**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:

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.

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

**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**  **PING from node3 to node1**

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 Tracepath from node 3 to node 1

**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**:

**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.

**Routing table of router 2**  :

**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*

**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:**

**Pinging from node4 on LAN2 to node1 on LAN1:**

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:**

**After configuration on router 2:**

**After configuration on router 3:**

**Ping from node1 to 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:**

**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.