University of Northern Iowa

UNI ScholarWorks

2014

Development of a multi-mode self-adaptive algorithm to create an efficient wireless network on a university campus

Melanie L. Abbas
University of Northern Iowa

Let us know how access to this document benefits you

Recommended Citation

DEVELOPMENT OF A MULTI-MODE SELF-ADAPTIVE ALGORITHM TO

CREATE AN EFFICIENT WIRELESS NETWORK ON A UNIVERSITY CAMPUS

An Abstract of a Dissertation

Submitted

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Technology

Approved:

_____

Dr. Jin Zhu, Committee Chair

_____

Dr. Michael J. Licari
Dean of the Graduate College

Melanie L. Abbas

University of Northern Iowa

August 2014

ABSTRACT

The expanding use of ubiquitous computing has created a significant demand on existing network infrastructures. The demands of voice, video, and data on the same medium require a quality of service (QoS) at a level acceptable to users. Many network providers simply scale their networks to increase bandwidth and hardware to deal with the increasing demands. However, a network may still reach its design limits with peak traffic or malicious overuse of resources. In addition, with technology changing at a rapid pace, it is difficult to provide sufficient staffing to monitor and adjust the network settings to avoid issues during periods of network saturation.

One of the common method to address these issues involves implementing a traffic shaper. A traffic shaper is a computer network management technique by which data sent across the network is delayed or routed in a way to accommodate a specific level of traffic to reach a desired QoS. There are many existing traffic shaping algorithms, each performing well under specific circumstances improving some QoS measures. The algorithms make use of queuing schemes to sort and send traffic based on the parameters provided to the system.

To determine the need for this research, a survey was administered which revealed dissatisfaction with QoS of the wireless network. The purpose of this study focused on the development of a traffic shaping algorithm that would improve the QoS on a local area network on a university campus. The goal of the research was to create a new architecture that would allow a router to dynamically shift between different queuing mechanisms to improve network delay and packet loss without negatively impacting data

throughput. The Multi-Mode Self-Adaptive (MMSA) algorithm was proposed to define a mechanism for this architecture. The MMSA was implemented within the code of a Cisco® router in the OPNET Modeler software and tested in a simulated university network environment. The results of the simulation revealed an improvement in end to end delay and packet loss rate with an insignificant change in average transmit rate between the router and the external server.

The results of this research can be used as a basis for future research to create a new QoS framework. The new framework could be implemented in a router to allow configurations tailored to the network requirements of a service provider.

DEVELOPMENT OF A MULTI-MODE SELF-ADAPTIVE ALGORITHM TO

CREATE AN EFFICIENT WIRELESS NETWORK ON A UNIVERSITY CAMPUS

A Dissertation

Submitted

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Technology

Approved:

_____
Dr. Jin Zhu, Chair

_____
Dr. Paul Gray, Co-Chair

_____
Dr. Shashi Kaparthi, Committee Member

_____
Dr. Ali Kashef, Committee Member

_____
Dr. Hong Nie, Committee Member

Melanie L. Abbas

University of Northern Iowa

August 2014

DEDICATION

Dedicated to my late father Marvin William Bennett, who encouraged me to continue my education and instilled in me the importance of learning something new every day.

ACKNOWLEDGEMENTS

I would like to take a moment to express my appreciation for the following people:

First and foremost I would like to thank my husband Rick for the many personal sacrifices that he made while I worked on my coursework and research. Without the support of my friends and family, this would not have been possible.

And a special thanks to Dennis Lindner, Director of Information Systems and the technical staff of the ITS Information Systems department who provided lab space, support and equipment that were critical in generating my simulations.

I would like to thank Aaron Howard, Ken Connelly, and Dr. Paul Gray for providing a means to capture actual network flows from the residence hall network to use as a basis for modeling the campus traffic.

Finally, I would like to thank my Graduate Advisory Committee, especially Dr. Jin Zhu, for all of their support and patience as I completed this arduous process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Wireless Local Area Networks (WLANs) have become the predominant means of gaining broadband access in the past decade. These networks are based on the IEEE 802.11 standards using basic service sets (BSS) which shares network channels across multiple access points. Since the medium of transport for these networks is not a physically controlled device but simply using the existing environment, there are various opportunities for interference that could cause issues with performance. In order to continue with success, several areas need to be addressed in 802.11 wireless networks. Some of those challenging areas are security issues (Hassell, 2004) and Quality of Service (QoS; Portoles, Zhong, & Choi, 2003).

The security concerns of wireless networking are often addressed by the IEEE standards committee creating new methods for wireless network transmission. Within a university environment, these security concerns are the ongoing task of the staff that support the infrastructure in place. The ability for support of other mobile devices on the network is driven by the demand of those that frequent the higher education campus. Historically, educational environments are not the first to incorporate new mobile devices on their networks, but wait until best practices are available to guide the process. The same holds true for implementing network controls that improve the QoS of the campus network. There are many standard practices for monitoring what appears to be inappropriate network traffic and stopping or limiting the bandwidth given to those systems behaving in that manner. In a wired network, the physical number of connections

is static as devices are not allowed to join the network without some intervention by the technology staff of the university. However, with wireless devices, the ability to change locations causing disconnections from one network access point and transitioning to another, makes it increasingly difficult to control the amount of traffic being presented to an individual network routing device. This increases the complexity of the rules necessary to develop an efficient network that provides sufficient QoS to the users of the mobile devices. In addition, wireless devices that should not gain access to the network have the ability to send transmissions on the same channels as the access points and interfere with legitimate users of the wireless network.

The objective of this study is to develop a new traffic shaping algorithm that will be applied to a wireless network to improve the QoS for users on a university campus. This study evaluated existing needs of wireless networking at the University of Northern Iowa, created a simulation environment to test various traffic shaping strategies, developed a new algorithm within the scope of the existing infrastructure, and tested this strategy in a network simulation. The results of this study can be used to determine full implementation of this algorithm at the University of Northern Iowa or other institutions of higher education.

<div align="center">Statement of the Problem</div>

The problem this study addressed is poor QoS issues, in particular end-to-end delay, in a university that has not implemented a traffic shaper to address unpredictable network traffic patterns.

Statement of Purpose

The major purpose of this study was to develop a new self-adaptive network traffic control algorithm for maintaining a QoS based network experience on a university campus.

Need for the Study

The justification for the study is based on the lack of research guidance to university professionals on implementing a network traffic shaper to enhance the user experience. Providing reliable network access for a campus-wide community is critical in higher education. In order to provide a reliable network, it is necessary to put in place systems that would prevent saturation of the network bandwidth whether it is intentional or accidental.

There are many commercial products available to implement traffic shaping, however, they can become expensive and complicated to maintain. Some campuses are purchasing additional bandwidth in an effort to eliminate traffic shaping requirements, which still leaves issues with network performance when the network is being abused. Many of the pieces of equipment that are owned by the university have the ability to use traffic shaping schemes but are rigid in their design functions. The Cisco® routers that are commonly used include software to implement QoS using policing and shaping. This allows limiting of traffic based on rates inbound and/or outbound from the router. The shaping that is provided is called Generic Traffic Shaping (GTS) which provides a limited set of traffic shapers. If a new algorithm were applied to the existing hardware that was dynamic to the environment, it could be cost effective enough for the campus to

use and would provide the flexibility needed to provide a robust environment for students and staff.

Statement of Hypotheses

The goal of this research study was to evaluate existing traffic shaping strategies and develop a new algorithm to improve the QoS of a simulated network.

The research hypotheses were:

1. Students and staff are unsatisfied with the performance of the wireless network at the University of Northern Iowa.

2. Traffic on a university campus various significantly in content based on the time of day and day of the week based on course offerings.

3. A method to allow a traffic shaper to change behavior dependent on the demand instead of the time of day would be more effective for abnormal uses outside of the normal traffic patterns.

Hypotheses Tests

The hypotheses were tested using the following tests:

1. A wireless satisfaction survey was conducted at the University of Northern Iowa in the spring of 2012 to evaluate the opinions of the students and staff. An executive summary was created and provide to Network Services. The information provided by the survey reported averages of respondents by question.

2. Network Services collected data on the ResNet segment of the university network for a period of ten days to provide data for this study. The data

was reviewed with a graphical product for data flows to show times of
heavy bandwidth usage. The most prominent traffic types, speeds, and
protocols were gathered by using system commands on the netflow data
capture files.

3. An OPNET simulation was developed to represent a small network using
   a Cisco® router capable of QoS settings. A series of tests were conducted
   with various QoS schemes and compared for performance using Ethernet
   delay times, TCP segment delays and network utilization levels. A custom
   QoS scheme was created and tested on an expanded network to attempt to
   improve performance.

Assumptions

Assumptions in this study were based on previous experiences as a faculty, staff,
and student at a university. Those assumptions would include:

1. The OPNET simulation software used accurately represents real traffic as
   modeled.
2. The gathered network data was representative of a normal ten day window of
   time on the ResNet segment.
3. The computer running the simulation was not impacted by any other system
   processes that would impact the results of the simulations.
4. The custom scheme created in the OPNET simulation could be coded into a
   real Cisco® router.

Limitations

The nature of this study involves some items outside the control of the researcher.
These limitations include:

1. Due the nature of simulations, the results may not represent exact traffic
   when implemented on real devices.

2. Simulations longer than 30 minutes were not explored due to the amount of processing time to complete a single 30 minute scenario was 8 hours.

<div align="center">Statement of Procedure</div>

The procedures used were conducting a survey, network traffic data collection and analysis, development of a new traffic control methodology, and an experiment to apply the new control algorithm in a network simulation and measure the results. The first step surveyed the campus to determine concerns with the current wireless network. The next step gathered data from normal wireless and wired network traffic. The data was analyzed to determine the type of traffic mix and load to test with the new algorithm. A network model was created using the OPNET Modeler software to test existing traffic shaping methods. Based on the results, a proposed algorithm was created to dynamically change the QoS method on the router to take advantage of each of the strengths of the baseline traffic shaping methods. A larger simulation was developed to provide more traffic and five simulations were run for each QoS method to compare results.

Wireless Services Satisfaction Survey

A survey was administered to all UNI enrolled students and current employees regarding the wireless services that are provided to them because of their relationship to the campus. This survey was distributed using the MyUniverse online survey tool system. The purpose of the survey was to determine the experiences that the respondents had with wireless in general and more specifically how they have used the campus systems. The key question on the survey asked about issues that had been experienced using the wireless system and what tasks they were trying to complete. The survey results were

calculated using descriptive statistics on the multiple-choice answers and the grounded theory on the open-ended questions. There survey was completed by 325 individuals.

Gathering of Network Data

In order to design the best QoS for this research, it was important to have a valid set of data that represents the existing network traffic at the university. The department of Information Technology Services - Network Services agreed to provide data that was representative of the campus for this purpose. The information did not identify any individuals. The data was used with simulation software to analyze the flow of a similar load and type of traffic repeatedly.

Development of a Traffic Control Algorithm

The next procedure included creating a new algorithm to use on a traffic control device that would produce a desirable QoS on a wireless network. Existing algorithms were evaluated during this time and a new procedure was proposed. The development was completed in the OPNET simulation tool by extending the model structure provided with the software.

<div align="center">Definition of Key Terms</div>

The following terms are defined to clarify their use in the context of the study:

- Bandwidth – the maximum amount of data in bps that is consumed or available to a network device at one time.

- End-to-end delay – the time it takes for a network packet to be sent from the source to the destination

- FIFO – first in/first out – a method for queuing network packets for transmission which ensures that the first packet received is the first packet to exit the queue.

- Multi-Mode Self-Adaptive – in a network, the ability to be flexible enough to alter itself as necessary to harden the flow of traffic as circumstances change.

- Throughput – the amount of data that passes through the network at any point in time usually measured in bits per second (bps).

- Traffic shaping – a computer network management technique by which data sent across the network is delayed or routed in a way to accommodate a specific level of traffic to reach a desired quality of service (QoS).

- Utilization – the amount of network traffic measured in a percentage of the maximum amount allowed based on the bandwidth of the network infrastructure.

- Quality of Service – the user experience of receiving a service; in regards to computer networks: the speed, stability, and efficiency of a network service that is requested by the user of a device on the network.

<p style="text-align:center">Outline of the Dissertation</p>

The dissertation is organized as follows. The second chapter is a review of existing literature related to the subjects of traffic modeling, network QoS, traffic shaping and network simulations. In Chapter 3, the background on different traffic shaping algorithms is discussed leading up to the development of a proposed new self-adaptive algorithm. Chapter 4 is a detailed explanation of the implementation of the new algorithm within a network simulation tool and the results from those simulations. The final chapter is a summary of conclusions and suggestions for further research related to this subject area.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter will review the use of networking on a university campus, traffic modeling, network QoS, traffic shaping and network simulations. The chapter begins with a basic overview of the increase in wireless network use on a university campus and the amount of traffic caused by this increase. This chapter discusses how traffic is modeled, network performance management with QoS and current research in network traffic algorithms. The chapter concludes with an overview of current network simulation software uses in QoS research.

<u>Background</u>

Statistics confirm that since 2007, the number of wireless data services that are available to users has increased at an annual rate of 40%. The number of wireless devices sold annually has increased from near zero in 1999 to hundreds of millions each year (Ciampa, 2013). Along with the increase of users there is a growing theory of study in the socio-technical arena of domestic space (Howard, Kjeldskov, & Skov, 2007). Since a University often has students and staff that reside on the campus, the domestic space includes the typical campus. A study involving students has been performed on the domestication of laptop use at a college in Finland. The study was intended to include the wireless network, yet the students reported low use or concern over the wireless services they were provided due to performance issues. In addition, the completed study was in 2005, which suggests that it should be repeated with more recent data to verify the results (Vuojarvi, Isomaki & Hynes, 2010). This statement shows a need to study the

performance of wireless services on a campus by students. This need can be expanded to also include faculty, staff and guest users.

Students, faculty, and staff are coming to universities with prior expectations and knowledge of using wireless networks. The services that these constituents desire require implementing services in ways that are robust and capable of large amounts of traffic with peaks between classes and into the night time study hours. Other studies have shown that meetings are impaired when sufficient data is not available which can be remediated by the use of wireless networking (Ciampa, 2006). This has driven the desire of mobile networking into the business world. A university is a very large operation with many meetings that fit into this category. In addition, it is a learning environment in which students expect access to information from multiple locations and professors to answer questions on demand. The technological answer to this issue is to install high performing wireless technologies to support these needs (Ciampa, 2013).

<div align="center">Traffic Modeling</div>

In order to properly design network infrastructure pieces, it is important to model network traffic model that relates to the specific topology being researched. A traffic model is an abstract representation of a network designed to study or modify real devices for a variety of purposes. One of the main reasons to model traffic is to properly implement a network that can handle the amount of bandwidth expected from a given number of nodes and servers. Another reason is to configure QoS parameters using queuing theory to the routing devices expected in the physical connections of the network.

Many traffic models are creating by predicting traffic based on a time series using bandwidth use over a period of time. In short periods of time, this type of model can hold up, however, some research has shown that high-speed networks long correlation cannot be handled accurately with this method (Juqiong, Chu-Ho, & Xu-dong, 2013). In research completed by Juqion, Chu-Ho and Xu-dong, the use of chaos theory was used to predict network traffic which was then run through an experiment for validation. The proposed prediction method using chaos and regression simple multiple kernel learning (RSMKL) showed promise in more accurately modeling complex network traffic over time which can be generalized and applied to complex networks. This predictor could be used in network performance tuning development.

In traffic engineering, development is aimed at dynamically analyzing the data transmissions to reduce congestion (Jain, Kumar, & Chaubey, 2011). In a study by Jain et al., a dummy server is placed into a mesh or ring network that becomes active only when needed based on queues on other nodes becoming full.

In a study of traffic modeling using the Brownian traffic model, samples were gathered from TCP/IP traffic traces over a period of three years from two businesses that were globally connected. However, it was determined that a monofactual traffic model cannot accurate represent modern network traces and as such are better defined by multifractal analysis due to the burstiness of the traffic (Vieira, Bianchi, & Lee, 2010). In similar studies, researchers tried to estimate unobserved point-to-point traffic based on aggregate data gathered from businesses. The proposed traffic model was a multilevel

state-space strategy which was shown to out predict the standard practice of modeling based on normal time-series calculations (Airoldi & Blocker, 2013).

<div align="center">Network QoS</div>

In studying QoS in a packet-switched network, there are several measurements that are typically used. End-to-end delay, also called packet transfer delay, is one of the most common statistics gathered. To accurately gather this number on a network, a timestamp is taken when a packet leaves the source system and another when the full packet is received at the target system. This is a one-way delay measurement which is useful in understanding the total time of transmissions (Ash, 2007). Another popular measurement related to delay is packet delay variation which compares a particular packets delay with a defined reference delay from the source to destination. Delay variation can cause issues with traffic such as streaming video which may cause buffer overflows and lag time in playback.

As noted in research by Chaudhuri, there are differing demands for QoS in regards to real-time transmissions. The performance of those applications are more sensitive to delays and packet loss than non-real-time traffic, such as email. These are most likely to be limited by the buffering abilities of the network devices such as routers or switches (Chaudhuri, 2012). There can be multiple layers of QoS implemented on a network, one at the application layer and another at the network layer. While application layer implementations are concerned with jitter, the network layer is focused on bandwidth performance measurements. The metrics that can be used to specify and gauge QoS level offered by a network are summarized in Table 1 (Mehmood & Alturki, 2011).

Due to the increase in multi-media traffic and its impact on performance, it is a

popular current research topic. Video is often chosen for research because of its high

demand in terms of bandwidth, buffering and timing requirements (Ma, Barto, & Bhatia,

2011). The aim is to achieve acceptable response to watch videos on wireless devices.

However, since video is rendered at a specific rate, it is significantly different than other

traffic that is more easily managed with common network shaping devices. Video

involves caching, at a client or server level. A video can be sent from a cache server to a

client or from another client in a peer-to-peer network. There are other combinations of

these two basic scenarios that are in use depending on the number of servers and the

location of the video caching.

QoS is often studied in wireless networking due to the high amount of traffic

overhead from devices associating with access points as they move through the

environment. Research suggests that admission control mechanisms and management of

bandwidth are necessary to provide the desired QoS on a wireless network to meet user

expectations (Soudani, Divoux, & Tourki, 2012).

In research conducted by Cahlan, a new algorithm was proposed for the selection

of wireless access points in a WBAN. The basis for the algorithm was a development of

parameters used to in a fuzzy logic approach that provided a score to each wireless access

point within the range of the coordinating node used to transmit information from a set of

body sensor networks on a moving subject. In performing a series of case studies, it was

determined that the fuzzy logic selections were superior to the standard RSSI-based

algorithms in providing reduced end-to-end delays in the transmissions (Calhan &

Atmaca, 2013).

Table 1. *QoS common measurements*

Quality of Service Measurements	Definition
Packet loss	When a packet is dropped because there is insufficient buffer space in the network queue to hold any more packets. This is a numeric representation over a period of time.
Retransmission count	This represents packets that were retransmitted because of network errors. This includes packets dropped and other transmission errors between the source and destination.
End-to-end delay	The time it takes for a network packet to be sent from the source to the destination
TCP segment delay	A measurement of the time between when a TCP sends a segment of data from source to destination and when the acknowledgement (ACK) is received back at the source. This is a time measurement often in milliseconds.
Jitter or Packet delay variation	Measured as the variability over time of packet latency between the source and destination on the network. The measurement is time based in milliseconds.
Throughput	The rate of successful data delivery over the network measured over a period of time. Typically this is measured in bits per second (bps) or packets per second (pps).
Queuing delay	This is the time that a packet spends in a queue on a network interface before it is transmitted.

Traffic Shaping

In performing a review of the literature related to network traffic issues, there are

several main areas of research. Most are aimed at specific types of network or how to

address a limited scope of problems related to QoS.

Ma et al. discuss three techniques of network management that have been used to

improve the QoS for Internet-based video transmission: network load reduction, network

interruption mitigation, and network load distribution. Each of these schemes handle multi-media traffic differently at the server or client location and include modifying how the data is encoded and the rates that are used during transmission. However, researchers state that continual investigation of current network data traffic would is necessary due to the increasing reliability of networks over time (Ma et al., 2011). They suggest that future research look into two primary methods for delivering video: streaming and segmented.

Research by Soudani et al. (2012) provides a suggested load balancing algorithm that would specifically address wireless networks and video transmission performance. The structure would include a load balancing server that would assign access points (AP) in a fair method to improve overloading of any particular AP.

The 802.11 network protocol uses Distributed Coordination Function (DCF), which senses traffic and tries to avoid collisions from multiple devices broadcasting at the same time. Due to nature of DCF, there are often retransmissions of the packets for those with lower signals, such as those at the far reaches of a wireless network. These can cause excess traffic and wait times for the other nodes on the wireless access point (AP) and thus affect the QoS for all of those connected to the AP (Bellavista, Corriadi, & Foschini, 2008).

While trying to improve video traffic, the researchers suggest ways to detect a specific anomaly and modify the traffic so that it is less detrimental to the network. This can be completed with a proxy server using something called Mum that was developed by the researchers (Bellavista et al., 2008). This server looks for video content and modifies the encoding and distribution to the nodes based on criteria so that the video is

usable at all locations without impacting performance. This research did not address any other types of network traffic besides videos. They suggest further research into QoS improvements for different types of video encoding.

In a more recent traffic shaping study by Bellavista, a survey of recent video network transmission papers is performed. The papers were those developed for the mobile middleware conference in Mobilware '10. The summary provided abstracts for a few papers based on issues related to heterogeneous wireless devices with different operating systems. A common theme revolved around creating 'weblets' to separate out pieces of a particular piece of software that they could run independent of the physical device. One example was a GPS/location application that was used to detect traffic accidents. The research focused on trying to decrease the amounts of false positives reported due to network transmission problems (Bellavista, Cai, & Magedanz, 2011).

Several areas of research involve proposing new algorithms to improve network traffic flow and thus QoS. While developing these algorithms, different types of network data are classified or prioritized to allow a means to shape the network during periods of congestion. Network Service Providers (NSP) often have certain service level requirements in agreements with those who purchase their services. In order to maintain those levels, it is important that the users of the wireless systems be provided with fair access to desired applications no matter what their location. Usui, Kitatsuji, and Yokota contend, that in order to maintain fairness, accommodations would be necessary to guarantee a specific level of capacity between any pair of edge routers in the NSP (Usui, Kitatsuji, & Yokota, 2011).

Yet some network traffic needs to have priority over others, especially when it involves something as important as human health information. In a study about Wireless Body Area Networks (WBANs) a prioritization algorithm was created that was based on the contents of the data being transmitted about a patient across a Wireless Sensor Network (WSN; Cheng, Huang & Tu, 2011). The Random Contention-based Resource Allocation (RACOON) medium access control (MAC) protocol was proposed to support the QoS in a medical treatment environment. This type of prioritization looked much deeper into the traffic to bump the priority of a node on network if a patient were to have a sudden abnormal reading. The protocol also allowed higher prioritization by patient type, such as those that had entered the hospital from an injury.

One newly proposed algorithm to improve traffic flow on Internet connections aims at the very core of basic network protocols (Dhanaskaran, Jayachandran, Rajan, KharmegaSundararaj, & SudarsonRamaPerumal, 2011). The article states that the current transmission control protocol (TCP) is based on not losing packets at the price of performance. Dhanaskaran and others do not feel that TCP is holding up to what is needed by users today due to the changes in network use. They have proposed an algorithm called adaptive RED to change the flow of network data to use node-to-node paths with weights. The proposed algorithm calculates the best path to the destination based on information from the transmitting node and adjoining router instead of information that is currently transferred from one server host to another on the Internet as is currently executed by TCP (Dhanaskaran et al., 2011). The researchers present some sound theory, but have yet to show actual implementation of this algorithm change.

Another set of researchers looked at creating a dynamic algorithm that was based on the traffic on an 802.16 WiMAX network to improve speed of transmissions while still maintaining fairness. Geetha and Jayaparvathy evaluated three common algorithms, fixed priority, equal priority and dynamic weighting. The two priority based methods are often used when creating traffic classes giving priority by the type of data while giving each either a specific amount of bandwidth either equally by class or differentiating the priority of the class. In contrast, the dynamically weighted algorithm determined an extra value associated with each class of traffic but allowed it to change the weight assigned to the traffic depending on the load at that point in time. These three algorithms were evaluated using a network simulator. The researchers determined that when simulating with three classes of traffic, two classes of network traffic were significantly affected when the prioritization algorithms were used individually. The conclusion was that the dynamically weighted method resulted in the best QoS in regards to delay times. For the purposes of this study, the QoS was evaluated in terms of delay times (Geetha & Jayaparvathy, 2011).

Similar studies involving 802.11 ad hoc networks transmitting multi-media content were completed using the OPNET simulator. However, the researchers did not use delay times as primary measurement for QoS in this study because they contended that network throughput was more important due to the high data requirements of multi-media (Mehmood & Alturki, 2011). The authors believe that the future of network performance depends on multiple network layers in the OSI model, and suggest that future research be completed in cross layer QoS architectures.

In research completed by Portoles et al., an algorithm was process to increase QoS in a BSS wireless network system to address issues with a wireless device that was having network issues (Portoles et al., 2003). The algorithm was meant to limit the amount of packets that were allowed to be transmitted to any specific network devices downloading from the network so that if it was not receiving all the data it requested, it would not continue to use all the bandwidth until the transmission was complete. The algorithm was not concerned with uploaded network traffic as the basic contention based protocols on the wireless access point already manage the ability to upload through the network.

In a round robin approach to network shaping, queues are configured and processed in a rotation to provide equal transmissions to each queue. In a Modified Weighed Round Robin (MWRR) approach proposed by Mardini and Alfoul, the different queues would be weighted such that those with higher weights received a great amount of exits from their queue than those with lower weights. This approach is aimed at decreasing jitter and end-to-end delays without negatively impacting average throughput (Mardini & Alfoul, 2011).

One of the most popular traffic shaping algorithm is the hierarchical token bucket (HTB) method. The basic premise of this method is to have all traffic flow through a process where the data is classified by type and prioritized for transmission. The traffic all flows into what is considered a bucket and does not leave the bucket unless it has been granted a token to do so. The researchers applied this algorithm to the 802.11b wireless network to show that if implemented between the Internet Protocol (IP) layer and the

Media Access Control (MAC) layer, it can improve the inherit QoS problems of this wireless standard traffic (Valenzuela, Monleon, San Esteban, Porteles & Sallent, 2004).

Another hierarchical algorithm used in Asynchronous Transfer Mode (ATM) networks is based on a timing queue traffic shaping architecture. Research showed that when added a resorting piece to that design, it that allowed priority to be given to higher speed connections, thus reducing the delay for transmissions (Zeng, Uzen, & Papvassiliou, 2001).

Another example involved QoS and traffic management in a broadband network environment focusing on optical network units (ONUs). The researchers compared different methods to use a traffic-shaping device that splits the type of network traffic into different classes. These classes of traffic, such as e-mail, web browsing, video on demand, were then limited based on the standard methods for each of six algorithms (Hwang & Yoo, 2011). One issue they studied is starvation. Starvation occurs when one class of network traffic is limited by a priority that is lower than another class that is overloaded, causing the lower priority transmissions to be halted altogether. For example, if video on demand was given higher priority to bandwidth than e-mail traffic, during times of large video viewing on the network, the algorithm would not allow e-mail to transmit until the video use declined. This causes unfairness in the handling of the users of the wireless network and thus unacceptable QoS.

In conclusion of the simulation results by Hwang and Yoo, it is suggested that a new algorithm be designed that contains parts from the different algorithms they explored. The ideal features would include regularity in cycle time, inter-class bandwidth

allocation for service differentiation and intra-class bandwidth allocation for fairness

(Hwang & Yoo, 2011).

<div align="center">Network Simulations</div>

For simulation purposes, if an available software package does not exist, some

researchers have written their own in languages, such as C++ (Guerrero-Ibáñez,

Contreras-Castillo, Barba, & Reyes, 2011). The simulator was written to process around

two hours of network traffic from wireless cellular transmissions for a study on allowing

customers to pick a network based on the services and QoS provided at any point in time.

There are several network simulator packages used by education for research

purposes, including Network Simulator 2 (ns-2), Network Simulator 3 (ns-3), Matlab,

and OPNET. The OPNET Modeler can be configured to represent a large amount of

network traffic sent through a simulated set of network devices.

Within the OPNET Modeler simulation environment, there are predefined QoS

schemes that can be used on a router to evaluate network performance. These different

methods are implemented at the scheduling level by setting up queues and determining

which traffic goes into which queue and the order in which to service each queue. The

most basic method is first-in-first-out (FIFO) processing. In this scheme, packets are

placed into a queue in the order which they are received and exit the queue in the same

order. If the queue is full at the time of arrival, the packet is dropped and must be

retransmitted. Priority Queuing is small modification to the FIFO queue where each

packet is marked with a priority level. There are different queues created for each priority

level and packets are placed in the queue and released from the queue in a FIFO manner. The packets in the higher priority queue are processed before those in the lower queues.

The weighted fair queue (WFQ), also referred to as flow-based WFQ, implementation in OPNET manages a separate queue for each flow that enters the router. Packets with the same source IP address, destination IP address, source or destination TCP port or UDP port are considered in a single flow. The queues are given weights which defines the percentage of the bandwidth that can be used by that particular queue. The queues are served in a round robin fashion to produce fairness of traffic handling.

One type of aggregate traffic that has been studied in OPNET is communication that generates a high volume of variable data rates creating what is called burst within burst phenomena (Daniel-Simion, 2012). One example of this is the software BitTorent which is used for distribution of large files. This type of traffic needs a large amount of bandwidth in order to avoid disrupting other network activities. The study concluded that traffic shaping could improve the QoS and support service level agreements, but did not address issues of self-similarity or bursts (Daniel-Simion, 2012).

Another research project related to ZigBee networking in a hospital environment looked at differing configurations to compare end-to-end delay and system data throughput. In creating a simulation composed of three scenarios, the results showed that a star topology showed the best performance with bursty data traffic, while mesh and tree topologies were weaker. The traffic being pushed was about 5000 bps and thus only a workstation in the center of the star topology of the Zigbee network could receive the full traffic when large bursts of data were received (Hu, Dung, Liu, Han, & Sun, 2013).

Research conducted by Athanasiou used the OPNET simulation modeler to investigate proposed changes to improve performance in a wireless network. Performance improvements in device associations with wireless access points is a continuing area of research. The existing RSSI algorithms often result in poor network performance and inefficient resource utilizations (Athanasiou, 2012). Athanasiou's article continues to provide an overview of existing suggested performance enhancements to the existing WLAN standards. The author provides and an overview of research on a cross-layer association mechanism to be used in mesh networks (Hiertz et al., 2010). In this technique, each wireless device would receive information on the current uplink and downlink airtime costs based on a complex algorithm.

However, this can lead to some access points becoming too popular because it does not consider the costs of the associations/dissociations, so to maintain the best end-to-end QoS for the network, each wireless device sends probe frames to the access point to determine if the connection would be the best option available. The author proposes a new algorithm whereby each wireless device maintains a table on APs within range and shares this information with other nearby wireless devices. The table stores the same cost information that was used in the cross-layer technique, however, it is now no longer broadcast or maintained by the access point. The end results showed that the throughput, average transmission delays, and dropped data were all improved by these changes to the normal 802.11 association process (Athanasiou, 2012). The author concludes by further work on load balancing of the wireless network in conjunction with the new association process to provide even greater QoS in the simulated mesh network.

Researchers looking at lower level wireless control systems mention using a combination of simulation products. They used a platform called PiccSIM which simulated the control systems in Simulink and the network in Ns-2 to create a rapid wireless development of new wireless technologies. Another well-known alternative to Simulink was developed at Lund University. The Simulink based simulator is used to evaluate real-time scheduling of embedded processors and wireless network such as Bluetooth (Bjorkbom, Nethi, Eriksson, & Jantti, 2011).

CHAPTER 3

DEVELOPMENT OF THE MULTI-MODE SELF-ADAPTIVE TRAFFIC SHAPER

Introduction

The main objective of this research study was to study the performance of a

campus network and how to improve its efficiency. The campus network at the

University of Northern Iowa (UNI) was used as an example for the study.

The first step of the research was to study the performance of the current UNI

campus network and identify existing problems and issues. Next, different traffic shaping

algorithms were studied to evaluate their effect on the performance. Finally, a proposed

self-adaptive traffic shaping algorithm was designed to improve the network efficiency

according to QoS metrics. The remainder of this chapter will explain the details in this

experiment and development of a proposed algorithm to improve selected network

metrics.

Wireless Satisfaction Survey

The study started by performing a survey of all UNI students, faculty and staff to

review the satisfaction with the wireless network that is in place at UNI. Questions on the

survey were aimed at determining which features of the network were considered

unacceptable for use due to performance concerns. The survey was completed by 258

students and 67 faculty/staff members. Of the 325 respondents, 320 had used wireless

devices on the Internet at various locations. The top four uses of wireless for both groups

were in the categories of communicating with others, emailing, general web browsing,

and social networking. The other uses of the wireless network varied in percentages but

were similar in order including watching movies, posting web content, online banking,

and online video games coming in the next group. The results show that students used the

campus wireless network much more frequently than the faculty and staff. However,

both groups had experienced issues with wireless performance on campus with 84.2% of

students and 61.2% of staff reporting very slow response from the UNI wireless networks

or the inability to perform a task such as viewing a video. When asked which specific

activities they would avoid on the campus wireless system, the top complaints varied

among the groups. Students' top complaints were videos, webcam or phone software, and

web browsing. The top items for faculty/staff were email, web browsing, and online

videos.

In reviewing the survey conducted, it is noted at 71.9% of students responded that

they are avoid watching videos on the wireless network due to the speeds available. Yet,

60.8% of students claimed to watch movies and several times a week or daily. Using

Skype or other video conferencing systems is also too slow according to 60.8% of the

students surveyed. Full survey results can be found in Appendix A and Appendix B.

It is evident from the survey and the network data obtained later in this chapter

that the use of video on the wireless network is heavy but also unsatisfactory. When

evaluating performance of video traffic on the network, there are a couple QoS metrics

that are valuable. When referring to speeds, there are multiple network components that

can be investigated to improve the perceived response times. One item to consider is the

throughput as is often seen waiting for a video clip to buffer in the video player software

before it is available to view. With regards to video streams, another common issue to

impact performance is packet delay. This research will focus on finding a method to decrease the network delay without negatively impacting the throughput in bits/second.

<p align="center">OPNET Simulations</p>

In order to study the network performance, a simulation tool was selected to model the UNI network. The simulation tool chosen was the OPNET Modeler.

Introduction to OPNET Modeler

The chosen network simulation tool provides is a suite of protocols and hardware models that allow for complicated development of new technologies. The OPNET software is a product available for educational research and commercial use by Riverbed Technology®. The software allows creation of simulated networks using hardware such as workstations, routers, wireless stations, and network links to represent small to enterprise level network environments. OPNET Modeler is used to improve research and development (R&D) by demonstrating a new design before it is implemented. To do so, the software provides complete access to the source code which can be modified by a researcher to test new enhancements with differing sets of network configurations.

The OPNET Modeler is a hierarchical system created to represent a real network of equipment, applications, and protocols. The simulation tool allows creation of projects using the project editor to graphically represent the network scenarios used under investigation. Within the project editor, an object palette is utilized to add nodes to the network which define the equipment in the simulation. Each node has underlying software that defines the processes that node can perform. These processes are defined using a powerful finite state machine approach to allow detailed specifications for the

protocols, resources, applications, algorithms, and queuing policies for that node. The

underlying C/C++ code that is associated with the processes can be edited for

customization to quickly develop new hardware or software solutions within the network

infrastructure. This hierarchy structure allows drilling from the Network -> Node-

>Process->Source code as displayed in Figure 1.
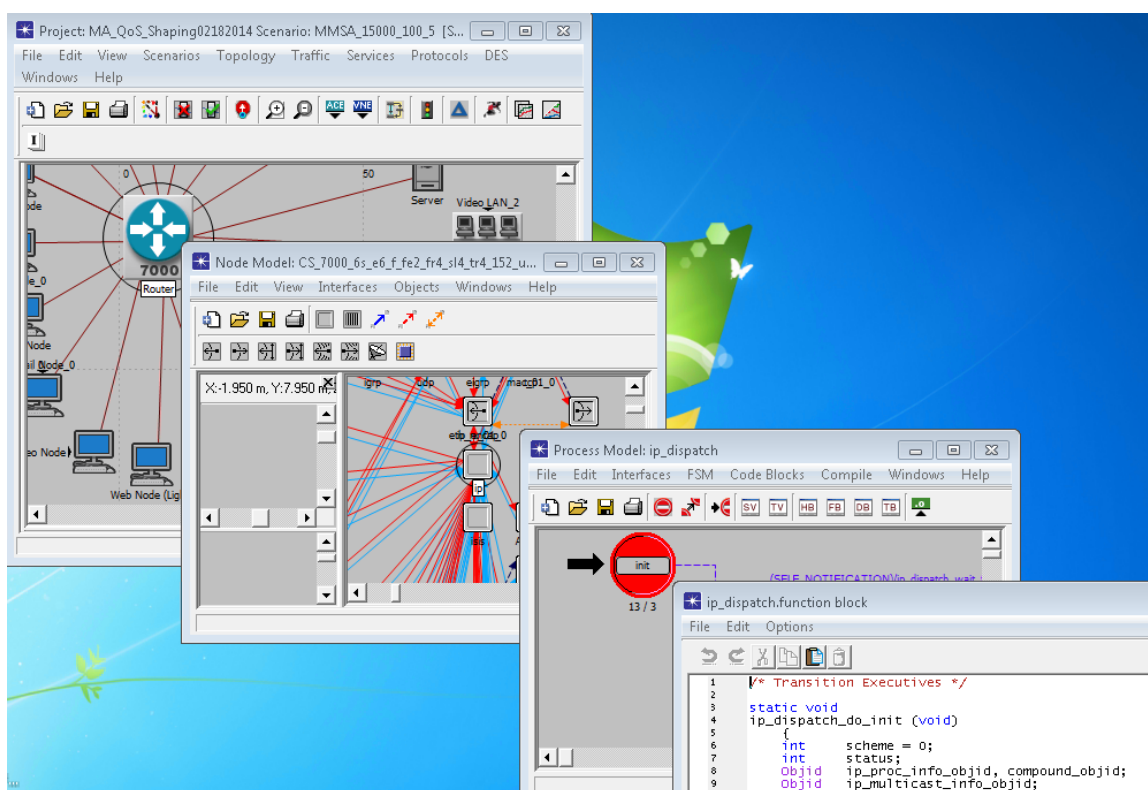
Figure 1. OPNET hierarchy structure.

     The software also provides several QoS settings that can be tested and

customized. There are a series of steps involved in creating the OPNET simulation

environment, including setting up the network devices, configuring the application

definitions, application profiles, and QoS attributes, applying them to network devices,

and selecting simulation statistics. These steps are detailed in Appendix C.

<u>QoS Schemes</u>

Traffic shaping is a computer network management technique by which data sent

across the network is delayed or routed in a way to accommodate a desired QoS level

defined by a traffic profile (Blake et al., 1998). There are several basic schemes that are

used in the industry.

<u>First in first out (FIFO).</u> The most basic packet handling method is FIFO

processing. In this scheme, packets are placed into a queue in the order which they are

received and exit the queue in the same order.
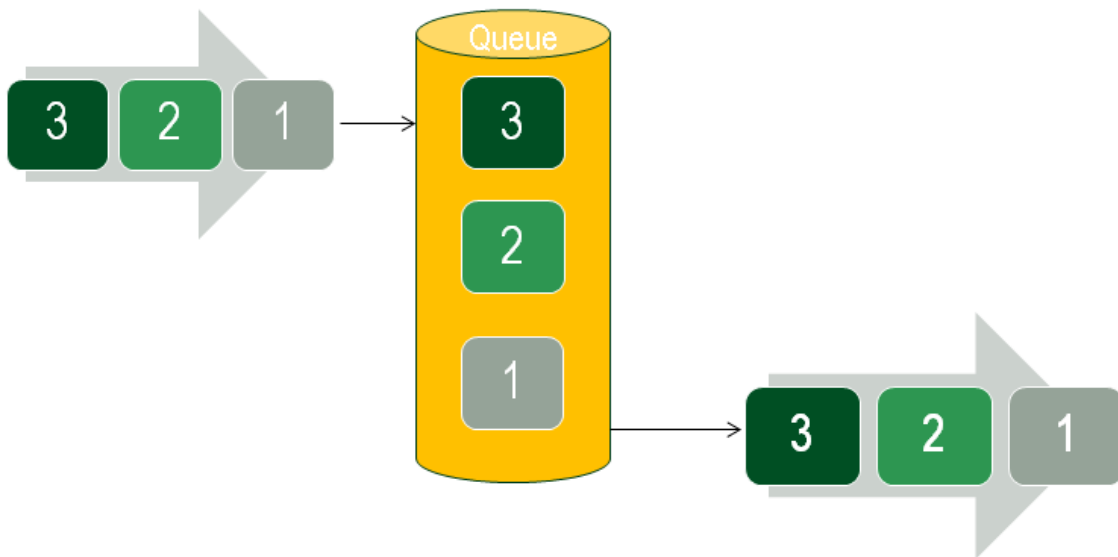
F*igure 2*. First In First Out Packet Processing.

If the queue is full at the time of arrival, the packet is dropped and must be retransmitted or it will be lost. The packets are processed in a single file flow through a single queue as shown in Figure 2.

Due to the simplicity of FIFO queuing, the overhead placed on the network system is extremely low (Semeria, 2001). However, a limitation with FIFO is the single queue is used for all traffic, thus all classes of service are impacted during periods of congestion due to high network demands.

Priority queuing (PQ). Priority Queuing is small modification to the FIFO queue where each packet is marked with a priority level. The priority level can be based on different attributes of the network flow including the source ports, destination ports, or application content of the packets. The packets are marked with a Type of Service (ToS) indicator that is used by the scheme to determine the priority level of that particular packet. There are different queues created for each priority level and packets are placed in the appropriate queue and released from that queue in a FIFO manner, as shown in Figure 3. All of the packets in the higher priority queue are processed before those in the lower queues. The process checks the high priority queue for packets, and if they exist, it sends them, until the queue is exhausted.

This scheme has very little overhead on the router so it is relatively quick, however, if the amount of high priority data flows becomes excessive, the lower queues can start seeing delays in service or dropped packets due to overflowing queues. This issue is commonly referred as starvation. In addition, if a high priority flow is

misbehaving, the flows sharing the same queue can experience longer delays and

increased jitter causing reduced QoS (Semeria, 2001).
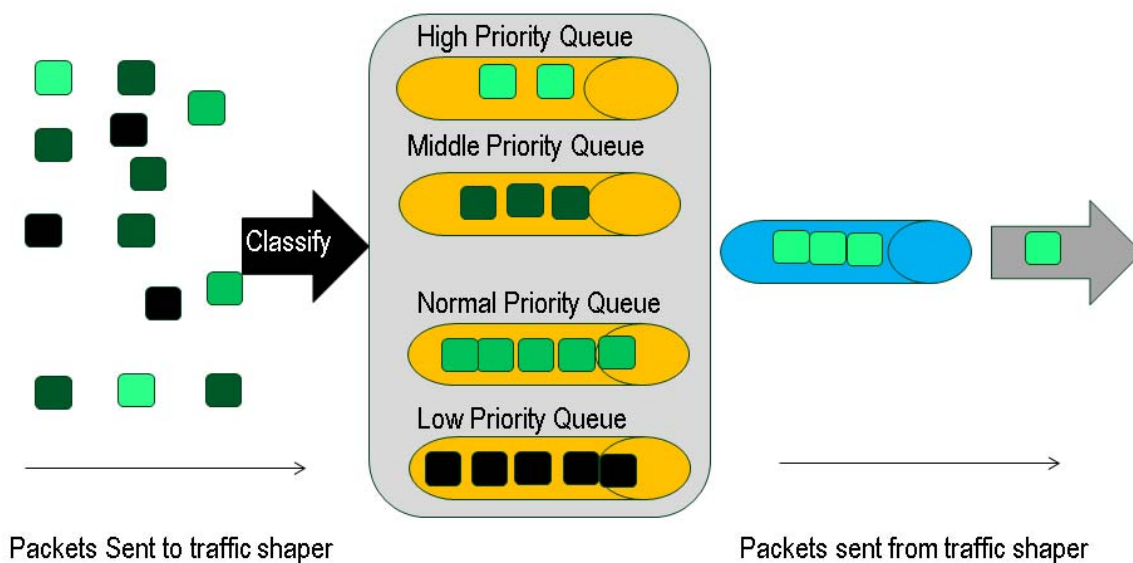
Figure 3. Priority Queuing packet processing.

Weighted fair queuing (WFQ). The weighted fair queue (WFQ), also referred to

as flow-based WFQ, implementation   manages a separate queue for each flow type that

enters the router. Packets with the same source IP address, destination IP address, source

or destination TCP port or UDP port are considered in a single flow. When the packets

are placed in the queue, they are stamped with an expected exit time based on a

complicated matching scheme. Unlike the previous methods, the queues are given

weights which define the maximum percentage of the bandwidth that can be used by that

particular queue (Kurose & Ross, 2010). As shown in Figure 4, the queues are served in a

round robin fashion going from queue to queue, choosing to send on the packets that have

the lowest exit timestamp and forwarding on the amount of traffic depending on the

weight of that particular queue. This is done to produce fairness of traffic handling.

The WFQ implementation has a larger impact on the base router performance

because of the complexity of the queuing mechanism. Therefore, it can cause slower

traffic rates during lower network demands than the previous methods (Semeria, 2001).
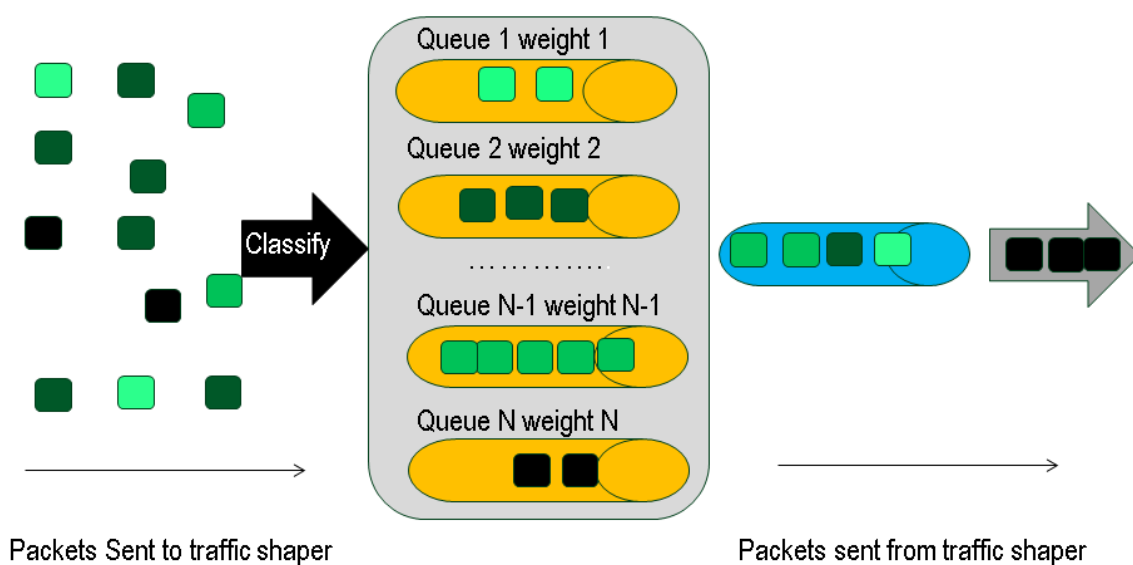
Figure 4. Weighted Fair Queuing packet processing.

Hierarchical token bucket (HTB). One of the most popular traffic shaping

algorithms is the hierarchical token bucket method. The basic premise of this method

sends all traffic flow through a process where the data is classified by type and prioritized

for transmission. The traffic all flows into what is considered a bucket (queue) that

matches the classification. The buckets are defined to have a continual flow of what are

called 'tokens' into the bucket which represent the packet rate at which the data should be

handled. The packet grabs an available token from the bucket before it can proceed. If there are no tokens available, the packet drops out of the bucket. In a hierarchical design, the packet may drop into a lower bucket which may or may not have tokens available. A visual representation of a single bucket process is shown in Figure 5.

Although this is among the most commonly used, it is complex to maintain and has the same problem in added overhead to the network as WFQ.

Packets Sent to traffic shaper   Packets sent from traffic shaper
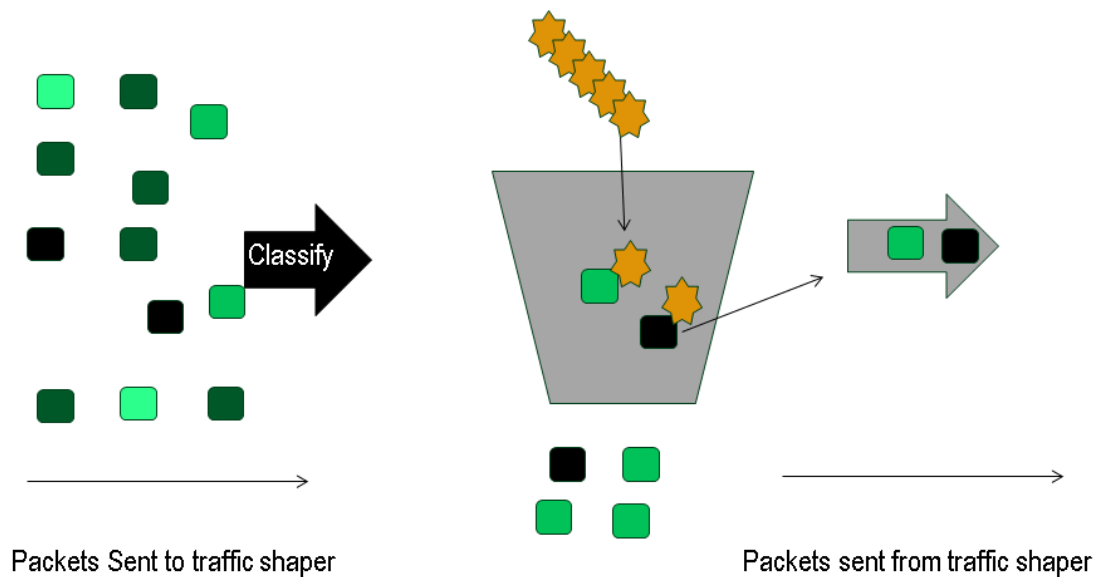
Figure 5. Hierarchical Token Bucket packet processing.

Baseline Queuing Comparisons

In order to develop a new algorithm, baseline comparisons of the standard QoS schemes were completed in the OPNET simulator. To define the baseline network, a small model was developed of the type of traffic that is used on the UNI campus. In order to create a realistic model, the Network Services staff were involved to obtain a sample of

the network traffic over a ten day period on the residence hall network. The traffic was gathered at a flow level to provide information on usage and types of traffic, no actual traffic data was captured. Representations of the data collected are provided in Appendix D. It was used as a baseline of the different types, sizes, and lengths of network transmissions experienced by students. The OPNET simulator was be configured to transmit similar baseline traffic to use different QoS algorithms for comparisons.

One of the most common routers used in the industry are made by Cisco®. The Cisco® routers include software to implement QoS using policing and shaping. These implementations allow limiting of traffic based on specific profile. A policy is used to drop traffic that matches a specific characteristic, such as a malicious attack or secured network location, whereas a traffic shaper delays traffic using a buffer or queuing scheme to shape the flow to meet a threshold. For example, a maximum value can be set for the rates inbound and/or outbound from the router. Cisco® routers provide three types of shaping: Generic Traffic Shaping (GTS), Distributed Traffic Shaping (DTS) and Frame Relay Traffic Shaping (FRTS). GTS uses a weighted fair queue to delay packets in order to shape the flow, and DTS and FRTS use either a weighted priority queue, a priority queue, a custom queue, or a FIFO queue, depending on how you configure the router (Cisco®, 2008) The FIFO, PQ and WFQ methods are all implemented in the FRTS traffic shaping mechanism and available in the OPNET Modeler environment.

Simulation Experimental Trials

The experimental trials followed by a procedure that involved creating a simulation of a small network with a single router and server. When reviewing the ResNet netflow data collected, the highest peaks in the network were from online video traffic, such as movies, YouTube or Skype communications. Other standard loads of traffic involved email and web browsing. So, the scenario was configured under varying amounts of video, web browsing, and email traffic over a period of 250 seconds. The simulation involved a separate scenario for each of the QoS schemes that were being compared. The simulation software used was OPNET Modeler. Three QoS schemes that were provided by OPNET as a base that were used for comparison were FIFO, PQ, and WFQ. The PQ scheme also had two options, one to prioritize based on port and another on ToS indicators, named as PQ port-based and PQ ToS respectively. Both PQ modes were used for comparison. The same simulation was run without any QoS scheme for comparison.

The simulation network consisted of a Cisco® 7000 router, four workstation nodes performing heavy web browsing traffic, four nodes performing light video conferencing, and two nodes doing light email traffic. A single server was placed in the scenario and configured to support all applications of the workstations. This single server was meant to represent all traffic leaving the subnet for external destinations. A diagram of the small network is shown in Figure 6. In order to simulation application traffic, data streams are configured as application profiles which are then assigned to the workstations in the simulation. The data streams were set to start up at uniform times in the simulation.
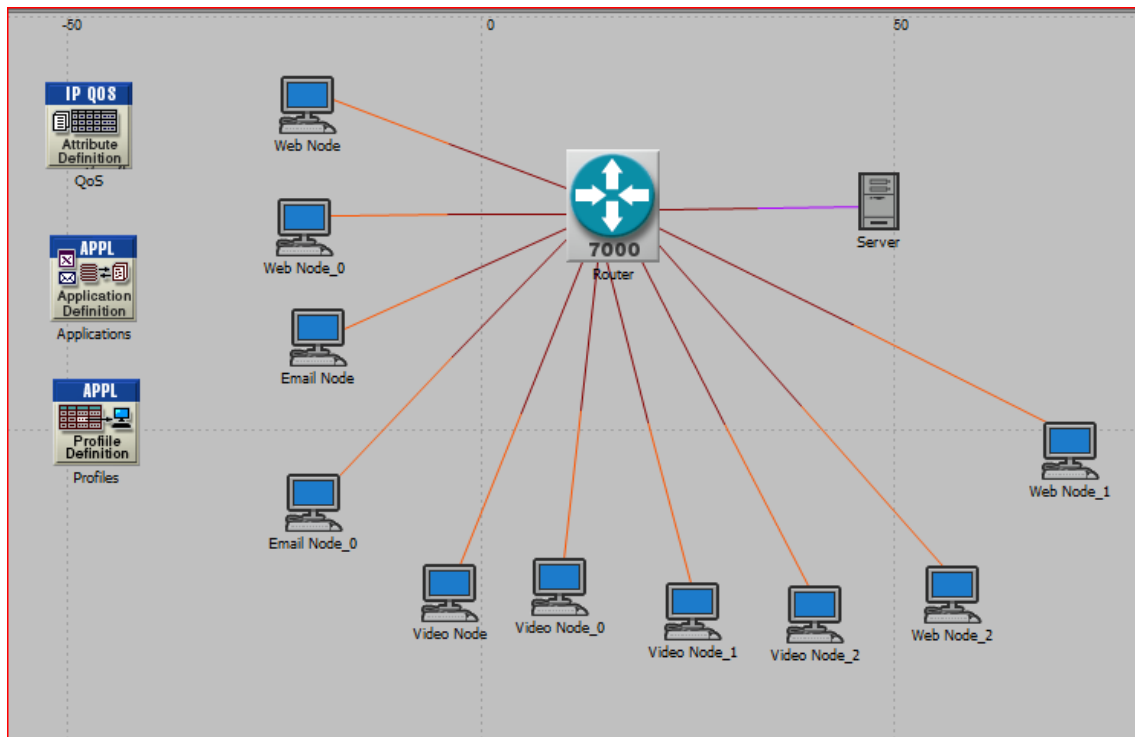
Figure 6. Baseline QoS network model.

The web traffic was set to start at between 5 and 30 seconds and continue until the end of the simulation. The http traffic was defined to have a constant page interval of .0055 seconds and object size of 1000 bytes. It was to represent 5 large images per page. This traffic is set with a Type of Service (ToS) setting of Standard (2) for QoS purposes.

The email traffic was set to start at 5 to 10 seconds into the simulation and continue until the end of the simulation. The send interarrival time was a constant .0055 seconds with receive interarrival time of .010 seconds. The group size on the email was set to a constant 3 with the email size set at 1000 bytes. The email traffic was set with a ToS of Best Effort (0).

The video conferencing traffic was set to start at 60 seconds into the simulation and continue until the end. The standard video conferencing settings were used on this data stream with frame interarrival time set at 10 frames/second with the frame size set at 128x120 pixels of All Discrete traffic. The ToS for this data stream was set at Interactive Voice (6) for QoS use. The ToS level is used by the classification logic to place items into queues, typically the higher ToS level has high priority. This simulation resulted in supporting three ToS levels as shown in Table 2.

Table 2. *Baseline simulation ToS applications and levels.*

Application traffic	ToS level	Packet size
Light email	0	1400 bits
Heavy web browsing	2	3120 bits
Light video conferencing	6	12000 bits

In initial trials of these schemes, the global statistic for Ethernet delay was gathered for comparison. The Ethernet delay statistic is collected in the simulation to represent the end-to-end delay of all packets received by all stations in the simulation. Due to the small amount of traffic, there was no noticeable difference in the Ethernet delay between the WFQ, FIFO, and PQ ToS simulation results causing the single graph line representing all three items in Figure 7. However, the PQ port-based scheme run did some a higher Ethernet delay. The simulation without any QoS showed the highest Ethernet delay. The other metric being reviewed was the throughput in packets/second. In comparing the results for this same simulation, the PQ port-based scheme provided

much higher transmission rates than FIFO or no QoS. Again, the other three schemes

produced identical results as can be seen as the middle line in Figure 8.
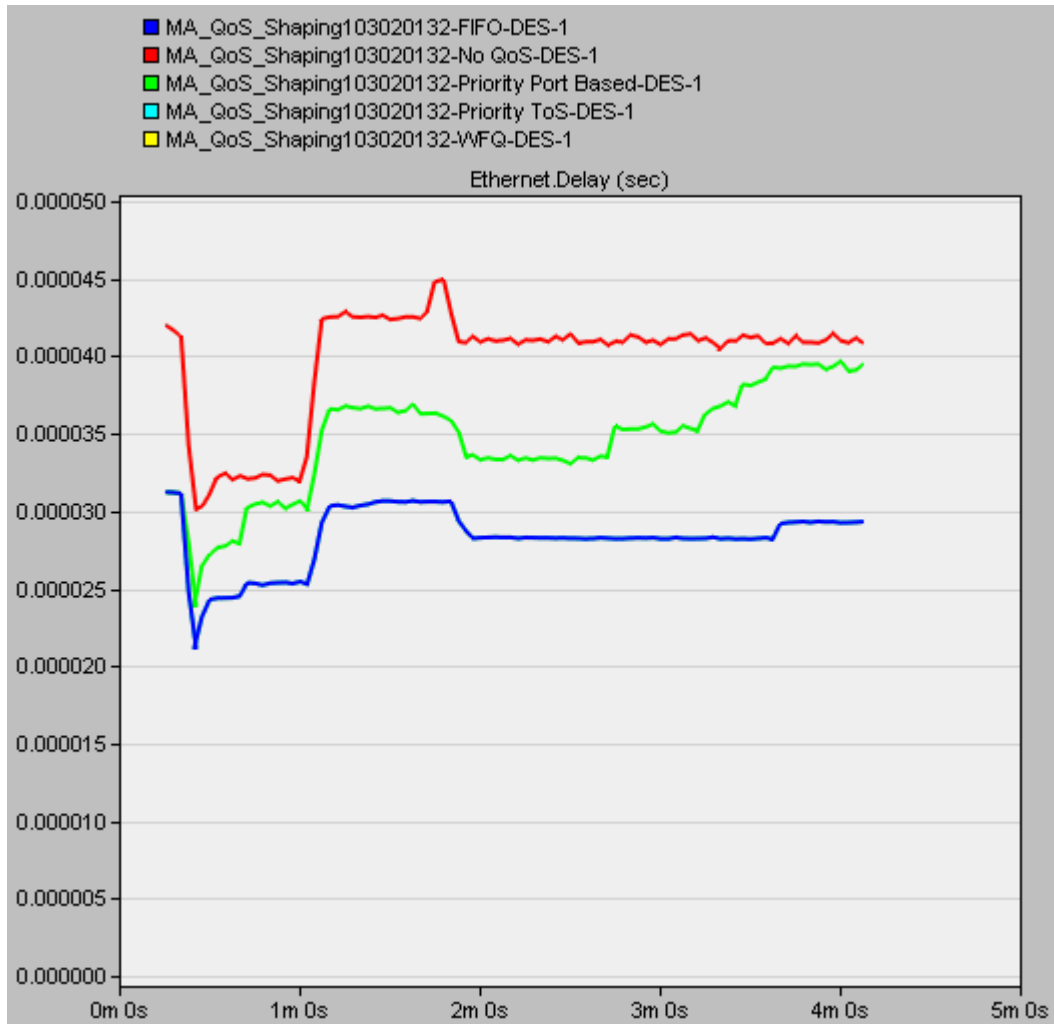
Figure 7. Baseline Ethernet Delay comparison for no QoS, FIFO, WFQ, PQ ToS, and PQ port-based.
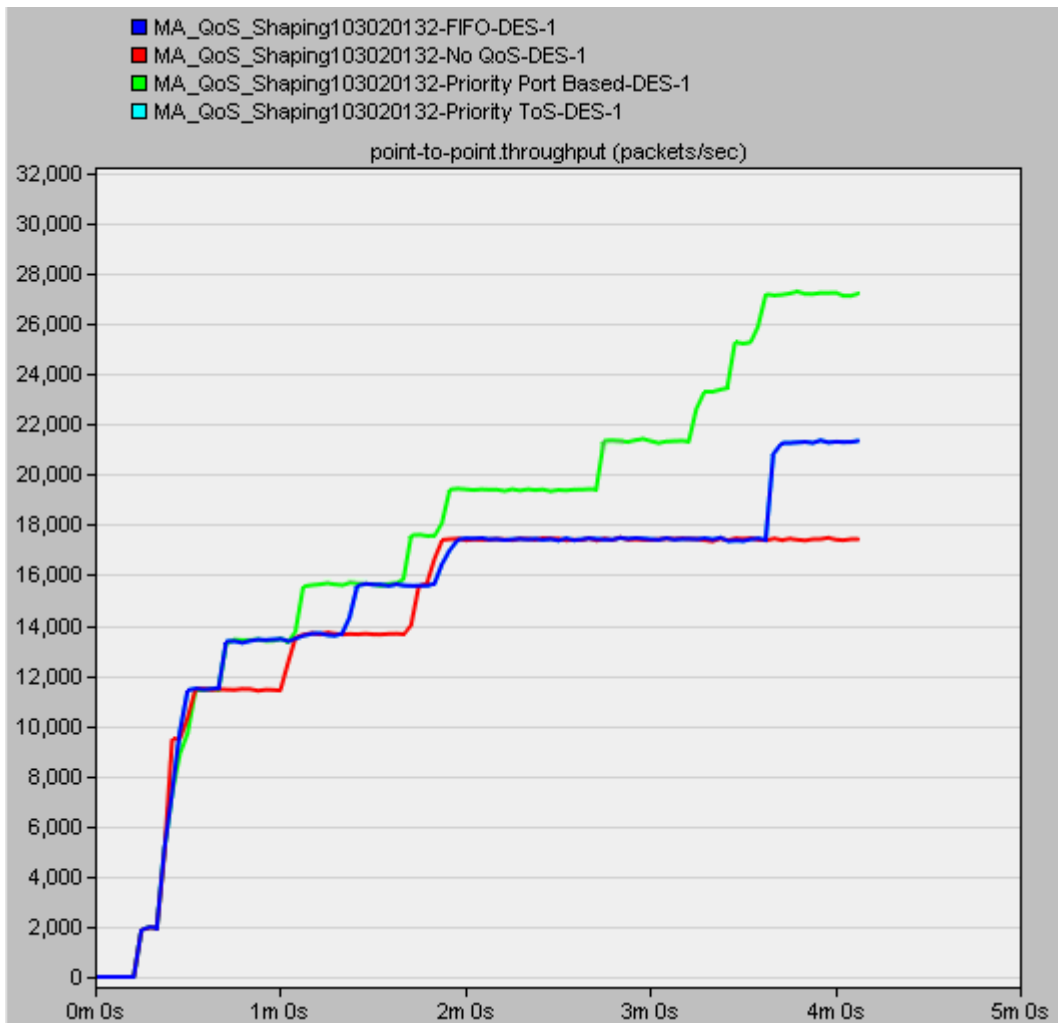
Figure 8. Baseline Throughput Comparison for no QoS, FIFO, WFQ, PQ ToS and PQ port-based.

In reviewing these results, with the queuing schemes, there appears to be is a trade-off between delay times and throughput. Due to the larger amount of traffic in the scenarios related to video conferencing which is rated as a higher priority, noticeable differences occur once the video traffic increases. As can be seen, the PQ port-based scenario was able to produce larger throughputs when the traffic increased from the video traffic. This can be attributed to the queue differences between PQ port-based and PQ

ToS. The traffic is distributed to the queues differently, whereas the PQ port-based

placed most of the traffic into queue 0, the PQ ToS placed the majority of the traffic into

queue 1. However, as shown in Figure 14, the overall delay for PQ port-based increases

because of the pile up of email while the higher priority queue is being processed. The

Ethernet delay for FIFO, WFQ and PQ ToS based simulations are almost identical

Preliminary Experiments in QoS Mode Switching

In looking at improvements to both of these metrics, there is a desire to decrease

the delay while increasing the bandwidth. Since the FIFO method had the lowest delay

and the PQ port-based had the highest throughput, these schemes were selected for more

testing. Further experiments were conducted on a PQ port-based scenario where the

system would to force a change in the queuing to switch to FIFO mode when high traffic

is introduced. In this simulation, the traffic is configured to increase quickly after 45

seconds into the scenario, accordingly this switching was implemented by setting a timer

so that after 45 seconds, all traffic was placed into queue 3, which was the highest

priority queue. As can be seen from the Figures 9 and 10 below, the baseline priority

queuing scheme is placing the traffic into 3 different queues, 0, 1 and 3. In the

experimental change, when the traffic load increases, at 45 seconds, all packets are placed

in queue 3.

The simulation results indicated that the change allows the router to increase the

bits/second transferred to the server and also the bits/second from the server back to the

router as shown in Figure 11. However, Figure 12 shows the overall Ethernet delay does

increase compared to using standard PQ port-based QoS.
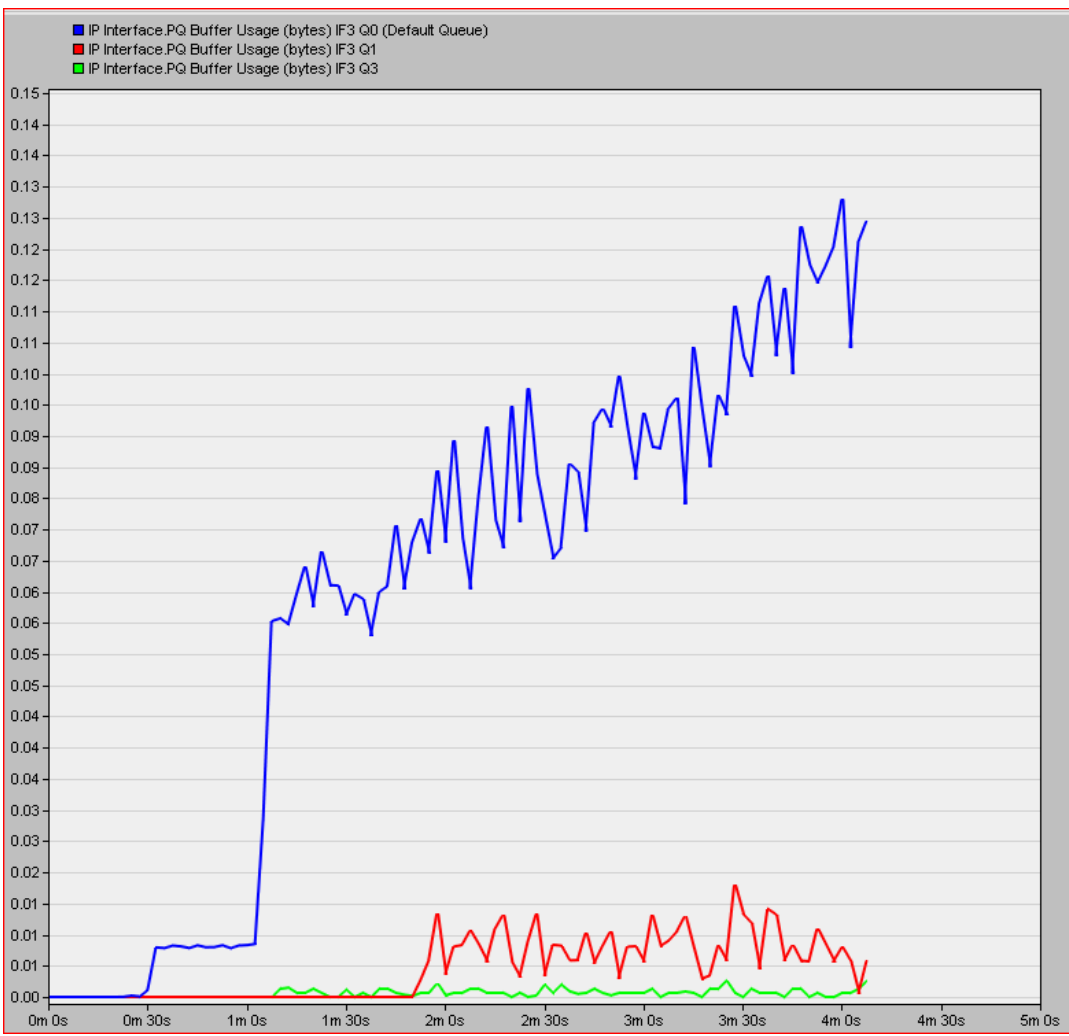
Figure 9. PQ port-based buffer usage by queue in bytes during baseline simulation.
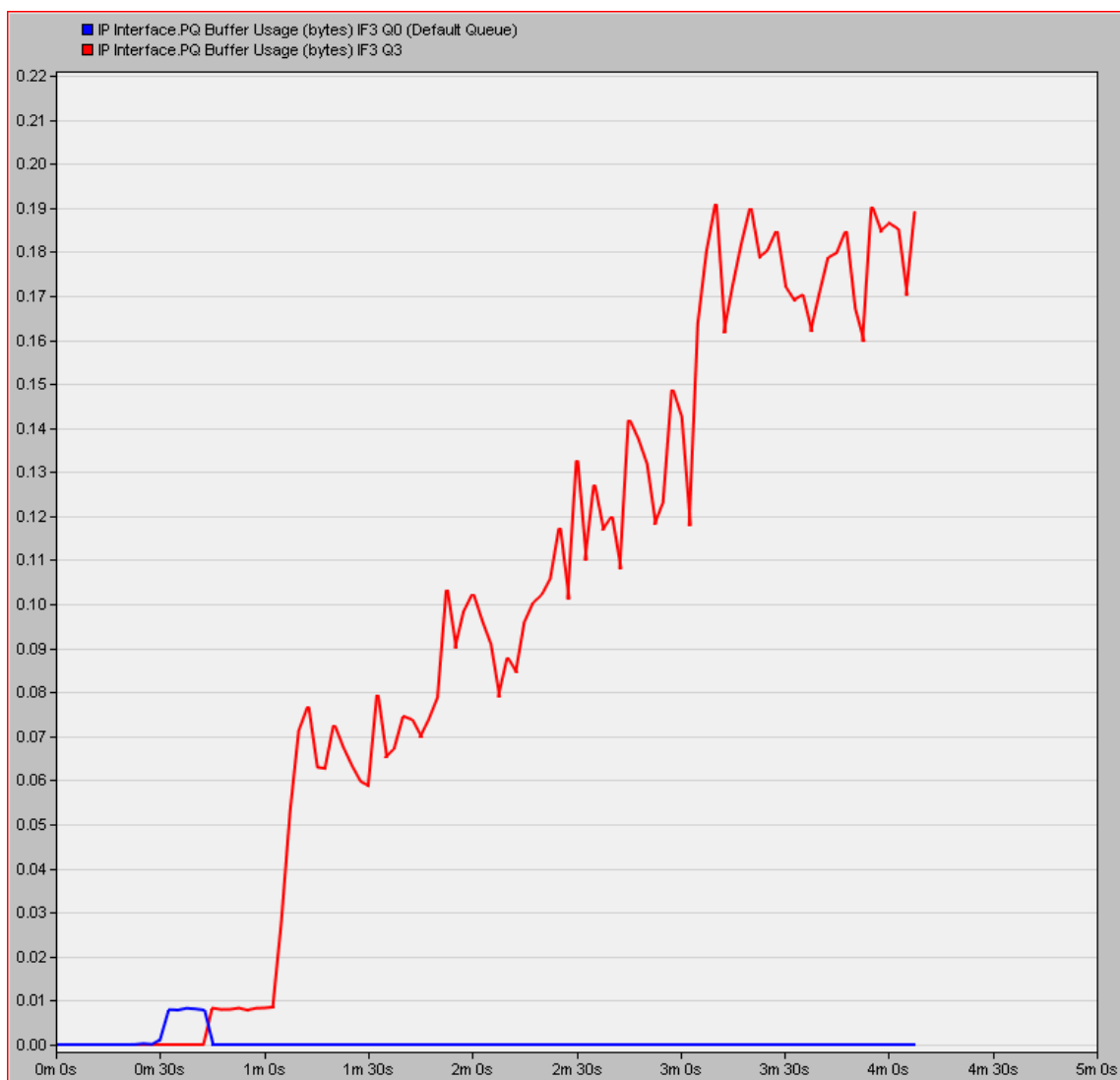
Figure 10. PQ port-based buffer usage by queue in bytes during experimental simulation forcing traffic into queue 3.
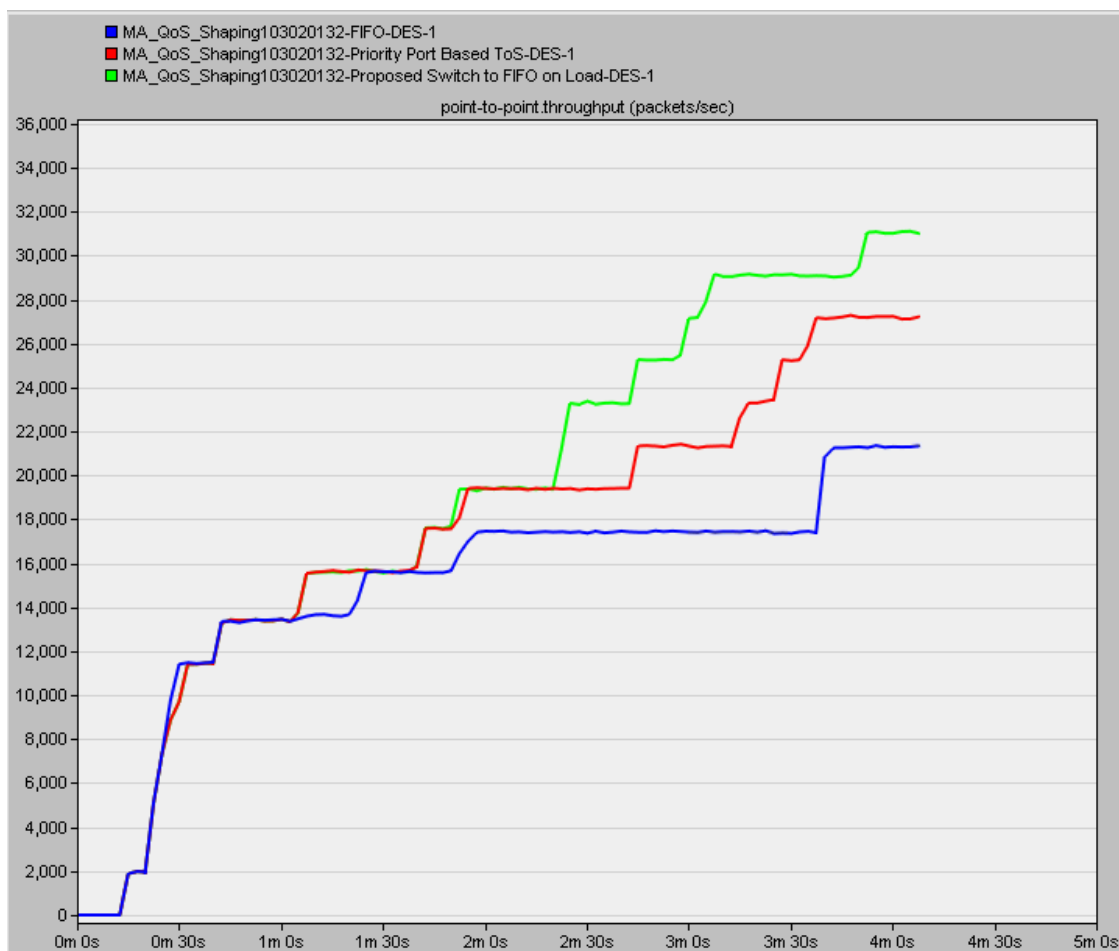
Figure 11. Comparison of packets/second sent from the router to the server including the new proposed priority queuing.
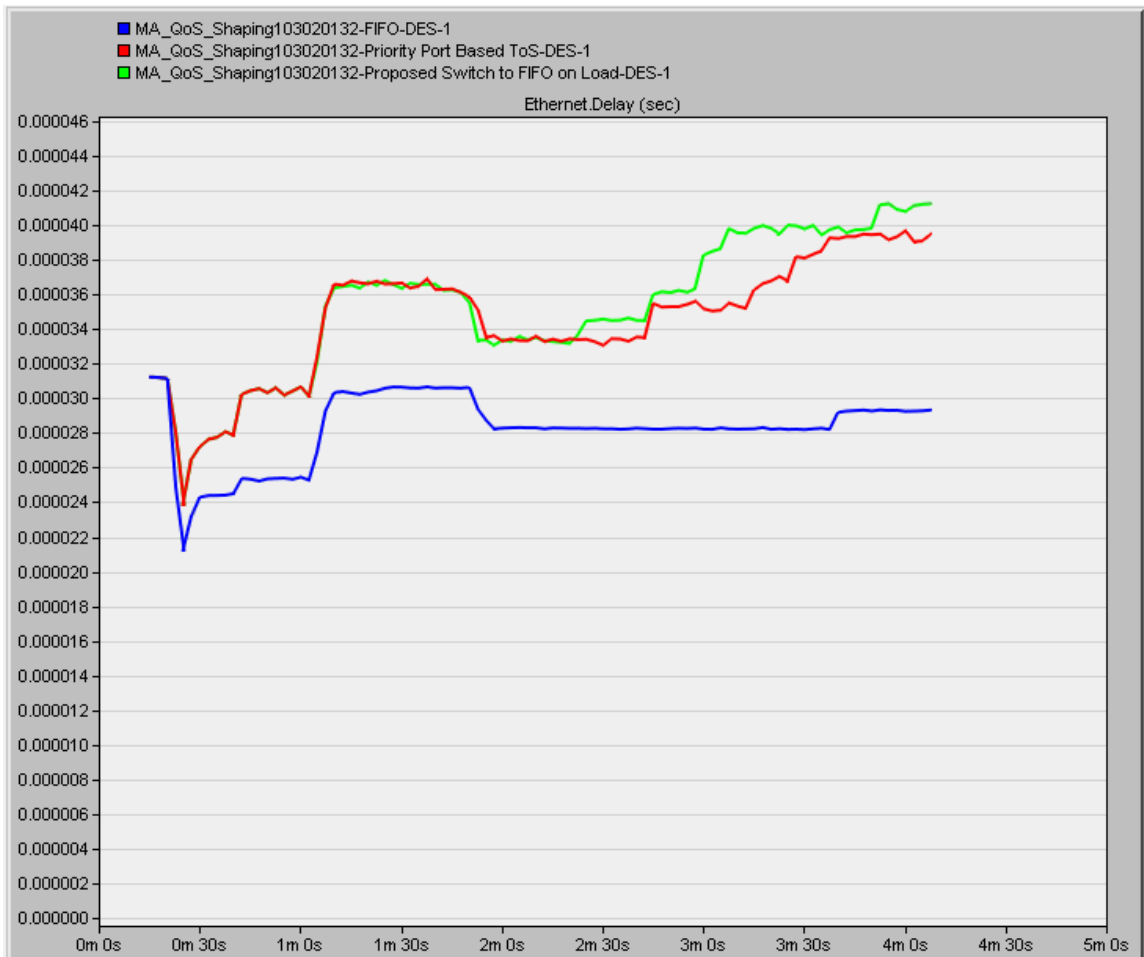
Figure 12. Comparison of Ethernet delay including proposed changed scheme.

In this network simulation, when the network traffic includes a lot of video traffic,

the QoS schemes behave differently resulting in differing throughput and delays. In the

priority queuing model, the video traffic is placed in the highest queue and processed

before all other queues which include the web and email packets. The amount of traffic

being serviced is higher, but the overall delay of all traffic is also higher because the

lower priority email and web traffic are left unprocessed as the video streams take over

the bandwidth. However, in the proposed change, if all packets are placed into the highest

queue starting when the load increases at 45 seconds, then the mixture of packets are processed similar to FIFO.

Another set of experiments were completed with a different queue sizes to find other differences in three of the baseline scenarios. Since this study was focused on improving delays, the PQ ToS method was chosen over the PQ port-based method going forward. The same scenario and traffic pattern were used, however, the queue sizes were adjusted to make the queuing schemes more equivalent for comparison queue sizes. Since the ToS for the three types of traffic fall into 0, 2, and 6, the queues were setup as shown in Table 3. Max packets is defined as the maximum number of packets that can be stored in that queue before the packets are dropped.

Table 3. *QoS scheme queue settings for baseline comparison.*

QoS Scheme	Queue Number	ToS handled	Max packets	Weight
FIFO	0	0-7	10	N/A
PQ ToS (a total of 4	0	0,1	5	N/A
queues)	1	2,3	3	N/A
	2	4,5	2	N/A
	3	6,7	2	N/A
WFQ (a total of 8	1.0	0	5	1.0
queues)	10	1	4	10
	20	2	3	20
	30	3	2	30
	40	4	2	40
	50	5	2	50
	60	6	2	60
	70	7	1	70

The listed weights are only used on the WFQ queues and represent the maximum percentage of the bandwidth that the queue can consume at any point in time. The weight is used to determine how many packets to release from the WFQ queue when it is selected in the round-robin method. The same traffic defined in Table 2 was run again for the FIFO, PQ ToS, and WFQ scenarios which resulted in differences in the Ethernet delay with WFQ performing the best when the video traffic was introduced into the scenario as shown in Figure 13.
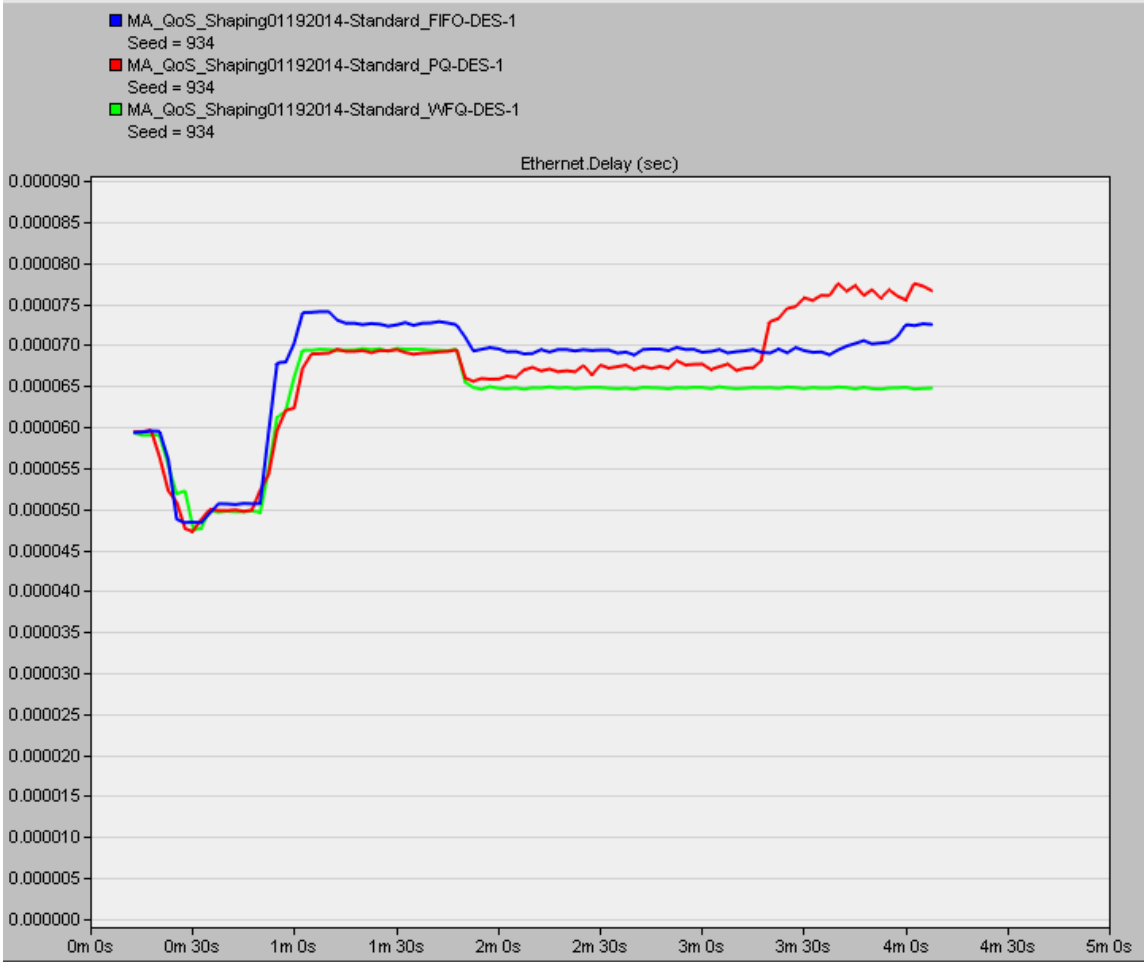
Figure 13. Baseline comparison of Ethernet delay with similar queue sizes.

However, when the throughput was evaluated, the WFQ performed lowest and the

FIFO had the highest. With the queue sizes adjusted, each scheme had a max of 10

packets that could be in a queue at any point in time. So when the higher priority traffic

started to increase for PQ and WFQ, more of the packets would be dropped because of

full queues. This can be seen by looking at the packets dropped/second statistics captured

in Figure 14.
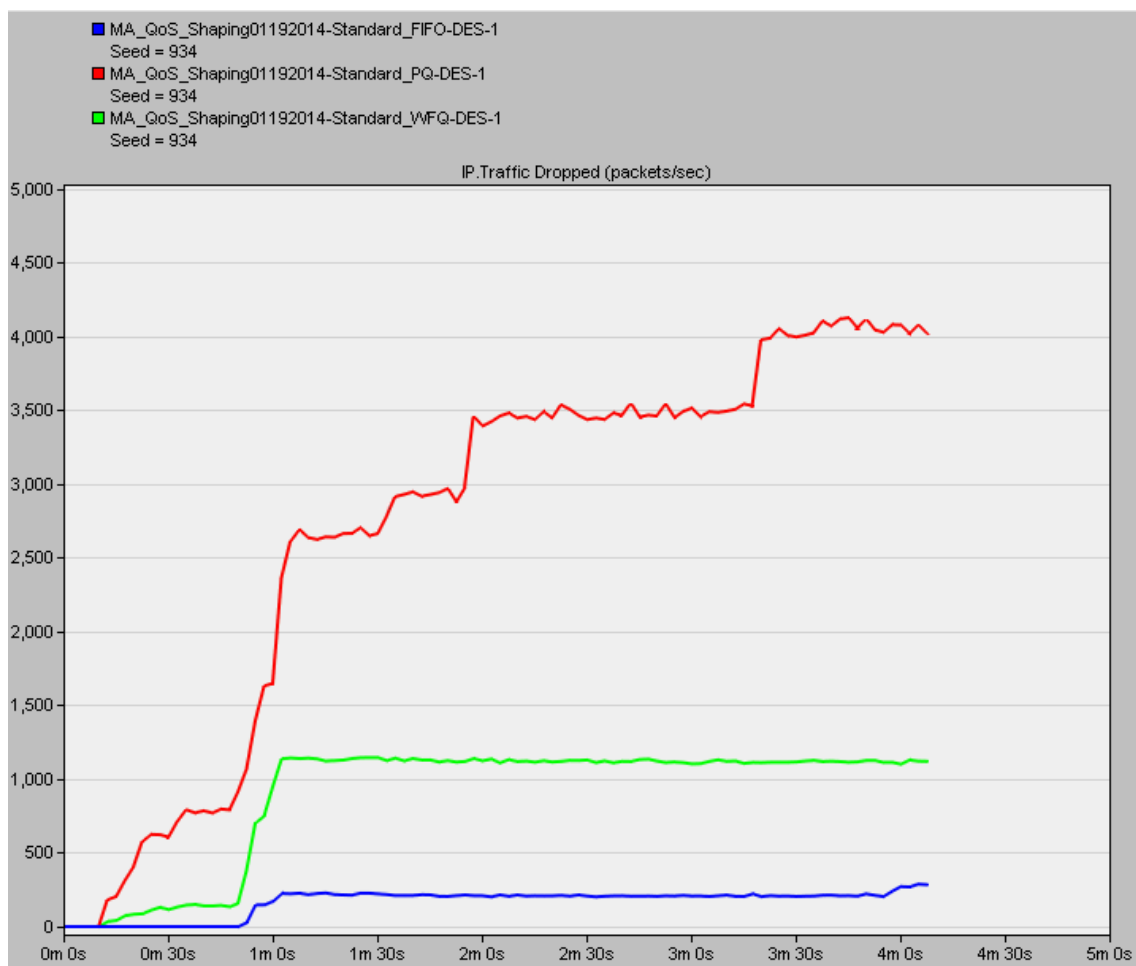
Figure 14. Baseline comparison packets dropped/second.

In summary, the results from the experiments showed that some performance increases could be made if the router could dynamically switch the queuing scheme to PQ when there is a medium amount of traffic to reduce the delay. Also when there begins to be a lot of packet drops, it could be beneficial to switch to WFQ or FIFO depending on the amount of traffic on the network.

<u>Proposed Multi-Mode Self-Adaptive Algorithm</u>

Due to the advantages and disadvantages of each of the existing algorithms, developing an approach that was dynamic depending on the condition of the network traffic at a point was proposed to provide better QoS. The proposed algorithm, named the Multi-Mode Self-Adaptive (MMSA) algorithm, would cause the router to change working modes based on the real-time network status. Since the queuing system must be flexible enough to switch modes within the same infrastructure, the three schemes of FIFO, PQ and WFQ were utilized and modified for the proposed MMSA algorithm. The initial queue structure developed supports the N queue structure of WFQ, with queue N set as the highest priority queue for the PQ mode and an associated weight to use when in the WFQ mode. This was necessary since the WFQ structure is the most complicated and allows the network to employ all the functions of all three schemes in one structure. The proposed algorithm switches among these three modes according to the measurements of the network throughput and packet drop rate in the run. The details of the three modes are given in the following sections.

Multi-Mode Self-Adaptive FIFO Mode

In the Multi-Mode Self-Adaptive FIFO Mode (MMSA_FIFO), all traffic is placed

into queue 1 similar to standard FIFO. However, on dequeuing, the queues are checked

for packets in the highest queue first, then second highest, etc. This is done to flush out

all packets that were in the system prior to switching to the MMSA_FIFO to avoid

leaving packets in the queues 2 through N from being delivered. In the example shown in

Figure 15, the next packet to leave the queues would be the packet in the top queue N.

This would removing all previous prioritized packets and continue with those in queue 1.

Priority N Queue N weight N

Priority N-1 Queue N-1 weight N-1

Priority 2 Queue 2 weight 2

Priority 1 Queue 1 weight 1

Packets Sent to traffic shaper

Packets sent from traffic shaper
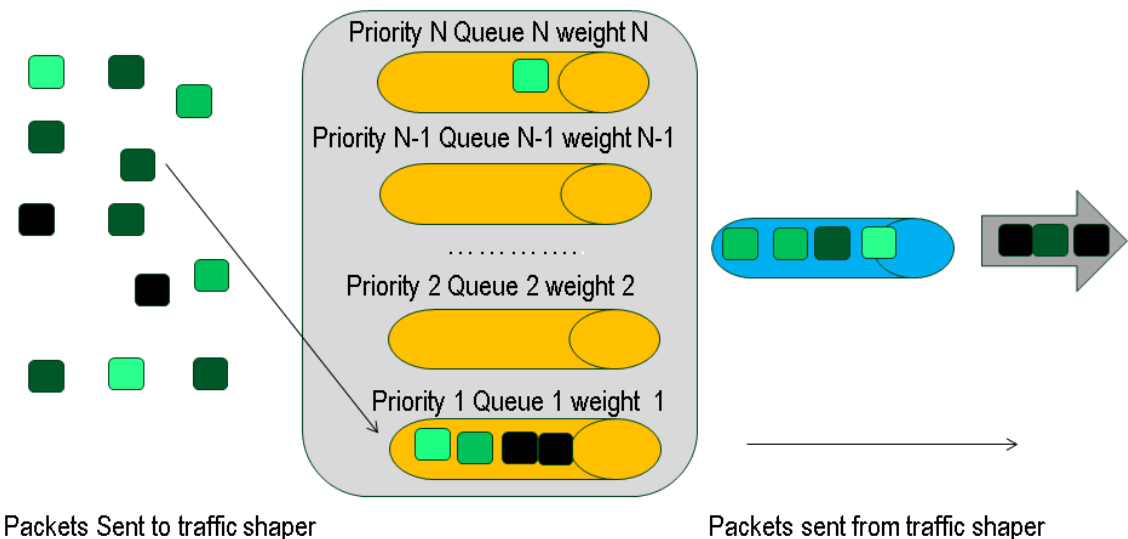
Figure 15. Multi-Mode Self-Adaptive First In First Out packet processing.

Multi-Mode Self-Adaptive PQ Mode

In Multi-Mode Self-Adaptive PQ Mode (MMSA_PQ), the packets are classified and placed into the queues using the same match algorithm as the standard WFQ mode in order to ensure that exit timestamps are created and placed on the packets. When the matching criteria was for the ToS on the packet, the flows considered highest priority were placed into the corresponding queue N. When the packets are dequeued, the process is the same as in standard PQ, starting with queue N. If there are packets they are sent on, going down to queue 1 only after all other queues are empty.
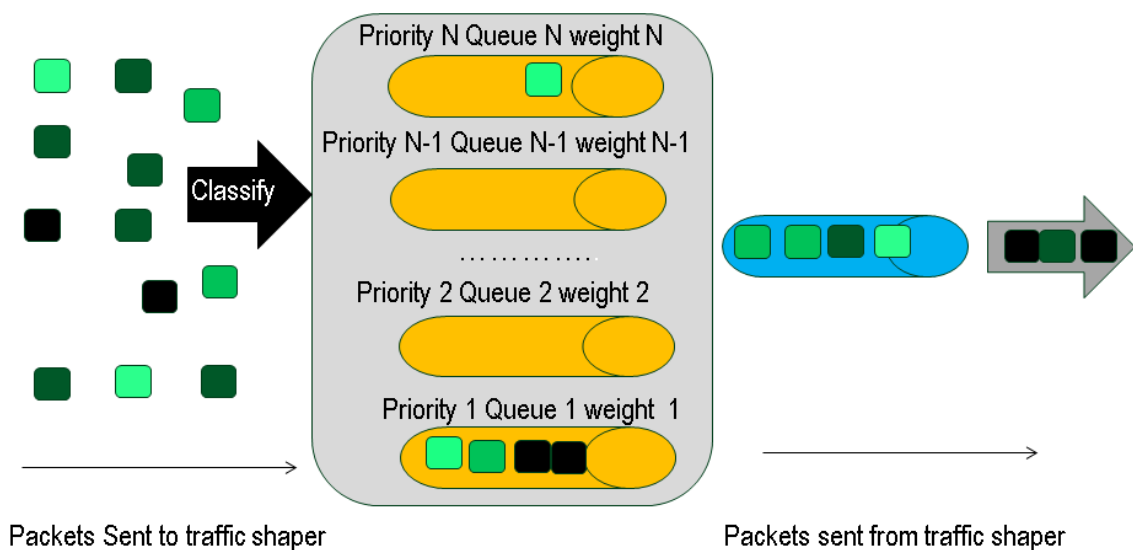
Figure 16. Multi-Mode Self-Adaptive Priority Queuing packet processing.

Multi-Mode Self-Adaptive WFQ Mode

The Multi-Mode Self-Adaptive WFQ mode (MMSA_WFQ) queues and dequeues in the normal WFQ scheme. Flows are classified and placed into the queues 1 through N

depending on defined match criteria. On dequeuing, the queues are accessed as a standard

WFQ scheme. In the example shown in Figure 17, the queues would be cycled through

from queue 1 to queue N allowing packets to leave the queue based on the exit

timestamps on each packet in groups determined by the weight on the queue.
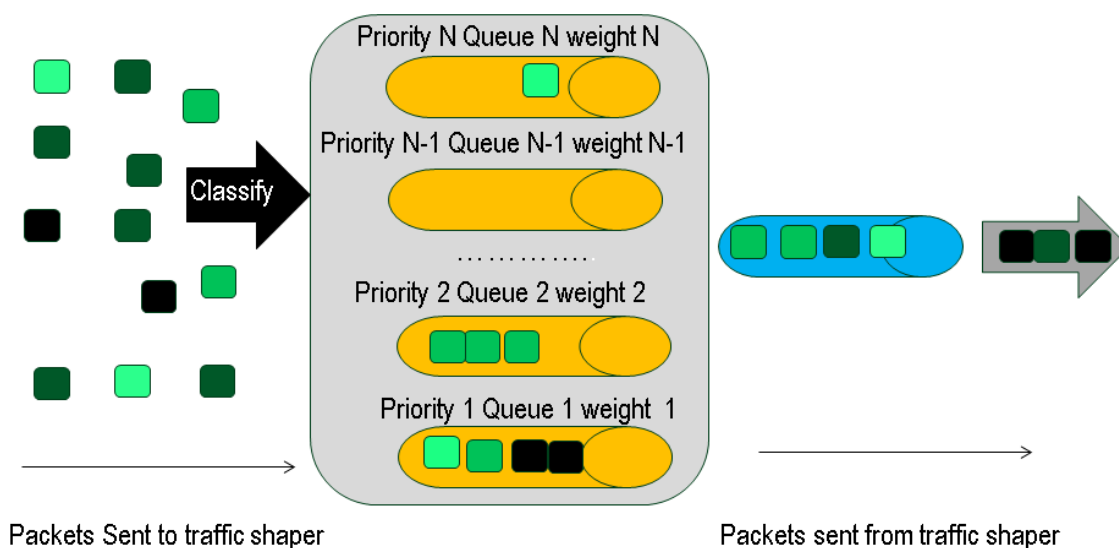
Figure 17. Multi-Mode Self-Adaptive Weighed Fair Queuing packet processing.

Mode Switch Algorithm

In order to verify the MMSA algorithm, some form of measurements had to be

selected for the thresholds on which to switch the self-adaptive algorithm between the

three modes. When in FIFO mode as shown from the baseline results in Figure 15,

Ethernet delays are generally lower, so when traffic is not heavy, it is beneficial to stay in

FIFO mode to reduce delays introduced due to complicated queuing classification.

However, once traffic reaches a certain level, using a PQ method will assist in increasing

throughput with fewer dropped packets. However, in heavy traffic scenarios, PQ tends to

force dropped packets in lower queues causing overall longer TCP segment delays. In looking at the advantages and disadvantages of these three methods the algorithm was setup as follows

if (P/S) < PT then

Set to MMSA_FIFO

else if (P/S) >PT and DP < DT then

Set to MMSA_PQ

else if (P/S) > PT and DP > DT then

Set to MMSA_WFQ

end if;

Where:

P = Packets on the interface over the time period S

S = Periodical measurement interval (seconds)

PT = Threshold of packets transmit rate for the interface considered low usage (packets/second)

DP = Dropped packets on the interface over the time period of S

DT = Threshold of packets dropped over S seconds

The algorithm is evaluated every S seconds over to determine if the mode should be switched. Packets per second (P/S) are used to determine when the network interface is experiencing low or heavy traffic based on the threshold (PT) numbers provided. When the system is experiencing a low amount of traffic, the MMSA_FIFO mode is employed. If the network is considered in a heavy load situation, then the algorithm

compares the number of packets dropped since the last time check interval (S) and if it is

under the threshold (DT), the QoS scheme is changed to use the MMSA_PQ settings.

However, if it is over the threshold set (DT), the QoS scheme is changed to the

MMSA_WFQ mode. The flow of the process is shown in Figure 18.
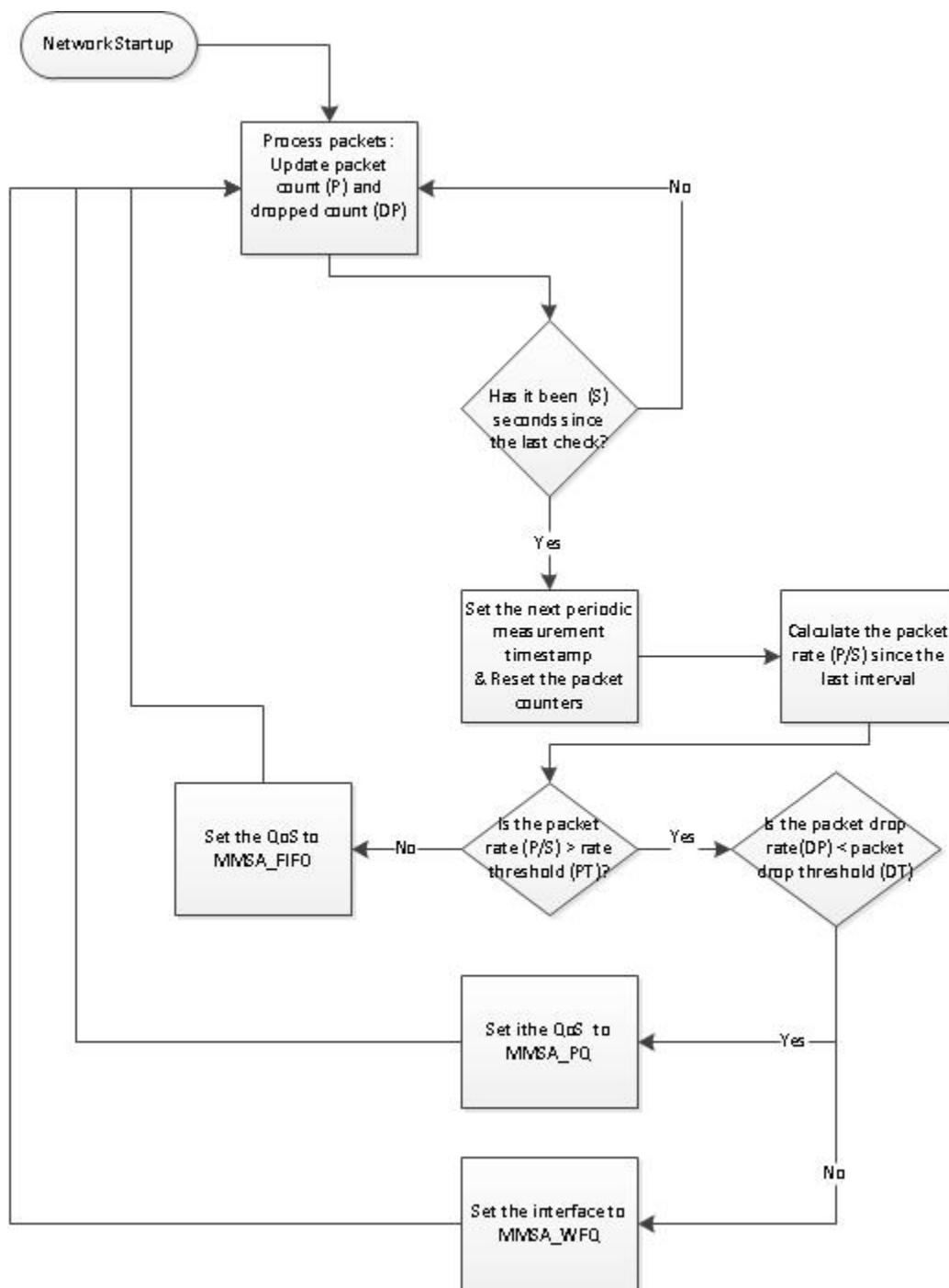
Figure 18. MMSA Algorithm flowchart.

CHAPTER 4

ALGORITHM IMPLEMENTATION AND PERFORMANCE ANALYSIS

Implementation of Proposed Algorithm in OPNET

The OPNET software provides the source code to the modeler in a form that is compiled with C++. This allows customization of the software to implement new functionality. In order to be able to switch modes for the proposed algorithm and also support the existing QoS schemes, the software needs to have the ability to turn the MMSA on and off. It also needs to be able to check the mode to determine if the standard code or custom code should be used in the queuing and dequeuing processes. The steps were performed in a bottom up design as listed in Table 4.

The structure for the queue management in OPNET is defined in the oms_qm.h file in the structure OmsT_Qm_Info. To complete step 1, a new attribute was added to that structure called mmsa_mode of type MMSA_Queuing_Scheme. The MMSA_Queuing_Scheme was enumerated with the following values:

MMSA_Standard_Queuing = -1

MMSA_FIFO_Queuing = 0

MMSA_WFQ_Queuing = 1

MMSA_Priority_Queuing = 2

Table 4. *Steps to OPNET modifications for MMSA algorithm.*

Step	Description of modification	Location of modification	Notes
1	Modify queuing structure-add new mode attribute	C:\OPNET\16.1.A\models\std\include\oms_qm.h	External file compiled into oms_qm_ex.c
2	Modify the queue initialization process - set the new attribute	C:\OPNET\16.1.A\models\std\utilities\oms\oms_qm_ex.c Function Oms_Qm_Info_Create()	External file. Called from the ip_output_iface process model
3	Modify the queuing process for MMSA_FIFO to always use queue 0	C:\OPNET\16.1.A\models\std\utilities\oms\oms_qm_ex.c Function Oms_Qm_Packet_Process()	External file. Called from the ip_output_iface process model
4	Modify the dequeuing process	C:\OPNET\16.1.A\models\std\utilities\oms\oms_qm_ex.c Function Oms_Qm_Wfq_Queue_Select()	External file. Called from the ip_output_iface process model
5	Add counter for dropped packets	Process model ip_output_iface function block. Function enqueue_packet()	Called from the process model ip_dispatch which is called from the Node model for the router
6	Set MMSA attribute startup values	Process model ip_output_iface function block. Functions do_init()	Called from the process model ip_dispatch which is called from the Node model for the router
7	Add Self-Adaptive code	Process model ip_output_iface function block. Function extract_and_send()	Called from the process model ip_dispatch which is called from the Node model for the router

In the process that initializes the queue management structures, the value mmsa_mode is set to MMSA_Standard_Queuing on startup. This setting represents using whatever standard QoS scheme was selected in the OPNET Modeler for the scenario being run. The queue structure is created for each network interface that has been selected for QoS management. The process model for the router calls the external process Oms_Qm_Info_Create to initialize the queues for the interface based on the attributes, such as maximum buffered packets. In all simulations, the QoS scheme has been configured on all interfaces that are directly connected to the central router. Therefore, each interface could be in a different mode depending on the algorithm's application to that particular interface.

Next changes were made to the queuing and dequeuing procedures to use the proposed algorithm. The enqueuing process was modified for step 2 by editing the code in the location listed in Table 4 so that if the current mode was MMSA_FIFO_Queuing, the packet would always be placed into the lowest level queue. The process used for queuing the packets was called Oms_Qm_Packet_Process which uses code to determine based on the QoS mode which queue should be used for the packet passed into the procedure. The only change to the process was that when in MMSA_FIFO_Queuing mode all packets are forced into queue 0 to bypass any complex logic that would be used in the classification code as shown:

```
if (qm_info -> mmsa_mode == MMSA_FIFO_Queuing)
        {
        enq_id = 0;
        }
```

The enqueuing process for MMSA_PQ_Queuing and MMSA_WFQ_Queuing used the classification procedure for the provided WFQ scheme. The classification procedure determines the queue depending on the ToS information on the packet. It also assigns the timestamps necessary for dequeuing in the MMSA_WFQ_Queuing method so those attributes are available when running in MMSA_WFQ_Queuing mode.

The dequeuing process contains separate functions for each method. Since the simulation must start in the standard WFQ structure to function, the process model calls the function Oms_Qm_Wfq_Queue_Select. The code was modified so that if the simulation was running in MMSA_FIFO_Queuing, the traffic is processed cycling from the highest priority down the lower priority queues which is the same method as MMSA_PQ_Queuing. Since there are most likely still packets sitting in the higher level queues when switching from MMSA_PQ_Queuing or MMSA_WFQ_Queuing to MMSA_FIFO_Queuing, this change will prevent packets that were in queues higher than 0 from being abandoned. The code was changed in the Oms_Qm_Next_Queue_Select code so that depending on the mode, the queue would be selected in either of two modes, by priority or by weight. When in MMSA_FIFO_Queuing or MMSA_PQ_Queuing, the queue is selected in priority mode as explained previously. The MMSA_WFQ_Queuing mode used the normal WFQ dequeue method of selecting the queue with the next lowest timestamp and taking however many packets allowed according to the assigned weight. The code for this change is provided in Figure 19.

```
FIN (Oms_Qm_Next_Queue_Select (qm_info));
        /* Get the number of queues.*/
        i = qpool_ptr->attributes_ptr->no_queues;
                switch (qm_info->mmsa_mode)
                {
                case MMSA_FIFO_Queuing:
                case MMSA_Priority_Queuing:
        /* Loop through the priorities from the high priority queue to the low        */
        /* priority queue until we find a non-empty queue.
        */
                do
                {
                i --;
                queue_id = i;
                }
                while ((i > 0) && oms_qm_container_is_empty (qpool_ptr, queue_id));
                break;
        case MMSA_WFQ_Queuing:
                /* Find out the minimum finish time among all non-empty subqueues.*/
                for (queue_id = 0;
                        queue_id < qpool_ptr->attributes_ptr->no_queues; queue_id ++)
                                {
                                qinfo_ptr = qpool_ptr->queue_info_parray[queue_id];
                wfq_vars_ptr    = (OmsT_Qm_WFQ_Vars*)
                                qinfo_ptr->qscheme_vars_ptr;
                                num_packets = oms_buffer_num_packets_get (qinfo_ptr-buffer_handle);

                                if (num_packets > 0.0)
                                        {
                                /* Check if the finish time has not been already been calculated*/
                                /* for this queue. Otherwise compute it again.                */
                if (wfq_vars_ptr->finish_time ==   OMSC_UNCOMPUTED)
                                        {
                                        wfq_vars_ptr->finish_time =
                                        oms_qm_finish_time_compute(queue_id, qm_info, qpool_ptr);
                                        }
                                /* Compare the finish time of the other packets and take the        */
                                /* smallest one among all of them.
                                */
                                        if ((minimum_finish_time == -1.0) ||
                                                (wfq_vars_ptr->finish_time < minimum_finish_time))
                                                {
                                                minimum_finish_time = wfq_vars_ptr->finish_time;
                                                minimum_finish_time_queue_index = queue_id;
                                                }
                                        }
                                }
                                queue_id = minimum_finish_time_queue_index;
                        break;
                default:
                        break;
                }
        /* Return the queue which has to be served.      */
        FRET (queue_id);
        }
```

Figure 19. Dequeue modifications for MMSA algorithm.

The changes required for steps 1 through 4 required changing the external C++ program in the file oms_qm.ex. The changes were compiled outside of the OPNET software using Microsoft Visual C++ 2010 Express®.

The next change to complete step 5 required adding code to keep track of the dropped packets for use later in the self-adaptive algorithm. The code was modified in the process model ip_output_iface function block in the function enqueue_packet. The code already had a special section to check if the packet was dropped so a counter was incremented in that section. The code was modified as shown in Figure 20.

```
if (qm_signal == OmsC_Qm_Dropped_Packet)

            {

            /* In case of a congestion the packet is dropped. Statistics      */

            /* are updated and a simulation notification is written to         */

            /* warn that some packets might be dropped.                        */

            /* Print a notification log message to tell                        */

            /* that a packet has been dropped for congestion         */

            /* reasons. It is displayed only once.                            */

            mmsa_pkt_dropped = mmsa_pkt_dropped +1;

... continued
```

Figure 20. Code to increment dropped packet counter.

The most important piece to implementing this algorithm was attaching it to the model, which was completed by modifying the ip process model associated with the node model used for the Cisco® router. The node model associated with the router contains an ip process that calls the ip_dispatcher process model shown in Figure 21. The ip_dispatcher process code calls the ip_output_iface process model shown in Figure 22.
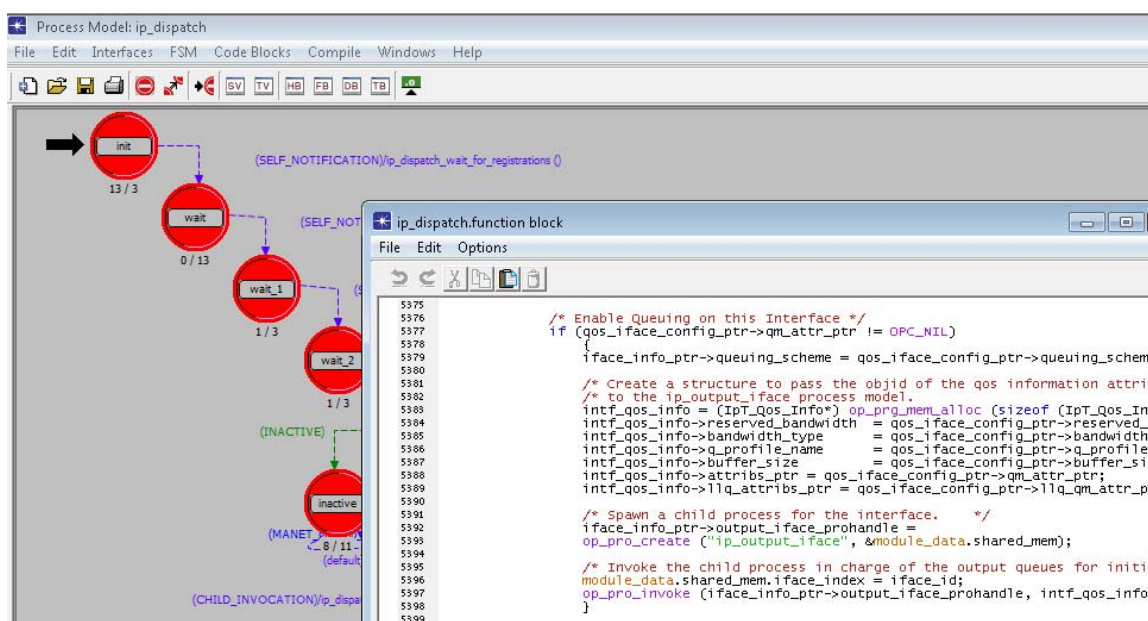
Figure 21. OPNET process model for ip_dispatcher.

Lastly, the self-adaptive code was implemented at the ip_output_iface process mode. In order to accomplish this, the process model that is used for QoS need to be modified. This model is used for setting up queues and managing all of the traffic when QoS is enabled on the model. Every process has a structure for the code to create and store variables, define the code that the process runs, etc. For this modification, there were a few state variables that were added to keep track of the custom thresholds and

counters. Figure 23 shows how these variables were placed in the state variables section

of the process.

This child process is spawned for each IP interface in order to model a FIFO, WFQ, priority queuing or custom queuing mechanism.

Congestion avoidance mechanisms like RED/WRED (Weighted Random Early Detection) are also handled by this process.

This child process is also RSVP-aware. Queue configuration can be managed upon RSVP requests.

(RECEIVE_PACKET)/enqueue_packet ()

(CREATE_RSVP_QUEUE)/create_rsvp_queue ()

(MODIFY_RSVP_QUEUE_PARAMS)/modify_rsvp_queue ()

(DELETE_RSVP_QUEUE)/delete_rsvp_queue ()

(ADD_RSVP_SENDER)/add_rsvp_sender ()

(DELETE_RSVP_SENDER)/delete_rsvp_sender ()

Figure 22. OPNET process model ip_output_iface for QoS code.

int	mmsa_pkt_counter	/* This is the total packets processed to use for aver...
int	mmsa_set_checktime	/* This is a time to be set when the next check shoul...
int	mmsa_interval_secs	/* This is the interval used in seconds to determine h...
Boolean	mmsa_self_anneal	/* This is a flag to determine to use the self-annealing...
int	mmsa_pkt_dropped	/* This is a counter of dropped packets */
int	mmsa_pkt_sec_threshold	/* This is the threshold for packets/second before s...
int	mmsa_pkt_dropped_threshold	/* This is the threshold for packets dropped in an inte...

Figure 23. OPNET ip_output_iface custom state variables.

Code then was placed into the procedure of forwarding packets to check the

thresholds and change the mode if necessary. An overall flag called mmsa_self_anneal

was created as a Boolean that is set when the scenario is run to either OPC_TRUE or

OPC_FALSE to use the MMSA code. If the flag is true, then the code provided in

Appendix E is processed. The pseudo code and flow chart that represents the code

changes are presented at the end of Chapter 3.

<u>Expansion of the Network Model</u>

The experimental model used as a baseline was expanded to test the new

algorithm on a traffic pattern that was more similar to what could be found on the traffic

on a university campus. When looking at traffic that was collected at UNI, there are small

peaks every day between class periods with large peaks at night which consist of online

movies and other video traffic. The simulation was expanded to include more

workstations, traffic types, and traffic patterns to include standard background traffic and

a couple major peaks within a thirty minute interval of simulated traffic. The simulation

time was chosen due to the limitations in the amount of time to run a scenario. The thirty

minute timeframe was enough to show a couple peaks as would be seen in a typical day

on campus. A single thirty minute simulation completed in six to eight hours. A visual

representation of the network is shown in Figure 24.

The data streams were modified in order to see peaks and valleys in the network

traffic. The Web Heavy stream traffic included a period of 60 seconds of heavy web

browsing followed by a break for 120 seconds then 20 seconds of light email traffic,

followed by a 5 second break and 10 seconds of file printing. The stream was set to start

at between 5 and 30 seconds and continue through this loop until the end of the

simulation. The http traffic was defined to have a constant page interval of .0055 seconds

and object size of 1000 bytes. It was to represent 5 large images per page. This traffic is

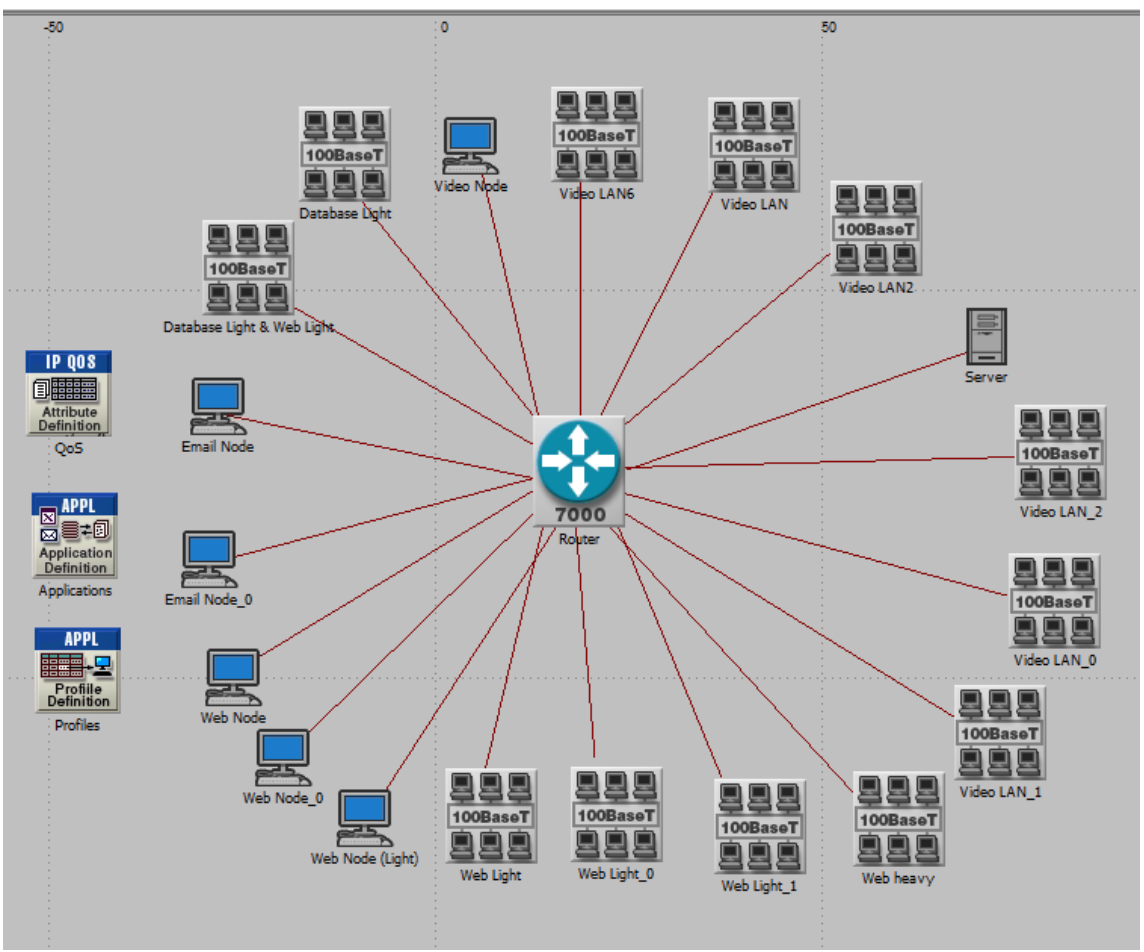set with a ToS setting of Standard (2) for QoS purposes.

Figure 24. Network for proposed algorithm simulations.

The network was expanded by adding subnets of workstations connected to the same router. The environment contained a total of 209 workstations going through a router connected to a single server representing all outbound traffic. The distribution of the traffic by workstation is listed in Table 5.

Table 5. *Simulated network workstation distribution.*

Data Stream	Single workstations	Subnets of workstations	Total workstations
Web Heavy	2	1	25
Web Light	1	4	56
Video	1	6	76
Email	2	0	2
Database Light	0	2	50

The Web Light traffic was set to start at between 5 and 10 seconds lasting for 10 seconds and repeating every 30 seconds until the end of the simulation. The http traffic was defined to have a constant page interval of .0055 seconds and object size of 1000 bytes. It was to represent 1 small image per page accessing 10 pages per server. This traffic is set with a ToS setting of Standard (2) for QoS purposes.

The video conferencing traffic was set to start between 5 and 60 seconds into the simulation, running for 60 seconds and repeating every 900 seconds until the end of the simulation. The video conferencing settings were changed on this data stream with frame interarrival time set at 15 frames/second with the frame size set at 128x240 pixels of All Discrete traffic. The ToS for this data stream was set at Interactive Voice (6) for QoS use.

The email traffic was set to start between 10 and 30 seconds into the simulation and continue until the end of the simulation. The send interarrival time was a constant

.0055 seconds with receive interarrival time of .010 seconds. The receive group size on the email was set to a constant 3 with the email size set at 1000 bytes. The email send group size was set to a constant 1. The email traffic was set with a ToS of Best Effort (0).

Database light traffic was set to start at a uniform 5 to 10 seconds into the simulation continue until the end of the simulation. The application definition was set to a 50% mix of queries per total transactions with a 30 second transaction interarrival time. The transaction size was set at the default light setting of 16 bytes with the ToS set to Excellent Effort (3) for QoS classification.

Simulation Results

The configured simulation created a traffic pattern with some background traffic and two definite peaks from the video streams in a thirty minute simulation. One of the background traffic types was the database streams. The statistics in Figure 25 show the total of the two database flows totaled less than 4000 bytes/second. Another contributor to the background traffic was the email traffic which produced a solid 2Mbps of total sent and received traffic as shown in Figure 26.

The web traffic generated a larger amount of traffic causing small peaks and valleys. The range was between 8Mbps and 34Mbps as shown in Figure 27. The traffic that caused the large peaks in the simulation was generated by video traffic.Video traffic caused the largest peaks in the sample data provided by UNI Network Services staff which was used to build this model. The peaks reached up to 135 Mbps as shown in Figure 28. In a real network, these peaks would be larger amounts of time as movies are watched of a matter of hours, however, due to the limitations of the simulation times, this

has been scaled into shorter period. The total utilization of the network connection from the router to the server never surpassed 34% for outgoing from the router to the server and 38% for incoming from the server. The network never hit max capacity, so any packet drops that were seen were associated with individual interfaces into the router running out of queue space for a particular ToS traffic.
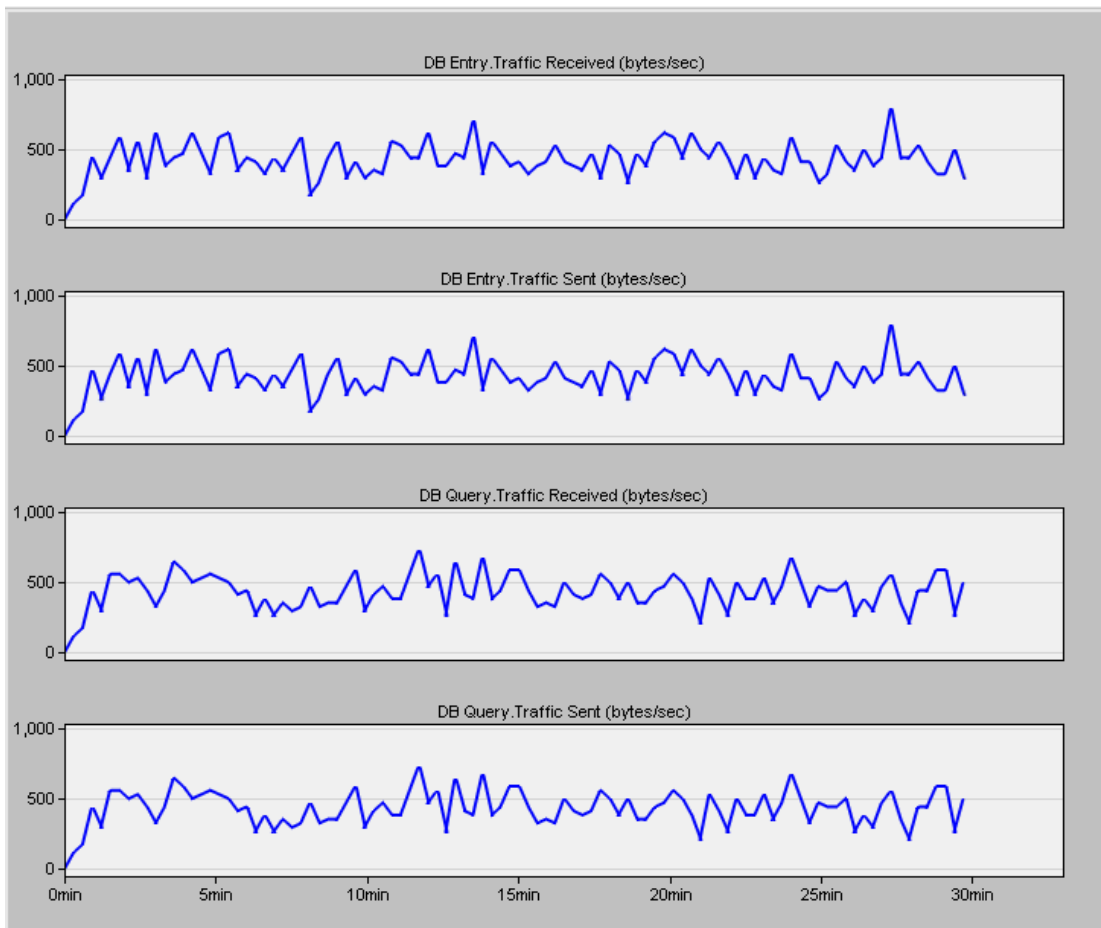
Figure 25. Database stream traffic bytes/second.
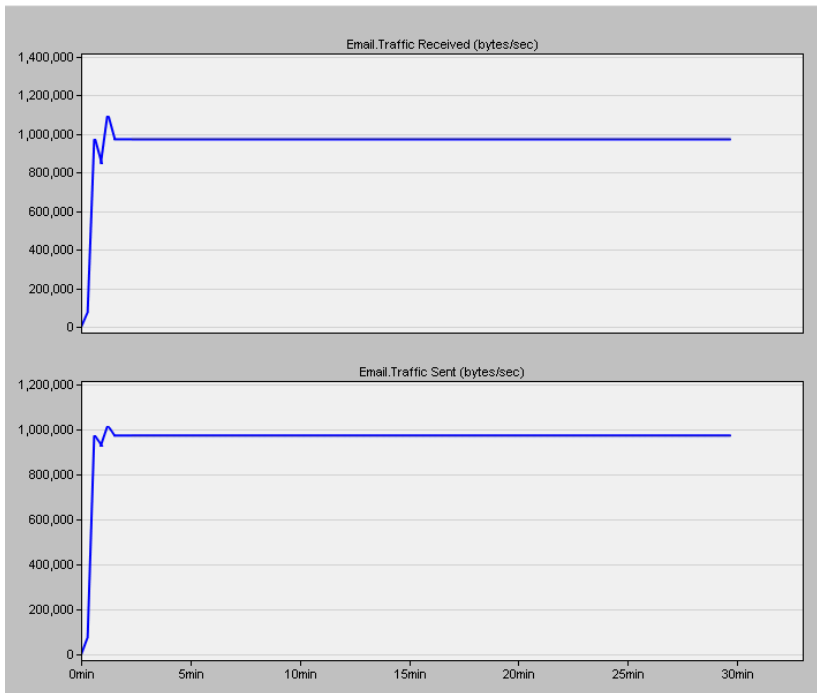
Figure 26. Email stream traffic bytes/second.
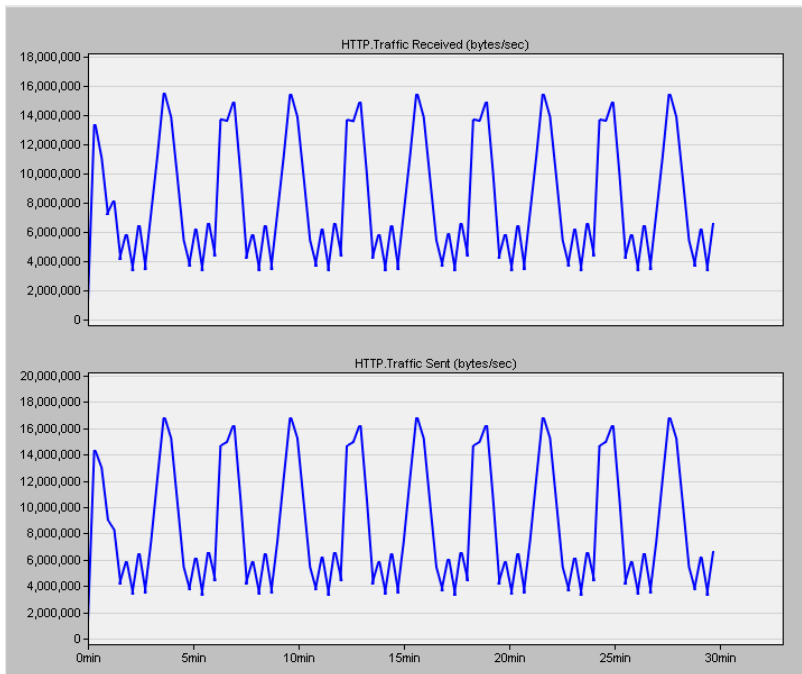
Figure 27. Web stream traffic bytes/second.

When reviewing network traffic shapers, all methods work well under low traffic since there is no contention for queues or bandwidth. If a network is saturated with an overload of network traffic, all QoS schemes have difficulty due to the sheer number of packets to process. The simulation traffic for this study was configured to represent a normal load with some sudden peaks in datat which could affect performance.
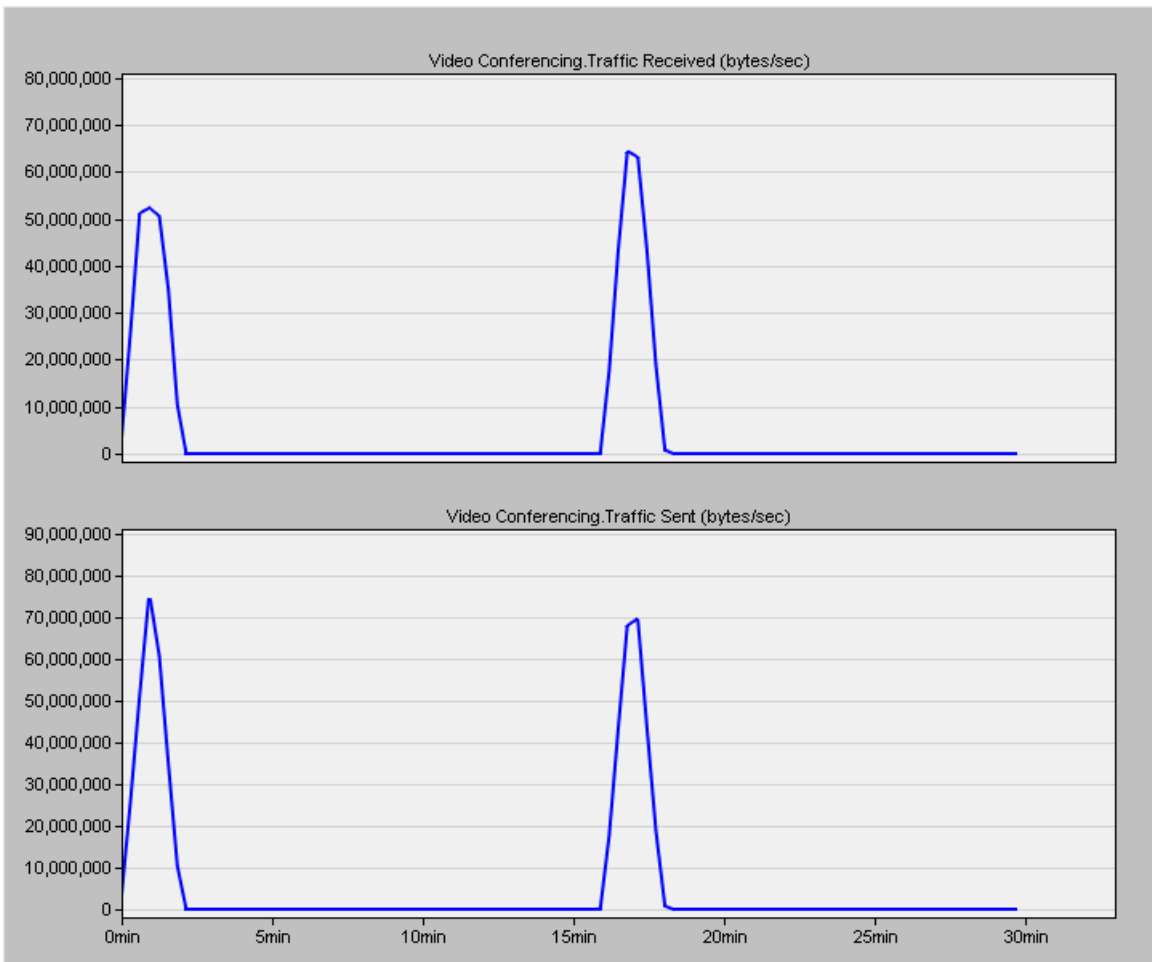
Figure 28. Video stream traffic bytes/second.
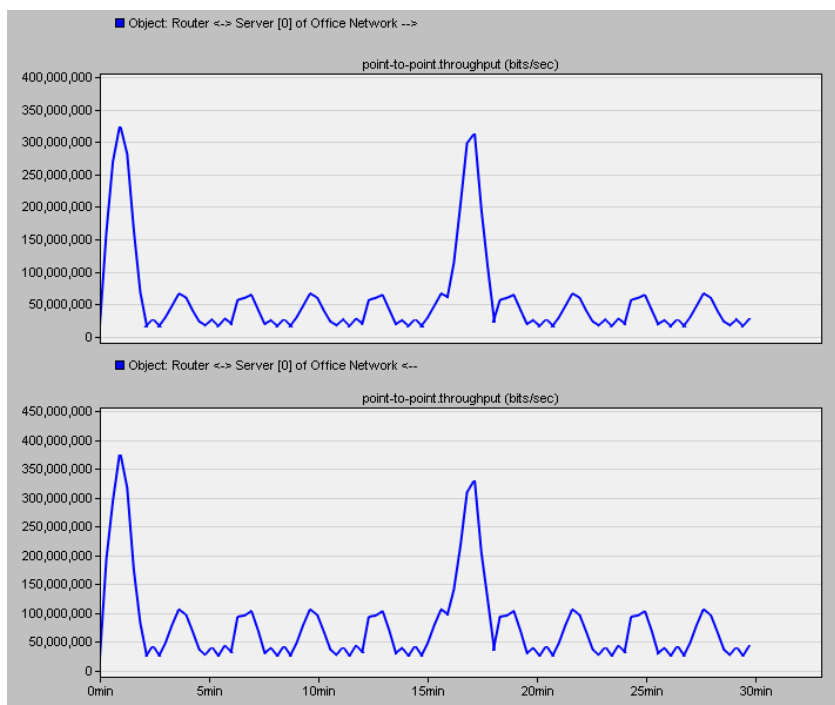
Figure 29. Router to server throughput for simulations (bits/second).
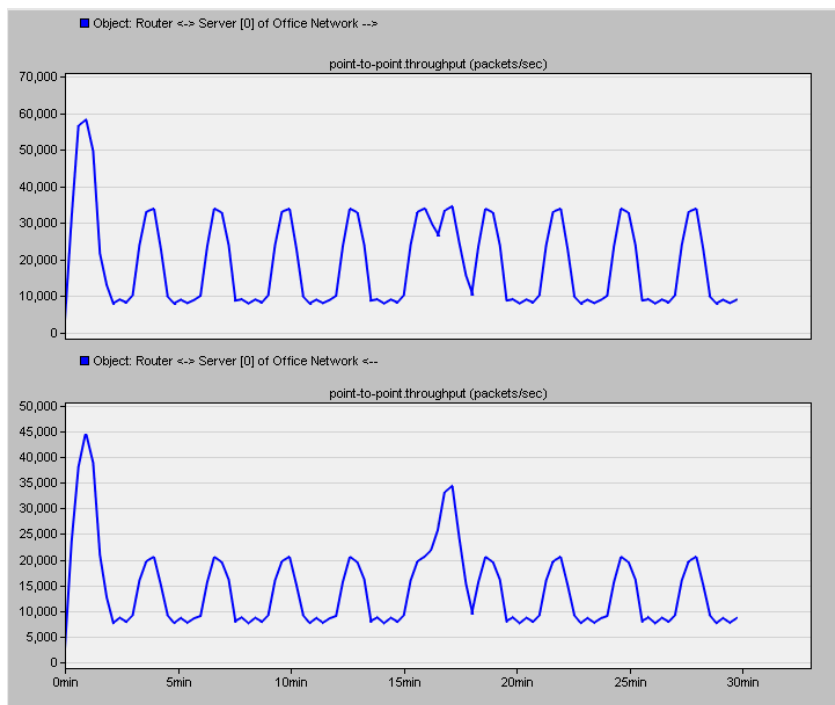
Figure 30. Router to server throughput for simulations (packets/second).

The MMSA mode was tested several times with different settings for the packet threshold (PT), dropped packet threshold(DT), and seconds in check interval(S). By looking at Figure 30, a level of 15,000 packets was chosen to represent a light load on the network to start testing. In trying to determine a threshold for packet loss, a level of 150 dropped per second was tested as a 1% packet loss level is considered the maximum allowed for network stability. A periodical measurement interval of 30 seconds was chosen since the nework changes were more than 30 seconds in length. Other simulations were completed by differing these three values. Changing the packet loss from 150 to 100 showed no difference returning the exact same statistics. Increasing the PT to 20000 also returned the same values for all statistics collected.

However, changing the packet loss to 50 packets with the interval at 30 seconds appears to have significant improvements in delays and no packet loss, but the throughput was significantly lower in reviewing Figures 34 and 35. For this scenario, looking at the DES logs, it shows that even though the packets were not dropped, the service queue was full and could not longer process data causing the network to become unstable. This was experienced when the peaks of video streaming were introduced and because network wasn't checked for 30 seconds. In the following figures, the simulation runs are indicated with the values for PT, DT, and S in the format of PT-DT-S. For example 15000-150-30 was a simulation run with a packet threshold (PT) of 15000 packets, a dropped threshold (DT) of 150 packets, and a periodic check interval (S) of of 30 seconds. Changing the periodic interval from 30 seconds to 5 seconds showed some consistent improvements in Ethernet delay as shown in Figure 31.
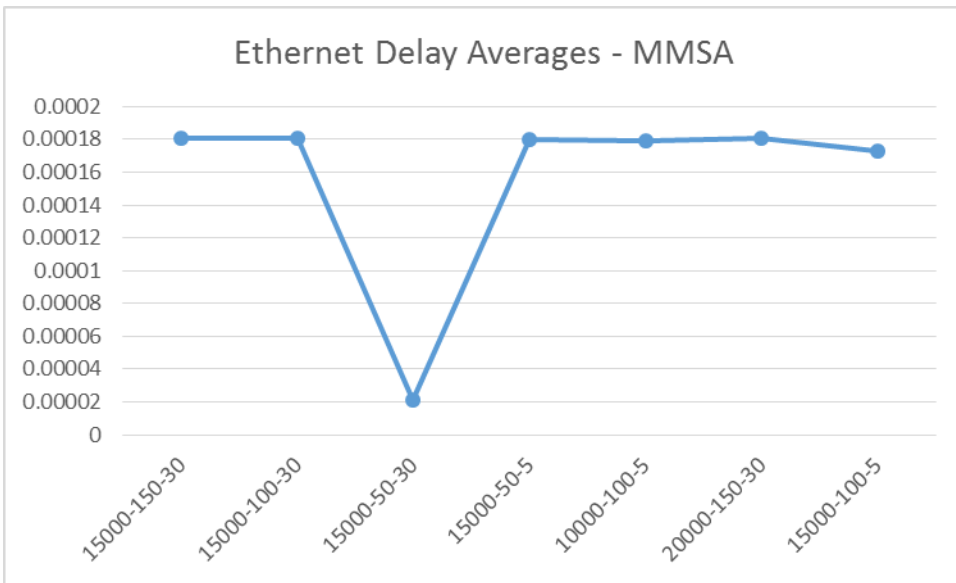
Figure 31. MMSA Ethernet delay averages.
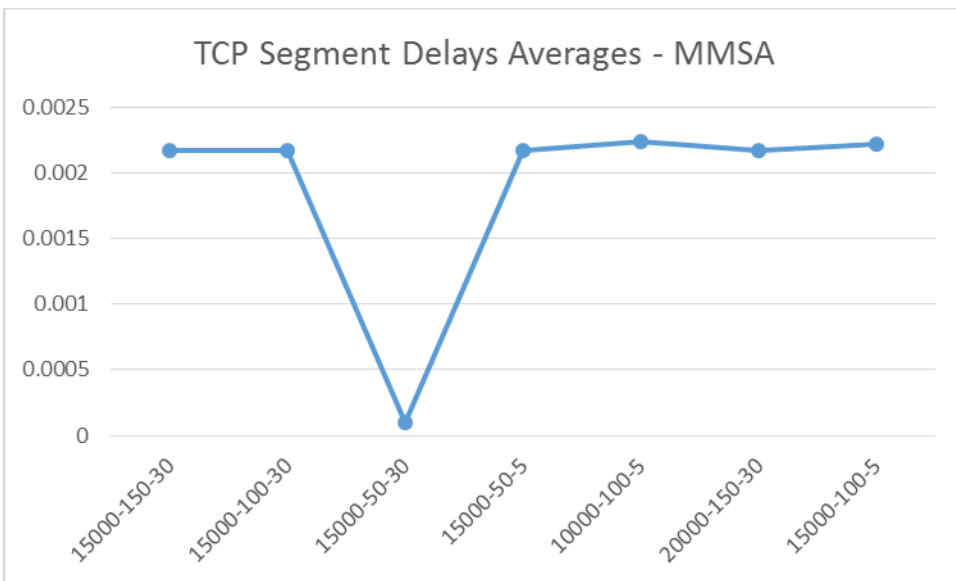
Figure 32. MMSA TCP segment delay averages.
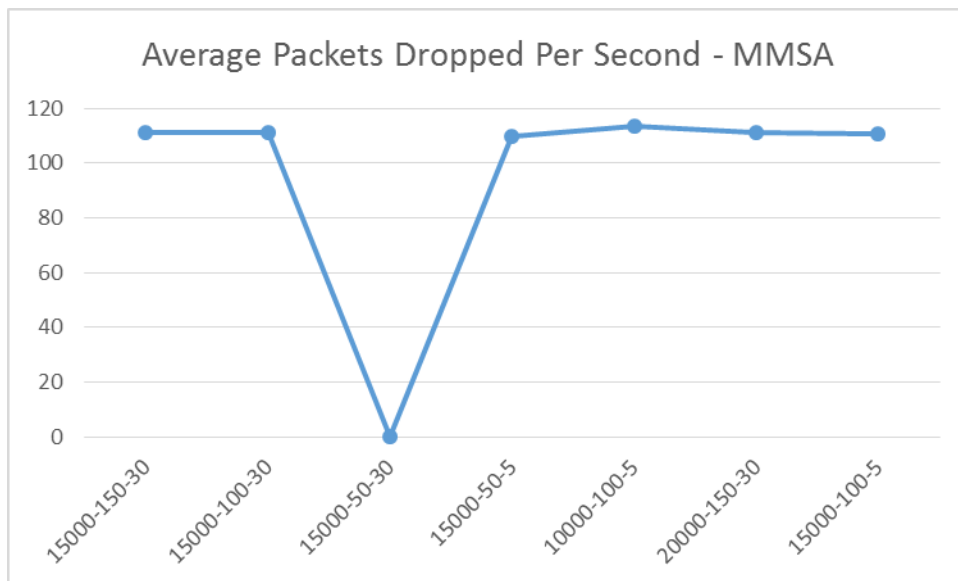
Figure 33. MMSA Packet dropped per second.
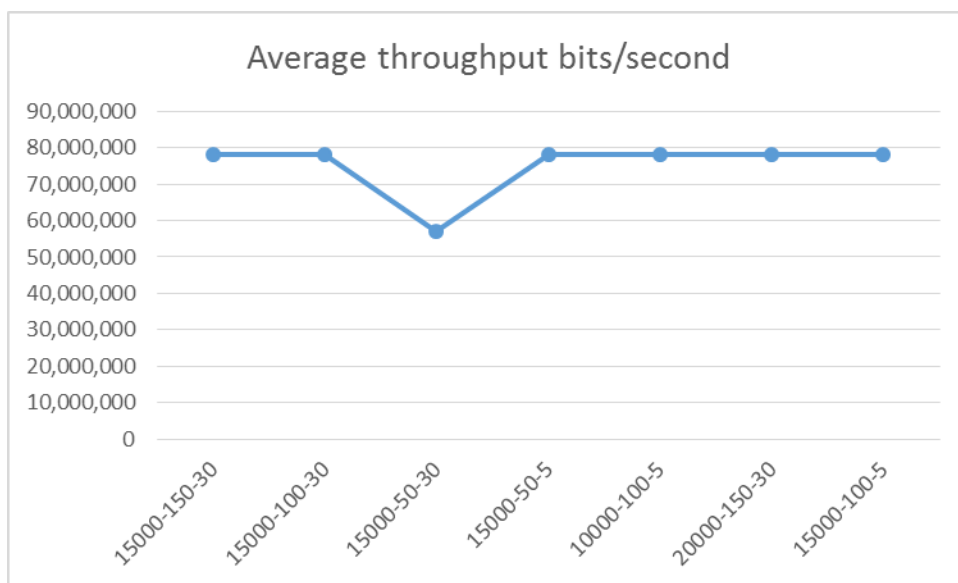
Figure 34. MMSA Throughput bits/second.
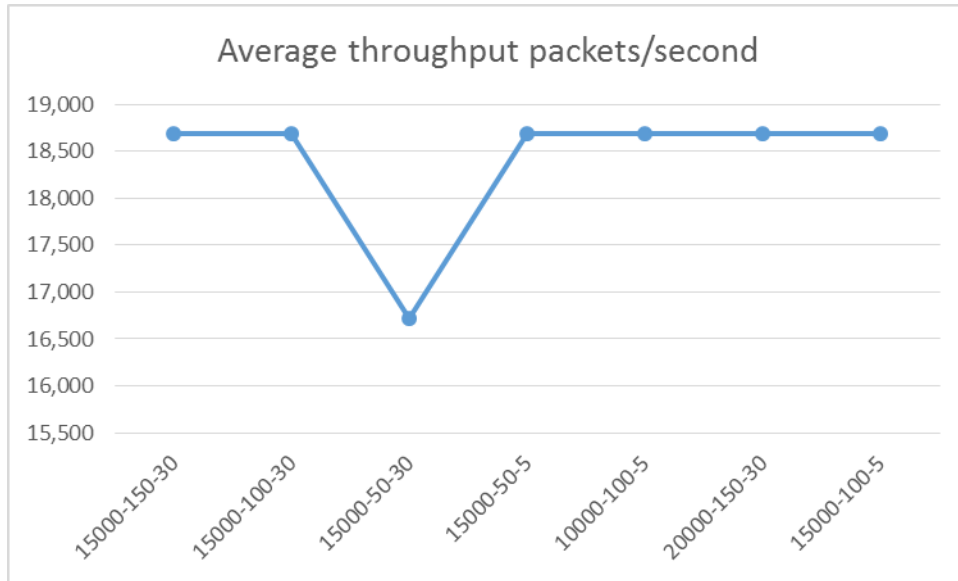
Figure 35. MMSA Throughput packets/second.

Due to the results shown in Table 6, it indicates that optimal Ethernet delay

results were obtained when PT = 15000 packets/second, DT=100 packets and S=5 for

this specific traffic pattern. Since the overall goal was to decrease delay without

impacting throughput, these settings were used to further test this algorithm.

Table 6. *MMSA algorithm detailed comparison.*

30 Min runs	Ethernet delay	Packets dropped/sec	TCP segment delay	Throughput bits/sec	Throughput packets/sec
15000-150-30	0.0001803	111.1	0.00217	78,195,807	18,689
15000-100-30	0.0001803	111.1	0.00217	78,195,807	18,689
15000-50-30	0.000020949	0	0.00010207	56,936,962	16,715
15000-50-5	0.0001801	109.9	0.00217	78,203,039	18,690
10000-100-5	0.0001791	113.4	0.00224	78,212,467	18,695
20000-150-30	0.0001803	111.1	0.00217	78,195,807	18,689
15000-100-5	0.0001727	110.8	0.00222	78,206,853	18,684

The simulation was modified to check on each interface every 5 seconds to determine if a threshold had been met that would change the MMSA mode between MMSA_PQ, MMSA_FIFO, and MMSA_WFQ. The simulator allows a seed value to be used that allows for slight variations in the traffic patterns. The simulations were run for each scenario with five different seed values to compare the averages of each scenario in regrads to Ethernet delay, TCP segment delay, and packets dropped. The details of the results are showin in Table 7.

Overall Performance Analysis

As expected, each of the existing QoS schemes had a weaker measurement and a stronger measurement. For example, the WFQ showed the lowest average dropped packets of the existing methods, but showed the worst Ethernet delay numbers. And although the PQ showed the lowest Ethernet delays, the TCP segment delays were higher. Whereas FIFO returned the worst packet drops but the best TCP segment delay average. The MMSA mode resulted in approximately 9% improvement in Ethernet delay, 6% decrease in packet loss, and 4.6% reduction in TCP segment delay. These results show a significant improvement based on those QoS measurements as shown in Figures 36, 37, and 38.

Table 7. *Run results for five simulations using five scenarios.*

30 Min runs	WFQ	PQ	FIFO	MMSA
Average Ethernet Delay(seconds)	0.0001786	0.0001789	0.0001777	0.0001727
	0.0002061	0.0002056	0.0002062	0.0002055
	0.0002543	0.000254	0.0002543	0.0002532
	0.0002304	0.0001616	0.0001824	0.0001874
	0.0001867	0.0001644	0.0002002	0.0001423
Average over 5 runs	0.00021122	0.00019290	0.00020416	0.00019222
Packets dropped/sec	110.7	108.20	110.3	110.8
	130	139.00	136	133.2
	178.3	173.4	174.9	172.1
	179.4	136.2	176.9	132
	134.2	138.7	135.6	141.6
Average over 5 runs	146.52	139.10	146.74	137.94
Average TCP segment delay (seconds)	0.00223	0.00223	0.00224	0.00222
	0.00263	0.00257	0.00255	0.00261
	0.00283	0.00291	0.00278	0.00273
	0.00278	0.00284	0.00283	0.00249
	0.0025	0.00253	0.00248	0.00244
Average over 5 runs	0.002594	0.002616	0.002576	0.002498
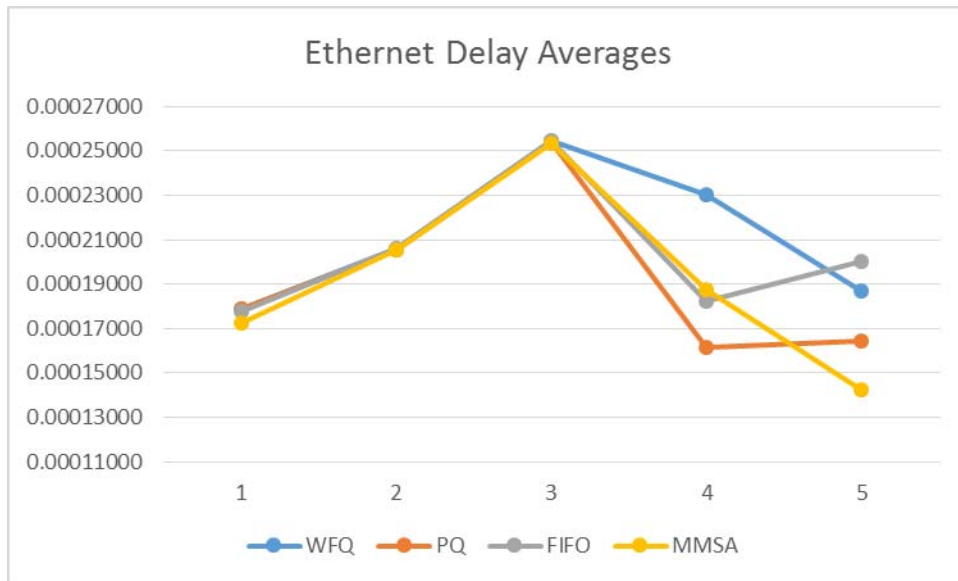
Figure 36. Ethernet delays averages for five simulation runs in seconds.
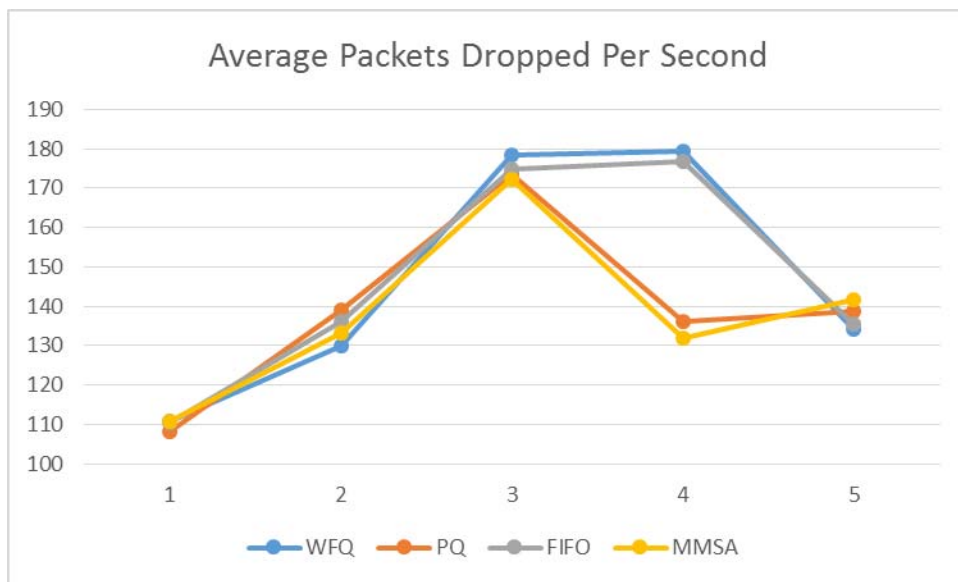
Figure 37. Packets dropped per second averages for five simulation runs.
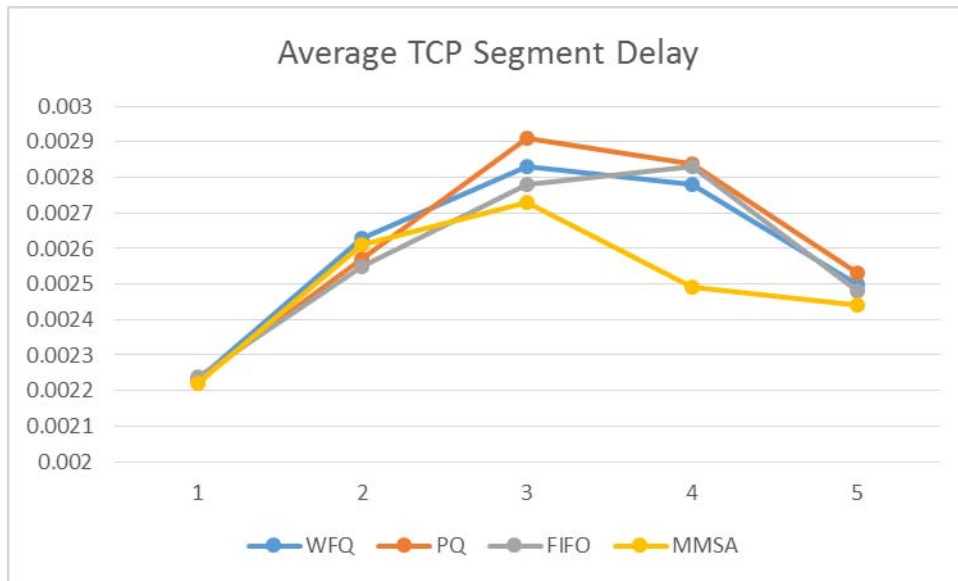
Figure 38. TCP segment delay averages for five simulation runs in seconds.

During the baseline studies, it was noted that there was a perceived inverse relationship between the delay and network throughput. In reviewing these larger scenarios, it is important to verify that the MMSA improvements are not adversely affecting the data transfer rate of the network. In the results, the throughput differences were negligible between the multiple runs showing less than a .03% difference in bits/second and less than .007% reduction in packets/second sent between the router and the server.
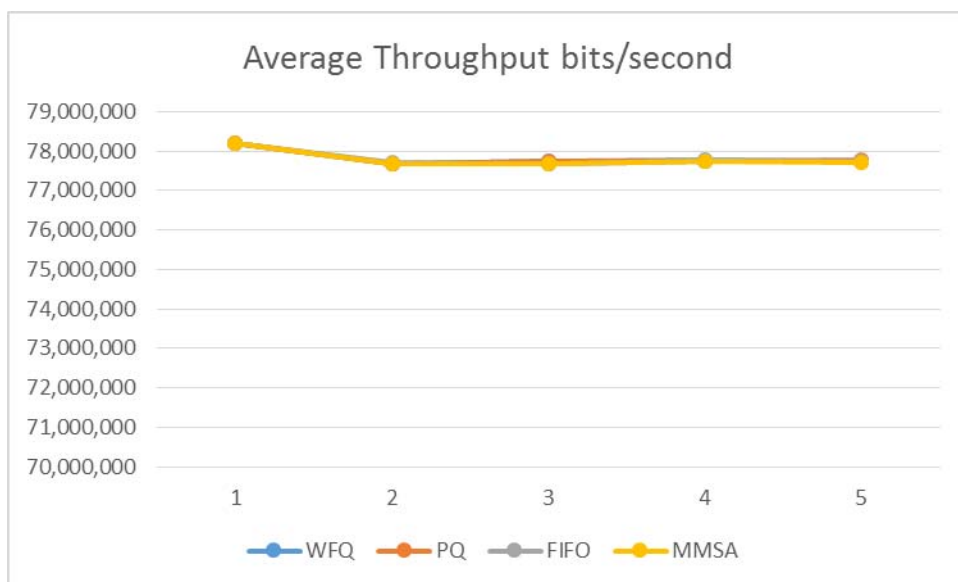
Figure 39. Average throughput over five simulation runs.
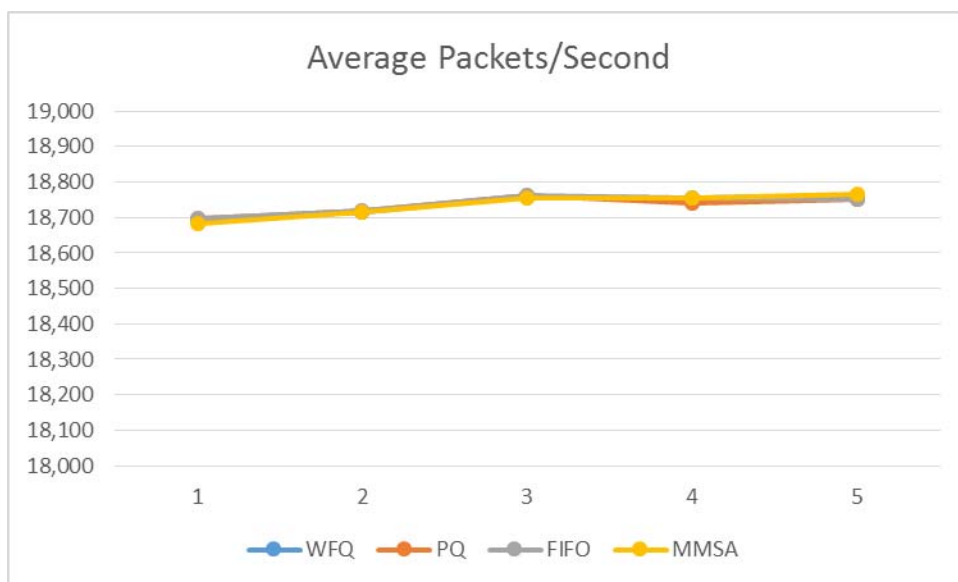
Figure 40. Average packets per second over five simulation runs.

CHAPTER 5

SUMMARY AND CONCLUSIONS

This study included conducting a survey, network traffic data collection and analysis, development of a new traffic control methodology, and an experiment to apply the new control algorithm in a network simulation to measure the results. The wireless survey showed that only 51% of the respondents were satisfied with the current network, most noting speed problems with video applications. Based on the survey results, data was collected for a two week period of time to use as a model for the traffic patterns for the remainder of the student. However, the primary focus of this study was on the development of a traffic shaping algorithm that would improve the QoS on a wireless network on a university campus. The goal of the research was to create a new architecture for QoS that would allow a router to dynamically shift between different queuing mechanisms to improve network delay and packet loss without negatively impacting data throughput. The Multi-Mode Self-Adaptive algorithm was proposed to define a mechanism for this architecture. The MMSA was implemented within the code of a Cisco® router in the OPNET Modeler software to simulate a university network.

The MMSA algorithm used two thresholds that were evaluated on a set interval to determine which mode should be employed for each interface on the router. The thresholds used were packets per second and packets dropped per time interval. The existing QoS schemes of FIFO, PQ ToS, and WFQ were compared with the MMSA implementation to review the desired QoS measurements of Ethernet delay, TCP segment delay, and packet loss for improvements. The results indicated that the MMSA algorithm

provided a 9% improvement in Ethernet delays compared to WFQ, a 6% decrease in dropped packets over FIFO, and 4.6% reduction in TCP segment delay over the PQ ToS scheme while having an insignificant impact on the data rate between the router and the server.

In general, all QoS schemes would work well under low traffic and have difficulties under complete saturation. Therefore the scenarios used were based on a medium traffic load with a few high peaks. For further studies, the algorithm should be test be tested under differing network traffic mixes and loads. Another area that could be investigated would be the use of a different measurement type for the threshold to switch modes. Other suggested measurements would be bytes per second since packets per second could behave differently based on the type of traffic being simulated. In addition, a complete implementation of this architecture into Cisco® firmware to verify the validity of the simulation would be desirable.

REFERENCES

Airoldi, E. M., & Blocker, A. W. (2013). Estimating latent processes on a network from indirect measurements. *Journal of the American Statistical Association, 108*(501), 149-164.

Ash, G. R. (2007). *Traffic Engineering and QoS Optimization of Integrated Voice & Data Networks*. Amsterdam: Elsevier/Morgan Kaufmann Publishers.

Athanasiou, G. (2012). Dynamic resource management in 802.11 wireless mesh networks. *Journal of Computer Systems, Networks, and Communications, 2012*.

Bellavista, P., Cai, Y., & Magedanz, T. (2011). Recent advances in mobile middleware for wireless systems and services. *Mobile Networks & Applications*, *16*(3), 267-269.

Bellavista, P., Corradi, A., & Foschini, L. (2008). Dynamic and context-aware streaming adaptation to smooth quality degradation due to ieee 802.11 performance anomaly. *The Journal of Supercomputing*, *45*(1), 15.

Bjorkbom, M., Nethi, S., Eriksson, L., & Jantti, R. (2011). Wireless control system design and co-simulation. *Control Engineering Practice*, *19*(9), 1075.

Blake, S., Black, D., Carlson, M. Davies, E. Wang, Z., & Weiss, W. (1998). *IETF RFC 2475 - An Architecture for Differentiated Services section 2.3.3.3 - Internet standard definition of "Shaper."* Retrieved from http://tools.ietf.org/html/rfc2475#section-2.3.3.3

Calhan, A., & Atmaca, S. (2013). A new Network Coordinator Node Design Selecting the Optimum Wireless Technology for Wireless Body Area Networks. *Ksii Transactions On Internet And Information Systems*, *7*(5), 1077-1093.

Chaudhuri, S. (2012). *Design and implementation of a differentiated service based qos model for real-time interactive traffic on constrained bandwidth ip networks*. Retrieved from http://arxiv.org/abs/1205.3319

Cheng, S., Huang, C., & Tu, C. (2011). Racoon: A multiuser qos design for mobile wireless body area networks. *Journal of Medical Systems*, *35*(5), 1277.

Ciampa, M. (2006). *CWNA Guide to Wireless LANs: Second Edition.* Boston, MA: Thomson Course Technology.

Ciampa, M. (2013). *CWNA Guide to Wireless LANs: Third Edition.* Boston, MA: Course Technology.

Cisco® (2008). *Configuring Generic Traffic Shaping*. Retrieved from http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfpolsh.html

Dhanaskaran, R., Jayachandran, A., Rajan, P., KharmegaSundararaj, G., & SudarsonRamaPerumal, T. (2011). Performance and evaluation of congestion control and routing algorithm for improving internet traffic control system. *International Journal of Computer Science and Network Security, 11*(6), 57-61.

Daniel-Simion, D. (2012). Traffic shaping and traffic policing impacts on aggregate traffic behavior in high speed networks. *International Journal of Advanced Computer Science, 2*(10), 389-393.

Geetha, S., & Jayaparvathy, R. (2011). Dynamic bandwidth allocation for multiple traffic classes in ieee 802.16e wimax networks: A petrinet approach. *Journal of Computer Science*, *7*(11), 1717.

Guerrero-Ibáñez, A., Contreras-Castillo, J., Barba, A., & Reyes, A. (2011). A qos-based dynamic pricing approach for services provisioning in heterogeneous wireless access networks. *Pervasive and Mobile Computing*, *7*(5), 569.

Hassell, J. (2004, June 2). *Wireless attacks and penetration testing (part 1 of 3)*. Retrieved from http://www.symantec.com/connect/articles/wireless-attacks-and-penetration-testing-part-1-3

Hiertz, G.R., Denteneer, D., Max, S., Taori, R., Cardona, J., Berlemann, L., & Walke, B. (2010). IEEE 802.11s: the WLAN mesh standard. *IEEE Wireless Communications*, *17*(1), 104-111.

Howard, S., Kjeldskov, J., & Skov, M.B. (2007). Pervasive computing in the domestic space. *Personal and Ubiquitous Computing, 11*(5), 329-333.

Hwang, J., & Yoo, M. (2011). Class-limited algorithm for supporting qos and fairness in ethernet passive optical network. *Photonic Network Communications*, *22*(3), 221.

Hu, L., Dung, O., Liu, Q., Han, T., & Sun, Y. (2013). Integration of Wireless Body Area Networks (WBANs) and WAN, WiMAX and LTE. *Ksii Transactions On Internet And Information Systems*, *7*(5), 980-997.

Jain, A., Kumar, M., & Chaubey, V. K. (2011). Traffic Modeling of Ring and Mesh Networks with Improved Quality of Service. *IUP Journal Of Telecommunications, 3*(3), 7-20.

Juqiong, Y., Chu-Ho, T., & Xu-dong, Z. (2013). Network Traffic Prediction Using the Combination of Chaos Theory and Simple-MKL. *Journal Of Networks, 8*(8), 1750. doi:10.4304/jnw.8.8.1750-1756.

Kurose, J., & Ross, K. (2010). *Computer networking: A top-down approach.* (5th ed.). Boston, MA: Pearson Education.

Ma, K., Bartos, R., & Bhatia, S. (2011). A survey of schemes for internet-based video delivery. *Journal of Network and Computer Applications*, *34*(5), 1572.

Mardini, W., & Alfoul, M. (2011). Modified wrr scheduling algorithm for wimax networks. *Network Protocols and Algorithms*, *3*(2), 24.

Mehmood, R., & Alturki, R. (2011). A scalable multimedia qos architecture for ad hoc networks. Multimedia *Tools and Applications*, *54*(3), 551.

Portoles, M., Zhong Z., & Choi,S.(2003). IEEE 802.11 downlink traffic shaping scheme for multi-user service enhancement. *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings, 2,* 1712-1716.

Semeria, C. (2001). Supporting Differentiated Service Classes: Queue Scheduling Disciplines, *Juniper Networks*, pp.11-14.

Soudani, A., Divoux, T., & Tourki, R. (2012). Data traffic load balancing and QoS in IEEE 802.11 network: Experimental study of the signal strength effect.(Report). *Computers & Electrical Engineering*, *38*(6), 1717. doi:10.1016/j.compeleceng.2012.07.016

Usui, T., Kitatsuji, Y., & Yokota, H. (2011). A study on traffic management cooperating with ims in mpls networks. *Telecommunication Systems*.

Valenzuela, J. L., Monleon, A., San Esteban, I., Porteles, M., & Sallent, O. (2004). *A hierarchical token bucket algorithm to enhance qos in ieee 802.11 :proposal, implementation and evaluation. In IEEE Vehicular Technology Conference 2004, 4*, pp. 2659-2662. doi: 10.1109/VETECF.2004.1400539

Vieira, F., Bianchi, G., & Lee, L. (2010). A network traffic prediction approach based on multifractal modeling. *Journal Of High Speed Networks*, *17*(2), 83-96. doi:10.3233/JHS-2010-0334

Vuojarvi, H., Isomaki, H., & Hynes, D. (2010). Domestication of a laptop on a wireless university campus: a case study. *Australasian Journal of Educational Technology*, *26*(2), 250-267.

Zeng, S., Uzun, N., & Papvassiliou, S. (2001). An improved fair traffic shaping algorithm for high speed packet switches. *High Performance Switching and Routing, 2001 IEEE Workshop*,68-72. doi:10.1109/HPSR.2001.923606

APPENDIX A

WIRELESS SURVEY RESULTS OF STUDENTS

University of Northern Iowa
Online Voting and Surveys Results by Question
UNI Wireless Service Survey (Survey ID: 7350)
04/02/2012 12:00 AM - 04/23/2012 12:00 AM

1. **Do you currently use or are interested in using wireless devices to connect to the Internet?**

	Percentage	Responses
Yes	98.8%	255
No	1.2%	3
	Total Responses:	**258**

2. Check all locations where you have ever successfully connected a wireless device to the Internet

	Percentage	Responses
Restaurant or Coffee Shop	70.9%	180
Store	37.0%	94
Airport	45.7%	116
Hotel	74.4%	189
Outdoor hotspot	31.5%	80
Residence or Home	92.9%	236
Public Library	63.8%	162
School or University or College	92.9%	236
Hospital	21.3%	54
Other	1.2%	3

Other Responses:
home
RIDER HALL!!!!!
Charter Bus

	Total Responses:	**254**

3. Check all wireless devices which you have successfully connected to the Internet

	Percentage	Responses
Laptop	99.6%	253
Tablet (iPad, Xoom,etc)	28.7%	73
Desktop computer	23.2%	59
iPod	55.9%	142
Cell phone	54.7%	139
Game console(Wii,PS3,Xbox,etc)	32.7%	83
Handheld game system	8.3%	21
Other	0.8%	2

Other Responses:
Kindle
Blu-ray

4.   In general, how often do you conduct the following online using a wireless device?

	Never	Once a month or less	Several times each month	Several times each week	At least once a day
Conduct research for classwork	5 (2.0%)	23 (9.1%)	42 (16.5%)	83 (32.7%)	101 (39.8%)
Communicate with students, teachers, or co-workers	3 (1.2%)	4 (1.6%)	20 (7.9%)	61 (24.0%)	166 (65.4%)
Send email	1 (0.4%)	4 (1.6%)	17 (6.8%)	55 (22.0%)	173 (69.2%)
Chat in chat rooms	147 (58.1%)	38 (15.0%)	28 (11.1%)	22 (8.7%)	18 (7.1%)
Online banking or financial transactions	26 (10.3%)	31 (12.3%)	94 (37.2%)	74 (29.2%)	28 (11.1%)
Post content to a website	24 (9.5%)	39 (15.4%)	53 (20.9%)	69 (27.3%)	68 (26.9%)
Watch videos, movies or TV shows	11 (4.4%)	27 (10.7%)	61 (24.2%)	75 (29.8%)	78 (31.0%)
Take classes online	99 (39.3%)	46 (18.3%)	34 (13.5%)	40 (15.9%)	33 (13.1%)
Make or receive phone calls through the internet	107 (42.6%)	66 (26.3%)	47 (18.7%)	18 (7.2%)	13 (5.2%)
Search for jobs	55 (21.7%)	84 (33.2%)	59 (23.3%)	40 (15.8%)	15 (5.9%)
Download music	37 (14.6%)	53 (20.9%)	90 (35.6%)	48 (19.0%)	25 (9.9%)
Read blogs or general web browsing	16 (6.4%)	14 (5.6%)	38 (15.1%)	47 (18.7%)	136 (54.2%)
Buy products, travel,etc.	20 (8.0%)	83 (33.1%)	91 (36.3%)	36 (14.3%)	21 (8.4%)
Play online games	66 (26.2%)	67 (26.6%)	43 (17.1%)	39 (15.5%)	37 (14.7%)
Social networking, Facebook, MySpace,etc.	10 (4.0%)	7 (2.8%)	6 (2.4%)	32 (12.7%)	197 (78.2%)
Browse the UNI web pages for information	4 (1.6%)	10 (3.9%)	48 (18.9%)	78 (30.7%)	114 (44.9%)

Total Responses: **254**

5. Do you own wireless devices that you have connected to the Internet at other locations, but have not done so on the UNI campus?

	Percentage	Responses
Yes	39.1%	101
No	60.9%	157
	Total Responses:	**258**

6. Check all wireless devices which you have successfully connected to the Internet on the UNI campus

	Percentage	Responses
Laptop	87.4%	83
Tablet (iPad, Xoom, etc)	11.6%	11
Desktop computer	5.3%	5
iPod	31.6%	30
Cell phone	33.7%	32
Game console(Wii,PS3,Xbox,etc)	11.6%	11
Handheld game system	0.0%	0
Other	3.2%	3

Other Responses:
I have not seccessfully connected to UNI's wireless
none
None, haven't been able to maintain a connection.

	Total Responses:	**95**

7. Select all locations where you have successfully connected to the Internet on the UNI campus?

	Percentage	Responses
Library	73.1%	68
Academic building	73.1%	68
Administrative building	23.7%	22
Maucker Union	66.7%	62
Outside buildings on central campus	19.4%	18
Athletic facilities	9.7%	9
Dorm	76.3%	71
Other	3.2%	3

Other Responses:
GBPAC
I have not seccessfully connected to UNI's wireless
Residence Hall

	Total Responses:	**93**

8. Have you experienced issues with the performance of an application on a wireless device while using one of the UNI wireless networks(i.e. very slow web page loads or inability to view a video)?

	Percentage	Responses
Yes	84.2%	202
No	15.8%	38
	Total Responses:	**240**

9. **What type of activities are too slow on the UNI wireless network that you would refrain from using that service?**

	Percentage	Responses
YouTube or other videos	71.9%	156
NetFlix or other movies	47.9%	104
Video gaming	24.4%	53
Web browsing	50.7%	110
Email access	42.4%	92
Downloading music	31.3%	68
Social networking or chatting	34.6%	75
Skype or webcam or phone software	60.8%	132
Other	4.6%	10

Other Responses:
Sometimes the wireless just doesn't work
blackboard
cant get on or connect most of time use labs and leave laptops at home
always freezes
Conneting
MyUniverse
basic connection
Could be any of them depends on the day
cannot connect my phone to uni wireless
access to UNI myuniverse and elearning.

Total Responses: 217

10. In general, how satisfied are you with the wireless network services provided on the UNI campus

	Percentage	Responses
Very Satisfied	10.5%	27
Satisfied	43.8%	113
Neither Satisfied nor unsatisfied	20.2%	52
Unsatisfied	19.0%	49
Very Unsatisfied	6.6%	17

Total Responses: 258

APPENDIX B

WIRELESS SURVEY RESULTS FACULTY/STAFF

University of Northern Iowa
Online Voting and Surveys Results by Question
UNI Wireless Service Survey (Fac/Staff) (Survey ID: 7370)
04/02/2012 12:00 AM - 04/23/2012 12:00 AM

1. **Do you currently use or are interested in using wireless devices to connect to the Internet?**

	Percentage	Responses
Yes	98.5%	66
No	1.5%	1
	Total Responses:	**67**

2. **Check all locations where you have ever successfully connected a wireless device to the Internet**

	Percentage	Responses
Restaurant or Coffee Shop	87.9%	58
Store	34.8%	23
Airport	69.7%	46
Hotel	90.9%	60
Outdoor hotspot	33.3%	22
Residence or Home	90.9%	60
Public Library	39.4%	26
School or University or College	77.3%	51
Hospital	33.3%	22
Other	3.0%	2

Other Responses:
pub
Rest Area

	Total Responses:	**66**

3. **Check all wireless devices which you have successfully connected to the Internet**

	Percentage	Responses
Laptop	97.0%	64
Tablet (iPad, Xoom,etc)	56.1%	37
Desktop computer	24.2%	16
iPod	30.3%	20
Cell phone	62.1%	41
Game console(Wii,PS3,Xbox,etc)	13.6%	9
Handheld game system	3.0%	2
Other	4.5%	3

Other Responses:
Roku (for Netflix)
kindle
tv, bluray, tivo,

	Total Responses:	**66**

4. **In general, how often do you conduct the following online using a wireless device?**

	Never	Once a month or less	Several times each month	Several times each week	At least once a day
Conduct research for classwork	19 (30.2%)	14 (22.2%)	2 (3.2%)	15 (23.8%)	13 (20.6%)
Communicate with students, teachers, or co-workers	2 (3.1%)	5 (7.8%)	6 (9.4%)	14 (21.9%)	37 (57.8%)
Send email	0 (0.0%)	3 (4.6%)	4 (6.2%)	14 (21.5%)	44 (67.7%)
Chat in chat rooms	47 (73.4%)	12 (18.8%)	4 (6.2%)	1 (1.6%)	0 (0.0%)
Online banking or financial transactions	18 (28.1%)	9 (14.1%)	24 (37.5%)	13 (20.3%)	0 (0.0%)
Post content to a website	19 (29.7%)	15 (23.4%)	11 (17.2%)	13 (20.3%)	6 (9.4%)
Watch videos, movies or TV shows	17 (27.0%)	16 (25.4%)	16 (25.4%)	9 (14.3%)	5 (7.9%)
Take classes online	48 (76.2%)	8 (12.7%)	3 (4.8%)	2 (3.2%)	2 (3.2%)
Make or receive phone calls through the internet	33 (51.6%)	20 (31.2%)	7 (10.9%)	4 (6.2%)	0 (0.0%)
Search for jobs	40 (62.5%)	14 (21.9%)	4 (6.2%)	6 (9.4%)	0 (0.0%)
Download music	27 (42.9%)	20 (31.7%)	15 (23.8%)	1 (1.6%)	0 (0.0%)
Read blogs or general web browsing	6 (9.4%)	4 (6.2%)	7 (10.9%)	20 (31.2%)	27 (42.2%)
Buy products, travel,etc.	5 (7.8%)	14 (21.9%)	32 (50.0%)	13 (20.3%)	0 (0.0%)
Play online games	31 (50.0%)	14 (22.6%)	7 (11.3%)	6 (9.7%)	4 (6.5%)
Social networking, Facebook, MySpace,etc.	11 (17.2%)	4 (6.2%)	14 (21.9%)	12 (18.8%)	23 (35.9%)
Browse the UNI web pages for information	2 (3.1%)	13 (20.3%)	21 (32.8%)	18 (28.1%)	10 (15.6%)

Total Responses: **65**

5. Do you own wireless devices that you have connected to the Internet at other locations, but have not done so on the UNI campus?

	Percentage	Responses
Yes	65.7%	44
No	34.3%	23
	Total Responses:	**67**

6. Check all wireless devices which you have successfully connected to the Internet on the UNI campus

	Percentage	Responses
Laptop	87.9%	29
Tablet (iPad, Xoom, etc)	39.4%	13
Desktop computer	18.2%	6
iPod	18.2%	6
Cell phone	42.4%	14
Game console(Wii,PS3,Xbox,etc)	3.0%	1
Handheld game system	0.0%	0
Other	0.0%	0
	Total Responses:	**33**

7. Select all locations where you have successfully connected to the Internet on the UNI campus?

	Percentage	Responses
Library	21.9%	7
Academic building	81.2%	26
Administrative building	37.5%	12
Maucker Union	43.8%	14
Outside buildings on central campus	25.0%	8
Athletic facilities	18.8%	6
Dorm	9.4%	3
Other	0.0%	0
	Total Responses:	**32**

8. Have you experienced issues with the performance of an application on a wireless device while using one of the UNI wireless networks(i.e. very slow web page loads or inability to view a video)?

	Percentage	Responses
Yes	61.0%	36
No	39.0%	23
	Total Responses:	**59**

9. What type of activities are too slow on the UNI wireless network that you would refrain from using that service?

	Percentage	Responses
YouTube or other videos	51.6%	16
NetFlix or other movies	22.6%	7
Video gaming	3.2%	1
Web browsing	58.1%	18
Email access	64.5%	20

Downloading music	9.7%	3
Social networking or chatting	16.1%	5
Skype or webcam or phone software	12.9%	4
Other	9.7%	3

Other Responses:
email
Google Docs rus slow
Poor ability to connect in Baker Hall

	Total Responses:	**31**

10. In general, how satisfied are you with the wireless network services provided on the UNI campus

|	Percentage	Responses
Very Satisfied	4.5%	3
Satisfied	35.8%	24
Neither Satisfied nor unsatisfied	29.9%	20
Unsatisfied	23.9%	16
Very Unsatisfied	6.0%	4
	Total Responses:	**67**

APPENDIX C

OPNET SIMULATION CONFIGURATION STEPS

<u>Setting up the Network Devices</u>

The first step in setting up a simulation in OPNET requires you to define the

physical network structure. The software comes with an object palette that allows the

selection of a variety of industry standard network devices called nodes. There are

servers, workstations, switches, routers, subnets, etc. The object palette also provides a

selection of communication links for wired networking such as a 1000BaseX Ethernet

link. The items can be placed individually or the network can be created using a wizard in

the software. A very small scenario is shown in Figure C1.
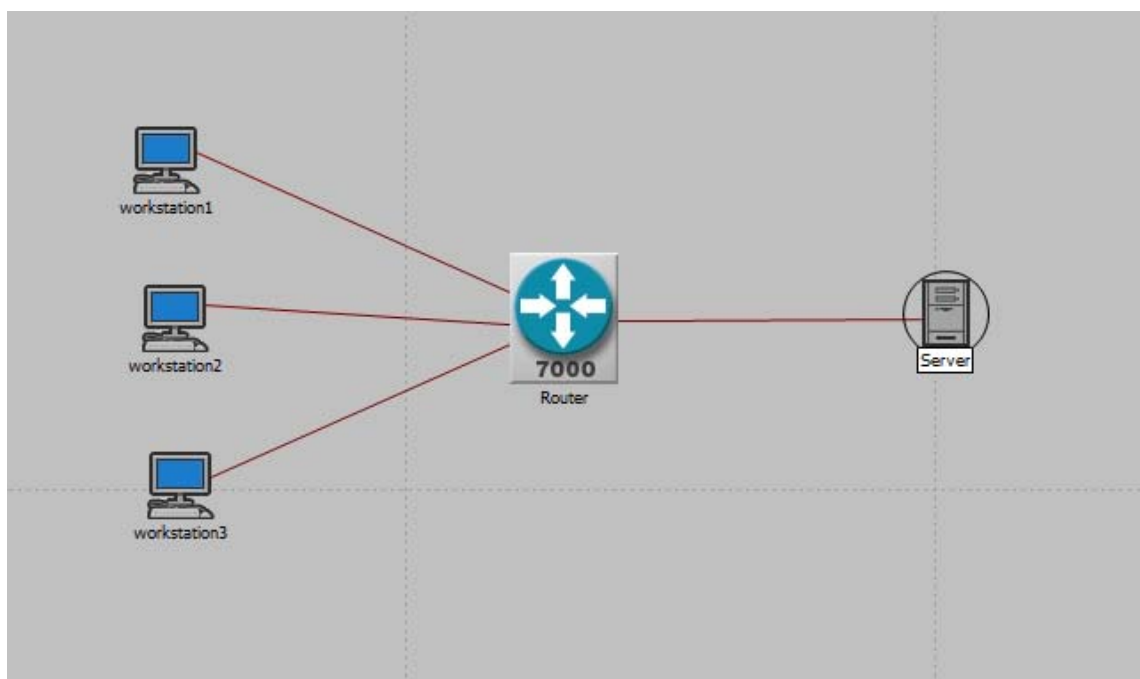
Figure C1. Sample small OPNET network scenario.

Configuring the Application Definitions

The next step in configuring the simulation involves setting up the application

traffic that will be used to evaluate the network configuration. To perform these steps, the

application definition object is added to the scenario. The object has attributes that all

modification to the delivered applications to change the size and behavior of the traffic.

For example, the definition provided for the Web Browsing (Heavy HTTP1.1) allows

modification of the page properties, the server settings, the page interarrival times and the

type of service indicator that would be used for QoS measurements as shown in Figure
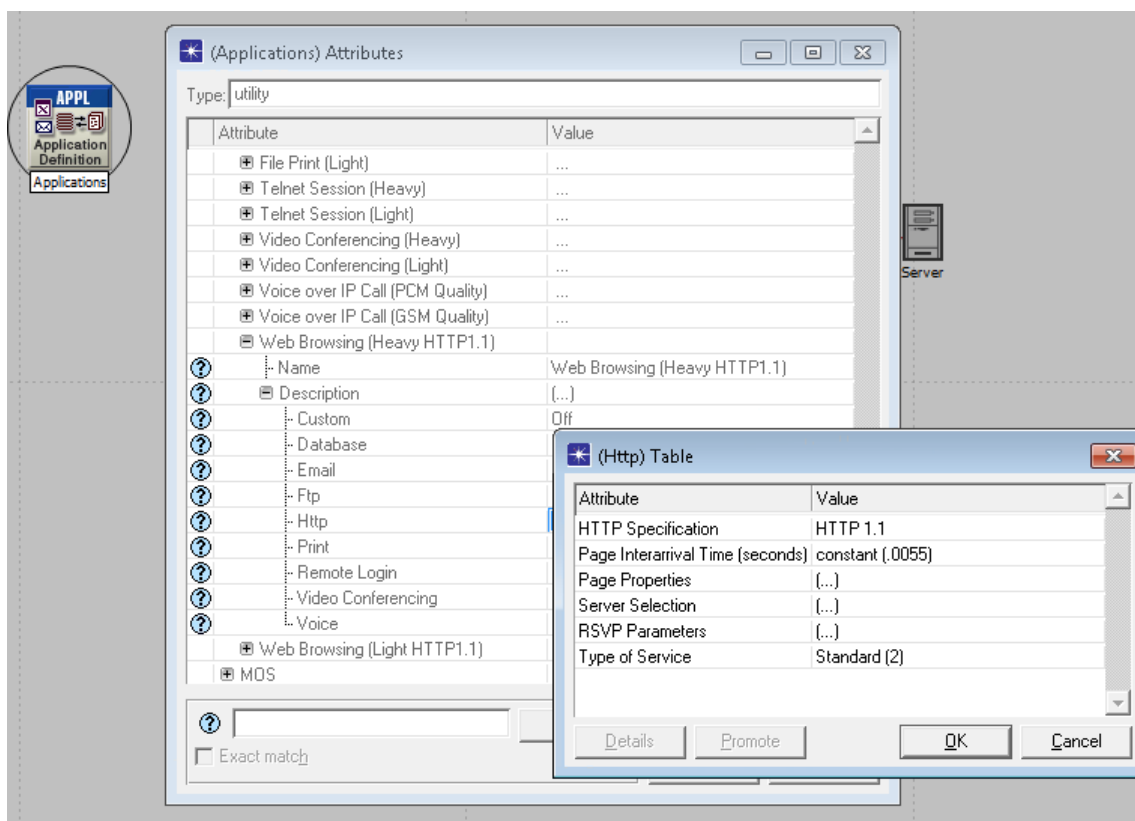
C2.

Figure C2. Sample application definition attributes for OPNET.

Configuring the Application Profiles

Once the attributes of applications are configured for the traffic desired, the

application profile object is used to define the traffic patterns to apply to the nodes in the

network. The application profile defines the application definitions to implement, when

they start, how long they last, how often they repeat, etc. A single profile can consist of a

number of applications running in serial or parallel to simulate specific traffic patterns. In

the sample shown in Figure C3 the only application supported is the same web browsing
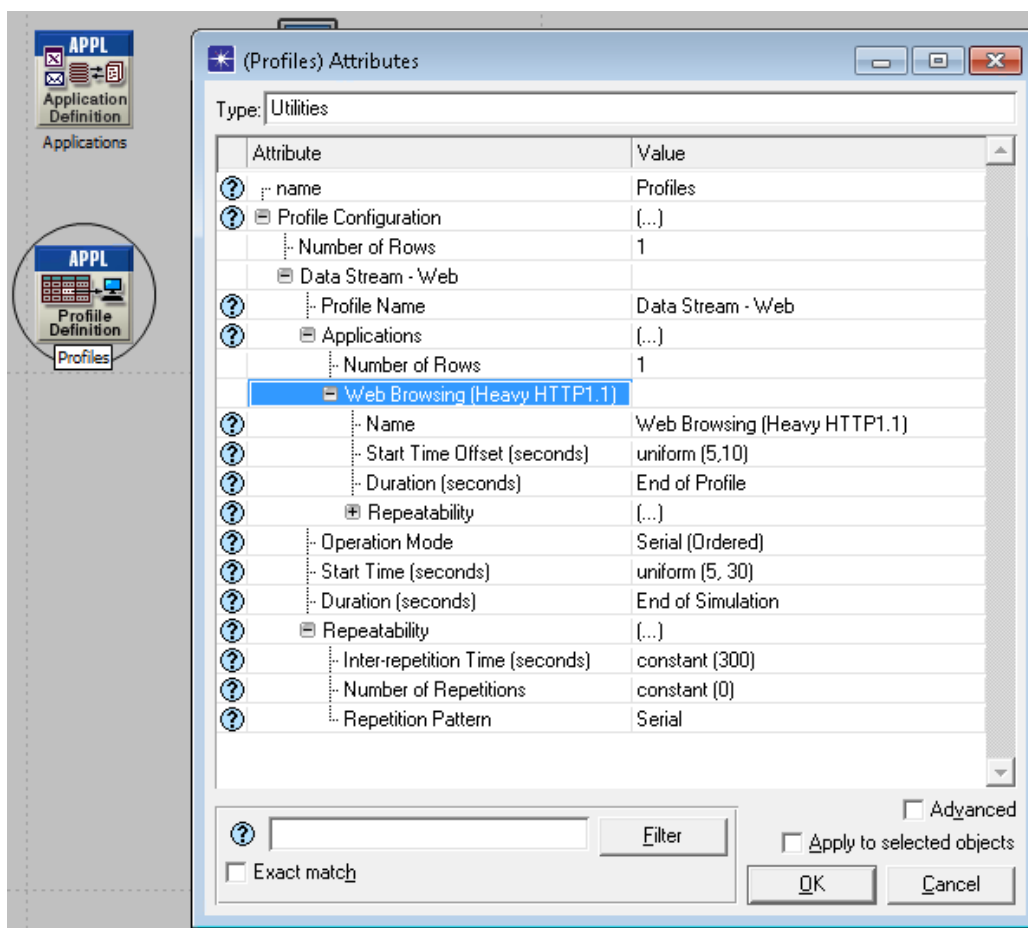
shown in Figure C2.

Figure C3. Sample OPNET application profile attributes.

Configuring the QoS Attribute Definitions

Because QoS is not always implemented on a network, the configuration of the

QoS attributes is completely optional. For the purposes of this research, this is a critical

component of the simulation environment. The OPNET simulation comes with a handful

of QoS profiles with attributes that can be configured similarly to the applications used as

seen in Figure C4. In the standard FIFO, the maximum queue size is 500 packets,

meaning that the most traffic that could be stored at any point in time is 500 packets. Any

traffic received after the queue is full would be dropped during the simulation. More

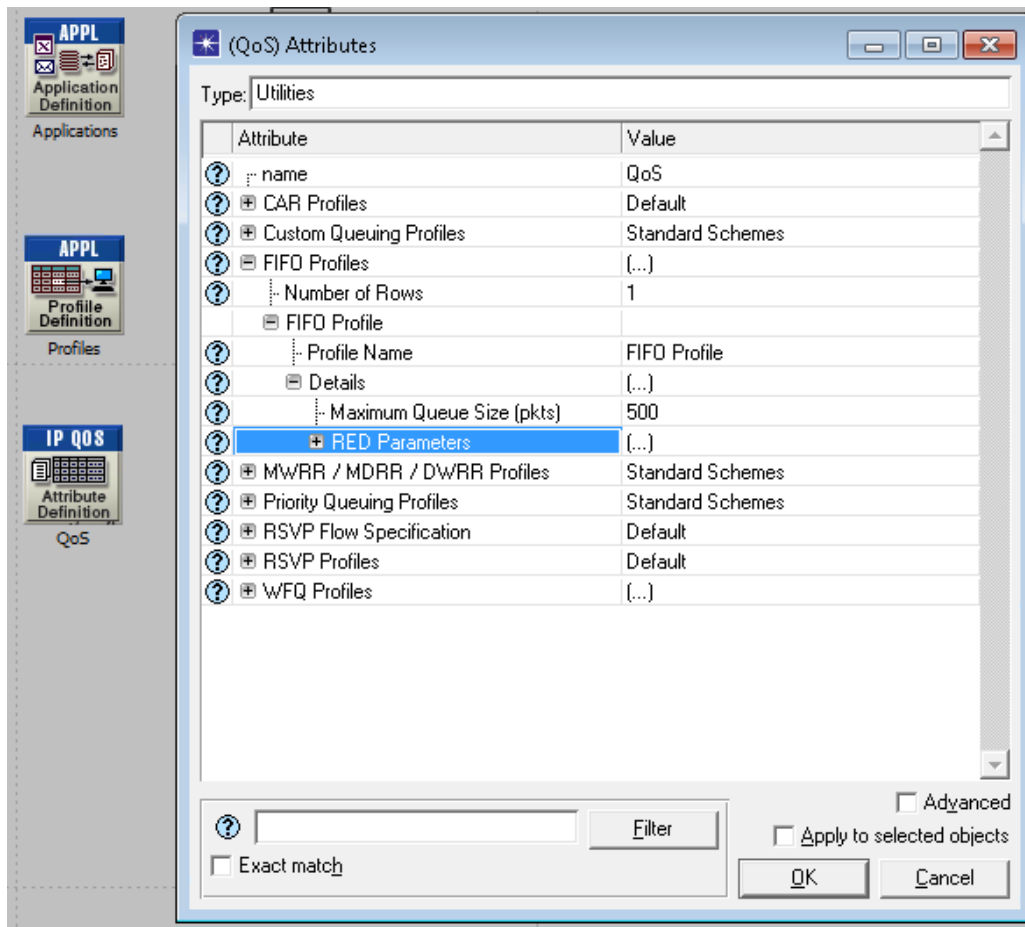information on QoS schemes will be covered later in this chapter.

Figure C4. Sample QoS attribute configuration.

Applying the Profiles and QoS to Network Devices

At this point, it is time to assign the application and QoS profiles to the network devices in the scenario to allow the network simulation to run with these settings. On each node, there are also many attributes that can be set. For the workstation nodes, the application profiles are assigned in the attribute panel for each node as shown in Figure C5.
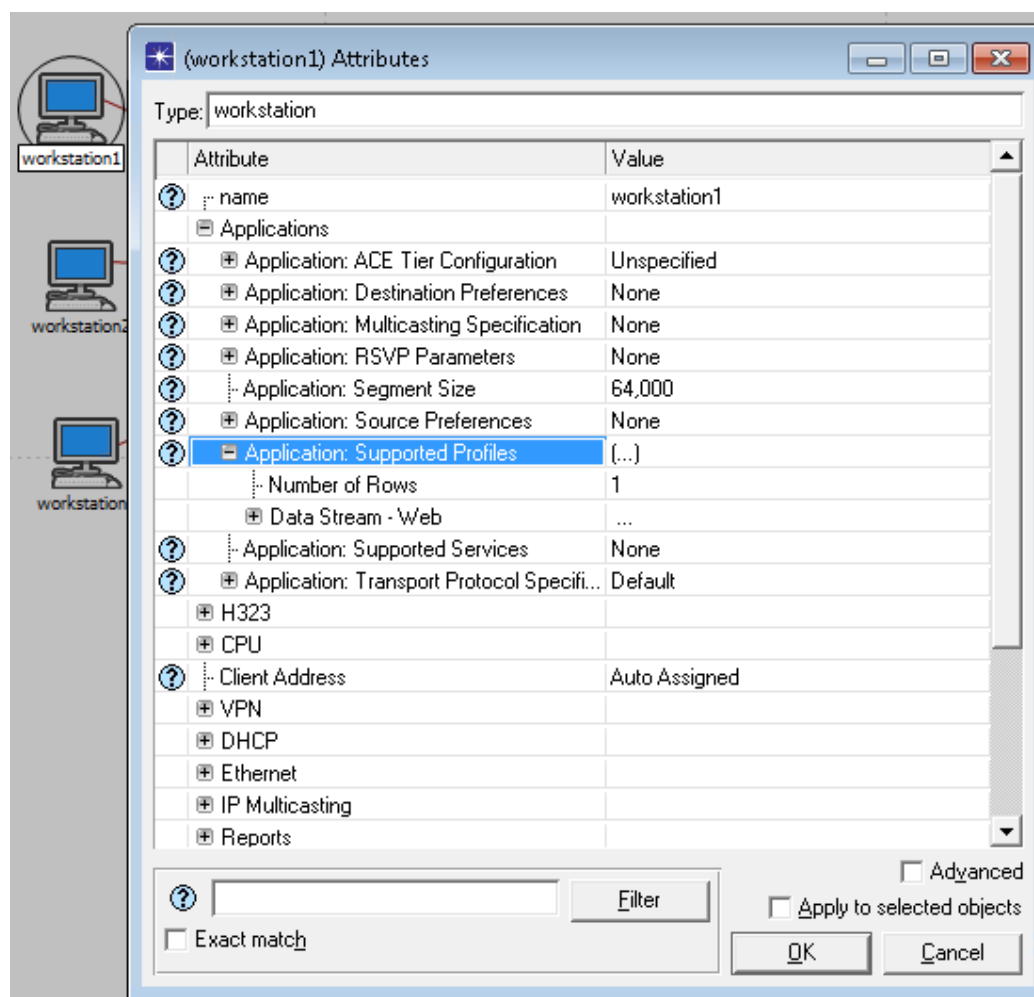

Figure C5. Sample OPNET node application attributes.

In order for the server in the model to respond to the traffic, it must also be configured to support the applications that have been defined in the application profiles assigned to the workstation nodes. For the example used, the server is setup to support the application Web Browsing (Heavy HTTP1.1) as shown in Figure C6.
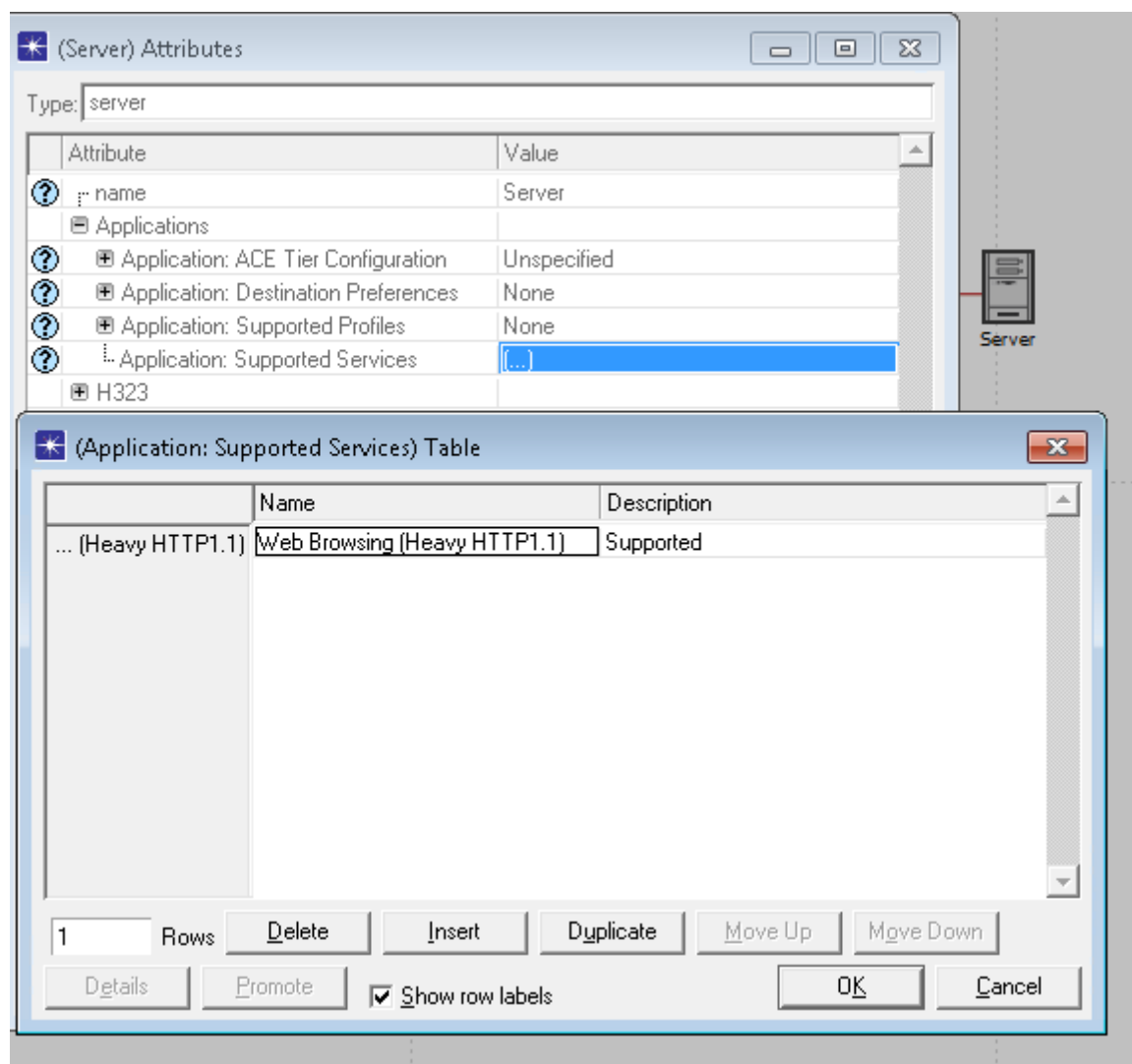

Figure C6. Sample OPNET server supported applications attributes.

A final item would need to be configured for QoS to be implemented on the

interfaces in the scenario. This can be done by using the attributes on each individual

node that has been defined or by using the menu option that allows the QoS profile to be

associated with every interface on the scenario as shown in Figure C7 As with any of the

attributes applied, they can be changed any time for comparison purposes.
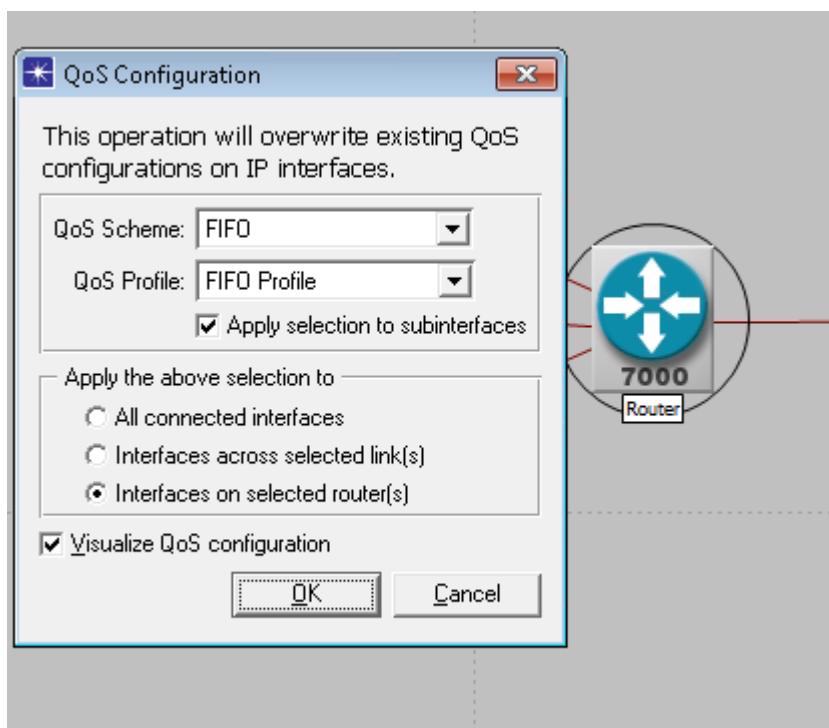

Figure C7. Sample OPNET applying QoS profiles.

<div align="center">Selecting Statistics for Collection</div>

At this point, the scenario is ready to run simulations using the Discrete Event

Simulator (DES). However, in order to perform an evaluation on the configuration, it is

necessary to configure the system to collect statistical information on the metrics that are

important to the situation under investigation. Depending on the applications and network

configuration, different statistics are available for selection. For example, if the purpose

of the simulation is to verify throughput on each network link in the simulation, statistics

can be selected as shown in Figure C8 that will allow that data to be collected during the

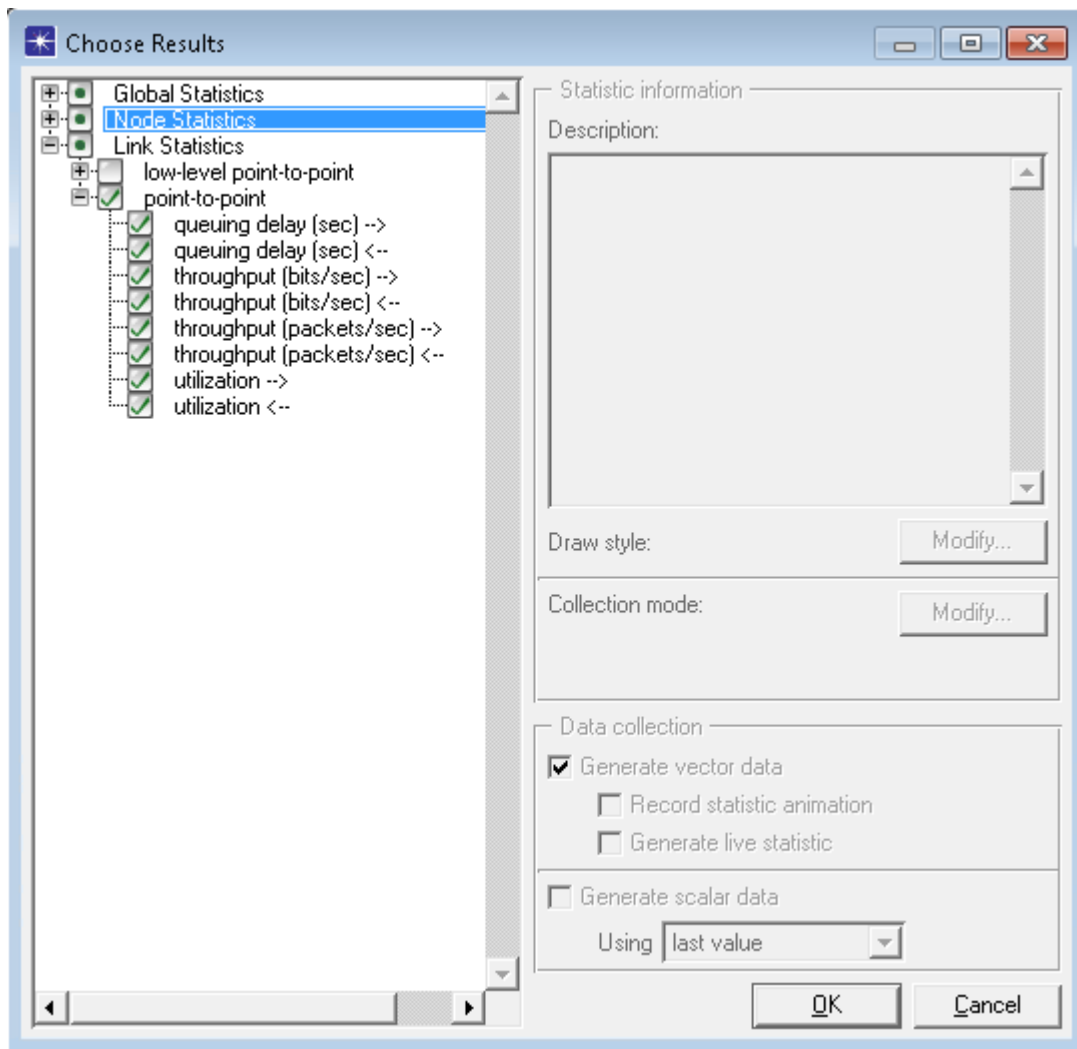simulation.

Figure C8. Sample OPNET statistics selection configuration.
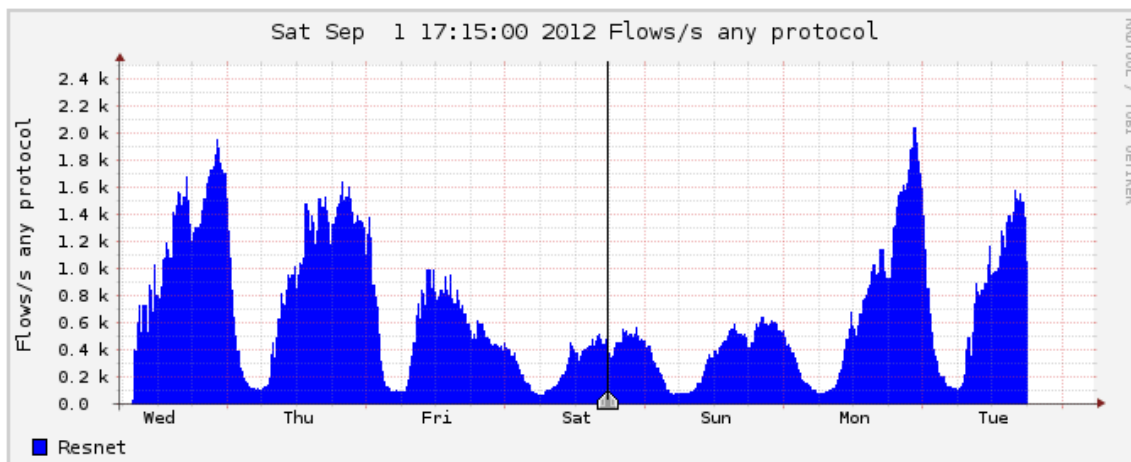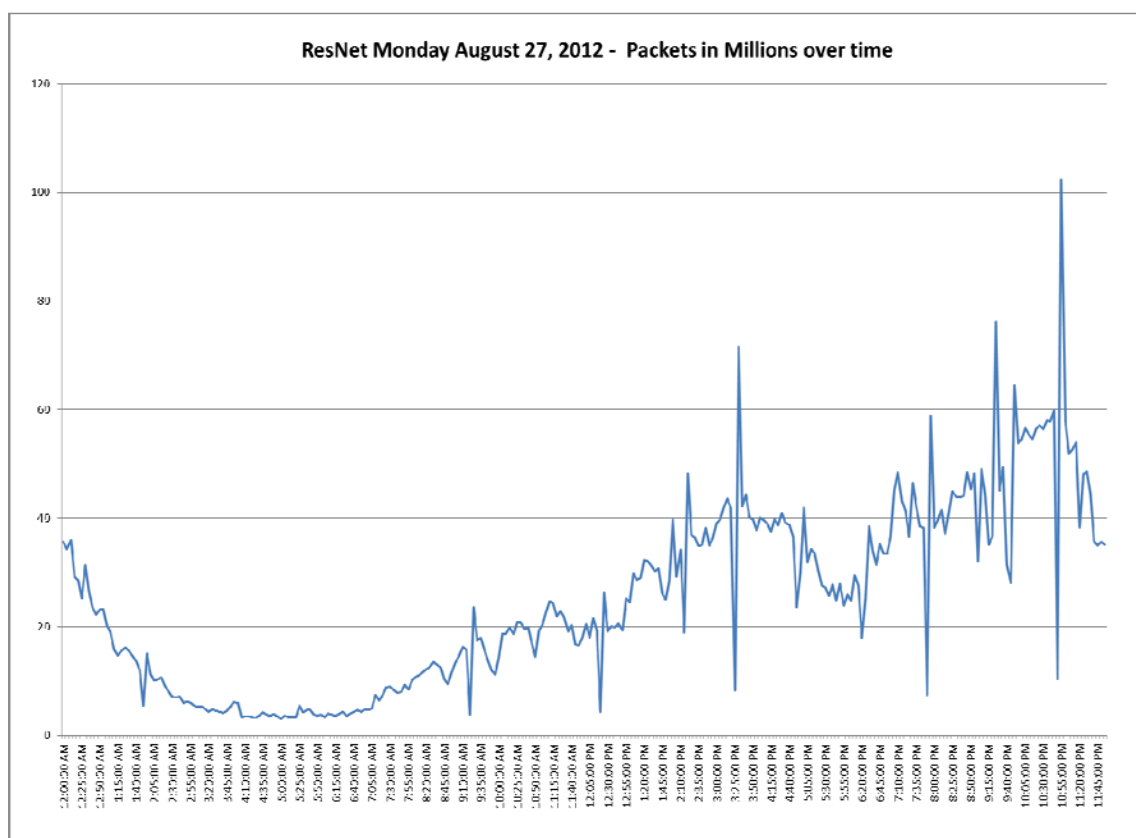
APPENDIX D

UNI GATHERED NETWORK DATA

Figure D1. Network flow for week of August 29,2012 to September 4, 2012.

Figure D2. Single day network traffic flow in packets.

APPENDIX E

OPNET MODELER CUSTOM CODE

```
static void
extract_and_send (void *state_ptr, int q_id)
        {
        OmsT_Qm_Queue_Pool* qpool_ptr;
        Packet*                 pkptr = OPC_NIL;
        double          pkt_svc_completion_time = 0.0;
        OmsT_Qm_Queue_Info*                        deq_info_ptr = OPC_NIL;
        OmsT_Qm_Queue_Info*     qinfo_ptr = OPC_NIL;
        OmsT_Qm_WFQ_Vars*       wfq_vars_ptr = OPC_NIL;
        /* Below Added by MLA */
        Log_Handle mla_log_handle;
        char ma_str[200];
        double ma_stamp_time;
        double ma_sim_time;
        int ma_sim_time_int;
        double          num_packets = 0.0;
        double          num_total_packets = 0.0;
        double mmsa_pkts_sec = 0;


        /** This function is called when the QM package is done processing a      **/
        /** packet. QM package interrupts this invoker with a process interrupt **/
        FIN (extract_and_send (void *state_ptr, int q_id));

        /* Get the qpool_ptr from the state ptr.                 */
        qpool_ptr = (OmsT_Qm_Queue_Pool *) state_ptr;

        /* Check whether the queue is now empty when RSVP is enabled.  */
        /* RSVP might have sent a request to delete this queue when          */
        /* this queue was in service.
        */
        if (qm_info->rsvp_flag == OPC_TRUE)
                {
                deq_info_ptr = qpool_ptr->queue_info_parray [q_id];

                /* If the queue had packets and got emptied, then     */
                /* the queue is removed after all packets are          */
                /* removed. But if the queue was empty when RSVP      */
```

```
                    /* sent the delete request, but was in service,          */
                    /* it needs to be deleted now.                                      */
                    if ((deq_info_ptr != OPC_NIL) && (deq_info_ptr->queue_to_delete ==
OPC_TRUE) &&
                            (oms_buffer_is_empty (deq_info_ptr->buffer_handle)))
                            {
                            /* Schedule an interrupt to delete this queue. */
                            op_intrpt_schedule_self (op_sim_time (), q_id);

                            /* Set the RSVP flag to indicate that a queue needs to be deleted.
         */
                            queue_to_be_deleted = OPC_TRUE;

                            qpool_ptr->queue_to_service = OMSC_QUEUE_DELETED;
                            }
                    }

        /* Dequeue next packet from the queue interface */
        Oms_Qm_Packet_Dequeue (&qpool_ptr, &q_id, &pkt_svc_completion_time,
OPC_NIL);

        /* Packet to be serviced will be NIL if */
        /* the buffer pool has been exhausted.          */
        pkptr = qm_info->sending_packet_ptr;
        mmsa_pkt_counter = mmsa_pkt_counter +1;
        if (pkptr != OPC_NIL)
                {

                /* Schedule the completion of this packet dequeue */
                output_iface_dequeue_schedule (pkptr, qpool_ptr,
pkt_svc_completion_time);

                /* Sends the packet.     */
                output_iface_packet_send (pkptr, q_id);

                if (mmsa_self_anneal == OPC_TRUE) {

                ma_stamp_time = op_pk_stamp_time_get (pkptr);
                ma_sim_time = op_sim_time();
                ma_sim_time_int = op_sim_time();

                /* mmsa_avg_pkt_delay = ((mmsa_avg_pkt_delay * mmsa_pkt_counter)+
(pkt_svc_completion_time - ma_stamp_time))/(mmsa_pkt_counter +1);*/
```

```
/*          for (queue_id = 0;
                    queue_id < qpool_ptr->attributes_ptr->no_queues;
queue_id ++)
                    {
                    qinfo_ptr                 = qpool_ptr-
>queue_info_parray[queue_id];
                    wfq_vars_ptr   = (OmsT_Qm_WFQ_Vars*) qinfo_ptr-
>qscheme_vars_ptr;

          /* If the subqueue is not empty we compare its head packet
     */
          /* to the current minimum finish time. If its head packet finish      */
          /* time is lowest it becomes the current minimum finish time.
     */
                    /*          num_packets =  oms_buffer_num_packets_get (qinfo_ptr-
>buffer_handle);
                    num_total_packets = num_total_packets + num_packets;
                    if (queue_id <3) {num_low_packets = num_low_packets +
num_packets;}
                    } */

     if (ma_sim_time_int > mmsa_set_checktime)
          {
          mmsa_set_checktime = mmsa_set_checktime + mmsa_interval_secs;
          mmsa_pkts_sec = mmsa_pkt_counter / mmsa_interval_secs;
          mmsa_pkt_counter = 1;
          switch (qm_info->mmsa_mode)
          {
          case MMSA_Standard_Queuing :
               if ((mmsa_pkts_sec >= mmsa_pkt_sec_threshold) &&
(mmsa_pkt_dropped < mmsa_pkt_dropped_threshold))
                    {
     qm_info->mmsa_mode = MMSA_Priority_Queuing;
          sprintf (ma_str, "switched to PQ %f",mmsa_pkts_sec);
                         op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");


                    }
               if ((mmsa_pkts_sec >= mmsa_pkt_sec_threshold) &&
(mmsa_pkt_dropped >= mmsa_pkt_dropped_threshold))
                         {
                    qm_info->mmsa_mode = MMSA_WFQ_Queuing;
                    sprintf (ma_str, "switched to WFQ %i",mmsa_pkt_dropped);
```

```
                              op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");


                    }
            break;
           case MMSA_FIFO_Queuing:
           if ((mmsa_pkts_sec >= mmsa_pkt_sec_threshold) &&
(mmsa_pkt_dropped < mmsa_pkt_dropped_threshold))
                         {
     qm_info->mmsa_mode = MMSA_Priority_Queuing;
           sprintf (ma_str, "switched to PQ %f",mmsa_pkts_sec);
                         op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");


                    }
            if ((mmsa_pkts_sec >= mmsa_pkt_sec_threshold) &&
(mmsa_pkt_dropped >= mmsa_pkt_dropped_threshold))
                              {
                    qm_info->mmsa_mode = MMSA_WFQ_Queuing;
                    sprintf (ma_str, "switched to WFQ %i",mmsa_pkt_dropped);
                         op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");


                       }

            break;

            case MMSA_Priority_Queuing :
            if (mmsa_pkts_sec < mmsa_pkt_sec_threshold)
                   {
                   qm_info->mmsa_mode = MMSA_FIFO_Queuing;
                   sprintf (ma_str, "switched to FIFO: %f",mmsa_pkts_sec);
                         op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");


                   }
             else
                     if (mmsa_pkt_dropped >= mmsa_pkt_dropped_threshold)
                            {
```

```
                            qm_info->mmsa_mode = MMSA_WFQ_Queuing;
                  sprintf (ma_str, "switched to WFQ %i",mmsa_pkt_dropped);
                              op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");


                          }

                  break;
                  case MMSA_WFQ_Queuing :
                  if (mmsa_pkts_sec < mmsa_pkt_sec_threshold)
                          {
                          qm_info->mmsa_mode = MMSA_FIFO_Queuing;
                          }
                  else
                          if (mmsa_pkt_dropped < mmsa_pkt_dropped_threshold)
                                  {
                                  qm_info->mmsa_mode = MMSA_Priority_Queuing;
                                  }

                  default:
                          break;
                  }

        mmsa_pkt_dropped = 0;

/*              if (mmsa_set_checktime %(mmsa_interval_secs*10) == 0)
                      {
                mmsa_pkt_counter = 0;
                mmsa_avg_pkt_delay = 0;

                 sprintf (ma_str,"reset at %f",200);
                op_prg_log_entry_write_options
(mmsa_log_handle,OPC_LOG_ENTRY_OPTION_NO_NODE |
OPC_LOG_ENTRY_OPTION_NO_EVENT,ma_str,"info","dequeue");
                      }
 */

                  }
                  }
                  }

        FOUT;
        }
```