Laboratory 6: Quagga Internet Router

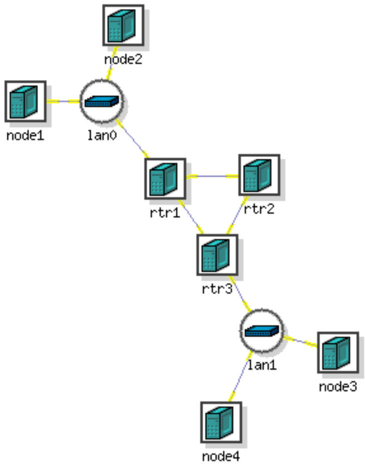
Esha Desai

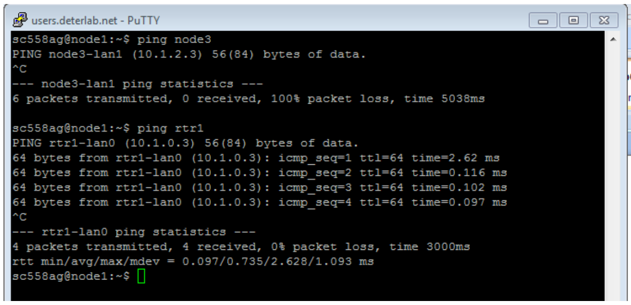
USC ID: 6993245898

In this tutorial I have learnt that Quagga software runs on top of most modern UNIX-like operating systems and is made up of several encapsulated software pieces called daemons. Quagga does not actually route packets, it uses information from the implemented protocols to update the kernel routing table. Zebra is a piece of quagga software,it is the management daemon responsible for updating the kernel routing table. When used with the other routing daemons (ospfd, ripd, bgpd) zebra takes updates from these daemons and applies them to the kernel routing table. In this tutorial we use zebra for managing static routes.

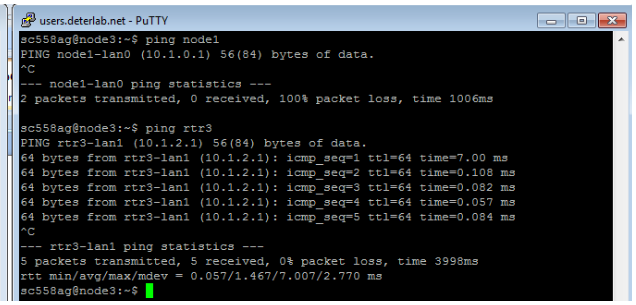
Part 1

The topology constructed by the provided NS file is as follows:





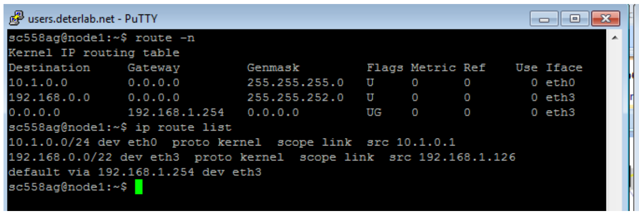
```
sc558ag@node1:~$ ping node3
PING node3-lan1 (10.1.2.3) 56(84) bytes of data.
^C
--- node3-lan1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5038ms

sc558ag@node1:~$ ping rtr1
PING rtr1-lan0 (10.1.0.3) 56(84) bytes of data.
64 bytes from rtr1-lan0 (10.1.0.3): icmp_seq=1 ttl=64 time=2.62 ms
64 bytes from rtr1-lan0 (10.1.0.3): icmp_seq=2 ttl=64 time=0.116 ms
64 bytes from rtr1-lan0 (10.1.0.3): icmp_seq=3 ttl=64 time=0.102 ms
64 bytes from rtr1-lan0 (10.1.0.3): icmp_seq=4 ttl=64 time=0.097 ms
^C
--- rtr1-lan0 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.097/0.735/2.628/1.093 ms
sc558ag@node1:~$
```



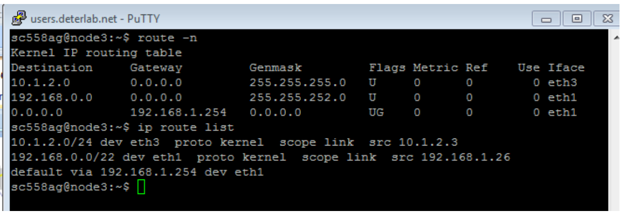
```
sc558ag@node3:~$ ping node1
PING node1-lan0 (10.1.0.1) 56(84) bytes of data.
^C
--- node1-lan0 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1006ms

sc558ag@node3:~$ ping rtr3
PING rtr3-lan1 (10.1.2.1) 56(84) bytes of data.
64 bytes from rtr3-lan1 (10.1.2.1): icmp_seq=1 ttl=64 time=7.00 ms
64 bytes from rtr3-lan1 (10.1.2.1): icmp_seq=2 ttl=64 time=0.108 ms
64 bytes from rtr3-lan1 (10.1.2.1): icmp_seq=3 ttl=64 time=0.082 ms
64 bytes from rtr3-lan1 (10.1.2.1): icmp_seq=4 ttl=64 time=0.057 ms
64 bytes from rtr3-lan1 (10.1.2.1): icmp_seq=5 ttl=64 time=0.084 ms
^C
--- rtr3-lan1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.057/1.467/7.007/2.770 ms
sc558ag@node3:~$
```

Node1 cannot ping Node3. But can ping Router1. While Node3 cannot ping node1 but it can ping router3. Being on the same LAN, the nodes on one LAN can communicate with each other but both the LANs cannot communicate yet.

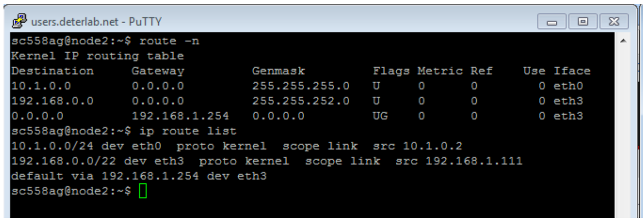
Kernel IP Routing tables for all nodes **before** changing the default route to their default routers. Default gateway is 192.168.1.254.



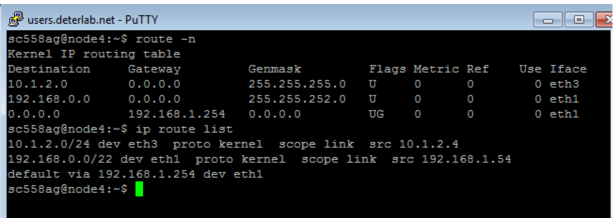
```
sc558ag@node1:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth3
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth3
sc558ag@node1:~$ ip route list
10.1.0.0/24 dev eth0  proto kernel  scope link  src 10.1.0.1
192.168.0.0/22 dev eth3  proto kernel  scope link  src 192.168.1.126
default via 192.168.1.254 dev eth3
sc558ag@node1:~$
```



```
sc558ag@node3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@node3:~$ ip route list
10.1.2.0/24 dev eth3  proto kernel  scope link  src 10.1.2.3
192.168.0.0/22 dev eth1  proto kernel  scope link  src 192.168.1.26
default via 192.168.1.254 dev eth1
sc558ag@node3:~$
```

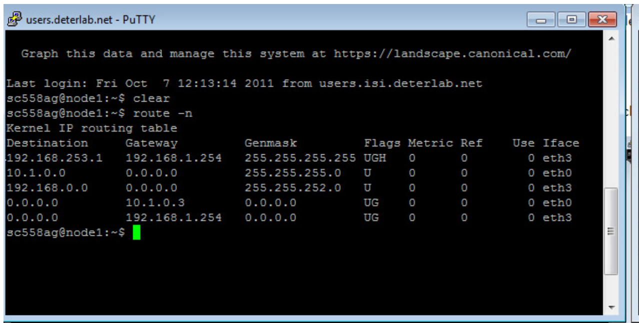


```
sc558ag@node2:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth3
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth3
sc558ag@node2:~$ ip route list
10.1.0.0/24 dev eth0  proto kernel  scope link  src 10.1.0.2
192.168.0.0/22 dev eth3  proto kernel  scope link  src 192.168.1.111
default via 192.168.1.254 dev eth3
sc558ag@node2:~$
```

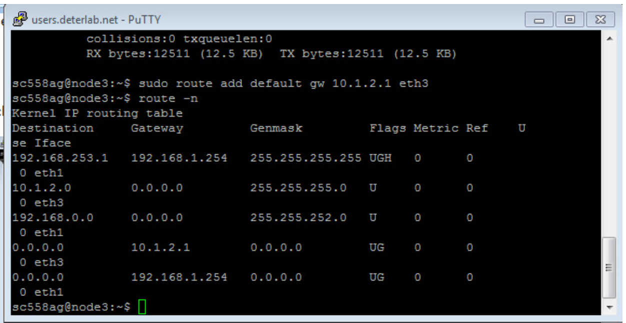


```
sc558ag@node4:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@node4:~$ ip route list
10.1.2.0/24 dev eth3  proto kernel  scope link  src 10.1.2.4
192.168.0.0/22 dev eth1  proto kernel  scope link  src 192.168.1.54
default via 192.168.1.254 dev eth1
sc558ag@node4:~$
```

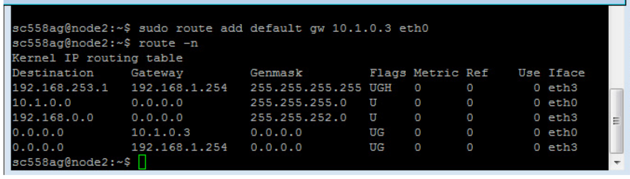
Kernel IP Routing tables for all nodes **after** changing the default routes to their respective routers.



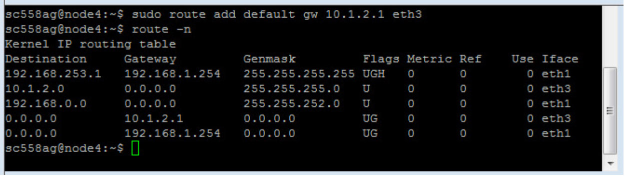
```

  Graph this data and manage this system at https://landscape.canonical.com/

Last login: Fri Oct  7 12:13:14 2011 from users.isi.deterlab.net
sc558ag@node1:~$ clear
sc558ag@node1:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.253.1   192.168.1.254   255.255.255.255 UGH   0      0        0 eth3
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth3
0.0.0.0         10.1.0.3        0.0.0.0         UG    0      0        0 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth3
sc558ag@node1:~$
```



```
          collisions:0 txqueuelen:0
          RX bytes:12511 (12.5 KB)  TX bytes:12511 (12.5 KB)

sc558ag@node3:~$ sudo route add default gw 10.1.2.1 eth3
sc558ag@node3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    U
se Iface
192.168.253.1   192.168.1.254   255.255.255.255 UGH   0      0
 0 eth1
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0
 0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0
 0 eth1
0.0.0.0         10.1.2.1        0.0.0.0         UG    0      0
 0 eth3
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0
 0 eth1
sc558ag@node3:~$
```



```

sc558ag@node2:~$ sudo route add default gw 10.1.0.3 eth0
sc558ag@node2:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.253.1   192.168.1.254   255.255.255.255 UGH   0      0        0 eth3
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth3
0.0.0.0         10.1.0.3        0.0.0.0         UG    0      0        0 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth3
sc558ag@node2:~$
```



```
sc558ag@node4:~$ sudo route add default gw 10.1.2.1 eth3
sc558ag@node4:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.253.1   192.168.1.254   255.255.255.255 UGH   0      0        0 eth1
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         10.1.2.1        0.0.0.0         UG    0      0        0 eth3
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@node4:~$
```

After building Quagga software on all the routers:

Zebra.conf Configuration file for **Router 1**:

```
!
! Zebra configuration saved from vty
!   2011/10/07 21:30:28
!
hostname zebrad
password zebra
```

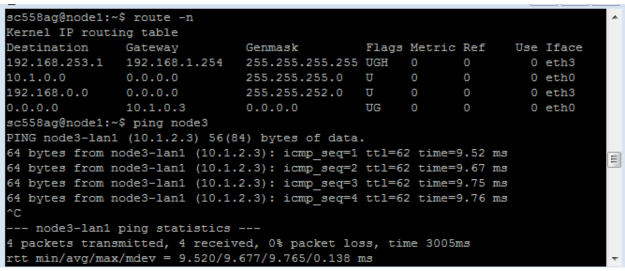
```
log stdout
!
interface eth0
 ip address 10.10.1.0/24
 ipv6 nd suppress-ra
!
interface eth1
 ipv6 nd suppress-ra
!
interface eth2
 ipv6 nd suppress-ra
!
interface eth3
 ip address 10.1.0.0/24
 ipv6 nd suppress-ra
!
interface eth4
 ipv6 nd suppress-ra
!
interface eth5
 ip address 10.10.2.0/24
 ipv6 nd suppress-ra
!
interface lo
!
ip route 10.1.2.0/24 10.10.1.2
!
ip forwarding
!
!
line vty
!
```

Zebra.conf Configuration file for **Router 3**:

```
!

! Zebra configuration saved from vty
!   2011/10/07 21:29:37
!
hostname zebrad
password zebra
log stdout
!
interface eth0
 ip address 10.10.1.0/24
 ipv6 nd suppress-ra
!
interface eth1
 ipv6 nd suppress-ra
```

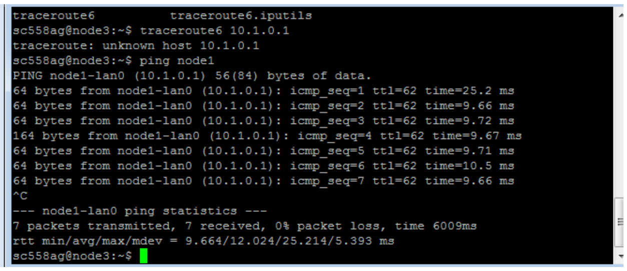
```
!
interface eth2
 ipv6 nd suppress-ra
!
interface eth3
 ip address 10.1.2.0/24
 ipv6 nd suppress-ra
!
interface eth4
 ipv6 nd suppress-ra
!
interface eth5
 ip address 10.10.3.0/24
 ipv6 nd suppress-ra
!
interface lo
!
ip route 10.1.0.0/24 10.10.1.1
!
ip forwarding
!
!
line vty
!
```

PING from node1 to node3



```
sc558ag@node1:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.253.1   192.168.1.254   255.255.255.255 UGH   0      0        0 eth3
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth3
0.0.0.0         10.1.0.3        0.0.0.0         UG    0      0        0 eth0
sc558ag@node1:~$ ping node3
PING node3-lan1 (10.1.2.3) 56(84) bytes of data.
64 bytes from node3-lan1 (10.1.2.3): icmp_seq=1 ttl=62 time=9.52 ms
64 bytes from node3-lan1 (10.1.2.3): icmp_seq=2 ttl=62 time=9.67 ms
64 bytes from node3-lan1 (10.1.2.3): icmp_seq=3 ttl=62 time=9.75 ms
64 bytes from node3-lan1 (10.1.2.3): icmp_seq=4 ttl=62 time=9.76 ms
^C
--- node3-lan1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 9.520/9.677/9.765/0.138 ms
```

PING from node3 to node1



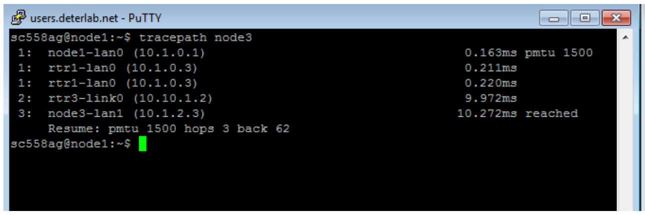
```
traceroute6          traceroute6.iputils
sc558ag@node3:~$ traceroute6 10.1.0.1
traceroute: unknown host 10.1.0.1
sc558ag@node3:~$ ping node1
PING node1-lan0 (10.1.0.1) 56(84) bytes of data.
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=1 ttl=62 time=25.2 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=2 ttl=62 time=9.66 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=3 ttl=62 time=9.72 ms
164 bytes from node1-lan0 (10.1.0.1): icmp_seq=4 ttl=62 time=9.67 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=5 ttl=62 time=9.71 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=6 ttl=62 time=10.5 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=7 ttl=62 time=9.66 ms
^C
--- node1-lan0 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 9.664/12.024/25.214/5.393 ms
sc558ag@node3:~$
```

RTT turned out to be about 10ms as expected.

According to the ns file script, the delay on links on LAN0 and LAN1 is 0ms. And the delay on the link between router1 and router3 is 5ms each. So the RTT matches with the practical observations.

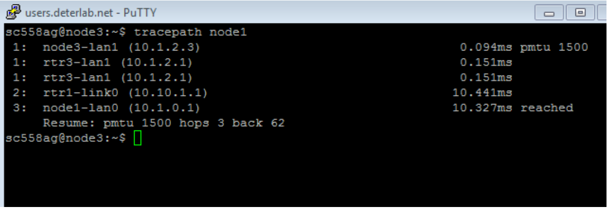
0ms+0ms+5ms+0ms+0ms+0ms+0ms+5ms+0ms+0ms=10ms

Tracepath: from node 1 to node 3



```
sc558ag@node1:~$ tracepath node3
 1:  node1-lan0 (10.1.0.1)                                 0.163ms pmtu 1500
 1:  rtr1-lan0 (10.1.0.3)                                  0.211ms
 1:  rtr1-lan0 (10.1.0.3)                                  0.220ms
 2:  rtr3-link0 (10.10.1.2)                                9.972ms
 3:  node3-lan1 (10.1.2.3)                                10.272ms reached
     Resume: pmtu 1500 hops 3 back 62
sc558ag@node1:~$
```

Tracepath from node 3 to node 1



```
sc558ag@node3:~$ tracepath node1
 1:  node3-lan1 (10.1.2.3)                                 0.094ms pmtu 1500
 1:  rtr3-lan1 (10.1.2.1)                                  0.151ms
 1:  rtr3-lan1 (10.1.2.1)                                  0.151ms
 2:  rtr1-link0 (10.10.1.1)                               10.441ms
 3:  node1-lan0 (10.1.0.1)                                10.327ms reached
     Resume: pmtu 1500 hops 3 back 62
sc558ag@node3:~$
```

The commands used to configure the routes were:

At router1#*enable*

> *Zebrad(config)#configure terminal*
> *Zebrad(config)#ip route 10.1.2.0/24 10.10.1.2*
> *Zebrad(config)#show running-config*
> *Zebrad(config)#quit*

copy running-config startup-config

Configuration saved to /mnt/conf/zebra.conf

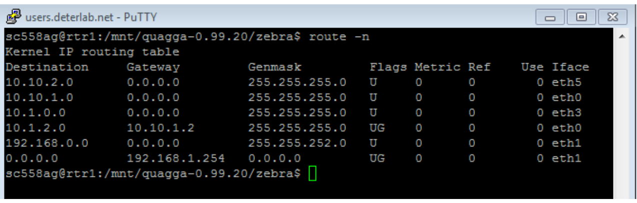
At router3#*enable*

> *Zebrad(config)#configure terminal*
> *Zebrad(config)#ip route 10.1.0.0/24 10.10.1.1*
> *Zebrad(config)#show running-config*
> *Zebrad(config)#quit*

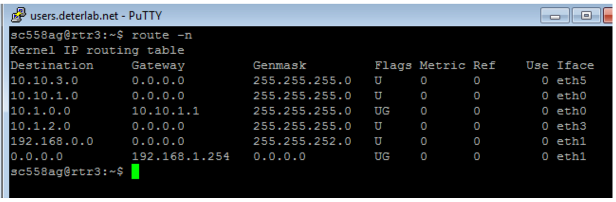
copy running-config startup-config

Configuration saved to /mnt/conf/zebra.conf

The routing tables of rtr1 and rtr3 after configuration and **pinging**:

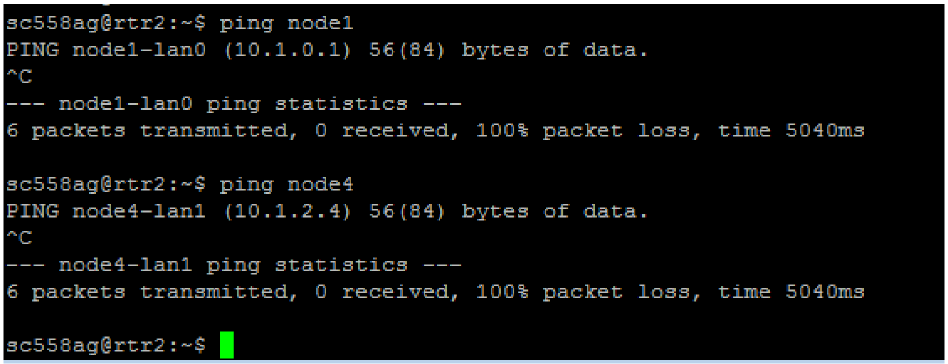


```
sc558ag@rtr1:/mnt/quagga-0.99.20/zebra$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.2.0       0.0.0.0         255.255.255.0   U     0      0        0 eth5
10.10.1.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
10.1.2.0        10.10.1.2       255.255.255.0   UG    0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@rtr1:/mnt/quagga-0.99.20/zebra$
```



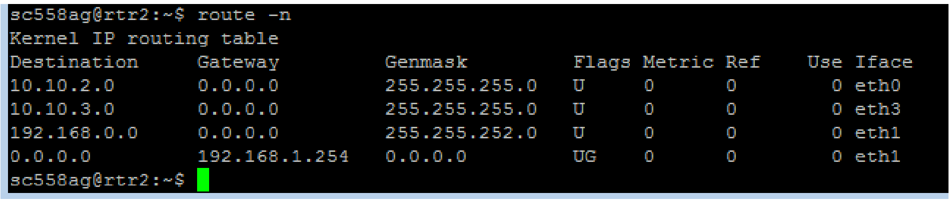
```
sc558ag@rtr3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.3.0       0.0.0.0         255.255.255.0   U     0      0        0 eth5
10.10.1.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.1.0.0        10.10.1.1       255.255.255.0   UG    0      0        0 eth0
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@rtr3:~$
```

Router2 cannot ping any nodes on either LANs. Because its routing table does not have entry for a path to forward any packets to the LANs.



```
sc558ag@rtr2:~$ ping node1
PING node1-lan0 (10.1.0.1) 56(84) bytes of data.
^C
--- node1-lan0 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5040ms

sc558ag@rtr2:~$ ping node4
PING node4-lan1 (10.1.2.4) 56(84) bytes of data.
^C
--- node4-lan1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5040ms

sc558ag@rtr2:~$
```

Routing table of router 2 :



```
sc558ag@rtr2:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.2.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.10.3.0       0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@rtr2:~$
```

Adding a secondary route through rtr2 on rtr1 and rtr3 .

Configuration commands to change routes:

At router1#*enable*

Zebrad(config)#configure terminal

Zebrad(config)#ip route 10.1.2.0/24 10.10.2.1

copy running-config startup-config

Configuration saved to /mnt/conf/zebra.conf

At router3#*enable*

Zebrad(config)#configure terminal

Zebrad(config)#ip route 10.1.0.0/24 10.10.3.2

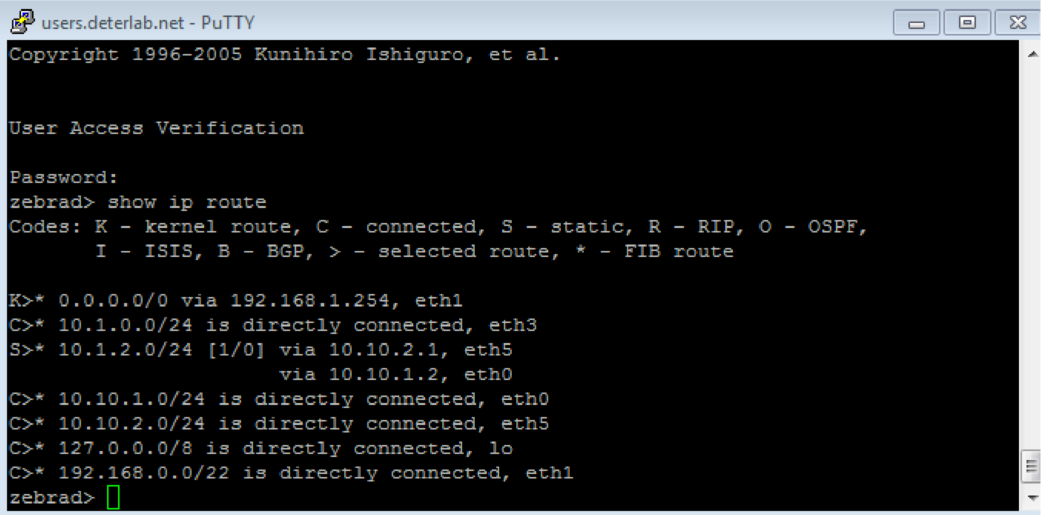
copy running-config startup-config

Configuration saved to /mnt/conf/zebra.conf

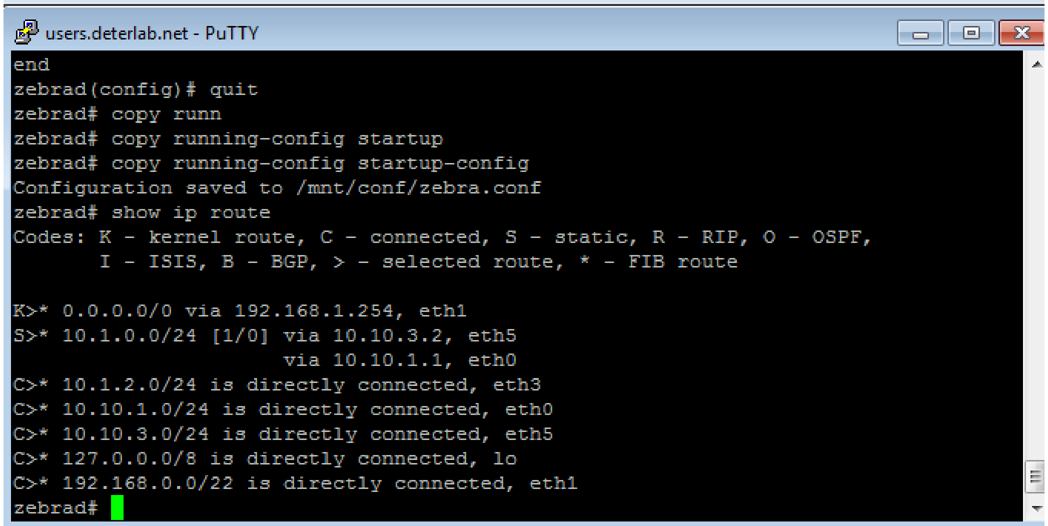
On doing **show ip route** on rtr1 pr rtr3 we see two lines in the routing table for one entry as follows.

Router1:

```
S>* 10.1.2.0/24 [1/0] via 10.10.2.1, eth5
                      via 10.10.1.2, eth0
```



```
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
zebrad> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.168.1.254, eth1
C>* 10.1.0.0/24 is directly connected, eth3
S>* 10.1.2.0/24 [1/0] via 10.10.2.1, eth5
                      via 10.10.1.2, eth0
C>* 10.10.1.0/24 is directly connected, eth0
C>* 10.10.2.0/24 is directly connected, eth5
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.0.0/22 is directly connected, eth1
zebrad> 
```

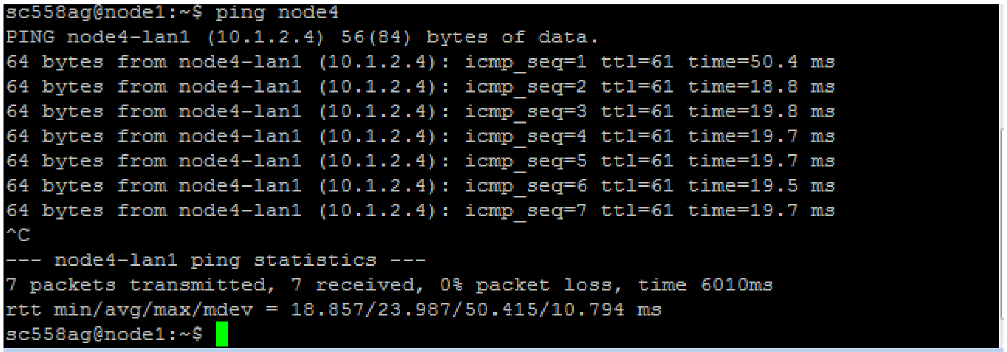


```
end
zebrad(config)# quit
zebrad# copy runn
zebrad# copy running-config startup
zebrad# copy running-config startup-config
Configuration saved to /mnt/conf/zebra.conf
zebrad# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.168.1.254, eth1
S>* 10.1.0.0/24 [1/0] via 10.10.3.2, eth5
                      via 10.10.1.1, eth0
C>* 10.1.2.0/24 is directly connected, eth3
C>* 10.10.1.0/24 is directly connected, eth0
C>* 10.10.3.0/24 is directly connected, eth5
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.0.0/22 is directly connected, eth1
zebrad# 
```

Router3:

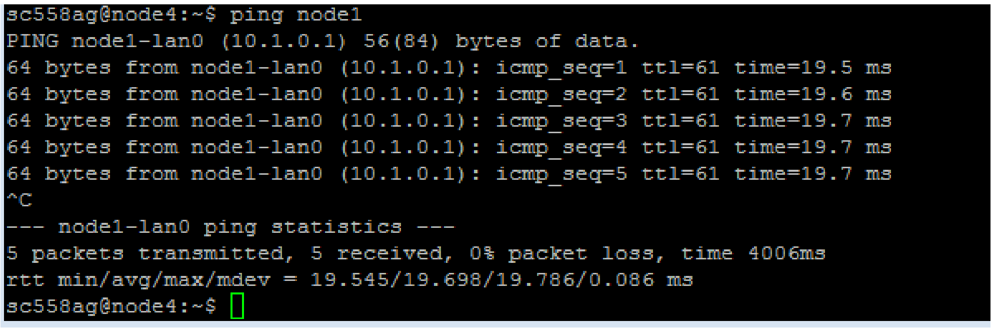
S>* 10.1.0.0/24 [1/0] via 10.10.3.2, eth5
Via 10.10.1.1, eth0

Pinging from node1 on LAN1 to node4 on LAN2:



```
sc558ag@node1:~$ ping node4
PING node4-lan1 (10.1.2.4) 56(84) bytes of data.
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=1 ttl=61 time=50.4 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=2 ttl=61 time=18.8 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=3 ttl=61 time=19.8 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=4 ttl=61 time=19.7 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=5 ttl=61 time=19.7 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=6 ttl=61 time=19.5 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=7 ttl=61 time=19.7 ms
^C
--- node4-lan1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 18.857/23.987/50.415/10.794 ms
sc558ag@node1:~$ 
```

Pinging from node4 on LAN2 to node1 on LAN1:



```
sc558ag@node4:~$ ping node1
PING node1-lan0 (10.1.0.1) 56(84) bytes of data.
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=1 ttl=61 time=19.5 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=2 ttl=61 time=19.6 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=3 ttl=61 time=19.7 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=4 ttl=61 time=19.7 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=5 ttl=61 time=19.7 ms
^C
--- node1-lan0 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 19.545/19.698/19.786/0.086 ms
sc558ag@node4:~$ 
```

Even on adding the secondary path to the router2 routing table, the tracepath shows that the packets are travelling through the link0 i.e. link between router1 and router3.

Changing the loss ratio to 1 on link0, we **cannot** ping between the nodes on both the LANs because it is static routing and the router1 will not know that the path is lossy and will keep sending packets via link0 (i.e. between rtr1 and rt3).

Changing the distance metric on link0 between router 1 and router 3 to 5 and the distance metric on links between router1 and router 2 and on link between router 2 and router 3 each to 1, makes the distance of link0 longer than the other route. So the packets travel through the other route and it was confirmed by using tracepath. Here by distance we mean the '**administrative distance'** as decided by the router.The smaller the administrative distance, the higher the preference.

Here next I have simply tried changing the path of packets through only via router 2. The link0 is never used and that is done by updating the routing tables of routers1, 2 and 3 accordingly as shown below.

Configuration commands to change routes:

```
At router1#enable
        Zebrad(config)#configure terminal
        Zebrad(config)#ip route 10.1.2.0/24 10.10.2.1
        Zebrad(config)#show running-config
        Zebrad(config)#no ip route 10.1.2.0/24 10.10.1.2
        Zebrad(config)#show running-config
        Zebrad(config)#quit
copy running-config startup-config
Configuration saved to /mnt/conf/zebra.conf
```

```
At router2#enable
        Zebrad(config)#configure terminal
        Zebrad(config)#ip route 10.1.2.0/24 10.10.3.1
```

Zebrad(config)#ip route 10.1.0.0/24 10.10.2.2

Zebrad(config)#show running-config

Zebrad(config)#no ip route 10.1.2.0/24 10.10.1.2

Zebrad(config)#show running-config

Zebrad(config)#quit

copy running-config startup-config

Configuration saved to /mnt/conf/zebra.conf

At router3#*enable*

Zebrad(config)#configure terminal

Zebrad(config)#ip route 10.1.0.0/24 10.10.3.2

Zebrad(config)#show running-config

Zebrad(config)#no ip route 10.1.0.0/24 10.10.1.1

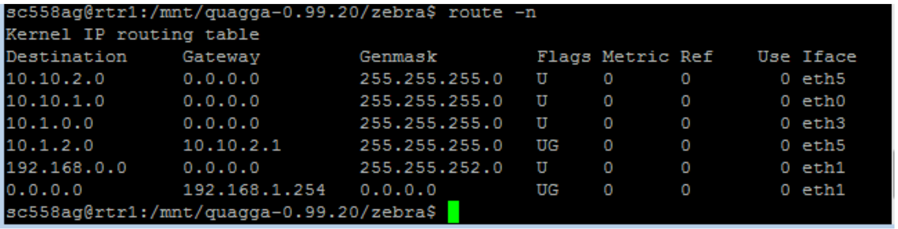
Zebrad(config)#show running-config

Zebrad(config)#quit

copy running-config startup-config

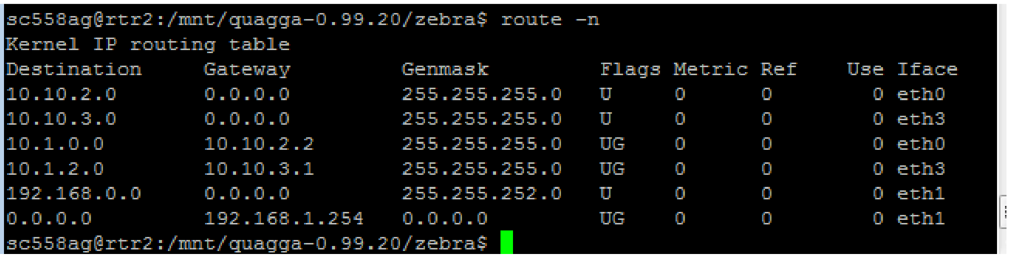
Configuration saved to /mnt/conf/zebra.conf

After configuration on router 1:



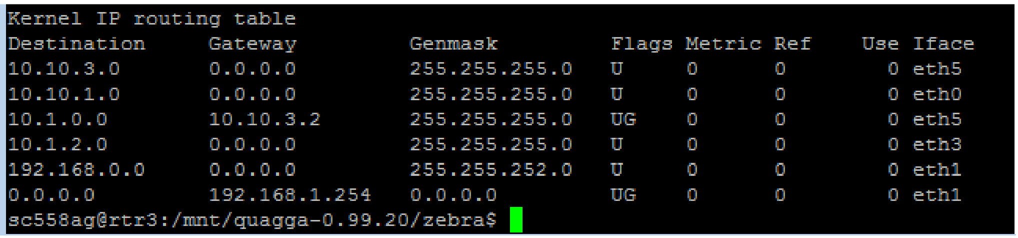
```
sc558ag@rtr1:/mnt/quagga-0.99.20/zebra$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.2.0       0.0.0.0         255.255.255.0   U     0      0        0 eth5
10.10.1.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.1.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
10.1.2.0        10.10.2.1       255.255.255.0   UG    0      0        0 eth5
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@rtr1:/mnt/quagga-0.99.20/zebra$ 
```

After configuration on router 2:



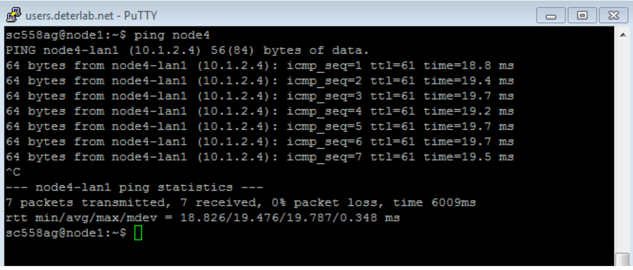
```
sc558ag@rtr2:/mnt/quagga-0.99.20/zebra$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.2.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.10.3.0       0.0.0.0         255.255.255.0   U     0      0        0 eth3
10.1.0.0        10.10.2.2       255.255.255.0   UG    0      0        0 eth0
10.1.2.0        10.10.3.1       255.255.255.0   UG    0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@rtr2:/mnt/quagga-0.99.20/zebra$ 
```

After configuration on router 3:

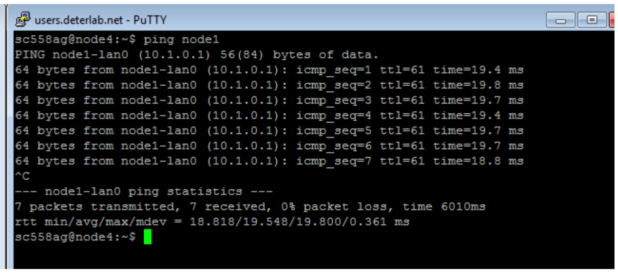


```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.3.0       0.0.0.0         255.255.255.0   U     0      0        0 eth5
10.10.1.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.1.0.0        10.10.3.2       255.255.255.0   UG    0      0        0 eth5
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.0.0     0.0.0.0         255.255.252.0   U     0      0        0 eth1
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth1
sc558ag@rtr3:/mnt/quagga-0.99.20/zebra$
```

Ping from node1 to node4 :



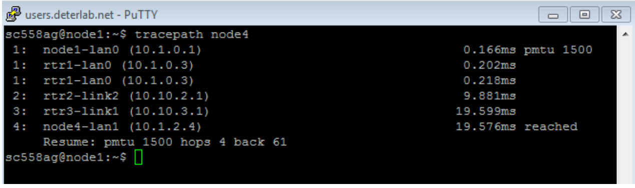
```
sc558ag@node1:~$ ping node4
PING node4-lan1 (10.1.2.4) 56(84) bytes of data.
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=1 ttl=61 time=18.8 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=2 ttl=61 time=19.4 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=3 ttl=61 time=19.7 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=4 ttl=61 time=19.2 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=5 ttl=61 time=19.7 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=6 ttl=61 time=19.7 ms
64 bytes from node4-lan1 (10.1.2.4): icmp_seq=7 ttl=61 time=19.5 ms
^C
--- node4-lan1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 18.826/19.476/19.787/0.348 ms
sc558ag@node1:~$
```



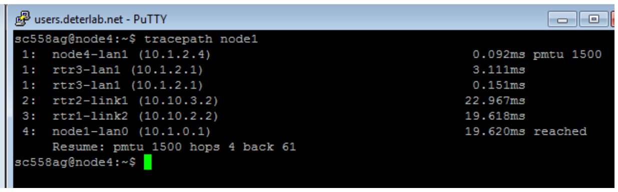
```
sc558ag@node4:~$ ping node1
PING node1-lan0 (10.1.0.1) 56(84) bytes of data.
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=1 ttl=61 time=19.4 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=2 ttl=61 time=19.8 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=3 ttl=61 time=19.7 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=4 ttl=61 time=19.4 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=5 ttl=61 time=19.7 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=6 ttl=61 time=19.7 ms
64 bytes from node1-lan0 (10.1.0.1): icmp_seq=7 ttl=61 time=18.8 ms
^C
--- node1-lan0 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 18.818/19.548/19.800/0.361 ms
sc558ag@node4:~$
```

The RTT can be seen to be around 20ms as expected. Since the delay on the links on LANs is 0ms and the delay on the links between router1 and router2 is 5 ms and the delay on the link between router 2 and router 3 is also 5ms, the delay turns out to be:
0 + 0 + 5+ 5 + 0 + 0 + 5 + 5 + 0 + 0 =20ms.

Tracepath from node1 to node4 and from node4 to node1:



```
sc558ag@node1:~$ tracepath node4
 1:  node1-lan0 (10.1.0.1)                                 0.166ms pmtu 1500
 1:  rtr1-lan0 (10.1.0.3)                                  0.202ms
 1:  rtr1-lan0 (10.1.0.3)                                  0.218ms
 2:  rtr2-link2 (10.10.2.1)                                9.881ms
 3:  rtr3-link1 (10.10.3.1)                               19.599ms
 4:  node4-lan1 (10.1.2.4)                                19.576ms reached
     Resume: pmtu 1500 hops 4 back 61
sc558ag@node1:~$
```



```
sc558ag@node4:~$ tracepath node1
 1:  node4-lan1 (10.1.2.4)                                 0.092ms pmtu 1500
 1:  rtr3-lan1 (10.1.2.1)                                  3.111ms
 1:  rtr3-lan1 (10.1.2.1)                                  0.151ms
 2:  rtr2-link1 (10.10.3.2)                               22.967ms
 3:  rtr1-link2 (10.10.2.2)                               19.618ms
 4:  node1-lan0 (10.1.0.1)                                19.620ms reached
     Resume: pmtu 1500 hops 4 back 61
sc558ag@node4:~$
```

Part 1.1

Setting the link loss of link0 to 1 makes the packet loss 100% on the link. So pinging does not work. The router process in static routing is ignorant about the loss parameter on that link and hence it keeps sending the packets on that link. This is a disadvantage that even on providing a secondary path for a lossy primary link, the router would not route the packets to the secondary path because of its ignorance.

If I were to manually have to change the routing tables for the large number of routers, it would be really difficult for me to keep changing each entry of the routing table. And if we are talking at an ISP level, it would surely be much more difficult to maintain each routing table with the increase in the number of entries. So manually updating the routers does not make sense. Rather using various existing protocols would be much helpful. The importance of routing protocols can only be realized after knowing how complex and difficult it is to maintain each routing table at an ISP level where the number of routers is large.

OSPF is an interior gateway protocol that gathers link state information from available routers and constructs a topology map of the network. The topology determines the routing table presented to the Internet Layer which makes routing decisions based on the destination IP address found in IP packets. OSPF detects changes in the topology, such as link failures, very quickly and converges on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm. The **link-state** information is maintained on each router as a link-state database (LSDB) which is a tree-image of the entire network topology. Identical copies of the LSDB are periodically updated through flooding on all OSPF routers. The OSPF routing policies to construct a route table are governed by link cost factors (*external metrics*) associated with each routing interface. Cost factors may be the distance of a router (round-trip time), network throughput of a link, or link availability and reliability, expressed as simple unitless numbers. This provides a dynamic process of traffic load balancing between routes of equal cost. In this way OSPF can solve problems in static routing which ignore the losses on the link and other metrics.

Conclusion:
The aim of this tutorial was getting acquainted to Quagga router software and the zebra piece of software which in this tutorial was used for static routing. A hands-on experience of using Cisco like commands over a CISCO like router for network administration and network maintenance was attained. The importance of routing protocols to be used for dynamic routing over static routing could be realized.