Queuing Disciplines:
Analysis of Packet Transmission and Dropping in Routers

Tabish, CSE-09/16

Department of Computer Science Engineering, National Institute of Technology Srinagar

As part of the resource allocation mechanisms, routers must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. Different algorithms are used to determine which packets are loaded onto router memory and which ones are discarded. Queuing within routers also determines the delay before which packets may be transmitted. Examples of the common queuing disciplines are first-in, first-out (FIFO) queuing, priority queuing (PQ), and weighted fair queuing (WFQ). The queuing process present in routers gives rise to the inherent Queuing Delay over a network. Further, the maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is. The router queue of packets waiting to be sent also introduces a potential cause of packet loss. Since the router has a finite amount of buffer memory to hold the queue, a router which receives packets at too high a rate may experience a full queue. In this case, the router has no other option than to simply discard excess packets.

Keywords: Queuing Delay | Queuing Disciplines | Router Buffer | First in First Out | Riverbed Modeller | First-in-First-Out | Priority Queues | Quality of Service (QoS)

Introduction.

A **Queue** is used to store traffic in a router till the time it can be serialized. Both switch and router interfaces have ingress (inbound) queues and egress (outbound) queues. An ingress queue stores packets until the switch or router CPU can forward the data to the appropriate interface. An egress queue stores packets until the switch or router can serialize the data onto the physical wire. This project will focus on analysing the various mechanisms of queuing and observe how the choice of the queuing discipline in the routers can affect the performance of the applications and the utilization of the network resources.

Queue Congestion.

Switch (and router) queues are susceptible to congestion by virtue of the limited buffer memory made available by them. When the rate of ingress traffic becomes larger than the amounts that can be successfully serialized on an egress interface, congestion is observed. The basic, factors that promote congestion include:

- The speed of an ingress interface is higher than the egress interface.
- The combined traffic of multiple ingress interfaces exceeds the capacity of a single egress interface
- The switch/router CPU is insufficient to handle the size of the forwarding table.

By default, if an interface's queue buffer fills to capacity, new packets will be dropped. This condition is referred to as tail drop, and operates on a first-come, first-served basis. If a standard queue fills to capacity, any new packets are indiscriminately dropped, regardless of the packet's classification or marking. QoS provides switches and routers with a mechanism to queue and service higher priority traffic before lower priority traffic. This project covers the analysis various queuing methods in detail.

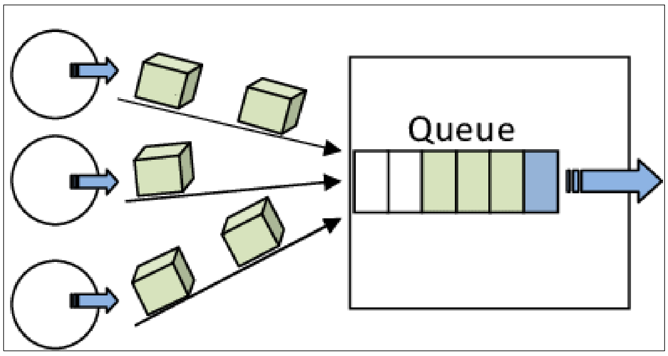

Figure 1: Multiple Packets Incoming into a Router Input Interface and being buffered

Types of Queuing Disciplines:

When receiving packets of data that cannot be accommodated in the buffer memory of a routing device, packets are dropped. This is done without regard to which flow the packet belongs to or how important the packet is. **Priority Queue** (PQ) is a simple variation of the basic FIFO queuing. The idea is to mark each packet with a priority; the mark could be carried, for example, in the IP Type of Service (**ToS**) field. The routers then implement multiple **FIFO** queues, one for each priority class. Within each priority, packets are still managed in a **FIFO** manner. This queuing discipline allows high priority packets to cut to the front of the line. The idea of the **fair queuing** (FQ) discipline is to maintain a separate queue for each flow currently being handled by the router. The router then services these queues in a **Round Robin** manner. **WFQ** allows a weight to be assigned to each flow (queue). This weight effectively controls the percentage of the link's bandwidth each flow will get. We could use ToS bits in the IP header to identify that weight.

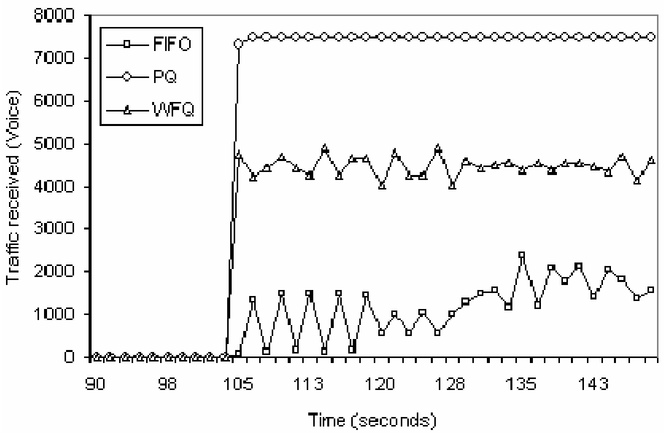

Figure 2: Estimated Voice Traffic Received with time in the various Queuing Algorithms

First-In First-Out Queuing:

The default form of queuing on nearly all interfaces is First-In First-Out (**FIFO**). This form of queuing requires no configuration, and simply processes and forwards packets in the order that they arrive. If the queue becomes saturated, new packets will be dropped (**tail drop**). This form of queuing may be insufficient for real-time applications, especially during times of congestion. FIFO will never discriminate or give preference to higher-priority packets. Thus, applications such as **VoIP** can be starved out during periods of congestion.

Hardware queues always process packets using the FIFO method of queuing. In order to provide a preferred level of service for high-priority traffic, some form of software queuing must be used. Software queuing techniques can include:

- First-In First-Out (FIFO) (default)
- Priority Queuing (PQ)
- Custom Queuing (CQ)
- Weighted Fair Queuing (WFQ)

Priority Queuing (PQ).

Traffic must be assigned to these queues, usually using access-lists. Packets from the High queue are always processed before packets from the Medium queue. Likewise, packets from the Medium queue are always processed before packets in the Normal queue, etc. Remember that traffic within a queue is processed using FIFO. As long as there are packets in the High queue, no packets from any other queues are processed. Once the High queue is empty, then packets in the Medium queue are processed but only if no new packets arrive in the High queue. This is referred to as a strict form of queuing.

The obvious advantage of PQ is that higher-priority traffic is always processed first. The nasty disadvantage to PQ is that the lower-priority queues can often receive no service at all. A constant stream of High priority traffic can starve out the lower-priority queues.

Weighted Fair Queuing (WFQ).

Weighted Fair Queuing (WFQ) dynamically creates queues based on traffic flows. Traffic flows are identified with a hash value generated from the following header fields:

- Source and Destination IP address
- Source and Destination TCP (or UDP) port
- IP Protocol number

Type of Service value (IP Precedence or **DSCP**) Traffics of the same flow are placed in the same flow queue. By default, a maximum of **256** queues can exist, though this can be increased to **4096**. If the priority (based on the ToS field) of all packets are the same, bandwidth is divided equally among all queues. This results in low-traffic flows incurring a minimal amount of delay, while high-traffic flows may experience latency.

Packets with a higher priority are scheduled before lower-priority packets arriving at the same time. This is accomplished by assigning a **sequence number** to each arriving packet, which is calculated from the last sequence number multiplied by an inverse weight (based on the ToS field). In other words, a higher ToS value results in a lower sequence number, and the higher-priority packet will be serviced first.

Setup of Project.

In this project, a network that carries three applications: FTP, Video, and VoIP, has been set up. The aim is to observe how the various queuing mechanisms affect the various network parameters in terms of network resource utilization, effectiveness and data loss. Analysis are done on the basic of software-based simulations. The results are use to justify the effects of using a particular scheme. In particular, we try to observe which schemes are better for what application.

Software Used for Simulations: *Riverbed Modeler Academic Edition*

Components and Simulation Methodology:

1. ***Application Config*** (*one*): Used to describe 3 basic applications services that will be present and used over the network. The applications are: **High Load FTP Service**, **Low Resolution Video Conferencing**, **Interactive VoIP Application**.

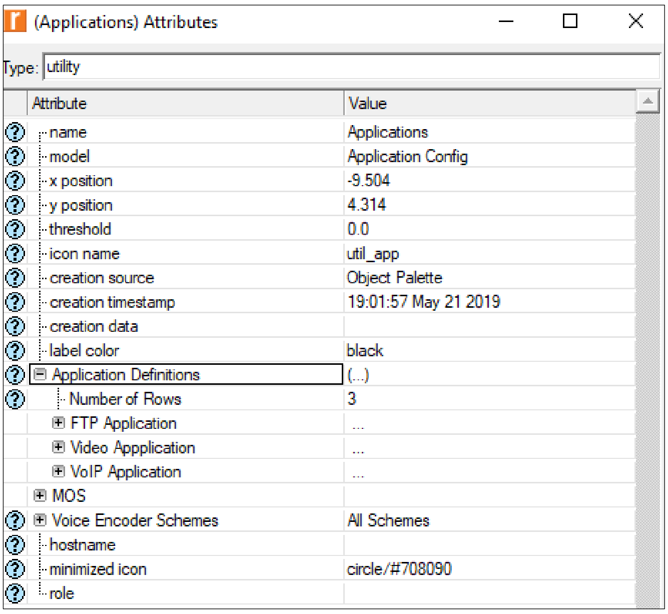

Figure 3: Application Profile Created for the Simulation. Includes 3 application (FTP, Video & VoIP)

2. ***Profile Config*** (*one*): Used to apply the different Application profiles to the nodes on the network

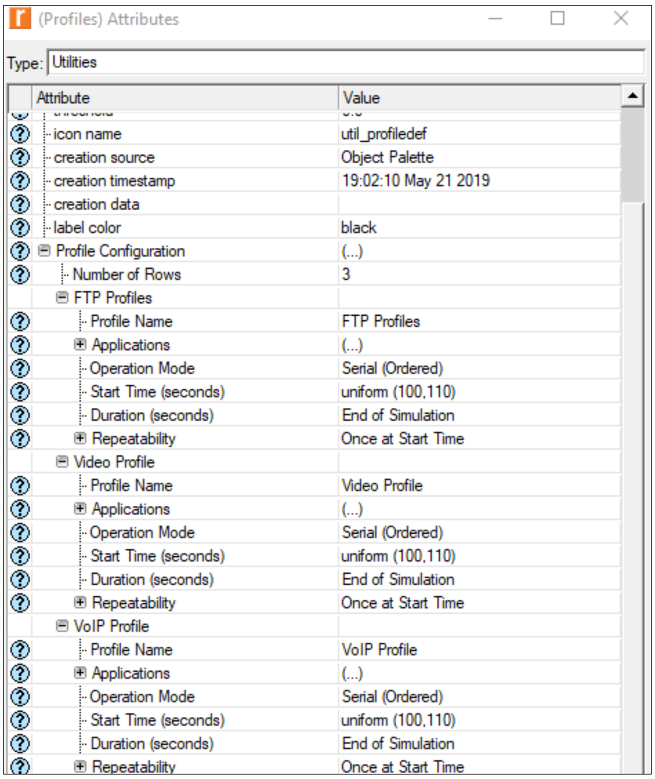

Figure 4: Profile Configuration. Separate Profiles are created for FTP, Video & VoIP Applications

3. ***QoS Attribute Config*** (*one*): The **QoS** Attribute Config node defines attribute configuration details for protocols supported at the IP layer. These specifications can be referenced by the individual nodes using symbolic names. It defines different queuing profiles such as **FIFO**, **WFQ**, **priority queuing**, **custom queuing**, **MWRR**, **MDRR**, and **DWRR**.
4. ***ethernet_wkstn*** (*five*): Designed to emulate the following: FTP Client, Video Client, VoIP Clients and Video Server
5. ***ethernet_server*** (*one*): Used to emulate functionality of the FTP Server on the network
6. ***ethernet4_slip8_gtwy*** (*two*): Allow interconnection of the nodes in the network
7. ***10Base_T***: All connections are assumed to be wired ethernet connections using the 10-Base-T ethernet wired connections.

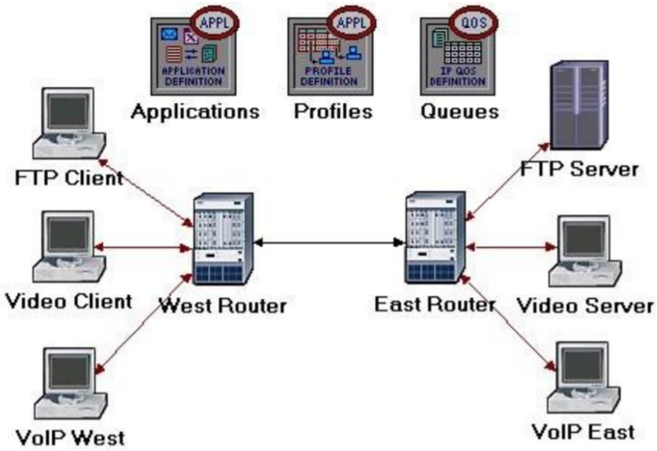

Figure 5: Network Setup for Simulation. Involves Clients and Servers utilizing the designed Applications.

Scenarios and Collected Statistics

The analysis involves 3 distinct scenarios that employ the **FIFO**, **PQ** and **WFQ** implementations on the same network. Analysis is performed on the performances of the network in these different scenarios.

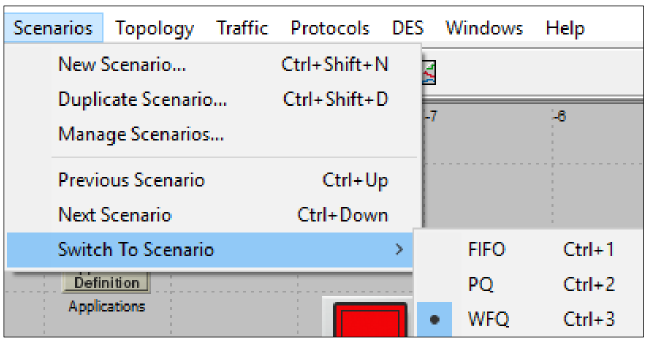

Figure 6: Three separate scenarios have been created for comparitive analysis of Queuing Disciplines

This analysis is based on the collection and subsequent comparison of the following metrics:

- **IP Traffic Dropped** (*packets/sec*). The number of IP datagrams dropped by all nodes in the network across all IP interfaces. The reasons for dropping an IP datagram can be any one of the following: Insufficient space in the queue, Maximum number of hops exceeded by an IP datagram, on non-routing nodes, a local router interface was not found to be used as the next hop, on routing nodes, the route table lookup failed to yield a route to the destination.
- **Traffic Received in the Voice Application** (*bytes/sec*)
- **Packet Delay Variation**, **Packet End-to-End delay** (*sec*) in the Voice app.

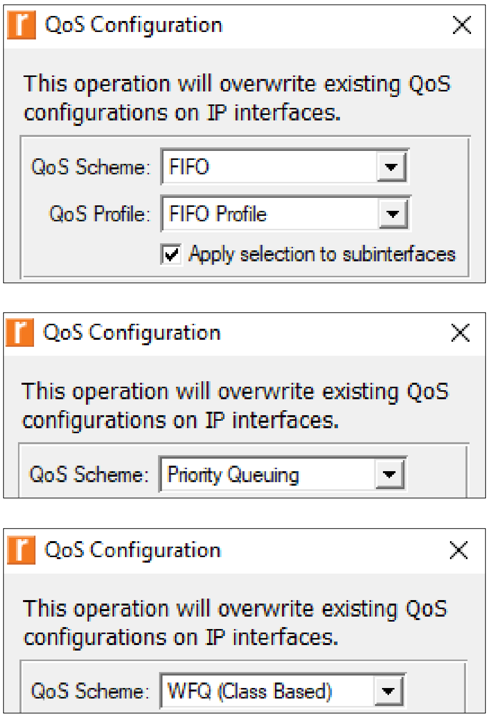

Figure 7: QoS Configuration menu allows selection of Queuing Discipline being utilized on the network

Simulation Duration: 150 seconds

Simulation Results & Analysis.

The system has been designed in a way such that the overall incoming traffic rate is greater than the rate at which the router empties the inbuilt buffer. This is done to analyse the behaviour of the different queuing mechanisms when buffer capacity is reached. Thus, the combined traffic originating from all the application (VoIP, FTP and Voice), beyond a certain point, cannot be accommodated and thus packets must be dropped. The trends of when packet drops start and how the progress has been evaluated in the project.

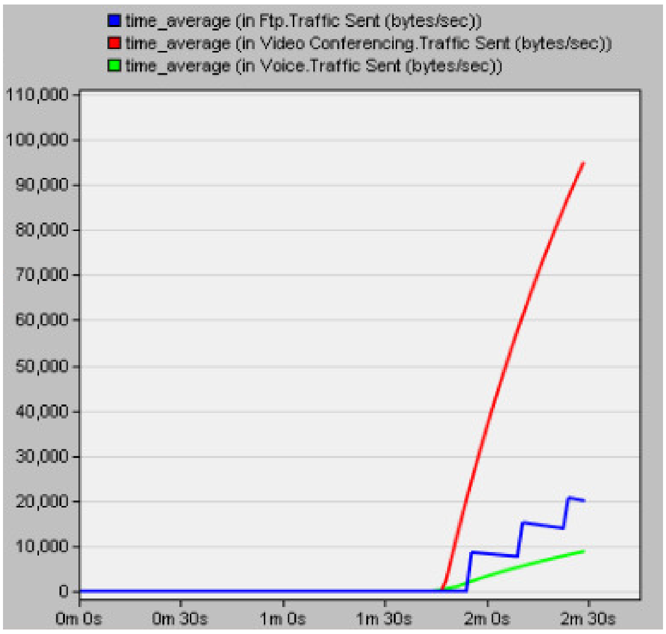

Figure 8: Traffic sent by the different applications

Analysis of the data being sent by the applications indicates that the 3 applications send out data on the network in the following order (descending):

I. Video Conferencing (Highest Load)
II. FTP (Medium Load)
III. VoIP (Smallest Load)

Since Video App has the highest probability of causing packet drops and it is based on **UDP**, which is an unreliable service, packets from Video Application will be assigned a higher priority level. FTP will have a lower priority as it is based on **TCP**, a protocol that allows for packet recovery in case of drop.

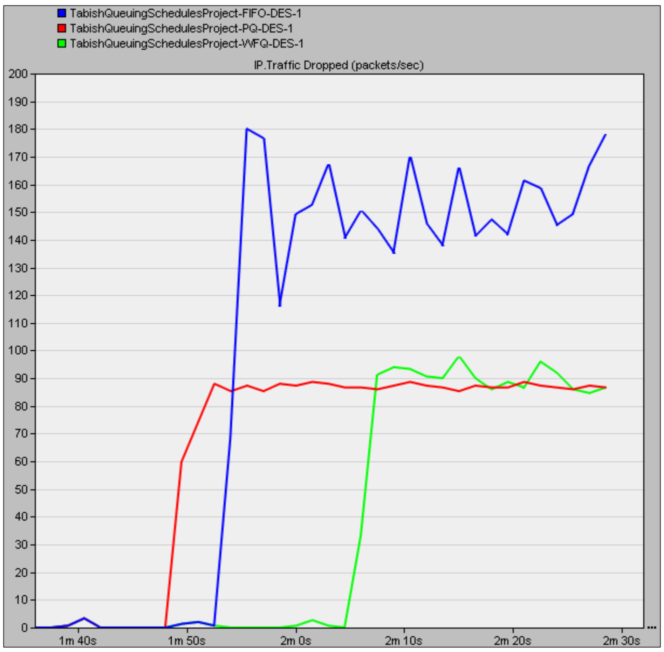

Figure 8: Dropped IP Traffic measurements for FIFO, PQ and WFQ Schemes (As Is)

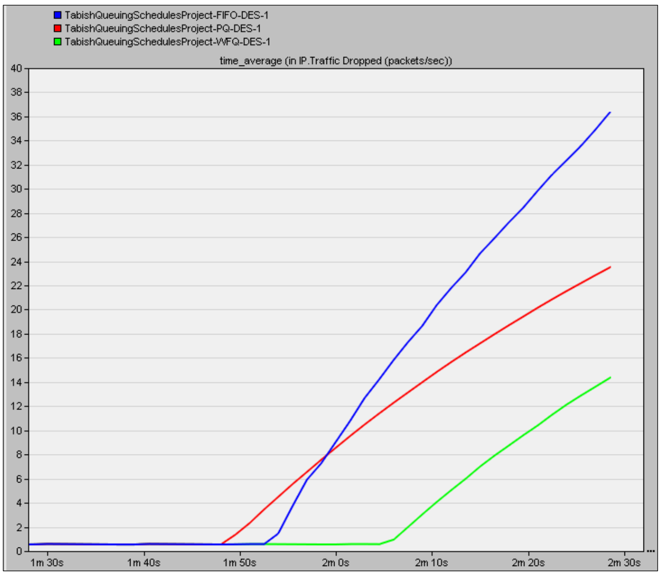

Figure 9: IP Dropped Traffic (Time Averaged)

Measuring the overall packet drop in the network for the three schemes points to the following results:

- In all the mechanisms, there are no packet drops up to a certain point. This is because of the fact that it takes a finite time for buffer memory to be filled up. Since packet drops occur only after the buffer is full, thus there is an initial period when the buffer is not full and hence there are no drops.
- In FIFO scheme, the packet drop starts after PQ but before WFQ. More prominently, the number of packets being dropped is the greatest in the case of FIFO. This is by virtue of the fact that once congested, all incoming traffic from all the apps is dropped all together without any discrimination. This is because FIFO treats all packets as the same and enqueues/dequeues them in the same order as they are received.
- In PQ scheme, the packet drops start the earliest. Since PQ divides the queue based on priority levels, the overall size of the individual queues is divided up. Assuming a simple division of the memory into an “Important” Queue and a “Less Important” Queue, the queue size is halved. Thus, the packets of Video being directed to the “Important” Queue will cause the queue to be filled up earlier (because of the smaller capacity) and hence packet drop will start earlier
- The amount of packet drop in PQ scheme is lesser than that in FIFO. This is simply because the algorithm proceeds in a Round Robin fashion for the different queues. Thus, unlike in FIFO, whose single queue gets congested because of all FTP, VoIP and Video packet and causes all further packets to be dropped, only the heavy applications traffic is dropped in PQ. Thus, application packets with lesser loads are saved from dropping. Hence, overall packet drop is lesser than FIFO.
- In case of **WFQ**, separate queues are constructed for each application. Thus, per queue size is reduced even further. However, packet drop starts later than both FIFO & PQ. This is because of **weighted processing of the Queues**. Thus, the more demanding queue is allotted greater bandwidth which results in the problematic queue being emptied faster than usual and delaying the buffer filling up time. The degree of packet drops, however, is not altered too much as compared to PQ.

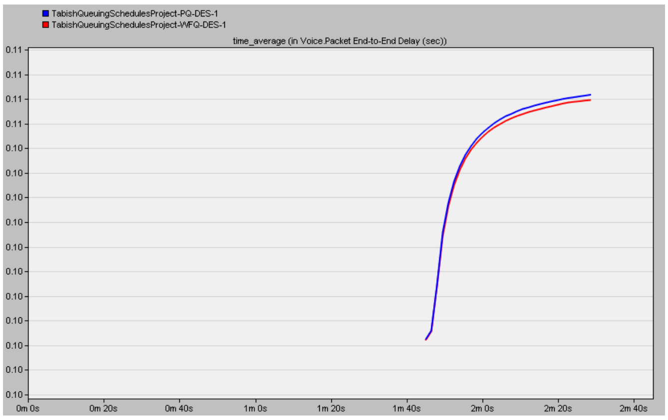

Figure 10: Delay Variation measured for Voice Application client in PQ and WFQ

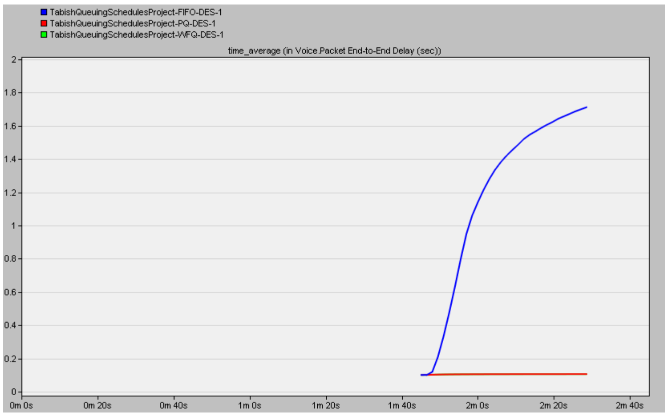

Figure 9: Delay Variation measured for Voice Application client in FIFO, PQ and WFQ

- Measuring the end-to-end delay for the low priority VoIP application indicates that using a PQ or WFQ mechanism results in drastic reduction in delays. Voice traffic is considered a higher priority traffic than FTP because of it utilizing the **UDP** protocol. Hence, separating the FTP flow from VoIP traffic flow allows voice data to be transmitted at greater rates.
- Thus, **real time applications** such as VoIP apps benefit greatly when using a Priority based service since they are prioritized achieving transmission of packets lacking any recovery method.

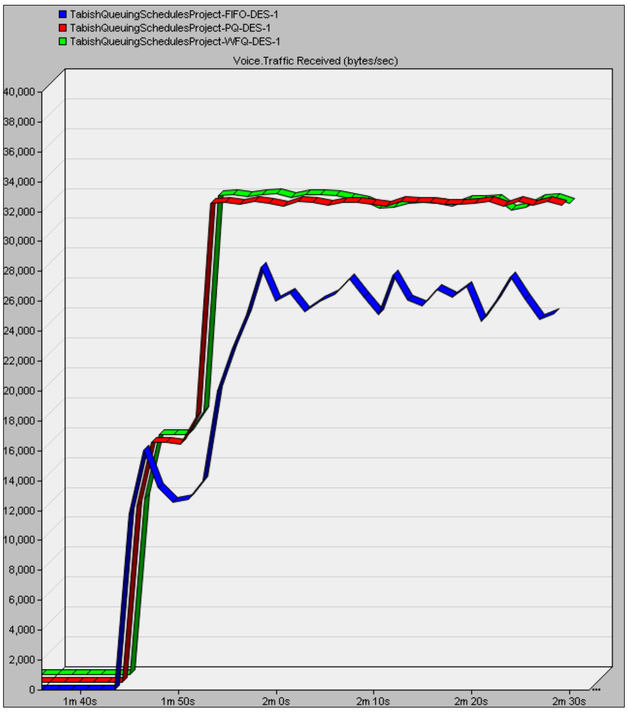

Figure 11: Voice Traffic Received (As Is)

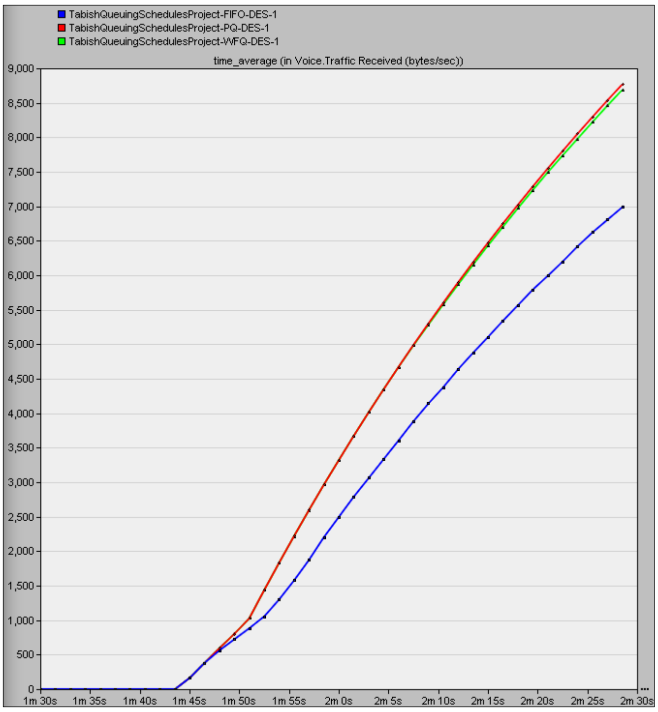

Figure 12: Voice Traffic Received (Time Averaged)

Voice Traffic Received (bytes/sec) by all nodes on the network indicates:

- In FIFO, the overall reception is the poorest. This is because of no special priority being assigned to Voice traffic. Thus, voice suffers greater drops
- In PQ and WFQ, voice is prioritized and separated from flow of other traffic. As such, more packets are successfully inserted into the router memory and consequently sent out on to the outgoing router links. Thus, the overall packet-drop amount lowers.

Conclusions.

Clearly, the use of a particular Queueing Mechanisms greatly affects Network Performance in certain apps. The choice of which scheme to use should be based on important factors such as Transport Level Protocols used, requirement for reliable delivery, requirement of real time or non-real time delivery, load imposed by the application, etc.

The performance of Weighted Fair Queuing and Priority Queuing appears to be identical in certain situations when dealing with router buffer overflows. However, WFQ performs better in terms of withstanding packet drops for longer periods by ensuring timely emptying of the various queues. Further, it prevents the possibility of overwhelming a low priority sender in case the buffer is being sent an excess of high priority packets from some contending application/applications.

Usage of priority-based schemes might appear appealing in most circumstances; however, FIFO schemes implement a simple, fair and low-overhead system that might be attractive to certain applications.

References

Iby & Aladar Fleischman Faculty of Engineering, “Queueing Discipline: An analysis of queueing in routers.” http://www.eng.tau.ac.il/~netlab/resources/

Intronetworks, “Queuing & Scheduling and QOS.” http://intronetworks.cs.luc.edu/current/html/queuing.html

T. Velmurugan ; Himanshu Chandra ; S. Balaji, “Comparison of Queuing Disciplines for Differentiated Services Using OPNET”, *2009 International Conference on Advances in Recent Technologies in Communication and Computing* https://ieeexplore.ieee.org/abstract/document/5328831/citations?tabFilter=papers#citations

Cisco, “Configuring Priority Queuing in Routers” https://www.cisco.com/c/en/us/td/docs/routers/10000/10008/configuration/guides/qos/qoscf/

Systemsapproach.org, “An Analysis of Queuing and Congestion in Networks” https://book.systemsapproach.org/congestion/queuing.html

Wikipedia Page on Queuing Delay in Routers https://en.wikipedia.org/wiki/Queuing_delay