Design of Graphical Quagga simulator for analysis of Quagga Routing and QoS Framework

Veena M.
Department of Compuetr Science and Engineering
S.D.M.College of Engineering and Technology
Dharwar-580002,India

Dr. S. M. Joshi
Professor and Head,
Department of Compuetr Science and Engineering
S.D.M.College of Engineering and Technology
Dharwar-580002

Kiran B.Malagi
Asst. Professor, Dept. of Computer science & Engg
K.L.E. Institute of Technology
Hubli-580030

Abstract- Quagga is a Unix based network routing protocol that runs as a deamean (background process). It is a fork of popular Zebra framework.. One of the distinguished features of Quagga framework is inbuilt classes for QoS provisioning. The framework utilizes fuzzy based techniques for detecting and resolving network anomalies. However Quagga is a text based network suite, it is essential to build a simulation model that can be utilized by the administrator with QoS features of Quagga understand how Quagga actually works. As there is lack of simulation test bed for Quagga we develop a Quagga simulation framework. The objective here is to incorporate both Routing and QoS provisioning features of Quagga. The results of logging service are stored in a trace files. These traces are aggregated and comparative graphs are drawn. Results of Quagga are found better than non-Quagga based flat routing under overloaded condition.

Keywords – Java Network Simulator (JNS), Latency, PDR (Packet Delivery Ratio), Mean Squared Error (MSE), Packet per node.

I. INTRODUCTION

Many networks communicate with each other with the help of a Router. The router consists of a routing table which is a data table stored in a router that lists the routes to particular network destinations and also the distance associated with these routes. The routing table contains the topology of the network around it. The construction of the routing table is the primary goal of the routing protocols.

Whenever a host needs to send data to another host on a network, it must know where to send it. If the host cannot directly connect to the destination host, it has to send it via other hosts along a proper route to the destination host. For this it uses a routing table. Routing table keeps track of paths, like a map, and allows the gateway to provide this information to the host requesting this information. With hop-by-hop routing, each routing table lists all reachable destinations, the address of the next device along with path to that destination.

Routing table consists of at least three information fields:

1. The Destination network ID.
2. The cost of the path through which the information has to be sent
3. The next hop address to which the packet is to be sent.

Internet is very essential for this generation. To have a reliable high performance switching connection, routers are expected to be able to support increased transmission speed and switching speed

Contrary to this pc's were standardized to open multivendor market but routers were particular to architectures. That means incompatibility with routers and personal computers especially with the configuration and management procedures and also requirement of trained administrators to manage several architectural setups or to stick to a single vendor, which would lead us to purchase high cost networking devices, which comes with high performance and also setup complexity. Instead of sticking to single vendor or single hardware we can move to a open source software router which scales up the networking with the present architecture and low cost also. As we have large availability of open source for networking application like Linux, Berkley software distribution (BSD), XORP, Quagga.

Providing GUI interface to Quagga helps to analyze and troubleshoot the problem easily. Even the network animator software's gives us the Graphical User Interface to analyze the network but it does not allow us to modify or configure the network. In this stage our project fits in frame.

In this paper we are not only thinking of designing GUI interface to Quagga we are also proposing a system were consumption of bandwidth to all the node who are actively participating should be equally distributed among others and if any particular node which is consuming more bandwidth from other node are made to stop transmitting the packets for some time and resumed. This system is proposed to do fair decision among every node. For doing this Quagga provides a mechanism called fuzzy logic. Using this technique we can avoid unnecessary delays, jitter in the network.

II. Literature Review

Quagga is a fork of GNU Zebra which was developed by Kunihiro Ishiguro. Quagga is routing protocol suite which implements OSPFv2, OSPFv3, RIPv1, RIPv2, RIPng, BGP for unix, linux, Free BSD, Solaris and Net BSD. All these software share the same architecture for managing the various protocol daemons. Traditional routing software (such as GateD [1]) is made as one process program that provides all of the routing protocol functionalities as a whole.

Software implementations of routers based on standard PC hardware have been recently made available in the "open software" and "free software" world [3]. Here are some open source routers like Quagga: Modular Router: a software architecture based on Linux, and developed at the MIT, well documented, and freely distributed. -Xorp: an open router software platform under development at UC Berkeley. Xorp supports different hardware platforms, from simple PC, to specialized network processors, to dedicated hardware architectures. Xorp also supports a variety of routing protocols and control interfaces. -LRP: a free and open distribution of Linux that supports the main routing protocols. It allows a flexible and dynamic configuration of the router functionalities. -Freesco: a free and open distribution of Linux, similar to LRP, with reduced functionalities and simpler configuration.

In paper [3] the author discuss about the incompatibility with the PC- Architectures and the software and how the growing network environment is dependent on the networking devices. In order to fulfill this we are dependent on the network business model which leads to high cost. The author explains here that by switching to open source software which comes with open source routing stack implementation such as Quagga ,XORP, Bird etc., which run on any of the standard PC hardware. Quagflow is a transparent combination of Quagga routing protocol suite and openflow enabled hardware where moving completely from legacy protocol stack to logically centralized controllers using openflow protocol as the only communication channel with routers as the forwarding engines..

A propose of design of a bug-tolerant router that significantly reduces the likelihood of a software error affecting the network. Internally, bug-tolerant router consists of several virtual routers running in parallel. Each of these virtual-router instances is made different from the others, by modifying their execution environment (e.g., by reordering the routing updates they receive, by changing their configuration, or by modifying their layout in memory) or by modifying their internal structure (e.g., by running router code implemented by different programmers).[6]

In virtual routers as a service the author speaks about RouteFlow which is innovated by providing interfaces to multi-vendor networking hardware which are commercially available and open source software development. RouteFlow, is an architecture following the software-defined networking (SDN) [5] standard based on a programmatic approach topologically centralize the network control, unify state information, and decouple forwarding logic and configuration from the hardware elements. RouteFlow is a centralized way of enabling remote IP routing as it decouples the forwarding information base and control planes .by this way IP network becomes flexible and allowing us to make changes in customizing protocols and algorithms, making way to virtual routers and IP network as a service in software defined network.

One metric to evaluate the performance of router is the switching time. In OSPF(open shortest path first) routing protocol delay has been observed when there is a change in inter domain and intra domain routing topology change. This delay is observed under a standard PC running open source routing software Quagga. To provide the better performance to Quagga 9.8 the OSPF code is altered. Earlier in Quagga 9.8 in OSPF they were using Dijskstra's algorithm to find the shortest path when there is a topology change in router. The solution for this is provided by replacing the Dijskstra's algorithm by binary heap data structure. This optimization leads to better performance than conventional OSPF routers[8].

In the growing network era there is a need for reliable, efficient and high speed routers to maintain the communication or network unbroken. To fulfill this we need to deploy software routers for the betterment of services provided by router. This issue is discussed in paper DROP [9]. This paper explores the developing and giving a new distributed model for IP router control and management. DROP is partially based on the main guidelines of the IETF ForCES standard (open-source realization), and it allows building logical network nodes through the aggregation of multiple software routers, which can be used to forward the packets or to control different services provided by the router. According to ForCES guidelines DROP plans to extend router distribution and aggregation concepts to geographical level in order to provide and support upcoming challenges and functionalities in next generation networks.

Software routers are gaining attention in the mind of technology developers as they can be developed on the standard PC with less cost [7]. But router based on single PC are suffering from limited bus and Central Processing Unit (CPU), bandwidth, high memory access latency, limited scalability in terms of number of NICs and lack of flexibility mechanisms. Solution for this is to consider Multi-stage architectures created by interconnecting several PC's which results in i) increase the performance of single software routers, ii) scale router size, iii) distribute packet-forwarding and control functionalities, iv) recover from single-component failures, and v) incrementally upgrade router performance. The paper [10] describes a control protocol for multistage architecture based on PC interconnection. The protocol allows information exchange among internal PC's which supports configuration of the interconnected architecture, packet forwarding routing table distribution. The protocol is OS independent, since it interacts with software routing suites such as Quagga, XORP which is under test on a small scale prototype of multistage router.

For any type of router whether a hardware router which is designed to particular standard PC architecture or a software router which is flexible with any Personal Computer would be accepted by the industry only when it proves itself to be the best. Here the performance metrics is measured and compared between the routers. In the paper [11] the author says that to measure the performance of the software routers we need to undergo simulation which creates the test-bed to identify and predict the bottlenecks and to improve the system performance. The simulation is performed using network simulator ns-3 which realizes the newly innovative node model which is meant for intra-node contention and for managing the resources on the network. Through this simulation we can evaluate and predict the current and future software architectures.

Open source software routers are the alternatives for highly expensive hardware network devices because they can be installed on normal personal computer. If we consider the performance of a single personal computer it suffers from lack of bus bandwidth, high memory access latency and so on. To overcome this limitation a multistage architecture is presented which combines a layer-2 load balancer front end and a layer-3 routing back end using standard PCs and open source software. The mechanism is explained in paper [12]. Here the architecture is multistage bidirectional. In the first stage layer-2 ie, Ethernet switch acts as a load balancers which distributes the incoming packets uniformly by looking into the NIC addresses and in the second stage, layer-3 IP routing protocols which supports as a back end in doing complex task such as filtering, classification, forwarding etc as a traditional router using a traditional IP paradigm. This paper suggest that using multistage architecture the performance can be achieved better than the single PC software router by providing parallel paths.

Advance modern technology provides an ultimate chance to achieve effective results on the ground of open internet devices which is also known as Open Routers or Ors. In the paper [13] author reports about the detail optimization and testing carried on a PC Open Router architecture based on Linux software and commercial off-the shelf hardware. In this paper the author mainly considers the forwarding plane performance evaluation which is performed on different open router Linux based architectures. The scrutiny is done with both throughput, latencies and profiling which are external internal measurements respectively.

Only purchasing a good featured router dose not assure you to give a better performance unless it is supported by an operating system. So the combined work of the router and the operating system gives us a better result [16]. In this paper authors speak that router should switch from closed environment to open, general purpose communicating environment. To accept this change even open source router should also show the same performance which is shown by the commodity hardware routers. To meet this challenge the paper specifies hardware and software architecture for general purpose routers. The architecture has two key innovations. To achieve a good scaling property and in order to improve performance of the data packets to utilize minimum processing cycle it integrate the routers switching capacity and compute cycles, which is the first key change in architecture. The second key change is implementing programmable paths in software instead of forwarding paths from fast or fixed paths implemented entirely in hardware.

III. METHODOLOGY

Here we explain the working of the system by defining the design, modules, elements, interfaces, and Figure 1

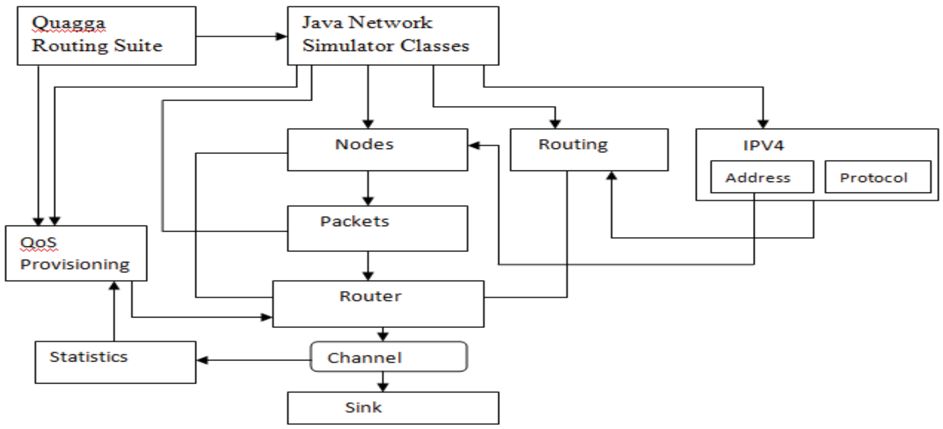

Figure1.Basic Diagram

shows the basic architecture of the network. Firstly group of nodes, distributed over a geographic area select a Router pertaining to that area. Once selected, this node acts as local data gathering and buffering node. It schedules all the domain member's reception and is responsible for channel allocation. It buffers the packets and forwards the packets to the sink node using IP routing. If any node is found to flood the network with too many packets, then the QoS framework of detects the node and bypasses the node for the current transmission cycle. The overall framework first compiles the Quagga library which is present as a C++ source. Basically Quagga framework is a daemon that runs Zebra, a unix based open source routing source. Using Java's native class support, the C++ source files are imported into java simulation using java network simulator package which in itself is an extension of NS2. This java package provides Quagga framework as four major components: IP addressing, IP protocol (DSDV and OSPF), QoS provisioning and packet structure. The simulator inherits Quagga's node structure into graphical nodes, thereby providing the graphics abilities to these nodes alongside the conventional networking being inherited from Quagga through JNS.

Nodes forwards packets which are IP packets encapsulating TCP frames. These packets are routed through the router which maintains a link to all the domain members as well as the routers of other domain. Therefore packets are routed using routing module extension through routers to the sink node. Sink is assumed to be gateway node that connects entire network with internet. Therefore the proposed simulation provides a realistic simulation of local network running Quagga where objective is to connect every node to the internet through their routers and the gateway node.

Finally the overall statistics of the simulation is gathered for analysis and comparison.

ALGORITHM

1. Make Network with N nodes over grid Topology of WIDTH x HEIGHT
2. Decide Number of Regions(domains).
3. Make the node with highest degree of neighborhood as Router
4. Other Nodes Join Router
5. Transmission

```
For( node=1:N)                                  Transmit data
      If(!node(i) is Router?)                   Transmit to Router
  Node(i) ->BW=Node(i)-> BW distance  between  node(i) , Router
  Else // Router
Gather data from other nodes.
Buffersize Data.
Transmit data to Destination
 Node(i) ->BW=Node(i)-> BW distance  between node(i) , Router
```

```
 if(Node(i)->BW<Threshold)
[OFN]=DetectOverfloodingNode();
For each ofn in OFN
WithdrawLink(ofn)
End   end
End
```

6. At the end of simulation, calculate pdr, packet per node, bandwidth.

IV. RESULTS AND DISCUSSION

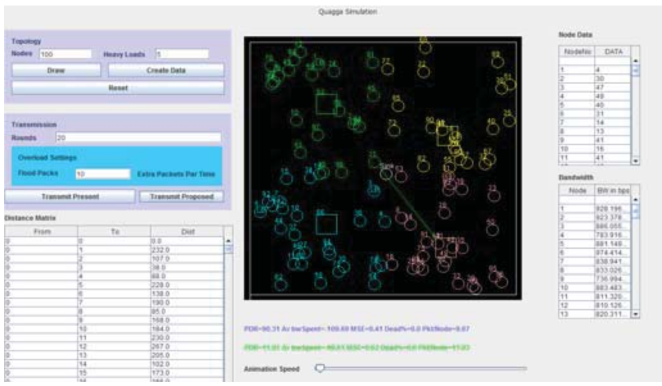

Figure 2 Snapshot 1

In Figure 2: A snapshot with node=100, Heavy loads=5, Rounds=20, Flood Packs=10

We can compare both present system and proposed systems performance matrices which is done as performance analysis to understand and analyze different performance parameters.

The Performance parameters of present system are displayed in green color and proposed system values are displayed in blue color

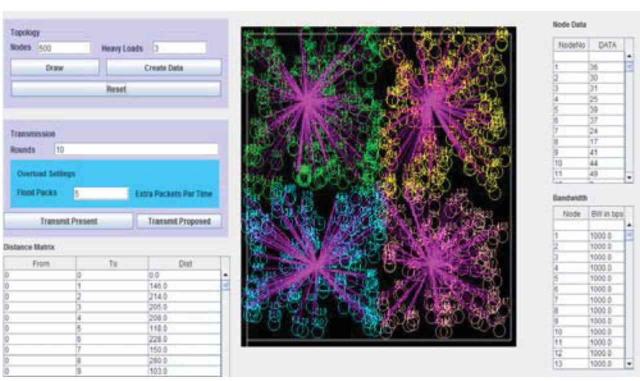

Figure 3: A snapshot with 500 nodes

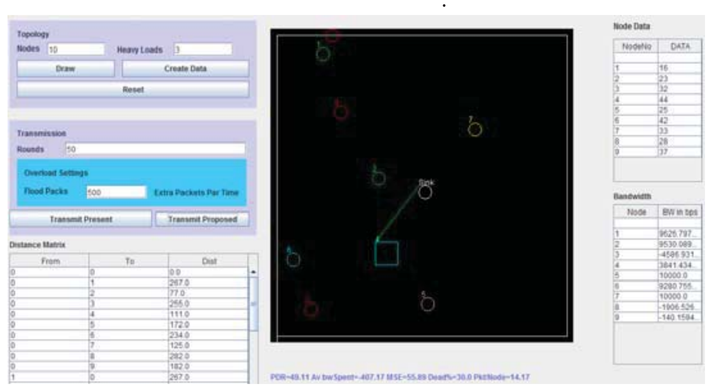

Figure 4 A snapshot with dead nodes

Figure 3 & 4 shows the results with node=500 and a snapshot with dead node displayed in red color respectively.

A) Performance Analysis

Here we are comparing the results by plotting the graph against the values obtained when the code is executed. Unless we compare and analyze the results we can't justify our work. In this section we discuss about the performance of the present and proposed systems. We are considering the attributes like number of Nodes, Packet delivery ratio, Mean square error, dead nodes and average bandwidth. For both present and proposed systems the resulting values are compared and graph is plotted. After the analysis for all the features considered the proposed system is showing better result. The following subsections present the analysis.

a.a) Packet Delivery Ratio (PDR)

PDR is the summation of number of packets received by all destinations and is divided by summation of number of packets sent by all the sources.

$$\text{i.e. PDR} = \frac{\sum \text{number of packets received by all destinations}}{\sum \text{number of packets sent by all sources}}$$

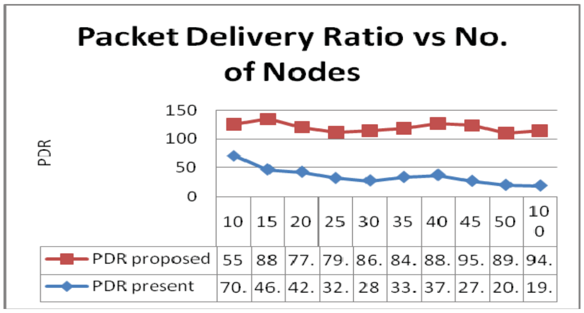

Figure 5: Graph of Packet Delivery Ratio

The PDR Vs No. of Nodes values are plotted(fig 5). x & y-axis represents number of nodes and PDR ratio respectively. Blue colour shows the present system and red colour graph line represents proposed system. By observing the graphs PDR is better in the proposed system.

a.b). Mean Squared Error (MSE)

Mean squared error measures the average of the squares of the "errors", that is, the difference between the estimator and what is estimated. MSE is a comparison of aggregated sink data with total network data. Actual data from all nodes is calculated and is expected that it should reach the sink. But the actual data that reaches sink is not the expected value. That is being compared and graph is plotted (fig 6) x & y-axis represents number of nodes and MSE respectively. We can conclude that the MSE of proposed system is reduced.

$$\text{MSE} = \sum(\text{ReceivedData} - \text{ActualData})^2/n$$

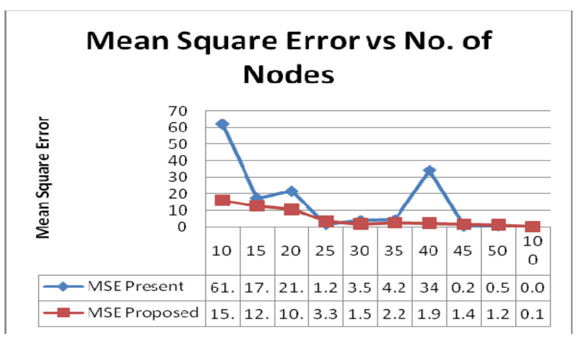

Figure 6: Graph of Mean Square Error

a.c). Packets Per Node (Pkt/Node) and Average Bandwidth utilization

A router is meant to send and receive the data packets. Packets per node are calculated based on the number of packets received at the sink to the number of packets generated by all the nodes. Present and proposed system and graph is plotted (figure 7 x & y-axis represents number of nodes and Pkt/Node respectively.

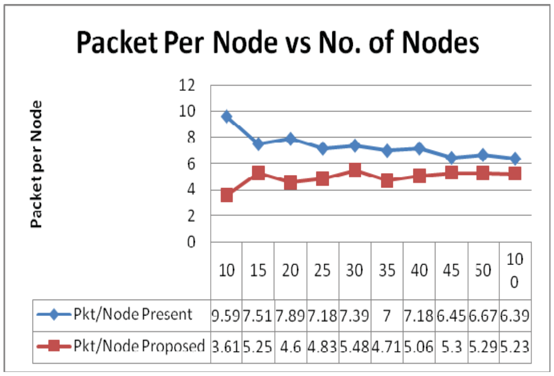

Figure 7: Graph of Packet Per Node

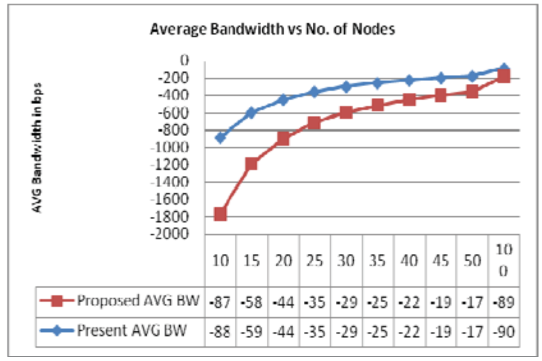

Figure 8: Graph of Average Bandwidth.

It is clearly seen that in the proposed system bandwidth utilization is efficient and hence the packets per node is also high. But in the present system the router aggregates the data and the router losess the bandwidth for transmission. The bandwidth utilization is shown in the graph (figure 8).

V. Conclusion and Future Work

Quagga is an open source network administration system that offers various routing modules shipped with QoS. Therefore unlike other networking suits, Quagga does not demand an additional QoS provisioning over the network. However the lack of simulation and visualization supports is major drawbacks of the system. In this work, this drawback is addressed and we have presented a system that offers near real time simulation for Quagga environment built on top of Network Simulator's Java build. We have shown that Quagga can better handle burst traffic and can offer better fairness to router nodes through bypassing bottleneck nodes. Results shows that the nodes can have better bandwidth with less packet errors and packet drops. The simulator is also tested for scaling factor and we have shown that the network simulation can handle heavy traffic and nodes both.

The work therefore justifies the adaptation of Quagga over conventional network and routing solution. This can be further improved by integrating administrative UI console with simulator and by provisioning a real time snapshot in the simulation environment that reflects the actual network.

References

[1] Gated®Enterprise2.0,NexthopTechnologies,Http://Www.Nexthop.Com/Products/Gated.Shtml.

[2] www.Quagga.net

[3] A. Bianco et al., "Open/Source Pc-Based Software Routers: A Viable To High Performance Packet Switching", proceeding of qos Ip, catania, 2-4 february 2005, pp. 353-366

[4] Marcelo Ribeiro Nascimento, Christian Esteve Rothenberg, Marcos Rogerio Salvador And Maurício Ferreira Magalhães "QuagFlow: Partnering Quagga With Openflow" Telecommunications Research And Development Center (CPqD) - Campinas - SP – Brazil University of Campinas (unicamp) - campinas-SP – Brazil {marcelon, marcosrs}@cpqd.com.br, {chesteve, mauricio}@dca.fee.unicamp.br

[5] K. Greene. Tr10: "Software-De Ned Networking". Mit Technology review, 2009.

[6] Matthew Caesar and Jennifer Rexford ,Princeton university " Building Bug-Tolerant Routers With Virtualization" Princeton, NJ Caesar@cs.uiuc.edu, jrex@cs.princeton.edu

[7] Katerina Argyraki, SalmanBaset ColumbiaUniversity, Byung-GonChun ICSI KevinFall "Can Software Routers Scale?", GianlucaIannaccone, AllanKnies Intel Research

[8] V. Eramo, M. Listanti, A. Cianfrani "Switching Time Measurement and Optimization Issues in GNU Quagga Routing Software " University of Roma "La Sapienza", INFOCOM Dept.

[9] "DROP: An Open-Source Project towards Distributed SW Router Architecture" Raffaele Bolla, Member, IEEE, Roberto Bruschi, Guerino Lamanna and Andrea Ranieri Department of Communications, Computer and Systems Science (DIST) University of Genoa

[10] "Control and Management Plane in a Multi-stage Software Router Architecture" A.Bianco *, R.Birke*, J .Finochietto t, L.Giraudo *, F.Marenco *, M.Mellia *, A.Khan:j:, D.Manjunath :j:* Dip. di Elettronica, Politecnico di Torino, Italy, Email: {firstname.lastname}@polito.it

[11] "Measurement and Simulation of High-Performance Packet Processing in Software Routers" Torsten Meyer, Florian Wohlfart, Daniel Raumer, Bernd E. Wolfinger and Georg Carle University Hamburg, Department of Computer Science, Telecommunications and Computer Networks {meyer|wolfinger} @informatik.uni-hamburg.deTechnische University München, Department of Computer Science, Network Architectures and Services {wohlfart|raumer|carle}@net.in.tum.de

[12] "Scalable Layer-2/Layer-3 Multistage Switching Architectures for Software Routers" Andrea Bianco, Jorge M. Finochietto, Giulio Galante, Marco Mellia, Davide Mazzucchi, Fabio Neri Department di Elettronica, Politecnico di Torino, 10129 Torino, Italy, Email:{firstname.lastname}@polito.it Networking Lab, Istituto Superiore Mario Boella, 10138 Torino, Italy, Email: galante@ismb.it

[13] "Linux Software Router: Data Plane Optimization and Performance Evaluation " Raffaele Bolla and Roberto Bruschi DIST - Department of Communications, Computer and Systems Science University of Genoa Via Opera Pia 13, 16145 Genoa, Italy Email: {raffaele.bolla, roberto.bruschi}@unige.it

[14] "The IP Lookup Mechanism in a Linux Software Router: Performance Evaluation and Optimizations" Bolla, R. ; Univ. of Genoa, Genoa ; Bruschi, R. Published in: High Performance Switching and Routing, 2007. HPSR '07.

[15] https://www.sdxcentral.com/articles/interview/quagga-project-martin-winter-interview.