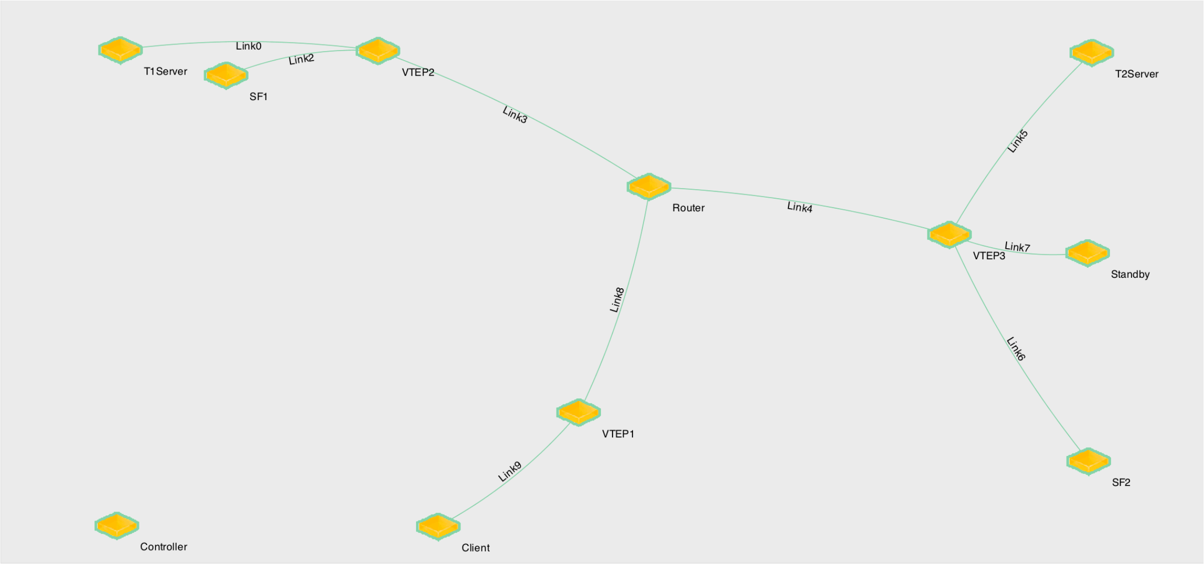
This ReadMe file contains the installation methodologies and packages to be installed on respective nodes. The corresponding codes to be run on each node is present along with the same zip file.

Topology on GENI:



Project website Link:
https://goo.gl/KA9lcI

Code names and where to run them:

1. t1s.py: This code is to be run on Tenant 1 server
2. t2s.py: This code is to be run on Tenant 2 server
3. sf1.py: This code is to be run on Service Function 1
4. sf2.py: This code is to be run on Service Function 2
5. sf1_standby: This code is to be run on Standby Service Function that would act as SF1 when SF1 fails
6. demo.py: This code is to be run on controller
7. demo_standby.py: This code is to be run on controller when we want to use standby SF instead of SF1 itself
8. probeSignal1.py: This code is to be run on Client to send a probe packet on behalf of tenant 1
9. probeSignal2.py: This code is to be run on Client to send a probe packet on behalf of tenant 2
10. sfSignal1.py: This code is to be run on Client to first send a signaling message for Client 1, followed by the payload (colored image)

11. sfSignal2.py: This code is to be run on Client to first send a signaling message for Tenant 2, followed by the payload (colored image)
12. unknownTenant.py: This code is to be run on Client, who is not subscribed to any of the services. The Client first send a signaling message for a tenant who is not subscribed, followed by the payload (colored image)

Installation method on Controller / VTEPs and OVS:

Controller:

```
apt-get update
apt-get -y install gcc make python-pip python-dev git
pip install --upgrade setuptools eventlet greenlet oslo.config webob==1.1.1
easy_install routes

git clone git://github.com/osrg/ryu.git
cd ryu; python ./setup.py install
sudo apt-get install sqlite3 libsqlite3-dev
```

VTEPs:

```
mkdir openvswitch
cd openvswitch
wget http://openvswitch.org/releases/openvswitch-2.4.0.tar.gz
tar -zxvf openvswitch-2.4.0.tar.gz
cd openvswitch-2.4.0/
sudo apt-get update
sudo apt-get -y install gcc make build-essential fakeroot debhelper autoconf automake libssl-dev pkg-config bzip2 openssl python-all procps python-qt4 python-zopeinterface python-twisted-conch dkms
DEB_BUILD_OPTIONS='parallel=2 nocheck' fakeroot debian/rules binary
cd ..
sudo dpkg -i openvswitch-common*.deb openvswitch-datapath-dkms*.deb openvswitch-testcontroller*.deb openvswitch-pki*.deb openvswitch-switch*.deb
sudo /etc/init.d/openvswitch-testcontroller stop
sudo update-rc.d openvswitch-testcontroller disable
sudo /etc/init.d/openvswitch-switch start

service neuca stop
```

Create OVS Bridge:

```
ovs-vsctl add-br br0
ovs-vsctl del-port br0 eth1
ovs-vsctl add-port br0 eth3
ovs-vsctl add-port br0 eth2
```

```
sudo ovs-vsctl set Bridge br0 protocols=OpenFlow14
```

```
ovs-ofctl show br0 -O OpenFlow14
sudo ovs-ofctl -O Openflow14 dump-flows br0
sudo ovs-ofctl -O Openflow14 del-flows br0
```

Unassign Ip address from OVswitch interfaces

```
 ifconfig eth2 down
 ifconfig eth2 0.0.0.0 up
 ifconfig eth3 down
 ifconfig eth3 0.0.0.0 up
 ifconfig eth4 down
 ifconfig eth4 0.0.0.0 up
 ifconfig eth5 down
 ifconfig eth5 0.0.0.0 up
 ovs-vsctl set-fail-mode br0 secure
 ovs-vsctl set-controller br0 tcp:10.103.0.2:6633
```

```
ovs-vsctl add-port br0 vtep -- set interface vtep type=vxlan option:remote_ip=flow option:key=flow ofport_request=10
ovs-ofctl set-frags br0 drop -O OpenFlow14
```

Add gateway IP address to reach other subnets "ip route add 10.0.0.0/8 via 10.0.1.3". We can configure OSPF also.

```
VTEP1: ovs-vsctl set bridge br0 other-config:datapath-id=0000000000000001
VTEP2:  ovs-vsctl set bridge br0 other-config:datapath-id=0000000000000002
VTEP3: ovs-vsctl set bridge br0 other-config:datapath-id=0000000000000003
```

Routers:

```
sudo apt-get update
sudo apt-get install gcc make quagga quagga-doc
```

Edit /etc/quagga/daemons file.

```
zebra=yes   <<<< -- Change to Yes
bgpd=no
ospfd=yes   <<<< -- Change to Yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
babeld=no
```

```
cp /usr/share/doc/quagga/examples/zebra.conf.sample /etc/quagga/zebra.conf
cp /usr/share/doc/quagga/examples/ospfd.conf.sample /etc/quagga/ospfd.conf
```

```
echo password cisco > /etc/quagga/zebra.conf
chown quagga.quaggavty /etc/quagga/*.conf
chmod 640 /etc/quagga/*.conf
```

Configure ospf:

Add config in "/etc/quagga/ospfd.conf" file

```
hostname R1
password cisco
!enable password please-set-at-here
!
router ospf
 network 172.168.0.0/16 area 0
 network 10.0.0.0/16 area 0
```

Stop and restart the quagga deamon

```
/etc/init.d/quagga stop
/etc/init.d/quagga start
```

Check Linux routing table and make sure all routes are dynamically learnt using OSPF.

```
route -n
root@R1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.103.0.1      0.0.0.0         UG    0      0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
10.0.2.0        10.103.0.13     255.255.255.0   UG    20     0        0 eth0
10.0.3.0        10.103.0.20     255.255.255.0   UG    20     0        0 eth0
10.103.0.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
172.168.18.0    0.0.0.0         255.255.255.252 U     0      0        0 eth2
172.168.19.0    0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.168.20.0    10.103.0.13     255.255.255.252 UG    20     0        0 eth0
root@R1:~#
```

Enable IP Forwarding: echo 1 > /proc/sys/net/ipv4/ip_forward