# Methodology – Ubuntu Server

Installation

TBD

Initial OS Configuration  
Start by speeding up boot/login.

Disable cloud-init:

sudo touch /etc/cloud/cloud-init.disabled

Disable motd:

sudo nano /etc/default/motd-news

Change these:

ENABLED=0

URLS=""

sudo chmod -x /etc/update-motd.d/\*

Force motd to update:

sudo /usr/lib/update-notifier/update-motd-updates-available --force

Disable waiting for network on boot:

sudo mkdir -p /etc/systemd/system/systemd-networkd-wait-online.service.d/

Create override file:  
sudo nano /etc/systemd/system/systemd-networkd-wait-online.service.d/override.conf

Add these lines to override file:

[Service]

ExecStart=

ExecStart=/lib/systemd/systemd-networkd-wait-online --any

Timeout=5

Reload daemons:

sudo systemctl daemon-reload

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 apt remove xserver-xorg-video-nouveau

Regenerate initramfs:

sudo update-initramfs -u -k all

### Package Installation

Download packages on an Ubuntu installation that is online:

apt download d-itg libsctp1 frr frr-pythontools libcares2 libyang2t64

Move .deb packages to an external drive.  
Separate into client and router directories.  
Then plug it into project PCs.

Mount drive:

sudo mount /dev/sda1 /mnt

On sender/receiver, install D-ITG:

sudo dpkg -i /mnt/packages/client/\*.deb

(Optional) On routers, install frr for dynamic routing:

sudo dpkg -i /mnt/packages/router/\*.deb

### Network Configuration

First, enable IP forwarding on both routers:

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

Edit these:

net.ipv4.ip\_forward=1

net.ipv6.conf.all.forwarding=1

Make sure the firewall is disabled:

sudo systemctl disable ufw.service && sudo systemctl stop ufw.service

Confirm it’s disabled:

sudo ufw status verbose

Next, set up netplan with static addresses:

sudo nano /etc/netplan/50-cloud-init.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

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

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

### NTP Configuration

Configure chrony:

sudo nano /etc/chrony.conf

Comment out the default NTP pools, e.g. #pool 2.ubuntu.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