

Universität Koblenz-Landau

Fachbereich 4

Institut für Informatik

Arbeitsgruppe Rechnernetze

Installation des Quagga RMTI Daemons auf einem Linksys WRT54G

Studienarbeit

Januar 2011

Bearbeiter: Josef Pennig
Matrikelnummer: 119920077
Betreuer: Prof. Dr. Ch. Steigner
Dipl. Inf. F. Bohdanowicz

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die vorliegende Arbeit selbständig und mit keinen anderen als den angegebenen Quellen und Hilfsmitteln verfasst zu haben. Alle aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet. Die Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Koblenz, den 21.01.2011

.....

Josef Pennig

Inhaltsverzeichnis

1. Einleitung	7
1.1. Motivation.....	8
1.2. Aufbau der Studienarbeit.....	8
2. Linksys-Modelle WRT54G/GL/GS	9
2.1. Entwicklungshistorie und Hardware-Spezifikation.....	9
2.2. Netzwerkarchitektur.....	12
3. Installation von OpenWRT	15
3.1. Installation von OpenWRT via TFTP.....	15
3.1.1. Installation von OpenWRT via AFTP.....	17
3.2. Installation von OpenWRT via MacTFTP.....	18
3.3. Installation von OpenWRT via Windows XP.....	18
3.4. Rettungsmodus des WRT54G/GL/GS.....	19
4. RIP/RMTI und Counting to Infinity	20
4.1. Das Routing Information Protocol.....	20
4.2. Das Counting-to-Infinity Problem.....	21
4.3. RIP with Metric based Topology Investigation.....	24
5. Konfiguration und Paketmanagement in OpenWRT	25
5.1. Zugriff auf den WRT54G/GL/GS.....	25
5.2. Konfiguration des Netzwerks.....	26
5.3. Installation der Quagga-Routing-Suite.....	30

6. Testszenario der Linksysrouter mit Quagga RMTI.....	31
6.1. Testszenario.....	31
6.2. Gegenüberstellung von RIP und RMTI.....	32
7. Fazit und Ausblick.....	35
Anhang A: Begriffe und Terminologie.....	36
Anhang B: Netzwerkkonfigurationsdateien.....	38
Anhang C: XML-Szenario für XTPeer	48

Abbildungsverzeichnis

Abbildung 2.2.1: Boardflags und Prozessorinfo.....	12
Abbildung 2.2.2: Blockdiagramm des WRT54GL.....	13
Abbildung 4.2.1: Beispiel Y-Szenario.....	22
Abbildung 4.2.2: Beispiel Y-Szenario mit CTI.....	23
Abbildung 5.2.1: Topologie des Szenarios.....	29
Abbildung 6.1.1: Demonstrationsplattform.....	33
Abbildung 6.2.1: Topologie des Testnetzwerkes.....	33
Abbildung 6.2.2: XTPeer-Graph von Szenario ohne RMTI.....	33
Abbildung 6.2.3: XTPeer-Graph von Szenario mit RMTI.....	34

Tabellenverzeichnis

Tabelle 2.1.1: Hardwarekonfiguration der WRT54G-Serie.....	10
Tabelle 2.1.2: Hardwarekonfiguration der WRT54GL-Serie.....	10
Tabelle 2.1.3: Hardwarekonfiguration der WRT54GS-Serie.....	11
Tabelle 3.4.1: Fehlerzustände nach Flash-Vorgang.....	18
Tabelle 5.2.1: Port-Nummern und Nummerierung.....	18
Tabelle 6.1.1: Schnittstellenbelegung.....	32

Kapitel 1

Einleitung

An der Universität Koblenz wurde eine Erweiterung zum RIP (Routing Information Protocol) entwickelt, der RMTI (RIP with Metric based Topology Investigation), dessen Einsatz zur Lösung des Counting-to-Infinity-Problems beitragen soll. Eine lauffähige Implementierung des RMTI Algorithmus auf Basis des RIP Daemons der Quagga Software Routing Suite, lag zu Beginn der Studienarbeit bereits vor.

Der Linksys¹ WRT54G, bzw. die in der Folge entwickelten Nachfolger WRT54GS und WRT54GL wurden als Testgeräte wegen ihrer Linux-Tauglichkeit gewählt. Entscheidend war dabei der öffentliche Zugang zum Quellcode der Router, der aber erst 2003 durch die Androhung rechtlicher Schritte ermöglicht wurde.

Linksys hatte durch die GPL² geschützten Code in seiner Firmware verwendet. Bis zur Firmware-Version 4.0 des Routers stehen damit die Endgeräte unter dem Schutz der GPL. Erst ein monatelang andauernder Appell brachte Linksys zum Einlenken und zur Veröffentlichung des Quelltextes, mit Ausnahme der Treiber für die WLAN-Schnittstelle. Die Veröffentlichung des Quellcodes hatte zur Folge, dass sich eine beträchtliche Anzahl von alternativen Firmware-Versionen entwickeln konnte, unter anderem das OpenWRT-Projekt, aber auch andere wie z.B. DD-WRT³, FreeWRT⁴, HyperWRT⁵, Talisman⁶ und Tomato⁷. Als Grundlage der Arbeit dient das OpenWRT-Projekt.

¹ <http://www.linksys.com/>

² GNU oder GPL - Die General Public License ist eine von der Free Software Foundation herausgegebene Lizenz, mit Copyleft für die Lizenzierung freier Software. Die Software bleibt dadurch für Entwickler frei und unterliegt keinen Lizenzunterwerfungen.

³ <http://www.dd-wrt.de/>

⁴ <http://freewrt.de/trac/>

⁵ <http://www.hyperwrt.org/>

⁶ <http://sveasoft.com/talismanbasic>

⁷ <http://www.polarcloud.com/tomato>

OpenWRT ist ein auf Linux basierendes, quelloffenes Betriebssystem für den Router WRT54G der Firma Linksys und nahezu uneingeschränkt kompatibel zu dessen Nachfolgern, bis einschließlich Version 4.0. Die Möglichkeit zur Konfiguration der Router mit OpenWRT gehen weit über die vom Hersteller mitgelieferte original Firmware hinaus.

1.1 Motivation und Ziel

Das Ziel dieser Studienarbeit war die Installation des RMTI-Daemons auf einem, bzw. mehreren Linksys WRT54G-Routern der AG Rechnernetze. Anschließend sollten typische Netzwerksituationen aufgebaut und das Verhalten der Router untersucht werden. Zur Veranschaulichung sollte eine Plattform mit WRT54G-Routern entstehen, um verschiedene Topologien aufbauen zu können.

1.2 Aufbau der Studienarbeit

Zunächst werden in Kapitel 2 die Router in ihrer Entwicklungshistorie und verschiedenen Modellen etwas näher beleuchtet. Danach wird auf das OpenWRT-Projekt und dessen Entwicklung eingegangen. Es folgt die Installation der Firmware, sowie der nötigen Pakete. Anschließend wird das Routing Information Protocol (RIP), mit dem Counting-to-Infinity Problem beschrieben und die Verbesserung bzw. Lösung des Problems mit dem RMTI. Im letzten Kapitel wird dann die Plattform zur Demonstration mit den Routern vorgestellt und ein vollständig aufgebautes Szenario erläutert.

Kapitel 2

Linksys-Modelle WRT54G/GL/GS

In diesem Kapitel wird die Hardware, bzw. die Entwicklungshistorie der Hardware von 2001 bis 2010 beschrieben, spezifiziert und die Unterschiede der einzelnen Versionen vorgestellt.

2.1 Entwicklungshistorie und Hardware-Spezifikation

OpenWRT ist ein alternatives Betriebssystem für sog. CPE⁸-Router. Die von verschiedenen Herstellern wie z.B. Linksys, Allnet, Asus, Belkin, Buffalo, Microsoft, Netgear und Siemens vorinstallierten Betriebssysteme können unter bestimmten Voraussetzungen durch ein „embedded Linux⁹“ ersetzt werden. Der Vollständigkeit halber sei noch erwähnt, dass auch an einer Lösung für die weit verbreiteten Geräte von AVM¹⁰, d.h. der Fritz!Box gearbeitet wird. Der große Vorteil liegt dabei in der Erweiterung der Funktionalität, gegenüber den ursprünglichen Softwareversionen der Hersteller.

OpenWRT wurde zunächst nur über die Kommandozeile über den SSH-Dienst gesteuert und bekam mit der Vollendung der Version Kamikaze 8.09 die Oberfläche LuCI, zur Seite gestellt. Auf die beiden Webinterfaces wird jedoch in der Folge nicht weiter eingegangen, sie seien der Vollständigkeit halber aber erwähnt.

Die Firma Linksys, die seit 2001 ein Tochterunternehmen von Cisco ist, erschien mit dem WRT54G Version 1.0 Ende 2002 auf dem Endverbrauchermarkt.

⁸ Der Begriff Customer Premises Equipment (CPE) bezeichnet ein Teilnehmer-Endgerät in einem Computernetz, einem Telefonnetz oder bei Telefonanlagen.

⁹ Als Embedded Linux bezeichnet man ein eingebettetes System mit einem auf einem Linux-Kernel basierenden Betriebssystem

¹⁰ <http://www.avm.de/de/>

Das Gerät beinhaltet einen integrierten Vier-Port-Switch,(10/100), sowie eine WLAN-Schnittstelle und bot Einstellungsmöglichkeiten via Webinterface, so dass auch unerfahrene Nutzer leichter Einstellungen vornehmen konnten.

In den folgenden Versionen wurde die Hardware des Routers immer wieder abgeändert und ab 2005 mit der Firmware VxWorks ausgestattet, so dass wegen der geringeren Hardwareanforderungen der Hauptspeicher von 32 MB auf 16 MB und die Flash-Speichergröße von 8 MB auf 2 MB reduziert wurden. Die Geräte ab der Version 5.0 sind daher für OpenWRT nicht mehr geeignet (s. durchgestrichene Einträge in den Tabellen). Zur eindeutigen Identifikation eines vorliegenden Geräts kann der auf der Unterseite des Gerätes vermerkte vierstellige Anfang der Seriennummer verwendet werden, beispielsweise CL7B1GC05683 für einen WRT54GL (s. Tabelle 2.1.2) [AP07].

Die folgenden Tabellen 2.1.1 für die Modelle der G-Serie, 2.2 für die Modelle der GL-Serie und 2.3 für die Modelle der GS-Serie geben einen Überblick über die verbaute Hardware der Router seit dem ersten Erscheinen 2001.

Modell / Seriennummer	Mainboard / Taktfrequenz	RAM / Flash-Speicher
Vers. 1.0 / CDF0 oder CDF1	BCM4702 / 125 MHz	16 MB / 4 MB
Vers. 1.1 / CDF2 oder CDF3	BCM4702 / 125 MHz	16 MB / 4 MB
Vers. 2.0 / CDF5	BCM4712 / 200 MHz	16 MB / 4 MB
Vers. 2.2 / CDF7	BCM4712 / 200 MHz	16 o. 32 MB / 4 MB
Vers. 3.0 / CDF8	BCM4712 / 200 MHz	16 MB / 4 MB
Vers. 3.1 / CDF9	BCM4712 / 200 o. 216 MHz	16 MB / 4 MB
Vers. 4.0 / CDFA	BCM5352 / 200 MHz	16 MB / 4 MB
Vers. 5.0 / CDFB	BCM5352 / 200 MHz	8 MB / 2 MB
Vers. 6.0 / CDFC	BCM5352 / 200 MHz	8 MB / 2 MB
Vers. 7.0 / CDFE	BCM5352 / 200 MHz	8 MB / 2 MB
Vers. 8.0 / CDFF oder CDFG	BCM5352 / 200 MHz	8 MB / 2 MB

Tabelle 2.1.1: Hardwarekonfiguration der WRT54G-Serie

Die in Tabelle 2.1 gelb markierte Konfiguration der Vers. 4.0 des WRT54G ist baugleich mit der Version des WRT54GL Rev. 1.0 (gelb markiert in Tabelle 2.2).

Modell / Seriennummer	Mainboard / Taktfrequenz	RAM / Flash-Speicher
Vers. 1.0 / CL7A	BCM5352 / 200 MHz	16 MB / 4 MB
Vers. 1.1 / CL7B oder CL7C	BCM5352 / 200 MHz	16 MB / 4 MB
Vers. 1.1 / CO61	BCM5352 / 200 MHz	32 MB / 8 MB

Tabelle 2.1.2: Hardwarekonfiguration der WRT54GL-Serie

Die in Tabelle 2.2 grün markierte Konfiguration des WRT54GL war die zum Zeitpunkt der Arbeit und darüber hinaus durchgängig im Handel angebotene Version. Die Version 1.1 mit Seriennummer CO61 ist ein Gerät, dass speziell für T-Mobile gebaut wurde und von der Grundausstattung an RAM und Flash-Speicher sehr interessant erschien, jedoch leider nicht mehr verfügbar war.

Modell / Seriennummer	Mainboard / Taktfrequenz	RAM / Flash-Speicher
Vers. 1.0 / CGN0 oder CGN1	BCM4712 / 200 MHz	32 MB / 8 MB
Vers. 1.0 / CGN2	BCM4712 / 200 MHz	32 MB / 8 MB
Vers. 2.0 / CGN3	BCM4712 / 200 MHz	32 MB / 8 MB
Vers. 2.1 / CGN4	BCM4712 / 200 MHz	32 MB / 8 MB
Vers. 3.0 / CGN5	BCM5352 / 200 MHz	32 MB / 8 MB
Vers. 4.0 / CGN6	BCM5352 / 200 MHz	16 MB / 4MB
Vers. 5.0 / CGN7	BCM5352 / 200 MHz	16 MB / 2MB
Vers. 5.1 / CGN8	BCM5352 / 200 MHz	16 MB / 2MB
Vers. 6.0 / CGN9	BCM5352 / 200 MHz	16 MB / 2MB
Vers. 7.0 / CGNA oder CGNB	BCM5354 / 200 MHz	16 MB / 2MB

Tabelle 2.1.3: Hardwarekonfiguration der WRT54GS-Serie

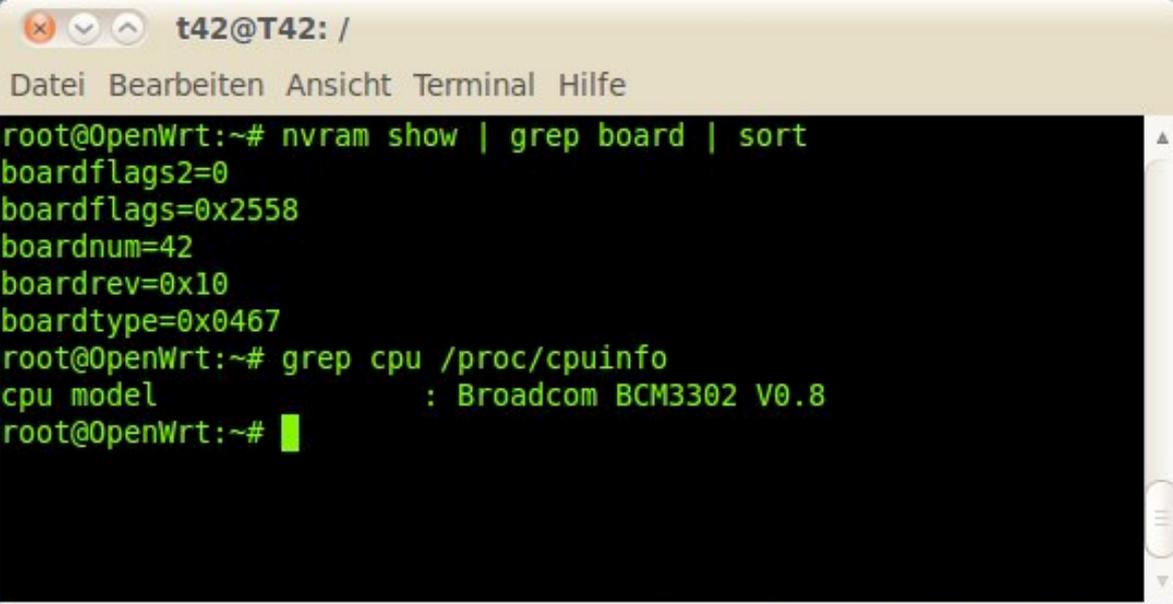
Die Geräte ab der Version 5.0 sind für eine Installation von OpenWRT nicht mehr geeignet, da in erster Linie der Flash-Speicher nur noch die Größe von 2 MB hat.

Alle Geräte sind mit einem Broadcom MIPS-Prozessor ausgestattet, ähnlich wie sie Cisco in seinen Switches, bzw. Sony in seinen Spielkonsolen Playstation und Playstation 2 verwendet. In der Regel handelt es sich um Prozessoren BCM47xx und BCM5352.

Dies sei an dieser Stelle erwähnt, da Linux-Software im Allgemeinen für Intel x86 Plattformen entwickelt wird, daher benötigt man eine Software zum „Cross-Compiling¹¹“, um die Software auf den Geräten zum laufen zu bekommen, oder man bekommt sie bereits fertig vorkompiliert [AP07].

2.2 Netzwerkarchitektur

Zur Überprüfung der vorhandenen Hardware können neben dem Beginn der Seriennummer folgende Auskünfte eingeholt werden, wie z.B. Board-Info und CPU-Modell.

A screenshot of a terminal window titled 't42@T42: /'. The terminal shows the following commands and output:

```
root@OpenWrt:~# nvram show | grep board | sort
boardflags2=0
boardflags=0x2558
boardnum=42
boardrev=0x10
boardtype=0x0467
root@OpenWrt:~# grep cpu /proc/cpuinfo
cpu model          : Broadcom BCM3302 V0.8
root@OpenWrt:~#
```

Abbildung 2.2.1: Boardflags und Prozessorinfo

¹¹ Cross-Compiling ist der Vorgang, beim dem ein Compiler auf einem System, bzw. einer bestimmten Hostplattform läuft und sog. Objektdateien oder ausführbare Programme für andere Plattformen erzeugt.

Der WRT54G-Router beinhaltet immer einen Ethernet-Switch, ein VLAN¹² und eine Bridge. Der standardmäßige 5-Port-Switch teilt sich dabei in den Grundeinstellungen in zwei VLAN's auf (VLAN 0 und VLAN 1).

Die WLAN-Schnittstelle, die in dieser Arbeit keine Berücksichtigung fand, sei der Vollständigkeit halber hier erwähnt und wird über eine Bridge an den Switch angebunden.

In dem Blockdiagramm in Abbildung 2.2 wird eine komplette Architektur eines WRT54GL skizziert.

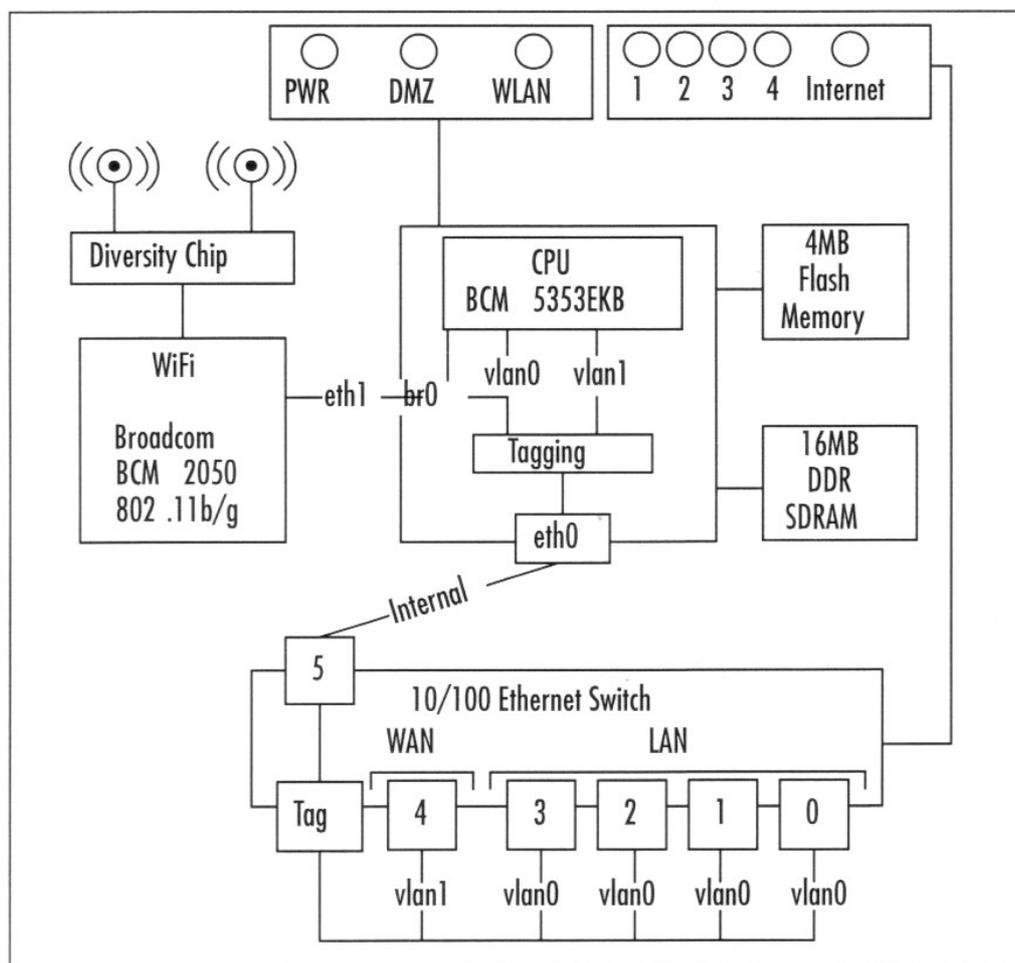


Abbildung 2.2.2: Blockdiagramm des WRT54GL

¹² Virtual Local Area Network - logisches Teilnetz innerhalb eines Switches oder eines physischen Netzwerks

Das VLAN0 referenziert dabei die LAN-Ports 0,1,2,3 die physikalisch als Ports 1,2,3,4 ausgewiesen sind. VLAN1 referenziert den WAN-Port.

Kapitel 3

Installation von OpenWRT

In diesem Kapitel wird das Aufspielen der Images auf den Linksys-Routern beschrieben. Die verschiedenen Modellserien unterscheiden sich dabei nicht im Vorgehen, lediglich die Auswahl des Images war zu beachten, da der Großteil der Modelle zwar WRT54GL waren, es lagen aber auch ein WRT54G sowie ein WRT54GS vor, die sich in den Prozessor- und Chipsatztypen unterschieden.

3.1 Installation von OpenWRT via TFTP

An dieser Stelle sei natürlich angemerkt, dass eine Installationsfunktion eines Images, via Webinterface¹³ durch Linksys bereits zur Standardfunktionalität aller Modelle gehört.

Da die Installation von selbst kompilierten Images jedoch nicht immer reibungslos funktionierte, wird die Installation per TFTP¹⁴ empfohlen. Der Vorteil liegt jedoch in der Reparatur eines defekten Systems, welches sich via TFTP stets noch erreichen lässt.

Zunächst ist der Router vom Strom zu trennen und der TFTP-Client zu starten. Danach ordnet man dem Router eine IP-Adresse zu, dort ist zunächst 192.168.1.1 voreingestellt.

Mit „set mode“ wird dann „binary“ als Wert eingetragen und dann mit einem „resend“ bestätigt.

Zuletzt wird mittels des Befehls „put“ die Datei an den Router gesendet. In diesem Zusammenhang gibt es einige Punkte zu beachten, die die Installation und den folgenden Anmeldeprozess erleichtern können.

¹³ grafische Oberfläche zur Installation eines Images, im Falle von OpenWRT z.B. „LuCI“ (Lua Configuration Interface)

¹⁴ Trivial File Transfer Protocol - Dateiübertragungsprotokoll zum Lesen und Schreiben von Dateien, ohne Rechtevergabe

Als wichtige Punkte wären zu nennen:

- a) der Router rebootet nach Beendigung des Flashvorgangs automatisch, erkennbar an der blinkenden DMZ LED
- b) die IP-Adresse wird auf den Default-Wert 192.168.1.1 zurückgesetzt
- c) zur Überbrückung der *boot-wait Period* empfiehlt sich ein Switch zwischen Router und Computer

Generell sollten alle Vorgänge als Root-Nutzer vorgenommen werden. Hier einmal die übliche Verfahrensweise zum Flashvorgang über ein Kommandozeileninterface:

```
-----  
tftp 192.168.1.1 // Verbindungsaufbau zum Router  
set mode binary  
rexmt 1 // Aufrecherhaltung des Sendens der Daten  
timeout 60 // Setzen des Timeouts auf 60 Sek.  
trace  
Packet tracing on  
tftp> put openwrt-xxx-x.x-xxx.bin  
-----
```

Die Einzeleingaben können auch zur Vereinfachung in einer Zeile eingegeben werden, die beispielsweise wie folgt aussehen könnte:

```
echo -e „binary\nrexmt 1\ntimeout 60\ntrace\nput openwrt-xxx.x-xxx.bin\n| tftp 192.168.1.1
```

3.1.1 Installation von OpenWRT via ATFTP

ATFTP¹⁵ sei an dieser Stelle noch kurz erwähnt, da es im Zusammenhang mit „Netkit¹⁶“ bereits eine gewisse Verbreitung fand.

Für die Installation der Router wurde es als Alternative mit hier aufgenommen und unterscheidet sich zunächst nur durch die Syntax beim Datenübertragungsprozess auf die Router.

Der Flashvorgang über ein Kommandozeileninterface lautet:

```
-----  
atftp  
connect 192.168.1.1  
mode octet  
trace  
timeout 1  
put openwrt-xxx-x.x-xxx.bin  
-----
```

Die Einzeleingaben können auch hier zur Vereinfachung in einer Zeile eingegeben werden, die beispielsweise wie folgt aussehen könnte:

```
atftp --trace --option timeout 1 --option mode octet --put --local-file  
openwrt-xxx.x-xxx.bin 192.168.1.1
```

¹⁵ Advanced TFTP - (atftp source code: <http://downloads.openwrt.org/sources/atftp-0.7.tar.gz>)

¹⁶ Netzwerkemulator der auf Open-Source-Software basiert und Bestandteil gängiger Linux-Distributionen ist

3.2 Installation von OpenWRT via MacTFTP

Die Installation von einem Mac aus funktioniert mittels TFTP analog zu der Beschreibung aus 3.1. Es besteht jedoch eine zusätzliche Alternative, mittels des Paketes MacTFTP¹⁷ eine Installation vorzunehmen. Die Besonderheit liegt dabei in dem Problem der Deaktivierung der Ethernetkarte nach der Abschaltung des Routers, was eine manuelle Konfiguration der Netzwerkkarte erforderlich macht.

Die Vorgehensweise sieht dann wie folgt aus:

- a) Installation des MacTFTP-Paketes
- b) Wahl des Send-Buttons
- c) Adresswahl (also in der Regel 192.168.1.1)
- d) Passworteingabe (falls gesetzt)
- e) Auswahl des Imagefiles openwrt-xxx-x.x-xxx.bin
- f) Konfigurieren der Netzwerkkarte - Auswahl manuell
- g) Geschwindigkeitsauswahl mit 10 BaseT/UTP
- h) Einschalten der Option Full-Duplex

Um eine manuelle Konfiguration zu umgehen, kann auch wie bereits beschrieben ein Hub bzw. Switch zwischen Host und Router geschaltet werden.

3.3 Installation von OpenWRT unter Windows XP

Die Installation eines Images unter Windows XP erfolgt am einfachsten mittels des kompatiblen TFTP-Clients¹⁸. Bei den durchgeführten Aufspielungen von Images verlief nach Abschalten der Firewall alles reibungslos.

¹⁷ Macintosh compatible Version des Trivial File Transfer Protocols - (<http://www.mactechnologies.com/index.php?page=downloads#mactftpclient>)

¹⁸ Trivial File Transfer Protocol-Client - (<http://martybugs.net/wireless/openwrt/flash.cgi>)

Die Installationen erfolgten nach nachstehendem Schema:

- a) Öffnen der Kommandozeile (cmd)
- b) `ping -t -w 12 192.168.1.1`
- c) Aufbau der Verbindung über `tftp - 192.168.1.1 PUT openwrt-xxx-x.x-xxx.bin`, diesen Befehl jedoch zunächst nicht bestätigen
- d) Anschließen des Routers
- e) Bestätigung mittels Enter nach der Ausgabe Hardware-Error

3.4 Rettungsmodus des WRT54G/GL/GS

Die Flashvorgänge der einzelnen Router haben vereinzelt zu Fehlern geführt, u.a. durch die Auswahl eines falschen Images. Der Router ist dann, auf den bisher beschriebenen Wegen, nicht mehr erreichbar. Linksys bietet in seinen Geräten jedoch nach einem Reset die Möglichkeit, innerhalb von fünf Sekunden auf den Router zuzugreifen, mittels TFTP. In diesem Zustand, kurz nach dem Reset ist der Router über die Adresse 192.168.1.1 erreichbar, unabhängig von etwaigen Voreinstellungen des Nutzers. Zur Unterscheidung der einzelnen Fehler-Szenarien daher hier eine kurze Übersicht der zur Rettung erforderlichen Maßnahmen:

Fehlergrad	Beschreibung	Abhilfe
1	Power-LED blinkt sehr schnell und nach 20 Sek. permanent, DMZ-LED leuchtet nicht, d.h.i.d.R. Bootloader und Kernel sind intakt	Mac-Filter und Firewall richtig einstellen, bzw. NVRAM zurücksetzen
2	Power-LED blinkt sehr schnell und DMZ-LED leuchtet 5 Sek., d.h.i.d.R. Bootloader intakt, Kernel bzw. Firmware defekt	erneute Installation über TFTP
3	Power-LED blinkt sehr schnell auch nach 2 Min., DMZ-LED leuchtet nicht, d.h.i.d.R. Bootloader defekt	Hardware-Reset durchführen, d.h. Öffnen des Routers erforderlich

Tabelle 3.4: Fehlerzustände nach Flash-Vorgang

Kapitel 4

RIP/RMTI und Counting-to-Infinity

4.1 Das Routing Information Protocol

Das Routing Information Protokoll (RIP) gehört zur Klasse der Distanzvektorprotokolle, die auch als Bellmann-Ford Protokolle bezeichnet werden, benannt nach Richard Bellmann und Lester Ford, deren Algorithmus die Berechnung des kürzesten Weges, ausgehend von einem Startknoten, in einem kantengewichteten Graphen ermöglichte [BEL57].

Die auf dem Distanzvektoralgorithmus basierenden Routing-Protokolle, zu denen RIPv1 und RIPv2 zählen, bilden die eine Klasse von dynamischen Routing-Protokollen, im Gegensatz zu den so genannten Link-State Protokollen. Beide sind über RFC's¹⁹ standardisierte, dynamische Routing-Algorithmen [TAN03].

RIPv1 wurde erstmals 1988 im RFC 1058 und die aktuelle RIPv2 im RFC 2453 spezifiziert.

Ein RIP-Router verwaltet in seiner Routing-Tabelle Routen aus dem Ziel-Subnetz, der Distanz zum Ziel-Subnetz (Metrik) und dem ersten Hop (Nachbarn) zum Ziel-Subnetz.

Als sogenannte Hop-Count-Metrik bezeichnet man beim RIP daher die Anzahl der Hops, bzw. RIP-Router, auf der Strecke Router zu Ziel-Subnetz.

Dabei wird jeder Router versuchen, immer die kürzeste Route, nämlich die mit der besten Metrik zum Ziel-Subnetz zu wählen. Diese kürzeste Route ist zusätzlich im Router gespeichert.

Alternative Routen mit einer höheren Metrik werden allerdings nicht gespeichert, so dass im Falle des Ausfalls einer Verbindung zu einem Subnetz, die Metrik der eingetragenen

¹⁹ RFC: Request for Comments: Offene Dokumentenreihe, in der Standards und Empfehlungen zu technischen Entwicklungen spezifiziert werden

Route auf RIP-Inifinity gesetzt wird. Dies entspricht in der Standard-Spezifikation von RIP dem Wert 16.

Anmerkung: Zwischen einem RIP-Router und seinem Ziel-Subnetz können somit maximal 15 weitere RIP-Router liegen [BOH08].

Die Kommunikation der RIP-Router untereinander beinhaltet zwei verschiedene Nachrichtentypen, die Request-Nachrichten und die Response-Nachrichten. Bei den Request-Nachrichten fragt ein Router alle Informationen aus der Tabelle seines Nachbarn an. Diese Anfrage wird gewöhnlich nur nach dem zuschalten eines Routers an das Netzwerk gestartet. Eine spezielle Anfrage für einzelne Subnetze ist in der RIP-Spezifikation auch möglich, die Verwertung aber nur zur Netzwerkdiagnose zu gebrauchen.

Die Response-Nachrichten beinhalten nun jene Routing-Informationen, die ein Router beim Updateprozess erhält. Diese Nachrichten enthalten die Informationen aus der Routing-Tabelle und werden in regelmäßigen Abständen an alle Nachbarrouter gesendet.

Interessant sind dabei vor allem jene Updates, die nach Änderungen an den Routing-Tabellen versendet werden. Diese werden als Triggered-Updates bezeichnet.

Eine Response-Nachricht ist also eine Antwort auf eine Anfrage, oder auf eine Request-Nachricht.

4.2 Das Counting-to-Infinity Problem

Als Folge einer ungünstigen Sendereihenfolge der periodischen Updates zwischen den einzelnen RIP-Routern kann das Counting-to-Infinity Problem auftreten.

Wenn also bei einem Updateprozess eine aktuellere Routinginformation über den Wegfall einer Route und die damit verbundene Unerreichbarkeit des Subnetzes von einer älteren Routing-Information, mit der alten Erreichbarkeit des Subnetzes überschrieben wird, dann kann es zu einer Routingschleife kommen.

In Abbildung 4.2 wird das Counting-to-Infinity Problem anhand eines Beispiels gezeigt. Wie es dann zum CTI²⁰ kommt, wird in der Folge in einzelnen Schritten erläutert. Als Szenario dient zunächst ein einfaches Y-Szenario.

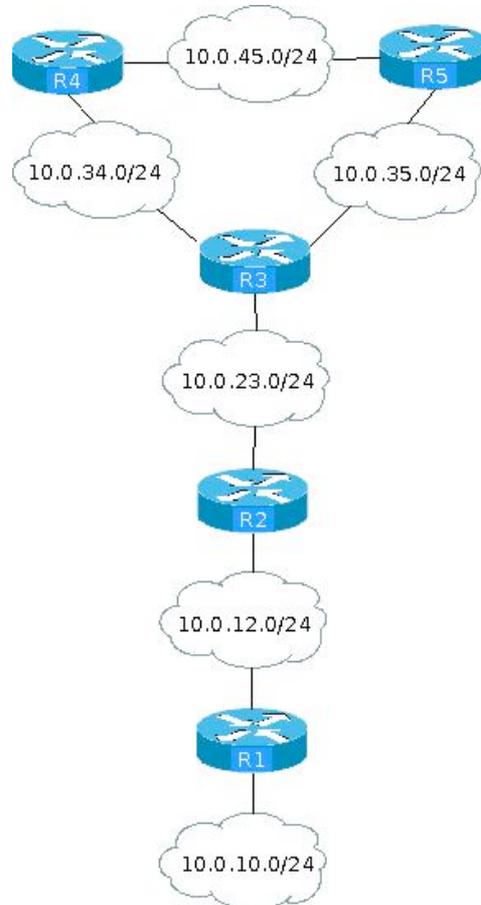


Abbildung 4.2.1: Beispiel Y-Szenario

Das Subnetz 10.0.10.0/24 wird in jedem Router als Zielnetz gespeichert, d.h. alle Router verfügen über eine gültige Route. Die beiden Router R4 und R5 kennen jeweils eine Route und zeigen sich diese gegenseitig, sie nutzen aber die kürzere über den Router R3.

Wenn nun das Netz 10.0.12.0/24, bzw. die Verbindung zwischen R1 und R2 ausfällt, erreicht der Timeout-Timer von R2 den Wert 16 (RIP-Infinity) und behandelt das Netz 10.0.10.0/24 von da an, als unerreichbar. Die Information verbreitet sich nun weiter an R3 und über R3 auch an R4 und R5. In diesem Fall übernimmt zunächst der Router R4 die Information von R3, während R5 das Routing-Update wegen der verbindungslosen

²⁰ Count-to-Infinity Problem

Kommunikation über UDP²¹ nicht bekommt. In der Routing-Tabelle von R5 bleibt also die alte Route über R3 gespeichert, die in der Folge auch an R4 weitergegeben wird.

Die von R5 als noch gültig angenommene Route wird dann auch an R3 weitergegeben, der wiederum leitet diese weiter an R2 und R5. Die Routing-Schleife entsteht genau zu dem Zeitpunkt der Weitergabe des Updates an R4 und der Eintragung in dessen Routing-Tabelle.

Die Router R3, R4 und R5 senden in der Folge die falsche Information untereinander weiter, so dass die Metrik bis zum höchstmöglichen Weg von 16 hochgezählt wird. Mit Erreichen des Wertes 16, gilt dann das Subnetz 10.0.10.0/24 als unerreichbar und die Routing-Schleife wird unterbrochen.

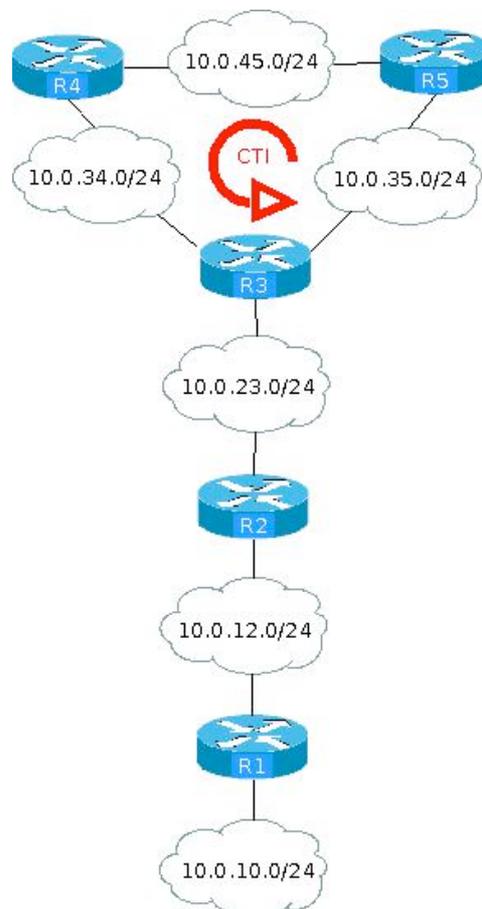


Abbildung 4.2.2: Beispiel Y-Szenario mit CTI

²¹ User datagram Protocol

Ein so entstandene Routing-Schleife führt bei den beteiligten Routern zu einer Mehrbelastung und kann mit Netzausfällen, sowie dem Ausfallen einzelner Netzwerkdienste einhergehen.

4.3 RIP with Metric based Topology Investigation

Die Idee hinter dem RMTI-Algorithmus basiert auf dem Erkennen von speziellen Pfaden zwischen Routern und Subnetzen einer Netzwerktopologie und der Möglichkeit diese voneinander zu unterscheiden. Durch das Ansammeln von Informationen über eine bestimmte Netzwerktopologie wird es für einen Router möglich, eine Schleife zwischen zwei seiner Netzwerkschnittstellen zu erkennen.

Eine ausführlichere Beschreibung findet sich erstmals bei [SCH99], der den RMTI als Lösung des CTI's vorstellt. In den Arbeiten von [KOC05] und [BOH08] werden die einzelnen Fälle die auftreten können näher beschrieben. Die in Kapitel 6 dargestellte Topologie, beschäftigt sich nur mit einer Y-Kombination des RMTI.

Kapitel 5

Konfiguration und Paketmanagement in OpenWRT

In diesem Kapitel wird die Konfiguration des Routers beschrieben und die Funktionsweise des Paketmanagers (OPKG²²). Anschließend werden einige nötige Pakete wie der RMTI und die Quagga-Suite installiert, die für die spätere Topologie benötigt werden.

5.1 Zugriff auf den WRT54G/GL/GS

Nach dem Flashvorgang und der Zuweisung einer IP-Adresse loggt man sich beim ersten Zugriff via Telnet auf dem Router ein. Der Benutzername lautet `root`; ein Passwort ist zunächst nicht vergeben. Der einfachen Erreichbarkeit und Konfiguration halber wurden alle Geräte mit den folgenden Zugangsdaten versehen:

Benutzername: `root`
Passwort: `admin`

Das Einloggen kann dann mittels `ssh` erfolgen und zwar durch Anmeldung als `root` am Router:

```
ssh root@192.168.1.1
```

Nach dem Einloggen wird zunächst die Firewall deaktiviert und zwar mittels:

```
/etc/init.d/firewall stop
```

²² Paketmanagement-System auf Basis von `ipkg`, findet Verwendung in verschiedenen Openembedded-Projekten und dem OpenWRT-Projekt

5.2 Konfiguration des Netzwerks

Die Einstellungen, der einzelnen Router werden in der `/etc/config/network` vorgenommen. Im folgenden Abschnitt befindet sich beispielhaft die Konfigurationsdatei von Router R5. Eine vollständige Einsicht, für alle Router findet sich im Anhang B.

Wie bereits in Kapitel 2 erwähnt (s. Abb. 2.2.2) sind bei den WRT54GL die fünf Netzwerkanschlüsse zu einem Switch zusammengeschaltet. Es besteht jedoch die Möglichkeit, einzelne VLAN's zu konfigurieren [AP07].

Innerhalb der Netzwerkkonfigurationsdateien gibt es verschiedene sog. Sektionen („config-sections“), in denen die Einstellungen für die einzelnen Geräte vorgenommen werden können. Diese bestehen aus den „option-statements“. Die in den folgenden Konfigurationsdateien verwendeten Typen von „config-sections“ sind „switch“, „switch_vlan“ und „interface“.

Damit nun ein oder mehrere VLAN's aktiviert werden können, muss zunächst durch die „Option enable=1“ der Switch des Routers aktiviert werden. Eine etwas verwirrende Besonderheit bei der Konfiguration von VLAN's auf den WRT54GL bildet die interne Vergabe der Port-Nummern, die im Gegensatz zur Nummerierung an den einzelnen Ports steht.

Port-Nummer	Nummerierung
0	4
1	3
2	2
3	1
4	Internet

Tabelle 5.2: Port-Nummernvergabe und Nummerierung

Alle Änderungen an den Dateien wurden mittels des Editors `vi`²³ vorgenommen.

```
1 config 'switch' 'eth0'
2     option 'enable' '1'           # Aktivieren des Router-Switches
3
4 config 'switch_vlan' 'eth0_0'
5     option 'device' 'eth0'
6     option 'vlan' '0'
7     option 'ports' '4 5'         # VLAN-Portzuweisung 4 und 5
8
9 config 'switch_vlan' 'eth0_1'
10    option 'device' 'eth0'
11    option 'vlan' '1'             # Namensgebung VLAN
12    option 'ports' '3 5'         # VLAN-Portzuweisung 3 und 5
13
14 config 'interface' 'loopback'   # Schnittstellenkonfiguration
15    option 'ifname' 'lo'
16    option 'proto' 'static'      # Zuweisung (dhcp oder static)
17    option 'ipaddr' '127.0.0.1'
18    option 'netmask' '255.0.0.0'
19
20 config 'interface' 'xtpeer'
21    option 'proto' 'static'
22    option 'ipaddr' '192.168.1.118' # IP-Adressen-Zuweisung
23    option 'netmask' '255.255.255.0' # Subnetzmaske
25    option 'defaultroute' '0'    # Standard-Route
26    option 'peerdns' '0'         # DNS-Server
```

²³ abgeleitet von „visual“, ist ein einfacher Texteditor in Linux, mit drei verschiedenen Arbeitsmodi, <http://de.wikipedia.org/wiki/Vi>

```
31 config 'switch_vlan' 'eth0_2'
32     option 'device' 'eth0'
33     option 'vlan' '2'
34     option 'ports' '2 5'
35
36 config 'switch_vlan' 'eth0_3'
37     option 'device' 'eth0'
38     option 'vlan' '3'
39     option 'ports' '1 5'
40
41 config 'switch_vlan' 'eth0_4'
42     option 'device' 'eth0'
43     option 'vlan' '4'
44     option 'ports' '0 5'
45
46 config 'interface' 'lan1'
47     option 'proto' 'static'
48     option 'ifname' 'eth0.1'
49     option 'ipaddr' '10.0.50.1'
51     option 'netmask' '255.255.255.0'
52     option 'defaultroute' '0'
53     option 'peerdns' '0'
54
55 config 'interface' 'lan2'
56     option 'proto' 'static'
57     option 'ifname' 'eth0.2'
58     option 'ipaddr' '10.0.45.2'
59     option 'netmask' '255.255.255.0'
60     option 'defaultroute' '0'
61     option 'peerdns' '0'
```

```

62 config 'interface' 'lan3'
63     option 'proto' 'static'
64     option 'ifname' 'eth0.3'
65     option 'ipaddr' '10.0.35.2'
66     option 'netmask' '255.255.255.0'
67     option 'defaultroute' '0'
68     option 'peerdns' '0'
69
70 config 'interface' 'lan4'
71     option 'proto' 'static'
72     option 'ifname' 'eth0.4'
73     option 'ipaddr' '10.0.15.2'
74     option 'netmask' '255.255.255.0'
75     option 'peerdns' '0'
76     option 'defaultroute' '0'

```

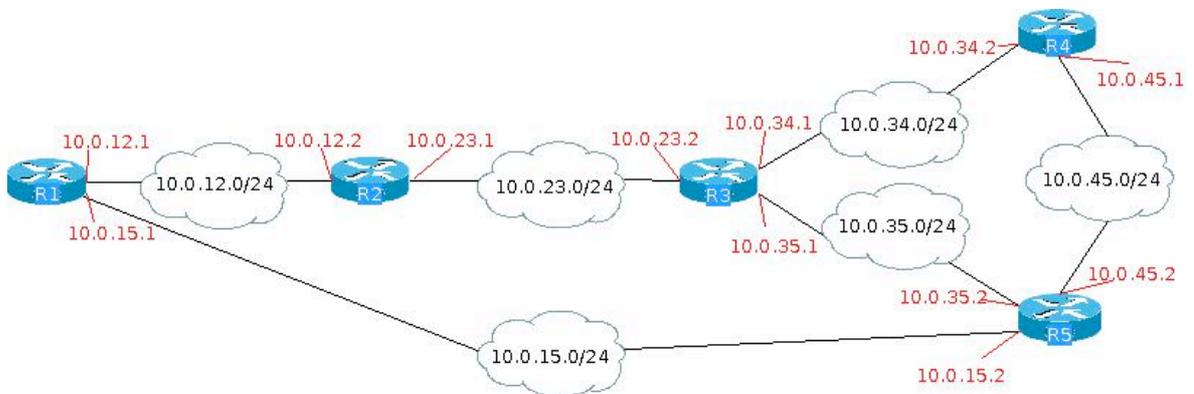


Abbildung 5.2.1: Topologie des Szenarios

Die Abbildung zeigt die Topologie ohne die Hosts an R1 bzw. R5. Die rot markierten Adressen sind die an den jeweiligen Schnittstellen konfigurierten Adressen, die schwarzen in „Wolken“ gezeigten stehen für die Netze zwischen den Routern.

5.3 Installation der Quagga-Routing-Suite

Die Quagga-Routing-Suite, die auf TCP/IP basiertes Routing unterstützt steht unter dem Schutz der GPL. Sie enthält verschiedene Routing-Dienstprogramme, für unterschiedliche Routing-Protokolle, wie z.B. RIP, OSPF und BGP.

Beim Quagga-Projekt handelt es sich um eine Weiterentwicklung des GNU-Zebra-Projektes von K. Ishiguro aus dem Jahr 1996 [BOH08], jedoch mit dem Ziel, eine breitere Unterstützung für Anwender zu ermöglichen.

Die Routing-Suite besteht aus dem RIP-Daemon „ripd“, der Routing über das RIP-Protokoll ermöglicht, dem „ospfd“ der OSPFv2-Routing ermöglicht, dem „bgpd“ der BGPv4-Routing ermöglicht, sowie dem Zebra-Daemon „zebra“, der die Schnittstelle zu vom Linux-Kernel verwalteten Routing-Tabelle bildet. Die Pakete für OSPF und BGP spielten für die Arbeit keine Rolle.

Wie bereits erwähnt, wird bei OpenWRT eine quelloffene Build-Umgebung verwendet, der OPKG-Paketmanager. Die Pakete werden für diesen Paketmanager gebaut und lagen in der Version 0.99.4 zu Beginn dieser Arbeit vor. Die Pakete konnten dann einfach installiert werden mittels:

1. Paket: `quagga-ripd_0.99.4-1_mipsel.ipk`

```
opkg install http://www.uni-koblenz.de/~vnuml/linksys/quagga-ripd_0.99.4-1_mipsel.ipk
```

2. Paket: `quagga-zebra_0.99.4-1_mipsel.ipk`

```
opkg install http://www.uni-koblenz.de/~vnuml/linksys/quagga-zebra_0.99.4-1_mipsel.ipk
```

3. Paket: `quagga_0.99.4-1_mipsel.ipk`

```
opkg install http://www.uni-koblenz.de/~vnuml/linksys/quagga_0.99.4-1_mipsel.ipk
```

Kapitel 6

TestszENARIO der Linksys-Router mit Quagga RMTI

In diesem Kapitel wird der Demonstrator bzw. die Testumgebung vorgestellt. Das Szenario wurde zu Labordemonstrationen und an Präsentationstagen der Universität Koblenz verwendet.

6.1 TestszENARIO

In Abbildung 6.1.1 sieht man zunächst die komplette Demonstrationsplattform. Zusätzlich waren für die Präsentationen zwei Laptop's angeschlossen, um die Veränderungen der TTL-Werte bei den jeweiligen Ping-Vorgängen nachvollziehen zu können.

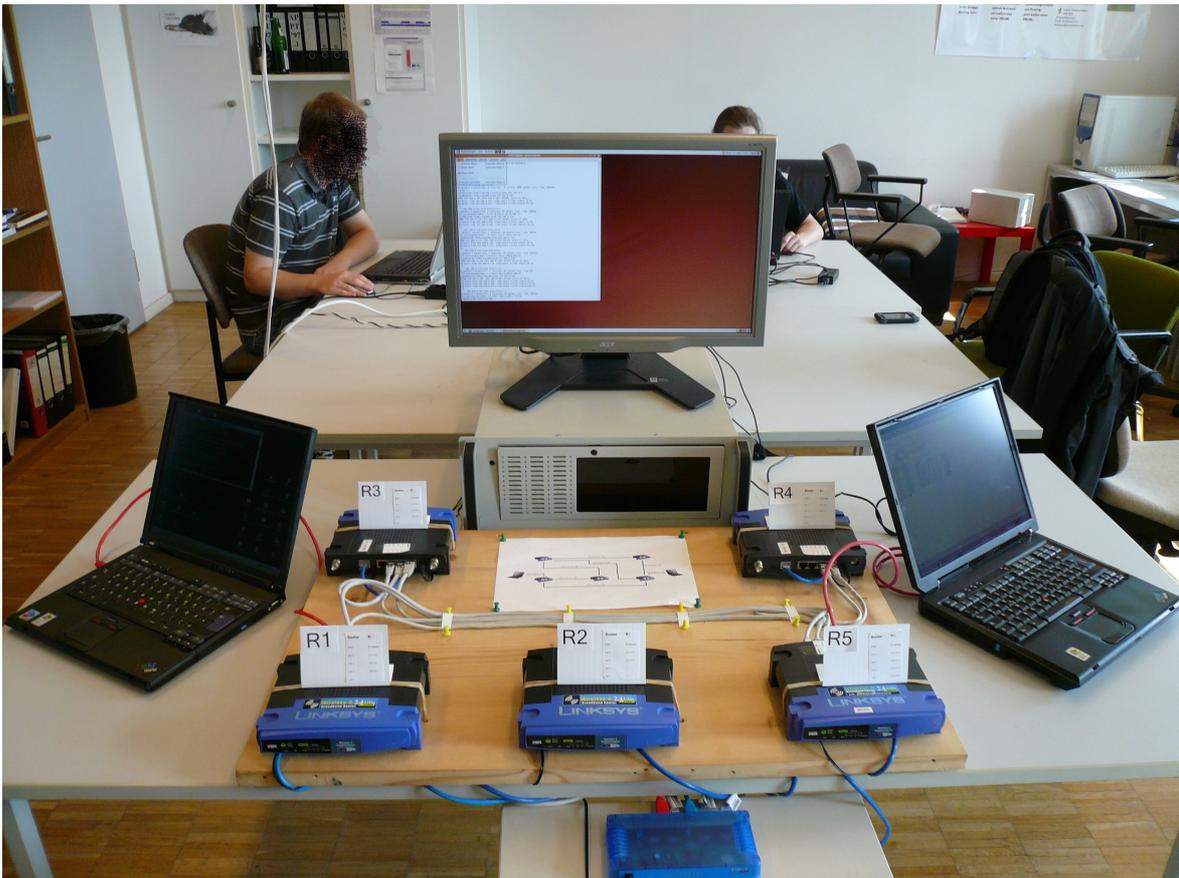


Abbildung 6.1.1: Demonstrationsplattform

Des Weiteren wurde ein zusätzlicher Host zur Veranschaulichung der Routing-Updates mit XTPeer²⁴ zugeschaltet. Die für das Szenario erforderliche XML-Datei befindet sich im Anhang C in Textform.

Ebenfalls zeigt Abbildung 6.1.1 einen Switch im Vordergrund, über den das Kontrollnetzwerk für den XTPeer zusammengeschaltet ist. Wie bereits im vorherigen Kapitel, im Abschnitt 5.2 Konfiguration des Netzwerk zu sehen ist, haben die Router R1-R5 folgende, für dieses Kapitel noch einmal kurz zusammengefasste Konfiguration, bzw. zeigt Tabelle 6.1.1 die benachbarten Router anhand der einzelnen LAN-Schnittstellen:

Router / Interface	R1	R2	R3	R4	R5
LAN 4	Host	R1	R2	R3	R1
LAN 3	R2	R3	R4	R5	R3
LAN 2	R5	-	R5	-	R4
LAN 1	-	-	-	-	Host
Internet	XTPeer	XTPeer	XTPeer	XTPeer	XTPeer

Tabelle 6.1.1: Schnittstellenbelegung

Der Eintrag „XTPeer“ steht für die am Internet-Port zugewiesene Adresse des Kontrollnetzwerkes.

6.2 Gegenüberstellung von RIP und RMTI

In diesem Abschnitt wird das in Abbildung 6.2.1 nochmals graphisch dargestellte Szenario erläutert, bzw. das Verhalten der Router einmal mit RIP-Routing, aber ohne RMTI in Abbildung 6.2.2 und dann nochmals mit RMTI, in Abbildung 6.2.3.

Es wird der Ausfall des Netzes 10.0.10.0/24 simuliert, und der damit verbundene CTI erzeugt.

²⁴ (eXternally Triggered Peer) Programm zur externen Steuerung und Analyse des Quagga RIP-Daemons

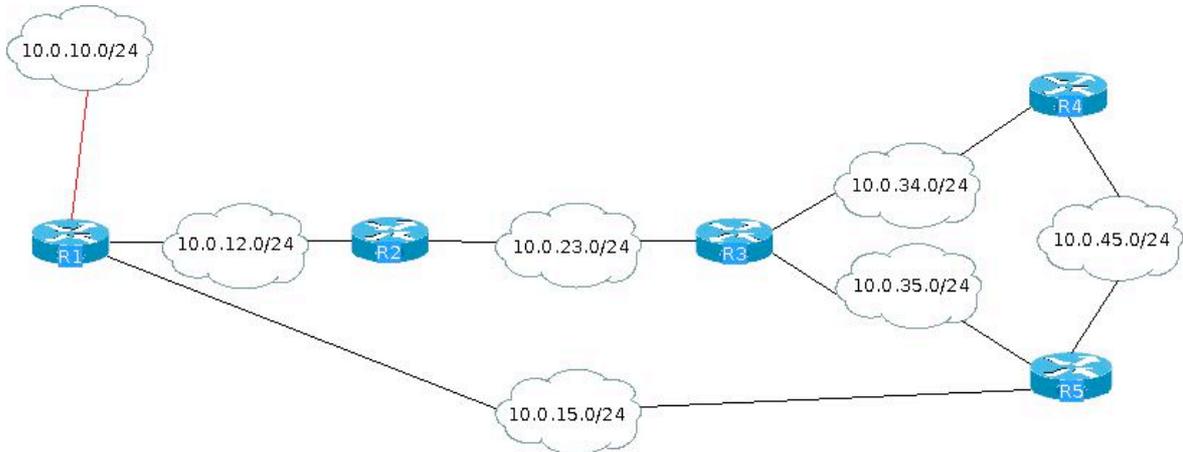


Abbildung 6.2.1: Topologie des Testnetzwerkes

Die Abbildung 6.2.1 zeigt die in dargestellte Testumgebung in Abbildung 6.1.1 dargestellte Netzwerk-Topologie, ohne die Hosts (Laptop's) das Kontrollnetz und den Host mit dem XTPeer.

In den beiden folgenden Abbildungen wird nun die Darstellung der XTPeer-Graphen gezeigt. Der Ausfall des Netzes 10.0.10.0/24, dargestellt durch den Ausschlag im Graphen von R2, sowie die „Trepchenbildung“ bei R3 und R5.

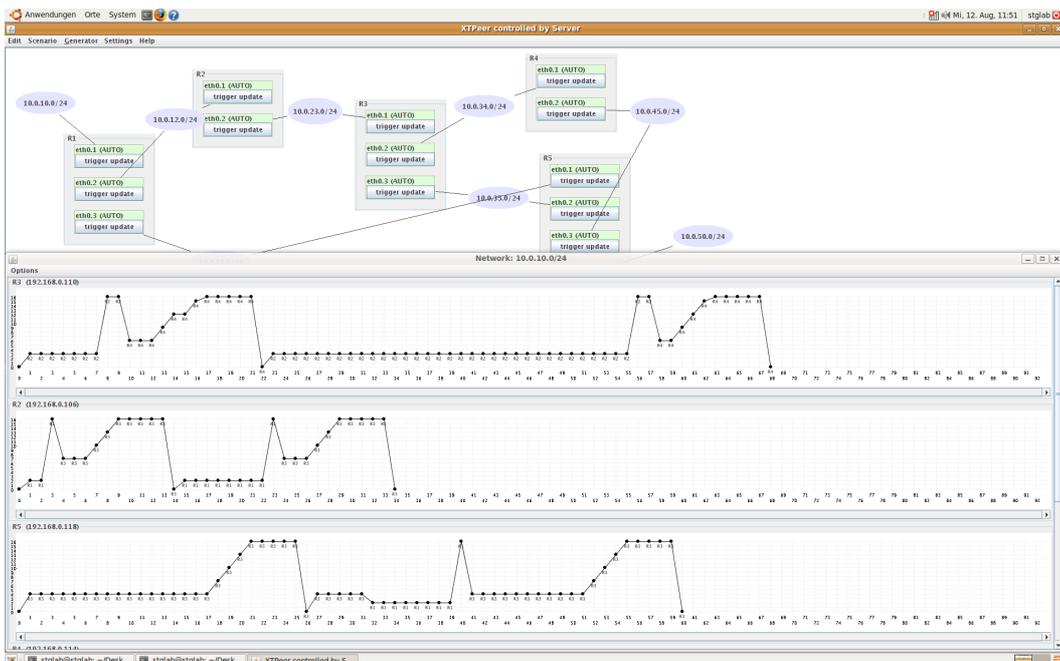


Abbildung 6.2.2: XTPeer-Graph von Szenario ohne RMTI

Hierbei handelt es sich um den bereits in Kapitel 4 beschriebenen CTI, der durch das Austauschen falscher Informationen zwischen R3, R4 und R5 zustande kommt. Bei Erreichen der Metrik 16 gilt dann das Netz als unerreichbar. (s. Abbildung 6.2.2)

In Abbildung 6.2.3 sieht man nun die Verhinderung des CTI anhand der verworfenen Information seitens von R4. Die falsche Information wird nicht weitergegeben. Das rote Dreieck steht dabei für die in der Routing-Tabelle ausgefallene Route. Das gelbe Dreieck steht für die Metrik der alternativ angebotenen Route, die ohne den RMTI akzeptiert worden wäre.

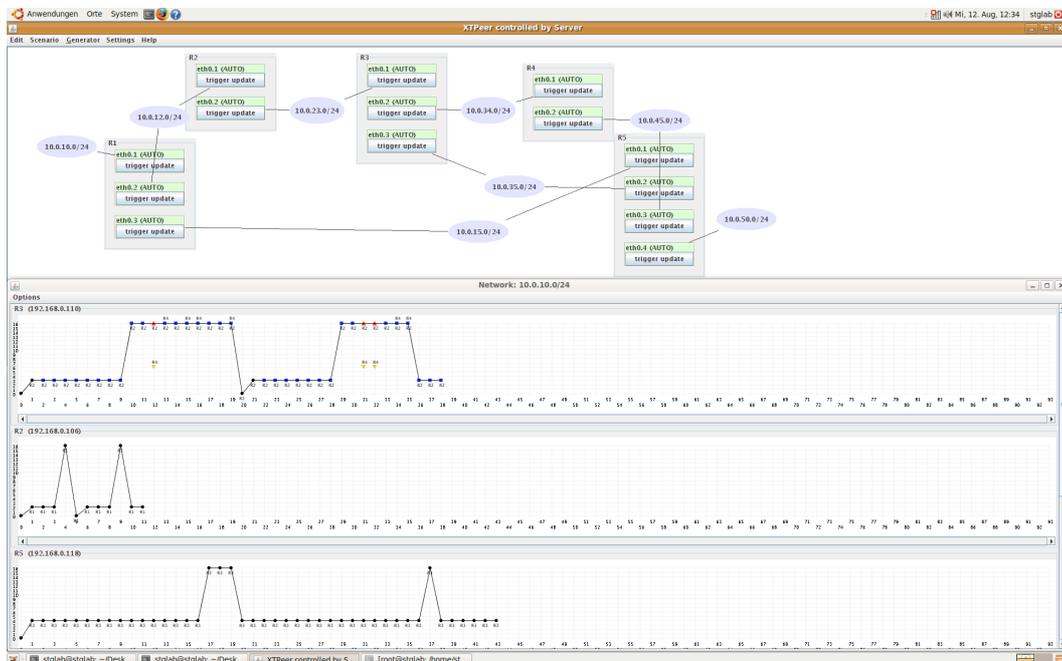


Abbildung 6.2.3: XTPeer-Graph von Szenario mit RMTI

Kapitel 7

Fazit und Ausblick

Der große Vorteil bleibt sicherlich die hohe Flexibilität in der Entwicklung am OpenWRT-Projekt. Die Plattform wird in Zukunft sicherlich, wenn auch in veränderter Form, zur Demonstration eingesetzt werden. Der Umfang der Topologie, sowie die auf den einzelnen Routern laufende Software unterliegen dabei den ständigen Änderungen der fortschreitenden Entwicklung.

Bei Fertigstellung der Arbeit lag bereits eine aktuellere Software mit der OpenWRT-Version 10.03 „Backfire“ vor. Die Installation des RMTI-Paketes, sowie des Zebra-Daemons funktionierte mit der Version 8.09rc1 „Kamikaze“ auf den Routern fehlerfrei. Innerhalb des Testszenarios konnte der CTI stets durch den RMTI-Algorithmus verhindert werden.

Interessant ist sicherlich in welcher Form OpenWRT-Router an VNUML-Netzwerke angebunden werden können, oder eine Analyse ohne virtuelle Netzwerke. Aufgrund der hohen Anzahl von anzuschaffenden Routern, mit den entsprechend hohen Anschaffungskosten, sowie aus Platzgründen für eine solche Demonstrationsplattform, dürfte die Variante mit Routerhardware wohl eher in den Hintergrund treten.

Anhang A: Begriffe und Terminologie

Router: Ein Router verbindet mittels Software, innerhalb eines Netzwerkes die verschiedenen Teilnetze miteinander. Die Verbindung über die Vermittlungsschicht ermöglicht die Kommunikation der Endgeräte, dabei werden die versendeten Datenpakete analysiert und anhand der Routing-Tabellen zu den jeweiligen Ziel-IP-Adressen gesendet. Der Vorgang der Zuweisung der Pakete wird als „Routing“ bezeichnet.

Subnetz: Als Subnetz oder Teilnetz bezeichnet man einen zusammenhängenden Teil von IP-Adressen aus der Gesamtanzahl aller verfügbaren Netze. Durch das sogenannte „Subnetting“, werden IP-Adressen mit der gleichen Netzanteil zu einem Subnetz zusammen gefasst. Der Vorteil dieser Unterteilung in Subnetze liegt in der Verhinderung von zu hohem Adressverlust.

Routing-Algorithmus: Ein Routing-Algorithmus ist ein Algorithmus, der die Selbstorganisation innerhalb eines Netzes steuert. Im wesentlichen existieren zwei Ansätze den Routingvorgang zu organisieren und zwar über Distanz-Vektor-Algorithmen, Pfad-Vektor-Algorithmen und die Link-State Algorithmen.

RIP-Router: Ein RIP-Router ist eine Netzwerkkomponente, deren Software die Ausführung des RIP-Algorithmus ermöglicht, um so mit anderen RIP-Routern zu kommunizieren.

Routing-Daemon: Ein Routing-Daemon ist ein Programm, dass Routing-Dienste zur Verfügung stellt. Der Daemon wird in der Regel beim Starten der Routersoftware geladen.

Switch: Im Gegensatz zum Hub (1. Schicht) arbeitet ein Switch auf der 2. Schicht oder 3. Schicht des OSI-Modells, d.h., dass eine zielgenaue Weiterleitung der Pakete über die

entsprechenden Port zu den Endgeräten möglich ist, während ein Hub Pakete zu allen Ports weiterleitet.

VLAN: Das Virtual Local Area Network ermöglicht die Trennung einzelner Teilnetze in einem Switch. Die konfigurierten VLAN's leiten dann trotz physikalischer Verbindung keine Datenpakete zueinander weiter, sondern nur innerhalb des jeweiligen VLAN's.

Interface: Der Begriff Interface bezeichnet die physikalische Netzwerkschnittstelle zwischen einer Netzwerkkomponente und dem Netzwerk.

Netzwerk-Schleife: Eine Netzwerk-Schleife bezeichnet einen Ring aus Routern, der aufgrund von redundanten Verbindungen entstanden ist, um die Ausfallsicherheit eines Netzwerks zu erhöhen. Sie besteht aus wenigstens zwei Routern, die über zwei getrennte, physikalische Leitungen miteinander verbunden sind.

Anhang B: Netzwerkkonfigurationsdateien

```
Router R1: (/etc/config/network)
1 config 'switch' 'eth0'
2     option 'enable' '1'
4 config 'switch_vlan' 'eth0_0'
5     option 'device' 'eth0'
6     option 'vlan' '0'
7     option 'ports' '4 5'
9 config 'switch_vlan' 'eth0_1'
10    option 'device' 'eth0'
11    option 'vlan' '1'
12    option 'ports' '3 5'
13
14 config 'interface' 'loopback'
15    option 'ifname' 'lo'
16    option 'proto' 'static'
17    option 'ipaddr' '127.0.0.1'
18    option 'netmask' '255.0.0.0'
20 config 'interface' 'xtpeer'
21    option 'proto' 'static'
22    option 'ipaddr' '192.168.1.102'
23    option 'netmask' '255.255.255.0'
25    option 'defaultroute' '0'
26    option 'peerdns' '0'
27 config 'switch_vlan' 'eth0_2'
28    option 'device' 'eth0'
29    option 'vlan' '2'
30    option 'ports' '2 5'
```

```
31 config 'switch_vlan' 'eth0_3'
32     option 'device' 'eth0'
33     option 'vlan' '3'
34     option 'ports' '1 5'
35 config 'switch_vlan' 'eth0_4'
36     option 'device' 'eth0'
37     option 'vlan' '4'
38     option 'ports' '0 5'
39 config 'interface' 'lan2'
40     option 'proto' 'static'
41     option 'ifname' 'eth0.2'
42     option 'ipaddr' '10.0.15.1'
43     option 'netmask' '255.255.255.0'
44     option 'defaultroute' '0'
45     option 'peerdns' '0'
46 config 'interface' 'lan3'
47     option 'proto' 'static'
48     option 'ifname' 'eth0.3'
49     option 'ipaddr' '10.0.45.2'
50     option 'netmask' '255.255.255.0'
51     option 'defaultroute' '0'
52     option 'peerdns' '0'
53 config 'interface' 'lan4'
54     option 'proto' 'static'
55     option 'ifname' 'eth0.4'
56     option 'ipaddr' '10.0.10.1'
57     option 'netmask' '255.255.255.0'
58     option 'defaultroute' '0'
59     option 'peerdns' '0'
```

```
Router R2: (/etc/config/network)
1 config 'switch' 'eth0'
2     option 'enable' '1'
4 config 'switch_vlan' 'eth0_0'
5     option 'device' 'eth0'
6     option 'vlan' '0'
7     option 'ports' '4 5'
9 config 'switch_vlan' 'eth0_1'
10    option 'device' 'eth0'
11    option 'vlan' '1'
12    option 'ports' '3 5'
13 config 'interface' 'loopback'
14    option 'ifname' 'lo'
15    option 'proto' 'static'
16    option 'ipaddr' '127.0.0.1'
17    option 'netmask' '255.0.0.0'
18 config 'interface' 'xtpeer'
19    option 'proto' 'static'
20    option 'ipaddr' '192.168.1.106'
21    option 'netmask' '255.255.255.0'
22    option 'defaultroute' '0'
23    option 'peerdns' '0'
24 config 'switch_vlan' 'eth0_2'
25    option 'device' 'eth0'
26    option 'vlan' '2'
27    option 'ports' '2 5'
```

```
28 config 'switch_vlan' 'eth0_3'
29     option 'device' 'eth0'
30     option 'vlan' '3'
31     option 'ports' '1 5'
32
33 config 'switch_vlan' 'eth0_4'
34     option 'device' 'eth0'
35     option 'vlan' '4'
36     option 'ports' '0 5'
37
38 config 'interface' 'lan3'
39     option 'proto' 'static'
40     option 'ifname' 'eth0.3'
41     option 'ipaddr' '10.0.23.1'
42     option 'netmask' '255.255.255.0'
43     option 'defaultroute' '0'
44     option 'peerdns' '0'
45
46 config 'interface' 'lan4'
47     option 'proto' 'static'
48     option 'ifname' 'eth0.4'
49     option 'ipaddr' '10.0.12.2'
50     option 'netmask' '255.255.255.0'
51     option 'defaultroute' '0'
52     option 'peerdns' '0'
```

```
Router R3: (/etc/config/network)
1 config 'switch' 'eth0'
2     option 'enable' '1'
4 config 'switch_vlan' 'eth0_0'
5     option 'device' 'eth0'
6     option 'vlan' '0'
7     option 'ports' '4 5'
9 config 'switch_vlan' 'eth0_1'
10    option 'device' 'eth0'
11    option 'vlan' '1'
12    option 'ports' '3 5'
13
14 config 'interface' 'loopback'
15    option 'ifname' 'lo'
16    option 'proto' 'static'
17    option 'ipaddr' '127.0.0.1'
18    option 'netmask' '255.0.0.0'
20 config 'interface' 'xtpeer'
21    option 'proto' 'static'
22    option 'ipaddr' '192.168.1.110'
23    option 'netmask' '255.255.255.0'
25    option 'defaultroute' '0'
26    option 'peerdns' '0'
27
28 config 'switch_vlan' 'eth0_2'
29    option 'device' 'eth0'
30    option 'vlan' '2'
31    option 'ports' '2 5'
32
```

```
32 config 'switch_vlan' 'eth0_3'
33     option 'device' 'eth0'
34     option 'vlan' '3'
35     option 'ports' '1 5'
36
37 config 'switch_vlan' 'eth0_4'
38     option 'device' 'eth0'
39     option 'vlan' '4'
40     option 'ports' '0 5'
41
42 config 'interface' 'lan2'
43     option 'proto' 'static'
44     option 'ifname' 'eth0.2'
45     option 'ipaddr' '10.0.35.1'
46     option 'netmask' '255.255.255.0'
47     option 'defaultroute' '0'
48     option 'peerdns' '0'
49 config 'interface' 'lan3'
50     option 'proto' 'static'
51     option 'ifname' 'eth0.3'
52     option 'ipaddr' '10.0.34.2'
53     option 'netmask' '255.255.255.0'
54     option 'defaultroute' '0'
55     option 'peerdns' '0'
56 config 'interface' 'lan4'
57     option 'proto' 'static'
58     option 'ifname' 'eth0.4'
59     option 'ipaddr' '10.0.23.2'
60     option 'netmask' '255.255.255.0'
61     option 'defaultroute' '0'
62     option 'peerdns' '0'
```

```
Router R4: (/etc/config/network)
1 config 'switch' 'eth0'
2     option 'enable' '1'
4 config 'switch_vlan' 'eth0_0'
5     option 'device' 'eth0'
6     option 'vlan' '0'
7     option 'ports' '4 5'
9 config 'switch_vlan' 'eth0_1'
10    option 'device' 'eth0'
11    option 'vlan' '1'
12    option 'ports' '3 5'
13
14 config 'interface' 'loopback'
15    option 'ifname' 'lo'
16    option 'proto' 'static'
17    option 'ipaddr' '127.0.0.1'
18    option 'netmask' '255.0.0.0'
20 config 'interface' 'xtpeer'
21    option 'proto' 'static'
22    option 'ipaddr' '192.168.1.114'
23    option 'netmask' '255.255.255.0'
25    option 'defaultroute' '0'
26    option 'peerdns' '0'
27
28 config 'switch_vlan' 'eth0_2'
29    option 'device' 'eth0'
30    option 'vlan' '2'
31    option 'ports' '2 5'
32
```

```
33 config 'switch_vlan' 'eth0_3'
34     option 'device' 'eth0'
35     option 'vlan' '3'
36     option 'ports' '1 5'
37
38 config 'switch_vlan' 'eth0_4'
39     option 'device' 'eth0'
40     option 'vlan' '4'
41     option 'ports' '0 5'
42
43 config 'interface' 'lan3'
44     option 'proto' 'static'
45     option 'ifname' 'eth0.3'
46     option 'ipaddr' '10.0.45.1'
47     option 'netmask' '255.255.255.0'
48     option 'defaultroute' '0'
49     option 'peerdns' '0'
50
51 config 'interface' 'lan4'
52     option 'proto' 'static'
53     option 'ifname' 'eth0.4'
54     option 'ipaddr' '10.0.34.2'
55     option 'netmask' '255.255.255.0'
56     option 'defaultroute' '0'
57     option 'peerdns' '0'
```

```
Router R5: (/etc/config/network)
1 config 'switch' 'eth0'
2     option 'enable' '1'
4 config 'switch_vlan' 'eth0_0'
5     option 'device' 'eth0'
6     option 'vlan' '0'
7     option 'ports' '4 5'
9 config 'switch_vlan' 'eth0_1'
10    option 'device' 'eth0'
11    option 'vlan' '1'
12    option 'ports' '3 5'
13 config 'interface' 'loopback'
14    option 'ifname' 'lo'
15    option 'proto' 'static'
16    option 'ipaddr' '127.0.0.1'
17    option 'netmask' '255.0.0.0'
18 config 'interface' 'xtpeer'
19    option 'proto' 'static'
20    option 'ipaddr' '192.168.1.118'
21    option 'netmask' '255.255.255.0'
22    option 'defaultroute' '0'
23    option 'peerdns' '0'
24 config 'switch_vlan' 'eth0_2'
25    option 'device' 'eth0'
26    option 'vlan' '2'
27    option 'ports' '2 5'
```

```
28 config 'switch_vlan' 'eth0_3'
29     option 'device' 'eth0'
30     option 'vlan' '3'
31     option 'ports' '1 5'
32
33 config 'switch_vlan' 'eth0_4'
34     option 'device' 'eth0'
35     option 'vlan' '4'
36     option 'ports' '0 5'
37
38 config 'interface' 'lan1'
39     option 'proto' 'static'
40     option 'ifname' 'eth0.1'
41     option 'ipaddr' '10.0.50.1'
42     option 'netmask' '255.255.255.0'
43     option 'defaultroute' '0'
44     option 'peerdns' '0'
45
46 config 'interface' 'lan2'
47     option 'proto' 'static'
48     option 'ifname' 'eth0.2'
49     option 'ipaddr' '10.0.45.2'
50     option 'netmask' '255.255.255.0'
51     option 'defaultroute' '0'
52     option 'peerdns' '0'
53
```

Anhang C: XML-Szenario für XTPeer

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE vnuml SYSTEM "/usr/local/share/xml/vnuml/vnuml.dtd">
<vnuml>
  <global>
    <version>1.8</version>
    <simulation_name>scenario</simulation_name>
    <ssh_version>2</ssh_version>
    <ssh_key>/root/.ssh/id_rsa.pub</ssh_key>
    <automac/>
    <vm_mgmt type="private" network="192.168.0.0" mask="24" offset="100">
      <host_mapping/>
    </vm_mgmt>
    <vm_defaults exec_mode="mconsole">
      <filesystem type="cow">/usr/local/share/vnuml/filesystems/root_fs_tutorial</
filesystem>
      <mem>50M</mem>
      <kernel>/usr/local/share/vnuml/kernels/linux-2.6.18.1-bb2-xt-4m</kernel>
      <console id="0">xterm</console>
      <!--xterm>gnome-terminal,-t,-x</xterm-->
      <forwarding/>
    </vm_defaults>
  </global>
  <net name="a" mode="uml_switch" type="lan" />
  <net name="b" mode="uml_switch" type="lan" />
  <net name="c" mode="uml_switch" type="lan" />
  <net name="d" mode="uml_switch" type="lan" />
  <net name="e" mode="uml_switch" type="lan" />
  <net name="f" mode="uml_switch" type="lan" />
  <net name="g" mode="uml_switch" type="lan" />
```

```
<net name="h" mode="uml_switch" type="lan" />
<!--
<net name="a" mode="virtual_bridge" type="lan" />
<net name="b" mode="virtual_bridge" type="lan" />
<net name="c" mode="virtual_bridge" type="lan" />
<net name="d" mode="virtual_bridge" type="lan" />
<net name="e" mode="virtual_bridge" type="lan" />
<net name="f" mode="virtual_bridge" type="lan" />
-->

<vm name="R1">
  <if id="1" net="d">
    <ipv4>10.0.10.1</ipv4>
  </if>
  <if id="2" net="a">
    <ipv4>10.0.12.1</ipv4>
  </if>
  <if id="3" net="b">
    <ipv4>10.0.15.1</ipv4>
  </if>
  <exec seq="start" type="verbatim">sysctl -w net.ipv4.conf.all.rp_filter=0</
exec>
  <exec seq="start" type="verbatim">zebra -f /etc/quagga/zebra.conf -d</exec>
  <exec seq="start" type="verbatim">ripd -f /etc/quagga/ripd.conf -x 5000 -d</
exec>
  <exec seq="stop" type="verbatim">killall zebra</exec>
  <exec seq="stop" type="verbatim">killall ripd</exec>
</vm>

<vm name="R2">
  <if id="1" net="a">
    <ipv4>10.0.12.2</ipv4>
  </if>
  <if id="2" net="c">
    <ipv4>10.0.23.1</ipv4>
```

```
</if>
exec> <exec seq="start" type="verbatim">sysctl -w net.ipv4.conf.all.rp_filter=0</
exec> <exec seq="start" type="verbatim">zebra -f /etc/quagga/zebra.conf -d</exec>
exec> <exec seq="start" type="verbatim">ripd -f /etc/quagga/ripd.conf -x 5000 -d</
exec> <exec seq="stop" type="verbatim">killall zebra</exec>
exec> <exec seq="stop" type="verbatim">killall ripd</exec>
</vm>
<vm name="R3">
  <if id="1" net="c">
    <ipv4>10.0.23.2</ipv4>
  </if>
  <if id="2" net="e">
    <ipv4>10.0.34.1</ipv4>
  </if>
  <if id="3" net="f">
    <ipv4>10.0.35.1</ipv4>
  </if>
  <exec seq="start" type="verbatim">sysctl -w net.ipv4.conf.all.rp_filter=0</
exec> <exec seq="start" type="verbatim">zebra -f /etc/quagga/zebra.conf -d</exec>
exec> <exec seq="start" type="verbatim">ripd -f /etc/quagga/ripd.conf -x 5000 -d</
exec> <exec seq="stop" type="verbatim">killall zebra</exec>
exec> <exec seq="stop" type="verbatim">killall ripd</exec>
</vm>
<vm name="R4">
  <if id="1" net="e">
    <ipv4>10.0.34.2</ipv4>
  </if>
  <if id="2" net="g">
    <ipv4>10.0.45.1</ipv4>
  </if>
  <exec seq="start" type="verbatim">sysctl -w net.ipv4.conf.all.rp_filter=0</
exec>
```

```
<exec seq="start" type="verbatim">zebra -f /etc/quagga/zebra.conf -d</exec>
<exec seq="start" type="verbatim">ripd -f /etc/quagga/ripd.conf -x 5000 -d</
exec>

<exec seq="stop" type="verbatim">killall zebra</exec>

<exec seq="stop" type="verbatim">killall ripd</exec>

</vm>

<vm name="R5">

  <if id="1" net="f">

    <ipv4>10.0.35.2</ipv4>

  </if>

  <if id="2" net="g">

    <ipv4>10.0.45.2</ipv4>

  </if>

  <if id="3" net="b">

    <ipv4>10.0.15.2</ipv4>

  </if>

  <if id="4" net="h">

    <ipv4>10.0.50.1</ipv4>

  </if>

  <exec seq="start" type="verbatim">sysctl -w net.ipv4.conf.all.rp_filter=0</
exec>

  <exec seq="start" type="verbatim">zebra -f /etc/quagga/zebra.conf -d</exec>

  <exec seq="start" type="verbatim">ripd -f /etc/quagga/ripd.conf -x 5000 -d</
exec>

  <exec seq="stop" type="verbatim">killall zebra</exec>

  <exec seq="stop" type="verbatim">killall ripd</exec>

</vm>

</vnuml>
```

Literaturverzeichnis

- [AP07] Paul Asadoorian and Larry Pesce. *Linksys WRT54G Ultimate Hacking*. Syngress Publishing, Inc. and Elsevier, Inc., Burlington, USA., 2007.
- [BEL57] R.E. Bellmann. *Dynamic Programming*. Princeton University Press. 1957.
- [BOH08] Frank Bohdanowicz. *Weiterentwicklung und Implementierung des RIP-MTI-Daemons*. Diplomarbeit, Universität Koblenz-Landau, 2008.
- [KEU07] Tim Keupen. *Generierung von Testfällen für den RIP-MTI-Algorithmus*. Diplomarbeit, Universität Koblenz-Landau, 2007.
- [KOC05] Tobias Koch, *Implementation und Simulation von RIP-MTI*. Diplomarbeit, Universität Koblenz-Landau, 2005.
- [PD04] Larry L. Peterson and Bruce S. Davie. *Computernetze*. Dpunkt Verlag, 3rd Edition, San Francisco, 2004.
- [RFC 1058] C. Hedrick. RFC 1058 - *Routing Information Protocol*. IEEE RFC, 1988.
- [SCH99] A. Schmid. *RIP-MTI: Minimum-effort loop-free distance vector routing algorithm*. Diplomarbeit, Universität Koblenz-Landau, 1999.
- [STD53] G. Marlin. *RIPv2*. IEEE RFC, 1998.
- [TAN03] Andrew S. Tanenbaum. *Computernetzwerke*. Prentice Hall. 3rd Edition. 2003.

[wika] Linksys WRT54G. Wiki. Online, Stand 10. August 2010.

[wikb] Kamikaze 8.09rc1. Open-WRT-Wiki. Online Stand 09. November 2010

[wikc] The uci-System, OpenWrt-Wiki. Online Stand 10. August 2010