INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(3):405-413, June 2016.

Influence of the QoS Measures for VoIP Traffic in a Congested Network

R.L. Luca, P. Ciotirnae, F. Popescu

Robert Luca, Petrica Ciotirnae*, Florin Popescu
Military Technical Academy
Romania, 050141 Bucuresti, Bdul George COSBUC, 39-49
robert.lucian.luca@gmail.com, ciotirnae@mta.ro, fdpopy@yahoo.com
*Corresponding author: ciotirnae@mta.ro

Abstract: The paper revolves around the subject regarding quality of service (QoS) in a telecommunication network. The chosen scenario is based on the transmission of data and voice packets using a WAN connection, which has a limited bandwidth and emphasize the need of implementing QoS mechanisms in order to fulfill the quality requirements of the traffic, especially for VoIP. This topology will outline the impact and importance of the QoS implementation, illustrated by the desired quality resulted through VoIP traffic simultaneously with maintaining the data conectivity using a lower bandwidth for applications which require a smaller amount of QoS properties, such as FTP.

Keywords: Quality of Service (QoS), VoIP, Weighted Fair Queuing (WFQ), Low Latency Queuing (LLQ), Class-Based Weighted Fair Queue (CBWFQ).

1 Introduction

This paper emphasize a method to implement mechanisms that provide Quality of Service (QoS) to VoIP services. The QoS problem is a subject that will be studied continuously, according to other papers which studied this matter [1], [2]. There are a few challenges that concern the VoIP implementation: finding solutions to transmit a real-time service using a data network, which offers no guarantees regarding the delivery of the service, by using the best effort mode. This type of technology is successful only if the features that inffluence the quality of service in data networks are being decreased: latency, jitter and packet loss (see [3]- [8]). Moreover, QoS represents one of the 4 characteristics which a network has to provide in order to fulfill the clients expectations, beside scalability, security and fault tolerance [9].

2 General description of the used mechanisms

In order to implement the QoS mechanisms, it is neccessary to use specific mechanisms depending on the type of network and service used. Some of them are described in the paragraphs below. The **Weighted Fair Queuing** (WFQ) [10] mechanism creates dinamically queues based on traffic flows. A packet can be asigned to a certain type of flow depending on the value obtained through applying a hash function on the source IP address and destination address, source port and destination port or ToS value.

This type of flow will have his own queue. The maximum number of queues that can be recorded on the interface is 4096, which is much higher than other similar mechanisms (ex: **Priority Queue** which allows the network administrator to configure four separate buffers with different priorities: high, normal, medium and low. Packets in buffer with higher priority are always processed before packets in lower priority buffer; **Custom Queue** - allows users to define up to 16 queues. Each queue is processed at a time and can be transmitted according to the parameter "weight" (set by user) a certain number of packets or bytes). In **WFQ**, a flow exists as long as there are packets in that stream waiting to be transmitted. In other words, when the queue allocated to that stream is empty, it is removed . For this reason the number of queues in the system is changing rapidly .

The mechanism which allocates bandwidth among different streams is based on a factor (cost, weight), which depends on the **Precedence** field for the given buffer. When there is a free space in the hardware buffer, the WFQ mechanism takes a packet from the software buffers and places it in the hardware buffer. It will be chosen the packet with the minimum "Sequence Number" (SN) among all queues. SN value is assigned to a packet before it is placed in the queue of the associated stream and also before taking a decision on its eventual disposal (the WFQ mechanism uses a modified "tail-drop" algorithm which takes into account the SN value before discarding packets).

The formula used to calculate the SN of a packet:

$$SN = SN_{ant} + (w \times dim) \tag{1}$$

where: SN_{ant} represents the SN of the previous packet, w represents the cost (weight)and has the formula:

$$w = \frac{32348}{I_{pp} + 1} \tag{2}$$

(I_{pp} represents the value of the IP Precedence Field) and dim represents the packet dimension. From the expression (1) we can have some conclusions:

- The calculation of the SN takes account of packet size, being smaller for smaller packets;
- Packets arriving in queues that already have a large number of packets will get a higher SN whereas the calculation of the SN takes account of the previous packet SN ;
- It can be seen that the SN varies inversely with the value of the Precedence field. The higher the priority of a flow, the lower the cost value "w" and hence the SN.

In **Fig. 1** it can be seen how the packets are scheduled for transmission in two different queues that have the same cost w.

If the A flow from **Fig. 1** would have a bigger cost than the cost of stream B it would be possible that SN A1 be greater than SN B1 and the packet "B1" may be transmitted before the transmission of packet "A1", situation which is graphically depicted in **Fig. 2**.

WFQ algorithm automatically creates eight queues with smaller weights (high priority) for management traffic generated by the router (messages from routing algorithms, control traffic of OSI layer 2, etc.). WFQ works well for networks where delay sensitive traffic requires less bandwidth than the average of other flows. The method has the advantage that it does not require manual configuration of classes and can thus be used as the default method when traffic characteristics are unpredictable and classification is difficult [12].

Disadvantages of this method: no guarantees for the delivery of a specific type of traffic and the fact that this algorithm doesn't have a high priority queue that would provide minimum values for delay and jitter for VoIP traffic.

Another mechanism that can be used for providing QoS measures is **Class-Based Weighted Fair Queue (CBWFQ)** [10], [11]. This congestion management mechanism permits the classification of packets into classes with costs (weights) according with the bandwidth specified by the administrator for each class. CBWFQ band division is performed inversely to the cost assigned

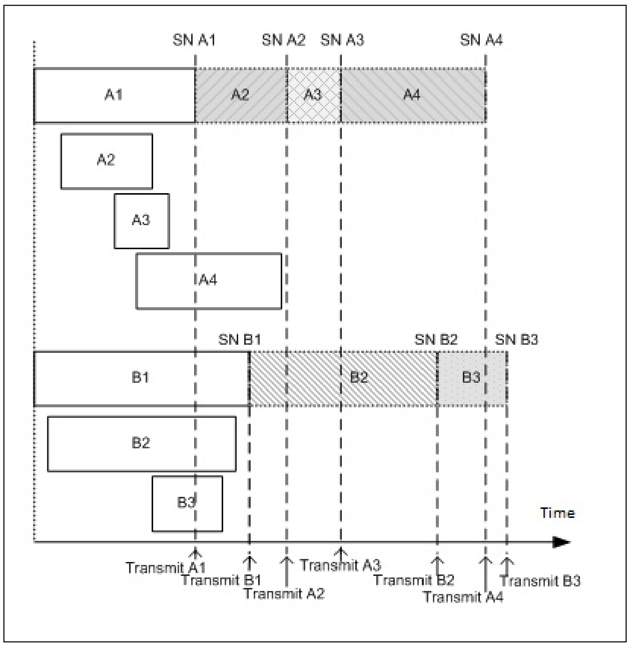


Figure 1: Programming packet transmission according to the SN (same cost w) [14]

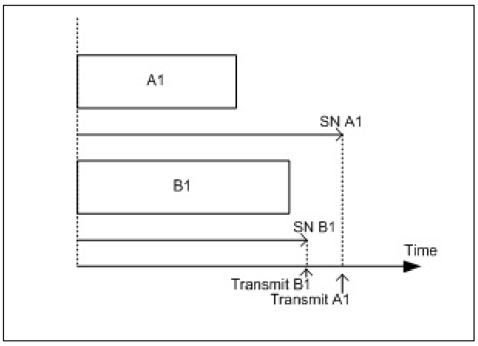


Figure 2: Programming packet transmission (different cost, $w(A) > w(B)$)

to the flow. The method in which bandwidth is shared between the streams is described by the following equation :

$$share(k) = \frac{w(1) + w(2) + ... + w(k) + ... + w(N)}{w(k)} \tag{3}$$

Equation *(3)* outlines the fact that the flows which have a smaller "w" cost get more bandwidth. Unclassified traffic (which does not fall within any user-defined class) is sent to a special class (default) using WFQ mechanism and the weights defined by it according to the equation *(2)*.

The cost of user-defined classes can be determined from the following formulas based on how the bandwidth command is applied to a certain class (as the number of bits or as a percentage of the total bandwidth) according to the following equations:

$$w(k) = ct\frac{B_{w_{tot}}}{B_w(k)} \tag{4}$$

,where ct depends on the number of queues in the system (inversely to the number), $B_{w_{tot}}$ represents the available bandwidth on an interface and $B_w(k)$ is the bandwidth configured for a specific class as the number of bits.

$$w(k) = \frac{ct \times 100}{B_{w_{percentage}}(k)} \tag{5}$$

$B_{w_{percentage}}(k)$ represents the bandwidth configured as a percentage from the total bandwidth.

Each manually configured class represents a separate queue managed by a FIFO algorithm that receives a percentage of the total bandwidth depending on how the packets are chosen for transmission to the hardware buffer. After configuration, the CBWFQ algorithm works similar to WFQ, which means that it is based on the SN value defined by the same formula as in WFQ, equation *(2)*. It can be demonstrated that any user-defined class dominates the dynamically determined flows (WFQ) almost all the time (except when certain classes are configured with a very low percentage 1.5%, which is unlikely) [13].

Same as WFQ, a predefined set of queues is allocated for the CBWFQ mechanism called "Link Queues" that the system uses for transmission of network management traffic. These buffers are set at a fixed cost $w = 1024$, a cost which is much better compared to the cost of any dynamic flow and about the same level as the traffic classified by the user. Due to the fact that this type of traffic is intermittent, the bandwidth distribution is not affected. Moreover, there is a rule that user-defined traffic classes should not reserve more than 75% of the available bandwidth on an interface, such that these special buffers do not remain without bandwidth [15].

Low Latency Queuing mechanism (LLQ) or "**CBWFQ with Priority Queue**" [11] complements CBWFQ mechanism and allows you to specify a traffic class with high priority and therefore minimal latency. The delay sensitive traffic can be transmitted before other types of traffic by using a special buffer with the highest priority (cost w = 0). In **Fig. 3** it can be seen the essential difference between LLQ and CBWFQ mechanism, namely the existence of the priority buffer. This is a buffer with the minimum cost (0) which means total priority. In other words, the traffic that will be selected to use this buffer will always be sent before other types of traffic. For this class it applies a "policing" mechanism (when the configured rate is exceeded, packets are automatically discarded) to avoid the situation in which traffic with priority monopolizes all available bandwidth on an interface and starves other traffic classes that have been configured for guaranteed bandwidth. Due to the fact that in the case of VoIP, traffic packets are sent at regular intervals, the transmission rate is relatively constant and for that reason VoIP is the best choice for the priority buffer. Due to the advantages of minimum values for delay and jitter,LLQ is one of the most recommended method of "queuing" in literature and represents the version tested in the practical application presented below.

On the low-capacity links, LLQ can be used with other QoS mechanisms such as header compression and fragmentation and interleaving (to avoid the situation in which voice packets must wait for the transmission of large data packets, which would lead to significant delays and jitter).

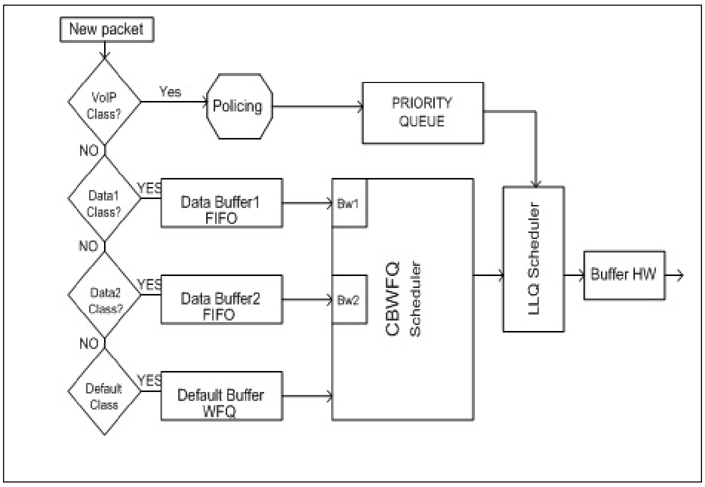


Figure 3: LLQ mechanism operation [16]

3 Experimental results

For laboratory testing the following scheme was used for the interconnection of the equipments (**Fig. 4**):

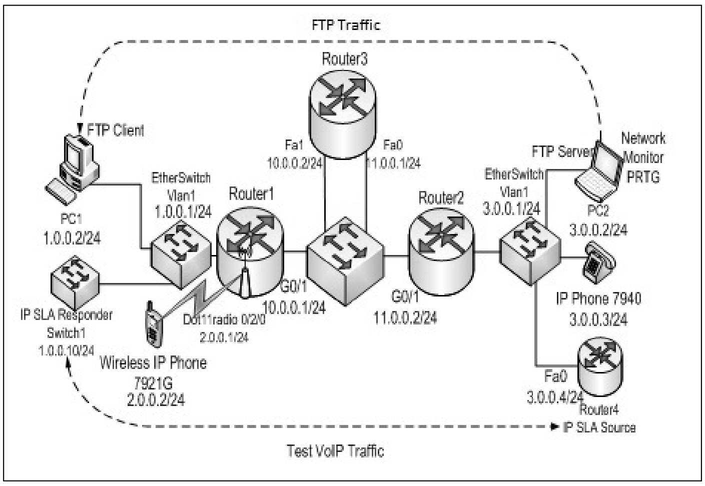


Figure 4: Logical scheme of the equipments interconnection

The aim of the laboratory tests is to highlight and compare the statistical values obtained for voice traffic in VoIP format in two distinct situations. The first case refers to a congested network in which it does not take into account the priority of traffic classes and the second situation where specific QoS measures are applied.

Functional laboratoty platform description:

- VoIP service is provided by the CME server (Call Manager Express) on router 1 for the wireless network 2.0.0.0/24 and CME server on router 2 for the network 3.0.0.0/24. Between the two servers there is a trunk for data link;
- computer at IP 1.0.0.2 will play the role of a FTP client that will receive multiple streams of data from the laptop with the address 3.0.0.2 that will be the FTP server. Between router 1 and router 2 it will be generated so much traffic as possible so that congestion occurs.
- in addition, on the link between router 1 and router 2 the rate is limited at 500 kbps. This limitation has been achieved using an additional router (router 3) configured in such a way that it receives all the traffic transmitted between router1 and router 2.
- traffic received by the router 3, regardless of its source or nature is selected by access lists and configured on router 3 as a single class of traffic. For this class it applies a shaping policy at 500 kbps (limiting the transfer rate at a given value while maintaining the packets in a buffer when the limit is exceeded).

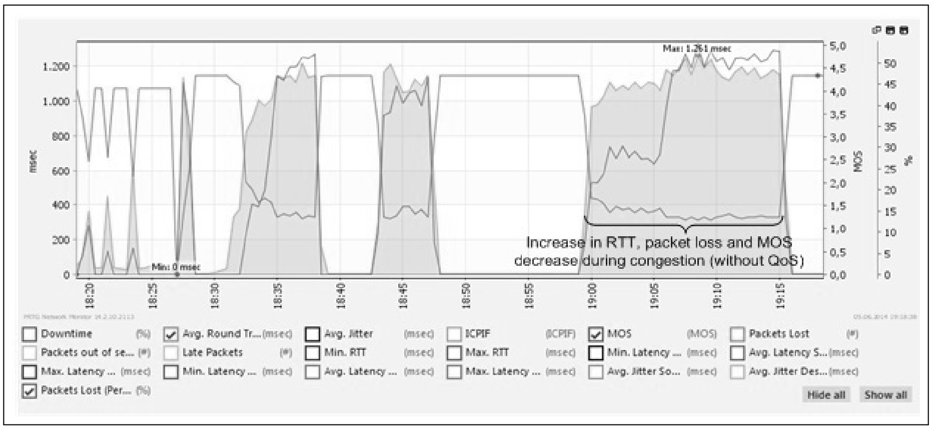
To analyze the quality of voice traffic was used an mechanism available on Cisco equipments called "Cisco IP SLA." This application may generate packets similar with VoIP packets with predetermined characteristics depending on the type of the codec used. Router 4 is used as the source and the probe and Switch 1 as the device addressed. Generated packets go through the network to the switch 1 and then return to the probe device to calculate the value for delay, jitter and number of packets lost.

Shortly after the initiation of data traffic congestion occurs in the network and the quality of voice traffic will be affected by high latency, variations in delay and packet loss. A range of objective assessments can be achieved by using a VoIP traffic generator (router 4) whose statistics are taken, analyzed and displayed in charts by a program called PRTG Network Monitor. This application identifies network devices by IP and assignes them "sensors" which return various statistics such as CPU load, system temperature, check the ping response time etc. Such a sensor that receives and interprets VoIP statistics is called Cisco IP SLA VoIP and can be associated with a router / switch Cisco that operate as transmitter of IP SLA test packets (probe). So, before this program can collect statistics from a VoIP device it must be configured to generate test traffic.

IP SLA VoIP UDP jitter mechanism can generate packets similar to VoIP packets with characteristics that are preset depending on the type of codec used, e.g. G.711, G.729. After the values for delay , jitter and packet loss have been collected an approximate value for the R factor is calculated which is an objective measure of voice quality (E-model), which in turn is converted into an equivalent value called MOS (MOS-CQE that Conversational Quality Estimated).

Therefore, IP SLA generates VoIP traffic statistics: delay, jitter, packet loss, MOS, and these statistics are retrieved and displayed in intuitive charts by the dedicated sensor in PRTG. Thus, the degradation of voice quality that comes with the installation of congestion can be confirmed by objective means (up to 1.2 seconds delay, packet loss up to 50% and jitter up to 20 ms.), shown in **Fig. 5** and **Fig. 6**:

Router 1 and router 2 are connected by a link that is limited in terms of available bandwidth. When the traffic rate tends to exceed that available bandwidth, every packet will be dropped regardless of the class to which it belongs. To avoid this, it is applied a policy to all the traffic



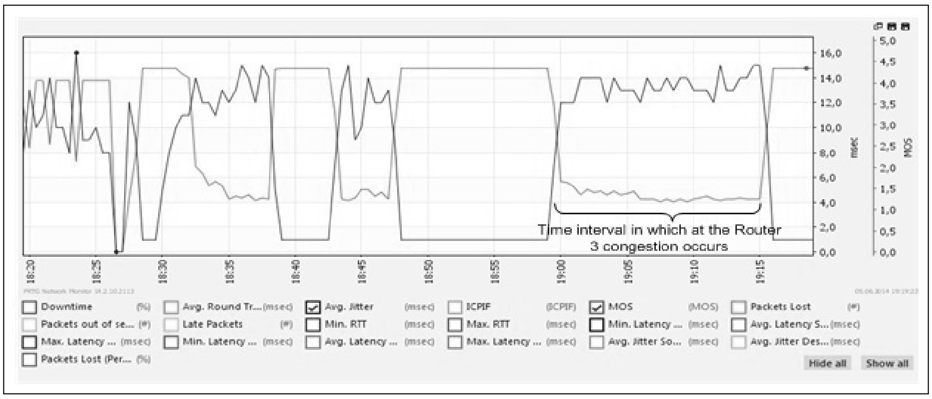
Increase in RTT, packet loss and MOS decrease during congestion (without QoS)

□ Downtime (%) ☑ Avg. Round Tr... (msec) □ Avg. Jitter (msec) □ ICPIF (ICPIF) ☑ MOS (MOS) □ Packets Lost (#)
□ Packets out of se... (#) □ Late Packets (#) □ Min. RTT (msec) □ Max. RTT (msec) □ Min. Latency ... (msec) □ Avg. Latency S... (msec)
□ Max. Latency ... (msec) □ Min. Latency ... (msec) □ Avg. Latency ... (msec) □ Max. Latency ... (msec) □ Avg. Jitter So... (msec) □ Avg. Jitter Des... (msec)
☑ Packets Lost (Per... (%)

Hide all Show all

Figure 5: **MOS**, **RTT** (average values) and **packet loss** statistics for a congested network without QoS measures (for VoIP SLA traffic)

leaving the interface (higher level policy), so the transfer rate is less than the maximum available bandwidth. In this way, it will move the congestion from the router 3 on the output interfaces of the border routers (router1 and router2) where the quality requirements of traffic can be managed (in a lower-level policy). In this application it has been studied Low Latency Queuing algorithm performance.



Time interval in which at the Router 3 congestion occurs

□ Downtime (%) □ Avg. Round Tr... (msec) ☑ Avg. Jitter (msec) □ ICPIF (ICPIF) ☑ MOS (MOS) □ Packets Lost (#)
□ Packets out of se... (#) □ Late Packets (#) □ Min. RTT (msec) □ Max. RTT (msec) □ Min. Latency ... (msec) □ Avg. Latency S... (msec)
□ Max. Latency ... (msec) □ Min. Latency ... (msec) □ Avg. Latency ... (msec) □ Max. Latency ... (msec) □ Avg. Jitter So... (msec) □ Avg. Jitter Des... (msec)
□ Packets Lost (Per... (%)

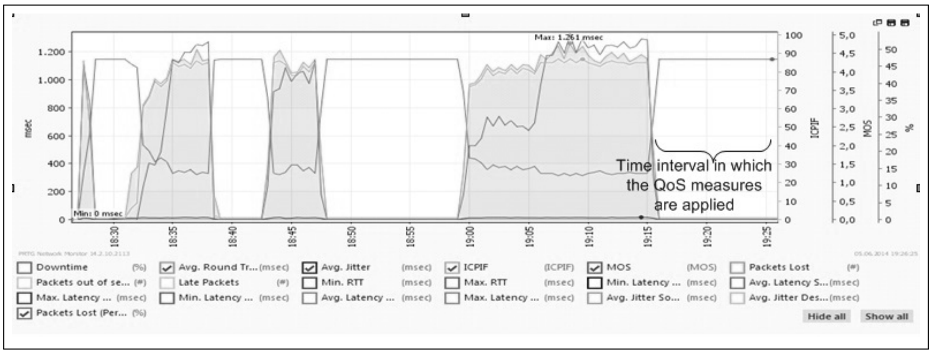
Hide all Show all

Figure 6: Highlighting the increse of **jitter** values during congestion

So there will be an aggregate traffic consisting mainly of VoIP traffic, FTP traffic and management traffic (ARP, DHCP) that will have a maximum output rate of 480 kbps (limitation imposed by the higher-level policy). From this rate a part will be allocated for VoIP traffic(with priority, 190 kbps), another part will be guaranteed for FTP traffic (150 kbps) and the remainder will be allocated to other types of traffic. This last category will share the remaining bandwidth with a WFQ policy type. LLQ algorithm must be carefully implemented because a poor setting may have unintended consequences (packet loss) on delay sensitive traffic (VoIP class in this case) due to the "policing" rule applying to it. If LLQ is implemented correctly, the traffic that is found in the priority buffer will never exceed the configured rate.

When hierarchical QoS policy is applied to the output interface of the two border routers it will be seen an immediate increase in MOS values and also the call quality (subjective) is much better in the absence of packet losses and delays as it can be seen in **Fig. 7**.

In **Fig. 7** it may be noted that packet losses have decreased from high levels of $40 - 50\%$ to 0 and the delays have also decreased to values of a few ms.



☐ Downtime (%) ☑ Avg. Round Tr... (msec) ☑ Avg. Jitter (msec) ☑ ICPIF (ICPIF) ☑ MOS (MOS) ☐ Packets Lost (#)
☐ Packets out of se... (#) ☐ Late Packets (#) ☐ Min. RTT (msec) ☐ Max. RTT (msec) ☐ Min. Latency ... (msec) ☐ Avg. Latency S... (msec)
☐ Max. Latency ... (msec) ☐ Min. Latency ... (msec) ☐ Avg. Latency ... (msec) ☐ Max. Latency ... (msec) ☐ Avg. Jitter So... (msec) ☐ Avg. Jitter Des... (msec)
☑ Packets Lost (Per... (%)

Figure 7: **MOS**, **ICPIF**, **jitter**, **RTT** statistics (average) and **packet loss** for VoIP SLA traffic with QoS measures applied

In **Fig. 8** it can be seen a comparison between the results obtained for delay and jitter when the mechanism used is CBWFQ and LLQ mechanism respectively.

Although apparently the differences between the two are small and don't affect speech quality in the lab scenario, in a real situation where there can be multiple nodes and multiple paths between source and destination the values for delay and jitter in the case of CBWFQ can grow significantly and can sum up and degrade the quality of voice traffic. Hence the importance of using a high priority buffer for voice traffic.

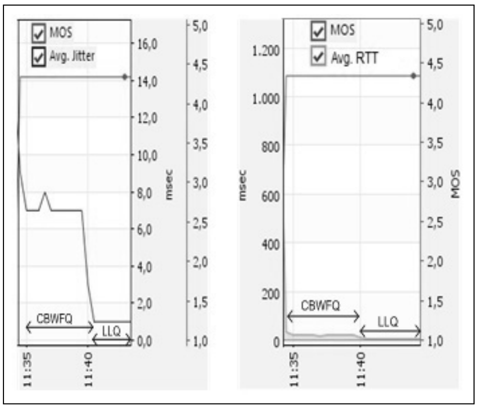


Figure 8: Highlighting the differences for **jitter** and **RTT** between CBWFQ and LLQ

4 Conclusions

Using appropriate QoS mechanisms it can be achieved the desired quality for VoIP traffic while maintaining the operation at lower rates for other types of traffic. Thus, by applying the appropriate QoS measures we have achieved a considerable improvement of MOS index from values of about 1.5 to ideal levels (about 4) which indicates a very good quality for VoIP traffic. The increase in MOS is due to improvements in the values of delays (from hundreds, thousands of milliseconds to a few milliseconds); lost packets (at levels of about $40 - 50\%$ to 0%) and jitter (from 15msec to 1 msec).

LLQ is a highly effective mechanism to meet QoS requirements because it allows control of service quality based on traffic classes that can be defined with precision.

Also, analyzing the mechanism behind CBWFQ (LLQ) it can be concluded that none of the classes is not restricted to the configuration set up if there is enough bandwidth available.

We also demonstrated the necessity of using a high priority buffer for VoIP traffic through a comparative analysis of the results obtained in the two cases LLQ and CBWFQ which showed that values for delay and jitter can be minimized by using Low Latency Queuing.

Bibliography

[1] P. Sharma (2014); Challenges of Quality of Services in Mobile Ad Hoc Networks, *International Journal of Advance Research in Computer Science and Management Studies*, 2(3): 342-347.

[2] R.D. Albu, I. Dzitac, F. Popentiu-Vladicescu, I.M. Naghiu (2010); Input Projection Algorithms Influence in Prediction and Optimization of QoS Accuracy, *International Journal of Computers Communications & Control*, 9(2): 132-139.

[3] V. Fineberg (2002); A practical architecture for implementing end-to-end QoS in an IP network, *IEEE Communications Magazine*, 40(1): 122-130.

[4] W. C. Hardy (2003); *VoIP Service Quality: Measuring and Evaluating Packet-Switched Voice*, McGraw-Hill, 2003.

[5] T. Wallingford (2005); *Switching to VoIP*, O'Reilly, 2005.

[6] O. Hersent, J.P. Petit, D. Gurle (2005); *Beyond VoIP Protocols: Understanding Voice Technology and Networking Techniques for IP Telephony*, John Wiley and Sons, 2005.

[7] J. Davidson, J. Peters (2000); *Voice over IP Fundamentals*, Cisco Press, 2000.

[8] Cisco Systems, Inc.(2001); *Cisco IP Telephony QoS Design Guide*, www.cisco.com.

[9] M. A. Dye, R. McDonald, A.W. Rufi (2008); *Network Fundamentals CCNA Exploration Companion Guide*, 2008 Cisco Press.

[10] Cisco Systems, Inc (2014); *QoS: Congestion Management Configuration Guide, Cisco IOS XE Release 3S*, www.cisco.com.

[11] T. Szigeti, R. Barton, C. Hattingh, K. Briley Jr. (2014); *End-to-End QoS Network Design: Quality of Service for Rich-Media and Cloud Networks*, Second Edition, Cisco Press, 2014.

[12] http://mynetworkingwiki.com/index.php/Weighted_Fair_Queuing

[13] http://myway2ccie.blogspot.ro/2009/04/qos-weighted-fair-queuing-and-class.html

[14] http://www.perihel.at/2/rno/03-QoS-Queuing-Methods.pdf

[15] http://blog.ine.com/2008/08/17/insights-on-cbwfq/?s=wfq

[16] http://josephmlod.files.wordpress.com/2010/12/llq-pq.jpg