QoS in IP Networks

TOBENNA UZOMA, IHEGBOROW
A00267346

Aim: The aim of this experiment is to evaluate the result of different queuing mechanisms on packet delivery and delays for three different services – voice, video and ftp, running on a small IP network with the use of OPNET Modeler 14.5.

Objectives:

- Set up a simulation in FIFO mechanism on three traffic types – video, voice & ftp, and compare the results.
- Using the same simulation, set up WFQ and PQ on the three traffic types and compare the results.

Introduction/Backgrounds:

The impact of surplus Quality of Service (QoS) along IP networks is leading to be an ever-growing area in the current enterprise IT infrastructure. It has become a necessary factor in maintaining the rising Internet of Things (IoT), not just supporting voice and video streaming along a network. (Froehlich, 2016)

Many applications running along a network are sensitive to delay. These kind of applications use UDP protocols most of the time compared to the TCP protocol. The main contrast between TCP and UDP relating to time sensitivity is that TCP resends packets that get lost in transit while UDP does not bother. For matters concerning file transfer from one end system to the other, TCP is commonly used because it assures best effort delivery of packets. In the sense that if packets get lost or arrive out of order, it resends and reorders these packets to reproduce the file on the destination host. However, in UDP resending packets are useless. Packets misplaced in applications such as internet telephone calls arrive in an ordered stream, hence retransmitting is not important. This makes applications supporting UDP protocol a real issue. In the internet call example stated earlier, losing even a tiny fraction of packets will lead to the voice service being scrambled and inaudible. Also, these packets are delicate to what is called Jitters. Jitter is known to be the disparity in the delay of received packets (Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms), 2006).

Packet loss, delays or jitters can be avoided in a network if that network has enough bandwidths and no traffic that surges beyond what it can carry. However, in enterprise networks, situations arise where links get over congested and the routers and switches begin to lose or drop packets because they are arriving or leaving quicker than it can be processed. In this situation, streaming applications will begin to buffer.

A provision for packet classification and marking is need to support QoS. There is CoS (Class of Service) bits at layer 2 for categorizing the traffic and ToS (Type of Service) bits at the network layer.

First-In-First-Out (FIFO) is the most rudimental queue scheduling mechanism. In this queuing, all packets are handled equally by putting them in a single tunnel and then servicing them in the same order with which they were placed into the queue. It is also commonly termed First Come First Served (FCFS) queuing (V & Thorenoor, 2009).

Priority Queuing (PQ) as the name implies, supports different numbers of queues ranging from high to low. Queues are placed in an inflexible order of priority where high queues are attended to first, then the next-lower priority all the way down to the very lowest priority. In a situation where a high priority packet enters a lower priority queue, this packet is serviced promptly. This is mechanism is good for important traffic but often times leads to queue starvation.

Weighted Fair Queuing (WFQ) is a network scheduling algorithm. It assigns a percentage of output bandwidth which is equal to the relative weight of each traffic class during periods of congestion (V & Thorenoor, 2009). It is an extension of Fair Queuing (FQ). (Wikipedia, 2018)

The simulation in this experiment is built using three traffic types which will be discussed in the following paragraphs below.

The File Transfer Protocol (FTP) is a standard network protocol used for sharing files a host/server and a client on a network (wikipedia, 2018). The File Transfer Protocol is developed on the client-server model architecture using different control and data connections linking the client and server. FTP users are allowed to validate themselves using a clear-text sign-in protocol usually in the appearance of a username and password and can connect incognito as long as the server is configured to allow it. FTP uses SSL/TLS(FTPS) or SSH File Transfer Protocol for secure transmissions that encrypts content and shields the username and password.

VoIP (Voice Over IP) is the conveyance of voice and multimedia content over the internet network. VoIP historically referred to using IP to connect private branch exchanges (PBXs), but the term is now used interchangeably with IP telephony (Rouse, n.d.). In VoIP, codecs are used to encapsulate audios into data packets, transmit these packets along the network and eventually decapsulated into audio at the destination host at the end of the connection. VoIP minimizes network infrastructural costs by eradicating the use of circuit-switched networks for voice, allowing providers deliver voice services over their broadband and private networks. VoIP also rides on the flexibility of IP based networks by permitting fast failover following outages and needless communications among endpoints and networks.

Method:

In this experiment, I used OPNET to create a small IP network and used QoS policies to examine the performance of the network in terms of packet delivery for FTP, voice and video services. The experiment was carried out in two parts;

- PART A
    1. Created a FIFO scenario on a new project.
    2. Configured the data, video and voice application to cause traffic over the IP network using FTP as the data application.
    3. Configured the profiles for the three services to be run on the network.
    4. Created the network using the object palette and deployed the applications on the clients.
    5. Configured the links connecting the routers and saved the network.

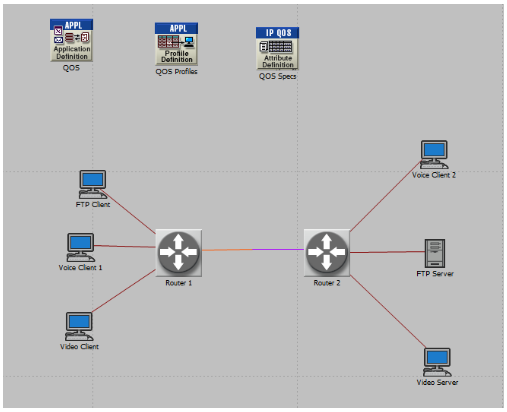
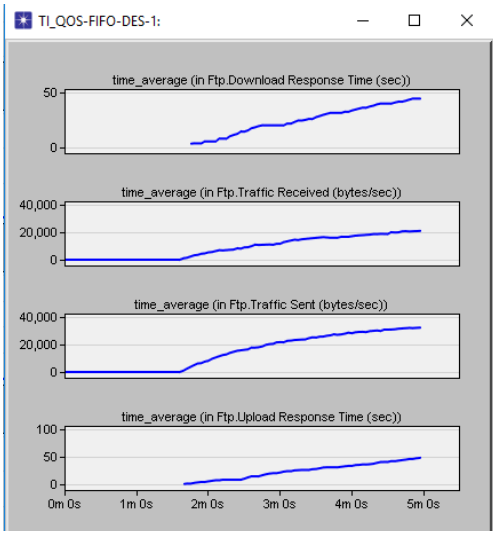


Illustration 1: Layout of the FIFO scenario

6. Set the global statistics for the three service levels and set the simulation duration to 300 seconds.
7. Created three scenarios – FTP only, Video only & Voice only.
8. Ran the four scenarios and compared the results of each application and the FIFO scenario.

- <u>PART B</u>
  1. Duplicated the FIFO scenario, created a PQ scenario and changed the protocol on the link connecting the routers.
  2. Saved the project.
  3. Duplicated the FIFO scenario again, created a WFQ scenario and changed the protocol on the link connecting the routers.
  4. Saved the project.
  5. Ran the simulation for the two new scenarios and compared the results of the performances by the three applications.

Simulation Results:



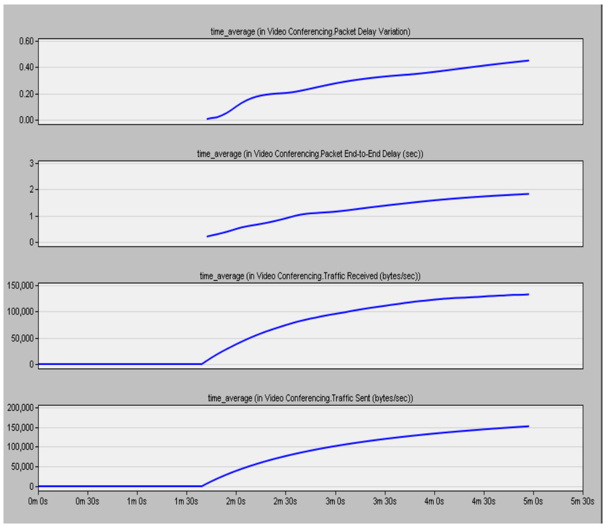
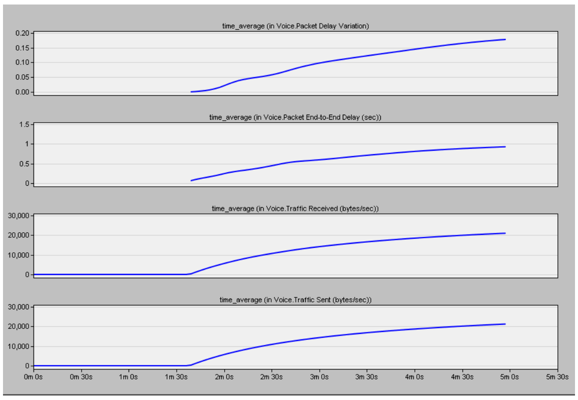


Fig 1: FIFO vs all the Ftp global statistics
Fig 2: FIFO vs all the Video global statistics



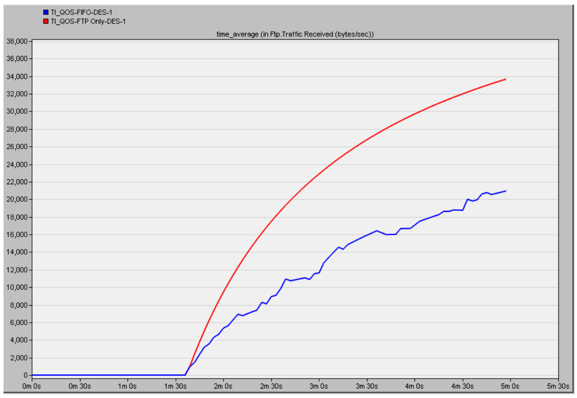
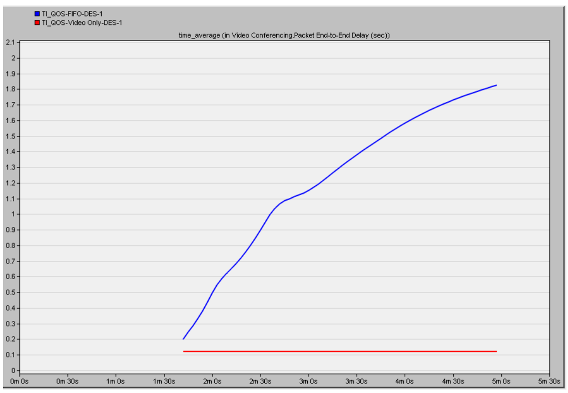


Fig 3: FIFO vs all the Voice global statistics
Fig 4: FIFO vs FTP Only scenario



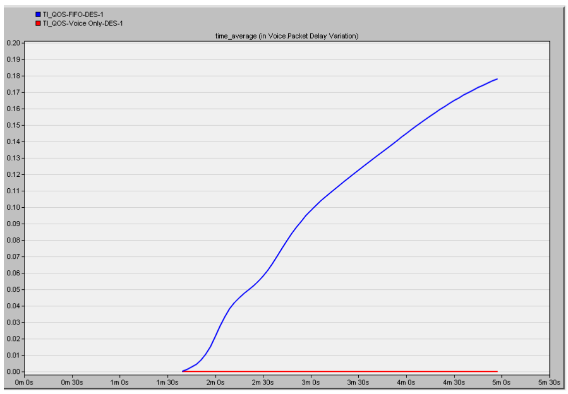
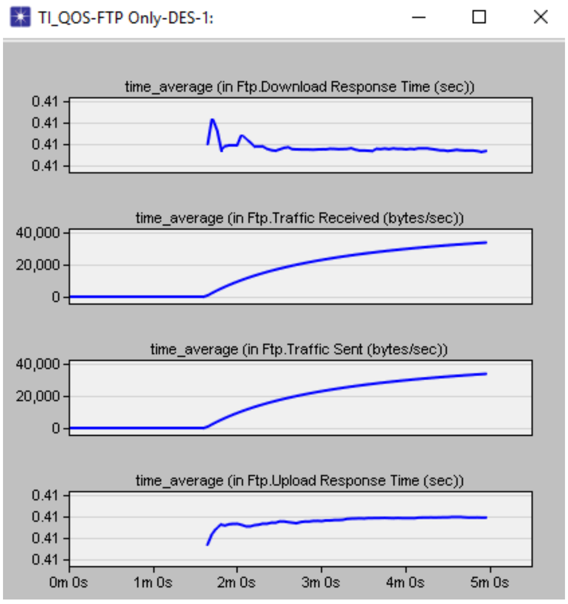


Fig 5: FIFO vs Video Only scenario
Fig 6: FIFO vs Voice Only scenario



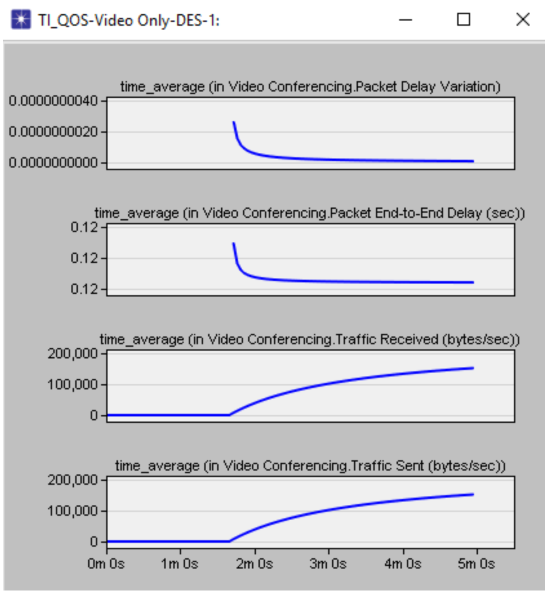
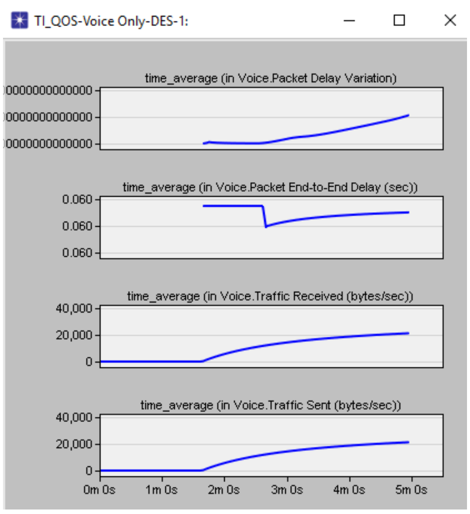


Fig 7: FTP Only scenario
Fig 8: Video Only scenario



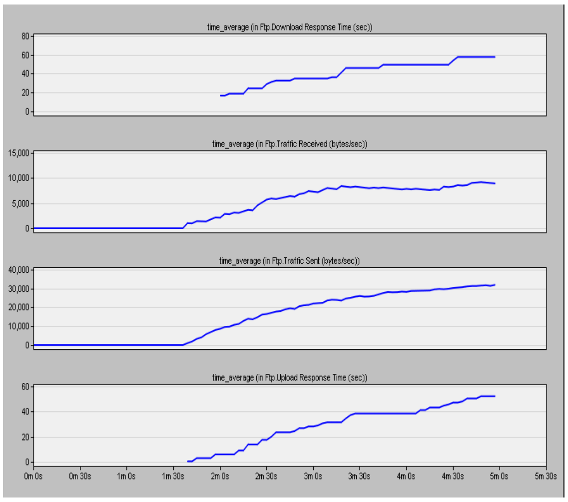
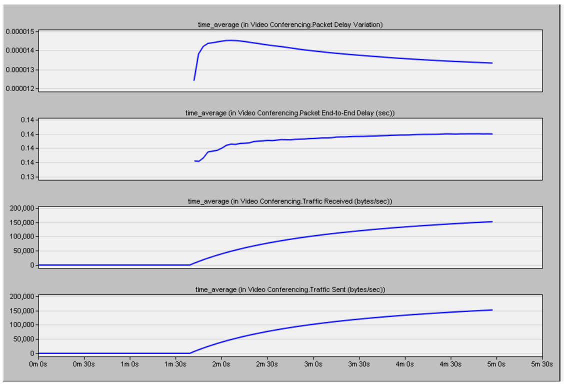


Fig 9: Voice Only scenario
Fig 10: PQ vs all the Ftp global statistics



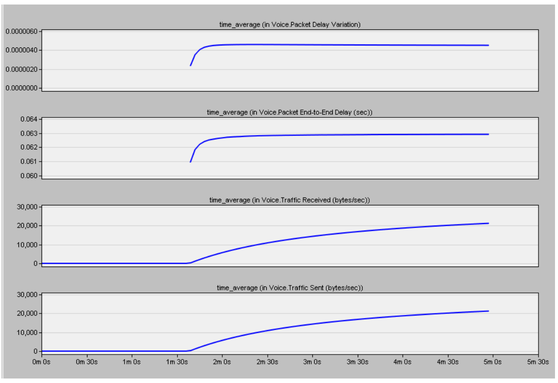
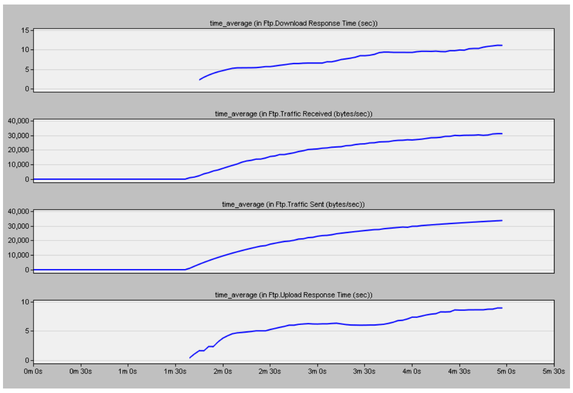


Fig 11: PQ vs all the Video global statistics
Fig 12: PQ vs all the Voice global statistics



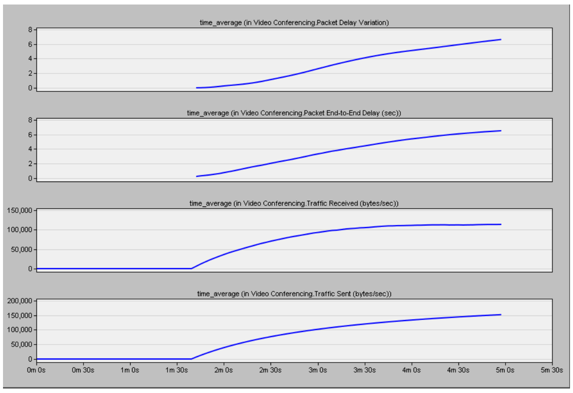
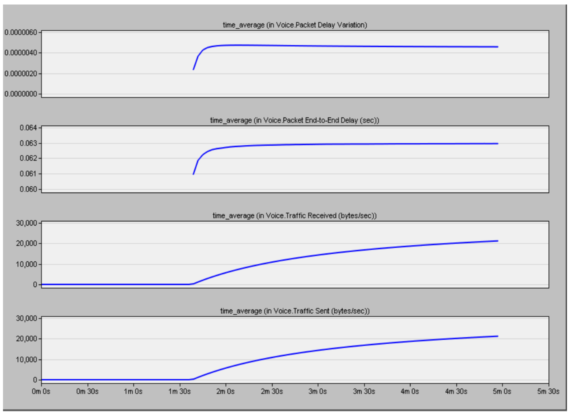


Fig 13: WFQ vs all the Ftp global statistics
Fig 14: WFQ vs all the Video global statistics



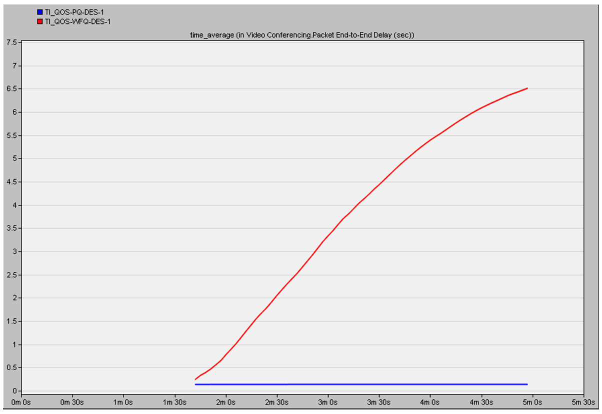
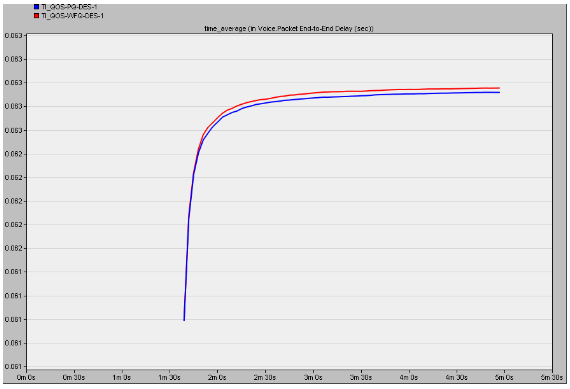


Fig 15: WFQ vs all the Voice global statistics
Fig 16: PQ vs WFQ on Video conferencing (packet end-to-end delay)



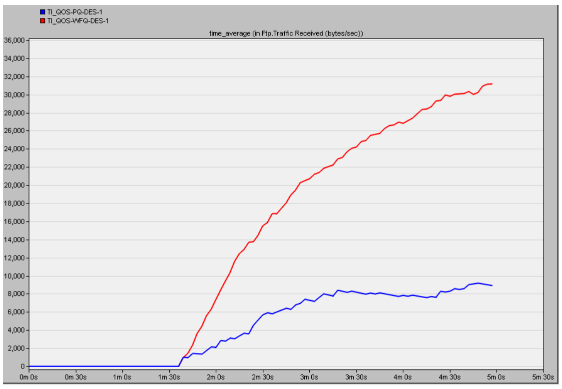
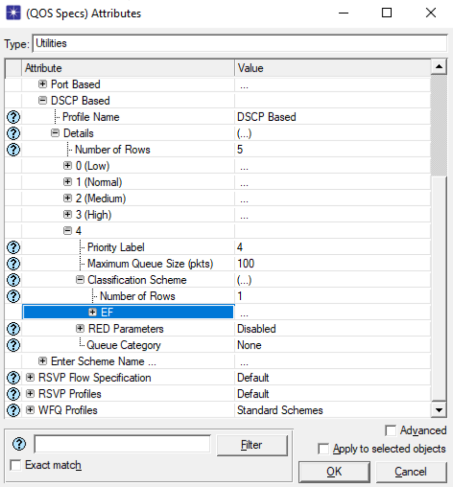


Fig 17: PQ vs WFQ on Voice (packet end-to-end delay)
Fig 18: PQ vs WFQ on FTP (traffic received)

Table 1: PQ for Voice (level 4)
Table 2: PQ for FTP (low)



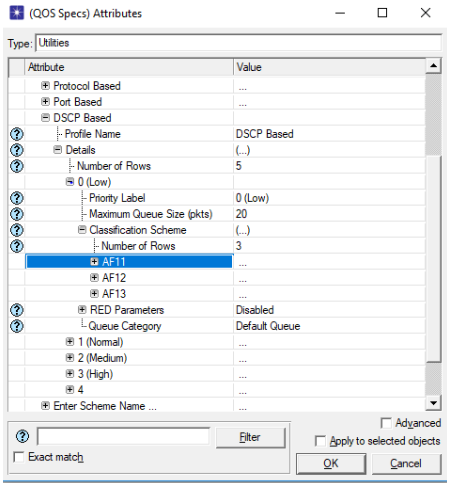
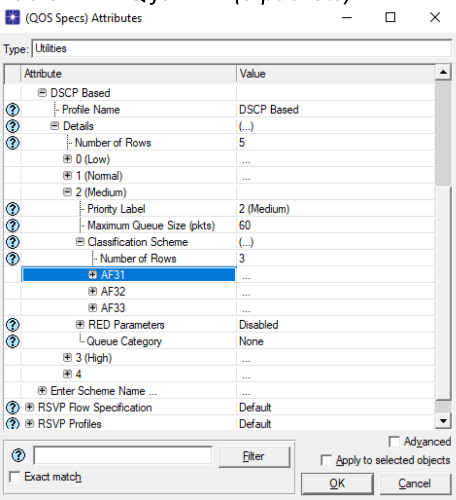


Table 3: PQ for Video (medium)
Table 4: WFQ for FTP (5 packets)



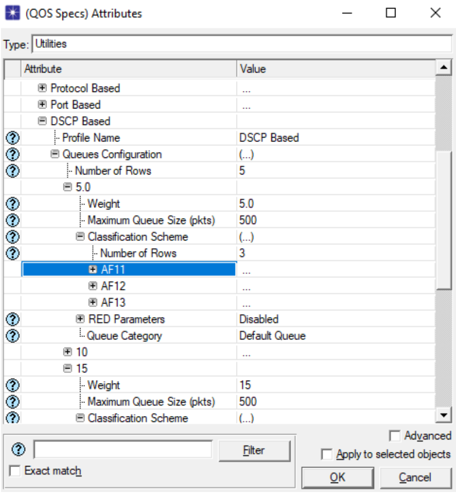
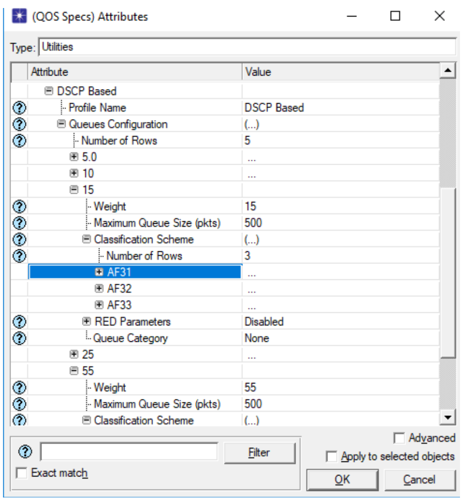
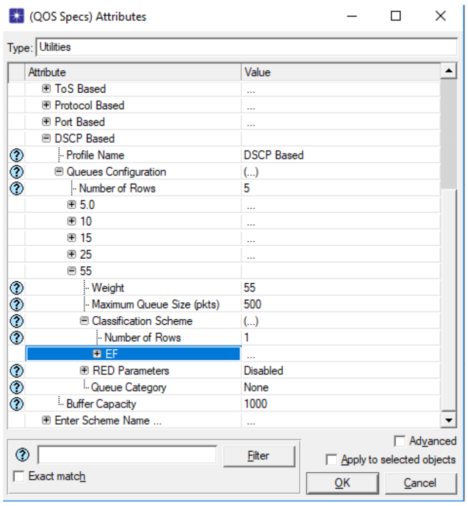


Table 5: WFQ for Video (15 packets)
Table 6: WFQ for Voice (55 packets)





Discussion:

The above graphs and tables are the simulation results from running the experiment using the OPNET Modeler. These will be discussed in details below;

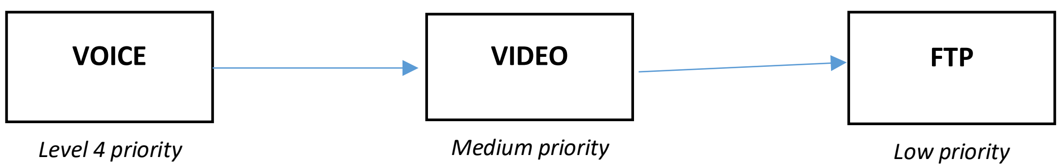
Fig 1, 2 & 3 are the simulation results for the FIFO scenario on the three different network services. It shows how these services are being sent and how long it takes them in the tunnel. Using the basic understanding of FIFO aka First Come First Serve, *fig 3* for example shows it takes about 5minutes to send about 20,000bytes of voice traffic received through the network which is also almost equal to the same time it takes the same amount of packets for FTP traffic received. In essence, in a FIFO network where ftp, voice and video are being sent, when about 20,000 packets of voice and ftp traffic are received, it'll take about 5 minutes for them to be sent to the destination host.

However, *Fig 7, 8 & 9* are the simulation results when the network is sending only any of the service types. There is no interruption or sharing of the tunnel, hence, all the packets are being sent smoothly.

Fig 4, 5 & 6 on the other hand are the simulation results comparing when the network sends only any of the service types alone with when operating in FIFO scenario. *Fig 4* for example compares the network when it is sending only FTP with when it's working in FIFO. As we can see, when it's on FIFO the network throttles or jitters. This is because the network is sending all other service types, hence the decline in the QoS. Looking at when it's on FTP only, you can observe the smooth packet transmission sending 34,000bytes of packets in 5minutes which is an increase from the previous 20,000bytes shown in *Fig1* when the network is on FIFO. *Fig 5* and *Fig 6* equally show difference when sending only video and voice respectively. In other words, QoS is improved when each of these network services (ftp, voice & video) are being sent individually on the network.

The second part of the experiment shows when the same FIFO scenario is duplicated and then changed to Priority Queuing and Weighted Fair Queuing scenarios.

Fig 10, 11 & 12 shows when the network is set on PQ for ftp, video and voice services. If you compare these three graphs, you will observe the slight differences in transmission of packets for each of them. The QoS for the packets transmitting for voice in *Fig 12* are very smooth, followed by the video conferencing having little throttling in *Fig 11*, while *Fig 10* on the other hand shows ftp services throttling heavily. This is because in the priority queuing, voice services are awarded the highest priority (*Table 1*), followed by video conferencing (*Table 3*) with priority set to medium and finally ftp having priority set to low (*table 2)*. The structure of the priority for the three services appear as follows;



In essence, whenever ftp, video and voice packets are being sent in a network with PQ, all the voice packets will be sent first before anything else, closely followed by video conferencing packets and finally ftp. Hence, all FTP services will be halted until every packets for voice and video have been transmitted. In a real life scenario using this PQ mechanism, no single FTP might be transmitted at all.

The last scenario considered is the WFQ where *Fig 13, 14 & 15* were examined closely. This time around ftp, voice and video packet have been compared showing similar smooth transmissions of packets in the network. This is because WFQ allows the sharing of services by splitting these services in weights, permitting all services to have a fair chance of getting transmitted unlike PQ where services are prioritized and adhered to strictly. This has been clearly shown in *tables 4, 5 and 6*. In this, FTP has been awarded 5 packets, video conferencing awarded 15 packets and voice awarded 55 packets by WFQ. This allows a "*round robin*" feel of service delivery where all three services are being sent in series or sequence.

Fig 16, 17 & 18 lastly compares PQ and WFQ against video conferencing, voice and ftp services respectively. Looking at *Fig 16* it takes about 6.5secs to send video conferencing packets along the tunnel using WFQ, while it only takes it about 0.2secs using PQ. This is because, as explained earlier, all the packets for priority queuing get sent all at once, hence it's smooth transmission in the network. But using WFQ, it takes a longer time because the network can only send 15packets at a time. The same observation can be seen in *Fig 17* where it takes almost the same time for end-to-end delay of voice packets to be sent along the network. Using PQ, all of the voice packets are sent all at once, while in WFQ only 55 packets can be sent at a time. Comparing 55 packets to the 15 and 5 packets for video and ftp respectively, it is way higher. As a result, it is sent smoothly along the tunnel though slightly higher than PQ because it is still sharing the tunnel with the remaining services. Finally, *Fig 18* compares when PQ and WFQ are used to send traffic received in FTP. Both queues appear to throttle because the network is operating on both queuing systems. Only 5packets can be sent at a time using WFQ while in PQ it can only be sent if and only when there are no voice or video packets. WFQ allows services to be shared in series (round robin), hence the throttling appearance of the FTP in PQ.

<u>Conclusion</u>:

I set up, ran and compared the results of the simulation for the FIFO scenario on three network services – FTP, video and voice. I set another scenario for PQ and WFQ, then compared results using the same network services.

The QoS provided when the network runs with Weighted Fair Queuing is much better than First-In-First-Out and Priority Queuing. This is because it allows a fair opportunity for each of the services to be sent across the network at the same time leaving no room starvation and packet loss.

Bibliography

Froehlich, A. (2016, August 15). *Network Computing*. Retrieved from The Basics of QoS: https://www.networkcomputing.com/networking/basics-qos/402199215

Rouse, M. (n.d.). *VoIP (voice over IP)*. Retrieved from TechTarget - Search Unified Communications: https://searchunifiedcommunications.techtarget.com/definition/VoIP

Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms). (2006, February 2). Retrieved from Cisco: https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/18902-jitter-packet-voice.html

V, A. K., & Thorenoor, S. G. (2009). *Analysis of IP Network for different Quality of Service.* Singapore. Retrieved from http://www.ipcsit.com/vol1/15-A1070.pdf

wikipedia. (2018, November 8). *File Transfer Protocol*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/File_Transfer_Protocol

Wikipedia. (2018, September 17). *Weighted Fair Queuing*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Weighted_fair_queueing