7
Address Resource Record 9
Adresse 6
  kodierte 40
Adresspräfix 9
  NLRI 10
Änderungsbericht 30
  Allow 32
  Block 32
  To_Excl 32
  To_Incl 32
Agent 19
Aggregation
  Diffserv 57
  DSMC-Nachrichten 109
  Joins und Prunes 41
  Reservierungen 56, 63
aktive und programmierbare Netze 21, 78
AND 94
Anfrage 29
  generelle 31
  gruppenspezifische 31
  gruppen- und quellenspezifische .. 31
Anwendungsschicht 3
Any Source Multicast 20, 21
Anycast-RP **45**, 131
Anycastadresse 20
API-Fähigkeiten 158, 190
  optionale 160
Application Layer Multicast 26
  NARADA 26
  NICE 26
ARP 7
ARPANET 14
AS-Grenzrouter 17
AS-Konföderationen 11
AS-Nummer 10
ASN.1 19
Assert 22, **42**
  -Cancel 42
  -Gewinner/-Verlierer 42
  -Metrik 42
  -Nachricht 42
Assoziativspeicher 37
Assured Forwarding PHB Group 61
asymmetrisches Protokoll
  IGMP/MLD 29
  SNMP 19
Ausgangsschnittstelle 38
Autonomes System 10

B

Bekanntheit (Gruppen) 20
Bellman-Ford-Algorithmus 14
Bericht 29
  Änderungs- 30
  Istzustands- 30
Best Effort 60
Best-Effort-Servicemodell 55
BGP 10
BGP4+ 13
Bi-directional PIM **49**, 114
Blackholing 53
Bootstrap-Nachricht 51
Bootstrap-Router-Mechanismus 50
Bootstraprouter 50

Border Gateway Protocol 10
Bridge 4
BRITE 172
Broadcast 20
Broadcast-and-Prune 23
Broadcastverfahren 22
    RPB 22
    RPF 22

C

Candidate-RP-Advertisement-Nachricht 40, 50
CBT **54**, 119
Class Based Queueing 62
Class Selector Codepoint **60**, 92, 189
Class Selector PHB Group 60
CLI 19
Codepoint-Regeln 89
Command Line Interface 19
Communities 19
Concast 20
Conf 81, **93**, 99
    Notation 94, 95
Conf 93
ConfReply **93**
    Notation 94, 95
Content Addressable Memory 37
    Single-Cycle Multiple Logical 39
Controlled Load 62
Count to Infinity 14, 52
Cryptographically Generated Address . 8

D

DAM 69
Datagram Congestion Control Protocol 8
Datagrammdienst 5
Datenübertragung
    datenstrom-/datagrammorientiert . 4
    unzuverlässig/zuverlässig 4
    verbindungslos/-orientiert 4
Datenstrom 9
DCCP 8
Defaultroute 9
Delay 57
Designated Forwarder
    BIDIR-PIM 50
    RPF-Prinzip 22
Designated Router
    OSPF 16
    PIM-SM 43
Designated-Router-Bit .. 81, **93**, 96, 190
DF-Election-Nachricht 50
Dienstgüte 55
    -management, zentrales/verteiltes 62
    -parameter 57
    -signalisierung 56
Dienstvertrag 59
Differentiated Services 57
Diffserv Codepoint 56–58
Diffserv Multicast 75
Diffserv-Prinzip 57
Diffservdomäne 58
Dijkstra-Algorithmus 15, 54, 173
direkt angeschlossener Empfänger **93**, 96
Distance Vector Routing 14
Distanz 15
DNS Resource Record
    Address A/AAAA 9
    Naming Authority NAPTR 9
    Service SRV 9
DNS-Name 9
Domäne
    Diffserv- 58
    Routing- 10
Domänenname 9
Domänenressourcenmanager 63
Domain Name System 9
Domain Ressource Manager 63
Downstream Join State Machine 72
Downstreamschnittstelle 21
DRbit(S,G,Oif) 81, **93**, 96, 190
DRbt 93
DRM 63
DSCP 57, 58
    Class Selector **60**, 92, 189
    einheitlicher ausgezeichneter . 84, **89**, 91
    hochpriorer 83
    niederpriorer 83, **89**
    Pool **58**, 189
DSCP 89
DSCP_KEEP 189
DSCP_LE 84, **89**, 91, 189
DSCP_SIG **89**, 92
dscp_signal 189

DSCP_SIGNAL 189
DSCP_UC 84, **89**, 91, 189
DSMC 75
  Basisverfahren 82, **89**
  DRM-seitiges Timing 104
    sofortiger SPT Switchover 131
  Erweiterung für PIM-SM ... 85, **125**
  inkrementeller Einsatz 150
  Interdomain 151
  routerseitiges Timing 106
    einfache Implementierung 109
DSMC Conf Header 124
DSMC Conf Option 125
DSMC-Grenzschnittstelle 87, **93**, 97, 150, 151
DSMC-Komponente 80, 87, 163, 167
DSMC-Nachrichten
  Aggregation von 109
  Auflistung aller 88
    Basisverfahren 94
    Erweiterung für PIM-SM 127
  Conf 81, **93**, 99
  ConfReply **93**
  Kodierung von 167
  Notation Conf & ConfReply .. 94, 95
  Trigger 81, **93**, 97
    je Routingprotokoll 121
    PIM-SM 126
    vom DR beim SPT Switchover 136
    von jedem Router 123
  TriggerRpt 86, **126**
  TriggerStop **93**, 98
    von jedem Router 122
  TriggerSwitched 86, **126**, 129
    mittels SNMP 137
  TriggerSwitchedAck ... 86, **126**, 129
  TriggerUC 86, **93**, 98
  Verlust von **100**, 148
  Vertauschung von **102**, 143
DSMC-Regeln
  Codepointregeln 89
    DSCP 89
    LE 89
  Informationsregeln 94, 127
    AND 94
    TSG 127
    TSGO 94
  Markierungsregeln 89
    Init 89
    Invd 89
    Keep 90
    Low 89
    Rmrk 90
    UC 89
  Nachrichtenregeln **93**, 122, 123
    Conf 93
    Rply 93
    Tr 93
    Tr′ 123
    TrSt 93
    TrSt′ 122
    TrUC 93
  PIM-SM-Regeln 126
    TrR 126
    TrSM 126
    TrSw 126
    Tsub 126
  Zustandsregeln 90, 93, 126
    DRbt 93
    NRS 90
    Sbit 126
DSMC-Signalisierung 80
Duplikate 22
Durchsatz 56
DVMRP **51**, 118
DVMRP-Nachrichten
  Graft 52
  Graft-Ack 52
  Neighbor-Probe 53
  Prune 52
  Report 52
Dynamic Host Configuration Protocol
  DHCPv4 26
  DHCPv6 ..... 8, 27
Dynamik (Gruppen) 20

E

ECMP **18**, 117
ECN-Bits **58**, 193
Egressrouter 60
  Tunnel 58
Eingangsschnittstelle 38
einheitlicher ausgezeichneter DSCP .. 84, **89**, 91
EMBAC 65
Embedded-RP-Adresse 29, 47

encap_send **157**, 194
Ende zu Ende
  Dienstgüte 56
  Multicastrouting 36
Ende-zu-Ende-Prinzip 5
Endsystem 4
Endsystem Admission Control 65
Equal Cost Multipath Routing .. **18**, 117
  Relaxed 18
Erweiterungskopf, IPv6- 6
Ethernet 5
Expedited Forwarding PHB 60
Explicit Congestion Notification 58
Explicit Multicast 25
Exponential Backoff 105, 142
Exterior Gateway Protocol 10
External BGP 11

F

Fast Leave 31
FEA 162
Filter
  IGMP/MLD 30
  verteilter MFC 38
Filtermodus
  Ausschließen 30
  Einbinden 30
First-Hop-Router 57
Flags
  IPv4-Kopf 7
  IPv6-Adresse 28
  kodierte Adresse 40
  Type of Service 56
Flash Updates 52
Flood-and-Prune 23
Flow 9
Flow Label 9
Fluten 22
  zuverlässiges 16
Forwarder 22
Forwarding 4
Forwarding Engine Abstraction 162
Forwarding Information Base 12
Frame Relay 5

G

Gültigkeitsbereich 27
Gültigkeitsdauer 40
Gebietsgrenzrouter 16
General Internet Signaling Transport 56
Generation ID
  DVMRP 53
  PIM 40
getsockopt 188
Gewicht 15
GIST 56
GLOP-Adressierung 26
Graceful Shutdown 52
Graft-Ack-Nachricht 48
  DVMRP 52
  PIM 40
Graft-Nachricht 48
  DVMRP 52
  PIM 39
Grafting 23
Grenzrouter 64
  AS- (OSPF) 17
  Autonomes System 12, 55
  Diffserv 57, 60
  Gebiets- (OSPF) 16
  HDVMRP 51
  MOSPF 53
  Multicastbereich 27
GRIP 71
Group-Membership-LSA 53
Gruppe
  anonym/bekannt 20
  dynamisch/statisch 20
  heterogen/homogen 20
  offen/geschlossen 20
  transient/permanent 20
Gruppenadresse 21, **26**
Gruppenadresspräfix 39, 50
Gruppenkennung 26
Gruppenkommunikation 20
Gruppenmanagement 29
Gruppenmanagementprotokoll
  IGMP 29
  MLD 29

H

Hashmaskenlänge 51
Hashtabelle 37
Hashwert 8, 18, 37
Hauptgebiet 17

HDE 70
HDVMRP 51
Hello-Mechanismus 40
  Gültigkeitsdauer 40
  Generation ID 40
Hello-Nachricht
  OSPF 16
  PIM 40
Hello-Optionen 40
  LAN Prune Delay 109
Hello-Protokoll 16
heterogene Gruppen 77
Heterogenität (Gruppen) 20
Heterogenous DSCP Header Encapsulation 69
Hierarchie
  HDVMRP 51
  IS-IS 17
  Multicastrouting 36
  OSPF 17
  ungenutzte 17
Highest Random Weight 18
HIP 36
hochpriorer DSCP 83
Hop Count 10
Hop-by-Hop Options
  DSMC Conf Option 125
  Router Alert Option 124
Host 4
Host Groups 21
Host Suppression 31
Hostroute 9
Hot Potato Forwarding 22
HPIM 36

I

IANA 6, 13, 21, 27
Identifikator 8
IETF 10
if_output 157
IGMP/MLD-Proxy 33
IGMP/MLD-Proxying 118
IGMPMSG_
  BW_UPCALL 157, 198
  DSCP_NOCONF **157**, 159, 191, 198
  NOCACHE 193
  PERIODIC_UPCALL **157**, 160, 191, 194
  WHOLEPKT 157, 194
  WRONGVIF 156, 193
Iif(S,G) 38
Informationsregeln 94, 127
InformRequest-PDU 19
Ingressrouter 60
  Tunnel 58
Init 89
Integrated Services 56
Inter-area Multicast Forwarder 53
Interdomänenmulticastrouting 36
Interdomänenrouting 10
Interior Gateway Protocol 10
Interleaving 71
Intermediate System to Intermediate System 17
Internal BGP 11
Internet Control Message Protocol 7
Internet Engineering Task Force 10
Internet Group Management Protocol 29
Internetprotokoll 5
Interoperabilitätsregeln 55
Intradomänenmulticastrouting 36
Intradomänenrouting 10
Invd 89
inverser Multicast 20
IP-Adresse 6, 7
  Anycast 20
  Multicast 26
  Unicast 7
  Schreibweise 8
IP-Kopf 6
IP-Paket 6
ip_input 155
ip_mdq 156, 192
ip_mforward 156
ip_mloopback 158
ip_mroute.c 192
ip_mroute.h 188
ip_output 157
IPv6-Erweiterungsköpfe 6
  DSMC Conf Header 124
IS-IS 17
ISO/OSI-Referenzmodell 3
Istzustandsbericht 30
  Excl 32
  Incl 33

J

J/P_Override_Interval **41**, 108
Jitter 57
Join Suppression 41
Join-Nachricht 24
Join/Prune-Nachricht 40

K

Kanal 27
Keep 90
Keepalive-Nachricht 47
KeepaliveTimer(S,G) 44
Kindrouter 22
Kleine-Welt-Netz 172
Knoten 3
komprimiertes Video 71
Kontingent 64
Kosten 15

L

LAN 108
Layered Multicast 78
LE 89
Leaky Bucket 58
Lebensdauer (Gruppen) 20
Lightweight User Datagram Protocol ..8
Limited Effort PHB 62
Link 4
Link State Advertisement 16
  AS-external- 17
  Group-Membership- 53
  Intra-Area-Prefix- 17
  Network- 16
  Opaque 17
  Summary- 17
Link State Routing 15
Lokator 9
Longest Prefix Match 9
Loopbackschnittstelle **4**, 12, 183
Loss 57
Low 89
Lower Effort PDB 60, **62**

M

M-ISIS 17
MADCAP 26
Management Information Base 19
Manager 19
Markieren 58
Markierungsregeln 89
MASC 26
MaxDrmProcessingTime 106
MaxDrmRtt 104
Maximum Residual Bandwidth 65
MaxSignalingDelay 106
MBone 51
mbuf 192
Medium 4
Mehrfeldklassifizierer 58
messungsbasierte Zugangskontrolle **65**, 71
Metrik 9
mf6cctl 159
mfc 191
MFC **36**, 164
MFC-Eintrag **37**, 191
  quellenunspezifischer **39**, 94, 96, 115, 146
  temporärer 156
mfcctl 189
mfcctl2 189
  mfc_flags 159
MFCentry(S,G) 37
MFEA 162, **163**
MfeaMrouter 164
MfeaNode 164
MLD/IGMP-Komponente 81
MOSPF **53**, 119
MPEG-2-komprimiertes Video 71
MPLS 76
Mrouter-Socket 156, 187
  API-Fähigkeiten 158, 190
    optionale 160
mrt6_dscp 159, 160
MRT6_DSCP 159
mrt6_dscp_noconf 159
MRT6_DSCP_NOCONF 159
MRT6_SET_DSCP 160
MRT6MSG_DSCP_NOCONF ..159, 160
MRT6MSG_PERIODIC_UPCALL ..159
MRT_API_CONFIG 158, 190
MRT_API_SUPPORT 190
mrt_dscp 158, 188
  .dscp_lowprio 159
  .dscp_unavail 159

MRT_DSCP 158, 188
mrt_dscp_noconf 158, 189
.msg_suppress 160
.msg_time 159
MRT_DSCP_NOCONF 158, 189
MRT_MFC_FLAGS_
DISABLE_DSCP_NOCONF ... **159**, 190, 194
KEEP_DSCP **159**, 190, 194
SLOW_DSCP_NOCONF .. **159**, 160, 190, 197
MRT_MFC_PERIODIC_UPCALL . **160**, 161, 190, 193
MSC 25
MSDP **47**, 131
MSDP-Nachrichten
Keepalive 47
Source Active 47
MT-OSPFv3 17
Multicast 20
Multicast Address-Set Claim 26
Multicast Forwarding Cache **36**, 164
Activebit(S,G,Oif) **38**
DRbit(S,G,Oif) 81
Iif(S,G) 38
kombinierte Ein-/Ausgangsfilter . 38
MFCentry(*,G) 39
MFCentry(S,G) 37
Modifikationen für DSMC 88
NRSstate(S,G,Oif) 81
Oif 38
Sbit(S,G) 81
Unix-Modell 38
verteilter 38
Multicast Forwarding Information Base 37
Multicast Listener Discovery 29
Multicast Routing Information Base . 13
Multicast Routing Table 36
Multicast Source Discovery Protocol **47**, 131
Multicast-Socket-API 30
Multicastadresse 26
All IGMPv3-capable Multicast Routers 30
All MLDv2-capable Multicast Routers 30
All PIM Routers 39
All Systems on this Subnet 31
All-DVMRP-Routers 53
Gültigkeitsbereich 27
IPv4 27
IPv6 28
Link-Local Scope All Nodes 31
Reichweite 27
Solicited Node 7
transiente/permanente 28
unicastpräfixbasierte 26–28
Multicastkanal 27
Multicastrouting 20, 35
Multicastrouting-API 164, 187
Erweiterung für DSMC 158, 188
Multicastroutingkomponente 82
Multicastroutingprinzip
Broadcast-and-Prune 23
Rendezvousstellen-basiert 24
Multicastverfahren
alternative 25
RPM 22
TRPB 22
multihomed
Autonomes System 12
Host 4
Ressourcenmanager 110
Multipeer 20
Multiprotocol Label Switching 76
Multitopologie-Erweiterung 17

N

Nachbarerkennung 7
Nachrichtenregeln **93**, 122, 123
Nachrichtenverlust 100
Nachrichtenvertauschung 102
Naming Authority Pointer 9
NARADA 26
Neglegted Reservation Subtree Problem 67
Neighbor-Probe-Nachricht 53
Network Control Server 64
Network Layer Reachability Information 10
Netzwerkmanagement 19
Netzwerkpräfix 6, 8
Netzwerkschnittstelle 4
Netzwerksocket 8
neuer Multicastpfad 84

Next Hop 9
Next Steps in Signaling 56
NICE 26
niederpriorer DSCP 83, **89**
NLRI 10
Nonbroadcast Multiple Access Network 5
NRS 90
NRS-Problem **67**, 79
  Lösung 90
NRS-Zustand ..80, 81, **90**, 160, 183, 190
  ungültiger 82, **90**
  vereinfachter **91**, 159, 190
  Zustandsübergangsdiagramm 91, **95**
NRSstate(S,G,Oif) 81, **90**
Null-Register 43

O

OCBT 36
Offenheit (Gruppen) 20
Oif 38
OMNeT++ 172
OMNI 26
Open Shortest Path First 15
Ordnung 71
OSPF 15
  OSPF-TE 17
  OSPFv2 15
  OSPFv3 17
Overlay Multicast 26
  OMNI 26
  TOMA 26

P

Paket 6
Paketverlustwahrscheinlichkeit 57
Paketvermittlung 5
Per-Domain Behavior 59
  Lower Effort 60, **62**
  Quick Forwarding 60
  Virtual Wire 60
Per-Flow-Zustand
  in Signalisierung 56
  in Weiterleitung 57
Per-Hop Behavior 57
  Assured Forwarding Group 61
  Class Selector Group 60
  Default/Best Effort 60
  Expedited Forwarding 60
  Limited Effort 62
Pfadattribut 10
  AS_PATH 10
  LOCAL_PREF 11
  NEXT_HOP 10
Pfadlänge
  Bestimmung 172
  durchschnittliche 177
Pfadvektor 10
PHB 59
phyint_send **157**, 194
PIM Dense Mode **48**, 113
PIM Source Specific Multicast ..**45**, 110
PIM Sparse Mode 42
PIM-Nachricht 39
  Assert 42
  Bootstrap 51
  Candidate-RP-Advertisement 40, 50
  DF-Election 50
  Graft 39, 48
  Graft-Ack 40, 48
  Hello 40
  Join/Prune 40
  Register 40, 43
  Register-Stop 40, 43
  State-Refresh 49
PIM-SM-Komponente 88
PIM-SM-Regeln 126
pim_register_send **157**, 194
pim_register_send_rp 157
pim_register_send_upcall 157
PimInterfaceEntry 97
PimMfc 168
PimMrt 164
PimNode 164
Poisoned Reverse **14**, 22, 48, 52
Policing 58
Policy
  -funktion PIM-SM 43
  BGP 11
Port(nummer) 8
Prüfsumme 7
Priorität
  -scheduler 61, 62
  -stufen 56, 60
  BSR-Mechanismus 50
  höchste 61

Verwurfs- 61, 65
Probingpaket 71, 73
Profil 58
Protocol Independent Multicast 39
BIDIR-PIM **49**, 114
BSR-Mechanismus 50
PIM-DM **48**, 113
PIM-SM **42**
PIM-SSM **45**, 110
Protokollnummer 6
Proxy
IGMP/MLD- 33
Overlay Multicast 26
Prune(S,G,rpt) 45
Prune-Echo 42
Prune-Nachricht 25
DVMRP 52
Pruning 23
Punkt-zu-Punkt-Link 15
Punkt-zu-Punkt-Verbindung 4

Q

QoS-NSLP 56
QoS-Routing 76
QSPEC 56
Quality of Service 55
QUASIMODO 72
Querier 29, 30
Quick Forwarding PDB 60

R

$1/R$ 106
Random Early Detect 61
Real-time Control Protocol 21
Receiver Initiated Marking 69
Referenzmodell, ISO/OSI- 3
Register-Nachricht 40, 43
Register-Stop-Nachricht 40, 43
Register-Tunnel **43**, 157
Reichweite 27
Relaxed ECMP **18**, 76
Rendezvous Point Interface Identifier 29
Rendezvousstelle 24
Rendezvousstellen-basiertes
Multicastrouting 24
Rendezvousstellenadresse 49
Rendezvousstellenbaum 25, 42
Rendezvousstellenkandidat 50
Rendezvousstellenlink 49
Repeater 4
Report-Nachricht 52
Ressourcen 55
Ressourcengraph 139
Ressourcenmanager 63
Ressourcenpartitionierung 62
Reverse Path Broadcasting 22
Reverse Path Flooding 22
Reverse Path Forwarding 21
Reverse Path Multicasting 22
RIM 69
RIP 14
Rmrk 90
Robustheit
Berichte (Gruppenmanagement) . 30
Join/Prune-Nachrichten 41
Robustheitsvariable 31
Route 9
BGP 10
Route Hold-down 52
Routen-Reflexion 11
Router 4
Router Alert Option 124
Routing 4
Routing Information Base 12
Routing Information Protocol 14
Routingdomäne 10
Routingebene 37
Routingmetrik
BGP 10
OSPF 15
Routingschleife
External BGP 10
Internal BGP 11
Relaxed ECMP 18
RIP 14
Routingtabelle 12
RPF-Prinzip 21
Rply 93
RSVP 56
RTCP 21

S

Sanduhrmodell 5
Sbit 126
Sbit(S,G) 81, **126**, 127, 190

Schichten 3
höhere anwendungsorientierte 3
transportorientierte 3
Schichtenmodell, TCP/IP- 3
Schnittstelle 4
Schnittstellenidentifikation 6, 8
SCTP 8
Secure Neighbor Discovery 8
Secure Shell 19
Service Level Agreement 59
Service Level Specification 59
Service Provisioning Policy 59
Service Resource Record 9
setsockopt 188
Shortest Path Tree 43
Sicherungsschicht 3
Simple Network Management Protocol 19
SimpleDSMC **123**, 172
SimpleDSMCext 123
SimpleDSMCwithStop 123
Single Point of Failure 62
Sitzung 11
skalenfreies Netz 172
Skalierbarkeit
BGP 11
bidirektionale Bäume 54
Diffserv 57
DVMRP 51
EAC 65
eingebettete RP-Adressen 29, 47
Intserv 56
Multicast 25
NSIS 56
Wegewahl 9
SNMPv2-Trap-PDU 19
Snooping Switch 7, 30
Socket 9
Socketoption 158, **188**
Soft-State-Prinzip 14
Solicited Node Multicast Address 7
Source Routed Multicast 25
Source Specific Multicast 20, 27
Source-Active-Nachricht 47
Spannbaum 25
Split Horizon with Poisoned Reverse 14
SPT Switchover 43
sofortiger 131
SPTbit(S,G) 44
SSH 19
State-Refresh-Nachricht 49
Stateless Address Autoconfiguration 8
Stream Control Transmission Protocol 8
stromabwärts 22
stromaufwärts 21
Structure of Management Information 19
Stub-AS 12
Stummellink 16
Subnet-Router-Anycastadresse 29
Subnetzwerkpräfix 6
Switch 4
Switchover-Bit 81, 85, **126**, 127, 190
SwitchToSptDesired(S,G) **43**, 125, 131

T

T_{check} 105
$T_{HoldDown}$ 105
T_{RTO} 105
T_S 106
Broadcast-and-Prune 108
PIM-DM 108
sofortiger SPT Switchover 132
TCP 8
TCP/IP-Schichtenmodell 3
Telnet 19
Time to Live 27
Token Bucket 58
TOMA 26
Tr 93
Tr′ 123
Traffic Class 7
Traffic Conditioning 58
Traffic Conditioning Agreement 59
Traffic Conditioning Specification 59
Traffic Engineering 76
Transitlink 15
Transmission Control Protocol 8
Transportschicht 3
Trigger 81, **93**, 97
je Routingprotokoll 121
PIM-SM 126
vom DR beim SPT Switchover 136
von jedem Router 123
Triggered Update 14
TriggerRpt 86, **126**
TriggerStop **93**, 98
von jedem Router 122

TriggerSwitched 86, **126**, 129
mittels SNMP 137
TriggerSwitchedAck 86, **126**, 129
TriggerUC 86, **93**, 98
TrR 126
TrSM 126
TrSt 93
TrSt′ 122
TrSw 126
TrUC 93
Truncated Reverse Path Broadcasting 22
TSG 127
TSGO 94
Tsub 126
TTL-Scoping **27**, 194
tunneln
BGMP 55
CBT 54
Diffserv 58
DVMRP 52
MSDP 48
PIM-SM 43
Type of Service 7, 55

U

UC 89
UDP 8
UDP-Lite 8
Ungleichbehandlung 55
Unicast 20
Unicastadresse 7
Unicastrouting 20
Unicastroutingprotokoll 9
Unix-Modell 38
unkontrollierte Ressourcennutzung ... 80
unspezifizierte Adresse **6**, 94, 140
Unterdrückungszeit **106**, 189
Register-Nachrichten 43
Update_SPTbit(S,G,iif) **44**, 128, 171
Upstreamschnittstelle 21
User Datagram Protocol 8

V

Verhaltensaggregat 57
verkehrsgesteuert 38
Verkehrskonditionierung 58
Verkehrskontrolle 58
Vermittlungsschicht 3
Verteilbaum 22
bidirektionaler 25, 33, 55
geteilter 25
quellenspezifischer 25
quellenspezifischer kürzester 43
Verzögerung
Dienstgüteparameter 57
Flag 56
Verzögerungsschwankung 57
Verzweigungsrouter 45, 85
Virtual Wire PDB 60
VLAN-Tagging 4
Vorwärtsfehlerkorrektur 71

W

Waxman-Topologie 172
Wegewahl 4
Weighted Fair Queueing 61
Weighted Traffic Conditioning 69
Weiterleitung 4
differenzierte 57
Weiterleitungsebene 37, 56, 80
Weiterleitungsschleife
IGMP/MLD-Proxying 34
RPF 22
Wildcard Receiver 53
Wireless LAN 5
WTC 69

X

Xcast 25
XORP 161
XORP Resource Locator 162
mfea/0.2/add_mfc4 ..... **166**, 168
mfea/0.2/add_mfc6 ..... **166**, 168
XRL 162

Z

Zugangskontrolle **58**, 77
messungsbasierte **65**, 71
zustandslose Autokonfiguration ... **8**, 27
Zustandsregeln 90, 93, 126
Zuverlässigkeit 56

ISBN: 978-3-89959-791-2 +++EUR[D] 29.90