# Methodology – Fedora Server

Installation

Under ‘Language Support’, set it to English (New Zealand).

Under ‘Time & Date’, set the time zone to Pacific/Auckland. Manually set the date. Time doesn’t matter for now.

Under ‘Network & Host Name’, set the hostname to router1 or router2, depending on which machine you are configuring. Ignore the network configuration for now.

Under ‘Installation Destination’, make sure ‘Storage Configuration’ is set to automatic, and check ‘Free up space my removing or shrinking existing partitions’. Delete all existing partitions in the ‘Reclaim Disk Space’ pop-up, then click ‘Reclaim space’ to finish.

Under ‘User Creation’, leave the defaults but set the user and full names to router. Enter the same password used on the other systems (Vaw’ovrang6).

Initial OS Configuration

To prevent the routers crashing, the default graphics driver needs to be removed.

Blacklist Nouveau driver:

sudo bash -c "echo blacklist nouveau > /etc/modprobe.d/blacklist-nvidia-nouveau.conf"

sudo bash -c "echo options nouveau modeset=0 >> /etc/modprobe.d/blacklist-nvidia-nouveau.conf"

Remove the driver:

sudo dnf remove xorg-x11-drv-nouveau

Regenerate initramfs:

sudo dracut ---regenerate-all --force

### Package Installation

Download packages on a Fedora installation that is online:

dnf download --resolve nano netplan netplan-default-backend-networkd system-networkd chrony

Move .rpm packages to an external drive, then plug it into project PCs (routers only).

Find drive partition containing files:

lsblk

(Look for the largest partition on the removable drive. Compare with USB drive plugged in/removed to be sure you’ve selected the right drive).

Mount drive:

sudo mount /dev/sda1 /mnt

On routers, install netplan, systemd-networkd, nano, and chrony:

sudo rpm -i /mnt/packages/\*.rpm

If using DNF to install, make sure to install and enable systemd-network first:

sudo dnf install system-networkd

sudo systemctl enable system-networkd

sudo systemctl start system-networkd

Then install netplan and netplan-default-backend-networkd using the --allowerasing flag:

sudo dnf install netplan netplan-default-backend-networkd --allowerasing

Disable and mask NetworkManager:

sudo systemctl mask NetworkManager

Disable the firewall:

sudo systemctl mask --now firewalld

Confirm the firewall is disabled:

sudo systemctl status firewalld

### Network Configuration

Enable IP forwarding on both routers:

sudo nano /etc/sysctl.d/99-sysctl.conf

Add these:

net.ipv4.ip\_forward=1

net.ipv6.conf.all.forwarding=1

Next, set up netplan with static addresses:

sudo nano /etc/netplan/99-netplan.yaml

Config must be correctly indented (tab = two spaces, with each level being two spaces).

PCs will use the following IPs:

PC1 (Sender)

192.168.10.2/24

fd00:0:0:10::2/64

Router1 (eth0)

192.168.10.1/24

fd00:0:0:10::1/64

Router1 (eth1)

192.168.20.1/24

fd00:0:0:20::1/64

Router2 (eth0)

192.168.20.2/24

fd00:0:0:20::2/64

Router2 (eth1)

192.168.30.1/24

fd00:0:0:30::1/64

PC2 (Receiver)

192.168.30.2/24

fd00:0:0:30::2/64

The YAML file will be configured with these IPs.

The following is the configuration for each YAML file:

PC1:

network:

version: 2

ethernets:

eno1:

dhcp4: no

dhcp6: no

addresses:

- 192.168.10.2/24

- fd00:0:0:10::2/64

routes:

- to: 192.168.20.0/24

via: 192.168.10.1

- to: 192.168.30.0/24

via: 192.168.10.1

- to: fd00:0:0:20::/64

via: fd00:0:0:10::1

- to: fd00:0:0:30::/64

via: fd00:0:0:10::1

Router1:

network:

version: 2

renderer: networkd

ethernets:

eno1:

dhcp4: no

dhcp6: no

addresses:

- 192.168.10.1/24

- fd00:0:0:10::1/64

ens5:

dhcp4: no

dhcp6: no

addresses:

- 192.168.20.1/24

- fd00:0:0:20::1/64

routes:

- to: 192.168.30.0/24

via: 192.168.20.2

- to: fd00:0:0:30::/64

via: fd00:0:0:20::2

Router2:

network:

version: 2

renderer: networkd

ethernets:

eno1:

dhcp4: no

dhcp6: no

addresses:

- 192.168.20.2/24

- fd00:0:0:20::2/64

routes:

- to: 192.168.10.0/24

via: 192.168.20.1

- to: fd00:0:0:10::/64

via: fd00:0:0:20::1

ens5:

dhcp4: no

dhcp6: no

addresses:

- 192.168.30.1/24

- fd00:0:0:30::1/64

PC2:

network:

version: 2

ethernets:

eno1:

dhcp4: no

dhcp6: no

addresses:

- 192.168.30.2/24

- fd00:0:0:30::2/64

routes:

- to: 192.168.10.0/24

via: 192.168.30.1

- to: 192.168.20.0/24

via: 192.168.30.1

- to: fd00:0:0:10::/64

via: fd00:0:0:30::1

- to: fd00:0:0:20::/64

via: fd00:0:0:30::1

Create a new systemd service unit to apply netplan at boot:

sudo nano /etc/systemd/system/netplan-apply.service

Insert the following:

[Unit]

Description=Apply netplan configuration at boot

After=network.target

[Service]

Type=oneshot

ExecStart=/usr/bin/netplan apply

[Install]

WantedBy=multi-user.target

Reload systemd units and enable/start the new service:

sudo systemctl daemon-reload

sudo systemctl enable netplan-apply.service

sudo systemctl start netplan-apply.service

### NTP Configuration

Configure chrony:

sudo nano /etc/chrony.conf

Comment out the default NTP pools, e.g. #pool 2.fedora.pool.ntp.org iburst

Add these lines:

local stratum 8

allow 192.168.10.0/24

allow 192.168.20.0/24

allow 192.168.30.0/24

manual

server 192.168.10.1 iburst

Save and reboot.

Check chrony is configured:

chronyc tracking

chronyc sources -av

### Testing with D-ITG

D-ITG needs certain flags when ran:

-a to specify IP address

-T to specify protocol (TCP/UDP)

-c to specify packet size (in bytes)

-t to specify duration of transmission (in milliseconds)

-l to specify log name on the sender (<name>.log)

-x to specify log name on the receiver (<name>.log)

The default meter is owdm (one-way delay meter), which only produces correct data if all PCs have clocks in sync (e.g. via timesyncd or chrony).

On PC2 (Receiver), run:

ITGRecv

On PC1 (Sender), run:

For IPv4:

ITGSend -a 192.168.30.2 -T <protocol> -c <pkt\_size> -t 30000 -m rttm -l send.log -x recv.log

For IPv6:

ITGSend -a fd00:0:0:30::2 -T <protocol> -c <pkt\_size> -t 30000 -m rttm -l send.log -x recv.log

After all packets are sent, logs can be copied over (e.g. via USB or SSH).

Logs can be decrypted with ITGDec:

To plaintext:

ITGDec <name>.log -l <name>.txt

To DATs (for use with Octave or MatPlot):

All data:

ITGDec <name>.log -o <name>.dat

Delay:

ITGDec <name>.log -d 100 <name>\_delay.dat

Jitter:

ITGDec <name>.log -j 100 <name>\_jitter.dat

Packet Loss:

ITGDec <name>.log -p 100 <name>\_packetloss.dat

Throughput:

ITGDec <name>.log -b 100 <name>\_throughput.dat