International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Test Usability of Routing Protocol for QoS Parameters in Cloud

Murtuza Petladwala

Department of Information Technology, SRM University, SRM Nagar, Kattankulathur, Tamil Nadu, India

Abstract: ***Cloud computing is emerging computer paradigm where computing storage and networking utilities are offered mainly to the business community. In this paper we use cloud computing in testing and simulating routing protocols. The benefit of simulating in the cloud is twofold: it provides hardware independence for the underlying test environment, in addition to better methods for monitoring the performance of the protocol. The transmission time, delay and packet loss parameters are measured in the network through Quagga code for testing the efficiency in the particular routing protocol. After obtained result we show that, if the routing software is optimized, the pc-based routers perform better than commercial router.***

Keywords: Cloud computing, routing protocol, software routers, Quagga, QoS.

1. Introduction

Cloud Computing is emerging method of computing where one can rent various services [1]. In other words it allows use of various services without the need of installing them on a local machine; the minimum requirement is only a computer connected on Internet and a web browser. Cloud computing services provided by Cloud Providers fall in one of the three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Cloud installations are classified in four categories: Private (provisioned for use by a single organization), Community (provisioned for exclusive use by a specific community of consumers), Public (provisioned for open use by general public) and Hybrid (composition of two or more distinct cloud infrastructures) clouds.

Routers are central devices in the contemporary networking world. Great demand is placed on them for providing shortest path inter-connections among various networks. In the majority of today's networks it is impossible for routers to find all the shortest paths without the help of the routing protocols. Though the method of finding the shortest path has been settled more than 60 years ago, and the choice of the appropriate metric seems well elaborated, the convergence is still troublesome for the network administrators. One way to deal with the issue of slow convergence is to segment the network into smaller domains; but this approach shifts the administrative burden toward the human. Quagga can help networking professionals build such custom solutions, in combination with other Open Source software packages.

Thus it is important for us to keep designing new and upgrading the existing routing protocols so that they can scale well with modern networking demands [2]. Cloud computing paradigm can offer several benefits in pursuing these efforts. The most obvious benefit is to provide scalable test environment that does not depend on the underlying hardware. In addition, synchronization and monitoring of the test nodes and network packets can be handled with plethora of existing tools.

In this paper we present a framework on how to build test network of routing nodes in a virtual environment. We use a VMware Workstation desktop version as virtual environment which run on top of the windows host machine, on which we have Quagga routing nodes. We will elaborate our framework using a simple analysis on packet delay and the distinct size of data transferred in the network with different routing protocols.

2. Simulating Routing Protocol

Our network consists of three Debian virtual machines interconnected with three virtual networks as shown in figure 1 and one public access point for each machine. Virtual machines act as virtual routers (VRs) with the installation of the Quagga routing software [3]. The VMware Workstation desktop version has the option to configure the virtual network in the system.

Quagga is open source routing software that implements most of today's routing protocols (RIP, OSPF and BGP). The software may offer no means to customize the routing functionality, there may be little means for data gathering or processing, and so on. The hardware may not have any storage capabilities, or sufficient spare capacity for additional data-processing.

Networking professionals and researchers sometimes face problems that go beyond the capabilities of such routing products. They may wish to combine routing functionality with existing open source software packages, but be constrained on how many separate devices can be deployed. Researchers and operators might wish to log routing protocol announcements, process them, and make them available, and hence require storage and scripting capacities that are not available in existing routing products. Researchers may wish to extend routing protocols, but not want to write complete implementations from scratch.

The Quagga Routing Suite is a package of Unix/Linux software implementing a number of common network routing protocols, including the "Routing Information Protocol" (RIP) [4], Open Shortest Path First (OSPF) [5],the Border Gateway Protocol (BGP) [6] and IS-IS [7]. The

package also includes a routing information management process, to act as intermediary between the various routing protocols and the active routes installed with the kernel. A library provides support for configuration and an interactive command line interface.

Quagga is available under the terms of the GPLv2 license. Quagga thus provides a useful addition to the tool box of networking professionals. Its existing routing protocols can be extended to enable experimentation, logging or custom processing. The libraries and kernel daemon provide a framework for easier development of new routing protocol daemon. It can be combined with arbitrary other available software packages, to allow a wide range of functionality to be integrated into a single device, or allow for imaginative solutions to networking problems.

Routing protocols are configured via CLI known as VTY or telnet. Particular useful feature of Quagga command is their similarity with the standard commands found on Cisco's routers. In addition, the ability to simulate loopback interfaces can also be very useful in our simulations. After initial configurations we verify network connectivity with successful pings among adjacent routers. Next we activate the RIP daemons on all VRs.

/opt/quagga/sbin/zebra –d
/opt/quagga/sbin/ripd –d

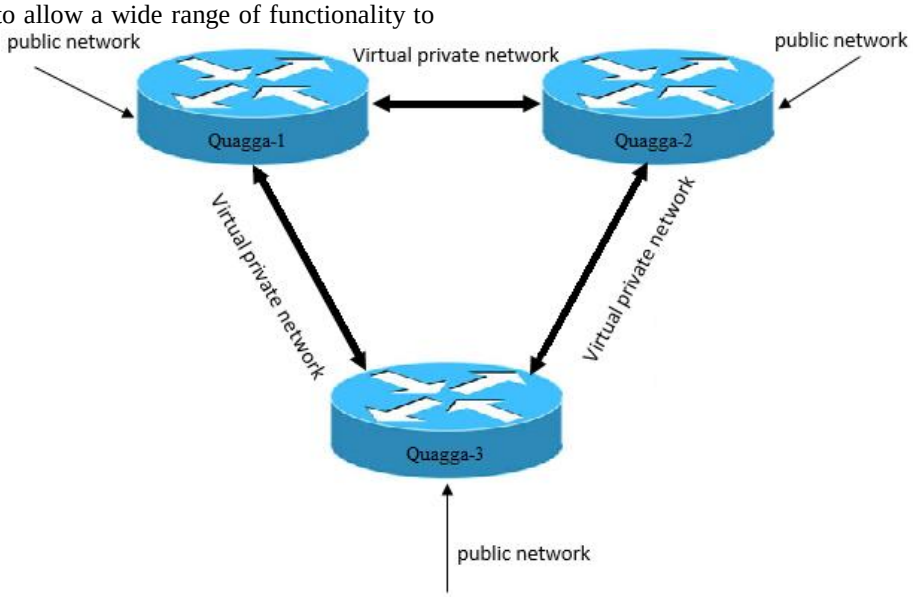

Figure 1: Topology of test network

And then connect to all CLI in order to configure the Routing Information Protocol. We verify network connectivity of distant networks from the routing tables or with extended pings.

Routing Information Protocol (RIP) is one of the oldest and still alive routing protocols. Its development began in the late '70s from the Xerox's XNS protocol. The first document that describes RIP was published in 1988, however recent RFC extensions that were proposed to support IPv6 and cryptographic authentications secured its future existence. RIP metric is an integer between 1 and 15, with 16 being reserved for infinity. The way the costs for traversing networks are associated is not specified in the standard, but due to the limit of 15, the cost is usually 1. This is the well-known *hop-count* metric used by RIP.

RIP packets are encapsulated in UDP segments before being sent over IP network. RIP configured routers send and receive RIP packets on port 520. RIP packet format is given on figure 2:

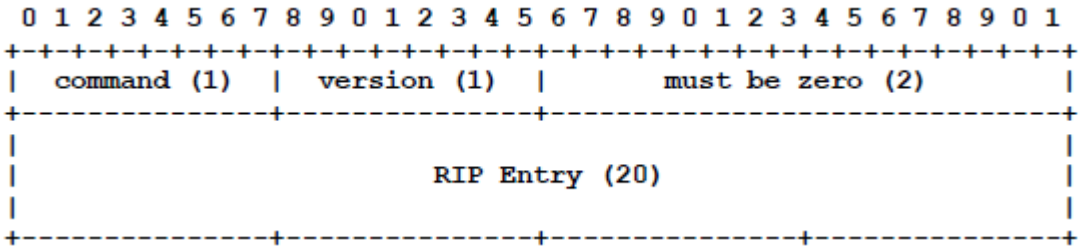

Figure 2: RIP packet

We can notice that RIP packets are aligned on the 32 bit boundaries. Version field (fig. 2) helps to distinguish between RIP version 1 and RIP version 2 packets. The command field defines two types of messages:

1) *Request* – from a neighbor router to send all or part of the routing table
2) *Response* – from the neighbor router with all or part of the routing table.

Each RIP packet (fig. 2) can carry information for up to 25 routes. Parameters requested or sent back for one route are carried with one RIP entry (fig.3).

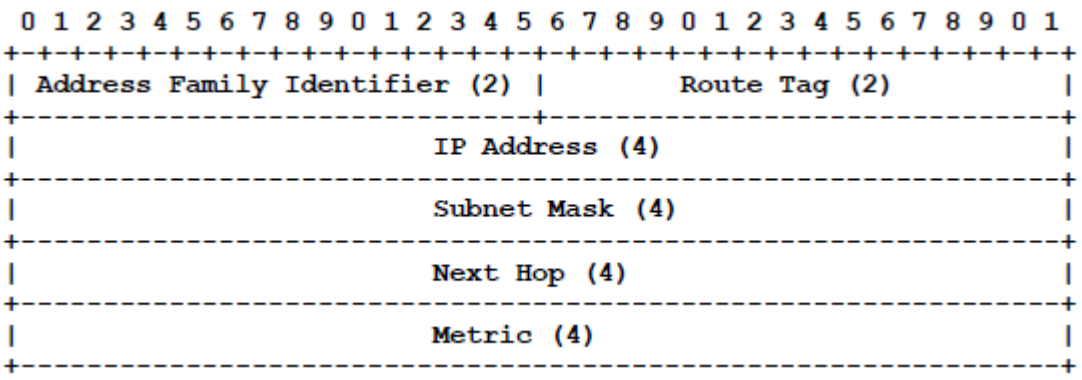

Figure 3: RIP Entry

In order to be able to monitor in real time packet flow through the network, we have to install Wireshark. Then we have to send data of different size and use the generated pcap file to analyze the packet delay, packet loss or any Quality of Service parameter that we are going to test in our framework. The RIP protocol performance is analyzed with the cisco router and software router i.e. quagga. The GNS3 software is used as the cisco router IOS images to create the topology and test the Quality of Service parameters. The various packet and the topology design are made to know the comparison between the quagga software router and the cisco router. Software routers are achieving a great interest in the last years because they represent an even more realistic alternative to commercial routers like we use in GNS3 or the other analyzing tool for network topologies. The routing protocols are tested and thus compared.

3. Analyzing Routing Protocol

Next, we will demonstrate the procedure for analyzing routing protocols. We want to use Quagga's RIP implementation so that routing packets can carry the payload i.e. data from a route. Additional fields in the routing packets can be used for various reasons, for example, to compare different metrics or to put time stamps in the packets. RIP uses hop-count to decide the best route, though better metrics exist. On the other hand hop-count cannot be removed from the RTE updates because mechanisms for routing loop detection (like split horizon) depend on it.

One of the most important aspect of simulating is real-time analysis of the protocol. For known protocols, this can be done with packet analyzers, but the problem is that packet analyzers cannot recognize new protocol, thus they will display only raw data in hexadecimal format. We have chosen Wireshark in our framework for real-time packet analysis due to its open source. All performed tests are based on fully meshed network topologies, with each router connected to each other through a different transit network.

A. Routing Information Protocol

Very similar how we get the terminal access from telnet, the same way is used to connect directly to this RIP daemon is by telnet the port 2602. Therefore, the VTY ask the user access verification saved in ripd.conf file. Then, typing the enable command and inserting the password is possible to access the core to configure the virtual router.

Routing Information Protocol (RIP) is widely deployed interior gateway protocol. RIP is a distance-vector protocol and is based on the Bellman-Ford algorithms. As a distance-vector protocol, RIP router send updates to its neighbors periodically, thus allowing the convergence to a known topology. In each update, the distance to any given network will be broadcasted to its neighboring router. The daemon ripd supports RIP version 2 as described in RFC2453 [8] and RIP version 1 as described in RFC1058.

The netmask features of ripd support both version 1 and version 2 of RIP. Version 1 of RIP originally contained no netmask information. In RIP version 1, network classes were originally used to determine the size of the netmask. Class A networks use 8 bits of mask, Class B networks use 16 bits of masks, while Class C networks use 24 bits of mask. Today, the most widely used method of a network mask is assigned to the packet on the basis of the interface that received the packet. Version 2 of RIP supports a variable length subnet mask (VLSM). By extending the subnet mask, the mask can be divided and reused. Each subnet can be used for different purposes such as large to middle size LANs and WAN links. Quagga ripd does not support the non-sequential netmask that are included in RIP Version 2. RIP protocol has several timers. User can configure those timer's values by timer's basic command.

B. Border Gateway Protocol

The connection to this daemon is by telnet the port 2605. Therefore, the VTY ask the user access verification saved in bgpd.conf file. After this, like a real Cisco router, typing the question mark it will shows a list of command that a standard user can invoke. Then, typing the enable command and inserting the password is possible to access the core to configure the virtual router.

BGP stands for a Border Gateway Protocol. The latest BGP version is 4. It is referred as BGP-4. BGP-4 is one of the Exterior Gateway Protocols and defact standard of Inter Domain routing protocol. BGP-4 is described in RFC1771, a Border Gateway Protocol 4 (BGP-4). Many extensions have been added to RFC1771. RFC2858, Multiprotocol Extensions for BGP-4 provides multiprotocol support to BGP-4. We must configure BGP router with router bgp command. To configure BGP router, we need AS number. AS number is an identification of autonomous system. BGP protocol uses the AS number for detecting whether the BGP connection is internal one or external one.

quagga-router (config)#router bgp asn
Enable a BGP protocol process with the specified asn. After this statement can input any BGP Commands. It cannot create different BGP process under different asn without specifying multiple-instance.

quagga-router (config-router)#bgp router-id A.B.C.D
This command specifies the router-ID. If bgpd connects to zebra it gets interface and address information. In that case default router ID value is selected as the largest IP Address of the interfaces. When router zebra is not enabled bgpd can't get interface information so router-id is set to 0.0.0.0

C. Open Shortest Path First

The only way to connect directly to this daemon is by telnet the port 2604. OSPF (Open Shortest Path First) version 2 is a routing protocol which is described in RFC2328. OSPF [9] is an IGP (Interior Gateway Protocol). Compared with RIP, OSPF can provide scalable network support and faster convergence times. OSPF is widely used in large networks such as ISP (Internet Service Provider) backbone and enterprise networks.

The OSPF standard for ABR behavior does not allow an ABR to consider routes through non-backbone areas when its links to the backbone are down, even when there are other ABRs in attached non-backbone areas which still can reach the backbone, this restriction exists primarily to ensure routing loops are avoided. With the "Cisco" or "IBM" ABR type, the default in this release of Quagga, this restriction is lifted, allowing an ABR to consider summaries learned from other ABRs through non-backbone areas, and hence route via non-backbone areas as a last resort when, and only when, backbone links are down. Note that areas with fully adjacent virtual-links are considered to be "transit capable" and can always be used to route backbone traffic, and hence are unaffected by this setting. Moreover though the definition of the ABR (Area Border Router) in the OSPF specification does not require a router with multiple attached areas to have a backbone connection, it is actually necessary to provide successful routing to the inter-area and external destinations. If this requirement is not met, all traffic destined for the areas not connected to such an ABR or out of the OSPF domain, is dropped.

A dedicated virtual machine is used to send the data from the given network and a java program is developed to initiate the network transactions. The program will create a TCP connection with the other quagga router i.e. software router running on top of the Ubuntu 12.04 machine which will send and receive the TCP traffic. From the generated traffic we can analyze the transmission time, packet delay and packet loss through Wireshark while testing different routing protocols.

4. Generating Traffic And Monitoring

This section will describe the traffic generation for testing and its use in this project. The next step after creating a test network is to test it, to see what happens when there is traffic incoming. For this use it has been necessary to generate traffic that goes through the virtual network. The tool to realize it are called packet generator. Packet generators create a discrete chunk of communication in a predefined format. A packet is a data block containing a header that includes destination address. All network communication that occur across a packet-switched system transmit packets. These packets are then reassembled by a recurring system from at the destination. The purpose of a packet generator is to permit users or network specialist to construct a packet (e.g.: WAN packets, VOIP packets) from one or more specific protocol stack areas for the purpose of testing security, communication effectiveness, or source-to-destination accuracy. A packet generator or packet builder is a type of software that generates random packets or allows the user to construct detailed custom packets. This software sends specific packets out a single or multiple network interfaces.

a. Iperf

Iperf [10] is a commonly used network testing tool that can create TCP and UDP data streams and measure the throughput of a network that is carrying them. Iperf allows the user to set various parameters that can be used for testing a network, or alternately for optimizing or tuning a network. So it can be defined as a tool to measure bandwidth and the quality of a network link. The network link is delimited by two hosts running Iperf. Iperf has a client and server functionality. The quality of network link can be tested as follows:

- Latency (response time or RTT): measured with the Ping command.
- Jitter (latency variation): measured with an Iperf UDP test.
- Datagram loss: measured with an Iperf UDP test.
- The bandwidth: measured through TCP tests.

In additional, the difference between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) is that TCP use processes to check that the packets are correctly sent to the receiver whereas with UDP the packets are sent without any checks but with the advantage of being quicker than TCP. Iperf uses the different capacities of TCP and UDP to provide statistics about network links. When used for testing UDP capacity, Iperf allows the user to specify the datagram size and provides results for the datagram throughput and the packet loss. As it is open source, the measurement methodology can be scrutinized by user.

b. Wireshark

Wireshark [11] is a free and open-source software protocol analyzer, or packet sniffer application, used for network troubleshooting, analysis, software and protocol development, and education. Wireshark has all of the standard features of a protocol analyzer. It runs on various Unix-like operating system including Linux, Mac OS, BSD and Microsoft Windows. Wireshark is very similar to tcpdump, but has a graphical front-end, and many more information sorting and filtering options. Wireshark allows the user to see all traffic being passed over the network by putting the network interface into promiscuous mode.

To perform all the tests, we have used one physical machine with the following requisites:

- Windows 7 Ultimate
- Service Pack 1

- Intel Core i5-2410M CPU @2.30GHz
- 4 GB Ram

The connection among routers is allowed configuring the 3 routing protocol (RIP, BGP and OSPF). First of all the configuration of Quagga to work as router and then it will be tested. The figure 10 shows the scenario that it will be created using quagga software. In this, there are three routers. To do that it is necessary three virtual machines, one for each router. Notice that to perform this in a real environment it is necessary three physical router, two physical switch plus four cables. In a virtual network environment only one physical PC is needed.

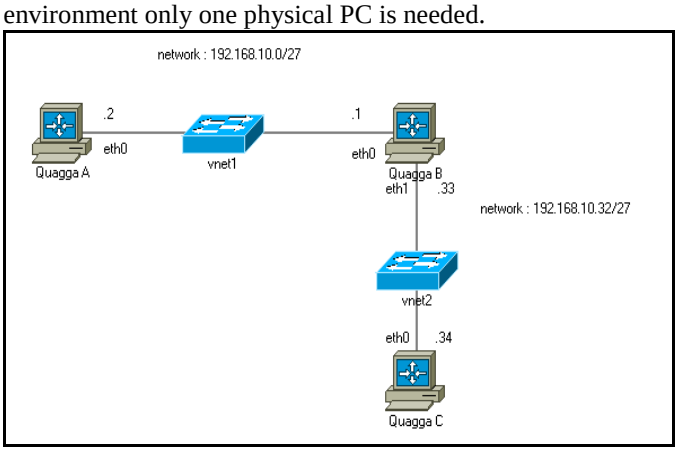

Figure 4: Scenario for testing in virtual environment

The virtual machines are created using VMware Workstation having the operating system Ubuntu Desktop version 12.04 LTS. Every virtual machines have one processor 256 MB of RAM and 8 GB of Hard Disk.

5. Conclusion

We have demonstrated a procedure for testing routing protocols. At early stages, testing in the cloud offers the benefit of hardware independence thus saving time with hardware related issues. However, we believe that cloud environment not only offers time saving, but also additional features not easily available in real environments. A routing protocol is characterized with its speed of convergence, percentage of redundant traffic through the network and CPU usage on the routers. The speed of convergence can be measured with the NTP protocol and the redundant traffic can be measured with Wireshark. The transmission delay, packet loss parameters are also the part of the service level agreements done in the cloud project or the virtual environment. The QoS helps in building Service level agreements SLA between the business companies. In particular, the most interesting feature of cloud computing is that CPU usage over the entire network can easily be monitored with the Sunstone [12] administrator's interface, thus gaining deeper insight into routing protocol's behavior.

In addition to obvious advantages in testing network protocols, simulating computer networks with cloud computing has advantages in education. For example, lab works in computer network classes can be based on simulations in the cloud. Thus we can easily dissect protocols and students will be more involved in interacting with network protocols. On the other hand, cloud environment gives the advantage for online course work and removes the constraint that students will have to be in the same physical location with the network equipment.

References

[1] https://community.emc.com/community/support/blog/2013/07/15/5-cloud-computing-trends-for-2013.
[2] Cisco Networking Academy: CCNP Advanced Routing Protocols.
[3] "Quagga software routing suite." [Online]. Available:http://www.quagga.net/
[4] G. Malkin, "RIP Version 2," RFC 2453 (INTERNETSTANDARD), Internet Engineering Task Force, Nov.1998, updated by RFC 4822. [Online]. Available:http://www.ietf.org/rfc/rfc2453.txt
[5] J. Moy, "OSPF Version 2," RFC 2328 (INTERNETSTANDARD), Internet Engineering Task Force,Apr. 1998, updated by RFCs 5709, 6549, 6845, 6860.[Online].Available:http://www.ietf.org/rfc/rfc2328.txt.
[6] Y. Rekhter, T. Li, and S. Hares, "A Border GatewayProtocol 4 (BGP-4)," RFC 4271 (Draft Standard),Internet Engineering Task Force, Jan. 2006, updatedby RFCs 6286, 6608, 6793. [Online]. Available:http://www.ietf.org/rfc/rfc4271.txt.
[7] D. Oran, "OSI IS-IS Intra-domain Routing Protocol,"RFC 1142 (Informational), Internet EngineeringTask Force, Feb. 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1142.txt.
[8] G. Malkin and R. Minnear, "RIPng for IPv6," RFC2080 (Proposed Standard), Internet EngineeringTask Force, Jan. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2080.txt.
[9] R. Coltun, D. Ferguson, J. Moy, and A. Lindem,"OSPF for IPv6," RFC 5340 (Proposed Standard),Internet Engineering Task Force, Jul. 2008, updatedby RFCs 6845, 6860. [Online]. Available: http://www.ietf.org/rfc/rfc5340.txt.
[10] Darren Johnson "Iperf Bandwidth Performance Testing v1.0".
[11] "Wireshark software" [Online]. Available:https://www.wireshark.org/.
[12] OpenNebula sunstone documents [Online]. Available:www.opennebula.org.