

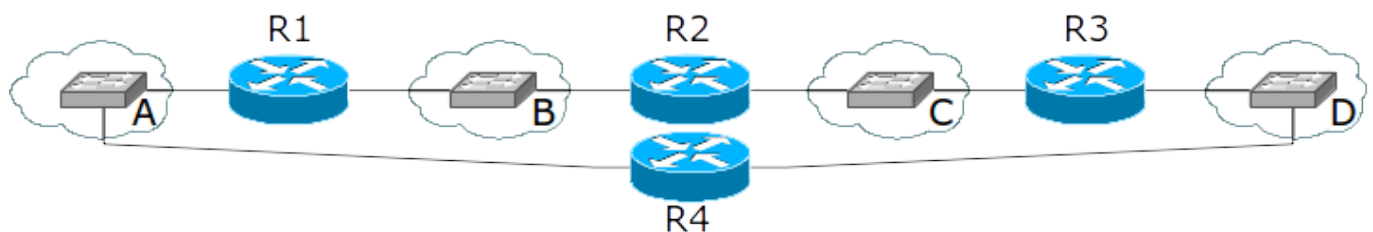
RIP-Laborübung im keinpasswort-Account

Aufgabe

Es sollen mehrere Router „in Reihe“ miteinander verschaltet werden, so dass sie einen Kreis bilden. Jeder Router hat genau zwei Nachbarn in jeweils unterschiedlichen Netzen. Die Netze können jeweils per Switch „simuliert“ werden.

Die folgende Abbildung zeigt den prinzipiellen Aufbau. Je nach Größe der Gruppe können entsprechend mehr Router und Netze zum Einsatz kommen.

Ziel ist es die Nachrichten zwischen den Routern mittels Wireshark zu analysieren und Routing-Informationsweitergabe zu überprüfen.



Hinweis-Seite:

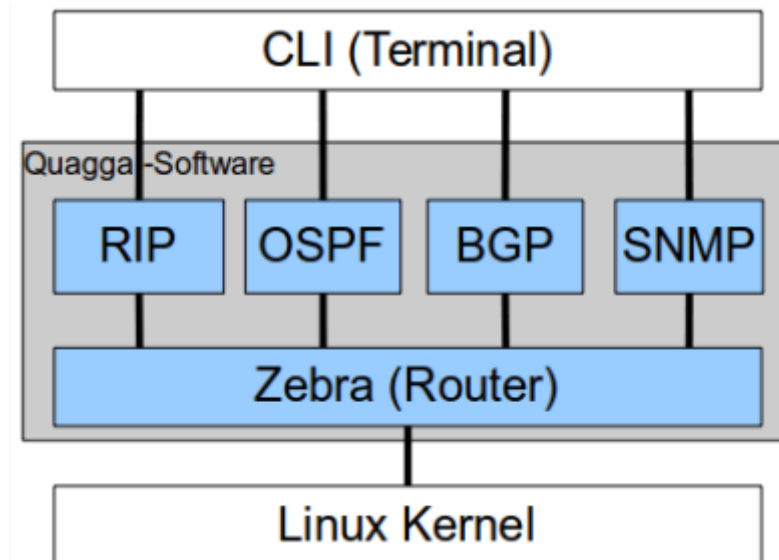
Offizielle Dokumentation zu quagga: <http://www.nongnu.org/quagga/docs.html>

HowTo: <http://opentodo.net/2012/08/configuring-routing-protocols-with-quagga/>

Schritte zum Einrichten von RIP

Zunächst sollte man sich mit dem Grundgerüst quagga vertraut machen. Dieser Daemon (Dienst) stellt die Basis für alle weiteren Konfigurationen dar. Der **quagga-Daemon** ist auf einer eigenen Seite erklärt. Über den quagga-Dienst werden indirekt die einzelnen Routing-Module (z.B. RIP, OSPF) gestartet.

Das folgende Bild zeigt den Aufbau der Module von Quagga.



Im zebra-Modul wird die grundlegende Konfiguration des Routers vorgenommen u.a. Name des Routers, IP-Adressen (unabhängig von bereits eingestellten Adressen), etc.

In weiteren Modulen (ripd, ospfd und bgpd) werden die Parameter für die einzelnen Routing-Protokolle festgelegt. Z.B. Split Horizont bei rip bzw. für welche Netze rip aktiviert werden soll usw.

Im Folgenden werden die grundlegenden Schritte aufgeführt, die nötig sind, um RIP unter Linux zu aktivieren.

1. [Routing aktivieren](#)
2. Grundlegende Daten im zebra-daemon [konfigurieren](#)¹⁾
3. Spezielle RIP-Parameter im rip-daemon [konfigurieren](#)

Routing aktivieren (FORWARDING)

Um einen PC zum Router zu machen, muss dieser in den FORWARDING-Modus versetzt werden. Das heißt der PC kann Datenpakete über Schnittstellen hinweg weiterleiten. Normalerweise behandelt ein PC jede Schnittstelle getrennt und würde die Datenpakete nicht weiterleiten.

Unter Linux geschieht dies über den folgenden Befehl:

```
sudo sysctl net.ipv4.ip_forward=1
```

Über den folgenden Befehl kann der aktuelle Status des FORWARDING überprüft werden:

```
cat /proc/sys/net/ipv4/ip_forward  
1: FORWARDING aktiviert  
0: FORWARDING deaktiviert
```

Konfiguration des zebra-Moduls

Man muss sich die einzelnen Module wie einen „Standalone“-Router vorstellen, der per Remote-Zugriff konfiguriert wird. Da man direkten Zugriff auf die Maschine selbst hat, ist die remote-Adresse der eigene Rechner also localhost:

```
telnet localhost 2601
```

AUSGABE:

```
Hello, this is Quagga (version 0.99.22.1).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
User Access Verification
```

```
Password:
```

Als Passwort verwendet man zebra bzw. das Passwort, das man in der zebra-Konfigurationsdatei (/etc/quagga/zebra.conf²⁾) gesetzt hat.

Alternativ zur direkten Eingabe der Kommandos, kann auch eine Textdatei als Konfigurationsdatei herangezogen werden. Der Befehl `show running-config` gibt die aktuelle Konfiguration aus, die für die Textdatei genutzt werden kann. Für das obige Beispiel würde demnach folgender Inhalt darin stehen:

[zebra.conf_sample](#)

```
!  
hostname tralala  
password zebra  
enable password zebra  
!  
interface eth0  
  ip address 10.0.0.1/8  
  ipv6 nd suppress-ra  
!  
interface eth1  
  ipv6 nd suppress-ra  
!  
interface lo  
!  
!  
!  
line vty  
!  
end
```

Die folgende Tabelle zeigt die wichtigsten Befehle des zebra-Moduls. Mit ? bzw. list kann jederzeit die Hilfe angezeigt werden ([vgl. STP-Übung](#)).

Befehl	Funktion	Beispiel
show	Zeigt die aktuelle Konfiguration an z.B. die Routing-Tabelle	show ip route
enable/disable	Aktiviert/deaktiviert den Router	enable oder disable
configure terminal	Wechseln in den Konfigurationsmodus. Neues Prompt:Router(config)	configure terminal
interface	Auswählen der Schnittstelle, die konfiguriert werden soll. Im Prompt:Router(config)	interface eth0
ip/no ip	Setzen/Löschen einer IP-Adresse für das ausgewählte Interface. Im Prompt:Router(config-if):	ip address 10.0.0.1/8
shutdown/no shutdown	Schnittstelle aktivieren/deaktivieren. Im Prompt:Router(config-if):	no shutdown
hostname	Setzen des Router-Namens.	hostname <NEUERNAME>

Beispiel Befehlsfolge im CLI³⁾:

```
telnet localhost 2601
Password: zebra

Router>enable
Password: zebra
Router#configure terminal
Router(config)#interface eth0
Router(config-if)#ip address 10.0.0.1/8
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#hostname tralala
tralala(config-if)#show running-config

Current configuration:
!
hostname tralala
password zebra
enable password zebra
!
interface eth0
 ip address 10.0.0.1/8
 ipv6 nd suppress-ra
!
interface eth1
 ipv6 nd suppress-ra
!
interface lo
!
!
!
line vty
```

```
!
end
```

Konfiguration des ripd-Moduls

[ripd.conf_sample](#)

```
! *- rip *-
!
!
!
hostname ripd
password zebra
!
! debug rip events
! debug rip packet
!
router rip
  network 11.0.0.0/8
  network eth0
! route 10.0.0.0/8
! distribute-list private-only in eth0
!
version 1
ip split-horizon
```

Das RIP-Modul ripd wird ähnlich konfiguriert. Hier legt man fest, auf welchen Schnittstellen das RIP-Protokoll ausgeführt werden soll. Bei der Telnet-Verbindung muss allerdings der Port 2602 verwendet werden.

Hier ein Auszug der wichtigsten Befehle:

Befehl	Funktion	Beispiel
show	Zeigt die aktuelle Konfiguration an z.B. die Routing-Tabelle	show running-config
enable/disable	Aktiviert/deaktiviert den Router	enable oder disable
configure terminal	Wechseln in den Konfigurationsmodus. Neues Prompt:Router(config)	configure terminal
hostname	Setzen des Router-Namens.	hostname <NEUERNAME>
router	Auswahl des Routers. Im Prompt:Router(config)	router rip oder router zebra
version	Version des RIP-Protokolls. Im Prompt:Router(config)	version 2
network <SUBNETZ>	Subnetz für das RIP aktiviert werden soll. Im Prompt:Router(config)	network 10.0.0.0/8
network <SCHNITTSTELLE>	Schnittstelle für die RIP aktiviert werden soll. Im Prompt:Router(config)	network eth0

Befehl	Funktion	Beispiel
<code>passive-interface / no passive-interface</code>	Das Interface konsumiert zwar alle RIP-Nachrichten, sendet aber selbst keine(DEFAULT).	<code>no passive-interface eth0</code>
<code>ip split-horizon / no ip split-horizon</code>	Das Split-Horizon wird aktiviert(DEFAULT) / deaktiviert. Mit no sendet das Interface aktiv.	<code>no passive-interface eth0</code>

Beispiel Befehlsfolge im CLI⁴⁾:

```
telnet localhost 2602  
<code>telnet localhost 2602
```

Per Konfigurationsdatei konfigurieren

ACHTUNG: Abweichend von der quagga-Beschreibung wurde für die Laborumgebung ein anderer Pfad für die Konfigurationsdateien gewählt. Dies ermöglicht es die Konfiguration im Account des Benutzers keinpasswort vorzunehmen und nach jedem Login/Logout bzw. Reboot zurückzusetzen.

Als Ordner für die zum quagga gehörenden Module wurde der folgende Pfad voreingestellt:

```
/home/keinpasswort/quagga/
```

Dort befinden sich die Beispielkonfigurationen für die beiden Daemons `ripd` und `ospfd` sowie den `zebra-daemon`. Die Dateien können von jedem Benutzer geändert werden. Durch Eingabe des nächsten Befehls wird der gesamte quagga-Daemon und seine Module (`ripd` und `ospfd`) neugestartet (`restart`), gestoppt (`stop`) bzw. gestartet (`start`).

```
sudo /etc/init.d/quagga [restart|stop|start]
```

1)
Daemon: Ein Dienst wird unter Linux auch als Daemon bezeichnet. So gibt es z.B. einen Drucker-Dienst, der `cups-daemon` heisst.

2)
hier kann auch der Name des Routers gesetzt werden

3) , 4)
CLI: **command line interface**; Kommandozeile des `zebra/ripd`-Daemons

From:
<https://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**

Permanent link:
https://www.kopfload.de/doku.php?id=lager:lok_netze:riplabor_keinpasswort&rev=1448549389

Last update: **2025/11/19 16:13**

