UNIVERSITÄT
KOBLENZ · LANDAU
Fachbereich 4: Informatik

Analyse des RIPng-Protokolls für IPv6

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science
vorgelegt von

Benjamin Hück

Erstgutachter:	Prof. Dr. Christoph Steigner Institut für Informatik
Zweitgutachter:	Dipl. Inform. Frank Bohdanowicz Institut für Informatik

Koblenz, im November 2011

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

.....

(Ort, Datum) (Unterschrift)

Abstrakt

Durch das fehlerhafte Vergabemanagement und das ständige Wachstum an internetfähigen Geräten sind die IPv4-Adressen ausgeschöpft. Aus diesem Grund und den neuen Anforderungen, die an die Technik des Internets gestellt werden, ist das IPv6-Protokoll entwickelt worden. Dieses bietet einen vielfach größeren Adressraum und wird das IPv4-Protokoll nach und nach ersetzen. Jedoch müssen für das neue Internetprotokoll die Routing-Protokolle angepasst werden. In dieser Ausarbeitung wird das dynamische Routing-Protokoll RIPng aus der Familie der Inter-Gateway-Protokolle (IGP) analysiert. Mithilfe dieses Protokolls tauschen die Router innerhalb eines Netzwerkes untereinander Informationen über ihre Verbindungen aus. Des Weiteren werden die Grundlagen des IPv6-Protokolls, der verwendeten Protokoll-Algorithmen und des RIPv2-Protokolls erläutert. Im praktischen Teil der Ausarbeitung werden Eigenschaften von RIPng sowie das Counting-to-Infinity-Problem genauer betrachtet.

Abstract

Due to the incorrect allocation management and the continuous increase of internet-capable devices, IPv4-adresses are nearly exhausted. For this reason and because of new requirements demanded by the technique of the internet the IPv6-protocol has been developed. The IPv6-protocol provides a larger adress space and will gradually replace the IPv4-protocol. However, the routing-protocols have to be adapted to the new internet-protocol. This paper will introduce and analyse the dynamic Routing-Information-Protocol RIPng which in itself is an inter-gateway protocol (IGP). The routers inside the network can exchange information about their different connections over this protocol. Furthermore, this paper will explain the basics of the IPv6-protocol, of the used protocol-algorithm as well as of the RIPv2-protocol. In the practical part of the paper the Counting-to-Infinity-Problem and some attributes of RIPng will be reviewed under the IPv6-protocol.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

Das zu Anfang der 80er Jahre entwickelte Internetprotokoll-Version 4 (IPv4) bildet die Grundlage des heutigen Internets. Bei der Entwicklung des IPv4-Protokolls wurde jedoch nicht an eine weltweite Verbreitung dieses Protokolls gedacht. Zu diesem Zeitpunkt war diese enorme Entwicklung und Verbreitung des IPv4-Protokolls nicht absehbar. Deswegen wurden die möglichen IPv4-Adressräume ohne besondere Beachtung vergeben. Dies führt heutzutage zu einem Engpass, da internetfähige Geräte in immenser Stückzahl produziert und verlangt werden. Jedoch benötigt jedes dieser Geräte zur Kommunikation mit anderen Geräten eine eindeutige IP-Adresse.

Aus diesem und weiteren Gründen wurde der Nachfolger des IPv4-Protokolls, das IPv6-Protokoll, entwickelt. Dieses bietet einen größeren IP-Adressbereich und kann das aktuelle IPv4-Adressproblem beheben. Die Anzahl der zur Verfügung stehenden IPv6-Adressen ist dabei immens und wird einem weiteren Adressproblem in Zukunft vorbeugen. Außerdem liefert es neue Möglichkeiten zur effizienteren Übertragung von Datenpakete.

Die schrittweise Einführung von IPv6 bedingt allerdings die Entwicklung neuer dynamischer Routing-Protokolle. Diese dienen dem automatischen Austausch von Nachrichten und Datenpaketen innerhalb der autonomen Systeme (AS), um ein optimales Routing vornehmen zu kön-

nen. Aus diesem Grund wurde das dynamische Routing-Protokoll RIPng (Routing-Information-Protocol–Next-Generation), welches zu der RIP-Familie gehört, entwickelt und an die IPv6-Adressen angepasst. Mithilfe dieses Protokolls können die Router, basierend auf dem IPv6-Protokoll, Informationen über ihre Routing-Tabellen untereinander austauschen und abgleichen. Die Router sind somit über die Topologie innerhalb des Netzwerkes informiert und kennen durch den ständigen Austausch mit ihren Nachbarroutern und dem Distanz - Vektor - Algorithmus die kürzesten Routen innerhalb des Netzwerkes.

Diese wissenschaftliche Ausarbeitung untersucht das RIPng-Protokoll anhand einer ausführlichen Analyse inklusive eines Testszenarios. Im theoretischen Teil wird das Protokoll in Hinblick auf dessen Aufbau, das Verfahren zur Verarbeitung der Datenpakete sowie die speziellen Protokolleigenschaften einer genauen Analyse unterzogen und ausführlich dokumentiert. Im praktischen Teil erfolgt die Betrachtung eines auf IPv6- und RIPng-basierenden Testszenarios.

Zu Beginn der Ausarbeitung wird in Kapitel 2 auf die allgemeinen Grundlagen eingegangen. Diese werden für das Verständnis der weiteren Ausarbeitung vorausgesetzt. Dabei wird das IPv6-Protokoll, bekannte Protokollalgorithmen sowie das Vorgängerprotokoll von RIPng, RIP–Version 2 für IPv4, erklärt. Im darauf folgenden Kapitel 3 wird das neue RIP-Protokoll RIPng für IPv6 explizit untersucht und erläutert. Dabei werden die verschiedenen Timer, das neue Nachrichtenformat sowie die Verarbeitung der ein- und ausgehenden Datenpakete ausführlich erklärt. In Kapitel 4 werden verschiedene Szenario auf Grundlage von IPv6 und RIPng aufgebaut und getestet. In den Szenarien wird das Routing-Verhalten sowie das Counting-to-Infinity-Problem unter RIPng untersucht und analysiert. Abschließend erfolgt in Kapitel 5 das Fazit sowie ein kurzer Ausblick, welche Möglichkeiten RIPng in der Zukunft bieten kann.

Kapitel 2

Allgemeine Grundlagen

In diesem Kapitel werden allgemeine Grundlagen für das weitere Verständnis der Ausarbeitung erläutert. Dabei wird das Internetprotokoll-Version 6 (IPv6) und dessen Routing-Header erklärt. Außerdem werden wichtige Informationen zu dem Routing-Information-Protokoll–Version 2 (RIPv2), dem Distanz-Vektor- und Link-State-Algorithmus aufgezeigt.

2.1 IPv6

Der folgende Abschnitt erläutert die Grundlagen zu dem Internetprotokoll-Version 6 (IPv6). Nach einer kurzen Einführung in das Thema wird die neue Adresslänge sowie die Adressarchitektur beschrieben. Des Weiteren werden der IPv6-Header, die Extension-Header unter besonderer Betrachtung des Routing-Header vom Typ 0 sowie das Routing unter IPv6 näher betrachtet. Die Informationen wurden, soweit nicht anders vermerkt, [Hag09] entnommen.

2.1.1 Allgemein

Die erste Verwendung fand das Internetprotokoll mit Version 4 Anfang der achtziger Jahre. Damals wurde das Internetprotokoll-Version 4 entwickelt, um staatliche und universitäre Einrichtungen miteinander zu verbinden und ein Netzwerk aufzubauen. Jedoch fand IPv4 eine immer grö-

ßere Verbreitung auf der Welt und stellt bis heute die Grundlage des Internets dar. Nach fast 30 Jahren ist das IPv4-Protokoll in die Jahre gekommen und Verbesserungen stehen an. Durch die schnelle Wachstumsrate des Internets und das falsche Vergabemanagement der IPv4-Adressen zur Anfangszeit ist der IPv4-Adressbereich ausgeschöpft. Es wurden verschiedene Techniken wie das *Classless Inter-Domain Routing (CIDR)* [Ful93] oder das *Network-Adress-Translation (NAT)*-Verfahren [Ege01] entwickelt, um dem Adressproblem vorbeugen zu können und die festen IP-Klassen aufzuheben und variabler bei der Zuweisung der IP-Adressbereiche zu sein. Jedoch reichen selbst diese Techniken für die große Anzahl an internetfähigen Geräten nicht mehr aus. Des Weiteren können neue Adressblöcke nicht ohne den Verlust der Kompatibilität in das bestehende Protokoll IPv4 eingebaut werden, was die Entwicklung eines neuen Protokolls bedingt [Ger09].

Aus diesen Gründen wurde der Nachfolger IPv6 entwickelt. Die ersten Arbeiten an IPv6 begannen im Jahr 1991. Zur Anfangszeit der Entwicklung wurde IPv6 unter dem Namen IPng (Internet-Protocol–Next Generation) entwickelt. Unter diesem kollektiven Namen entwickelt die *Internet Engineering Task Force (IETF)* neue IP-Nachfolgerprotokolle, in diesem Fall das Nachfolgerprotokoll zu IPv4. Nachdem in den Jahren 1992 und 1993 mehrere Erneuerungen und Vorschläge eingearbeitet wurden, führte dies zur Bezeichnung IPv5 [ITW]. IPv5 bietet zwar gegenüber IPv4 eine verbesserte Echtzeitfähigkeit, wurde jedoch aus Kosten-Nutzen-Erwägungen eingestellt und ein Protokoll mit dem offiziellen Namen IPv5 existiert nicht [Enz11a]. Deshalb wurde im Dezember 1995 IPv6 offiziell als Internet-Standard durch [Dee95] eingeführt. Im Jahre 1998 wurde das obsolete RFC durch [Dee98] ersetzt.

Dabei sind bewährte Eigenschaften beibehalten und die Skalierbarkeit und Flexibilität erweitert worden. Außerdem wurde sichergestellt, dass IPv6 den zukünftigen Anforderungen von Diensten im Internet und dessen Wachstumsrate standhalten kann. Durch den vergrößerten Adressbereich auf 128 Bit steht eine immense Anzahl an möglichen IPv6-Adressen bereit, welche einer weiteren Adressknappheit in ferner Zukunft vorbeu-

gen soll.

Die Umstellung von IPv4- zu IPv6-Adressen wird sich in der weiteren Zukunft langsam, schrittweise ineinander übergehend vollziehen und IPv4 wird noch einige Jahre neben IPv6 existieren.

Viele Hardwarehersteller haben ihre IPv6-Implementationen bis dato weitestgehend getestet und sind bereit für den Umstieg von IPv4 auf IPv6 durch den Betrieb ihrer Produkte und Webseiten im Dual-Stack-Modus. Dabei müssen jedoch nicht nur die Hardwarehersteller ihre Systeme umstellen, sondern auch die Betriebssysteme der Endbenutzer sowie die Infrastrukturen auf Provider- und Backbone-Seiten müssen IPv6-fähig gemacht werden. Diverse Probleme sind vorprogrammiert und es kommt immer wieder zu verschiedenen Schwierigkeiten bei der Verwendung des IPv6-Protokolls. So haben Betriebssysteme von Apple (Mac OS X und iOS) große Probleme sich im Dual-Stack-Betrieb für die richtige IP-Adresse zu entscheiden, sodass der Benutzer des öfteren eine Fehlermeldung statt der Internetseite in seinem Browser angezeigt bekommt. Außerdem schaltet sich die Windows-eigene IPv6-taugliche Firewall bei der Installation einer Personal-Firewall ab. So kann der IPv6-Verkehr den Computer unkontrolliert erreichen [End10].

Ein weiterer Kritikpunkt ist die stark gewachsene Komplexität, da eine Vielzahl an neuen Leistungsmerkmalen die Möglichkeit von Fehlern bei der Implementierung und Konfiguration von IPv6 drastisch erhöhen. Außerdem wurde in der Vergangenheit wenig Wert auf die Abstraktion bei der Behandlung von Netzwerkadressen gelegt. Dies bedingt ein kompliziertes Einarbeiten der IPv6-Unterstützung für die Programmierer in ihre Software [Ger09].

Ein weiteres Problem des IPv6-Protokolls stellt die Abschaffung der NAT-Technik dar. Jeder Rechner kann über eine „echte“, global gültige IPv6-Internetadresse auf der ganzen Welt erreicht werden. Ebenfalls ist die interne Netzwerkinfrastruktur im Internet sichtbar. Jedoch wird an einer Implementierung des NAT-Verfahrens für IPv6 erwartungsvoll gearbeitet und eine erfolgreiche Implementierung scheint möglich [Kap11]. Diese und viele weitere Gründe, wie die schrittweise Umstellung der In-

frastruktur auf das IPv6-Protokoll oder die neue Denkweise, die das IPv6-Protokoll mit sich bringt, verhindern eine große Akzeptanz bei dem Einsatz des Protokolls durch Administratoren und Benutzer [Ger09].

Jedoch bringt das IPv6-Protokoll trotz einiger Probleme und Schwierigkeiten durch seine Neuerungen einige Vorteile mit sich. Die wichtigste Erneuerung ist die Erweiterung des Adressraumes durch die 128 Bit Adressen. Durch eine intelligente hierarchische Strukturierung und optimale Vergabe der Adressräume kann ein optimales Routing in Zukunft vollzogen werden und der immense IPv6-Adressraum wird einer Knappheit an IPv6-Adressen vorbeugen.

Eine weitere Änderung wurde am Aufbau des Header-Formates vorgenommen. Dieses hat im Gegensatz zu den IPv4-Headern eine fest definierte Länge von 40 Bytes. Auf Grund des schlanken Headers kann eine schnellere und effizientere Verarbeitung der Datenpakete erfolgen. Weitere Optionen werden bei dem IPv6-Protokoll in den Extension-Headern transportiert. Durch diese flexible Extension-Header-Struktur wird die Entwicklung sowie Anpassung an neue Dienste vereinfacht.

Außerdem bietet das Protokoll neu, zu einem Netzwerk hinzugefügten Routern die Möglichkeit, sich selbst konfigurieren zu können (*Stateless-Adress-Autoconfiguration*) und sich dabei eigenständig IPv6-Adressen zu berechnen und zuzuweisen. Dabei verwenden die Router zur Erstellung der Link-Local IPv6-Adressen lokale Informationen wie die *Media-Access-Control (MAC)*-Adresse und verknüpfen diese mit den Präfixinformationen des Routers. Eine detaillierte Beschreibung zur automatischen Erstellung einer IPv6-Adresse wird in dem *RFC2464* beschrieben.

Da sich eine autokonfigurierte IPv6-Adresse anhand der eindeutigen MAC-Adresse ableiten lässt, wurde in der Öffentlichkeit der Schutz der Privatsphäre diskutiert, da die Möglichkeit besteht, Benutzer im Internet anhand der abgeleiteten IPv6-Adresse einfacher verfolgen und identifizieren zu können. Aus diesem Grund wurde im *RFC4941* ein neues Verfahren zur Autokonfiguration von IPv6-Adressen unter der Beachtung der Privatsphäre vorgeschlagen. Dabei basiert die IPv6-Adresse nicht mehr länger auf der MAC-Adresse, sondern wird anhand einer periodisch neu ge-

nerierten Zufallszahl erstellt. Dieses Verfahren ist in den meisten neueren Betriebssystemen schon implementiert.

Des Weiteren wurde bei der Entwicklung von IPv6 darauf geachtet, dass IPv6-Adressen beziehungsweise IPv6-Adressbereiche wieder zurückgezogen werden können und somit eine Neuverteilung möglich ist. Außerdem beruht das IPv6-Protokoll auf den „Ende-zu-Ende“-Verbindungen. Dadurch werden die Zwischenrouter, welche die Datenpakete weiterleiten, entlastet, da die Fragmentierung von überlangen Datenpaketen auf den Absenderroutern mithilfe des Fragment-Headers (Abschnitt 2.1.4 erfolgt [Hag09].

Das IPv6-Protokoll bringt zwei Sicherheitsmechanismen mit sich: den IP-Authentification-Header und die IP-Encapsulating-Security-Payload. Durch diese Mechanismen, welche auf dem IPsec-Framework beruhen, kann die Integrität, die Authentisierung sowie die Vertraulichkeit der Datenpakete garantiert werden. Jedoch sind in der Vergangenheit Angriffsmöglichkeiten wie *Denial-of-Service (DoS)*-Attacken mithilfe des Routing-Headers vom Typ 0 entdeckt worden. Auf dieses Problem wird im Abschnitt 2.1.5 genauer eingegangen.

Eine ausführliche Beschreibung der Sicherheitsmechanismen wird in diesem Dokument nicht vorgenommen. Weitere Informationen zu den Sicherheitsmechanismen sind in [Atk95b] und [Ken05b] zu finden.

2.1.2 IPv6-Routing

Unter IPv6-Routing ist die Festlegung des gesamten Weges für einen Nachrichtenstrom durch das Netzwerk zu verstehen. Die Weiterleitung (*Forwarding*) eines IPv6-Datenpaketes erfolgt innerhalb der einzelnen Knoten. Diese entscheiden, über welchen Nachbarknoten das Datenpaket weitergeleitet werden soll [Enz11b]. Die Weiterleitung kann entweder im lokalen Netz über Hubs/Switches realisiert werden oder die sie erfolgt über einen Router in andere Netze. Im folgenden Text wird für Hubs/Switches und Router zur Vereinfachung der einheitliche Begriff Router verwendet.

Jeder Router besitzt und verwaltet eine eigene Routing-Tabelle (For-

warding-Table). In dieser Tabelle werden folgende Informationen zu jeder IPv6-Route festgehalten [Hag09]:

Metrik
Gibt die gesamten Kosten zur Erreichung eines Zielnetzwerkes an. Die Metrik ist dabei von dem jeweiligen Routing-Protokoll abhängig. Die Router wählen bei mehreren möglichen Routen zu einem Ziel die Route mit der kleinsten Metrik.

Timer
Gibt die abgelaufene Zeit an, seitdem eine Route das letzte Mal aktualisiert wurde. Die Arten der Timer, die jeder Route zugeordnet werden, sind von dem jeweiligen Protokoll abhängig.

IPv6-Präfix und Präfixlänge
Durch die Präfixlänge wird die Anzahl der relevanten Bits für das IPv6-Präfix angegeben. Dabei werden bei der Suche nach einer übereinstimmenden Zieladresse nur diese relevanten Bits des IPv6-Präfixes berücksichtigt.

Next-Hop-Adresse
Gibt die IPv6-Adresse (auch Link-Local-Adresse genannt) des nächsten Routers auf dem Weg zum Ziel an. Eine Next-Hop-Adresse ist nicht nötig, falls die Route sich an einer direkt am Router angeschlossenen Schnittstelle befindet.

Next-Hop-Interface
Stellt die lokale angeschlossene, physikalische Schnittstelle dar. Über dieses Feld ist die Next-Hop-Adresse erreichbar.

Route-Source
Dieses Feld gibt den Namen des Prozesses an, der die Route in die Routing-Tabelle eingetragen hat. Dies kann der Name einer direkt angeschlossene Route oder eines Routing-Protokolls sein.

Beim Eintreffen eines Datenpaketes an einem Router kann dieser anhand der IPv6-Zieladresse des Paketes durch einen Vergleich mit den Einträgen in der Routing-Tabelle eine Weiterleitung veranlassen. Dafür wird die entsprechende Präfixlänge auf die Zieladresse angewendet um das Zielnetzwerk zu berechnen. Falls das berechnete Zielnetzwerk einem Präfix einer Route innerhalb der Routing-Tabelle entspricht, gibt es eine Übereinstimmung und die Weiterleitung erfolgt über diese Route. Dabei wird immer die Route mit der längsten übereinstimmenden Präfixlänge bevorzugt. Falls keine Übereinstimmung der Zieladresse in der Routing-Tabelle gefunden wird oder das Hop-Limit gleich 0 ist, wird das IPv6-Datenpaket verworfen und es folgt eine ICMPv6-Fehlermeldung an den absendenden Router.

Die Einträge der Routing-Tabellen können entweder manuell eingegeben werden (*statisches Routing*) oder mithilfe von Routing-Protokollen (*dynamisches Routing*) erstellt werden.

Des Weiteren enthält jede Routing-Tabelle mindestens eine Default-Route. Diese leitet Datenpakete weiter, deren Zieladressen nicht in der Routing-Tabelle aufgelistet sind. Die Weiterleitung der Datenpakete erfolgt an den eingetragenen Default-Router, der seinerseits die möglichen Routen kennen sollte. Falls nicht, werden die Pakete ebenfalls per Default-Route an dessen weiteren Default-Router weitergeleitet. Default-Routen, welche ebenfalls statische Routen sind, können nur durch den Administrator erstellt werden. Eine Default-Route ist bei der Suche nach einer Übereinstimmung für die Weiterleitung eines Paketes immer die letzte Route, die betrachtet wird.

Falls die Eingabe von statischen Routen umgangen werden soll beziehungsweise nicht benötigt wird, können verschiedene Routing-Protokolle eingesetzt werden. Diese besitzen den großen Vorteil, dass ein dynamischer Informationsaustausch zwischen den Routern und dessen Routing-Tabellen stattfinden kann. Dabei wird zwischen einem Informationsaustausch innerhalb eines autonomen Systems oder zwischen zwei autonomen Systemen unterschieden. Bei einem Informationsaustausch unter den Routern innerhalb eines autonomen Systems (*Intradomain-Routing*) wer-

den die sogenannten Interior-Gateway-Protokolle (IGP) wie RIPng und OSPF verwendet. Für den Austausch zwischen zwei unterschiedlichen ASes (*Interdomain-Routing*) sind die Exterior-Gateway-Protokolle (EGP) wie BGP-4 mit der IPv6-Erweiterung verantwortlich.

Die Informationen aus diesem Abschnitt wurden [Hag09] entnommen.

2.1.3 Adresslänge und Adressarchitektur

Einer der ausschlaggebenden Beweggründe für die Entwicklung von IPv6 war die Erweiterung der Adresslänge auf 128 Bit.

Die IPv6-Adressen werden in acht hexadezimalen, 16 Bit langen Blöcken dargestellt. Die einzelnen Blöcke werden durch Doppelpunkte getrennt. Die ersten vier Blöcke (64 Bit) stellen das Präfix dar. Das Präfix kann eine variable Länge zwischen 0 Bit und 64 Bit besitzen. Die letzten vier Blöcke (64 Bit) stehen für den Schnittstellen-Identifier (*Interface-ID*). Dieser besteht immer aus 64 Bit.

Da das IPv6-Protokoll über die Präfix-Schreibweise verfügt, müssen die Adressen nicht in feste Klassen unterteilt werden wie zu Anfang des IPv4-Protokolls. Die Präfixlänge gibt die Anzahl der hochwertigen (*high-order*) Bits der Adresse an und definiert die Adressbereiche. Die allgemeine Schreibweise hat das Format `IPv6-Adresse/ Präfixlänge`.

Ein Beispiel für eine IPv6-Adresse:

```
4711:0A40:0000:0000:1235:FA30:84E3:98AA
```

Zur Vereinfachung dürfen führende Nullen in einem Block gekürzt werden. Somit ändert sich das Beispiel wie folgt:

```
4711:0A40:0:0:1235:FA30:84E3:98AA
```

Eine weitere Vereinfachung zur besseren Übersicht ist das Ersetzen von aufeinanderfolgenden Null-Blöcken durch zwei Doppelpunkte. Jedoch darf dies nur einmal in einer kompletten IPv6-Adresse angewendet werden, da sonst nicht mehr bekannt ist, welche Stellen mit Nullen wieder aufgefüllt werden müssten. Am Beispiel sieht dies folgendermaßen aus:

```
4711:0A40::1235:FA30:84E3:98AA
```

Da die Umstellung von den IPv4-Adressen zu den IPv6-Adressen nur schrittweise erfolgen wird beziehungsweise momentan erfolgt, sind zwei Adresstypen für die Rückwärtskompatibilität möglich. Weitere Informationen zu der Einbettung von IPv4-Adressen in IPv6-Adressen können dem *RFC4291* entnommen werden.

IPv4-kompatible IPv6-Adressen

Mit diesem Adresstyp können IPv6-Datenpakete dynamisch über IPv4-Routinginfrastrukturen transportiert werden. Dafür besitzen die IPv6-Router spezielle IPv6-Unicast-Adressen, welche die IPv4-Adressen in den letzten 32 der 128 Bit der IPV6-Adressen speichern. Jedoch wurde dieser Adresstyp abgeschafft.

IPv4-abgebildete IPv6-Adressen

Um Adressen von IPv4-only Routern als IPv6-Adressen repräsentieren zu können, kommt dieser Adresstyp zum Einsatz. Die IPv4-Adresse wird ebenfalls in den letzten 32 Bit der IPv6-Adresse eingebettet. Jedoch werden die vorigen 16 Bit mit dem Wert 0xFFFF initialisiert. Die Abbildung 2.1 zeigt die Einbettung einer IPv4-Adresse in eine IPv6-Adresse mit diesem Adresstyp.

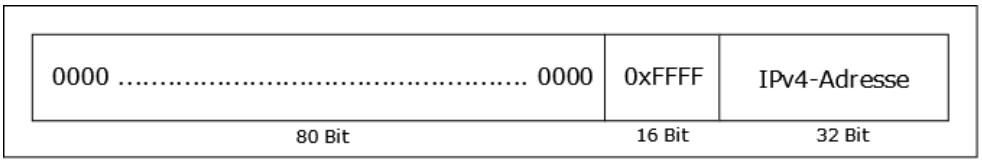

Abbildung 2.1: Abgebildete IPv4-Adresse in IPv6-Adresse [Hag09]

Des Weiteren wird bei dem IPv6-Protokoll zwischen drei verschiedenen Adressarten unterschieden.

Die erste Adressart sind die Unicast-Adressen. Diese Adressen identifizieren eine Schnittstelle eines IPv6-Routers eindeutig. Bei dem Versand eines IPv6-Datenpaketes an eine Unicast-Adresse wird das Datenpaket an

die Schnittstelle ausgeliefert, die mit dieser Adresse konfiguriert ist. Dabei wird bei den Unicast-Adressen zwischen Link-Local Unicast-Adressen, Site-Local Unicast-Adressen sowie Global Unicast-Adressen unterschieden.

Die Link-Local Unicast-Adressen und die Site-Local Unicast-Adressen stellen den Ersatz für die privaten Adressen dar. Die Link-Local Unicast-Adressen (Hex-Präfix: `FE80::/10`) werden automatisch durch die MAC-Adresse oder eine Zufallszahl konfiguriert. Jede IPv6-Schnittstelle muss im Besitz einer Link-Local Unicast-Adresse sein. Diese Adressen werden von den Routern nach außen hin nicht weiter geleitet und sind nur im gleichen Teilnetz erreichbar [Enz11a].

Ein weiterer Typ der Unicast-Adressen sind die Site-Local Unicast-Adressen (Hex-Präfix: `FEC0::/10`). Diese standortlokalen Adressen dürfen nur innerhalb der gleichen Organisation geroutet werden. Die Zuweisung der Adressen erfolgt durch die Konfiguration des lokalen Präfixes auf dem Router oder durch eine automatische Zuweisung des *Dynamic-Host-Configuration-Protokoll Version 6 (DHCPv6)*. Da die Wahl des zu verwendenden Adressraumes innerhalb des Adressbereiches beliebig war, konnte es bei Zusammenlegungen oder *Virtual-Private-Network (VPN)*-Verbindungen bei Organisation zu Überschneidungen der Adressräume kommen. Deshalb sind die Site-Local Unicast-Adressen inzwischen veraltet und finden in zukünftigen Standards keine Verwendung mehr [Enz11a]. Der Nachfolger der Site-Local Unicast-Adressen sind die Unique Local-Adressen. Diese Adressen sollten nur in lokalen, abgegrenzten Bereichen eingesetzt werden und Router sollen diese Adressen nicht in das globale Internet durchreichen. Diese Adressen stellen die privaten Adressen in IPv6-Netzwerken dar. Weitere Informationen zu diesen Adressen sind in [Hin05] zu finden.

Die Global Unicast-Adressen (Hex-Präfix: `2000::/3`) werden durch das binäre Präfix bx0010 gekennzeichnet (Abbildung 2.2) und sind weltweit einzigartige, eindeutige IP-Adressen. Eine Global Unicast-Adresse besteht aus drei Teilen: dem globalen Routing-Präfix (n Bit), der Subnetz-ID (64 - n Bit) und der Interface-ID (64 Bit). Anhand des globalen Routing-

Präfixes wird der einer Site zugewiesene Adressbereich definiert. Die Subnetz-ID (auch Subnetz oder Subnetzpräfix) identifiziert den Link innerhalb einer Site. Dabei können mehrere Subnetz-IDs einem Link zugeordnet werden. Die Interface-ID dient der Identifizierung einer eindeutigen Schnittstelle innerhalb des Subnetzes. Weitere Informationen zu den Bereichen der Global Unicast-Adressen sind auf der Webseite der IANA[1] zu finden.

Eine weitere Adressart sind die Multicast-Adressen (Hex-Präfix: `FF00::/8`). Mithilfe dieser Adressen können mehrere IPv6-Schnittstellen angesprochen werden. Bei dem Versand eines IPv6-Datenpaketes wird dieses von allen Mitgliedern der Multicast-Gruppe verarbeitet. Eine einzige Schnittstelle kann zu mehreren Multicast-Gruppen gehören. Die Multicast-Adresse setzt sich folgendermaßen zusammen: die ersten 8 Bit zeigen an, dass es sich um eine Multicast-Adresse handelt. Die nächsten 4 Bit werden für Flags, beispielsweise ob es sich um eine fest definierte, von der IANA zugewiesene Adresse handelt, verwendet. Danach wird durch die folgenden 4 Bit der Scope für die Reichweite der Multicast-Adresse angegeben. Die weiteren 112 Bit geben die Multicast Gruppen-ID an. Diese und weitere Informationen zu den Multicast-Adressen sind in [Tha02] und [Hab02] zu finden.

Die dritte Adressart stellen die Anycast-Adressen dar. Diese Adressen sind mehreren Schnittstellen an verschiedenen Routern zugewiesen. Jedoch wird ein an eine Anycast-Adresse adressiertes Datenpaket nur an eine der Schnittstellen gesendet, üblicherweise an die Nächstliegende. Anycast-Adressen werden in Situationen eingesetzt, bei denen mehrere Server oder Router denselben Dienst zur Verfügung stellen. Jedoch muss beachtet werden, dass bei der Verwendung von Anycast-Adressen der Absender keine Kontrolle darüber hat, an welche Schnittstelle das Datenpaket ausgeliefert wird. Diese Entscheidung wird auf Ebene des Routing-Protokolls getroffen. Da es sich bei der Anycast-Adresse um eine reguläre Unicast-Adresse handelt, setzt sich diese durch ein Subnetzpräfix (n Bit) sowie ein Schnittstellen-Identifier (128 - n Bit) zusammen. Jedoch sind bei dem

[1] Internet-Assigned-Numbers-Authority, URL: http://www.iana.org/

Schnittstellen-Identifier alle Bits mit dem Wert 0 belegt. Dadurch wird ein an diese Adresse gesandtes Datenpaket an einen beliebigen Router innerhalb des Subnetzes ausgeliefert. Diese und weitere Informationen zu den Anycast-Adressen können [Joh99] entnommen werden.

Generell muss jede Schnittstelle eine Unicast-IPv6-Adresse zugewiesen haben. Jedoch kann eine Schnittstelle auch mehrere IPv6-Adressen besitzen, die von verschiedenen Adresstypen sein können. Eine Kombination aus den oben genannten Adresstypen ist möglich. Außerdem besteht die Möglichkeit, eine Unicast-Adresse mehreren Schnittstellen zuzuweisen, um eine Lastenverteilung (*Load-Sharing*) über diese Schnittstellen vorzunehmen.

Bei dem IPv6-Protokoll gibt es zwei spezielle Adressen, welche aus dem reservierten Adressbereich mit dem binären Präfix bx00000000 vergeben wurden. Zum einen die unspezifizierte Adresse (*unspecified-adress*) mit dem Wert `0:0:0:0:0:0:0:0`. Diese gibt an, dass keine gültige IP-Adresse vorhanden ist und kann unter der Verwendung der oben beschriebenen Schreibweise auch folgendermaßen `::` abgekürzt werden. Während des Boot-Vorganges wird diese Adresse beispielsweise als Absenderadresse verwendet.

Zum anderen stellt die Loopback-Adresse eine spezielle Adresse dar. Sie dient bei Tests auf dem IP-Stack, ohne Datenpakete in das Netz zu senden. Dabei hat sie den Wert `0:0:0:0:0:0:0:1` beziehungsweise abgekürzt `::1` und sollte nie einer Schnittstelle zugewiesen werden.

Bestimmten IPv6-Präfixen wird eine spezielle Bedeutung zugeordnet. Die in Abbildung 2.2 dargestellte Tabelle gibt einen Überblick über die ursprünglichen Zuweisungen von reservierten Präfixen.

Anhand der Tabelle lassen sich die unterschiedlichen Präfixe durch die binäre und hexadezimale Darstellung der jeweiligen Adresstypen erkennen. Für die Anycast-Adressen besteht kein spezielles Präfix. Sie werden dem Unicast-Adressbereich entnommen. Durch die Zuweisung einer Unicast-Adresse an mehrere Schnittstellen wird diese zu einer Anycast-Adresse.

Zuweisung	Binär	Präfix in Hex
Global Unicast-Adressen	001	2000::/3
Link-Local Unicast-Adressen	1111 1110 10	FE80::/10
Unique-Local IPv6-Adressen	1111 1101	FC00::/7
Multicast-Adressen	1111 1111	FF00::/8

Abbildung 2.2: Reservierte IPv6-Präfixbereiche [Hag09]

Außerdem wird bei der IPv6-Adressarchitektur zwischen zwei verschiedenen Typen von Adressen unterschieden:

Stabile IP-Adressen
IPv6-Adressen, welche manuell, per DHCPv6 oder über die Autokonfiguration unter der Verwendung der MAC-Adresse zugewiesen worden sind.

Befristete vorübergehende IP-Adressen
Diese IPv6-Adressen werden bei der Autokonfiguration anhand einer periodisch neu generierten Zufallszahl der Schnittstelle zugewiesen.

Die Informationen aus diesem Abschnitt wurdens [Hag09] entnommen.

2.1.4 IPv6- und Extension-Header

Bei der Entwicklung von IPv6 wurde das Format des IP-Headers neu definiert. Der IPv6-Header besitzt einen einfacheren Aufbau und kann somit effizienter verarbeitet werden. Der Header besteht aus einer festen Größe von 40 Byte, von denen je 16 Byte für die Empfänger- und Absender-

adresse reserviert sind. Somit verbleiben lediglich acht Byte für die allgemeinen Header-Informationen. Um mögliche Optionen wie Security- oder Source-Routing-Optionen mitsenden zu können, werden diese in Form der neuen Extension-Header angehangen. Durch die neue Architektur der Extension-Header ist es möglich, auch in Zukunft neue, angepasste Extension-Header für weitere Anforderungen bei der Datenübertragung zu definieren.

Des Weiteren werden im folgenden Text die einzelnen Felder eines IPv6-Datenpaketes näher erläutert. Zur Anschauung des Aufbaus eines IPv6-Headers dient die Abbildung 2.3.

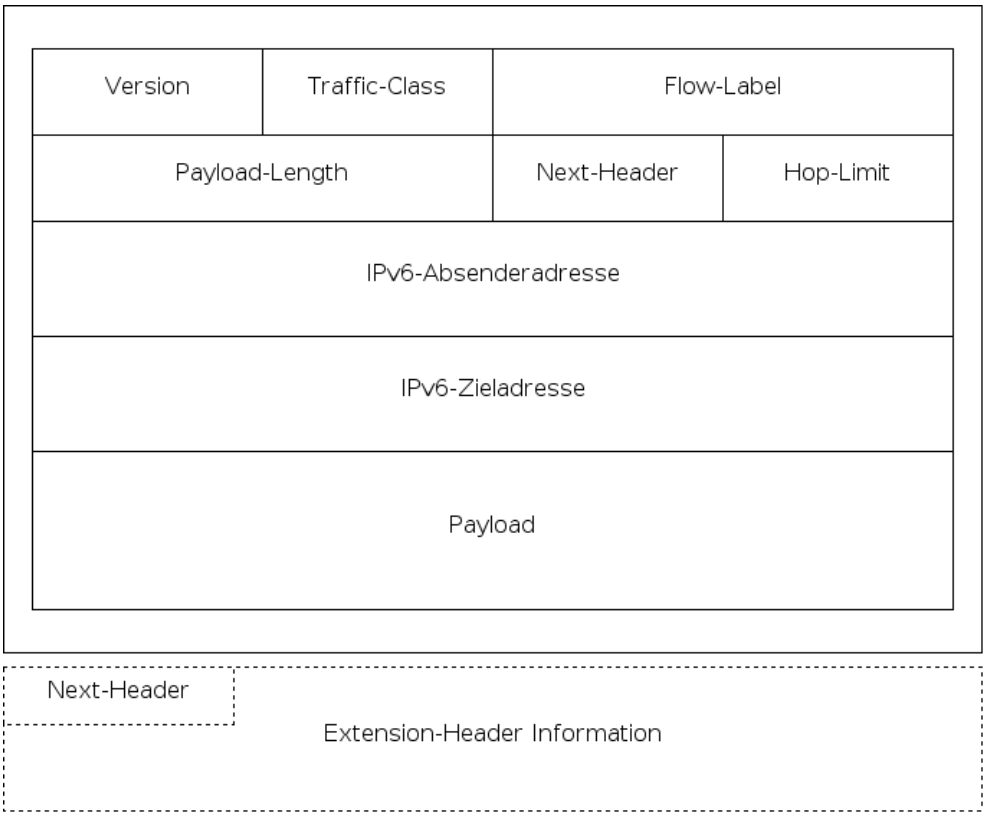

Abbildung 2.3: Aufbau eines IPv6-Headers [Hag09]

Ein IPv6-Header setzt sich aus den folgenden Feldern zusammen:

Version (4 Bit)

In diesem Feld wird die Versionsnummer des Protokolls angegeben. Bei IPv6 enthält dieses Feld dem zufolge die Zahl sechs beziehungsweise den Binärwert bx0110.

Traffic-Class (1 Byte)

In diesem Feld kann eine Angabe über die verschiedenen Prioritäten der IPv6-Pakete erfolgen. Diese Prioritäten werden von den weiterleitenden Routern erkannt. Somit können durch dieses Feld Einstellungen des *Quality of Service (QoS)* vorgenommen werden, welches Vorteile bei der Weiterleitung bestimmter Datenpakete beinhaltet.

Flow-Label (20 Bit)

Dieses Feld dient der Kennzeichnung der Pakete, die gleich behandelt werden sollen. Es stellt eine ähnliche Funktion wie das Traffic-Class-Feld dar und dient ebenfalls den *QoS*-Einstellungen. Die Verwendung des Flow-Label-Feldes befindet sich jedoch zum jetzigen Zeitpunkt noch in einem experimentellen Zustand.

Payload-Length (2 Byte)

Mithilfe dieses Feldes wird die Länge der Nutzdaten (Payload) angegeben. Dabei entspricht die Länge der Payload den Daten, die auf den IPv6-Header folgen. Datenpakete, die Steuer- oder Protokollinformationen enthalten, werden nicht dazu gezählt. Die Angabe erfolgt in Byte. Bei der Berechnung der Payload werden die Extension-Header mit einbezogen. Durch die Größe von 2 Byte ist die maximale Größe der Payload auf 64 Kilobyte beschränkt. Jedoch können für den Transport von größeren Paketen *Jumbogramme* verwendet werden.

Next-Header (1 Byte)

Dieses Feld enthält eine Protokollnummer, die den auf den IPv6-Header folgenden Header angibt. Dies kann entweder ein Protokoll-

oder ein Extension-Header sein. Die aktuellen Protokoll- und Headerwerte sind auf der IANA-Webseite[2] zu finden.

Hop-Limit (1 Byte)

Dieses Feld gibt die Anzahl der möglichen Hops wieder, über die ein Datenpaket noch weitergeleitet werden kann. Bei einer Weiterleitung des Paketes durch einen Router wird die Anzahl der Hops standardmäßig um den Wert eins dekrementiert. Falls ein Router das Paket auf den Wert null dekrementiert, schickt dieser dem Absender mit Hilfe des Internet-Control-Message-Protocol Version 6 (ICMPv6) eine Fehlermeldung zurück. ICMPv6 dient generell dem Austausch von Informationen und Fehlern innerhalb eines Netzwerkes. Der maximale Wert des Hop-Limits ist abhängig von dem jeweilig verwendeten Protokoll.

IPv6-Absenderadresse (16 Byte)

In diesem Feld ist die IPv6-Adresse des Absenders gespeichert. Weitere Informationen zu den Adresslängen und der Adressarchitektur können dem Abschnitt 2.1.3 entnommen werden.

IPv6-Zieladresse (16 Byte)

Dieses Feld enthält die IPv6-Adresse des Empfängers. Jedoch muss in diesem Feld bei dem IPv6-Protokoll nicht zwangsläufig die IP-Adresse des endgültigen Empfängers gespeichert sein. Falls ein Routing-Extension-Header zum Einsatz kommt, wird lediglich die IP-Adresse des nächsten Hops gespeichert. Durch den Routing-Header wird der weitere Verlauf des Datenpaketes definiert.

Falls mehrere Extension-Header innerhalb eines IPv6-Datenpaketes auftreten, wird eine bestimmte Anordnung für die Extension-Header vorgeschrieben. Die Extension-Header werden zwischen dem sich zu Anfang eines IPv6-Datenpaketes befindlichen IPv6-Header und dem letzten Header, dem Header der nächst höheren Schicht, eingebaut. Die sequenzielle

[2]IANA: http://www.iana.org/protocols

Abarbeitung der einzelnen Header erfolgt in der Reihenfolge, wie sie in dem gesamten IPv6-Datenpaket erscheinen. Ein Router darf sich nicht vor der Abarbeitung der Extension-Header einen Überblick über diese verschaffen, um nur einen bestimmten Extension-Header zu verarbeiten. Jeder Extension-Header darf außerdem in einem Paket nur einmal vorkommen, mit Ausnahme des Destination-Options-Headers. Dieser darf höchstens zweimal auftreten.

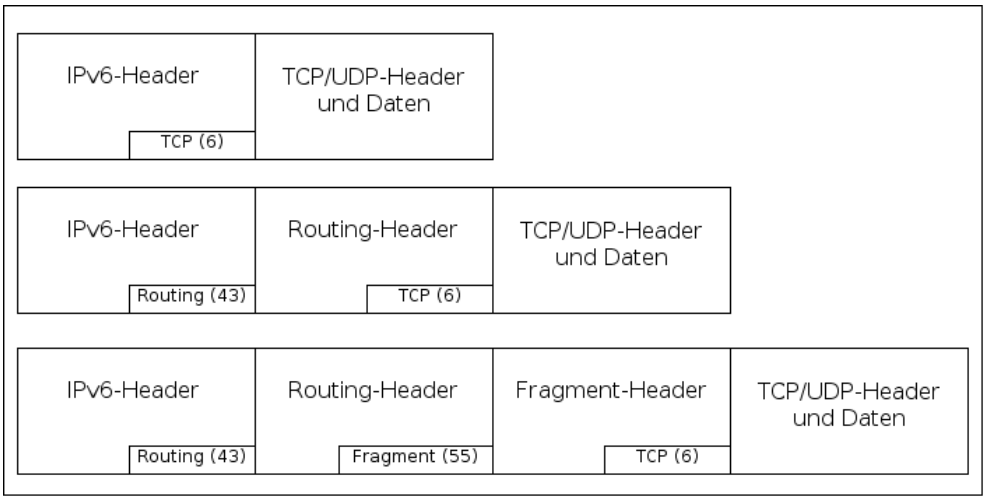

Abbildung 2.4: IPv6-Header mit Extension-Header [Bio07]

Die Abbildung 2.4 zeigt mehrere IPv6-Datenpakete mit verschiedenen Extension-Header. Dabei wird durch das Next-Header-Attribut der darauf folgende Extension-Header und dessen Protokollnummer angegeben. Der folgende Extension-Header wird in den kleineren, inneren Rechtecken angezeigt.

Die aktuellen sechs Extension-Header der IPv6-Spezifikation sind:

- Hop-by-Hop-Options-Header
- Routing-Header
- Fragment-Header
- Destination-Options-Header

- Authentication-Header
- Encapsulating-Security-Payload-Header

Der Routing-Header wird im nächsten Abschnitt genauer erläutert. Informationen zu den weiteren Headern sind in [Ken05a], [Ken05b] und [Dee98] zu finden.

2.1.5 Routing-Header

Mithilfe des Routing-Headers können die IP-Adressen der Router angegeben werden, über die das zu transportierende Datenpaket auf dem Weg zu seinem Ziel weitergeleitet werden soll. Unter IPv4 war eine Bestimmung des Weges durch den Absenderknoten ebenfalls per *Source-Routing* möglich [Enz11b]. Alle in der Liste aufgeführten Router müssen den Routing-Header verarbeiten können. Somit kann der Pfad definiert werden, über welches das Datenpaket gesendet wird. Der Routing-Header wird durch den Wert 43 im vorhergehenden Next-Header-Feld angegeben.

Die Abbildung 2.5 zeigt den Aufbau des Routing-Headers.

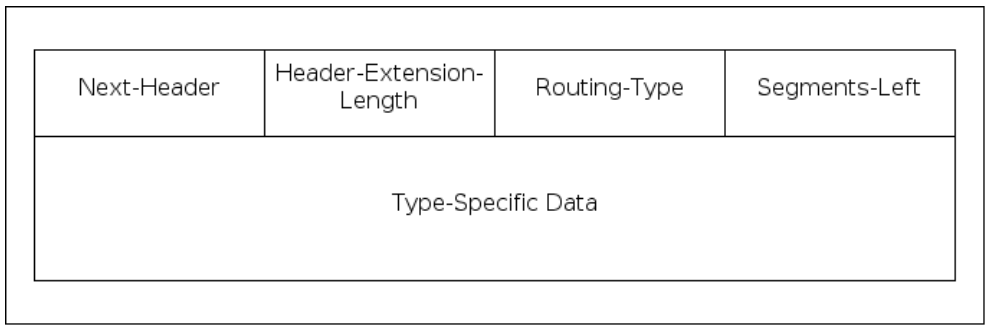

Abbildung 2.5: Aufbau eines Routing-Headers [Dee98]

Dabei setzt sich ein Routing-Header aus folgenden Feldern zusammen:

Next-Header (1 Byte)

Dieses Feld gibt den Typ des nächsten Headers an, der auf den Routing-Header folgt. Die aktuellen Protokoll- und Headerwerte sind auf der IANA-Webseite zu finden.

Header-Extension-Length (1 Byte)

Die Länge des Routing-Headers wird in diesem Feld angegeben. Die Angabe erfolgt dabei in acht Byte Einheiten. Dabei werden die ersten acht Byte nicht mitberechnet.

Segments-Left (1 Byte)

Diese Feld gibt die Anzahl der restlichen Router an die passiert werden müssen, bis das Datenpaket sein endgültiges Ziel erreicht.

Routing-Type (1 Byte)

Durch dieses Feld wird der Routing-Typ definiert. Die möglichen Typen werden im weiteren Text dieses Abschnittes erläutert.

Type-Specific Data (variable Größe)

Durch den Routing-Typ wird der Aufbau und die Größe dieses Feldes definiert. Dabei ist die Größe des Feldes immer ein Vielfaches von acht Byte. In dieses Feld werden Daten wie die IP-Adressen der weiteren Router für den Transport geschrieben.

Falls ein angegebener Router auf dem Weg zum Ziel den Routing-Typ nicht identifizieren kann, ist das weitere Verhalten abhängig von dem Wert im Segments-Left-Feld. Bei dem Wert null ignoriert der aktuelle Knoten den Routing-Header und fährt mit dem nächsten Header fort. Dieser wird im Next-Header-Feld des Routing-Headers angegeben. Bei einem Wert ungleich null im Segments-Left-Feld wird das Datenpaket verworfen und es erfolgt die Benachrichtigung durch eine ICMPv6-Nachricht „Parameter Problem“ an den Absender.

Falls ein Knoten nach der Verarbeitung des Routing-Headers merkt, dass die MTU zur Erreichung des nächsten Knotens kleiner als die Größe des Datenpaketes ist, wird das Datenpaket ebenfalls von dem Knoten verworfen und eine ICMPv6-Nachricht „Too Big“ an den Absender des Datenpaketes gesendet.

Bei dem Routing-Typ wird zwischen drei Typen unterschieden:

Typ 0

Dieser Typ stellt eine erweiterte Version des IPv4 Loose-Source-Routing dar. Jedoch können durch diesen Routing-Header-Typ DoS-Angriffe erzeugt werden. Deswegen wird dieser Header-Typ von Betriebssystemen sowie Routern nicht mehr verarbeitet (deprecated [3]) und Datenpakete, welche diesen Typen enthalten, verworfen werden [Bio07] [J.A07].

Typ 1

Wurde in einem Projekt namens *Nimrod* von DARPA[4] definiert. Dieser Typ wird seit Mai 2009 [5] nicht mehr verwendet.

Typ 2

Dieser Typ wird für Mobile-IPv6 (MIPv6) benutzt. Der Header-Typ erlaubt über eine temporäre IPv6-Adresse (*care-of adress*) das direkte Routing zu dem Mobilknoten. Die temporäre IPv6-Adresse wird als IPv6-Adresse angegeben. Wenn ein Datenpaket die temporäre Adresse erreicht, erhält der mobile Knoten die endgültige IPv6-Adresse von dem Routing-Header. Außerdem dürfen Routing-Header vom Typ 2 nur eine IPv6-Adresse beinhalten und jeder Router, der solch einen Header empfängt, muss prüfen, ob die Absenderadresse die richtige IP-Adresse des Absenders ist [Per11]. Somit sind DoS-Attacken mit diesem Header-Typ nicht möglich.

Die folgende Abbildung 2.6 zeigt einen beispielhaften Verlauf eines Datenpaketes in der in Abschnitt 4.1 beschriebenen Y-Topologie. Die Route des IPv6-Datenpaketes wird durch die Einträge der IPv6-Adressen im Routing-Header vorgegeben. Dabei wird im *src*-Feld (source) die IPv6-Adresse `4711::12` angegeben, von der das Datenpaket versendet wird. Das Feld *dst* (destination) enthält die nächste Zieladresse. Dies muss nicht

[3]IANA: http://www.iana.org/assignments/ipv6-parameters/
[4]DARPA: Defense-Advanced-Research-Projects-Agency, http://www.darpa.mil/
[5]IANA: http://www.iana.org/assignments/ipv6-parameters/

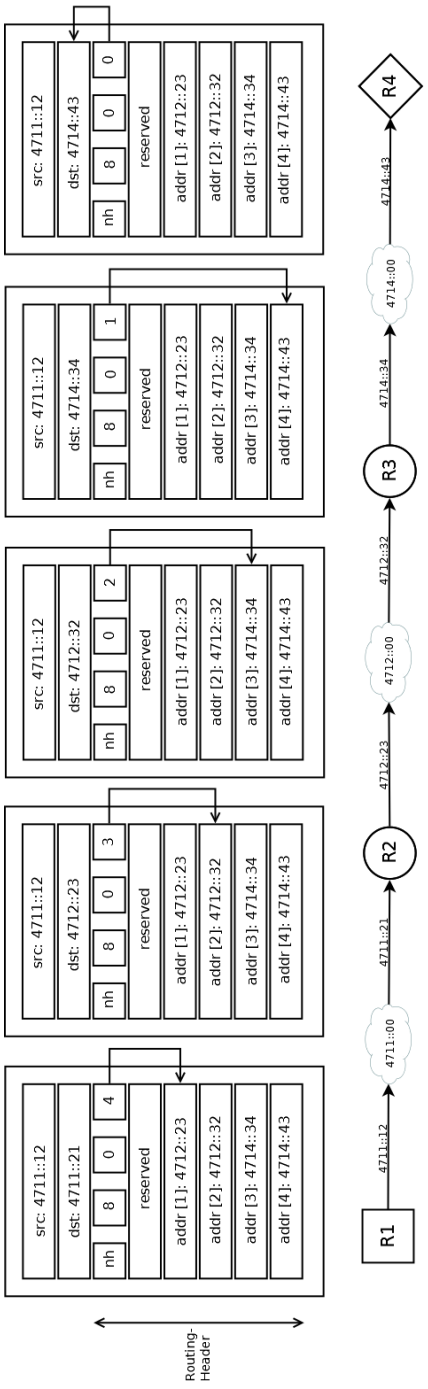

Abbildung 2.6: RH-Verlauf [Bio07]

die endgültige Zieladresse sein. Abhängig von dem Wert im Segments-Left-Feld wird das Datenpaket anhand der nächsten IPv6-Adresse in den Adressfeldern des Routing-Headers weitergeleitet. Nach der Abarbeitung einer IPv6-Adresse, wird mit der nächsten IPv6-Adresse fortgefahren. Somit kann der Verlauf des IPv6-Datenpaketes durch das Netzwerk gesteuert werden.

Das Segements-Left-Feld gibt die Anzahl der noch folgenden Router auf Grund der Anzahl der IPv6-Adressen im Routing-Header an. Nach dem Erreichen einer weiteren Zieladresse wird das Segment-Left-Feld dekrementiert. Falls eine Dekrementation auf den Wert null vorliegt, ist das Datenpaket bei seinem Ziel angekommen und passiert keine weiteren Router mehr.

Da in dem Routing-Header vier IPv6-Adressen angegeben werden, muss die Header-Extension-Length mit dem Wert von acht Byte initialisiert werden. Somit können vier Adressen angesprochen und adressiert werden. Die Angabe des Routing-Header Typ 0 erfolgt durch den Wert null im Routing-Header-Feld des Datenpaketes.

In dem mit *nh*-initialisiertem Feld wird der Next-Header angegeben. Der Verlauf des Datenpaketes beruht auf der in Abschnitt 4.1 dargestellten Y-Topologie.

Der Routing-Header Typ 0 (RH0) stellt dahingehend eine Bedrohung dar, da eine eindeutige IPv6-Adresse mehrmals innerhalb des Routing-Header-Datenpaketes angegeben werden kann. Somit können die IPv6-Datenpakete bei einem Angriff so aufgebaut werden, dass diese Datenpakete zwischen zwei Routern abwechselnd hin und her gesendet werden. Dies bedingt eine Überlastung der Routen und kann somit als DoS-Attacke eingesetzt werden. Dadurch kann es zum Ausfall oder zu Leistungseinbrüchen innerhalb des Netzwerkes kommen.

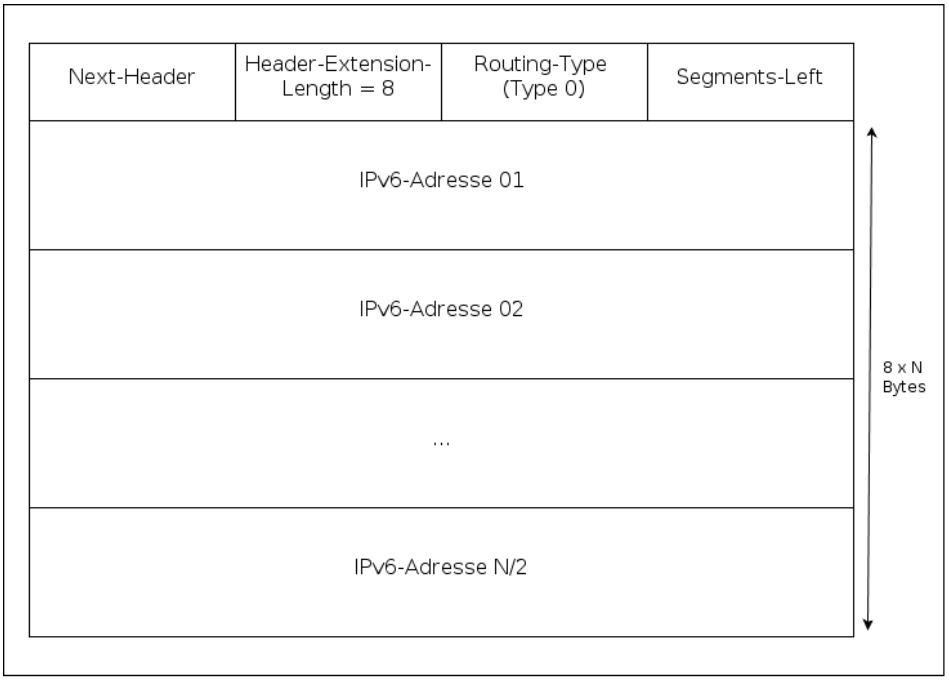

Abbildung 2.7: Aufbau eines Routing-Headers Typ 0 [Bio07]

In Abbildung 2.7 wird ein Routing-Header vom Typ 0 dargestellt. Die Felder sind identisch zu denen in Abbildung 2.5. Jedoch ist das Typ-Feld mit dem Wert null initialisiert und die in Abbildung 2.5 typenspezifischen Daten stellen die einzelnen IPv6-Adressen dar, welche auf dem Weg zum Ziel passiert werden sollen.

Zwar bietet das IPv4-Protokoll durch seine Source-Route Optionen eine ähnliche Option an, jedoch kann nur eine geringe Anzahl an zwischenliegenden IP-Adressen im Gegensatz zu dem IPv6-Protokoll und dessen Routing-Header angegeben werden.

Da die Probleme des Routing-Headers Typ 0 schon im Jahre 2001 entdeckt worden sind, wurden im Jahr 2002 neue Vorschläge für die Verarbeitung der Routing-Header gemacht. Diese Änderungen sind in die Spezifikation des mobilen IPv6-Protokolls eingeflossen. Aus diesem Grund wurde der Routing-Header Typ 2 entwickelt, um Routing-Header vom Typ 0 strikt abzulehnen und Angriffe zu verhindern.

Eine weitere Sicherheitsvorkehrung ist die richtige Einstellung der Firewall. Zwar können Firewalls alle IPv6-Datenpakete, welche einen Routing-Header mit sich führen, blocken, jedoch würde damit eine fortschreitende Entwicklung weiterer IPv6-Routing-Header unterdrückt. Deshalb muss standardmäßig die Weiterleitung der IPv6-Datenpakete, welche den Typ 0 Routing-Header enthalten, unterbunden werden. Alle IPv6-Datenpakete mit Routing-Header-Typen ungleich dem Typ 0 werden weitergeleitet. Außerdem wird eine Verarbeitung von Routing-Headern Typ 0 durch die aktuellen Betriebssysteme komplett abgelehnt beziehungsweise besteht die Möglichkeit, die Verarbeitung der Header standardmäßig abzuschalten.

Die Informationen zu diesem Abschnitt wurden [Bio07], [J.A07] und [Dee98] entnommen.

2.2 Routing-Algorithmen

In diesem Abschnitt werden zwei Routing-Algorithmen vorgestellt. Zum einen der Distanz-Vektor-Algorithmus und zum anderen der Link-State-Algorithmus. Beide Algorithmen dienen der Berechnung kürzester Routen innerhalb eines Netzwerkes. Sie unterscheiden sich hinsichtlich des Verfahrens und der Vorgehensweise zur Berechnung der Routen.

2.2.1 Distanz-Vektor-Algorithmus

Bei dem Distanz-Vektor-Algorithmus besitzt jeder Router eine eigene Datenbasis, seine Routing-Tabelle. In dieser Tabelle werden Routen zu anderen Netzwerken oder Routern abgespeichert. Jeder Router besitzt eine eigene Sichtweise auf seine Nachbarrouter. Die Router tauschen in periodischen Zeitintervallen, abhängig von dem verwendeten Routing-Protokoll, Informationen untereinander über die Erreichbarkeit der verschiedenen Netzwerke aus. Dabei enthalten die Datenpakete eine Menge an Paaren (P,D). Der Buchstabe P gibt das Präfix für das Zielnetz an, der Buchstabe D repräsentiert die Distanz (Metrik) aus Sicht des Senders zu dem Zielnetz. Die Metrik wird durch die Anzahl der Router auf dem Weg zum Zielnetz festgelegt. Die direkten Nachbarrouter haben standardmäßig einen Metrikwert von eins; eine Route, bei der die Metrik größer gleich dem Wert 16 ist, gilt als unerreichbar und kann nicht verwendet werden. Routen dürfen daher maximal eine Metrik von 15 aufweisen.

Da bei dem Distanz-Vektor-Algorithmus jeder Router den Inhalt seiner Routing-Tabelle als Distanz-Vektor in periodischen Zeitintervallen an seine Nachbarrouter sendet, kann ein Abgleich der kürzesten Wege innerhalb des Netzwerkes erfolgen. Die Berechnung dieser Wege beruht dabei auf dem mathematischen Bellman-Ford-Algorithmus. Dieser wird in der Graphentheorie zur Berechnung der kürzesten Wege eingesetzt. Falls Router A einen kürzeren Weg zu einem bestimmten Ziel kennt, der Router B bis zu dem Empfang der Update-Nachricht von Router A noch nicht bekannt war, aktualisiert Router B seinen Distanz-Vektor-Eintrag in seiner Routing-Tabelle. Dafür passt er die neue Entfernung für das Zielnetz entsprechend an.

Durch die regelmäßigen Updates der einzelnen Router wird verhindert, dass die mit den Routen assoziierten Timer ablaufen und somit gelöscht werden (*Timeout-Mechanismus*). Falls für einen Eintrag innerhalb der Routing-Tabelle in einer bestimmten Zeit keine Aktualisierung empfangen wird, wird der Distanz-Vektor und damit der Eintrag in der Routing-Tabelle gelöscht. Dies soll verhindern, dass ein Router weiterhin eine un-

gültige Route an seine Nachbarrouter verteilen kann.

Der Distanz-Vektor-Algorithmus findet vor allem Verwendung bei den Routing-Protokollen der RIP-Familie und wird für kleinere Netzwerke mit geringen Ausfallwahrscheinlichkeiten genutzt. Für größere Netzwerke ist der Algorithmus auf Grund seiner hohen Konvergenzzeit bei Veränderungen des Netzwerkes ungeeignet, da Änderungen, wie zum Beispiel ein Ausfall eines Routers, durch die regelmäßigen Updates im Netzwerk nur langsam verbreitet werden [Klo05].

Die Informationen stammen, soweit nicht anders vermerkt, aus [Car02] und [Hed98].

2.2.1.1 Count-to-Infinity

Ein Problem des Distanz-Vektor-Algorithmus ist das Count-to-Infinity-Problem (CTI-Problem). Dieses Problem beschreibt das Hochzählen der Hop-Anzahl/Metrik bis zur Unendlichkeit. Die Unendlichkeit wird durch den Grenzwert des verwendeten Protokolls angegeben. Wenn der Grenzwert erreicht wird, wird der Löschvorgang für die Route gestartet. Der Grenzwert bei den Protokollen der RIP-Familie beträgt immer 16 Hops.

Bei dem Count-to-Infinity-Problem werden in bestimmten Situationen falsche Einträge über die Hop-Anzahl/Metrik in die Routing-Tabellen geschrieben. Ein ausführliches Beispiel für das CTI-Problem unter der Verwendung von RIPng sowie IPv6 wird im Testszenario 4.4 aufgeführt.

Des Weiteren trägt das CTI-Problem dazu bei, dass sich Informationen innerhalb eines Netzwerkes nur sehr langsam verbreiten. Somit vergeht bei dem Ausfall einer Route beziehungsweise eines Routers in größeren Netzwerken viel Zeit, bis die Nachricht über den Ausfall allen Routern bekannt ist. Diese verstrichene Zeit ist die sogenannte Konvergenzzeit innerhalb eines Netzwerkes.

Die Konvergenzzeit und das CTI-Problem können durch folgende Maßnahmen reduziert werden:

- Split-Horizon (Abschnitt 2.2.1.2)
- Split-Horizon mit Poison-Reverse (Abschnitt 2.2.1.2)

- Triggered-Updates (Abschnitt 3.6.1.2)

Die Maßnahmen werden in den angegebenen Abschnitten ausführlich erläutert. Dabei stellt das Split-Horizon-Verfahren optional mit Poison-Reverse eine Maßnahme dar, um das Auftreten des CTI-Problems auf ein Minimum zu beschränken. Dieses Verfahren wird in dem nächsten Abschnitt genauer erläutert. Die Triggered-Updates helfen die Konvergenzzeit zu verringern, indem Änderungen innerhalb der Routing-Tabelle sofort weitergesendet werden.

Die Informationen aus diesem Abschnitt wurden, soweit nicht anders vermerkt, [Mah07] entnommen.

2.2.1.2 Split-Horizon und Poison-Reverse

Eine Lösung zur Verhinderung des Count-to-Infinity-Problems und der damit verbundenen hohen Konvergenzzeit stellt das Split-Horizon-Verfahren dar. Mit diesem Verfahren wird verhindert, dass ein Router Datenpakete zu einem benachbarten Router über die gleiche Schnittstelle (Next-Hop-Schnittstelle) zurücksenden darf, über welche er die Datenpakete des Nachbarrouters erhalten hat. Somit können Schleifen zwischen zwei Routern (*Two-Hop-Loops*) vermieden werden.

Des Weiteren bietet das Split-Horizon-Verfahren eine zusätzliche Option, das sogenannte Poison-Reverse-Verfahren (blockierte Rückroute). In diesem Verfahren darf die gleiche Route über die Next-Hop-Schnittstelle unter der Bedingung zurückgesendet werden, dass die Metrik den Wert 16 enthält. Somit kann eine schnellere Verbreitung über den Ausfall eines Routers im Netzwerk stattfinden und die Konvergenzzeit verkleinert werden [Mah07]. Jedoch werden für das Poison-Reverse-Verfahren zusätzliche Response-Nachrichten benötigt und damit verbunden auch zusätzliche Bandbreite. Außerdem dürfen die beiden Verfahren in bestimmten Topologien wie den Punkt-zu-Punkt Topologien nicht eingesetzt werden, da sonst ein reibungsloser Austausch der Datenpakete nicht mehr möglich ist.

Bei beiden Verfahren besteht das Risiko, dass diese bei größeren Schleifen in komplexen Netzwerken nicht mehr zuverlässig arbeiten [Mah07]. Des Weiteren muss das Split-Horizon-Verfahren mit oder ohne dem Poison-Reverse-Verfahren an jeder Schnittstelle separat eingeschaltet werden.

2.2.2 Link-State-Algorithmus

Bei dem Link-State-Algorithmus werden die Routing-Tabellen und die Zustände der einzelnen Verbindungen der Router nach dem mathematischen Dijkstra-Algorithmus berechnet, welcher ebenfalls zur Berechnung der kürzesten Pfade innerhalb eines Netzwerkes eingesetzt wird und unter die Shortest-Path-Algorithmen fällt. Jeder Router kennt die gesamte Topologie des Netzwerkes. Somit haben alle Router die gleiche Sicht auf das Netzwerk. Deshalb kann jeder Router die kürzeste Strecke zu einem anderen Router auf Basis seiner eigenen Routing-Tabelle und der Topologie des Netzwerkes berechnen. Das IS-IS-Protokoll (Intermediate-System-to-Intermediate-System-Protocol) sowie das OSPF-Protokoll (Open-Shortest-Path-First) gehören zu den Link-State-Protokollen und basieren auf dem Link-State-Algorithmus.

Die Speicherung der gesamten Topologie und aller Routen stellt einen Hauptunterschied zu dem Distanz-Vektor-Protokoll dar, da hier nur die Routen zu den Nachbarroutern in den Routing-Tabellen festgehalten werden. Außerdem enthalten die Routing-Nachrichten bei dem Versand nicht eine Menge an erreichbaren Zielnetzen wie bei dem Distanz-Vektor-Protokoll, sondern Informationen über die Zustände der Verbindungen zu den direkten Nachbarroutern. Um den Zustand der Verbindungen zu den Nachbarroutern zu testen, werden fortlaufend in bestimmten Zeitintervallen kurze Nachrichten durch das Hello-Protokoll ausgetauscht. Dadurch kann ausgewertet werden, ob eine Verbindung noch funktionstüchtig (*up*) oder gestört (*down)* ist. Da die Hello-Nachrichten über jeden Link ausgesendet werden, können somit neue Nachbarrouter entdeckt werden und eine Aushandlung über die Parameter der Verbindung ist möglich.

Des Weiteren treten bei diesem Algorithmus keine Routing-Schleifen

auf. Dies wird durch die gleiche Sicht aller Router auf das Netzwerk und dessen Topologie bedingt. Falls jedoch der Abgleich der Netzwerkpläne nicht erfolgreich war, kann dies nicht mehr garantiert werden und Routing-Schleifen können die Folge sein.

Zu Beginn erhält jeder neue Router im Netzwerk alle Informationen aus der Routing-Tabelle der Nachbarrouter, um sich möglichst schnell in das Netzwerk integrieren zu können und seine Routing-Tabelle zu füllen. In dieser Zeit kann es zu einem hohen Synchronisationsaufwand und damit verbunden zu einer hohen Netzwerkbelastung kommen, da die Router sich mit der Topologie des Netzwerkes vertraut machen müssen.

Im weiteren Verlauf reicht es jedoch aus, wenn die Router sich nur noch bei Zustandsänderungen innerhalb ihrer Routing-Tabellen gegenseitig informieren. Bei Änderungen der eigenen Zustände in der Routing-Tabelle sendet der Router Nachrichten an alle benachbarten Router, in denen die Zustände vorhanden sind und einer Änderung bedürfen. Diese Vorgehensweise ist unter dem Namen *Link-State-Updates* bekannt. Nach einer Überprüfung des neuen Zustandes durch die Router senden diese die Änderungen an ihre Nachbarrouter weiter. Dies geschieht solange, bis das gesamte Netzwerk mit den Änderungen durchflutet wurde (*Flooding*). Bei Eingang eines solchen Link-State-Updates verwendet der Router die darin enthaltenen Informationen, um seinen Netzwerkplan zu aktualisieren.

Verbindungen, bei denen sich der Zustand durch ein Update ändert, werden mit Hilfe des Shortest-Path-Algorithmus von Dijkstra neu berechnet. Diese Berechnungen beruhen auf der Darstellung des Netzwerkes als ein Graph mit einer bestimmten Menge an Knoten (Router) und einer bestimmten Menge an Kanten (Verbindungen zwischen den Routern). Die Berechnung für den „besten“ Weg zu einem anderen Knoten hängt von der konkreten Maßeinheit ab, mit der die Verbindungen innerhalb des Graphen gekennzeichnet sind. Der genaue Algorithmus von Dijkstra ist unter [Car02] zu finden.

Der Link-State-Algorithmus bietet somit einige Vor- wie auch Nachteile im Vergleich zu dem Distanz-Vektor-Algorithmus. Ein großer Vorteil sind die internen Berechnungen der kürzesten Routen auf Grund des

Topologiewissens und der Statusinformationen eines jeden Routers. Somit sind die Router nicht von den Berechnungen anderer Router abhängig und die Wahrscheinlichkeit, eine schnelle Konvergenz innerhalb des Netzwerkes zu erreichen, ist um ein Vielfaches höher. Jedoch bringt dies einen erhöhten Rechenaufwand bei der Berechnung der Routen mit sich, da die Berechnungen nach jeder Änderung der Topologie durchgeführt werden müssen und bei größeren Netzwerken eine nicht zu vernachlässigende Belastung des Netzwerkes auftreten kann. Außerdem müssen die Router für die Speicherung der kompletten Netzwerkinformationen und der Topologie eine größere Anzahl an internem Speicherplatz vorweisen.

Die Informationen aus diesem Abschnitt wurden [Car02] entnommen.

2.3 RIP-Version 2

Das dynamische Routing-Information-Protokoll–Version 2 (RIPv2) basiert auf dem Distanz-Vektor-Algorithmus zur Berechnung der kürzesten Routen innerhalb des Netzwerkes. Durch den Einsatz von dynamischen Routing-Protokollen wie RIPv2 können die Router innerhalb des Netzwerkes ihre Routing-Tabellen selbstständig füllen und Informationen über die Routen untereinander austauschen. Eine manuelle Konfiguration der Routen zwischen den Routern entfällt. Da das RIPv2-Protokoll nur dem Routing innerhalb eines autonomen Systems dient, wird es den Interior-Gateway-Protokollen (IGP) zugeordnet. Außerdem wurde das RIPv2-Protokoll für den Einsatz in kleinen bis moderaten Netzwerken entwickelt.

RIPv2 verwendet zur Übertragung der Datenpakete das UDP-Protokoll. Jeder Router, der das RIPv1- oder RIPv2-Protokoll anwendet, erhält die zu verarbeitenden Datenpakete über den RIPv1-/RIPv2-Port (520). Die Update-Nachrichten werden ebenfalls von diesem Port aus versendet. Da RIPv2 einige Erneuerungen in Hinsicht zu RIPv1 mit sich bringt, musste das Nachrichtenformat entsprechend angepasst werden.

RIPv2 verwendet ein dem RIPv1 ähnliches Nachrichtenformat. Die beiden Protokolle unterscheiden sich lediglich in dem Aufbau und der da-

mit verbundenen Funktionalität der RIP-Einträge. Das Nachrichtenformat wird in Abbildung 2.8 dargestellt. Jedes Datenpaket wird durch einen RIP-Header gekennzeichnet. Dieser setzt sich aus dem command-Feld und dem version-Feld zusammen. Mithilfe des command-Feldes kann zwischen einer Request- oder einer Response-Nachricht unterschieden werden. Eine Request-Nachricht wird durch den Wert eins, eine Response-Nachricht durch den Wert zwei gekennzeichnet. Das version-Feld gibt an, ob es sich bei den Einträgen um RIPv1- oder RIPv2-Einträge handelt. Hier muss für die Verwendung des RIPv2-Protokolls der Wert zwei eingetragen sein. Die restlichen Einträge dienen der Angabe der *Routing-Tabellen-Einträge (RTE)*. Der Inhalt und Aufbau der RTE ist vom verwendeten Protokoll abhängig.

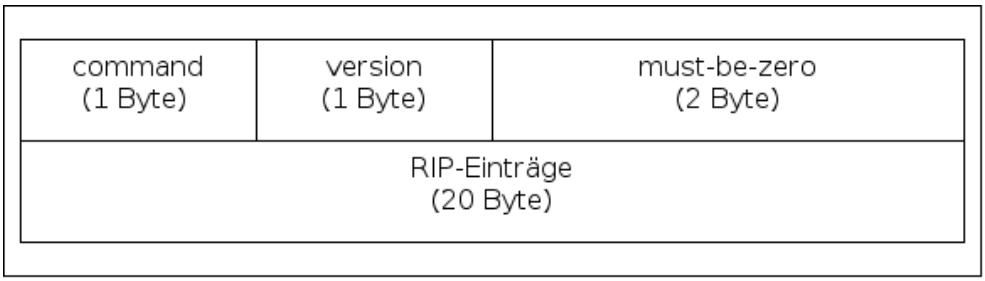

Abbildung 2.8: Aufbau eines RIPv2-Nachrichtenpaketes [Mal98]

Ein RIP-Datenpaket darf einen bis einschließlich 25 RIP-Einträge enthalten. Jeder RIP-Eintrag besteht aus der Angabe eines Adress-Family-Identifiers (AFI), einer IPv4-Zieladresse, der Subnetz-Maske, der Next-Hop-Adresse sowie der Metrik. Mithilfe des AFI wird der Typ der Adresse bestimmt. Das Metrikfeld enthält einen Wert zwischen eins und einschließlich 15 zur Angabe der Routingkosten zum Ziel. Ein Wert von 16 zeigt eine unerreichbare Route auf. Durch die IPv4-Adresse und die Subnetz-Maske wird das Zielnetzwerk angegeben. Die Next-Hop-Adresse gibt den nächsten Router bei der Weiterleitung des Datenpaketes zum Ziel hin an. Die Absicht des Next-Hop-Feldes ist, dass Datenpakete auf dem Weg zu ihrem Ziel unnötige Router bei der Weiterleitung vermeiden. Des Weiteren wird das must-be-zero-Feld von RIPv1 durch das Route-Tag-

Feld ersetzt. Bei den RIPv1-Einträgen wird dieses Feld nicht verwendet. Das Route-Tag-Feld dient der Unterscheidung von internen und externen RIPv2-Routen. In diesem Feld können Router, die ein anderes Protokoll als das RIP-Protokoll verwenden, ihre Route-Tags setzen. Abbildung 2.9 stellt den Aufbau eines RIPv2-RTEs dar.

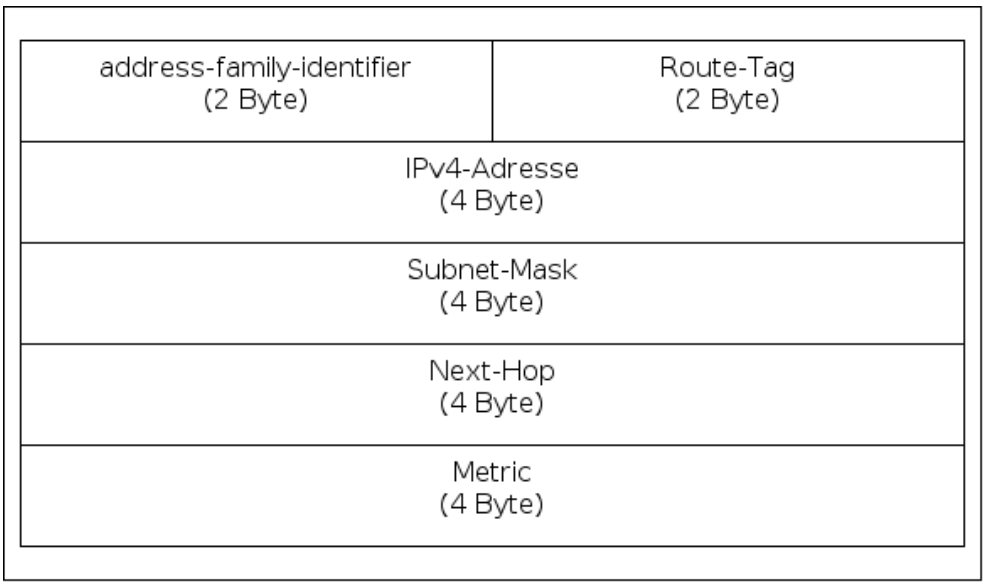

Abbildung 2.9: Aufbau eines RIPv2-RTEs [Mal98]

Bei der Weiterentwicklung von RIPv1 zu RIPv2 wurde das Routing-Protokoll nicht komplett neu gestaltet. Lediglich wichtige neue Erweiterungen wurden der RIP-Version 2 hinzugefügt. Die wichtigste Erweiterung ist die Unterstützung der CIDR-Technik. Durch diese Technik sind keine starren Einteilungen der IPv4-Adressen in bestimmte Klassen (A/-B/C/-Klassen) mehr nötig und die verfügbaren Netzbereiche können besser ausgenutzt werden.

Eine weitere Erneuerung ist die mögliche Authentifizierung. Für die Authentifizierung wird ein kompletter RIPv2-Eintrag benötigt. Dieser wird im AFI-Feld mit dem Wert 0xFFFF initialisiert und muss der erste RIPv2-Eintrag der Nachricht sein. Somit können noch maximal 24 weitere RIPv2-Einträge in der Nachricht angegeben werden. Das must-be-zero-Feld wird in diesem Fall zu dem Authentification-Type-Feld. Dieses Feld gibt den

Authentifizierungstypen an. Dies ist ein einfaches Passwort vom Type 2. Die weiteren 16 Bytes enthalten das Passwort im Klartext. Falls das Passwort kleiner als 16 Byte ist, wird es linksbündig ausgerichtet und mit angehängten Nullen aufgefüllt. In Abbildung 2.10 wird ein Authentifizierungseintrag aufgezeigt.

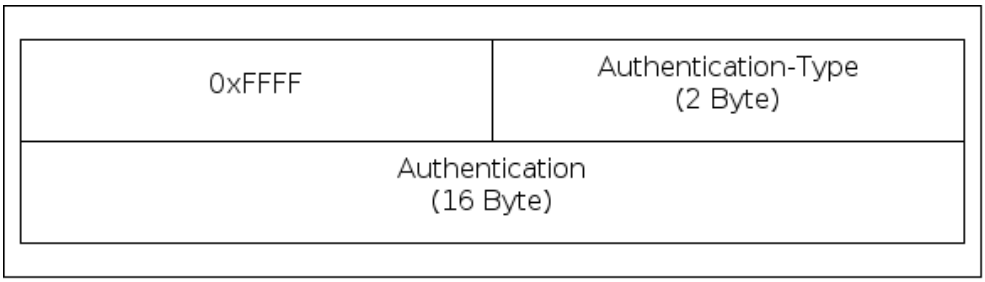

Abbildung 2.10: Aufbau eines Authentifizierungseintrages [Mal98]

Weitere Erneuerungen des RIPv2-Protokolls sind die Übermittlung der Datenpakete per Multicast und die Einführung von Triggered-Updates. Diese verkürzen die Konvergenzzeit innerhalb der Netzwerke und leiten Änderungen innerhalb der Routing-Tabellen sofort weiter.

Jedoch enthält das RIPv2-Protokoll noch einige Beschränkungen. Zum einen darf der längste Pfad innerhalb des Netzwerkes maximal 15 Hops betragen. Außerdem basiert das Protokoll bei der Entdeckung von Schleifen innerhalb des Netzwerkes auf dem Counting-to-Infinity-Problem (Abschnitt 2.2.1.1). Eine weitere Einschränkung ist die Benutzung von fixen Metriken für die Angabe der Distanzen der Routen. Bei diesen Metriken werden Echtzeitparameter wie die Zuverlässigkeit oder die Latenzzeiten der Routen nicht beachtet. Deswegen sollte es sich um homogene Netzwerke handeln, bei denen die Verbindungen gleiche Eigenschaften aufweisen.

Zur Begründung der Entwicklung von RIPv2 ist zu sagen, dass durch das Erscheinen anderer dynamischer Routing-Protokolle wie OSPF oder IS-IS die Meinung vertreten wurde, dass Protokolle der RIP-Familie für manche Anwendungen und Einsatzzwecke zu alt und nicht mehr brauchbar seien. Allerdings sind die Protokolle der RIP-Familie einfach zu im-

plementieren und auf fast allen Routern verfügbar. Somit werden die Protokolle der RIP-Familie heutzutage noch in vielen Bereichen des Routings innerhalb autonomer Systeme eingesetzt und konnten sich mit den neuen eingebauten Features gegenüber anderen Routing-Protokollen beweisen.

RIPv1 und RIPv2 sind ausschließlich für den Einsatz innerhalb von IPv4-Netzwerken gedacht. Da jedoch das IPv6-Protokoll in den letzten Jahren mehr und mehr an Bedeutung eingenommen hat und eine schrittweise Integration des IPv6-Protokolls auf Grund der in Abschnitt 2.1.1 beschriebenen Probleme unausweichlich ist, wurde für die RIP-Familie und unter Einsatz des Distanz-Vektor-Algorithmus das dynamische RIPng-Protokoll entwickelt. Dieses wird in Kapitel 3 ausführlich analysiert und erläutert. RIPng stellt somit das Routing-Protokoll der RIP-Familie für IPv6-Netzwerke dar.

RIPv2 konnte nicht an die IPv6-Adressen angepasst werden, da es auf der Grundlage der IPv4-Adressen aufgebaut wurde und neue Adressblöcke nicht ohne den Verlust der Kompatibilität zu den IPv4-Adressen möglich gewesen wäre [Ger09]. Jedoch basiert RIPng auf der Grundlage von RIPv1 und RIPv2, auch wenn einige Erweiterungen wie die Authentifizierung von RIPv2 standardmäßig im IPv6-Protokoll implementiert sind und somit für das RIPng-Protokoll überflüssig geworden sind.

Die Informationen aus diesem Abschnitt wurden [Hed98] sowie [Mal98] entnommen.

Kapitel 3

RIPng

RIPng (Routing-Information-Protocol–Next-Generation) ist ein dynamisches Routing-Protokoll der RIP-Familie. Die Versionen RIPv1 und RIPv2 sind dynamische Routing-Protokolle, die für das Routing mit dem Internet-Protokoll Version 4 (IPv4) ausgelegt sind. Das neue Protokoll RIPng hingegen wurde ausschließlich für das Internet-Protokoll Version 6 (IPv6) entwickelt.

Da RIPng eine Weiterführung des RIPv2-Protokolls darstellt, ist es konzeptuell nicht völlig neu entwickelt worden. Bei der Entwicklung von RIPng wurden nur die nötigsten Änderungen in Hinblick auf IPv6 an RIPv2 vorgenommen. Des Weiteren wurden bewährte Eigenschaften der RIP-Familie beibehalten. Die IPv4-Adresse ist zu einer IPv6-Adresse erweitert und die 32-Bit IPv4-Subnetzmaske durch ein acht Bit großes IPv6-Präfixlängenfeld ersetzt worden. Die Adressfelder der Routing-Tabellen-Einträge (RTE) sind von 32-Bit auf 128-Bit vergrößert worden, um kompatibel zu den neuen IPv6-Adressen zu sein. Das Next-Hop-Feld aus dem RIPv2-Protokoll wurde entfernt. Die Funktionalität wird durch die neuen Extension-Header weiterhin unterstützt. Außerdem wurde das Authentifikationsfeld von RIPv2 entfernt, da IPv6 diesen Mechanismus standardmäßig beinhaltet. Das Route-Tag-Feld sowie der maximale Metrikwert von 15 sind geblieben [Mal97]. Außerdem werden die Routing-Datenpakete durch das RIPng-Protokoll nicht mehr in ihrer Größe beschränkt. Dies be-

dingt, dass weniger Datenpakete zum Übermitteln einer RIPng-Nachricht benötigt werden und somit ein Performancevorteil bei dem Versand der Datenpakete entsteht.

RIPng kommt vor allem in Bereichen zum Einsatz, in denen RIPv2 mit IPv4 angewendet wird und nun eine Umstellung auf das IPv6-Protokoll erfolgen soll. Mit RIPng wird nicht beabsichtigt, einen Ersatz für das OSPFng zu liefern, da RIPng wie auch RIPv2 durch verschiedene Beschränkungen nicht die Funktionalität für komplexere Netzwerke erbringen können [Mal97].

Das Protokoll basiert auf dem Distanz-Vektor-Algorithmus (Abschnitt 2.2.1). Durch den Einsatz des Distanz-Vektor-Algorithmus kann die kürzeste Strecke zu einem Ziel innerhalb eines IPv6-Netzwerkes berechnet werden. Da es sich bei dem RIPng-Protokoll um ein IGP (Interior-Gateway-Protocol) handelt, wird dieses Protokoll nur intern in kleinen bis mittelgroßen Netzwerken verwendet.

RIPng wird auf den einzelnen Routern implementiert. Dabei benötigen die Router Zugriff auf die Informationen anderer Router im Netzwerk, vor allem auf deren Metrik, um die kürzesten Strecken berechnen und das Protokoll erfolgreich ausführen zu können. RIPng läuft als einzelner Prozess auf einem Router.

Damit möglichst vollständige Routing-Informationen zur Berechnung der kürzesten Routen und dem Ausführen des Protokolls zu Verfügung stehen, sollte jeder Router innerhalb des Netzwerkes das RIPng-Protokoll implementieren. Falls mehrere verschiedene IGP-Protokolle innerhalb eines Netzwerkes verwendet werden, muss mindestens ein Router die Informationen zwischen den Protokollen übersetzen können, um ein Zusammenspiel der verschiedenen Routing-Protokolle zu gewährleisten.

Im Verlauf dieses Kapitels wird das Routing-Protokoll RIPng explizit analysiert und dokumentiert. Dabei wird das verwendete Nachrichtenformat, mögliche Einschränkungen sowie die Adressierung und das Routing von RIPng erläutert. Des Weiteren wird auf die Next-Hop-RTE sowie die Sicherheitsmaßnahmen und die Verarbeitung der Datenpakete eingegangen.

Die Informationen zu diesem Kapitel wurden [G.M97] entnommen.

3.1 Einschränkungen

RIPng und die früheren Versionen von RIP sind ursprünglich als Interior-Gateway-Protokolle entwickelt worden. Somit kommen diese Protokolle bei kleinen bis mittleren autonomen Systemen zum Einsatz, da sie für größere AS aus folgenden Gründen eher ungeeignet sind:

Eine Begrenzung der maximalen Distanz einer gültigen Route bei RIPng wird durch den Metrikwert 15 gegeben. Standardmäßig setzen die Router einer Route die Metrik um jeweils den Wert eins hoch. Daraus folgt, dass im besten Fall genau 15 Hops/Router passiert werden können. Jedoch lässt das RIPng-Protokoll ebenfalls eine manuelle Konfiguration der Metrik für die jeweiligen Router zu. Dadurch kann die Anzahl der zu passierenden Router stärker beschränkt werden, da den einzelnen Routen eine höhere Metrik zugewiesen wird. Routen mit einer Metrik von 16 gelten als unerreichbar.

Eine weitere Einschränkung stellen Routing-Schleifen dar. Zwar bietet RIPng die Möglichkeit, Routing-Schleifen in moderaten Netzwerken aufzudecken, jedoch sind damit hohe Konvergenzzeiten verbunden. Bei dem Ausfall eines Routers werden diese Informationen durch die Nachbarrouter verteilt. Dabei kann es unter bestimmten Umständen vorkommen, dass die Metrik wegen der Unerreichbarkeit des Zieles nach und nach um den Wert eins auf den Nachbarroutern erhöht wird, bis der Grenzwert 16 erreicht wird. Dieses Problem wird das Counting-to-Infinity-Problem (Abschnitt 2.2.1.1) genannt. Durch die Beschränkung der Metrik auf den Wert 15 wird bei Überschreitung dieser Metrikgrenze die Löschung der Route veranlasst. Die Löschung einer Route ist allerdings abhängig von der Verbreitungsgeschwindigkeit der Informationen innerhalb des Netzwerkes. Die Informationen werden durch Update-Nachrichten oder Triggered-Updates weitergeleitet. RIPng und das CTI-Problem werden anhand des Testszenarios in Abschnitt 4 genauer beschrieben.

Da durch die Metrik in den meisten Fällen keine Bewertung der Leistung der Route vorgenommen werden kann, stellt dies eine weitere Einschränkung dar. Es werden feste Metriken verwendet (im Normalfall sind die Routen mit dem Metrikwert eins versehen), um alternative Routen miteinander vergleichen zu können. Dabei werden Angaben über messbare Parameter wie die Zuverlässigkeit, die Verzögerung, die Last oder die Bandbreite einer Route nicht berücksichtigt. Es besteht somit keine Möglichkeit, die Routen auf Grund ihrer Echtzeitparameter auszuwählen.

3.2 Routing und Adressierung

Da das RIPng-Protokoll auf dem IPv6-Protokoll aufbaut, muss keine Unterscheidung zwischen Netzwerk-, Subnetzwerk- und Host-Routern vorgenommen werden. Diese sind durch ihre eindeutige IPv6-Adresse (Abschnitt 2.1) gekennzeichnet.

Des Weiteren besitzt jeder Router eine eigene Routing-Tabelle. Innerhalb dieser Routing-Tabelle wird je ein Eintrag für ein erreichbares Netzwerk mit weiteren Informationen angelegt, um die Datenpakete weiterleiten zu können. Ein Eintrag innerhalb der Routing-Tabelle besteht aus folgenden Informationen:

Es erfolgt die Angabe des IPv6-Netzwerkes und der Präfixlänge. Mithilfe der Präfixlänge wird das Netzwerk bestimmt. Weitere Angaben innerhalb eines Eintrags sind die Next-Hop-Adresse, welche den nächsten Router zum Netzwerk angibt, sowie die Metrik zur Angabe der Kosten für die Erreichung des Zielnetzwerkes. Die Metrik darf im RIPng-Protokoll einen Wert zwischen eins und einschließlich 15 annehmen. Routen, die mit dem Integer-Wert 16 gekennzeichnet sind, gelten als unerreichbar. Des Weiteren beinhaltet ein Routing-Eintrag verschiedene, mit der Route assoziierte Timer (Abschnitt 3.5) und ein Flag-Feld. Durch das Flag-Feld (auch Route-Change-Flag genannt) werden Änderungen an der Route dargestellt.

Die Einträge der Routing-Tabellen können entweder durch dynami-

sche Routing-Protokolle wie RIPng erzeugt oder manuell durch den Administrator ergänzt werden. Ein großer Vorteil bei dem manuellen Hinzufügen der Routen ist, dass die Routen durch die Angabe der Metrik indirekt durch den Administrator hinsichtlich ihrer Real-Time-Werte zur Datenübertragung bewertet werden können. Diese Unterscheidungen werden durch RIPng bei der automatischen Verwaltung der Routen nicht beachtet. Weitere Informationen zum statischen und dynamischen Routing sind im Grundlagenabschnitt 2.1.2 zu finden. Jedes Netzwerk innerhalb des AS hat einen IPv6-Zieladressenpräfix sowie eine bestimmte Präfixlänge. Durch diese Angaben können die einzelnen Netzwerke anhand der Präfixe unterschieden werden. Dies dient der Weiterleitung der Datenpakete an das richtige Netzwerk.

Eine besondere Art von Routen stellen die Default-Routen dar. Diese werden bei dem RIPng-Protokoll mit dem IPv6-Präfix 0::0 angegeben. Die Präfixlänge wird mit dem Wert null initialisiert. Ein Router verwendet eine Default-Route, falls es für ihn nicht möglich ist, alle Netzwerke in der RIPng-Update-Nachricht aufzuzählen oder falls bestimmte Netzwerke, an die Datenpakete gesendet werden sollen, nicht in der Routing-Tabelle aufgeführt sind. In diesem Fall verwendet der Router seine Default-Route, in der Hoffnung, dass der in der Default-Route eingetragene Router eine Route zum Zielnetzwerk kennt. Der Administrator bestimmt, welche Router als Default-Router deklariert werden. Diese Router erzeugen und kündigen die Default-Routen-Einträge an. Außerdem können die Default-Routen durch den Administrator mit der passenden Metrik versehen werden. Dadurch ergibt sich die Möglichkeit, dass die Router intern eine Reihenfolge der Default-Routen auf Grund der Metrik anlegen können. Die Einträge der Default-Routen werden wie normale RTE behandelt.

3.3 Nachrichtenformat

RIPng basiert auf dem verbindungslosen UDP-Protokoll unter der Verwendung von Port 521, dem RIPng-Port. Im Vergleich zu dem TCP-Protokoll werden keine festen Verbindungen aufgebaut (*3-Way Handshake*). Dies bedingt, dass bei einer fehlerhaften Übertragung eines Datenpaketes diese Datenpaket nicht automatisch erneut gesendet wird. Diese Verantwortung bleibt der Applikation überlassen.

Alle Nachrichten, die an den RIPng-Port geleitet werden, werden durch den RIPng-Prozess empfangen und verarbeitet. Des Weiteren werden alle RIPng-Nachrichten, wie zum Beispiel Update-Nachrichten, von diesem Port aus versendet. Bei unaufgeforderten Update-Nachrichten sind der Absender- und Empfänger-Port gleich dem RIPng-Port. Auf eine Request-Nachricht gesendete Response-Nachrichten werden an den Port gesendet, von welchem die Request-Nachricht kam. Spezielle Request-Nachrichten können auch von anderen Ports aus zur Verarbeitung an den RIPng-Port gesendet werden.

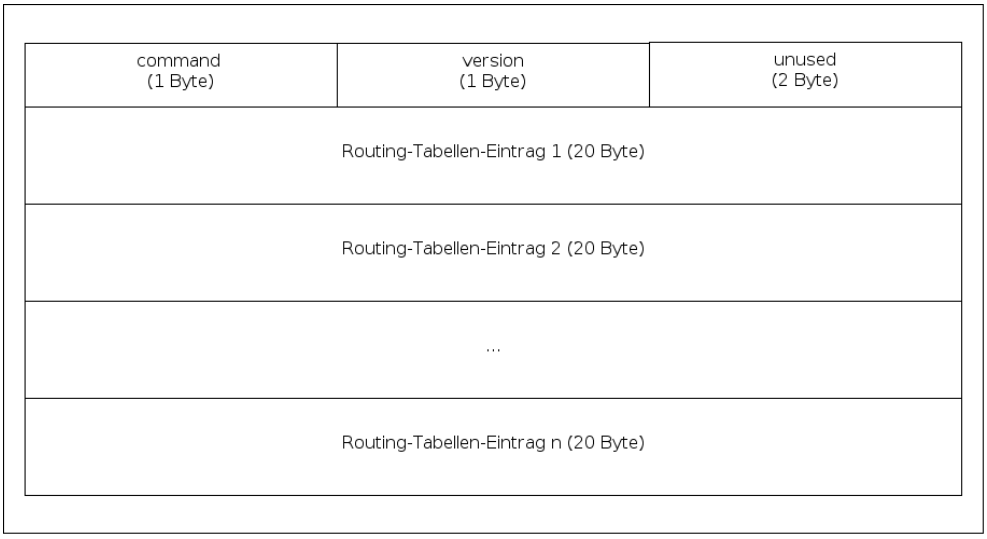

Abbildung 3.1: Aufbau eines RIPng-Nachrichtenpaketes [G.M97]

In Abbildung 3.1 wird der Aufbau eines RIPng-Datenpaketes dargestellt. Ein Paket setzt sich aus dem RIPng-Header sowie einer begrenzten

Anzahl von Routing-Tabellen-Einträgen (Route-Table-Entry *RTE*) zusammen. Das command-Feld, das version-Feld sowie das unused-Feld stellen den RIPng-Header dar. Der Aufbau eines einzelnen RTEs wird in Abbildung 3.2 genauer dargestellt und anschließend erläutert. Die Größen der jeweiligen Felder werden in Oktetten angegeben und nach dem *Big-Endian-Format* ausgewertet.

Im Folgenden werden die Bedeutungen der einzelnen Felder des RIPng-Headers erläutert. Die Feldgrößen werden in Klammern als Bytewerte dargestellt.

command (1 Byte)

Mithilfe dieses Feldes wird der Zweck des Datenpaketes angegeben. Dabei stehen in Version 1 des RIPng-Protokolls für das command-Feld die Werte *request* und *response* zu Verfügung. Durch einen request-Wert (1) wird dem antwortenden Router signalisiert, die gesamte beziehungsweise die partielle Routing-Tabelle zu senden. Die Verwendung des response-Wertes (2) in einer RIPng-Nachricht zeigt an, dass es sich um eine Nachricht handelt, welche die gesamte beziehungsweise Teile der Routing-Tabelle enthält. Response-Nachrichten können durch vorausgegangene request-Nachrichten ausgelöst oder durch ein unaufgefordertes Routing-Update versendet werden.

version (1 Byte)

Das version-Feld gibt die verwendete Version des RIPng-Protokolls an. Standardmäßig wird dieses Feld immer mit dem Wert eins initialisiert.

must be zero (2 Byte)

Dieses Feld wird im aktuellen RIPng-Protokoll nicht verwendet. Es kann eventuellen Erweiterungen in der Zukunft dienen.

Die maximale Datengröße eines RIPng-Datenpaketes und die damit verbundene maximale Anzahl von RTE innerhalb eines Datenpaketes werden durch die Maximum-Transmission-Unit (MTU) und das verwendete physikalische Übertragungsmedium beschränkt [Hag09].

Die Abbildung 3.2 zeigt den Aufbau eines Routing-Tabellen-Eintrages, wie er in den RIPng-Nachrichten vorzufinden ist. Jeder RTE-Eintrag hat eine feste Größe von 20 Byte.

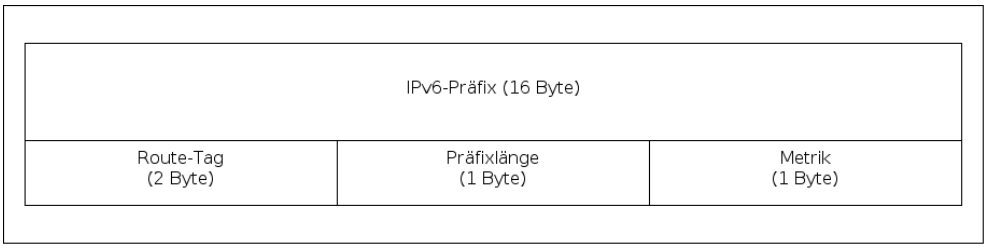

Abbildung 3.2: Aufbau eines Routing-Tabellen-Eintrags (RTE) [G.M97]

Ein Routing-Tabellen-Eintrag setzt sich aus dem IPv6-Präfix, dem Route-Tag, der Präfixlänge sowie der Metrik zusammen. Diese Felder werden im Folgenden erklärt.

IPv6-Präfix (16 Byte)

Das IPv6-Präfix-Feld ist gewöhnlich 128-Bit lang und gibt die Zieladresse beziehungsweise die IP-Adresse des nächsten Routers an. Das Format der IPv6-Adressen wird in Abschnitt 2.1.3 explizit erläutert.

Route-Tag (2 Byte)

Durch das Route-Tag-Feld können zusätzliche Informationen über die Route vermittelt werden. Somit können interne RIPng-Routen von Externen unterschieden werden. Interne Routen sind dabei die Routen, die innerhalb des Netzwerkes im RIPng-Bereich vorzufinden sind. Externe Routen sind Routen, welche der Router von anderen Routing-Protokollen, zum Beispiel BGP (*Border-Gateway-Protokoll*), gelernt hat. Deshalb stellt das Route-Tag ein Transitfeld für Informationen aus externen Routen dar. Es bietet dem Router die Möglichkeit, Routen von externen Routing-Protokollen zu importieren und das Tag gemäß der Anweisungen der externen Routing-Protokolle zu setzen.

Präfixlänge (1 Byte)

Die Präfixlänge gibt die Anzahl der signifikanten Bits innerhalb des IPv6-Präfixes an, um das IPv6-Netzwerk zu bestimmen. Die Auswertung der Bits erfolgt nach dem *Big-Endian-Format*.

Metrik (1 Byte)

In dem Metrik-Feld werden die Kosten des absendenden Routers für die Erreichung eines Zielnetzwerkes für die Route gespeichert. Dabei sind nur Werte zwischen eins und einschließlich 15 gültig. Routen, dessen Metrik einen größeren Wert aufweisen, werden als unerreichbar gekennzeichnet und aus der Routing-Tabelle gelöscht (Abschnitt 3.6).

Durch die einzelnen RTE können die verschiedenen Routen repräsentiert werden, die der Router in seiner Routing-Tabelle speichert. Die Route eines Datenpaketes wird mithilfe der Angaben von mehreren RTE und dessen IPv6-Präfixadressen bestimmt.

3.4 Next-Hop-Eintrag

RIPng bietet mit den Next-Hop-RTE die Möglichkeit, den nächsten Router auf dem Weg zum Ziel durch dessen IPv6-Adresse zu kennzeichnen. Somit kann die komplette Route der Datenpakete auf dem Weg zum Ziel bestimmt werden. Die Weiterleitung von RIPng ist der von RIPv2 ähnlich, da dort auch jedes RTE ein Next-Hop-Feld besitzt.

Da RIPng auf dem IPv6-Protokoll basiert und somit die IP-Adressen 128-Bit lang sind, würde ein jeweiliger Eintrag des Next-Hop-Feldes in jedem RTE die Größe des Datenpaketes nahezu verdoppeln und das Datenpaket somit unnötig vergrößern. Aus diesem Grund wird das Next-Hop-Feld bei dem RIPng-Protokoll durch einen speziellen RTE vorangestellt. Die Abbildung 3.3 stellt den Aufbau eines Next-Hop-RTEs dar.
Im Anschluss an die Abbildung erfolgt die Erklärung der einzelnen Felder.

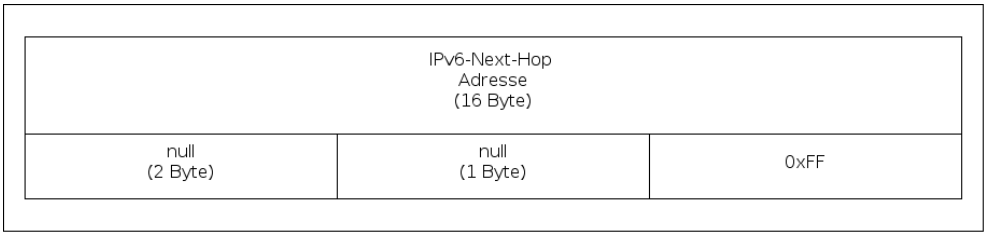

Abbildung 3.3: Aufbau eines Next-Hop-RTEs [Hag09]

Der Next-Hop-RTE enthält zur Identifizierung im Metrik-Feld den Wert 0xFF. Durch diesen Wert kann der RTE eindeutig als ein Next-Hop-RTE identifiziert werden und das IPv6-Präfix wird nicht mehr als Routenpräfix anerkannt. Die IP-Adresse im IPv6-Präfixfeld (IPv6-Next-Hop-Adresse) gibt den nächsten Hop beziehungsweise den nächsten Router an. Die Felder Route-Tag und Präfixlänge müssen den Wert null enthalten. Die IP-Adresse `0::0` im Präfix-Feld eines Next-Hop-RTE definiert, dass die Next-Hop-Adresse der Absender der RIPng-Nachricht selbst ist, da keine spezifische Next-Hop IPv6-Adresse gesetzt wurde. Falls alle RTE den Absender der RIPng-Nachricht als Next-Hop-Adresse verwenden, kann der Next-Hop-RTE komplett weggelassen werden.

Ebenfalls kann der Fall eintreten, dass eine Next-Hop-Adresse explizit eingetragen werden soll. Dies kommt zum Einsatz, wenn unnötige Routingwege während des Transportes der RIPng-Nachrichten vermieden werden sollen. Zugleich muss eine Next-Hop-Adresse immer eine Link-Local-Adresse sein, welche an ihrem Präfix zu erkennen ist. Falls es sich bei der Next-Hop-Adresse nicht um eine Link-Local-Adresse handelt beziehungsweise in der RIPng-Nachricht kein Next-Hop-RTE definiert ist, erfolgt eine Behandlung der Next-Hop-Adresse mit dem Präfixwert `0::0`.

Um die speziellen Next-Hop-RTE auf mehrere Routing-Tabellen-Einträge anwenden zu können, stehen die Next-Hop-RTE immer zu Anfang einer RTE-Sequenz. Dadurch wird ein Next-Hop-RTE auf alle folgenden RTE des gesamten Datenpaketes angewendet beziehungsweise bis ein neuer spezieller Next-Hop-RTE erscheint. Ein Beispiel für solche RTE-Sequenzen und die passenden Next-Hop-RTE wird in Abbildung 3.4 dargestellt.

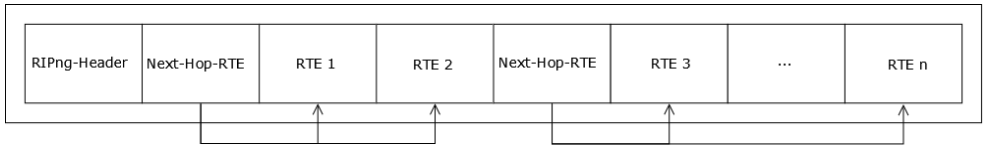

Abbildung 3.4: Next-Hop-RTE in der Anwendung [Hag09]

Nach dem RIPng-Header folgt ein erster Next-Hop-RTE. Dieser enthält die IPv6-Adresse des nächsten Routers für die Routing-Tabellen-Einträge RTE1 und RTE2. Der folgende Next-Hop-RTE gibt eine weitere IPv6-Adresse für die noch verbleibenden RTE an.

3.5 Timer

Das RIPng-Protokoll implementiert verschiedene Arten von Timer. Diese dienen der Kontrolle von Änderungen an den Routen in den Routing-Tabellen. Es wird zwischen drei verschiedenen Timer unterschieden, die im Folgenden näher erläutert werden

Der Update-Timer veranlasst die Router alle 30 Sekunden dazu, unaufgefordert ihre komplette Routing-Tabelle in einer Response-Nachricht über alle vorhandenen Interfaces an alle Nachbarrouter zu senden. Der Timer wird für jedes Interface eigenständig ausgeführt. Routen, die unter die Split-Horizon-Maßnahme fallen, werden nicht mitgesendet. Des Weiteren kann es vorkommen, dass für die Übertragung der Routing-Tabelle mithilfe der Response-Nachricht mehrere RIPng-Pakete benötigt werden. Dies wird durch die Einstellungen der MTU und des physikalischen Übertragungsmediums (Abschnitt 3.3) bestimmt. Falls eine hohe Anzahl an Routern in einem einzigen Netzwerk vorhanden sind, besteht die Gefahr, dass die Router sich zeitlich untereinander synchronisieren. Dadurch werden Update-Nachrichten der verschiedenen Router zur selben Zeit versendet. Solch eine Synchronisation kann durch die Beeinflussung der Systembelastung auf den Zeitgeber entstehen. Dieser Zustand ist unerwünscht, da dies eine unnötige Kollision der Datenpakete mit sich führen kann.

Um dies zu verhindern, sind zwei Schutzmechanismen entwickelt wor-

den. Mindestens eine dieser Schutzmaßnahmen muss bei der Implementation berücksichtigt werden, um eine Synchronisation der Router zu vermeiden. Der erste Schutzmechanismus besagt, dass die regulären Updates von einem Zeitgeber ausgelöst werden, der nicht von der Systembelastung oder von der benötigten Zeit des vorangehenden Update-Timers abhängig ist. Der zweite Schutzmechanismus bietet durch eine kurze Random-Time, welche den Update-Timer um eine gewisse Zeit versetzt, einen Schutz vor der Synchronisierung. Die zufällige Zeit berechnet sich aus der Update-Periode multipliziert mit dem Wert 0,5.

Des Weiteren besitzt jede Route einen Timeout- sowie Garbage-Collection-Timer. Diese beiden Timer dienen der Löschung einer nicht mehr erreichbaren Route und werden unter gewissen Umständen gestartet.

Bei dem Timeout-Timer wird eine Zeitperiode in Sekunden für den Ablauf einer RIPng-Route angegeben. Die Zeitperiode beträgt standardmäßig 180 Sekunden. Falls bei einer Route diese Zeit abgelaufen ist, wird die entsprechende Route als nicht mehr verfügbar betrachtet und mit der Metrik 16 versehen. Bei jeder Aktualisierung einer Route durch eine Update-Nachricht oder ein Triggered-Update wird der Timeout-Timer wieder auf seinen Ausgangswert null zurückgesetzt und beginnt erneut zu inkrementieren. Der Lebenszyklus dieser Route auf dem Router beginnt von Neuem. Falls jedoch keine Aktualisierung innerhalb der Zeitperiode erfolgt und die Route als nicht mehr erreichbar deklariert ist, wird für diese Route der Garbage-Collection-Timer gestartet. Außerdem wird bei der Änderung einer Route auf den Metrikwert 16 das Change-Flag gesetzt, um eine sofortige Weiterleitung der Änderungen durch ein Triggered-Update (Abschnitt 3.6.1.2) zu veranlassen. Diese Maßnahme trägt zur Minderung der Konvergenzzeit innerhalb des Netzwerkes bei.

Der Garbage-Collection-Timer startet nach Ablauf des Timeout-Timers beziehungsweise wenn eine Route auf den Metrikwert 16 gesetzt wird. Der Timer wird für die zu löschende Route mit einem Zeitwert von 120 Sekunden initialisiert und beginnt rückwärts zu zählen. Falls die Metrik der Route noch nicht mit dem Wert 16 versehen wurde, wird dieser Wert nachträglich gesetzt. Der Metrikwert 16 ist Voraussetzung für eine spätere

Löschung der Route.

Solange der Garbage-Collection-Timer nicht abgelaufen ist, wird die Route in alle Update-Nachrichten mit einbezogen. Erfolgt eine Response-Nachricht in Form einer Update-Nachricht oder eines Triggered-Updates, so wird der Garbage-Collection-Timer von der Route entfernt und der Timeout-Timer auf den Wert null zurückgesetzt.

Eine Löschung der Route wird veranlasst, falls beim Ablauf der Zeit keine Aktualisierung der Route vorgenommen wird. Bei Löschung der Route kann es zu einer hohen Konvergenzzeit kommen. Bei einer periodischen Verteilung der Routing-Informationen der Router in einem Abstand von 30 Sekunden kann die Verteilung von Informationen innerhalb eines größeren Netzwerkes deswegen mehrere Minuten betragen.

Falls eine neue Route während eines aktiven Garbage-Collection-Timers zu einem Netzwerk hinzugefügt wird, ersetzt diese die zu löschende Route. In diesem Fall muss der Garbage-Collection-Timer neu gesetzt werden.

3.6 Verarbeitung der Datenpakete

In diesem Abschnitt wird die Verarbeitung der ein- und ausgehenden RIPng-Nachrichten beschrieben. Eine Nachricht kann sich aus mehreren RIPng-Datenpaketen zusammensetzen. Die Verarbeitung der Nachrichten ist abhängig von dem Wert im command-Feld (Abschnitt 3.3). Durch dieses Feld wird zwischen response- und request-Nachrichten unterschieden, die einer individuellen Verarbeitung bedürfen. Ein- und ausgehende RIPng-Datenpakete erreichen beziehungsweise verlassen einen Router über den RIPng-Port 521.

3.6.1 Response-Nachrichten

Mithilfe von Response-Nachrichten werden gesamte oder bestimmte Teile der Routing-Tabelle an die Nachbarrouter gesendet, um die eigenen Informationen über die Routen mit den Nachbarroutern zu teilen. Daraus

können mithilfe des Distanz-Vektor-Algorithmus die kürzesten Strecken zu den jeweiligen Zielen anhand der Metrik berechnet werden.

Die Erstellung einer Response-Nachricht für einen Router kann folgende Gründe haben:

Antwort auf eine spezielle Anfrage
: Falls eine Request-Nachricht an einen bestimmten Router gesendet wird, erstellt dieser daraufhin eine Response-Nachricht und sendet diese zurück. In diesem Fall wird die Response-Nachricht nur an die Unicast-Adresse des anfragenden Routers gesendet.

Reguläre Updates
: Die regulären Updates, die standardmäßig alle 30 Sekunden von jedem Router versendet werden, werden ebenfalls als Response-Nachrichten versendet. Dabei wird die komplette Routing-Tabelle an die Nachbarrouter gesendet, um eventuelle Änderungen innerhalb der Routing-Tabelle weiterzugeben.

Triggered-Update
: Falls es zu Änderungen an einer Route innerhalb der Routing-Tabelle kommt, werden diese Änderungen mithilfe von Triggered-Updates an die Nachbarrouter gesendet. Die Triggered-Updates liegen dabei in Form von Response-Nachrichten vor und werden an alle über die Schnittstellen erreichbaren Nachbarrouter gesendet. Die Response-Nachricht beinhaltet in diesem Fall nur die geänderten Einträge.

Bei Erhalt einer Response-Nachricht wird diese auf ihre Gültigkeit hin überprüft. Dies ist nötig, da die Response-Nachrichten Änderungen an den Routing-Tabellen der Router bedingen können. Die Response-Nachricht muss von einem RIPng-Port (521) versendet worden sein. Ist dies nicht der Fall, wird sie ignoriert und verworfen. Außerdem wird überprüft, ob es sich bei der IPv6-Absenderadresse um eine Link-Local-Adresse handelt und ob die Response-Nachricht nicht von dem Router selbst ist. Der Hop-Count muss auf den Wert 255 gesetzt sein. Dieser Wert garantiert, dass die Response-Nachricht keine anderen Router passiert hat. So-

bald die Gültigkeit der einkommenden Response-Nachricht positiv überprüft wurde, kann die Verarbeitung der RTE beginnen.

Diese Überprüfungen gelten jedoch nicht für Response-Nachrichten, welche auf Grund einer spezifischen Request-Nachricht gesendet worden sind. Der Hop-Count darf in diesem Fall einen kleineren Wert als 255 aufweisen und die IPv6-Absenderadresse kann eine Global- oder Unique-Local-Adresse sein.

Vor der Verarbeitung eines Routing-Tabellen-Eintrages werden diese ebenfalls einer Überprüfung unterzogen und auf ihre Gültigkeit hin überprüft. Dabei besteht die Basisüberprüfung eines einzelnen RTE aus folgenden Schritten:

Es erfolgt eine Überprüfung des IPv6-Zielpräfixes und der Präfixlänge. Dabei darf es sich bei dem IPv6-Zielpräfix nicht um eine Link-Local- oder eine Multicast-Adresse handeln. Die Präfixlänge muss einen Wert zwischen null und 128 enthalten. Des Weiteren wird der Wert im Metrik-Feld überprüft. Dieser muss zwischen eins und einschließlich 15 liegen. Falls eine dieser Überprüfungen fehlschlägt, wird der Eintrag ignoriert und es wird mit dem nächsten Routing-Tabellen-Eintrag fortgefahren. Die Fehler sollten aus Gründen der Sicherheit und der Fehlerbehebung mitgeloggt werden.

Sobald ein RTE die Gültigkeitsprüfung erfolgreich bestanden hat, überprüft der Router, ob schon eine Route für das IPv6-Zielpräfix vorhanden ist. Falls noch keine Route für diese Zieladresse existiert, wird die Route nach folgendem Verfahren der Routing-Tabelle des Routers hinzugefügt:

Zuerst setzt der Router in seiner Routing-Tabelle die IPv6-Zieladresse und die Präfixlänge in den neuen RTE. Danach wird die Metrik berechnet und übernommen sowie die Next-Hop-Adresse im RTE eingestellt. Diese gibt dem Router an, von welchem Router das Datenpaket gesendet wurde oder die Next-Hop-Adresse, welche durch einen Next-Hop-RTE spezifiziert wird. Außerdem wird der Timeout-Timer für die Route initialisiert. Falls der Garbage-Collection-Timer aktiv ist, wird dieser gestoppt. Zum Schluss wird noch der Wert im Flag-Feld auf eins gesetzt und ein Triggered-Update ausgelöst. Durch das Setzen des Flag-Feldes wird eine

Änderung an der Route innerhalb der Routing-Tabelle signalisiert. Das Triggered-Update sendet die Änderungen sofort an seine Nachbarrouter weiter, um die neuen Informationen im Netzwerk möglichst schnell zu verbreiten.

Existiert jedoch eine Route für die IPv6-Zieladresse, wird die Next-Hop-Adresse mit der Adresse des Routers, von dem die Response-Nachricht gesendet wurde, verglichen. Es erfolgt eine Reinitialisierung des Timeout-Timers, falls die Response-Nachricht von dem gleichen Router wie die existierende Route innerhalb der Routing-Tabelle stammt. Des Weiteren werden die Metriken verglichen. Wenn die Metriken unterschiedlich sind oder die neue Metrik kleiner als die Metrik aus der Routing-Tabelle ist und das Datenpaket von dem gleichen Router wie die existierende Route stammt, wird die Route in der Routing-Tabelle durch die Route aus der Response-Nachricht ersetzt. Die Metrik in der Routing-Tabelle wird durch die neue, mitgesendete Metrik ersetzt und gegebenenfalls die Next-Hop-Adresse angepasst. Außerdem wird das Route-Change-Flag-Feld gesetzt und ein Triggered-Update ausgelöst.

Die Metriken werden nach folgender Funktion berechnet:

$$Metrik = MIN(Aktuelle\ Metrik + Kosten\ der\ Route,\ Infinity)$$

Falls jedoch die neue, mitgesendete Metrik außerhalb des gültigen Wertebereiches liegt und somit die Route unerreichbar ist, wird der Löschvorgang der Route innerhalb der Routing-Tabelle gestartet. Der Löschvorgang wird nur bei dem erstmaligen Setzen der Metrik auf den Wert 16 gestartet. Falls die Metrik schon zuvor einen Wert außerhalb des gültigen Bereiches angenommen hat, erfolgt kein neuer Löschvorgang.

Ein weiterer Fall ist die Gleichheit beider Metriken. Dabei wird aus Gründen der Einfachheit nichts unternommen, außer eine Reinitialisierung des Timeout-Timers. Es wird jedoch empfohlen, die im Folgenden erläuterte optionale Heuristik bei Gleichheit zweier Metriken anzuwenden:

Im Falle der Gleichheit beider Metriken ist es sinnlos, die existierende Route durch die neue Route zu ersetzen, da dies mit einer hohen Anzahl

von unnützigen Triggered-Updates verbunden ist. Falls die existierende Route jedoch Anzeichen von einem Ablaufen des Timeout-Timers aufweist, ist es besser, eine alternative, gleichwertige Route zu wählen. Deswegen werden bei der Gleichheit zweier Metriken die Timeout-Timer der Routen untersucht. Ist der Timer der existierenden Route zeitlich schon unter der Hälfte der abgelaufenen Zeit, wird zu der neuen Route gewechselt. Die alte Route wird gelöscht beziehungsweise mit den Einstellungen der neuen Route überschrieben.

Um eine Response-Nachricht an alle Nachbarrouter zu senden, wird die IPv6-Multicast-Gruppe `FF02::9` verwendet. Es wird eine Response-Nachricht für alle Nachbarrouter vorbereitet und versendet. Jedoch kann es vorkommen, dass nicht alle Nachbarrouter erreicht werden. Dies kann der Fall sein, wenn es sich nicht um ein Broadcast-Netzwerk handelt oder die benachbarten Router stumm sind, also nicht angesprochen werden können. Hier muss der Administrator eine Liste der direkten Nachbarrouter erstellen, um Datenpakete an diese Router senden zu können. Somit wird erreicht, dass die benachbarten Router die Response-Nachricht durch einen Unicast erhalten können.

Des Weiteren wird zwischen aufgeforderten und unaufgeforderten Response-Nachrichten unterschieden. Bei den durch Triggered-Updates oder periodischen Updates versendeten Response-Nachrichten handelt es sich um unaufgeforderte Nachrichten. Diese Nachrichten werden durch die IPv6-Multicast-Gruppe `FF02::9` an alle benachbarten Router gesendet. Als Absenderadresse wird in den Response-Nachrichten die Link-Local-Adresse der Absenderschnittstelle eingetragen. Der Quell- sowie Zielport ist der RIPng-Port.

Bei den unaufgeforderten Response-Nachrichten durch ein Triggered-Update werden nur die geänderten Routen versendet. Diese werden durch das Setzen des Route-Change-Flags gekennzeichnet. Die Response-Nachricht wird über alle Schnittstellen unter der Berücksichtigung des Split-Horizon-Verfahrens an die benachbarten Router gesendet [Hag09].

Bei den periodischen Updates werden die Response-Nachrichten nach dem Ablauf des Update-Timers versendet. Die ebenfalls unaufgeforder-

ten Response-Nachrichten, welche in diesem Fall die komplette Routing-Tabelle enthalten, werden nur über die entsprechende Schnittstelle und unter Berücksichtigung des Split-Horizon-Verfahrens versendet [Hag09].

Die aufgeforderten Response-Nachrichten werden nur verwendet, um eine Antwort auf eine Request-Nachricht zu senden. Dabei dient die Absenderadresse der Request-Nachricht als Zieladresse. Die Absenderadresse wird durch eine Link-Local- oder Global-Local-IPv6-Adresse angegeben. Als Zielport muss der RIPng-Port, als Absender der entsprechende UDP-Port angegeben werden.

Die Unterschiede zwischen aufgeforderten und unaufgeforderten Response-Nachrichten werden in Abbildung 3.5 noch einmal kurz zusammengefasst.

Nachrichtentyp	Absenderadresse	Zieladresse	Absender-Port	Ziel-Port
Unaufgeforderte Response-Nachricht (periodisches/ Triggered-Update)	Link-Local-Adresse des Absendeinterfaces	FF02::9 (Multicast-Adresse für RIPng)	RIPng-Port 521	RIPng-Port 521
Aufgeforderte Response-Nachricht (Antwort auf Request-Nachricht)	Link-Local-Adresse (generelles Request) Global-/ Unique-Local-Adresse (spezifischer Request)	Absenderadresse der entsprechenden Request-Nachricht	RIPng-Port 521	Absender UDP-Port der entsprechenden Request-Nachricht

Abbildung 3.5: Response-Typen im Überblick [Hag09]

3.6.1.1 Erstellung von Response-Nachrichten

Bei der Erstellung einer Response-Nachricht muss die IPv6-Absenderadresse eine Link-Local-Adresse aus den möglichen Adressen der Schnittstellen des sendenden Routers sein. Eine Ausnahme stellt eine Response-Nachricht auf eine Unicast-Request-Nachricht von einem anderen Port als dem RIPng-Port dar. In diesem Fall muss die IPv6-Absenderadresse eine global gültige Adresse sein.

Die Verwendung einer Link-Local-Adresse ist Voraussetzung für ein erfolgreiches Routing, da die Absenderadresse aus der Response-Nachricht als Next-Hop-Adresse in die Routing-Tabelle des Routers eingetragen wird.

Falls es sich dabei um eine fehlerhafte Absenderadresse handelt, können andere Router möglicherweise die Pakete nicht korrekt weiterleiten.

In manchen Fällen besitzen die Router auch mehrere IPv6-Adressen für eine physikalische Schnittstelle. Damit können über ein physikalisches Medium mehrere verschiedene Netzwerke angesprochen werden. Somit kann ein Router mehrere Link-Local-Adressen für ein physikalisches Medium besitzen. In diesem Fall muss der Router nur eine einzige Response-Nachricht erstellen, welche als IPv6-Absenderadresse die gewünschte Link-Local-Adresse für eine bestimmte Schnittstelle besitzt. Die Auswahl der Link-Local-Adresse sollte sich nur ändern, wenn die aktuelle Link-Local-Adresse nicht mehr erreichbar ist. Diese Vereinbarung ist nötig, da Router, welche die Response-Nachricht empfangen, mithilfe der IPv6-Absenderadresse den Absender identifizieren.

Sendet ein Router mehrere Datenpakete mit unterschiedlichen IPv6-Absenderadresse an einen benachbarten Router , führt dies zu einem unerwünschten Zustand. Der Nachbarrouter geht davon aus, dass die Pakete von unterschiedlichen Routern stammen und kann somit keinen eindeutigen Zustand herstellen.

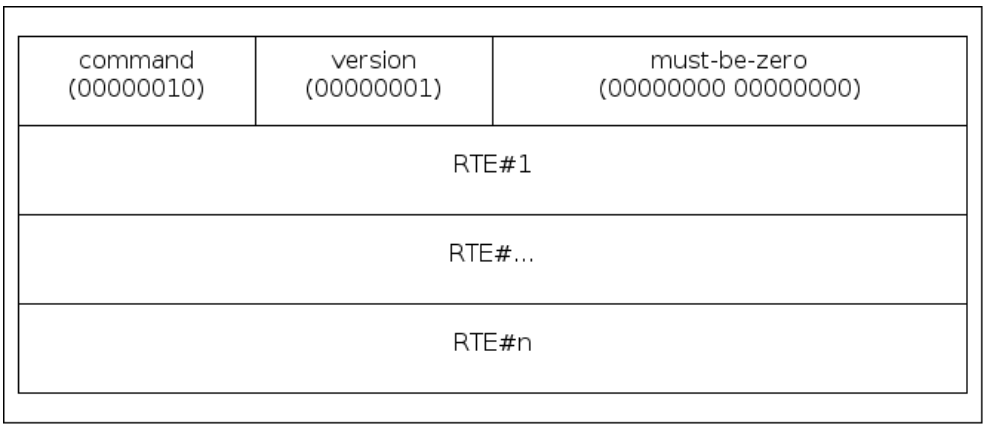

Abbildung 3.6: Response-Nachricht mit Routing-Tabellen-Einträgen

Nachdem die genannten Entscheidungen bezüglich der IPv6-Absenderadresse getroffen sind, wird bei der Erstellung einer Response-Nachricht folgendermaßen fortgefahren. Abbildung 3.6 stellt eine konfigurierte

Response-Nachricht dar. Dabei sind die expliziten Werte als Binärzahlen eingetragen.

Die Versionsnummer des aktuellen RIPng-Protokolls wird im Version-Feld des Datenpakets gesetzt, welche derzeit noch Version eins ist. Im Command-Feld wird der Response-Wert zwei eingetragen, um die Nachricht als Response-Nachricht zu deklarieren. Das Unused-Feld (auch must-be-zero-Feld genannt) wird mit dem Wert null initialisiert. Danach werden die Routing-Tabellen-Einträge gefüllt.

Bei der Übertragung der Response-Nachricht muss die maximal mögliche Größe beachtet werden. Diese wird durch die MTU und das physikalische Medium beschränkt. Gegebenenfalls muss die Response-Nachricht in mehrere Nachrichten aufgeteilt werden, um diese nacheinander zu versenden. Die Anzahl der Routen, welche eine Response-Nachricht beinhaltet, wird durch folgende Formel berechnet:

$$Anzahl\ der\ Routen\ in\ einer\ Response - Nachricht = (Laenge\ der\ Routing\ Daten / Groesse\ eines\ RTE)$$

Bei der Erstellung der Response-Nachricht und dem damit verbundenen Einfügen der RTE muss jede Route in der Routing-Tabelle einzeln untersucht werden. Routen, welche auf Link-Local-Adressen verweisen, dürfen niemals als ein RTE eingebaut werden. Response-Nachrichten, welche auf Grund eines Triggered-Updates erstellt werden, müssen nur die Routen berücksichtigen, bei denen das Route-Change-Flag-Feld gesetzt wurde.

Routen, die durch das Split-Horizon-Verfahren nicht mit einbezogen werden, müssen ausgelassen werden. Sollte die Route jedoch gültig sein, wird das IPv6-Zielpräfix, die Präfixlänge sowie die Metrik im RTE festgeschrieben. Des Weiteren wird das Route-Tag gesetzt. Routen, die mit einem unerreichbaren Metrikwert initialisiert sind, werden ebenfalls in den Response-Nachrichten berücksichtigt und mitgesendet bis die Route auf Grund des Garbage-Collection-Timers endgültig gelöscht wird.

3.6.1.2 Triggered-Updates

Durch Triggered-Updates werden Änderungen in den Routing-Tabellen direkt an die Nachbarrouter weitergesendet. Dies dient einer schnelleren Verbreitung von Änderungen und der Minderung der Konvergenzzeit innerhalb eines Netzwerkes. Mithilfe der Abbildung 3.7 und einem kurzen Beispiel werden die Triggered-Updates genauer erläutert.

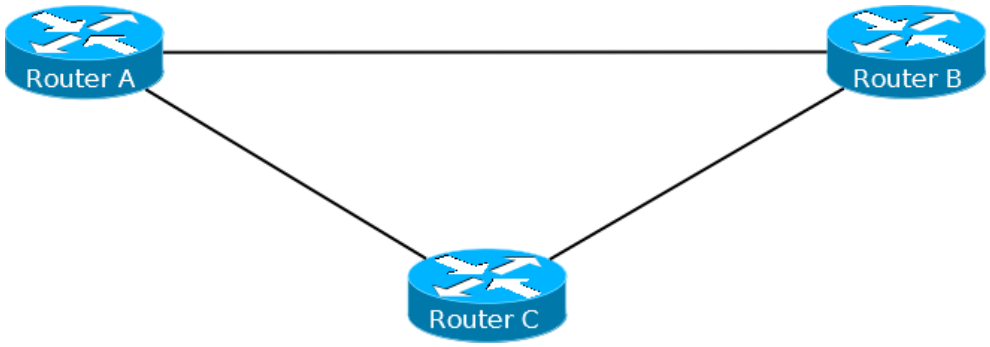

Abbildung 3.7: Triggered-Updates zwischen drei Routern

Falls Router A durch eine ankommende Response-Nachricht von Router B oder Router C Änderungen in seiner Routing-Tabelle vornimmt, wie das Setzen einer neuen Metrik für eine bestimmte Route, sendet dieser daraufhin automatisch seine geänderte komplette Routing-Tabelle an alle seine Nachbarrouter unter Beachtung der Split-Horizon-Maßnahme. Somit werden die Nachbarrouter sofort über die Änderungen informiert und können ihre Routing-Tabellen gegebenenfalls dementsprechend anpassen.

Triggered-Updates können jedoch das Netzwerk stark belasten, da kleinere Netzwerke mit beschränkten Netzwerkkapazitäten und vielen Routern massiv überlastet werden können. Ein kurzes Beispiel anhand der Abbildung 3.7 soll dies verdeutlichen.

Falls Router A an Router B und Router C ein Triggered-Update sendet, werden diese Router unter Umständen nach Vergleich der neuen Metriken Änderungen an den Routen innerhalb ihrer Routing-Tabellen vornehmen. Somit würden die Router B und C wieder, unmittelbar nach Erhalt des Triggered-Updates von Router A, ihre eigenen Triggered-Updates absen-

den. Je größer die Netzwerke sind, desto größer ist die Belastung durch die Triggered-Update-Maßnahme.

Um diese großen Belastungen zu verhindern, können Beschränkungen hinsichtlich der Häufigkeit der Triggered-Updates durch zeitlich definierte Abstände von dem Administrator definiert werden. Nach jedem gesendeten Triggered-Update wird ein Timer mit einer Zufallszahl zwischen einer und fünf Sekunden für jede Schnittstelle eines Routers initialisiert. Diese zufällige Zeit muss abgelaufen sein, bis das nächste Triggered-Update von dem Router aus versendet werden darf. Weitere Änderungen in der Routing-Tabelle innerhalb der Timerzeit lösen kein erneutes Triggered-Update aus. Danach wird der Timer mit einem neuen Zufallswert zwischen einer und fünf Sekunden initialisiert. Falls ein reguläres Update zu dieser Zeit stattfindet, wird ein Triggered-Update unterdrückt und nicht ausgeführt.

Des Weiteren senden Triggered-Updates nicht die komplette Routing-Tabelle. Eine Übermittlung der geänderten RTE reicht aus. Deswegen werden die Response-Nachrichten so generiert, dass nur die RTE versendet werden, bei denen das Route-Change-Flag gesetzt wurde. Der Administrator kann weitere RTE angeben, welche ebenfalls übermittelt werden sollen. Aus Performancegründen sollte jedoch nicht die komplette Routing-Tabelle mitgesendet werden, da dies in vielen Fällen das Netzwerk unnötig belastet. Das Auslassen der nicht veränderten Routen bei einem Triggered-Update stellt den einzigen Unterschied zu den regulären Updates dar. Nachdem alle Triggered-Updates erstellt und versendet wurden, kann das Route-Change-Flag wieder auf seinen Ausgangswert zurück gesetzt werden.

Falls die Verarbeitung von eingehenden RIPng-Nachrichten während der Erstellung der Response-Nachrichten erlaubt ist, müssen angepasste Schutzmechanismen gesetzt sein. Änderungen am Route-Change-Flag dürfen nicht als Ergebnis von der Verarbeitung eingehender Pakete während der Erstellung eines Triggered-Updates abhängig sein.

Bei der Erstellung von Triggered-Updates findet das Split-Horizon-Verfahren (Abschnitt 2.2.1.2) Anwendung. Dies dient der Vermeidung von

Routing-Schleifen innerhalb des Netzwerkes.

3.6.2 Request-Nachrichten

Mithilfe einer Request-Nachricht kann ein Router die gesamte oder partielle Teile der Routing-Tabelle eines anderen Routers anfordern. Die Router, welche die Requests empfangen, verarbeiten diese Eintrag für Eintrag. Falls kein Eintrag in einer Request vorliegt, wird keine Response-Nachricht erstellt, außer bei dem Spezialfall, dass die komplette Routing-Tabelle zurück gesendet werden soll. Der Zielport einer Request-Nachricht muss immer der RIPng-Port sein. Der Absendeport der Request-Nachricht kann beliebig sein.

Als Absenderadresse wird eine Global- oder Unique-Local-Adresse verwendet. Die Zieladresse ist analog dazu die Global- oder Unique-Local-Adresse des angefragten Routers.

Ein typisches Anwendungsgebiet einer solchen Request-Nachricht ist ein Multicast von einem neu installierten Router innerhalb eines Netzwerkes, um dessen Routing-Tabelle möglichst schnell zu füllen. Der Multicast bezweckt, dass eine Request-Nachricht über alle vorhandenen Schnittstellen an alle verbundenen Nachbarrouter gesendet wird, um dessen gesamte Routing-Tabelle in Form einer Response-Nachricht zu erhalten. Da der neue Router seine Nachbarrouter zu diesem Zeitpunkt noch nicht kennt, verwendet er die RIPng-Multicast-Adresse `FF02::9`. Als Absenderadresse wird die Link-Local-Adresse der Absenderschnittstelle verwendet. Die Response-Nachrichten der Nachbarrouter werden unter der Berücksichtigung der Split-Horizon-Maßnahme zurück gesendet. Die Kommunikation zwischen den Routern wird über den RIPng-Port des jeweiligen Routers geregelt. Diese Art von Request-Nachrichten werden generelle Request-Nachrichten genannt.

Um durch eine generelle Request-Nachricht die gesamte Routing-Tabelle abzufragen, enthält diese bestimmte Werte in den IPv6-Feldern und besitzt nur einen einzigen Routing-Tabellen-Eintrag. In diesem RTE hat das IPv6-Präfix den Wert `0::0` sowie eine Präfixlänge mit dem Wert null.

Die Metrik ist mit dem Wert 16 initialisiert. Durch diese Werte innerhalb der Request-Nachricht wird dem Router, der diese Nachricht erhält, angezeigt, dass er seine komplette Routing-Tabelle zurücksenden soll.

Eine generelle Request-Nachricht wird in Abbildung 3.8 dargestellt. Da die MTU sowie das physikalische Medium die Größe der RIPng-Nachrichten lokal beschränkt, werden in manchen Fällen auch mehrere RIPng-Datenpakete zur Übertragung der Routing-Tabellen benötigt.

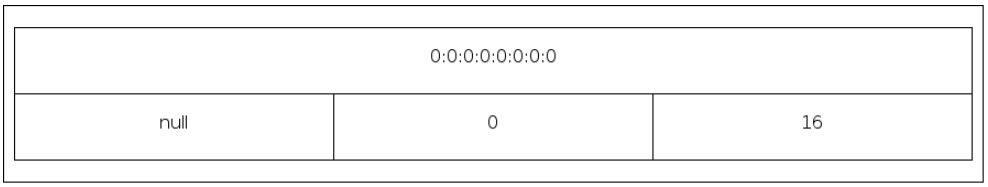

Abbildung 3.8: Generelle Request-Nachricht

Im Gegensatz zu der gesamten Routing-Tabelle können durch Request-Nachrichten auch nur bestimmte Routen auf dem jeweiligen Router abgefragt werden. Dies wird als eine partielle Abfrage der Routing-Tabelle bezeichnet. Bei der partiellen Abfrage werden die Datenpakete und die RTE des absendenden Routers der Reihe nach vom Router abgearbeitet. Dafür vergleicht der die Request-Nachricht erhaltende Router die Request-RTE mit seinen vorhandenen RTE. Falls er einen entsprechenden Eintrag innerhalb seiner Routing-Tabelle findet, der mit einem Request-RTE übereinstimmt, kopiert er die Metrik in den Request-RTE. Kann kein entsprechender Eintrag gefunden werden, wird die Metrik in dem Request-RTE auf den Wert 16 gesetzt. Sobald alle Request-RTE verarbeitet sind, wird im Header der RIPng-Nachricht das Command-Feld auf den Response-Wert gesetzt und die RIPng-Nachricht zurück an den Absender gesendet.

Falls nur die Routing-Tabelle eines bestimmten Routers benötigt wird, wird die Request-Nachricht direkt an den RIPng-Port des Routers über den UDP-Port, welcher ungleich dem RIPng-Port sein muss, gesendet. Diese Nachrichten werden spezifische Request-Nachrichten genannt und an die Unique-Local-Adresse des Routers gesendet. Als Absenderadresse wird die Global- oder Unique-Local-Adresse angegeben. Durch den Er-

halt einer solchen Request-Nachricht sendet der Router seine Response-Nachricht mit der gesamten beziehungsweise der partiellen Routing-Tabelle direkt anhand der IPv6-Adresse und des Ports zurück.

Die Split-Horizon-Maßnahme muss in diesem Fall nicht beachtet werden, da eine spezifische Request-Nachricht nur von einer Diagnosesoftware oder ähnlichem gestellt wird. Somit wird nur Wert auf den exakten Inhalt der Routing-Tabelle gelegt und nicht auf versteckte oder modifizierte Informationen.

Abbildung 3.9 stellt die Unterschiede zwischen den Request-Nachrichten noch einmal in zusammengefasster Form dar.

Request-Typ	Absenderadresse	Zieladresse	Absender-Port	Ziel-Port	Split-Horizon
Genereller Request	Link-Local-Adresse des Absenderinterfaces	FF02::9 (Multicast-Adresse für RIPng)	RIPng-Port (521)	RIPng-Port (521)	Ja
Spezifischer Request	Global- oder Unique-Adresse des Absenders	Global- oder Unique-Local-Adresse des angefragten Routers	Beliebig, jedoch kein RIPng-Port	RIPng-Port (521)	Nein

Abbildung 3.9: Request-Typen im Überblick [Hag09]

3.7 Routing-Policy und Sicherheit

Mithilfe einer Routing-Policy können die Einstellungen eines Routers wie die Filterfunktionen definiert werden, um somit eine höhere Sicherheit herzustellen. Zwar werden die möglichen Filterfunktionen bei dem RIPng-Protokoll nicht essentiell benötigt, jedoch bieten sie dem Administrator die Möglichkeit, eine gewisse Kontrolle über die Sicherheitseinstellungen und -maßnahmen zu haben.

Ein Beispiel für den Einsatz einer solchen Routing-Policy ist die Einschränkung der Router, von denen Update-Nachrichten empfangen beziehungsweise an die Update-Nachrichten gesendet werden dürfen. Diese Art von Policies können auf Grund der schlechten Sicherheitsvorkehrungen eines Netzwerkes getroffen werden, um eine Kommunikation mit

diesen entsprechenden Netzwerken zu unterbinden. Analog dazu können die Policies auch die Response-Nachrichten einschränken, damit diese nur an bestimmte Netzwerke zurückgesendet werden beziehungsweise nur Response-Nachrichten von bestimmten Netzwerken empfangen werden dürfen.

Im Folgenden werden zwei typische Anwendungen für Routing-Policies und Routing-Filter aufgezeigt. Das erste Beispiel ist die Erstellung einer Liste aller Nachbarrouter. Somit kann der Administrator für jeden Router die möglichen Nachbarrouter selbst bestimmen. Die Router akzeptieren nur noch Request- und Response-Nachrichten von den aufgelisteten Routern. Analog dazu kann der Administrator eine Liste erstellen, in der die möglichen Zielrouter angegeben werden.

Das zweite Beispiel ist die Erstellung einer Liste, in der erlaubte und unerlaubte Ziele anhand ihrer Präfixadresse angegeben werden. Diese Liste wird an ein bestimmtes Interface der Router für die ein- und ausgehenden Datenpakete gebunden. Somit erhalten nur in den Response-Nachrichten erwähnte Netzwerke die Datenpakete beziehungsweise werden nur eingehende Datenpakete von den erlaubten Netzwerken akzeptiert. Die Angabe der Präfixe erfolgt entweder für die erlaubten oder die nicht erlaubten Präfixe. Eine Angabe beider ist nicht möglich.

Da RIPng auf dem IPv6-Protokoll aufbaut, können die Sicherheitsfunktionen des Protokolls eine gesicherte Übertragung der Datenpakete gewährleisten. Zu den Sicherheitsfunktionen des IPv6-Protokolls gehören der IP-Authentification-Header und der IP-Encapsulating-Security-Payload-Header. Diese Sicherheitsfunktionen werden mithilfe der IPv6-Extension-Header beigefügt.

Durch den Authentification-Header wird die Integrität sowie die Authentisierung für IPv6-Datenpakete unterstützt. Jedoch wird durch diesen Sicherheitsmechanismus nicht die Vertraulichkeit hergestellt und überprüft. Weitere Authentisierungsmechanismen stehen zur Verfügung. Informationen zu dem Authentification-Header sind in [Atk95a] und [Ken05a] zu finden.

Der IP-Encapsulating-Security-Payload-Header bietet als Sicherheits-

mechanismus die gewünschte Vertraulichkeit sowie Integrität für die IPv6-Datenpakete des RIPng-Protokolls. [Atk95b] und [Ken05b] bieten weitere Informationen zu diesem Sicherheitsmechanismus.

Die gesicherte Datenübertragung wird nicht mehr durch eine direkte Verschlüsselungsfunktion des RIPng-Protokolls unterstützt, da dies von dem zu Grunde liegenden IPv6-Protokoll übernommen wird. Im Gegensatz dazu wurde im RIPv2-Protokoll eine direkte Verschlüsselungsfunktion eingeführt, jedoch auf Kosten der Header-Größe.

Kapitel 4

RIPng-Testszenarien

In diesem Kapitel wird RIPng auf verschiedene Eigenschaften untersucht. Die Grundlage stellt eine einfache Y-Topologie (Abschnitt 4.1) unter der Verwendung des Split-Horizon-Verfahrens (Abschnitt 2.2.1.2) und des IPv6-Protokolls (Abschnitt 2.1) dar.

Zur Erstellung der Testszenarien wird die Software *VNMUL - Virtual Network User Mode Linux* [1] verwendet. Mithilfe dieser Software können virtuelle Netzwerke lokal auf einem Computer erstellt werden.

Zur Konfiguration und Implementation von RIPng auf den Routern wird die Software *Quagga* [2] verwendet. Diese implementiert und aktiviert das dynamische Routing-Protokoll RIPng auf den jeweiligen Routern.

Zur Analyse der Datenpakete sowie zur Darstellung der Routing-Tabellen werden die bereitgestellten Bordmittel des kostenlos zur Verfügung stehenden Betriebssystems *Ubuntu 11.10* [3] verwendet.

Im kommenden Abschnitt wird der Aufbau der Y-Topologie betrachtet und näher erläutert. Es folgt eine Analyse des dynamischen Routings durch den Ausfall diverser Schnittstellen. Im letzten Abschnitt dieses Kapitels wird die Y-Topologie hinsichtlich des CTI-Problems analysiert. Die Timer sind im gesamten Szenario mit folgenden Werten belegt: Update-Timer 10 s, Timeout-Timer 30 s und Garbage-Collection-Timer 20 s.

[1] URL: http://neweb.dit.upm.es/vnumlwiki/index.php
[2] URL: http://www.quagga.net/
[3] URL: http://www.ubuntu.com/

4.1 Aufbau der Y-Topologie

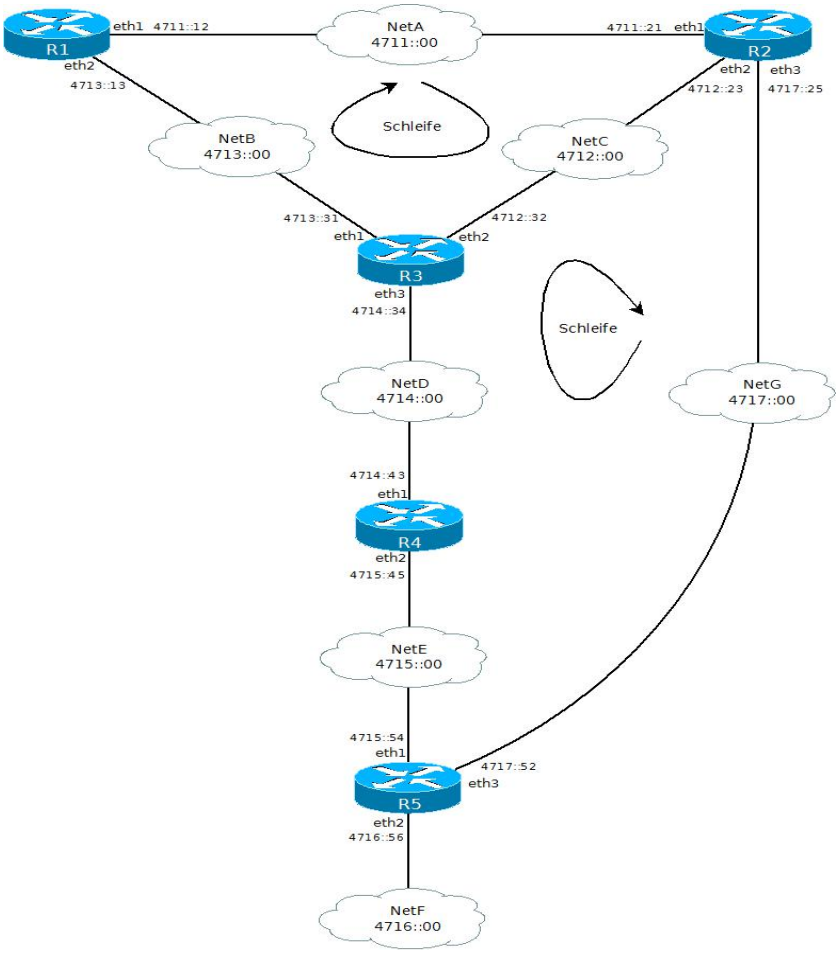

Abbildung 4.1: Aufbau der Y-Topologie

Die Y-Topologie wird in Abbildung 4.1 dargestellt. Dabei besteht die Topologie aus fünf Routern R1 bis R5 und sieben Netzwerken NetA bis NetG. Die Adressräume der Netzwerke reichen von `4711::00` (NetA) bis `4717::00` (NetG). Durch die Verbindung der Router R1, R2 und R3 zu einem Dreieck entsteht eine Schleife (*Loop*) innerhalb der Topologie. Die Verbindung zwischen Router R2 und Router R5 über das Netzwerk NetG bedingt eine weitere Schleife, jedoch wird das CTI-Problem in Abschnitt 4.4 an der Schleife zwischen Router R1, R2 und R3 aufgezeigt.

4.2 Szenario 01 - Ausfall der alternativen Route

Router R1 kann zur Erreichung des Netzwerkes NetF zwischen zwei verschiedenen Routen wählen. Die rote Route verläuft über Router R3, R4 und R5 und benötigt bis zum Ziel vier Hops. Eine weitere blaue Route ist über Router R2 und Router R5 an das Netzwerk NetF angebunden. Diese Route benötigt lediglich drei Hops bis zum Netzwerk NetF. In Abbildung 4.2 werden die zwei Routen aufgezeigt.

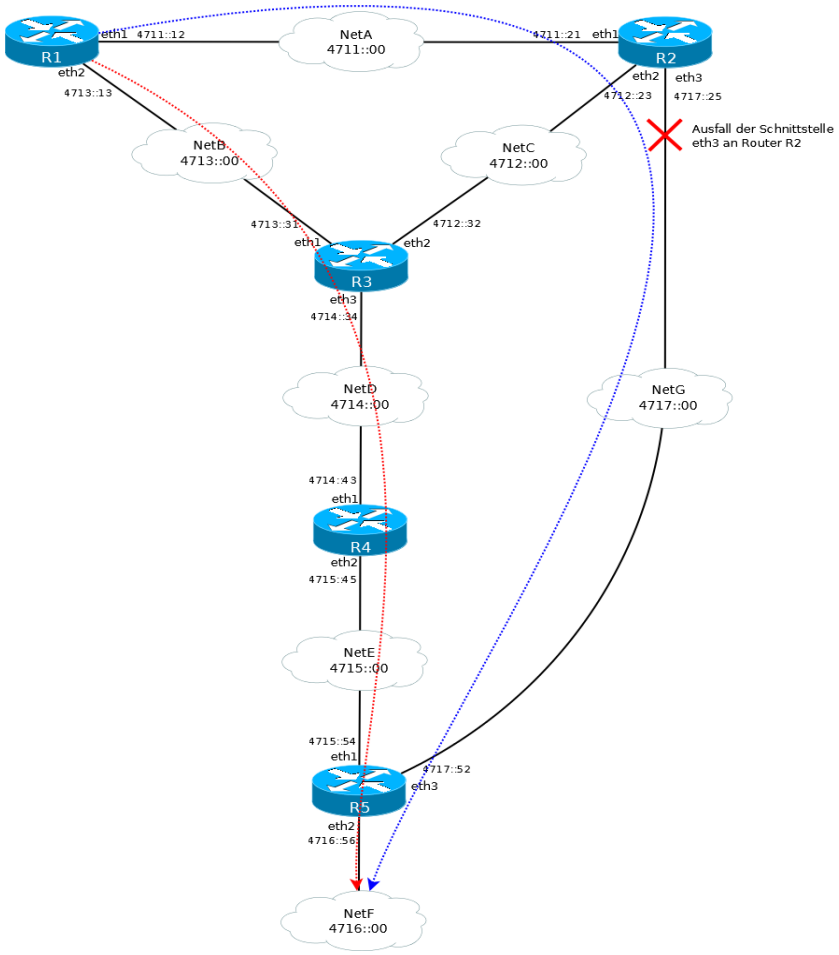

Abbildung 4.2: Übersicht Y-Topolgie - Szenario 01

Da RIPng die Metriken der möglichen Routen zu einem Ziel für die Wahl der kürzesten Route vergleicht, wird in diesem Fall automatisch die

Next-Hop-Adresse in der Routing-Tabelle von Router R1 mit der Scope-Link IPv6-Adresse des Routers R2 initialisiert, da über diesen Router und Router R5 das Netzwerk NetF zu den Metrikkosten von drei erreicht werden kann. Die Routing-Tabelle von Router R1 mit dem in dem Next-Hop-Feld eingetragenen Router R2 wird in Abbildung 4.3 dargestellt.

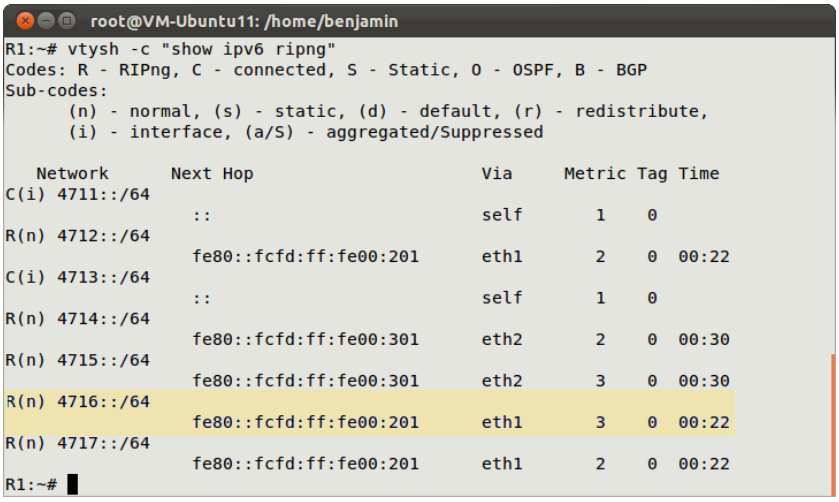

Abbildung 4.3: Routing-Tabelle R5 vor dem Ausfall

Im weiteren Verlauf des Testszenarios wird ein Ausfall der Schnittstelle eth3 an Router R2 mithilfe des Befehls `ifconfig eth3 down` simuliert. Der Ausfall wird in Abbildung 4.2 durch ein rotes Kreuz signalisiert. Dieser Befehl schaltet die Schnittstelle ab. Somit muss Router R1 über das RIPng-Protokoll eine neue Route zu dem Netzwerk NetF finden. Da nun keine alternativen kürzeren Routen mehr zu dem Netzwerk, außer die Route über Router R3, R4 und R5, vorliegen, passt Router R1 seine Routing-Tabelle dementsprechend an. Die Metrikkosten werden auf den Wert vier gesetzt, das Next-Hop-Feld mit der Scope-Link IPv6-Adresse des Routers R3 belegt. Dieser leitet die Datenpakete von Router R1 entsprechend an Router R4 weiter, welcher diese wiederum an Router R5 sendet. Router R5 verfügt über eine direkte Verbindung zu dem Netzwerk NetF. Die angepasste Routing-Tabelle von Router R1 wird in Abbildung 4.4 wiedergegeben.

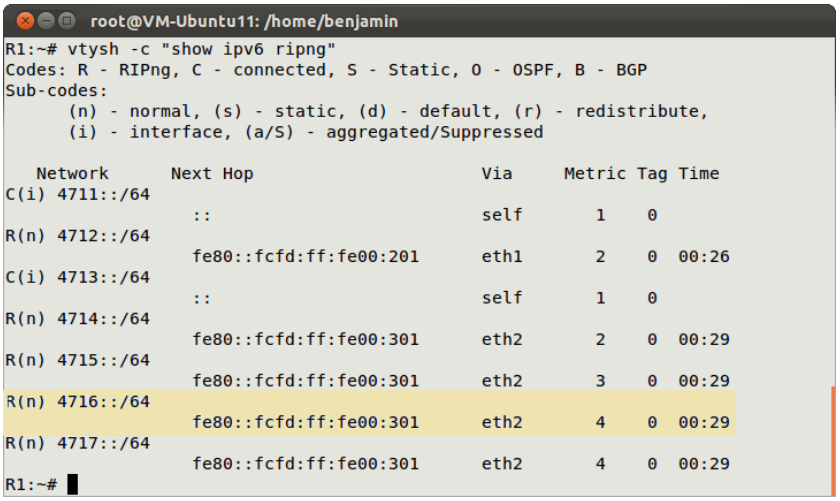

Abbildung 4.4: Routing-Tabelle R5 nach dem Ausfall

Dieses Szenario zeigt, dass ein problemloses Routing unter RIPng und IPv6 möglich ist. Es treten keine Schwierigkeiten bei der Berechnung der kürzesten Strecken zu einem Ziel durch den verwendeten Distanz-Vektor-Algorithmus auf. Ebenfalls werden Änderungen, wie der Ausfall einer Schnittstelle und der damit verbundene Ausfall einer Verbindung, durch das RIPng-Protokoll problemlos an die weiteren Router innerhalb des Netzwerkes weitergeleitet.

4.3 Szenario 02 - Ausfall der Hauptroute

Nachdem in Szenario 01 die alternative Route über das Netzwerk NetG ausgefallen ist, bleibt lediglich für die Übertragung von Datenpaketen von Router R1 zu dem Netzwerk NetF die Route über Router R3, R4 und R5. In diesem Szenarios fällt zusätzlich diese Route über R3 aus. Dafür wird die Schnittstelle eth3 an Router R3 durch den Befehl `ifconfig eth3 down` abgeschaltet. Die Verbindung zwischen dem Router R3 und dem Netzwerk NetD ist somit aufgehoben. Die Abbildung 4.5 stellt die angepasste Y-Topologie dar. Das weitere rote Kreuz markiert den Ausfall der Verbindung zwischen dem Router R3 und dem Netzwerk NetD.

Durch den Ausfall beider möglichen Routen wird die Y-Topologie in zwei Teilnetze separiert. Ein Teilnetz besteht aus den Routern R1, R2 und R3. Das andere Netz setzt sich aus den Routern R4 und R5 zusammen. Da durch den Ausfall der Schnittstelle eth3 an Router R3 die Topologie aufgeteilt wird, ist eine Kommunikation zwischen diesen beiden Teilnetzen nicht mehr möglich. Somit können keine Datenpakete untereinander ausgetauscht werden. Dies spiegelt sich in den Routing-Tabellen der Router wieder.

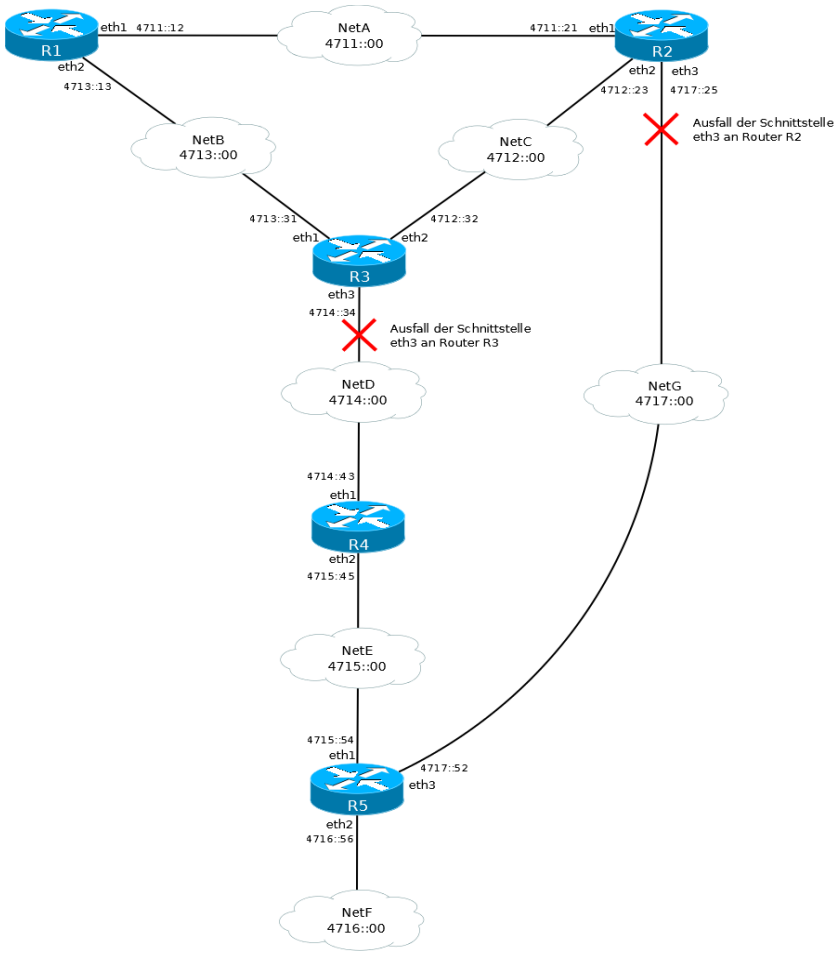

Abbildung 4.5: Übersicht Y-Topologie - Szenario 02

Dafür wird die Routing-Tabelle von Router R1 vor und nach dem Ausfall der Schnittstelle an Router R3 genauer betrachtet. Die Abbildung 4.4

aus Szenario 01 veranschaulicht die Routing-Tabelle von Router R1 vor dem Ausfall der Schnittstelle an R3, jedoch nach dem Ausfall der Schnittstelle eth3 an Router R2. Die Schnittstelle eth3 an Router R3 wird ebenfalls mit dem Befehl `ifconfig eth3 down` abgeschaltet und bedingt gewisse Änderungen in den Routing-Tabellen der Router. Diese sind in den folgenden Abbildungen für Router R1 und Router R5 aufgeführt.

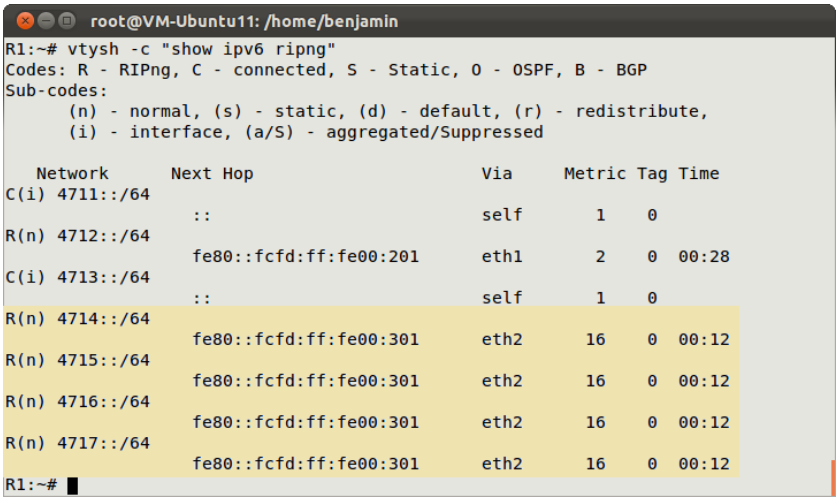

Abbildung 4.6: Routing-Tabelle R1 vor Ablauf Garbage-Collection-Timer

Nach der Abschaltung der Schnittstelle eth3 an Router R3 werden die Netzwerke NetD bis NetG mit der Metrik 16 in der Routing-Tabelle von Router R1 versehen. Dies bedingt den Start des Garbage-Collection-Timers für die eingetragenen Routen zu den Netzwerken. Die Änderungen der Metriken sind in Abbildung 4.6 aufgezeigt. Die Werte in der Time-Spalte zeigen an, dass die Routen in zwölf Sekunden aus der Routing-Tabelle von Router R1 entfernt werden, falls keine Update-Nachricht in dieser Zeit eintrifft. Nach Ablauf des Garbage-Collection-Timers werden die Netzwerke `4714::00` (NetD), `4715::00` (NetE), `4716::00` (NetF) sowie `4717::00` (NetG) aus der Routing-Tabelle des Routers R1 gelöscht. Die Netzwerke `4711::00` (NetA), `4712::00` (NetC) sowie `4713::00` (NetB) bleiben weiterhin für Router R1 erreichbar. Abbildung 4.7 zeigt die Routing-Tabelle von Router R1 nach Ablauf des Garbage-Collection-Timers.

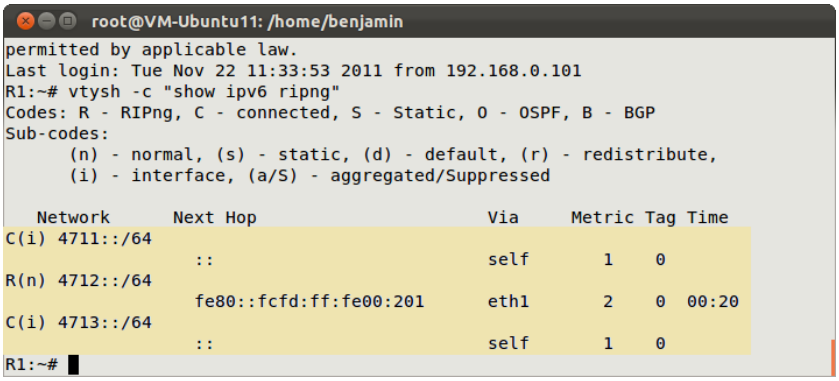

Abbildung 4.7: Routing-Tabelle R1 nach Ablauf Garbage-Collection-Timer

Ähnliche Auswirkungen sind in den Routing-Tabellen der Router R4 und R5 aus dem abgetrennten Teilnetz ersichtlich. Wie in Abbildung 4.8 zu sehen, besteht die Routing-Tabelle des Routers R5 nur noch aus den Netzwerken, die in den Routing-Tabellen der Router aus dem anderen Teilnetz nicht vorzufinden sind. Die Abbildung zeigt die Routing-Tabelle des Routers R5 nach dem Ausfall der Schnittstelle an Router R3 und dem Ablauf der Garbage-Collection-Timer. Router R5 kann die Netzwerke NetE, NetF sowie NetG direkt erreichen. Datenpakete, welche an das Netzwerk NetD gesendet werden, sind durch den Router R4 zu den Metrikkosten zwei weiterzuleiten.

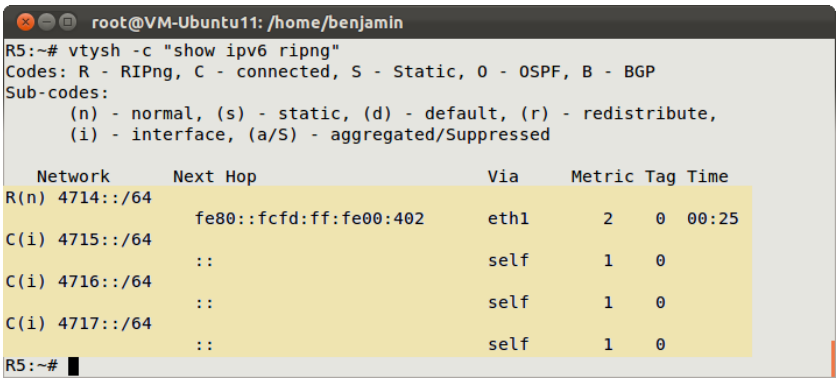

Abbildung 4.8: Routing-Tabelle R5 nach Ablauf Garbage-Collection-Timer

4.4 Szenario 03 - Counting-To-Infinity

In dem dritten Szenario wird die Y-Topologie hinsichtlich des CTI-Problems analysiert. Dafür bleibt die Schnittstelle eth3 an Router R2 weiterhin abgeschaltet. Des Weiteren wird im Verlauf des Testszenarios die Schnittstelle eth2 an Router R5 durch den Befehl `ifconfig eth2 down` abgeschaltet und Router R3 dazu veranlasst, für eine gewisse Zeit keine Datenpakete mehr über die Schnittstelle eth1 an den Router R1 zu senden. Diese Bedingungen sind notwendig, um ein Counting-To-Infinity-Problem herbeiführen zu können. Abbildung 4.9 zeigt die Änderungen der Y-Topologie kompakt auf.

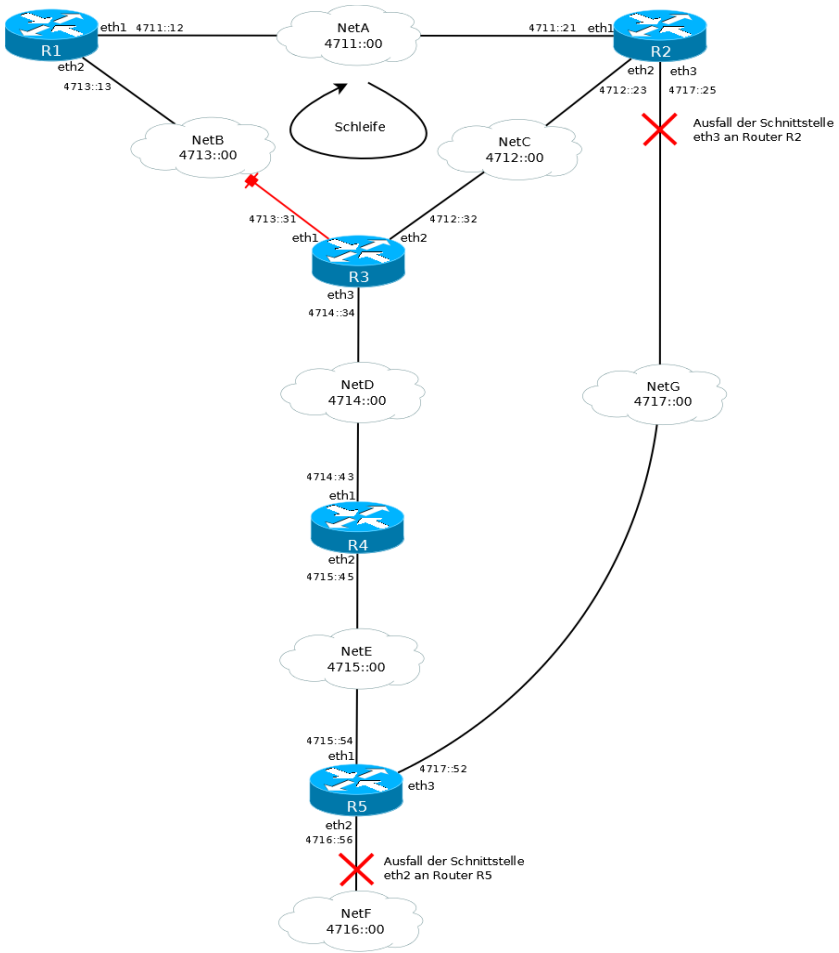

Abbildung 4.9: Übersicht Y-Topologie - Szenario 03

Die Abschaltung der Schnittstelle eth2 an Router R5 simuliert einen Ausfall der Verbindung von Router R5 zu dem Netzwerk NetF. Das Netzwerk NetF ist somit nicht mehr über die IPv6-Adresse `4715::56` erreichbar. Die folgenden Abbildungen geben einen Einblick in die Routing-Tabelle des Routers R5 und zeigen die Auswirkungen des Ausfalls der Schnittstelle eth2 auf.

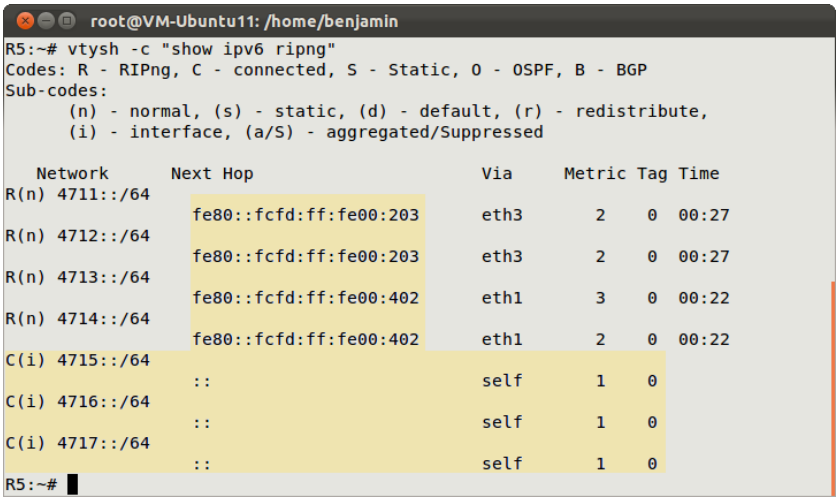

Abbildung 4.10: Routing-Tabelle R5 vor Ausfall der Schnittstelle eth2

Wie in Abbildung 4.10 dargestellt, kann der Router R5 die Netze NetE, NetF und NetG zu den Metrikkosten von eins direkt erreichen. Die anderen Netze in der Topologie sind durch das RIPng-Protokoll über den Router R2 oder den Router R4 vor dem Ausfall der Schnittstelle eth2 erreichbar. Dies wird durch die Next-Hop-Einträge, welche die Scope-Link IPv6-Adressen dieser Router aufzeigen, veranschaulicht.

Nach dem Abschalten der Schnittstelle eth2 wird die Metrik für das Netzwerk NetF in der Routing-Tabelle im Router R5 sofort auf den Wert 16 gesetzt. Das Netzwerk wird somit als nicht mehr erreichbar gekennzeichnet. Durch das Setzen der Metrik auf den Wert 16 wird außerdem der Garbage-Collection-Timer (Abschnitt 3.5) gestartet. Des Weiteren wird durch die Änderung der Metrikkosten innerhalb der Routing-Tabelle ein Triggered-Update (Abschnitt 3.6.1.2) in Form einer Response-Nachricht an die Nachbarrouter, in diesem Fall Router R4, gesendet, um weitere Router

innerhalb der Topologie von dem Ausfall der Verbindung und dem damit verbundenen Ausfall des Netzwerks NetF zu informieren. Ein Triggered-Update an Router R2 ist durch den Ausfall der Schnittstelle eth3 an Router R2 nur indirekt über die Router R4 und R3 möglich. Die Änderungen innerhalb der Routing-Tabelle von Router R5 nach der Abschaltung der Schnittstelle eth2 sind in Abbildung 4.11 sichtbar. Nach Ablauf des Garbage-Collection-Timers wird die Route aus der Routing-Tabelle gelöscht.

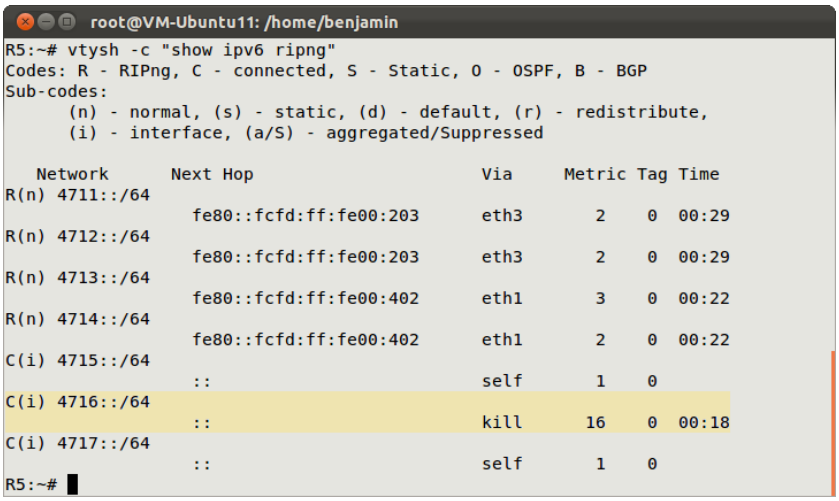

Abbildung 4.11: Routing-Tabelle R5 nach Ausfall der Schnittstelle eth2

Da die Änderungen durch das ganze Netzwerk versendet werden, ist jeder Router nach einer gewissen Zeit über den Ausfall der Verbindung informiert (Konvergenzzeit). Der Ausfall wird über den Router R4 an den Router R3 per Triggered-Update weitergeleitet. Dieser sendet die Informationen jedoch nur per Schnittstelle eth2 an Router R2 weiter, nicht an Router R1. Eine Weiterleitung über die Schnittstelle eth1 an den Router R1 wird durch die in Abbildung 4.12 dargestellte Einstellung in den ip6tables vermieden. Durch den Befehl `ip6tables -A OUTPUT -o eth1 -j DROP` auf Router R3 wird dieser dazu veranlasst, keine Datenpakete mehr über die Schnittstelle eth1 zu versenden.

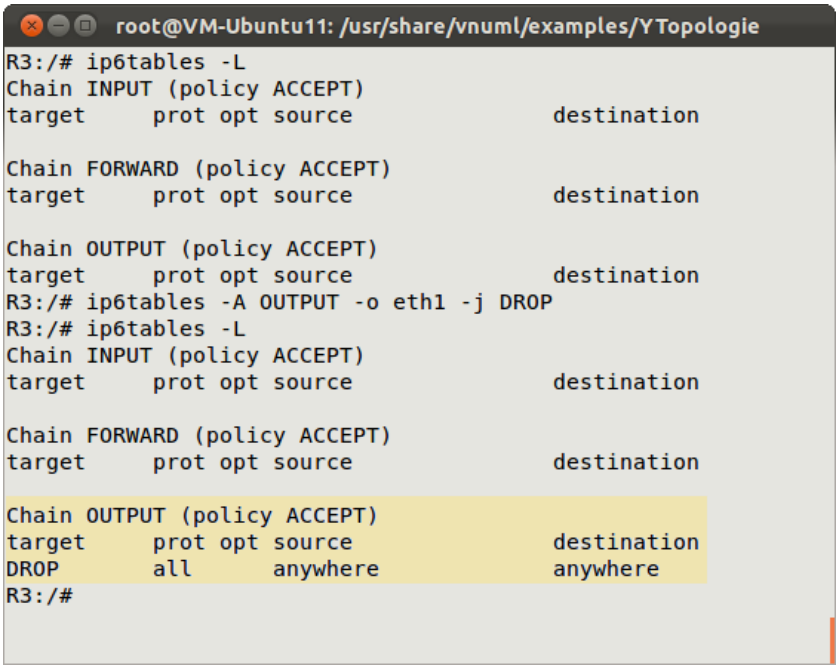

Abbildung 4.12: Blockierung der Schnittstelle eth1 an Router R3

Das von Linux bereitgestellte ip6table-Programm dient der Filterung des IPv6-Datenverkehrs. Diese Filterungseinstellung wird vorausgesetzt, um das CTI-Problem darstellen zu können. Außerdem müssen die Einstellungen vor dem Ausfall der Schnittstelle eth2 an Router R5 schon aktiviert sein, da die Änderungen in den Routing-Tabellen per Triggered-Update direkt weitergeleitet werden. Jedoch ist die Zeitspanne durch den Ablauf des Update-Timers in R1 begrenzt, da dieser nach Ablauf ebenfalls den Garbage-Collection-Timer aufruft, um die Route zu Router R3 zu löschen.

Durch die Filterung werden Datenpakete ausgehend vom Router R5 mit dem Metrikwert 16 nur an Router R2 weitergeleitet. Dieser passt seine interne Metrik in der Routing-Tabelle entsprechend an und sendet die neuen Änderungen per Triggered-Update weiter. Router R1 erhält die Änderungen in einer Response-Nachricht, ändert seine Metrik jedoch nicht auf den Wert 16, da er sich noch in dem Zustand befindet, dass er das Netzwerk NetF über den Router R3 zu den Metrikkosten vier erreichen kann. Durch das aktive Split-Horizon-Verfahren darf Router R1 diese Informationen nicht direkt an Router R2 zurücksenden. Erst mit dem nächsten

regulären Update werden die Informationen an den Router R2 gesendet.

Router R2 übernimmt die Metrik von Router R1 zur Erreichung des Netzwerkes NetF aus dem regulären Update, da die Metrik vier kleiner als die ihm vorliegende Metrik 16 ist. Dazu addiert er auf die Metrikkosten den Wert eins für die Routingkosten zu Router R1 und trägt die Metrikkosten fünf zur Erreichung des Netzwerkes NetF über Router R1 in seiner Routing-Tabelle ein. Diesen Wert sendet er per Multicast und Triggered-Update an seine Nachbarrouter, in diesem Fall an den Router R3. Router R1 kann wegen des Split-Horizon-Verfahrens nicht angesprochen werden.

Router R3 übernimmt nach einem Vergleich mit seinem aktuellen Metrikwert 16 die Metrik von Router R2 zur Erreichung des Netzwerkes NetF, in dem er ebenfalls den Wert eins zu den Metrikkosten für den Zwischenrouter R2 hinzu addiert und den Metrikwert sechs in seine Routing-Tabelle zur Erreichung des Netzwerkes NetF einträgt.

In der Zwischenzeit müssen die Filtereinstellungen auf Router R3 für Router R1 verworfen sein. Somit geht nun auch Router R1 davon aus, dass das Netzwerk NetF nicht mehr erreichbar ist. Jedoch befindet er sich nun in dem Zustand, dass er über den Router R3 zu den Metrikkosten sechs das Netzwerk NetF erreichen kann, gelangt jedoch nun über Router R3 in die eigentliche Schleife. Router R3 sendet seine Metrik für das Netzwerk NetF an Router R1. Dieser addiert wieder den Wert eins hinzu und übernimmt die Metrikkosten sieben in seiner internen Routing-Tabelle. Die drei Router zählen sich bis zu dem Wert 16 gegenseitig hoch.

Dieser Wert stellt den Grenzwert bei RIPng für das CTI-Problem dar. Deshalb wird bei Erreichung des Wertes 16 der Garbage-Collection-Timer für die Route gestartet und damit verbunden die endgültige Löschung des Routing-Eintrages veranlasst. Falls keine weitere Nachricht mit einer gültigen Metrik während dieser Zeit eintrifft, wird die Route gelöscht. Durch die Löschung der Route wird die Schleife innerhalb des Netzwerkes aufgelöst.

Der Vorgang des „Hochzählens“ der Router R1, R2 und R3 wird anhand der Routing-Tabelle von Router R3 in den folgenden Abbildungen verdeutlicht. Zu Anfang ist der Metrikeintrag für das Erreichen des Netz-

werkes NetF noch mit dem Wert drei initialisiert. Dies wird in Abbildung 4.13 dargestellt.

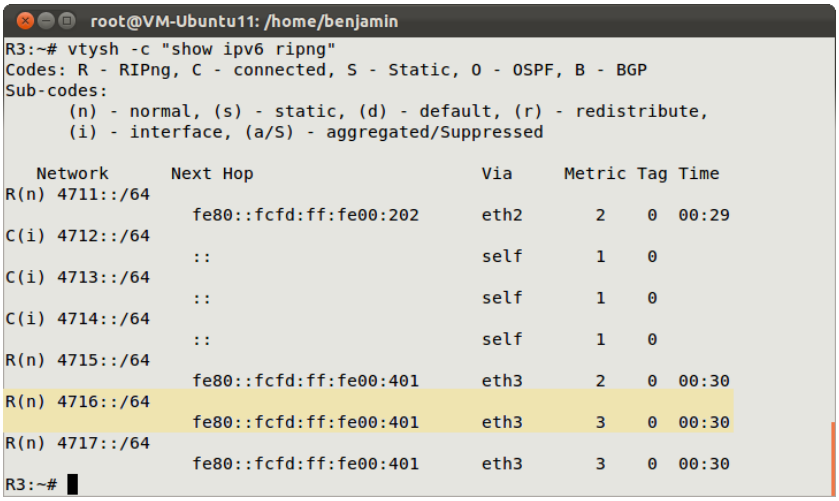

Abbildung 4.13: Routing-Tabelle R3 mit der Metrik 3

Nach dem Abschalten der Filterungseinstellungen an Router R3 wird in dessen Routing-Tabelle als Next-Hop-Adresse der Router R2 gesetzt, da Router R2 eine vermeintlich kürzere Route zu Netzwerk NetF kennt. Durch die Schleife zwischen den Routern R1, R2 und R3 erhöht sich der Metrikwert pro Schleifenrunde um den Wert drei. Deshalb nimmt die Metrik in der Routing-Tabelle des Routers R3 in den folgenden Durchläufen der Schleife die Werte sechs, neun, zwölf , 15 und 16 an.

In Abbildung 4.14 ist die Routing-Tabelle des Routers R3 zu dem Zeitpunkt dargestellt, in dem die Metrik zu dem Netzwerk NetF mit dem Wert zwölf belegt ist.

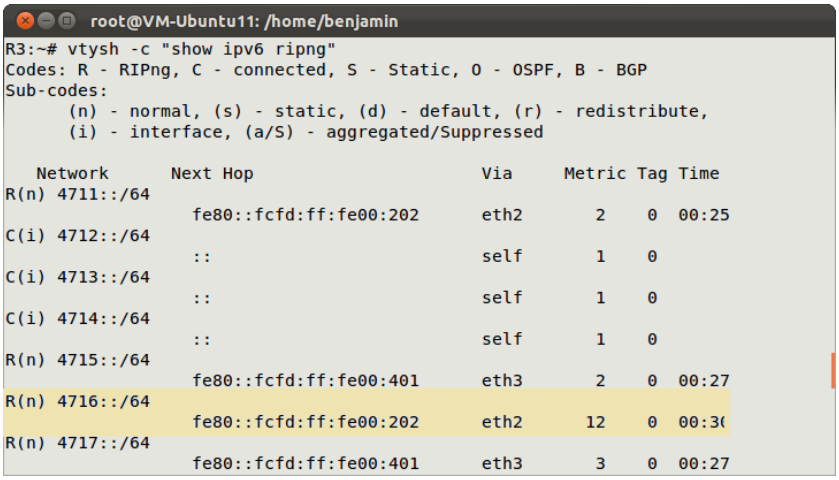

Abbildung 4.14: Routing-Tabelle R3 mit der Metrik 12

Die Metrik wird bis zu dem Wert 16 hochgesetzt. Dieser Wert stellt die Grenze der Gültigkeit einer Route im RIPng-Protokoll dar und aktiviert den Garbage-Collection-Timer. Dieser löscht nach Ablauf der Zeit die Route und damit verbunden die Schleife in der Topologie endgültig. Dies wird in der Routing-Tabelle von Router R3 in Abbildung 4.15 dargestellt. Die Metrik wurde bereits bis zu dem Wert 16 hochgezählt und der Eintrag für das Netzwerk NetF wird in 18 Sekunden gelöscht. Damit wird der Eintrag für Netzwerk NetF endgültig aus der Routing-Tabelle entfernt.

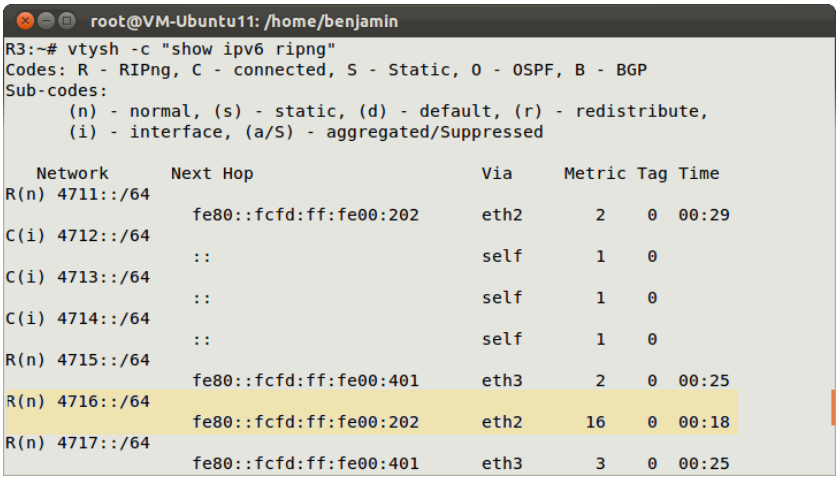

Abbildung 4.15: Routing-Tabelle R3 mit der Metrik 16

In kleinen Topologien wie dieser, können Schleifen mithilfe des Counting-to-Infinity-Verfahrens relativ schnell entdeckt und behoben werden. Je größer jedoch die Netzwerktopologien sind, desto mehr Zeit wird benötigt, um eine Schleife innerhalb des Netzwerkes feststellen und erfolgreich beheben zu können.

Kapitel 5

Fazit und Ausblick

In dieser Ausarbeitung wurde das dynamische Routing-Protokoll RIPng unter der Verwendung des IPv6-Protokolls einer genauen Betrachtung unterzogen. Nach einer kurzen Einführung in die Eigenschaften des IPv6-Protokolls, den Distanz-Vektor-Algorithmus sowie den Vorgänger für die IPv4-Adressen, RIPv2, wurden die theoretischen Eigenschaften des RIPng-Protokolls explizit erläutert. Dabei ist festgestellt worden, dass der Aufbau des RIPng-Protokolls dem generellen Aufbau der RIP-Protokoll für die IPv4-Adressen recht ähnlich ist. Zwar sind die IPv4-Adressen durch die IPv6-Adressen ersetzt worden und bedingen deshalb ein neues Routing-Protokoll der RIP-Familie, jedoch sind viele Eigenschaften wie die Berechnung der kürzesten Wege mithilfe des Distanz-Vektor-Algorithmus, die generellen Einschränkungen durch das RIP-Protokoll oder das Auflösen von Schleifen (CTI-Problem) innerhalb eines Netzwerkes die gleichen geblieben. Eine weitere Veränderung sind die neu hinzugefügten Extension-Header. Zwar bieten diese neue, flexible Möglichkeiten zur Übertragung von zusätzlichen Routing-Informationen und eine bessere Performance bei der Übertragung der Datenpakete, stellen aber zudem neue Sicherheitslücken wie die möglichen Denial-of-Service-Attacken durch den Routing-Header vom Typ 0 dar. Datenpakete mit diesem Routing-Header werden aus diesem Grund von den heutigen Betriebssystemen und Routern ignoriert und verworfen.

Die einzelnen, auf der Y-Topologie aufbauenden Testszenarien zeigen das problemlose Routing-Verhalten von RIPng unter IPv6 auf. Das Routing erfolgt wie in den vorigen bekannten RIP-Protokollen und es treten keine Schwierigkeiten durch die Verwendung des IPv6-Protokolls auf. Jedoch gelten diese Ergebnisse nur für die virtuellen Testszenarien. In der realen Welt können immer noch durch Hardwareunstimmigkeiten oder Implementationsproblemen Schwierigkeiten bei der Verwendung von IPv6 auftreten. Das CTI-Testszenario zeigt, dass unter RIPng und IPv6 durch die Schleife in der Y-Topologie die Metriken der Routen ebenfalls bis zu dem durch RIPng-gesetzten Grenzwert hochgezählt werden und die Löschung der Route durch den Garbage-Collection-Timer bedingen.

Abschließend ist zu festzustellen, das IPv6 bedingt durch das enorme Wachstum an internetfähigen Geräten eine immer größere Verbreitung in der Welt der Datenkommunikation findet und weiterhin finden wird. Dies ist dadurch zu erklären, dass die anfänglichen Kompatibilitätsprobleme der Hardware unter der Verwendung der neuen IPv6-Adressen sowie neue auftretende Sicherheitsfragen weitestgehend aufgeklärt worden sind und ein fehlerloses Routing auf Basis des IPv6-Protokolls stattfinden kann. Dies folgert die Entwicklung und Anpassung an das IPv6-Protokoll für neue und bestehende Routing-Protokolle. Die RIP-Familie hat durch das RIPng Protokoll diese Anpassung erfolgreich vollzogen und ist für den Einsatz unter IPv6 bestens geeignet.

Die Zukunft wird zeigen, ob sich das RIPng-Protokoll als Standardprotokoll unter den Interior-Gateway-Protokollen durchsetzen kann. Da nur die wirklich nötigsten Veränderungen für IPv6, wie die Erweiterung auf 128 Bit Adressen, eingebaut wurden, könnten eine große Anzahl der Administratoren, welche ihre IPv4-Netzwerke noch mit einem Protokoll der RIP-Familie betreiben, auf der Suche nach einem geeigneten Routing-Protokoll unter Umständen wieder zu einem Protokoll der RIP-Familie für IPv6 greifen, nämlich dem RIPng-Protokoll.

Danksagung

Bei allen, die mir bei der Ausarbeitung meiner Bachelor-Thesis zur Seite gestanden haben, möchte ich mich hiermit herzlich bedanken. Vor allem bei meinem Betreuer Frank Bohdanowicz, der mir bei jeglichen Fragen mit Rat und Tat zur Seite stand und mich während der gesamten Zeit der Erstellung der Bachelorarbeit bestens betreut hat.

Außerdem gilt Prof. Dr. Steigner ein großer Dank. Dieser lehrte mich, dass auch neue, weiterentwickelte Techniken, welche vermeintlich nur weitere Verbesserungen und Vorteile mit sich bringen, trotzdem kritisch betrachtet und hinterfragt werden müssen, da in der Literatur die Nachteile zum Teil schöngeschrieben werden.

Ein weiterer Dank gilt allen, die meine Arbeit trotz möglicher fehlender Sachkenntnisse Korrektur gelesen haben und mir den ein oder anderen Fehler aufzeigen konnten. Vor allem aber meiner Freundin, die mir ebenfalls bei meinen Fragen zum Ausdruck und zur Rechtschreibung mit Rat und Tat zur Seite stand. Danke!

Anhang A

XML-Datei für Y-Topologie

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE vnuml SYSTEM "/usr/share/xml/vnuml/vnuml.dtd">
3
4  <!--Aufbau der Y-Topogie fuer das Testszenario, Version 1.0
       -->
5
6  <vnuml>
7   <!--Globale Einstellungen fuer virtuelle Maschinen-->
8   <global>
9       <version>1.8</version>
10      <simulation_name>YTopologie</simulation_name>
11      <ssh_version>2</ssh_version>
12      <ssh_key>~/.ssh/id_rsa.pub</ssh_key>
13      <automac/>
14      <vm_mgmt type="private" network="192.168.0.0" mask="24"
            offset="100">
15      <host_mapping/>
16      </vm_mgmt>
17      <vm_defaults>
18          <filesystem type="cow">/usr/share/vnuml/filesystems/
               root_fs_tutorial</filesystem>
19          <kernel>/usr/share/vnuml/kernels/linux</kernel>
20          <forwarding type="ipv6" />
```

```
21      </vm_defaults>
22    </global>
23
24
25  <!--Erstellung der Virtuellen Netzwerken-->
26
27  <net name="neta" mode="virtual_bridge" />
28  <net name="netb" mode="virtual_bridge" />
29  <net name="netc" mode="virtual_bridge" />
30  <net name="netd" mode="virtual_bridge" />
31  <net name="nete" mode="virtual_bridge" />
32  <net name="netf" mode="virtual_bridge" />
33  <net name="netg" mode="virtual_bridge" />
34
35
36  <!--Erstellung der virtuellen Maschinen (Router)-->
37
38  <!--VM R1-->
39  <vm name="R1">
40      <if id="1" net="neta">
41          <ipv6>4711::12</ipv6>
42      </if>
43      <if id="2" net="netb">
44          <ipv6>4713::13</ipv6>
45      </if>
46      <route type="ipv6" gw="4711::31">default</route>
47      <filetree seq="start" root="/etc/quagga/">conf</
            filetree>
48      <exec seq="start" type="verbatim">/usr/lib/quagga/zebra
            -f /etc/quagga/zebra.conf -d</exec>
49      <exec seq="start" type="verbatim">/usr/lib/quagga/
            ripngd -f /etc/quagga/ripngd.conf -d</exec>
50      <exec seq="stop" type="verbatim">killall zebra</exec>
51      <exec seq="stop" type="verbatim">killall ripngd</exec>
52  </vm>
```

```
53
54 <!--VM R2-->
55 <vm name="R2">
56     <if id="1" net="neta">
57         <ipv6>4711::21</ipv6>
58     </if>
59     <if id="2" net="netc">
60         <ipv6>4712::23</ipv6>
61     </if>
62     <if id="3" net="netg">
63         <ipv6>4717::25</ipv6>
64     </if>
65     <route type="ipv6" gw="4711::32">default</route>
66     <filetree seq="start" root="/etc/quagga/">conf</
         filetree>
67     <exec seq="start" type="verbatim">/usr/lib/quagga/zebra
         -f /etc/quagga/zebra.conf -d</exec>
68     <exec seq="start" type="verbatim">/usr/lib/quagga/
         ripngd -f /etc/quagga/ripngd.conf -d</exec>
69     <exec seq="stop" type="verbatim">killall zebra</exec>
70     <exec seq="stop" type="verbatim">killall ripngd</exec>
71 </vm>
72
73 <!--VM R3-->
74 <vm name="R3">
75     <if id="1" net="netb">
76         <ipv6>4713::31</ipv6>
77     </if>
78     <if id="2" net="netc">
79         <ipv6>4712::32</ipv6>
80     </if>
81     <if id="3" net="netd">
82         <ipv6>4714::34</ipv6>
83     </if>
84     <route type="ipv6" gw="4711::43">default</route>
```

```
85      <filetree seq="start" root="/etc/quagga/">conf</
            filetree>
86      <exec seq="start" type="verbatim">/usr/lib/quagga/zebra
            -f /etc/quagga/zebra.conf -d</exec>
87      <exec seq="start" type="verbatim">/usr/lib/quagga/
            ripngd -f /etc/quagga/ripngd.conf -d</exec>
88      <exec seq="stop" type="verbatim">killall zebra</exec>
89      <exec seq="stop" type="verbatim">killall ripngd</exec>
90  </vm>
91
92  <!--VM R4-->
93  <vm name="R4">
94      <if id="1" net="netd">
95          <ipv6>4714::43</ipv6>
96      </if>
97      <if id="2" net="nete">
98          <ipv6>4715::45</ipv6>
99      </if>
100     <route type="ipv6" gw="4711::54">default</route>
101     <filetree seq="start" root="/etc/quagga/">conf</
            filetree>
102     <exec seq="start" type="verbatim">/usr/lib/quagga/zebra
            -f /etc/quagga/zebra.conf -d</exec>
103     <exec seq="start" type="verbatim">/usr/lib/quagga/
            ripngd -f /etc/quagga/ripngd.conf -d</exec>
104     <exec seq="stop" type="verbatim">killall zebra</exec>
105     <exec seq="stop" type="verbatim">killall ripngd</exec>
106 </vm>
107
108 <!--VM R5-->
109 <vm name="R5">
110     <if id="1" net="nete">
111         <ipv6>4715::54</ipv6>
112     </if>
113     <if id="2" net="netf">
```

```
114         <ipv6>4716::55</ipv6>
115     </if>
116     <if id="3" net="netg">
117         <ipv6>4717::52</ipv6>
118     </if>
119     <filetree seq="start" root="/etc/quagga/">conf</
            filetree>
120     <exec seq="start" type="verbatim">/usr/lib/quagga/zebra
            -f /etc/quagga/zebra.conf -d</exec>
121     <exec seq="start" type="verbatim">/usr/lib/quagga/
            ripngd -f /etc/quagga/ripngd.conf -d</exec>
122     <exec seq="stop" type="verbatim">killall zebra</exec>
123     <exec seq="stop" type="verbatim">killall ripngd</exec>
124 </vm>
125 </vnuml>
```

Anhang B

Zebra-Konfigurationsdatei

```
1  ! Konfiguration von Zebra unter Quagga
2  ! Version 1.0
3
4
5  ! Login-Einstellungen
6  hostname zebra
7  password xxxx
8  enable password xxxx
9
10 ! Logging-Datei zwecks Fehlerbehandlung
11 log file zebra.log
```

Anhang C

RIPng-Konfigurationsdatei

```
1  ! Konfiguration von RIPng unter Quagga
2  ! Version 1.0
3
4  ! Login-Einstellungen
5  hostname ripng
6  password xxxx
7
8  ! Spezifizierung des Routing-Protokolls, des Netzwerkes
9  ! sowie der Einstellungen der Timer
10 router ripng
11 network ::/0
12 timers basic 10 30 20
13
14 ! Logging-Datei zwecks Fehlerbehandlung
15 log file ripng.log
```

Literaturverzeichnis

[Atk95a] R. Atkinson (August, 1995): *RFC1826 - IP Authentication Header*.

[Atk95b] R. Atkinson (August, 1995): *RFC1827 - IP Encapsulating Security Payload (ESP)*.

[Bio07] Philippe Biondi, Arnaud Ebalard (2007): *IPv6-Routing-Header-Security*. *Vortrag*.

[Car02] Carsten Bormann and Jörg Ott (2002). *Konzepte der Internet Technik*.

[Dee95] S. Deering, R. Hinden (Dezember, 1995): *RFC1883 - Internet Protocol, Version 6 (IPv6) Specification*.

[Dee98] S. Deering, R. Hinden (Dezember, 1998): *RFC2460-Internet-Protocol-Version 6 (IPv6) Specification*.

[Ege01] K. Egevang, P. Srisuresh (Januar, 2001): *RFC3022 - Traditional IP Network Address Translator (Traditional NAT)* .

[End10] Johannes Endres (2010): *15 Jahre IPv6*. *Artikel der Zeitschrift CT*.

[Enz11a] Wikipedia Die freie Enzyklopaedie (2011). *IPv6*. Zugriffsdatum: 26. August 2011.

[Enz11b] Wikipedia Die freie Enzyklopaedie (2011). *Routing*. Zugriffsdatum: 28. November 2011.

[Ful93] V. Fuller, T. Li, J. Yu, K. Varadhan (September, 1993): *RFC1519 - Classless Inter-Domain Routing (CIDR):an Address Assignment and Aggregation Strategy*.

[Ger09] Alexander Gernler (2009): *Hürden für IPv6. Artikel der Zeitschrift Hackin9*.

[G.M97] G.Malkin, R. Minnear (Januar, 1997): *RFC2080 - RIPng for IPv6*.

[Hab02] B. Haberman (August, 2002): *RFC3307 - Allocation Guidelines for IPv6 Multicast Addresses*.

[Hag09] Silvia Hagen (2009): *IPV6-Grundlagen, Funktionalität, Integration*. Sunny Edition.

[Hed98] C. Hedrick (Juni, 1998): *RFC1058 - Routing Information Protocol*.

[Hin05] R. Hinden, B. Haberman (Oktober, 2005): *RFC4193 - Unique Local IPv6 Unicast Addresses*.

[ITW] ITWissen - Das große Online-Lexikon für Informationstechnologie. *IPng-Protokolle*. Zugriffsdatum: 28.11.2011, http://www.itwissen.info/definition/lexikon/IP-next-generation-IPnG-IPnG-Protokolle.html.

[J.A07] J.Abley and P.Savolo and G. Neville-Neil (Dezember, 2007): *RFC5095 - Decreption of Type 0 Routing-Headers in IPv6*.

[Joh99] D. Johnson, S. Deering (März, 1999): *RFC2526 - Reserved IPv6 Subnet Anycast Addresses*.

[Kap11] Reiko Kaps (2011): *Netfilter-Entwickler arbeiten an NAT für ip6tables. Newsmeldung Heise Netze*. Zugriffsdatum: 28.11.2011, http://www.heise.de/netze/meldung/Netfilter-Entwickler-arbeiten-an-NAT-fuer-ip6tables\-1384952.html.

[Ken05a] S. Kent (Dezember, 2005): *RFC4302 - IP Authentication Header*.

[Ken05b] S. Kent (Dezember, 2005): *RFC4303 - IP Encapsulating Security Payload (ESP)*.

[Klo05] Andreas Kloeber (2005): *Seminar User Mode Linux - RIP Network Laboratory*.

[Mah07] Peter Mahlmann, Christian Schindelhauer (2007): *Peer-to-Peer-Netzwerke: Algorithmen und Methoden*. Springer-Verlag.

[Mal97] G. Malkin (Januar, 1997): *RFC2081 - RIPng Protocol Applicability Statement*.

[Mal98] G. Malkin (November, 1998): *RFC2453 - RIP Version 2*.

[Per11] C. Perkins, D. Johnson, J. Arkko (Juli, 2011): *RFC6275 - Mobility Support in IPv6*.

[Tha02] D. Thaler, B. Haberman (August, 2002): *RFC3306 - Unicast-Prefix-based IPv6 Multicast Addresses*.