Analyses of QoS Routing Approach and the Starvation's Evaluation in LAN

Ariana Bejleri
Polytechnic University of Tirana
Faculty of Information Technology
Computer Engineering Department
Tirana, Albania
arianabejleri@yahoo.com

Igli Tafaj
Polytechnic University of Tirana
Faculty of Information Technology
Computer Engineering Department
Tirana, Albania
itafaj@gmail.com

Aleksander Biberaj
Polytechnic University of Tirana
Faculty of Information Technology
Electronic and Telecommunication Engineering Department
Tirana, Albania
a.biberaj@yahoo.com

Ermal Beqiri
Tirana University
Mathematics & Statistics & Applied Informatics Department
Tirana, Albania
ermalfr@yahoo.fr

Julian Fejzaj
Tirana University
Faculty of Natural Sciences
Department of Computer Science
Tirana, Albania
Julian.fejzaj@fshn.edu.al

ABSTRACT

This paper gives a survey in QoS Routing Architecture implemented by Dijkstra's algorithm. The performance of QoS Routing architecture is evaluated by made a comparison between the Shortest Path Routing and QoS one. A very important feature in QoS routing are the conditions for elimination of starvation. Experimentally we have evaluated the number of packets delivery from source node to destination one in QoS Routing architecture with high and low priority classes based on ns-2 simulator.

Keywords

QoS routing architecture, Dijkstra's algorithm, shortest path routing, starvation

1. INTRODUCTION

QoS Routing (Quality of Service Routing) is the architecture that takes into consideration some elements such as the necessary bandwidth, delay, throughput, jitter. Based on these elements this architecture selects the specified path that satisfy the Quality of Service requirements. In [4] QoS Routing gets the information about network state and resources availability. Also it based on some factors such as: Dynamic determination of the feasible paths which can accommodate the QoS requirements in data flow over the network, the optimization of resources uses etc. QoS helps workload balancer in network to manage the resources. So it should affect in performance improvements of the total throughput. QoS Routing has the ability to provide better throughput in the network than Best Effort Routing (Internet Communication).

QoS Routing finds the feasible path in order to guarantees the routing information but it cannot ensure the stability of selected path. Resources Reservation Protocols can be used to allocate the necessary resources during the data transmission in the selected path. One of the most popular protocol reservation is RSVP (Resource Reservation Protocol). It is receiver oriented protocol, which means that the receiver of the data flow is responsible for the reservation of resources.

Qos Routing has a higher level admission control mechanism [5] in order to ensure that the selected path does not utilize the resources network for a long time.

Shortest Path Forward (SPF) Routing based on number of hops from source to destination. Often the SPF Routing leads to unbalanced traffic, so it can cause the congestion of packets in network while the QoS Routing architecture provides a better performance in both of them during the data transmission. Traffic Engineering in QoS Routing is the process of traffic flow's arrange in network by avoiding the congestion.

Constraint-based Routing (CBR) architecture includes the QoS Routing architecture. Constraint-based Routing is a general definition, which combines QoS Routing (i.e bandwidth, delay, loss) and Policy Routing. It is an extension of QoS Routing architecture by take in consideration, QoS requirements, network policies and utilization of the network resources in order to prevent the congestion traffic [2].

The goal of CBR is to enable a new routing paradigm with special properties, such as Resource Reservation-Aware and Demand-Driven. The Information from different resources offer the possibility of selection specified path.

The first goal in this paper is the comparison between QoS Routing and Shortest Path First Routing which significantly based on the number of hops. This paper is organized as follows. In Section II, we presents the QoS Routing architecture based on Dijkstra Algorithm. At this topic we theoretically examine stability, robustness and scalability. In Section III, we briefly discuss some experiments based on ns-2 simulator and benchmark tool called Httperf. This tool will compare the performance of QoS Routing with Shortest Path Routing by presenting the Starvation results. At the end of this paper are the conclusions and references.

2. ROUTING ALGORITHMS APPROACH

Routing Algorithms are the basics elements in the selection of necessary path. Wrong selected path causes the leak of utilization resources and services. In our days a very important topic is utilization of services in order to give a satisfaction for every internet user. QoS Routing Architecture is a basic constrain which can implement in all kinds of Routing. The selection of a path is not easy, because there are a lot of diversity of the network topologies. Some of them are network mobile and most of them haven't any solid infrastructure (Such as Ad-Hoc Networks). Nevertheless it follows the same way for all routing communication approaches. In figure1 we give an example of path selection. In this figure presented a communication form between 2 users. All the communications are supported by ethernet interface with utp cat 7 technique. Also we have presented a network with two broadcasters which routes video packets stream based on Dijkstra Algorithm. As we presented in this figure, there are some possibility (links) for routing packets. The Dijkstra Algorithm does check of the whole available paths and finds the best one. To find the better solution it should supported from two features: Number of hops and link states. Some different protocols are built on these approaches, i.e RIP (Routing Information Protocol) which is based on Number of hops between hosts. This algorithm has a disadvantage because it generates a lot of traffic (for every path-finder, the router sends a full routing table to the neighbors router) . Another one is OSPF (Open Short Path First) which is based on Link State Routing information. This algorithm sends packets from source to a target destination based on state of the link such as throughput, delay, packets drop etc. These features are depth analyzes in QoS Architecture which is built on each router in figure 1. This architecture based on 4 elements: Bandwidth which is a concave metric. This denotes that some bandwidth should be available (reserved) for QoS flows. This resource evaluated as min/max approach, because the request for minimum (bottleneck) bandwidth on the path is necessary for transmission of packets. The goal of QoS Routing is to find the feasible path with the maximum bandwidth [1].

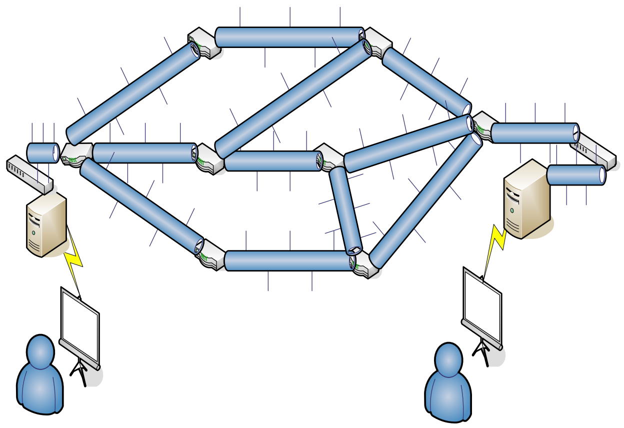

Figure 1: Communication between two users with Dijkstra's algorithm.

Delay, Reliability and Jitter. The last three elements are additive metrics. The Delay Metric causes the latency of packets which are transmitted between source and destination over the network. For delay-sensitive requests, some of the links can be pruned from the graph before selecting the path. The Reliability assigns the acceptable data loss rates, which previously are guaranteed through the reservation protocols. Jitter offers the variable time delay of packet transmitted from source to the destination.

In the former way [1], Additive and Concave metrics can presented:

Additive if w(P) = w(u1; u2) + w(u2; u3) + : : : + w(ul¡1; ul) (1)

Concave if w(P) = min(w(u1; u2);w(u2; u3); : : : ;w(ul¡1; ul)). (2)

Based on these features QoS put some numbers for each path destination. Dijkstra's algorithm take a decision for routing packets based on some numbers which are implemented for a QoS Routing. The Router device is a Forwarding Packet Core Architecture. Each packet which arrive at destination router, initially buffered, then it transmits according to Dijkstra Algorithm.
In our example in figure 2 video packet stream from source to the destination should be transmitted by following this path:
R1-R5-R6-R4-Switch-Server Destination

Let's give an algorithm for all required steps:

1. R1-R5 = 2, R1-R7 = 5,. Dijkstra Algorithm utilizes the shortest distance.
2. R5 – R6 because this is a single route to destination.
3. R6-R4 because there is smaller value in the path (valuated from QoS =1)
4. R6-R3 is equal 3. So the final path from source user to destination is: R1-R5-R6-R4.
5. If selected another path from R1 Router: R1-R7 (Value =5)
6. R7-R8 because these is the only path remained.
7. At this point there are 2 possibility: R8-R2 and R8-R4. Their values are 3.
8. If R8-R2 is selected, there are 2 possibilities: R2-R1 or R2-R3. If the first path is selected, an fatal error will occur, because R1 is the initially router. R2 router couldn't sends the packets in R1 router because at the first moment it's routing table is updated previously from R1. Thus there is just one possibility R2-R3.
9. Again in router R3 there are two possibilities: R3-R6 and R3-R4. As it shown in figure 2, values in R3-R6 = 3 and R3-R4=5. The small value is R3-R6. This is a temporarily destination.
10. The only way for forwarding packets from R6 Router is R6-R4 because routing table is previously updated from R3 and R5 [1],[6]. The destination packet should be sent to R4
11. R4 is previously updated from R8 Routing Table too.
12. Finally the selected path is R1-R5-R6-R4-Server

Dijkstra Algorithm is not the only algorithm for the selection of available path. One of the interesting algorithm is Bellman – Ford. Both algorithms are used to assign the policy of data communications between source and destination. Below we are presented the difference between those algorithms [7]:

Dijkstra's Algorithm
1) Dijkstra doesn't work for negative weight edges.
2) Time complexity of Dijkstra is O(|E| + |V|Log|V|)
3) Dijkstra's algorithm is usually the working principle algorithm which is behind link-state routing protocols, OSPF and IS-IS

Bellman-Ford's Algorithm
1) Some kinds of Bellman–Ford algorithm is used in distance-vector routing protocols.
2) Bellman-Ford works for non negative weight edges.
3) Time complexity of Bellman Ford takes O(|V||E|) time which can also be written as O(|V|^3)

Some CBR challenges which are based on QoS constrains are stability, robustness and scalability.

In the first constrain every node in a network must maintain local state information. Those information include available bandwidth, queuing management and propagation delays. A reasonable question is how frequently the Routing Table updates inside a Router. High frequency of updates, increase the traffic engineering and routing overhead, on the other hand if we minimize the frequency update, we will get the inaccurate information. To balance this tradeoff, generation of the update packets can be advertised whenever there is a significant change in the values of the resources [3]. There are two improvement ways: absolute scale, which divide the range of values into equivalence classes, and relative scale, which triggers the update when the percentage of change since the last advertisement exceeds a given threshold.

Robustness means allocation of an adaptive route. When resources are sending inaccurate information in the specified router, the QoS selection path is not the best one. Based on this inaccurate information a fatal error can occur. The Robust path means that the selected path offers permanent QoS application services during the different workloads and different times. After we commit some tests (as much as we can) we can select the robust path which satisfy the QoS requirement. This is an advantage in Virtual Circuit Switching Technology.

Scalability is the situation when growth and shrink of network edge and network core do not affect directly on the network and applications performance.

The final situation is evaluation of starvation in QoS routing environment. If some packets with different priority classes shares the network, packet with low priority can dropped. This happen because each packet class has it's own time of life. If the time for this packet located in buffer of any nodes exceeds, it means that this packet is not transmitted yet to the destination node, so the probability that packet drops, increases. In third topic we will study this situation by using some experiments.

3.EXPERIMENTAL PHASE

Initially the experiments do not belong to Dijkstra and Bellman Ford Alogrithm, but only with QoS Routing Algorithm and the ratio of this algorithm with Shortest Path Forward Algorithm. Second we have analyses the Starvation in QoS Routing Algorithm. These topic is organized in 2 phase. In the first one, we want to test QoS Constrains depends on network size and in the second phase we want to evaluate the necessary bandwidth for different packet classes in order to minimize the starvation.
At first we can show briefly what kind of tools we have used in our experiments. All simulations are organized on a host computer.

- Ns-2
- Nam
- Tools_Patch_Ns2_for_Graph

We have organized some experiments with this simulator, by using two algorithms: Shortest Path First algorithm, which is based on number of hops and QoS algorithm which is based on both: Distance Vector Routing and Link State Routing.
All graphs are generated from nam tool.
Initially, we can generate automatically by using a script in C Language up to 400 router nodes with ns2 simulator in Full Mesh topology, Full duplex link and supported by RIP version 2 policy of routing. In this script we have assign a IP's range of C class. This script automatically plays similar role with DHCP (Dynamic Host Control Protocol) service. We have created a network environment by using the nam tool. This script is located in /proc directory of Linux CENTOS 5.5 installed in my laptop. Each computer takes one IP. The results of simulation are presented in table 1.

Table 1: The Ratio Between Network Size and CPU Time Consuming by Using Two Algorithms

Network Size	CPU consuming (ms) in SPF	CPU consuming (ms) in QoS
50	100	230
100	720	990
150	1080	1700
200	1570	2500
250	1910	3010
300	2000	3560
350	2060	4020
400	2320	4900

As it looks in table 1 and in figure 2 SPF algorithm growth in linear form, as much as number of network size and time processing increases. Also the QoS algorithm has a linear curve but it consumes more times than SPF algorithm. The reasons are that QoS algorithm should execute more values than SPF algorithm (So it can processing not only the number of hops, but throughput, necessary bandwidth, delays of packets too). For these reasons it causes more delay than SPF Algorithm.
Another experiment is elimination of Starvation for three packet classes. This experiment is performed in a real application.

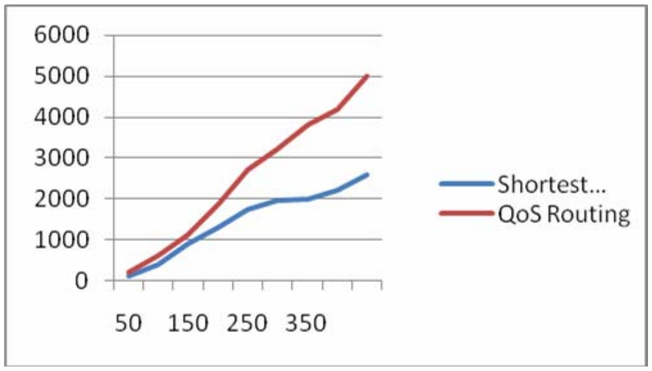

Figure 2: Ratio between network size and time consuming with the two algorithms.

In this graph in x-axis is network size and y-axis is time consuming in network. Graph with red color is referred to QoS algorithm and with blue is Shortest Path First algorithm

3.1 Experimental Environment

- There are 2 VM (Virtual Machines) above a Host Physical Machine with Win XP 32 bit (The physical host parameters are the same as we mentioned above)
- Type of Virtualization: Full Virtualization (VMWare Workstation 7.1)
- In the first VM is built XP Operating System
- In the second one is built CENTOS 5.5 Operating System
- Both of them uses 500 MB RAM.
- CPU 1 for each VM
- Core 2 for each VM
- WAMP5 1_4_7 installed on VM1 (WAMP-Windows, Apache, My SQL, Php)
- Httperf benchmark
- Client 192.168.1.10 255.255.255.0
- Server 192.168.1.100 255.255.255.0
- Video_Tooling_4_all_TKO

As it look from the description, we have used 2 virtual machines which can communicate with each other as a team in LAN. The experiment is organized for 2 VM in the same computer.
The experiment is repeated again with 2 physical machines which support respectively 2 VM. Both physical machines are connected by twisted pair utp cat 7 cable.
The final experiment is tested with 4 physical machines in LAN which are connected with switch gigabit Ethernet. All machines support respectively 2 VM, as it shows in figure 3.

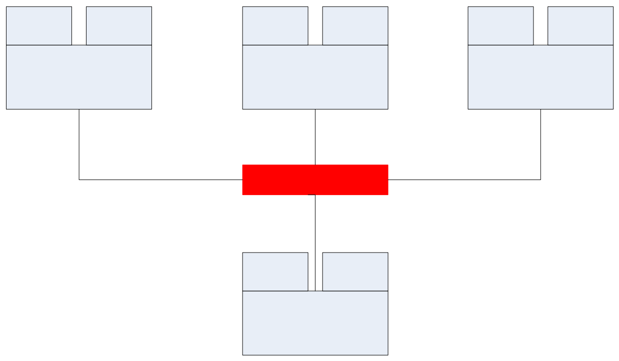

Figure 3: Four computers which are connected by gigabit switch. Each computer has two VM.

In our experiment we have used the manageable switch. The situation is the same if we use just a simple switch. Packets are transmitted in Layer 2 of OSI (Open System Interconnection) model, by using the broadcast technique. Each computer has its own gigabit Ethernet card and there are connected by utp cat 7 cable (Unshielded Twisted Pair Category 7). All computers have the same architecture as there are in the first experiment. Each computer is configured with C class IP (Internet Protocol) private.
There are three packet classes. In the first class there are Video Stream packets. In the second there are generated FTP (File Transfer Protocol) packets and in the third one the Email packets. In the video stream packet we have set the priority=5. In FTP=2 and Email=1. All these priority are assigned by "Video_Tooling_4_All_TKO. Bandwidth communication between 2 hosts is 1 Gb/sec in LAN and 5 Gb/sec in the physical machine.
For the first experiment , a benchmark, called httperf is used for generating the packets from Client (CENTOS 5.5) to Server (XP_OS)
From the client side we can generate randomly 6000 packets to Server destination:
Httperf –server 192.168.1.100 –uri /TKO.html –num-con 6000 –rate=10 –timeout5
TKO.html is a html file in Apache Web Server which is used from Video_Tooling_4_All_TKO. This method utilizes these packets based on our assigned priority. TKO.html randomly distributes packets classes.(An additive module is included in this benchmark - This is not the study of this paper)
From this experiment we introduced that Video Calling packet would generate 94% of total packets, FTP 5% and E-mail 1 %.
If we want to use Email more than 1% of time, a starvation will occur.
We can repeat the experiment for 2 physical machines with 2 virtual machines above respectively. As we mentioned previously, both physical machines can communicate with twisted pair utp cat. 7 cable. The results are:
Video calling packet would generate 92 % of total packets, FTP 6 % and E-mail 2%. If we make a comparison between first experiment and second one we can see that video stream packets reduce from 94 % to 92%, FTP packets from 5% growth to 6 % and Email packet from 1 % up to 2 %. The reason is the bandwidth. At the first experiment the speed of communication is 5 Gbit/sec but in the second case it is just 1 Gbit/sec.
The final experiment shows us the situation of 4 computers which are connected with a gigabit switch in LAN. In a computer we can generate randomly the traffic packets with httperf benchmark. The results change from both previously experiments. Generation of Video calling packets reduce to 90%, FTP packets reduce to 7% and E-mail packet reduce to 3%. We got these results because of the switch which introduces the complexity of hardware architecture. [6] . This situation is reflected in packets with big sizes. So the bandwidth for small packets with low priority will increases and the total number of these packets will increases too.

Table 2: Ratio Between Video Stream, FTP and E-mail Packets with Different Topology

Video Stream	FTP	E-mail	Topology
94%	5%	1%	Inside Physical Machine
92%	6%	2%	Twisted Pair
90%	7%	3%	Switch

4. CONCLUSIONS

As a conclusion of this paper we can show that the QoS Routing Architecture is a big factor in routing technology. It asks better performance than other routing architectures. As it looks from the experiments, if the time of processing is increases the number of network size will increases too. The evaluation of starvation is a very interesting thing which is very important in QoS architecture. For three different generated classes in server only email could be hurt from the starvation. Anyway the risk of email's starvation reduces, because of some troubles which are induced from packets with big size (i.e video stream packet). So the total consumed bandwidth from video stream packet will be reduced. The remain part of bandwidth can utilized from smaller packets (The reasons of this situation are not study in this paper)

In the future we will verify the starvation between different classes in order to satisfy QoS routing Architecture in Network with different topology**.**

5. REFERENCES

[1] Andrew Tanenbaum, "Computer Network 4th Edition", Chap 4 pp. 205–220, 2007

[2] Awduche, D. Malcolm, J. Agogbua, J. O'Dell, M. and Mc-Manus, J, "Requirements for Traffic Engineering over MPLS," *RFC 2702*, 2003

[3] Apostolopoulos, G. Williams, D. Kamat, S. Guerin, R. Orda, A. and Perzygienda, T, "QoS Routing Mechanisms and OSPF Extensions," *RFC 2676*, 2003

[4] Crawley, E. Nair, Rajagopalan, R. B. and Sandick, H,. "A Framework for QoS-based Routing in the Internet," *RFC 2386*, 1998

[5] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, "Quality of service based routing: a performance perspective", In Proceedings of ACM SIGCOMM'98 Conference, pp. 17–28, 1998

[6] James Curose "Network Computing", p.188, 2009