Basic FreeRouter Tutorial

FreeRouter is a control plane: Router OS process speaks various network protocols, (re)encap packets, and exports forwarding tables to hardware switches. Basically, it is only necessary to install the Java Runtime Environment (JRE). Below is demonstrated how to install it on operating systems: Linux, Windows and macOS.

Linux

For demonstration purposes, the Debian-based Linux installation was chosen.

```
#sudo apt-get install default-jre-headless --no-install-recommends
```

Windows

In order to install the Windows version of Java, you need to visit the official Java website and download the Windows executable. After the download, check if your user has permission to install and perform the installation through the graphical environment.

Mac

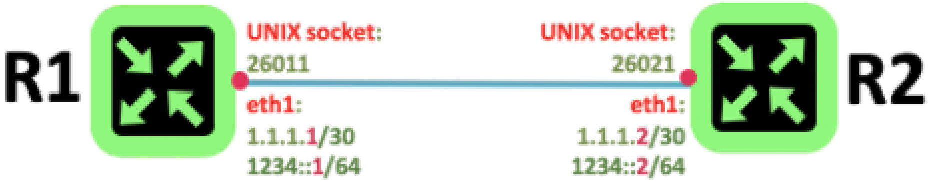
There are several options on how to install java, here we chose to install it in text mode.

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk list java
sdk install java 17.0.2-open
sdk default java 17.0.2-open
java -version
```

Install FreeRtr

The freeRouter homepage is at freertr.net. Starting from this page, you'll find various resources such as source code (there is also a GitHub mirror), binaries, and other images that might be of your interest. From there we just download the freeRouter jar files.

```
#wget freertr.net/rtr.jar
```



Hardware Router1

freeRouter hardware file: **r1-hw.txt**

```
int eth1 eth 0000.1111.0001 127.0.0.1 26011 127.0.0.1 26021
tcp2vrf 1123 v1 23
```

Software Router1

freeRouter software configuration file: **r1-sw.txt**

```
hostname r1
!
vrf definition v1
 exit
!
int eth1
 exit
!
server telnet tel
 security protocol telnet
 no exec authorization
 no login authentication
 vrf v1
 exit
!
```

Hardware Router2

freeRouter hardware file: **r2-hw.txt**

```
int eth1 eth 0000.2222.0001 127.0.0.1 26021 127.0.0.1 26011
tcp2vrf 2223 v1 23
```

Software Router2

freeRouter software configuration file: **r2-sw.txt**

```
hostname r2
!
vrf definition v1
 exit
!
int eth1
vrf forwarding v1
exit
!
server telnet tel
 security protocol telnet
 no exec authorization
 no login authentication
 vrf v1
 exit
!
```

Launch freeRouters R1 & R2

java -jar <path>/rtr.jar <parameters>

parameters:

```
  router <cfg>             - start router background
  routerc <cfg>            - start router with console
  routerw <cfg>            - start router with window
  routercw <cfg>           - start router with console and window
  routers <hwcfg> <swcfg> - start router from separate configs
  routera <swcfg>          - start router with sw config
  test <cmd>               - execute test command
  show <cmd>               - execute show command
  exec <cmd>               - execute exec command
```

R1 launch with supplied r1-hw.txt and r1-sw.txt with a console prompt

```
#java -jar <path>/rtr.jar routersc <path>/r1-hw.txt <path>/r1-sw.txt
```

R2 launch with supplied r2-hw.txt and r2-sw.txt with a console prompt

```
#java -jar <path>/rtr.jar routersc <path>/r2-hw.txt <path>/r2-sw.txt
```

R1 telnet access from port 1123

```
#telnet localhost 1123
```

R2 telnet access from port 2223

```
#telnet localhost 2223
```

R1 & R2 running configuration

```
r1#?
r1#conf t
r1(cfg)#hostname r1
r1(cfg)#int ethernet1
r1(cfg-if)#vrf forwarding v1
r1(cfg-if)#ipv4 address 1.1.1.1 255.255.255.252
r1(cfg-if)#ipv6 address 1234::1 ffff:ffff:ffff:ffff::
r1(cfg-if)#desc r1@e1 -> r2@e1
```

```
r1(cfg-if)#no shut
r1(cfg-if)#end
r1#sh run
r1#sh int
```

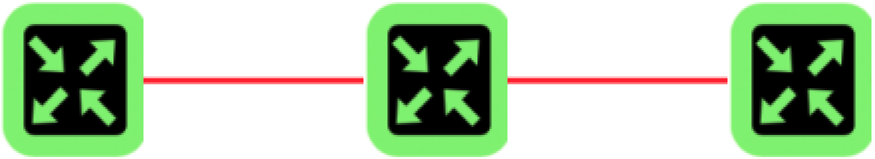
```
r2#conf t
r2(cfg)#hostname r2
r2(cfg)#int ethernet1
r2(cfg-if)#vrf forwarding v1
r2(cfg-if)#ipv4 address 1.1.1.2 255.255.255.252
r2(cfg-if)#ipv6 address 1234::2 ffff:ffff:ffff:ffff::
r2(cfg-if)#desc r2@e1 -> r1@e1
r2(cfg-if)#no shut
r2(cfg-if)#end
r2#sh run
r2#sh int
```

Connectivity test between R1 and R2 & R2 and R1

```
r1#ping 1.1.1.2 /vrf v1
r2#ping 1.1.1.1 /vrf v1
```

Implement a static network with 3 routers, use your registration number and registration year to define and assign your own IPv4 and IPv6 address.

Example:
Registration number = 45
year = 2021
Ipv4 address = 45.21.1.1 255.255.255.252
Ipv6 address = 2021:45::1 ffff:ffff:ffff:ffff::



Important configuration commands

```
router(cfg)#ipv4 route v1 <destination network> <netmask dest_network> <next hop>
router(cfg)#ipv4 route v1 0.0.0.0 0.0.0.0 1.1.1.2
router(cfg)#ipv6 route v1 <destination network> <netmask dest_network> <next hop>
router(cfg)#ipv6 route v1 :: :: 1234:2
```

Important troubleshooting commands

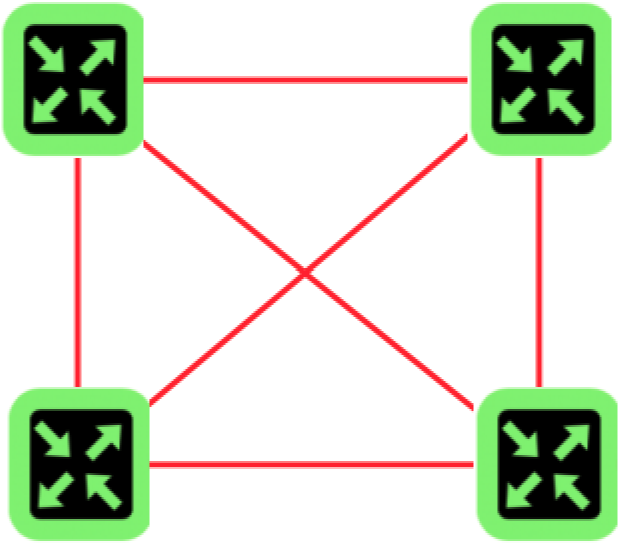
```
router#sh run
router#sh ipv4 route v1
router#sh ipv6 route v1
router#sh int
router#ping
router#traceroute
r1#ping 1.1.1.2 /vrf v1
```

Full Mesh Topology With Rip (Challenge)

implement a full mesh topology with the dynamic routing protocol RIP following the previous exercise pattern

Example:

Registration number = 45
year = 2021
Ipv4 address = 45.21.1.1 255.255.255.252
Ipv6 address = 2021:45::1 ffff:ffff:ffff:ffff::



Important configuration commands

```
router(cfg)#router rip4 <process_id>  (atribuir um número ao processo, ex. 1)
router(cfg-rip)#vrf v1
router(cfg-rip)#redistributed connect
-----
router(cfg)#int eth1
router(cfg-if)#router rip4 1 enable
-----
```

```
router(cfg)#router rip6 <process_id>
router(cfg-rip)#vrf v1
router(cfg-rip)#redistributed connect
router(cfg-rip)#net <network>
–----
router(cfg)#int eth1
router(cfg-if)#router rip6 1 enable
```

Important troubleshooting commands

```
sh run
sh ipv4 route v1
sh ipv6 route v1
sh int
ping
traceroute
r1#ping 1.1.1.2 vrf v1
```

Example of script tmux start all routers

```
#!/bin/bash
#by Everson

#Environment variable
RTR=<path freerouter "rtr.jar">
HWSW=<path files hardware and software>

tmux new-session -d -s rare 'java -jar '$RTR' routersc '$HWSW' r1-hw.txt '$HWSW' r1-sw.txt'
tmux split-window -v -t 0 -p 50
tmux send 'java -jar '$RTR' routersc '$HWSW' r2-hw.txt '$HWSW' r2-sw.txt' ENTER;
tmux split-window -h -t 0 -p 50
tmux send 'java -jar '$RTR' routersc '$HWSW' r3-hw.txt '$HWSW' r3-sw.txt' ENTER;
tmux select-layout tiled;
tmux a;
```

Example of script tmux stop all routers

```
#!/bin/bash
#by Everson

tmux kill-window -t rare
```

Tools:

Get freeRouter net-tools tarball

```
wget http://www.freertr.net/rtr-`uname -m`.tar -O rtr.tar
tar xvf rtr.tar .
```

Extra:

https://www.youtube.com/watch?v=yG6_HIRMXxE
https://youtu.be/J308z6PA86M

Link - Arquivos de Roteamento Protocolo ISIS - 25/01/2023
https://drive.google.com/file/d/1yWAqEZxyd31DtiI4ZZP2bqHcXuiBZtcg/view?usp=share_link

Link - Gravação - 25/01/2023
https://drive.google.com/file/d/1XzQ6eQOPaGiJFfwVpxGaGHH96oGKFkqO/view?usp=sharing

Reference:

http://www.freertr.net/
https://wiki.geant.org/