INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KANCHEEPURAM

DCN-8

Name : K.Nithesh

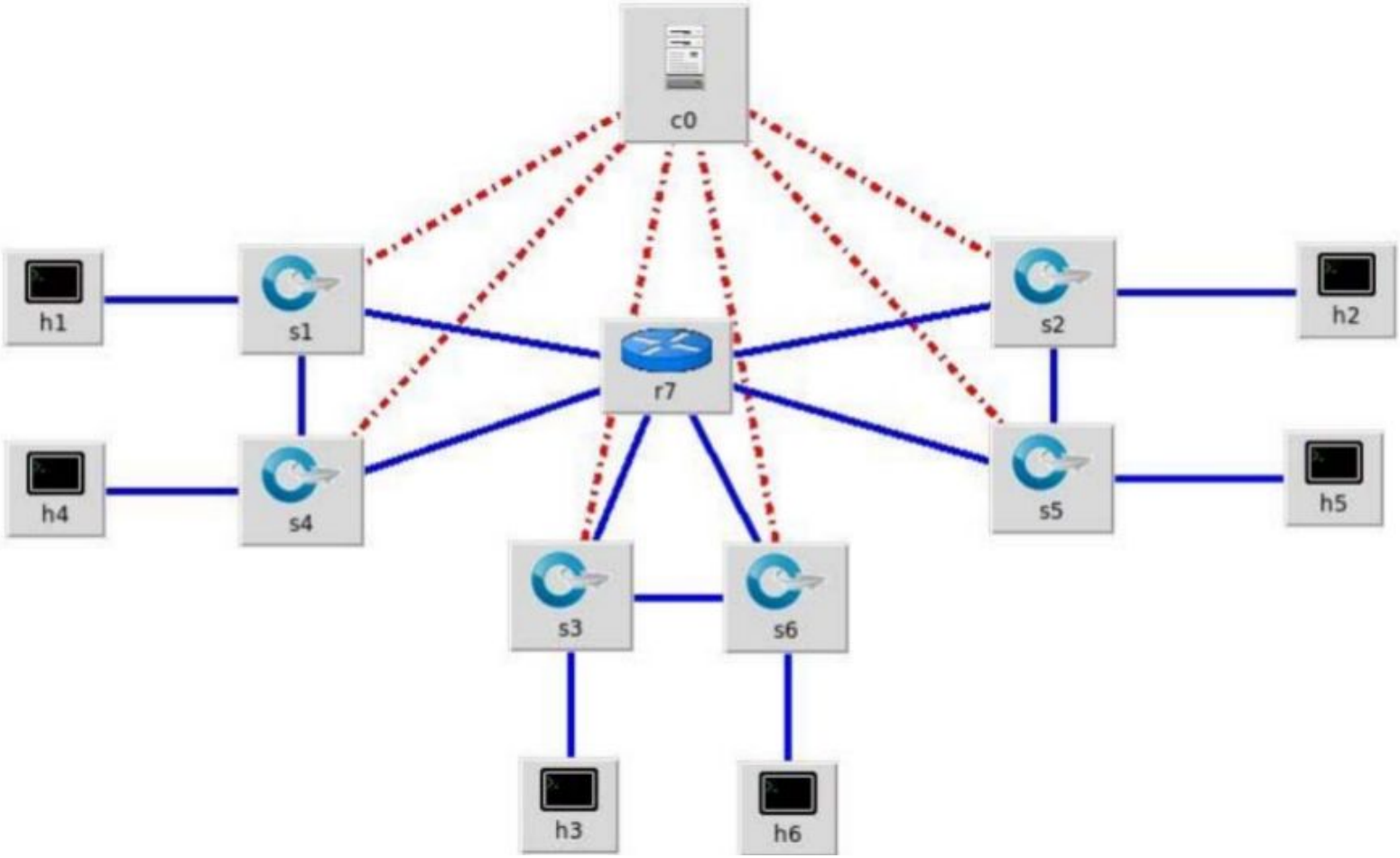
Roll N0: ESD19I008

AIM : To create linux router and subnetting

Theory:

Subnetting:

A subnetwork or subnet is a logical subdivision of an IP network. **The practice of dividing a network into two or more networks** is called subnetting. Computers that belong to the same subnet are addressed with an identical most-significant bit-group in their IP addresses.

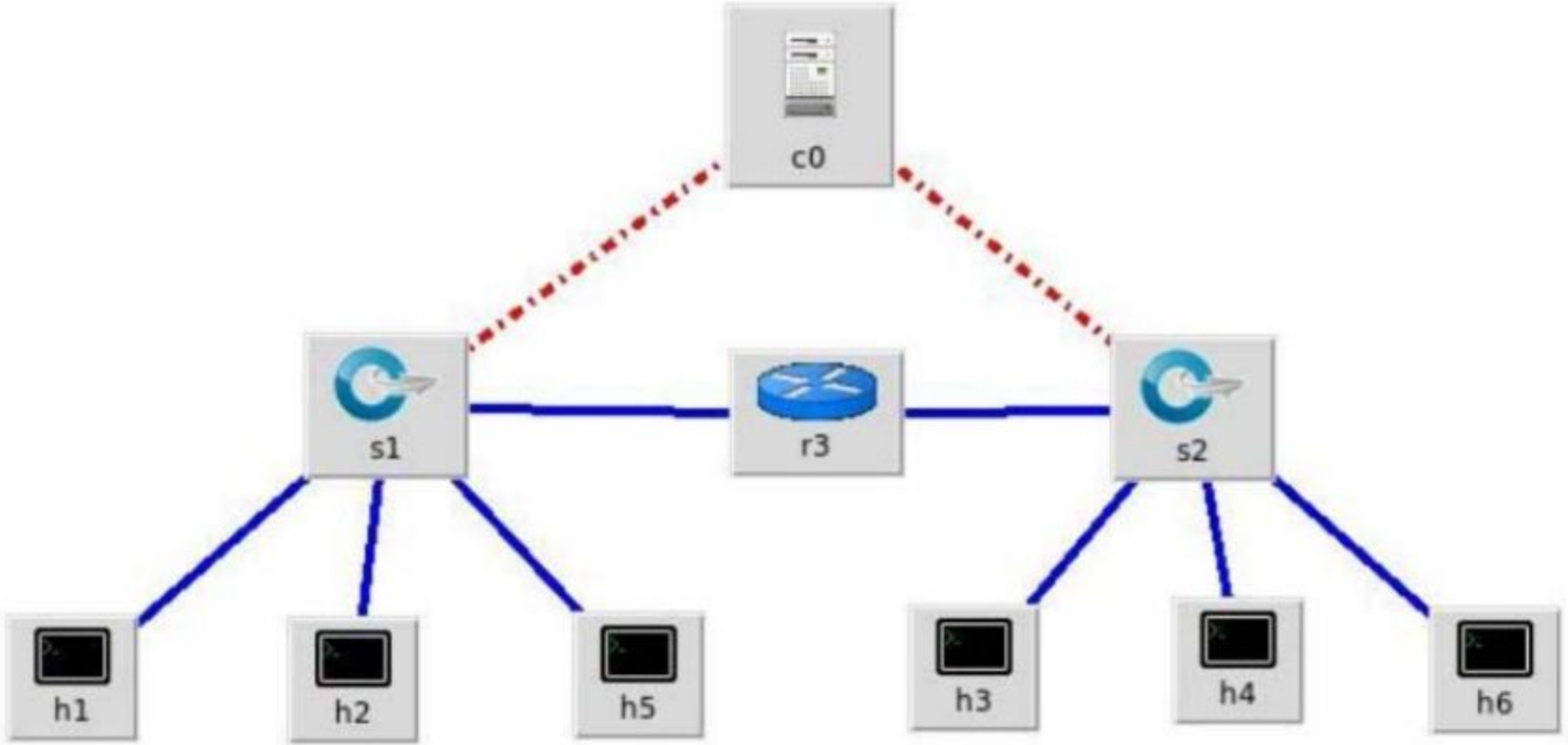

c0
h1
s1
r7
s2
h2
h4
s4
s5
h5
s3
s6
h3
h6

```
defaultIP = '192.168.1.1/24'  # IP address for r0-eth1
router = self.addNode( 'r0', cls=LinuxRouter, ip=defaultIP )

s1, s2, s3, s4, s5, s6 = [ self.addSwitch( s ) for s in 's1', 's2', 's3', 's4', 's5', 's6' ]

self.addLink( s1, router, intfName2='r0-eth1',
              params2={ 'ip' : defaultIP } )  # for clarity
self.addLink( s2, router, intfName2='r0-eth2',
              params2={ 'ip' : '172.16.0.1/12' } )
self.addLink( s3, router, intfName2='r0-eth3',
              params2={ 'ip' : '10.0.0.1/8' } )
self.addLink( s4, router, intfName2='r0-eth4',
              params2={ 'ip' : defaultIP } )
self.addLink( s5, router, intfName2='r0-eth5',
              params2={ 'ip' : '172.16.0.1/12' } )
self.addLink( s6, router, intfName2='r0-eth6',
              params2={ 'ip' : '10.0.0.1/8' } )

h1 = self.addHost( 'h1', ip='192.168.1.100/24',
                   defaultRoute='via 192.168.1.1' )
h2 = self.addHost( 'h2', ip='172.16.0.100/12',
                   defaultRoute='via 172.16.0.1' )
h3 = self.addHost( 'h3', ip='10.0.0.100/8',
                   defaultRoute='via 10.0.0.1' )
h4 = self.addHost( 'h4', ip='192.168.1.100/24',
                   defaultRoute='via 192.168.1.1' )
h5 = self.addHost( 'h5', ip='172.16.0.100/12',
                   defaultRoute='via 172.16.0.1' )
h6 = self.addHost( 'h6', ip='10.0.0.100/8',
                   defaultRoute='via 10.0.0.1' )

for h, s in [ (h1, s1), (h2, s2), (h3, s3), (h4, s4), (h5, s5), (h6, s6), (s1, s4), (s3, s6), (s2, s5) ]:
    self.addLink( h, s )
```

mininet@mininet-vm: ~/mininet/examples

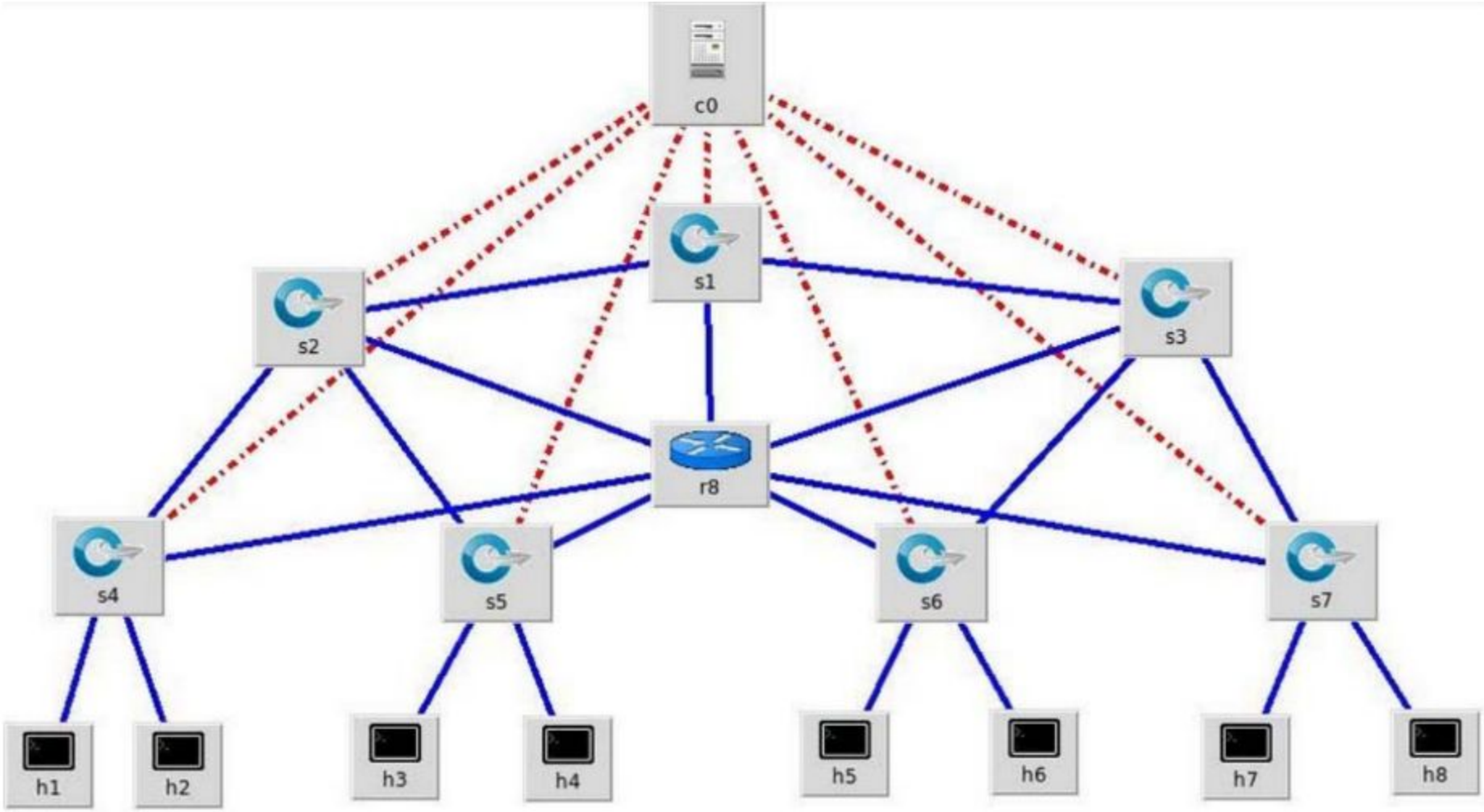
```
login as: mininet
mininet@192.168.56.101's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Sun Apr 10 14:59:16 2022 from 192.168.56.1
mininet@mininet-vm:~$ cd mininet
mininet@mininet-vm:~/mininet$ cd examples
mininet@mininet-vm:~/mininet/examples$ sudo python nitheshrouter.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 r0
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (h5, s5) (h6, s6) (s1, r0) (s1, s4) (s2, r0)
 (s2, s5) (s3, r0) (s3, s6) (s4, r0) (s5, r0) (s6, r0)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 r0
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Routing Table on Router:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        *               255.0.0.0       U     0      0        0 r0-eth3
10.0.0.0        *               255.0.0.0       U     0      0        0 r0-eth6
172.16.0.0      *               255.240.0.0     U     0      0        0 r0-eth2
172.16.0.0      *               255.240.0.0     U     0      0        0 r0-eth5
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth1
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth4
```

```
mininet@mininet-vm: ~/mininet/examples
login as: mininet
mininet@192.168.56.101's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Sun Apr 10 14:59:16 2022 from 192.168.56.1
mininet@mininet-vm:~$ cd mininet
mininet@mininet-vm:~/mininet$ cd examples
mininet@mininet-vm:~/mininet/examples$ sudo python nitheshrouter.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 r0
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (h5, s5) (h6, s6) (s1, r0) (s1, s4) (s2, r0)
 (s2, s5) (s3, r0) (s3, s6) (s4, r0) (s5, r0) (s6, r0)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 r0
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Routing Table on Router:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        *               255.0.0.0       U     0      0        0 r0-eth3
10.0.0.0        *               255.0.0.0       U     0      0        0 r0-eth6
172.16.0.0      *               255.240.0.0     U     0      0        0 r0-eth2
172.16.0.0      *               255.240.0.0     U     0      0        0 r0-eth5
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth1
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth4

*** Starting CLI:
mininet> h1 ping h2
PING 172.16.0.100 (172.16.0.100) 56(84) bytes of data.
64 bytes from 172.16.0.100: icmp_seq=1 ttl=63 time=8.69 ms
64 bytes from 172.16.0.100: icmp_seq=2 ttl=63 time=0.326 ms
64 bytes from 172.16.0.100: icmp_seq=3 ttl=63 time=0.056 ms
64 bytes from 172.16.0.100: icmp_seq=4 ttl=63 time=0.054 ms
64 bytes from 172.16.0.100: icmp_seq=5 ttl=63 time=0.060 ms
^C
--- 172.16.0.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.054/1.837/8.693/3.429 ms
mininet> h5 ping h4
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=7 ttl=63 time=1.80 ms
64 bytes from 192.168.1.100: icmp_seq=8 ttl=63 time=0.379 ms
64 bytes from 192.168.1.100: icmp_seq=9 ttl=63 time=0.063 ms
64 bytes from 192.168.1.100: icmp_seq=10 ttl=63 time=0.065 ms
64 bytes from 192.168.1.100: icmp_seq=11 ttl=63 time=0.061 ms
64 bytes from 192.168.1.100: icmp_seq=12 ttl=63 time=0.066 ms
^C
--- 192.168.1.100 ping statistics ---
12 packets transmitted, 6 received, 50% packet loss, time 11007ms
rtt min/avg/max/mdev = 0.061/0.406/1.803/0.635 ms
mininet>
```


c0
s1
r3
s2
h1
h2
h5
h3
h4
h6

```
defaultIP = '192.168.1.1/24'  # IP address for r0-eth1
router = self.addNode( 'r0', cls=LinuxRouter, ip=defaultIP )

s1, s2 = [ self.addSwitch( s ) for s in 's1', 's2' ]

self.addLink( s1, router, intfName2='r0-eth1',
              params2={ 'ip' : defaultIP } )  # for clarity
self.addLink( s2, router, intfName2='r0-eth2',
              params2={ 'ip' : '172.16.0.1/12' } )


h1 = self.addHost( 'h1', ip='192.168.1.100/24',
                   defaultRoute='via 192.168.1.1' )
h2 = self.addHost( 'h2', ip='192.168.1.101/24',
                   defaultRoute='via 192.168.1.1' )
h3 = self.addHost( 'h3', ip='172.16.0.100/12',
                   defaultRoute='via 172.16.0.1' )
h4 = self.addHost( 'h4', ip='172.16.0.101/12',
                   defaultRoute='via 172.16.0.1' )
h5 = self.addHost( 'h5', ip='192.168.1.102/24',
                   defaultRoute='via 192.168.1.1' )
h6 = self.addHost( 'h6', ip='172.16.0.102/12',
                   defaultRoute='via 172.16.0.1' )

for h, s in [ (h1, s1), (h2, s1), (h3, s2), (h4, s2), (h5, s1), (h6, s2) ]:
    self.addLink( h, s )
```

```
mininet@mininet-vm:~/mininet/examples$ sudo python nitheshrouter2.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 r0
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (h5, s1) (h6, s2) (s1, r0) (s2, r0)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 r0
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Routing Table on Router:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.16.0.0      *               255.240.0.0     U     0      0        0 r0-eth2
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth1

*** Starting CLI:
mininet> h1 ping h6
PING 172.16.0.102 (172.16.0.102) 56(84) bytes of data.
64 bytes from 172.16.0.102: icmp_seq=1 ttl=63 time=6.14 ms
64 bytes from 172.16.0.102: icmp_seq=2 ttl=63 time=0.314 ms
64 bytes from 172.16.0.102: icmp_seq=3 ttl=63 time=0.092 ms
^C
--- 172.16.0.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.092/2.184/6.146/2.803 ms
mininet>
```


c0
s1
s2
s3
r8
s4
s5
s6
s7
h1
h2
h3
h4
h5
h6
h7
h8

```
s1, s2, s3, s4, s5, s6, s7 = [ self.addSwitch( s ) for s in 's1', 's2', 's3', 's4', 's5', 's6', 's7' ]

self.addLink( s1, router, intfName2='r0-eth1',
              params2={ 'ip' : defaultIP } )  # for clarity
self.addLink( s2, router, intfName2='r0-eth2',
              params2={ 'ip' : '192.168.1.64/26' } )
self.addLink( s3, router, intfName2='r0-eth3',
              params2={ 'ip' : '192.168.1.128/26' } )
self.addLink( s4, router, intfName2='r0-eth4',
              params2={ 'ip' : '192.168.1.80/28' } )
self.addLink( s5, router, intfName2='r0-eth5',
              params2={ 'ip' : '192.168.1.96/28' } )
self.addLink( s6, router, intfName2='r0-eth6',
              params2={ 'ip' : '192.168.1.144/28' } )
self.addLink( s7, router, intfName2='r0-eth7',
              params2={ 'ip' : '192.168.1.160/28' } )

h1 = self.addHost( 'h1', ip='192.168.1.81',
                   defaultRoute='via 192.168.1.80' )
h2 = self.addHost( 'h2', ip='192.168.1.82',
                   defaultRoute='via 192.168.1.80' )
h3 = self.addHost( 'h3', ip='192.168.1.97',
                   defaultRoute='via 192.168.1.96' )
h4 = self.addHost( 'h4', ip='192.168.1.98',
                   defaultRoute='via 192.168.1.96' )
h5 = self.addHost( 'h5', ip='192.168.1.145',
                   defaultRoute='via 192.168.1.144' )
h6 = self.addHost( 'h6', ip='192.168.1.146',
                   defaultRoute='via 192.168.1.144' )
h7 = self.addHost( 'h7', ip='192.168.1.161',
                   defaultRoute='via 192.168.1.160' )
h8 = self.addHost( 'h8', ip='192.168.1.162',
                   defaultRoute='via 192.168.1.160' )
```

```
mininet@mininet-vm:~/mininet/examples$ sudo python nitheshrouter3.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 r0
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(h1, s4) (h2, s4) (h3, s5) (h4, s5) (h5, s6) (h6, s6) (h7, s7) (h8, s7) (s1, r0)
 (s2, r0) (s2, s1) (s3, r0) (s3, s1) (s4, r0) (s4, s2) (s5, r0) (s5, s2) (s6, r0
) (s6, s3) (s7, r0) (s7, s3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 r0
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Routing Table on Router:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth1
192.168.1.64    *               255.255.255.192 U     0      0        0 r0-eth2
192.168.1.80    *               255.255.255.240 U     0      0        0 r0-eth4
192.168.1.96    *               255.255.255.240 U     0      0        0 r0-eth5
192.168.1.128   *               255.255.255.192 U     0      0        0 r0-eth3
192.168.1.144   *               255.255.255.240 U     0      0        0 r0-eth6
192.168.1.160   *               255.255.255.240 U     0      0        0 r0-eth7
```

```
mininet@mininet-vm:~/mininet/examples$ sudo python nitheshrouter3.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 r0
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(h1, s4) (h2, s4) (h3, s5) (h4, s5) (h5, s6) (h6, s6) (h7, s7) (h8, s7) (s1, r0)
 (s2, r0) (s2, s1) (s3, r0) (s3, s1) (s4, r0) (s4, s2) (s5, r0) (s5, s2) (s6, r0
) (s6, s3) (s7, r0) (s7, s3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 r0
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Routing Table on Router:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.0   U     0      0        0 r0-eth1
192.168.1.64    *               255.255.255.192 U     0      0        0 r0-eth2
192.168.1.80    *               255.255.255.240 U     0      0        0 r0-eth4
192.168.1.96    *               255.255.255.240 U     0      0        0 r0-eth5
192.168.1.128   *               255.255.255.192 U     0      0        0 r0-eth3
192.168.1.144   *               255.255.255.240 U     0      0        0 r0-eth6
192.168.1.160   *               255.255.255.240 U     0      0        0 r0-eth7

*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 r0
h2 -> h1 h3 h4 h5 h6 h7 h8 r0
h3 -> h1 h2 h4 h5 h6 h7 h8 r0
h4 -> h1 h2 h3 h5 h6 h7 h8 r0
h5 -> h1 h2 h3 h4 h6 h7 h8 r0
h6 -> h1 h2 h3 h4 h5 h7 h8 r0
h7 -> h1 h2 h3 h4 h5 h6 h8 r0
h8 -> h1 h2 h3 h4 h5 h6 h7 r0
r0 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
```

Result:

Created subnetworks

Linux router is created using python code