**Node 1 Mininet Topology Script**

#!/usr/bin/python

from mininet.net import Mininet

from mininet.node import Controller, RemoteController

from mininet.cli import CLI

from mininet.log import setLogLevel, info

def MyNet():

NODE1\_IP='10.0.0.2'

NODE2\_IP='10.0.0.3'

NODE3\_IP='172.16.15.30'

NODE4\_IP='172.16.15.44'

NODE5\_IP='172.16.57.71'

NODE6\_IP='172.16.59.59'

CONTROLLER\_IP='10.0.0.2'

net = Mininet( topo=None,

build=False)

net.addController( 'c0',

controller=RemoteController,

ip=CONTROLLER\_IP,

port=6653)

h1 = net.addHost( 'h1', ip='11.0.0.1' )

h2 = net.addHost( 'h2', ip='11.0.0.2' )

h3 = net.addHost( 'h3', ip='11.0.0.3' )

h4 = net.addHost( 'h4', ip='11.0.0.4' )

h5 = net.addHost( 'h5', ip='11.0.0.5' )

h6 = net.addHost( 'h6', ip='11.0.0.6' )

h7 = net.addHost( 'h7', ip='11.0.0.7' )

h8 = net.addHost( 'h8', ip='11.0.0.8' )

h9 = net.addHost( 'h9', ip='11.0.0.9' )

h10 = net.addHost( 'h10', ip='11.0.0.10' )

h11 = net.addHost( 'h11', ip='11.0.0.11' )

h12 = net.addHost( 'h12', ip='11.0.0.12' )

h13 = net.addHost( 'h13', ip='11.0.0.13' )

h14 = net.addHost( 'h14', ip='11.0.0.14' )

h15 = net.addHost( 'h15', ip='11.0.0.15' )

h16 = net.addHost( 'h16', ip='11.0.0.16' )

h17 = net.addHost( 'h17', ip='11.0.0.17' )

h18 = net.addHost( 'h18', ip='11.0.0.18' )

h19 = net.addHost( 'h19', ip='11.0.0.19' )

h20 = net.addHost( 'h20', ip='11.0.0.20' )

h21 = net.addHost( 'h21', ip='11.0.0.21' )

h22 = net.addHost( 'h22', ip='11.0.0.22' )

h23 = net.addHost( 'h23', ip='11.0.0.23' )

h24 = net.addHost( 'h24', ip='11.0.0.24' )

h25 = net.addHost( 'h25', ip='11.0.0.25' )

s1 = net.addSwitch( 's1' )

s2 = net.addSwitch( 's2' )

s3 = net.addSwitch( 's3' )

s4 = net.addSwitch( 's4' )

s5 = net.addSwitch( 's5' )

s6 = net.addSwitch( 's6' )

s7 = net.addSwitch( 's7' )

net.addLink( h1, s4 )

net.addLink( h2, s4 )

net.addLink( h3, s4 )

net.addLink( h4, s4 )

net.addLink( h5, s4 )

net.addLink( h6, s4 )

net.addLink( h7, s5 )

net.addLink( h8, s5 )

net.addLink( h9, s5 )

net.addLink( h10, s5 )

net.addLink( h11, s5 )

net.addLink( h12, s5 )

net.addLink( h13, s6 )

net.addLink( h14, s6 )

net.addLink( h15, s6 )

net.addLink( h16, s6 )

net.addLink( h17, s6 )

net.addLink( h18, s6 )

net.addLink( h19, s7 )

net.addLink( h20, s7 )

net.addLink( h21, s7 )

net.addLink( h22, s7 )

net.addLink( h23, s7 )

net.addLink( h24, s7 )

net.addLink( h25, s7 )

net.addLink( s1, s2 )

net.addLink( s1, s3 )

net.addLink( s2, s4 )

net.addLink( s2, s5 )

net.addLink( s3, s6 )

net.addLink( s3, s7 )

net.start()

# Configure the GRE tunnel

s1.cmd('ovs-vsctl add-port s1 s1-gre1 -- set interface s1-gre1 type=gre options:remote\_ip='+NODE1\_IP)

s1.cmdPrint('ovs-vsctl show')

s2.cmd('ovs-vsctl add-port s2 s2-gre1 -- set interface s2-gre1 type=gre options:remote\_ip='+NODE2\_IP)

s2.cmdPrint('ovs-vsctl show')

s3.cmd('ovs-vsctl add-port s3 s3-gre1 -- set interface s3-gre1 type=gre options:remote\_ip='+NODE3\_IP)

s3.cmdPrint('ovs-vsctl show')

s4.cmd('ovs-vsctl add-port s4 s4-gre1 -- set interface s4-gre1 type=gre options:remote\_ip='+NODE4\_IP)

s4.cmdPrint('ovs-vsctl show')

s5.cmd('ovs-vsctl add-port s5 s5-gre1 -- set interface s5-gre1 type=gre options:remote\_ip='+NODE5\_IP)

s5.cmdPrint('ovs-vsctl show')

s6.cmd('ovs-vsctl add-port s6 s6-gre1 -- set interface s6-gre1 type=gre options:remote\_ip='+NODE6\_IP)

s6.cmdPrint('ovs-vsctl show')

CLI( net )

net.stop()

if \_\_name\_\_ == '\_\_main\_\_':

setLogLevel( 'info' )

MyNet()

**Node 2 Mininet Topology Script**

#!/usr/bin/python

from mininet.net import Mininet

from mininet.node import Controller, RemoteController

from mininet.cli import CLI

from mininet.log import setLogLevel, info

def MyNet():

NODE1\_IP='10.0.0.1'

NODE2\_IP='10.0.0.3'

NODE3\_IP='172.16.15.30'

NODE4\_IP='172.16.15.44'

NODE5\_IP='172.16.57.71'

NODE6\_IP='172.16.59.59'

CONTROLLER\_IP='10.0.0.2'

net = Mininet( topo=None,

build=False)

net.addController( 'c0',

controller=RemoteController,

ip=CONTROLLER\_IP,

port=6653)

h26 = net.addHost( 'h26', ip='11.0.0.26' )

h27 = net.addHost( 'h27', ip='11.0.0.27' )

h28 = net.addHost( 'h28', ip='11.0.0.28' )

h29 = net.addHost( 'h29', ip='11.0.0.29' )

h30 = net.addHost( 'h30', ip='11.0.0.30' )

h31 = net.addHost( 'h31', ip='11.0.0.31' )

h32 = net.addHost( 'h32', ip='11.0.0.32' )

h33 = net.addHost( 'h33', ip='11.0.0.33' )

h34 = net.addHost( 'h34', ip='11.0.0.34' )

h35 = net.addHost( 'h35', ip='11.0.0.35' )

h36 = net.addHost( 'h36', ip='11.0.0.36' )

h37 = net.addHost( 'h37', ip='11.0.0.37' )

h38 = net.addHost( 'h38', ip='11.0.0.38' )

h39 = net.addHost( 'h39', ip='11.0.0.39' )

h40 = net.addHost( 'h40', ip='11.0.0.40' )

h41 = net.addHost( 'h41', ip='11.0.0.41' )

h42 = net.addHost( 'h42', ip='11.0.0.42' )

h43 = net.addHost( 'h43', ip='11.0.0.43' )

h44 = net.addHost( 'h44', ip='11.0.0.44' )

h45 = net.addHost( 'h45', ip='11.0.0.45' )

h46 = net.addHost( 'h46', ip='11.0.0.46' )

h47 = net.addHost( 'h47', ip='11.0.0.47' )

h48 = net.addHost( 'h48', ip='11.0.0.48' )

h49 = net.addHost( 'h49', ip='11.0.0.49' )

h50 = net.addHost( 'h50', ip='11.0.0.50' )

s26 = net.addSwitch( 's26' )

s27 = net.addSwitch( 's27' )

s28 = net.addSwitch( 's28' )

s29 = net.addSwitch( 's29' )

s30 = net.addSwitch( 's30' )

s31 = net.addSwitch( 's31' )

s32 = net.addSwitch( 's32' )

net.addLink( h26, s29 )

net.addLink( h27, s29 )

net.addLink( h28, s29 )

net.addLink( h29, s29 )

net.addLink( h30, s29 )

net.addLink( h31, s29 )

net.addLink( h32, s30 )

net.addLink( h33, s30 )

net.addLink( h34, s30 )

net.addLink( h35, s30 )

net.addLink( h36, s30 )

net.addLink( h37, s30 )

net.addLink( h38, s31 )

net.addLink( h39, s31 )

net.addLink( h40, s31 )

net.addLink( h41, s31 )

net.addLink( h42, s31 )

net.addLink( h43, s31 )

net.addLink( h44, s32 )

net.addLink( h45, s32 )

net.addLink( h46, s32 )

net.addLink( h47, s32 )

net.addLink( h48, s32 )

net.addLink( h49, s32 )

net.addLink( h50, s32 )

net.addLink( s26, s27 )

net.addLink( s26, s28 )

net.addLink( s27, s29 )

net.addLink( s27, s30 )

net.addLink( s28, s31 )

net.addLink( s28, s32 )

net.start()

# Configure the GRE tunnel

s26.cmd('ovs-vsctl add-port s26 s26-gre1 -- set interface s26-gre1 type=gre options:remote\_ip='+NODE1\_IP)

s26.cmdPrint('ovs-vsctl show')

s27.cmd('ovs-vsctl add-port s27 s27-gre1 -- set interface s27-gre1 type=gre options:remote\_ip='+NODE2\_IP)

s27.cmdPrint('ovs-vsctl show')

s28.cmd('ovs-vsctl add-port s28 s28-gre1 -- set interface s28-gre1 type=gre options:remote\_ip='+NODE3\_IP)

s28.cmdPrint('ovs-vsctl show')

s29.cmd('ovs-vsctl add-port s29 s29-gre1 -- set interface s29-gre1 type=gre options:remote\_ip='+NODE4\_IP)

s29.cmdPrint('ovs-vsctl show')

s30.cmd('ovs-vsctl add-port s30 s30-gre1 -- set interface s30-gre1 type=gre options:remote\_ip='+NODE5\_IP)

s30.cmdPrint('ovs-vsctl show')

s31.cmd('ovs-vsctl add-port s31 s31-gre1 -- set interface s31-gre1 type=gre options:remote\_ip='+NODE6\_IP)

s31.cmdPrint('ovs-vsctl show')

CLI( net )

net.stop()

if \_\_name\_\_ == '\_\_main\_\_':

setLogLevel( 'info' )

MyNet()

**Mininet topology Implementation and GRE tunnel Configuration**

Mininet is tool which is used to create virtual SDN network topologies. It provides a way to build a custom topology with any number of host and switches. We can also connect a remote controller with our custom topology using only the controllers IP Address.

**Installation of Mininet**

To install mininet first clone the mininet github repository to your system using the following command

# git clone git://github.com/mininet/mininet

Then run apt-get command

# sudo apt-get install mininet

Then to check whether mininet is installed properly run the following command

# sudo mn

This command will the default topology with 1 switch and 2 host connected with a default OVS controller. You can ping host 1 and host 2 to check connectivity

# h1 ping h2

**Writing the Mininet Topology Script**

To start first import the Mininet libraries in the script.

from mininet.net import Mininet

from mininet.node import Controller, RemoteController

from mininet.cli import CLI

from mininet.log import setLogLevel, info

Then create a Mininet Object and attach a controller to this object.

net = Mininet( topo=None, build=False)

net.addController( 'c0', controller=RemoteController, ip=CONTROLLER\_IP, port=6653)

Here you should provide the IP Address of the controller where it is hosted and the port number on which it is connected.

Then to add host you should call the addHost function of Mininet object. If you required more host then you can call addHost function multiple times.

h1 = net.addHost( 'h1', ip='11.0.0.1' )

To add switches in your topology you can call the addSwitch function of Mininet object.

s1 = net.addSwitch( 's1' )

To add link between host and switch or to add link between two switches we can call addLink function of Mininet of Object specifying the entities which have be connected in its parameters.

net.addLink( h1, s4 )

At the end we have to start the topology which can be used to test our topology

net.start()

CLI( net )

net.stop()

**Connecting two different Mininet Topologies on two Different Physical Machines Using GRE Tunnels**

To connect two mininet topologies which are running on two different physical machines we can use GRE tunnel which help us to build a tunnel between two machines and route the packets through it.

To add GRE tunnel in our mininet topology you will choose a switch that will build a tunnel with another switch on second machine. Then you will run this command on this switch:

s1.cmd('ovs-vsctl add-port s1 s1-gre1 -- set interface s1-gre1 type=gre options:remote\_ip='+NODE\_IP)

s1.cmdPrint('ovs-vsctl show')

Here NODE\_IP represent the IP Address of the other machine to which we are connecting. This command has to run on both machine with NODE\_IP representing the IP Address of the other machine.