Programming Assignment 3

Routing Protocols

Names :

Mohamed Ahmed Abd-Twab (57)
Mohamed Samir shaaban (60)
Mohamed Mahmoud Adel (79)

Problem Statement:

- Create a script that simulates the following topology, with those nodes having point to points links, and internet stack installed.

- When computing the routes, the GlobalRouting implementation in NS3 keeps all routes to the node in the database, You are required to update this part to lter the routes keeping only the route with the min number of hops (least cost).

How to create topology (part one):

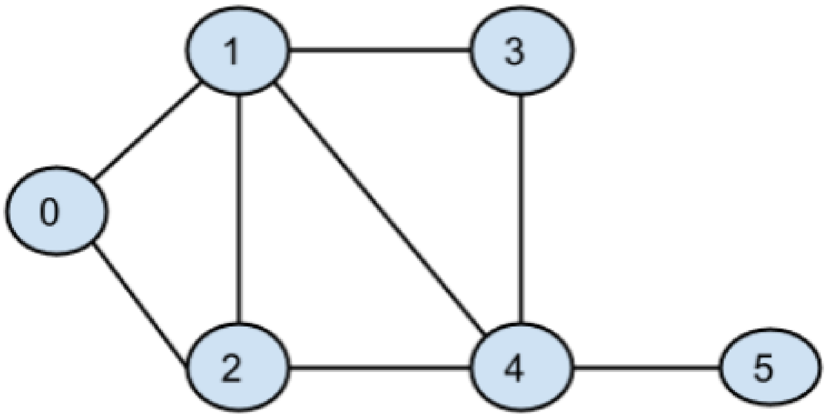


Figure 1: Network topology.

- We use ordinary class that explained on piazza also in tutorial :
  - * Node (simulate a host machine), and NodeContainer
  - * NetDevice (simulate a NIC), and NetDeviceContainer
  - * PointToPointHelper (installs devices on nodes, and sets up the channel connection between them)
  - * InternetStackHelper (installs the stack protocols on nodes such as TCP, IP,...)
  - * IPv4AddressHelper (sets IP addresses to nodes)
  - * IPv4GlobalRoutingHelper (builds routing tables for a topology)
  - * UDPEchoServerHelper, and UDPEchoClientHelper (install server/client application to a node)
- we create 6 nodes (number of nodes).
- we use 8 NodeContainer (first node on link, second node on link).
- we use PointToPointHelper to install 8 NetDeviceContainer.
- then assign ip address for each link between node

node(start , end) on link (p2p)	ip
0 , 1	10.1.1.0
0 , 2	10.1.2.0
1 , 2	10.1.3.0
1 , 4	10.1.4.0
1 , 3	10.1.5.0
2 , 4	10.1.6.0
4 , 5	10.1.7.0
3 , 4	10.1.8.0

- we will use this ip address to illustrate later.
- we use PopulateRoutingTables() to builds routing tables for a topology.

- we create echo server to create server application and make node 5 as a server .
- we create echo client to create client application and make node 0 , 1 , 2 , 3 and 4 as a client(one server and 4 client).

- client at 0 run in second 2 , 1 in second 3 , 2 in second 4 , 3 in second 5 and 4 in second 6.

Change in (part 2):

* two changes in global-route-manager-impl.cc file:

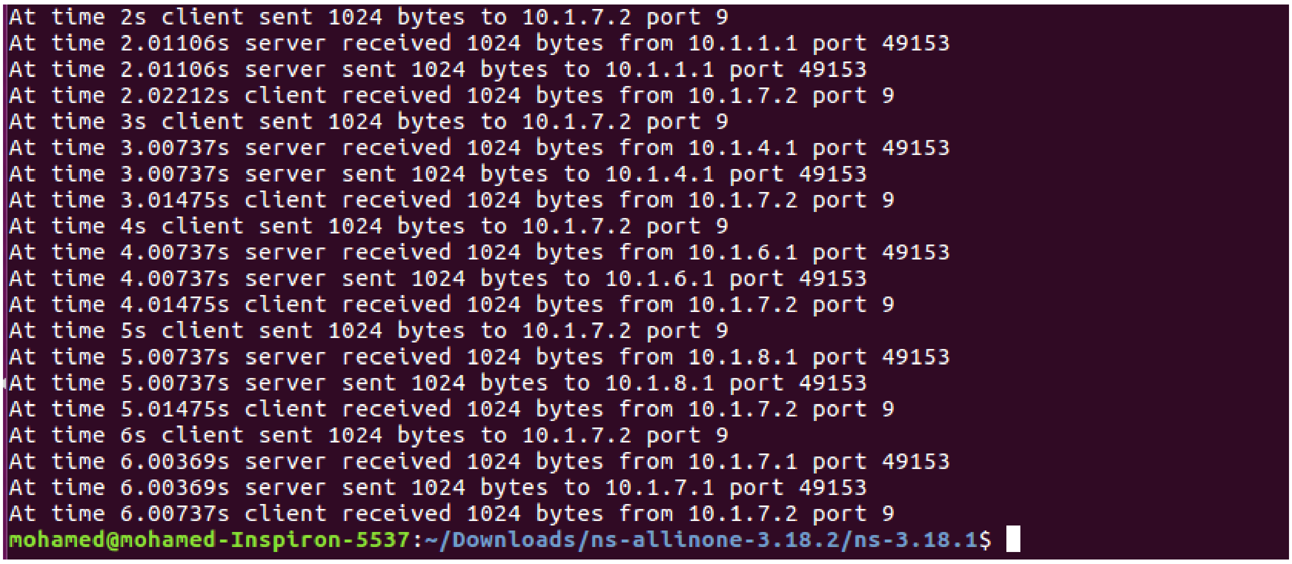
1) in method SPFCalculate():

edit the condition which check the if the node has only one interface through which another router can be reached and remove this condition, in order to get routing table for all nodes including nodes which have only one interface.

2) in method SPFNext():

it's required to only add the first min route to the m_lsdb database, not all of them, so we modified the condition which checks if equal cost to if there is another route equal cost to the min one don't do any thing reather than add it to database.

Explain printed output:



```
At time 2s client sent 1024 bytes to 10.1.7.2 port 9
At time 2.01106s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.01106s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.02212s client received 1024 bytes from 10.1.7.2 port 9
At time 3s client sent 1024 bytes to 10.1.7.2 port 9
At time 3.00737s server received 1024 bytes from 10.1.4.1 port 49153
At time 3.00737s server sent 1024 bytes to 10.1.4.1 port 49153
At time 3.01475s client received 1024 bytes from 10.1.7.2 port 9
At time 4s client sent 1024 bytes to 10.1.7.2 port 9
At time 4.00737s server received 1024 bytes from 10.1.6.1 port 49153
At time 4.00737s server sent 1024 bytes to 10.1.6.1 port 49153
At time 4.01475s client received 1024 bytes from 10.1.7.2 port 9
At time 5s client sent 1024 bytes to 10.1.7.2 port 9
At time 5.00737s server received 1024 bytes from 10.1.8.1 port 49153
At time 5.00737s server sent 1024 bytes to 10.1.8.1 port 49153
At time 5.01475s client received 1024 bytes from 10.1.7.2 port 9
At time 6s client sent 1024 bytes to 10.1.7.2 port 9
At time 6.00369s server received 1024 bytes from 10.1.7.1 port 49153
At time 6.00369s server sent 1024 bytes to 10.1.7.1 port 49153
At time 6.00737s client received 1024 bytes from 10.1.7.2 port 9
mohamed@mohamed-Inspiron-5537:~/Downloads/ns-allinone-3.18.2/ns-3.18.1$
```

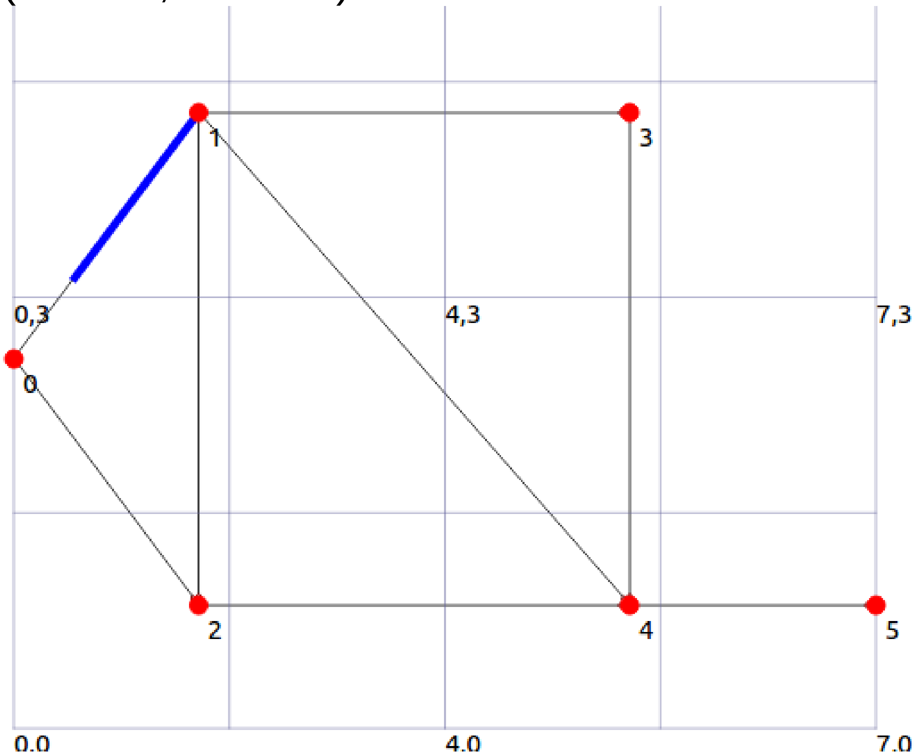
- as shown in second 2 client at node 0 (10.1.1.1) start run as mentioned above and send packet to server node 5 (10.1.7.2) then server received packet from 10.1.1.1(node 0) then send it to client again then client.
- as shown in second 3 client at node 1 (10.1.4.1) start send packet to server at node 5 (10.1.7.2) then server received it from client (10.1.4.1) then send it again to client and client received it.

Why node 1 take ip (10.1.4.1) not (10.1.5.1) not (10.1.3.1) not (10.1.2.2) because the path from it to client start from p2p link (channel) between it and node 4 (0 , 4) > (10.4.1.0) see table.

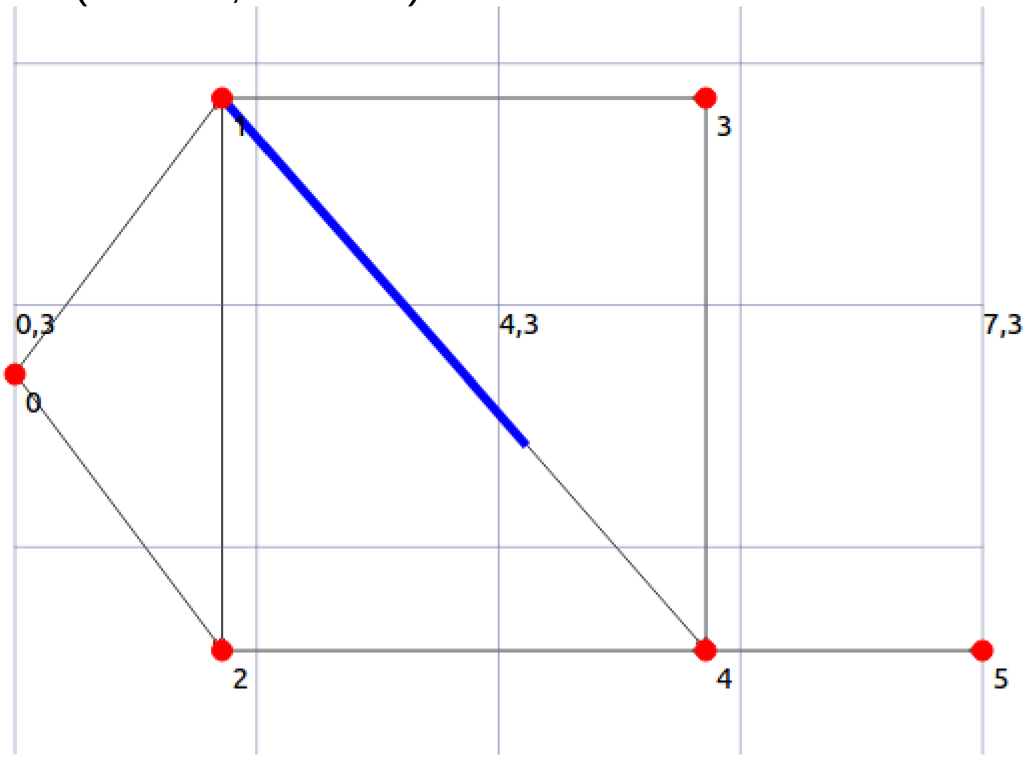
- all remaining client in node 2 , 3 , 4 make same thing as mentioned above.

Screen shoot in simulate to make it clear:

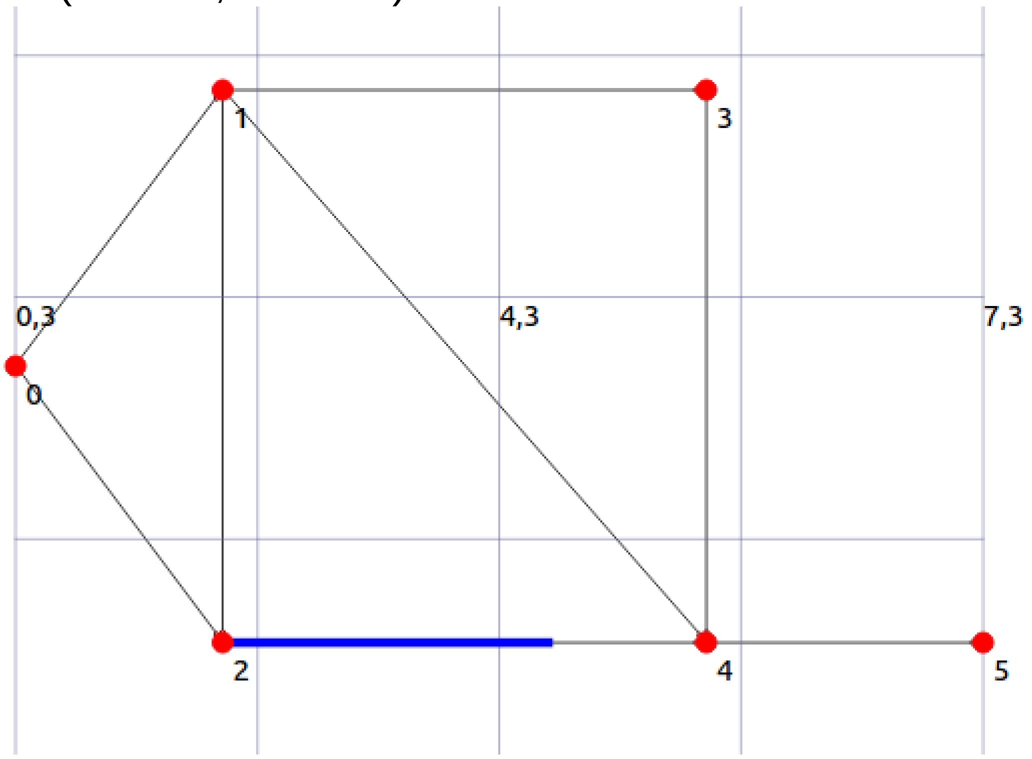
* client 0 (node 0 , 10.1.1.1):



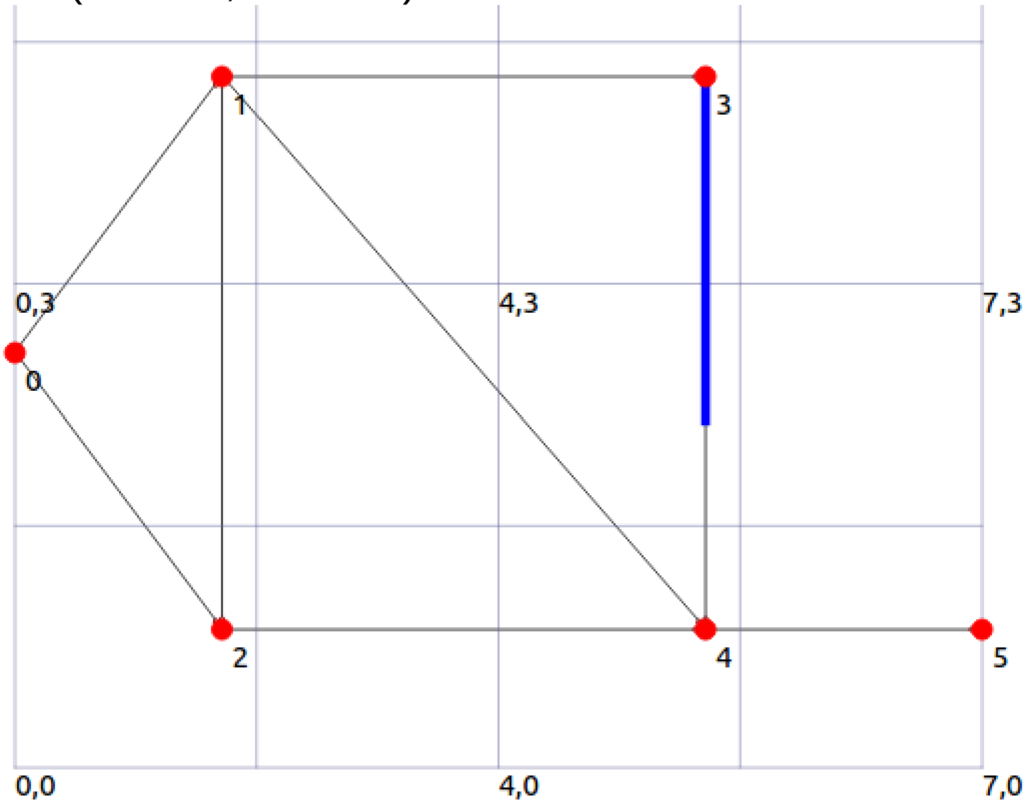
* client 1 (node 1 , 10.1.4.1):



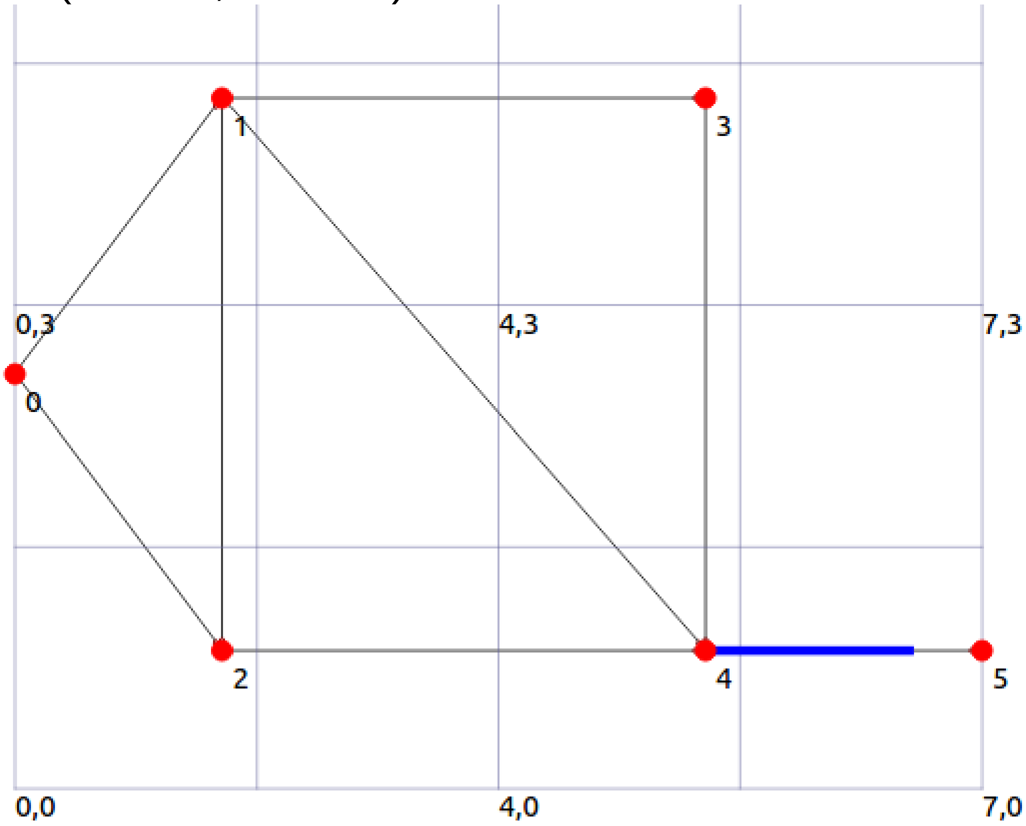
* client 2 (node 2 , 10.1.6.1):



* client 3 (node 3 , 10.1.8.1):



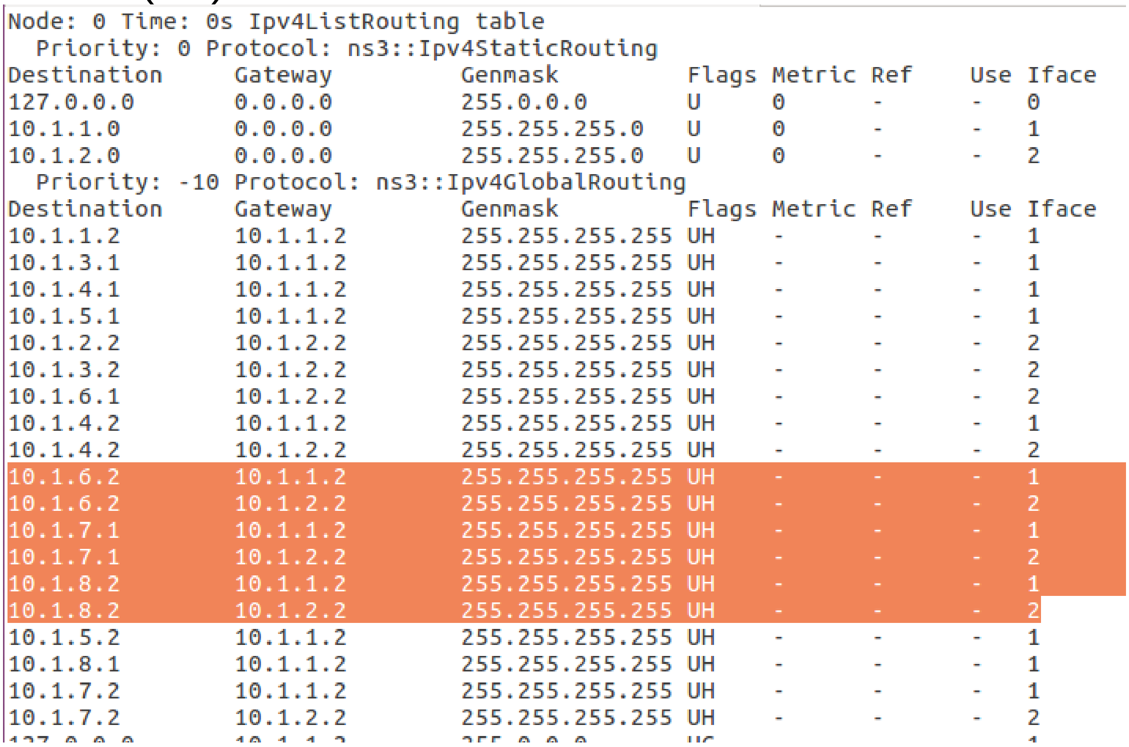
* client 4 (node 4 , 10.1.7.1):



I hope that explanation make it clear and enough.

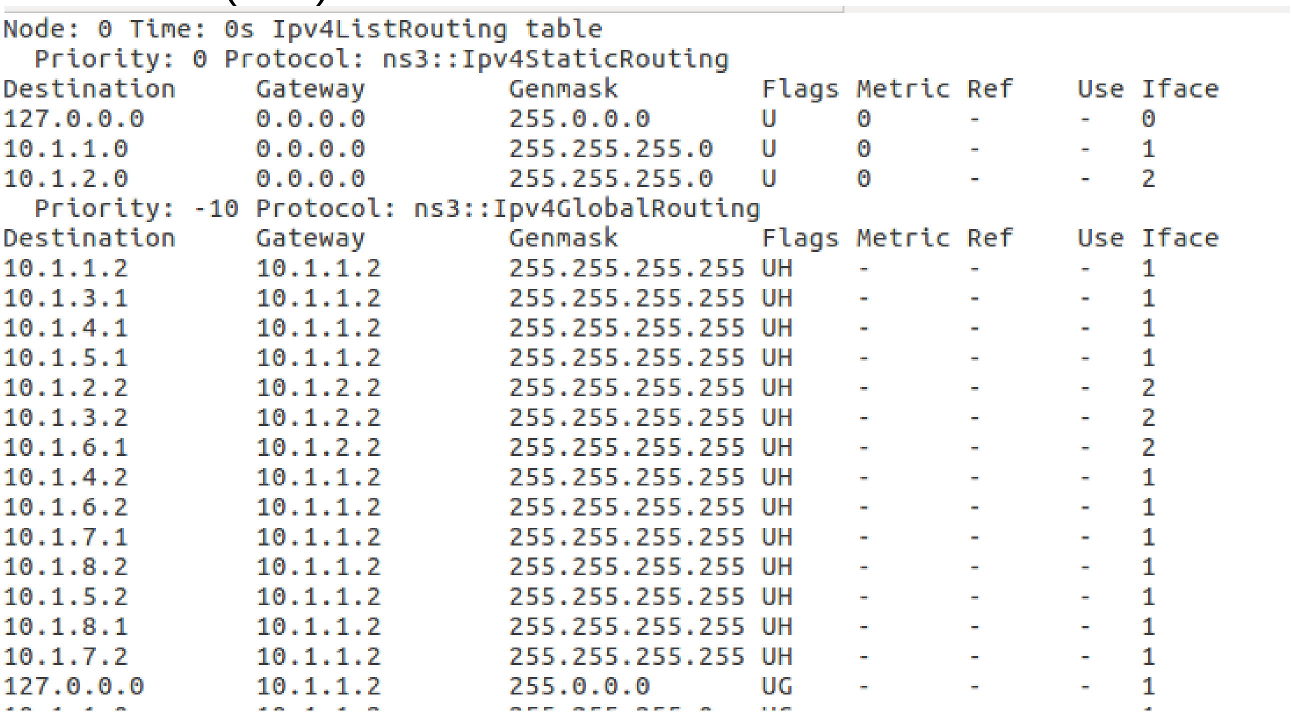
<u>Comparing between old printed route table and new:</u>

For node 0 (old):



```
Node: 0 Time: 0s Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Ipv4StaticRouting
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     0      -      -   0
10.1.1.0        0.0.0.0         255.255.255.0   U     0      -      -   1
10.1.2.0        0.0.0.0         255.255.255.0   U     0      -      -   2
  Priority: -10 Protocol: ns3::Ipv4GlobalRouting
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.3.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.4.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.5.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.2.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.3.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.6.1        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.4.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.4.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.6.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.6.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.7.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.7.1        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.8.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.8.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.5.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.8.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.7.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.7.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
127.0.0.0       10.1.1.2        255.0.0.0       UG    -      -      -   1
```

For node 0 (new):



```
Node: 0 Time: 0s Ipv4ListRouting table
  Priority: 0 Protocol: ns3::Ipv4StaticRouting
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     0      -      -   0
10.1.1.0        0.0.0.0         255.255.255.0   U     0      -      -   1
10.1.2.0        0.0.0.0         255.255.255.0   U     0      -      -   2
  Priority: -10 Protocol: ns3::Ipv4GlobalRouting
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.3.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.4.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.5.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.2.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.3.2        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.6.1        10.1.2.2        255.255.255.255 UH    -      -      -   2
10.1.4.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.6.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.7.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.8.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.5.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.8.1        10.1.1.2        255.255.255.255 UH    -      -      -   1
10.1.7.2        10.1.1.2        255.255.255.255 UH    -      -      -   1
127.0.0.0       10.1.1.2        255.0.0.0       UG    -      -      -   1
```

we can see duplicates are removed and less path.

Explain new route table :
- From node 0
- Destination (want to reach) node ip
- GateWay (the way gate to reach destination)
- Use Iface (node number) : we can see node 1 take more than ip depend on link (10.1.1.2 , 10.1.3.1 , 10.1.4.1 and 10.1.5.1) and so on.

- we upload old file and new file if you want to check and review them .

Thanks :D