Cluster computer simulation of buffer sharing schemes under bursty traffic load

Javier Castillo
University of Texas-Pan American

CLUSTER COMPUTER SIMULATION OF BUFFER SHARING SCHEMES UNDER BURSTY TRAFFIC LOAD

A Thesis

by

JAVIER CASTILLO

Submitted to the Graduate School of
The University of Texas-Pan American
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2014

Major Subject: Electrical Engineering

CLUSTER COMPUTER SIMULATION OF BUFFER SHARING SCHEMES UNDER BURSTY TRAFFIC LOAD

A Thesis
by
JAVIER CASTILLO

COMMITTEE MEMBERS

Dr. Sanjeev Kumar
Chair of Committee

Dr. Yul Chu
Committee Member

Dr. Weidong Kuang
Committee Member

August 2014

ABSTRACT

Castillo, Javier, Cluster Computer Simulation of Buffer Sharing Schemes under Bursty Traffic Load. Master of Science (MS), August, 2014, 99 pp., 9 tables, 117 figures, references, 56 titles.

In this thesis it is first analyzed the effect that different Average Burst Length, buffer size or number of ports have on the performance in terms of packet loss ratio on shared memory network switches using Complete Sharing as baseline for the memory allocation scheme. Three different shared memory allocation schemes - Sharing with a Minimum Allocation (SMA), Sharing with Maximum Queue lengths (SMXQ), and Dynamic Threshold (DT) - are then analyzed under varied traffic conditions in order to determine the best configuration for each tested scenario.

Having determined the best configuration for each individual scheme under all the tested scenarios, DT scheme is then compared against SMA scheme, as well as SMXQ scheme in order to determine which of the conventional shared memory allocation schemes presents a lower packet loss ratio on each tested scenario.

A new shared memory allocation scheme referred to in this thesis as ‘Shortest Queue First’ (SQF) scheme is evaluated. SQF aims at decreasing packet loss ratio while maintaining fairness of memory utilization. This proposed scheme is subjected to the same traffic conditions as the other schemes mentioned above; a comparison is then drawn against the conventional scheme with the lowest packet loss ratio for each scenario in order to determine the extent to which packet loss ratio decreases for a switch utilizing the SQF scheme.

DEDICATION

The completion of my master studies would not have been possible without the love and support of my family. My mother Isabel Rodriguez, my father Javier Castillo, my aunt Rosa Rodriguez, my grandmother Isabel Rubio, and my brother Jorge Castillo, whom wholeheartedly inspired, motivated and supported me by all means to accomplish this degree. My lovely fiancée, Wendy Hernandez, for the countless hours spent helping me and for always believing in me. Thank you all for your love and patience.

ACKNOWLEDGEMENTS

I will always be grateful to Dr. Sanjeev Kumar, chair of my thesis committee, for all his mentoring and advice. From NSF funding, research design, and data processing, to manuscript editing, he encouraged me to complete this process through his infinite patience and guidance.

My thanks go to my thesis committee members: Dr. Yul Chu, and Dr. Weidong Kuang. Their advice, input, and comments on my dissertation helped to ensure the quality of my intellectual work.

I would also like to thank my colleagues at the UTPA library who helped me locate supporting documents for my research. Also, I would like to acknowledge the many volunteers who participated in the focus group research.

Work in this thesis is supported in part by NSF's Major Research Instrumentation Program Award #CNS-1040430.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Background

Due to the advent of cloud services, social networks, and streaming media in the last decade, there has been an explosion of the use of the Internet. Data shows a high and steady rate of traffic increase [1] with no signs of slowing down in the near future, with an estimated compound annual growth rate (CAGR) of 35 percent through 2017 [2], or a 4.5 fold increase on IP traffic over the next five years.

Access to the Internet has become more ubiquitous, resulting in an increase of the number of users [4] thanks in no small part to the wide spread use of mobile devices such as tablets and smartphones, with an estimated CAGR of 66 percent through 2017, two thirds of which are expected to be video traffic by 2017 [3]. Next generation networks will have to handle a large amount of video traffic as IPTV becomes widely accepted by consumers [40]. Estimates show that by 2015 most of the video traffic will migrate to a service delivery almost entirely through packet switched networks such as the Internet [5].

Progress in optical transmission technologies such as dense wave division multiplexing (DWDM), optical add-drop multiplexers, ultralong-haul lasers, and optical amplifiers, have allowed for lower costs of digital transmission [7]. DWDM has allowed for fiber optics to double their bandwidth every seven to eight months [8]; however, this cycle is poised to slow down as we approach the maximum theoretical fiber capacity of 100 Tera-bits per second [9], making it important for future designs of packet switches and routers to handle traffic more efficiently in order to reduce packet loss, thus avoiding traffic increase due to the need for retransmission of lost packets.

In the following sections a few basic concepts are explained in an effort to facilitate the understanding of concepts more tightly related to this thesis introduced in Chapter 2.

1.2 Circuit Switching and Packet Switching

There are two different technologies available for the transmission of data between end users, namely circuit switching and packet switching. Circuit switching was the first one to be developed and it is still in use to this day, the telephone network being the most common example. In circuit switching a dedicated connection is established between both users for the duration of the transmission. At each node the data is retransmitted to its corresponding outgoing channel without delay [1][4].

On the other hand, we have packet switching, the most common example being computer-to-computer communications, which the basis of the Internet. In packet switching there is no dedicated channel between users, instead data is split into smaller chunks referred to as packets, and sent out through the network from node to node along a path that ultimately leads to their destination. At each node the packets are stored for a short period of time while its destination is being determined before it is sent out through the appropriate outgoing channel [1][4].

1.3 Routers and Switches

Computer networks allow for the transmission of data amongst connected devices, one common example is the Internet. A network is a web-like series of connections or links such as fiber optics, twisted pairs, or radio waves that provide a medium for data to be transmitted. The points where multiple links intersect are known as nodes, typically devices such as routers and switches, and they are responsible for receiving data and retransmitting it along the proper links, allowing data to reach its final destination.

1.3.1 Routers

Routers are devices that share physical and logical connections with multiple networks, and serve as a point of connection that provide three fundamental services. The first service is to calculate the best path that a packet should take through the network to reach its final destination, storing this information into its routing table. The second service provided by a router is to forward the packets received on an input interface to the proper output interface in order to be transferred across the networks. Finally, the third service is to provide a temporary storage via memory buffers for the packets when the arrival rate at the router's input interfaces is greater than the supported departure rate at the target output interface.

A router must be assigned at least two IP addresses, one for each network the router is part of. The reason behind this is because an IP address does not identify a specific computer, but the connection between said computer and a network. Hence, a device like a router connected to several networks requires an IP address for each network [4].

1.3.2 Switches

A switch is comprised of multiple ports, each one of them potentially connected to a single computer, which allows connected devices to send frames to each other. Switches consist of a series of intelligent interfaces, each connected to a single port, as well as a central fabric that provides simultaneous transfer between any two ports. Intelligent interfaces consist of a processor, memory, and the hardware required to receive any incoming packet, decipher its destination, and pass it on to the fabric for delivery, while also receiving packets from the fabric. Memory is available in the switch in order to allow the buffering of packets in instances when the arrival rate to the output port is greater than the available line speed allows outputting [4].

1.4 Buffering Strategies

Buffers are needed in switches because without them packets would be lost and have to be retransmitted, generating unnecessary extra traffic. When two or more arriving packets have the same destination in common only one of the packets is outputted, the remaining packets are stored to be outputted later. By implementing buffers in switches and routers packet loss is reduced, but at the same time average delay increases since packets received now spend time in the buffer.

1.4.1 Input Buffered Switches

In this type of switch, packets are queued at the input ports on an individual buffer belonging to each input port [13], where they wait for access to the switch fabric as shown in figure 1.1.

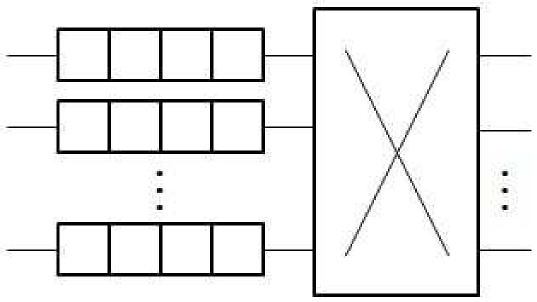


Figure 1.1-Input buffer switch diagram

The required bandwidth is equal to 2L [45][46], where L stands for the line speed. This type of switch is the simplest to implement, and it has the lowest bandwidth requirement, since each memory module is only required to carry out one read and one write operation during any given time slot. The downside however is its performance. Input buffer switches suffer from what is known as Head of Line (HoL) blocking, occurring when $k>1$ packets are destined to the same output, only one of them is allowed through the fabric, leaving $k-1$ packets waiting in the input queues. If more packets arrive on those queues waiting, then the rest of the packets received are blocked even though they may not precisely have as destination a busy output port [15][43], resulting in a maximum achievable throughput of 0.586 [13][45].

Because HoL blocking is only present in input buffer switches with FIFO queues, several mechanisms have been devised in order to mitigate or eliminate it by way of loosening the strict FIFO nature of the queues. Whenever HoL occurs, the most simple way to reduce it is by implementing a window policy [45][14], which allows the switch to look at the next packets queued inside a window for packets destined to a different output port. This method however is not effective under bursty traffic load.

A traditional method of reducing HoL blocking is Virtual Output Queueing (VOQ). This approach tackles the problem by implementing at the input ports a queue for each output port [17][18], effectively making this method analog to adding lanes to a street; VOQ provides a means for packets to pass packets being blocked. In addition to reducing HoL blocking and thus increasing throughput, VOQ also provides a certain degree of freedom from the strict FIFO nature of input queues, allowing for the implementation of varied scheduling strategies that reduce latency in a network [16][48]. The disadvantages of employing VOQ is that implemented at the network level it has very poor scalability, increasing the amount of memory required quadratically as the number of ports increases; whereas implemented at the switch level it no longer eliminates high order HoL blocking [16][47].

Destination Based Buffer Management (DBBM) [51] attempts to improve upon VOQ by maintaining a small number of queues at each input port, only a fraction of the number of ports in the switch. DBBM utilizes a mapping scheme that allows for an efficient utilization of the available resources, effectively reducing buffer requirements and improving scalability while maintaining a comparable performance to VOQ.

Regional Explicit Control Notification (RECN) [49][50][52] attempts to eliminate HoL blocking while reducing the requirement for the number of queues. This method was designed with networks with source-based routing and virtual cut-through switching in mind, although it requires the implementation of new logic at each switch. RECN is able to identify and sort congested traffic from non-congested traffic. It places non-congested traffic in normal queues located at each port, and congested traffic in dynamically allocated Set Aside Queues (SAQ), thus preventing long flows from blocking other incoming packets to that particular input queue.

1.4.2 Output Buffered Switches

This type of switch consists of an individual buffer located at each of the output ports where packets are stored while they wait to be transmitted into the media [20] as shown in figure 1.2. No two cells destined to different output ports are allocated in the same queue [21], meaning that in the case where k packets are destined to the same output port one packet is transmitted on the output line while k-1 packets are held on the buffer waiting for transmission [43].

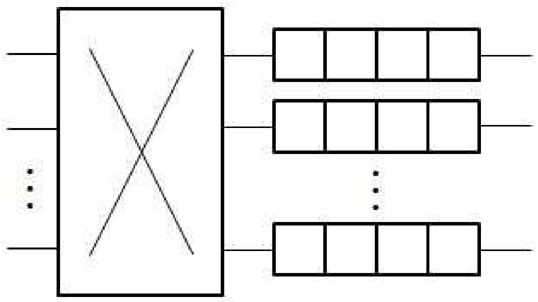


Figure 1.2-Output buffer switch diagram

Output buffer switches require more memory than input buffer switches, as well as a higher memory bandwidth, because in a worst case scenario there are N write operations and one read operation [46], requiring a memory bandwidth of L*(N+1). Output buffer switches do not need to implement complicated logic, utilize simple network resources management, and present no throughput degradation, making them capable of achieving a maximum throughput of 1 due to their non-blocking nature. These switches however, exhibit high cell loss under bursty traffic conditions [19].

1.4.3 Shared-Memory Switches

This type of switch is a variation of an output port switch in which all queues are combined into a single shared pool [43] as shown in figure 1.3.

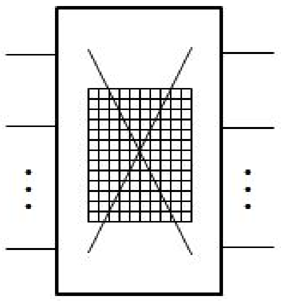


Figure 1.3-Shared-memory switch diagram

Memory is allocated to any output port in a first come first served basis until the buffer becomes overflown, in which case packets are dropped [26]. This characteristic however makes the switch unfair to inactive ports under bursty traffic loads, allowing highly active ports to occupy the majority of the available memory, effectively reducing throughput and increasing packet loss. This particular problem is tackled by setting rules or restrictions as to how memory might be allocated, some of which will be explained in detail in Chapter II.

The memory module in a shared-memory switch should be able to perform N reads and N writes in a single time cycle, requiring a memory bandwidth equal to 2NL [25][46], the fastest among all types of previously described switches. This switch also requires the use of a more complex controller to handle memory assignment [37], it does however lower the amount of memory required to provide a low packet loss ratio compared to both input and output buffer switches.

Parallel Packet Switch (PPS) [32][33] provides a way to overcome memory bandwidth limitations, allowing packet buffers to run slower than the line speed by increasing parallelism. Another way to implement shared-memory switches is with a Multistage Interconnection Network (MIN) [34][35] in which switching is done in stages, providing parallel access to memory read and write operations, mitigating the bottleneck produced when using only one memory module [38]. By switching in stages, an extra internal queueing delay is introduced, resulting in a throughput degradation of up to 2% under random traffic conditions [36][38]. Complete Sharing with Virtual Partitions (CSVP) [30] provides a more dynamic and adaptable implementation, deploying multiple memory modules and then virtually sharing them, effectively creating a single memory unit. In CSVP, the united buffers act like Complete Sharing when the buffer is not occupied; however, CSVP employs a push-out mechanism when the buffer is occupied, discarding cells already stored in the buffer when the arriving packet is destined to a queue that has not reached a virtual threshold. When subjected to heavy loads, CSVP behaves like an output buffer switch, allocating the same amount of memory to each output port. The Sliding Window [36][39] uses a similar approach to CSVP, allowing for a memory allocation that may range from completely partitioned to completely shared depending on the restrictions imposed on the switch.

1.5 Motivation and Problem Statement

As router and switch bandwidth increases at a faster pace than Moore's law [10], at 2.2 every 1.5 to 2 years due to advances in architecture and packet processing, the memory becomes the main limiting factor on switches and routers, as the quantity of memory available [12], the speed of the memory [11], as well as memory bandwidth [42] are not able to scale up fast enough to meet the increasing traffic demand.

In an effort to mitigate the limitations imposed on switches by its buffers, advances have been made with regard to the efficiency with which memory is utilized in order to maximize performance. In this thesis several conventional memory allocation schemes are subjected to a comparative performance evaluation under a wide array of scenarios in order to determine the optimal configuration for each of the different scenarios.

After extensive simulation it was found that the main factor contributing to improving the efficiency with which memory in a switch is utilized is fairness. It is observed that if a switch allocates memory to incoming packets in such a way that the distribution of memory amongst output ports is fair, throughput increases, and thus packet loss decreases. With this in mind, a new scheme is then proposed and subsequently subjected to the same scenarios and compared against the previously determined most efficient scheme, successfully proving this new scheme is optimal under each and all simulated scenarios.

1.6 Thesis Outline

In this thesis we test and evaluate four existing sharing memory schemes, subjecting them to a bursty traffic load under a wide array of scenarios in an effort to determine which of them has the best performance, a new scheme is then proposed and tested under the same scenarios in order to determine if there is an improvement with this new scheme.

Chapter I serves as an introductory chapter, providing explanation for a series of basic concepts in an effort to facilitate the comprehension of the materials covered in Chapter II and Chapter III, as well as defining the propose of the thesis. Chapter II contains the detailed descriptions of the simulated conventional sharing buffer schemes, as well as the scheme proposed in this thesis. Chapter III provides in-depth knowledge regarding how data was obtained, including descriptions of the traffic model utilized, how the buffer available in the switch was simulated, the scenarios under which the sharing memory schemes were tested, and the use of the cluster computer in order to speed up the process of gathering of data. Chapter IV first explores the effects that the variables ABL, number of ports and available buffer size have on switches using the scheme Complete Sharing (CS) as baseline; then the optimal configurations for the schemes Shared with Minimum Allocation (SMA), Shared with Maximum Queue lengths (SMXQ), and Dynamic Threshold (DT) are determined under each of the simulated scenarios. Chapter V compares the performance of SMA, SMXQ and DT in order to determine the best scheme for each of the tested scenarios. Chapter VI compares the best sharing memory scheme as defined in Chapter V to the proposed scheme Shortest Queue First (SQF) to demonstrate the improvement in performance it provides over the conventional sharing memory schemes. In Chapter VII we conclude this thesis and suggest possible future work.

CHAPTER II

CONVENTIONAL MEMORY SCHEMES AND THE PROPOSED SCHEME

In this chapter we provide a description for the conventional sharing memory schemes Complete Sharing (CS), Sharing with Minimum Allocation (SMA), Sharing with Maximum Queue lengths (SMXQ) and Dynamic Threshold (DT). We then propose a new scheme, Shortest Queue First (SQF) that tackles the issues described in the problem statement.

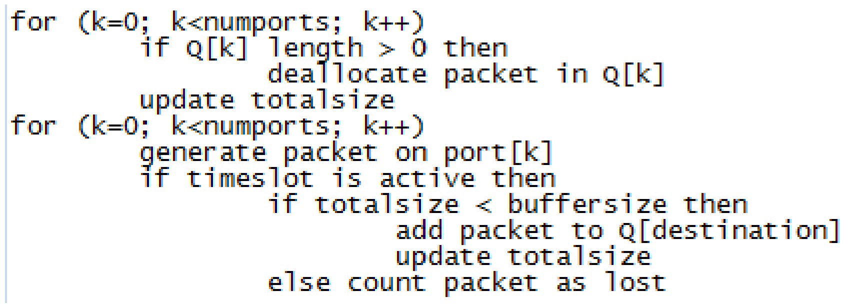
Out of the number of available conventional schemes, CS was chosen as baseline because it is the simplest form of sharing; SMA is also very simple, it is a hybrid between an output buffered switch and a shared-memory switch; DT is a very popular scheme because it dynamically adapts to incoming traffic conditions; SMXQ is DT’s static counterpart.

2.1 Conventional Shared Memory Schemes

2.1.1 Complete Sharing (CS)

This is the simplest form of sharing, there are no restrictions regarding how the memory is used, the memory is assigned to ports in a first come first served basis as long as there is storage space to be allocated [27][28][30][31]. This however tends to favor the most active ports, allowing them to occupy most of the available memory, so when a packet tries to reach an inactive port this packet will most likely be dropped.

Figure 2.1 shows the algorithm for the behavior of the scheme as it was modeled for simulation.



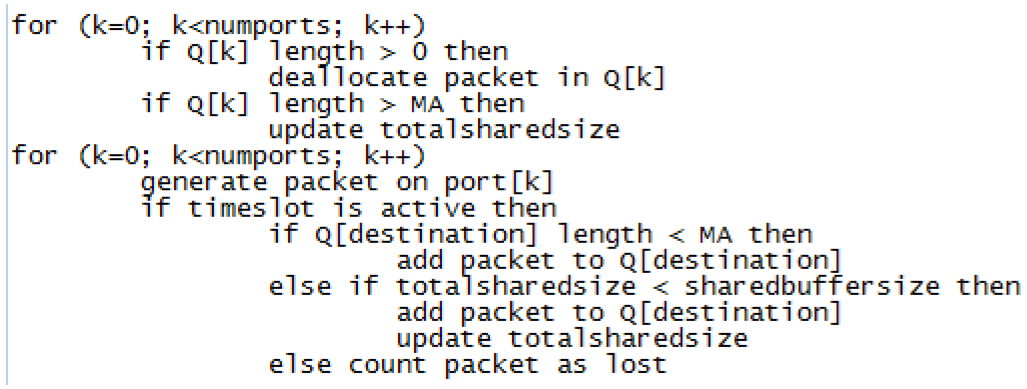
```
for (k=0; k<numports; k++)
        if Q[k] length > 0 then
                deallocate packet in Q[k]
        update totalsize
for (k=0; k<numports; k++)
        generate packet on port[k]
        if timeslot is active then
                if totalsize < buffersize then
                        add packet to Q[destination]
                        update totalsize
                else count packet as lost
```

Figure 2.1-Complete Sharing Pseudo-code

2.1.2 Sharing with a Minimum Allocation (SMA)

This scheme consists of two different memory segments. One is completely partitioned and allocated to each of the output ports. The other is completely shared, and just as with CS, this shared memory is allocated to ports in a first come first served basis. This set up allows inactive ports to have some memory available at all times, achieving fairness in the distribution of memory [27][28][30].

Figure 2.2 shows the algorithm for the behavior of the scheme as it was modeled for simulation.



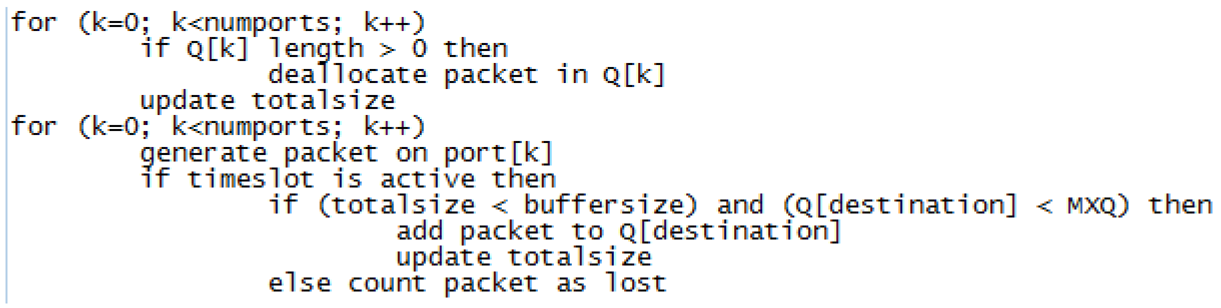
```
for (k=0; k<numports; k++)
        if Q[k] length > 0 then
                deallocate packet in Q[k]
        if Q[k] length > MA then
                update totalsharedsize
for (k=0; k<numports; k++)
        generate packet on port[k]
        if timeslot is active then
                if Q[destination] length < MA then
                        add packet to Q[destination]
                else if totalsharedsize < sharedbuffersize then
                        add packet to Q[destination]
                        update totalsharedsize
                else count packet as lost
```

Figure 2.2-Sharing with Minimum Allocation Pseudo-code

2.1.3 Sharing with Maximum Queue Lengths (SMXQ)

In this scheme the memory is allocated in a first come first served basis, but a static threshold is set in order to limit how big a queue may become at any given time, thus preventing the most active ports from hogging all the memory. In order for sharing to take place, the threshold times the number of ports must exceed the total available memory [27][28][30]. This particular scheme is not easy to configure, since setting a threshold too low will prevent sharing, whereas setting a threshold too high will not prevent extremely active ports from consuming most of the memory.

Figure 2.3 shows the algorithm for the behavior of the scheme as it was modeled for simulation.



```
for (k=0; k<numports; k++)
        if Q[k] length > 0 then
                deallocate packet in Q[k]
        update totalsize
for (k=0; k<numports; k++)
        generate packet on port[k]
        if timeslot is active then
                if (totalsize < buffersize) and (Q[destination] < MXQ) then
                        add packet to Q[destination]
                        update totalsize
                else count packet as lost
```

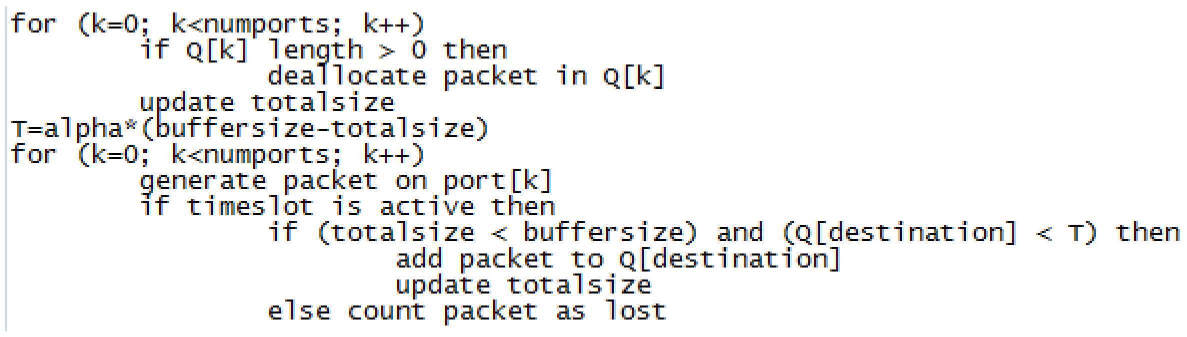
Figure 2.3-Sharing with Maximum Queue lengths Pseudo-code

2.1.4 Dynamic Threshold (DT)

The memory is allocated in a first come first served basis, but a limit is set as to how big a queue may become. Unlike SMXQ the threshold is dynamic, it changes every time slot depending on the amount of memory available at the moment, directly proportional to a factor of α [27][29][30][44]. If port-*i* occupancy is denoted by $Q^i(t)$ and it represents the length of the queue of port *i* at time *t*; the total occupancy $Q(t)$ represents the sum of every port queue lengths at time *t*; the total size of the shared buffer memory is denoted by *B*. Then the threshold $T(t)$ imposed on the queue lengths is given by the formula $T(t)=\alpha \cdot (B - Q(t))=\alpha \cdot (B - \sum_I Q^i(t))$. Each output port limits its queue length in such manner that $Q^i(t)$ never exceeds $T(t)$.

DT does not achieve full buffer occupancy in steady state. If *S* denotes heavily active queues and Ω denotes the occupied space in memory by queues below $T(t)$, then the amount of unused memory is given by $(B - \Omega) / (1 + \alpha \cdot s)$. The dynamic nature of this scheme allows it to adapt and respond to the ever-changing characteristics of the incoming traffic, while at the same time maintaining a reasonably small unused memory.

Figure 2.4 shows the algorithm for the behavior of the scheme as it was modeled for simulation.



```
for (k=0; k<numports; k++)
        if Q[k] length > 0 then
                deallocate packet in Q[k]
        update totalsize
T=alpha*(buffersize-totalsize)
for (k=0; k<numports; k++)
        generate packet on port[k]
        if timeslot is active then
                if (totalsize < buffersize) and (Q[destination] < T) then
                        add packet to Q[destination]
                        update totalsize
                else count packet as lost
```

Figure 2.4-Dynamic Threshold Pseudo-code

2.2 The Proposed Scheme (SQF)

Shortest Queue First (SQF) has the advantages of CS, not having any variable to adjust and being capable of achieving full occupancy of the shared memory space that is available to the switch. Like DT, SQF is able to adapt to incoming packet conditions, providing a fair allocation to inactive ports by giving them a higher priority, hence only discarding the packets destined to the longest queues when there is a buffer overflow. SQF has no restrictions regarding the size of queue lengths, however memory is not assigned in a first come first served basis, instead a priority list is kept in which a queue is given a higher priority the smaller its queue length; packets received during a time slot are accommodated on memory dependent on their priority, allowing inactive ports to be assigned memory locations when their packets arrive at its respective input.

Figure 2.5 shows the algorithm for the behavior of the scheme as it was modeled for simulation.

```
for (k=0; k<numports; k++)
        if Q[k] length > 0 then
                deallocate packet in Q[k]
        update totalsize
        populate priority list
        generate packet on port[k] and temporarly store
sort priority list
while (timeslot packets remaining) and (totalsize < buffersize) then
        add highest priority packet to Q[destination]
        if priority changes then
                update priority list
if timeslot packets remaining then
        count packets as lost
```

Figure 2.5-Shortest Queue First Pseudo-code

CHAPTER III

SIMULATION METHODOLOGIES

In this chapter we describe the model utilized in the generation of traffic, the way the buffer available in the switch is implemented, how the cluster computer helped with the acquisition of data, and the array of scenarios under which all sharing memory schemes were tested.

3.1 Bursty Traffic Model

As the Internet usage becomes more mobile, the average access time becomes smaller [6], having most users access the Internet in intervals of no more than 10 minutes. This makes the nature of the traffic highly bursty, which is the reason behind utilizing the model for this type of traffic in order to test the performance of the simulated buffer allocation schemes.

The bursty traffic is generated using a two-state ON-OFF model [36]. As shown in figure 3.1, it alternates between an idle period – a geometrically distributed period in which no packets arrive; and an active period – a geometrically distributed period in which packets arrive in a Bernoulli fashion. If p is used to represent the duration of the active period, and r is used to represent the duration of the idle period, then the probability that an active period lasts for a duration of i time slots is given by $P(i)=p\ (1 - p)^{i-1}$, for $i \geq 1$, and the corresponding average burst length (ABL) is given by $E_B[i]=1 / p$.

In a similar fashion, the probability that an idle period lasts for *j* time slots is given by $R(j)=r\,(1-r)^j$, for $j \geq 0$, and the corresponding mean idle period is given by $E_t[j]=(1-r)/r$. Hence, for a given *p* and *r*, the offered load *L* is given by $L=r/(r+p-rp)$.

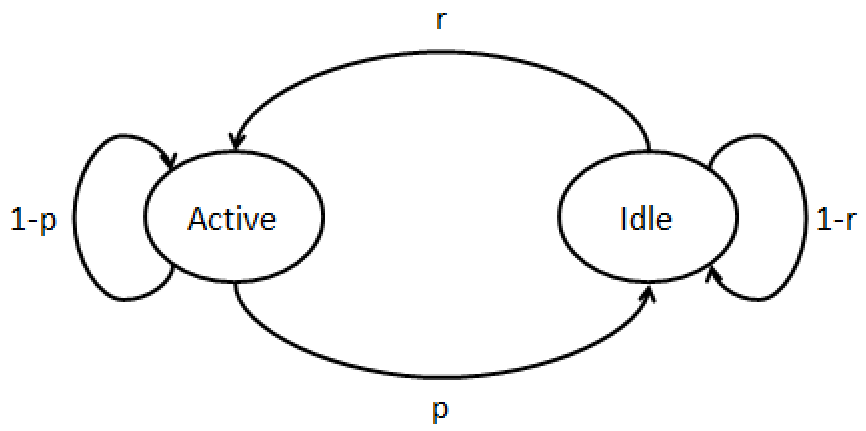


Figure 3.1-On/Off Model

3.1.1 Implementation

Defined as a class, it receives ABL and L as inputs, with that it is able to solve for the duration of active $p=1/ABL$, which in term is used to calculate the duration of idle $r=pL/(1+L(p-1))$.

For every time cycle a random number is generated in order to determine the next state. If the current state is Idle then the number is compared against *r*, if the random number is less than *r* then the state changes to active, otherwise it remains on idle. If the current state is Active then the number is compared against *p*, if the random number is greater than *p* then the state remains on active (same train), otherwise it changes to idle. However, once on idle a new random number is generated and analyzed, if this new random number generated is less than *r* then there is no idle period between trains, the states goes right back to active; if however, the random number is greater than *r* then the state remains on idle.

3.1.2 Functions Available

"initialize" receives as input variables ABL, load and number of ports in order to initialize the model; it determines what the starting state will be and outputs the corresponding packet.

"cycle" determines what the next state should be and generates the corresponding output.

"retout" returns the output generated for the current time slot, this output consists of a number ranging from 0 to the number of ports simulated where 0 represents an idle time slot and any other number represents an active time slot where the number represents its destination port.

"retL" provides the calculated Load *L = active / (actuve+idle)*.

"retABL" provides the calculated Average Burst Length *ABL = active / # of trains*.

"retactive" provides the number of active slots.

"retidle" provides the number of idle slots.

"rettrains" provides the number of trains or bursts.

3.2 Buffer

The memory or buffer available in switches is what allows for the storage of packets when more than one input port receives packets sharing as destination the same output port.

3.2.1 Implementation

The buffer is simulated as a group of singly linked lists or queues. One queue is used for each of the output ports in the switch, and the combined size of the queues is monitored to ensure that the total size does not exceed the maximum allowed.

Utilizing linked lists allows for a dynamic allocation of memory, providing for a more efficient use of the available resources while running the schemes on the cluster computer. Each element in the queue contains the number of time slots it has spent waiting inside the queue.

3.2.2 Functions Available

“retsize” returns the number of elements in the queue.

“retavgdelay” returns the calculated running average delay.

“add” adds an element at the end of the queue, sets the delay counter of that particular element to 0, and increments the size counter by 1.

“update” increments the delay counter on every element stored on the queue, updates the running average delay, deletes the element at the head of the queue and decrements the size counter of the queue by 1.

3.3 Cluster Computer

The reason for using the cluster computer was to reduce the total simulation time required. With the help of the cluster the time required for obtaining the results needed for this thesis was greatly reduced, since the cluster was able to run upwards of 300 jobs in parallel. The total number of jobs executed was 6930. The large amount of simulations allowed a wide array of scenarios to be analyzed, providing valuable information regarding the impact of the different variables that are part of the simulation.

3.3.1 Architecture

The cluster computer consists of 68 independent compute nodes, eigth nodes are used as a single Virtual SMP node, and four nodes are used for various managements tasks, making a total of 72 nodes. Each node has two Dual Intel 5650 2.67GHz processors and 48GB of RAM and a 250GB Disk. The system is connected via Mellanox Infiniband fabric (QDR) for fast data access. A diagram of the cluster computer is shown in figure 3.2.

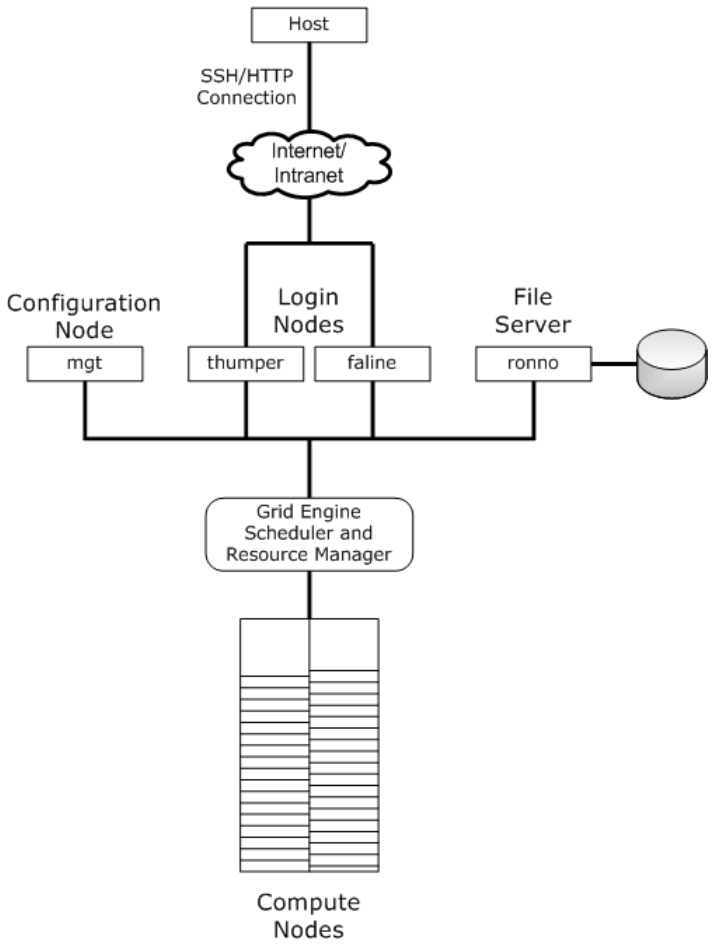


Figure 3.2-Diagram of Cluster Computer

The Sun Grid Engine scheduler is used to provide access to the compute nodes. Jobs are submitted with the ‘qsub’ command and their progress can be monitored with the ‘qstat’ command.

3.3.2 Job Submission

Two different interfaces are utilized to connect to the cluster. Gompute Xplorer provides an intuitive and user friendly interface that allows for the transfer of data between the user’s computer and the cluster, this program was used to upload the code to the cluster and afterwards download the data generated. In order to submit jobs to the cluster a Secure Shell (SSH) connection was established granting access to the cluster’s BASH shell.

After loading the code into the cluster, the command ‘g++ -o fileName fileName.cpp’ is inputted into the command line, this command compiles the code located in the file ‘fileName.cpp’ and generates the binary executable file ‘fileName’; the option ‘-o’ specifies where the generated file will be stored, in this case ‘fileName’ [53]. Once the required executable file has been generated, it is submitted to the cluster job scheduler to be assigned a node to run on by using the command ‘qsub –b y ~/path/fileName’, where the option ‘-b y’ indicates the format of the file being submitted is a binary, not a shell script [54]. Each job submitted for execution to the cluster generates a text file containing the data gathered for that specific run that can be then downloaded for analysis via Gompute Xplorer.

3.4 Simulation Scenarios

The simulated sharing memory allocation schemes were subjected to a wide arrange of scenarios including three different sizes of switches: 16, 32 and 64 ports; 7 different buffer sizes: 16, 32, 64, 128, 256, 512 and 1024 packets per port. For both SMA and SMXQ the minimum allocations and maximum queue lengths are 1/2, 1/4, and 1/8; while for DT the chosen alphas where 1, 2 and 4.

As for the traffic generated, the memory allocation schemes where subjected to 3 different average burst lengths: 64, 128 and 256 packets per burst, as well as loads ranging from 10% to 100% in step increments of 10%.

CHAPTER IV

DEFINING OPTIMAL CONFIGURATIONS FOR SMA, SMXQ AND DT

In this chapter we will first describe the effect that variables such as Average Burst Length (ABL), the number of ports on a switch, as well as the amount of memory in terms of cells—fixed size unit of memory—available for buffering in a switch have on the performance of a switch in terms of packet loss ratio, utilizing Complete Sharing (CS) as a baseline. We will then proceed to analyze the effect that variables particular to some of the sharing memory schemes have on the performance of the switch, namely Minimum Allocation (MA) for Sharing with a Minimum Allocation (SMA), Maximum Queue length (MXQ) for Sharing with Maximum Queue lengths (SMXQ), and Alpha for Dynamic Threshold (DT), this in order to determine the optimal configurations for each of these schemes under all tested scenarios. The last section of the chapter provides a summary of the results presented on this chapter.

4.1 Effect of Variables

There are a total of four variables not exclusive to any particular sharing memory scheme, these variables are the Average Burst Length (ABL), the offered load L, the number of ports and the total available buffer in the switch. We will analyze the effect the variables ABL, number of ports, and available buffer have over the complete range of offered loads L on a switch utilizing the sharing memory CS in order to determine the effect these variables have.

4.1.1 Effect of Average Burst Length

As it can be observed in figure 4.1, as the ABL increases so does packet loss. The reason behind this is that with higher ABL there is a greater tendency of arriving packets being director towards the same output port, effectively reducing the throughput of the switch, since a fewer number of ports being utilized translates into fewer packets outputted on a given time cycle. Having a higher arrival rate than departing rate causes the buffer in the switch to overflow, resulting in a higher packet loss.

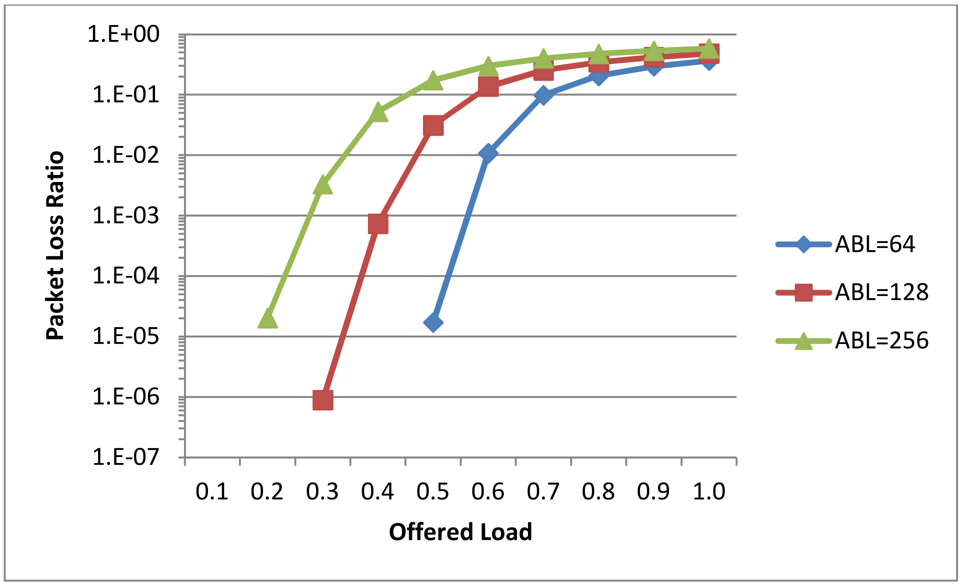


Figure 4.1-CS; Ports=64; Buffer=4096

4.1.2 Effect of Number of Ports

As it can be observed in figure 4.2, as the number of ports on any given switch increases so does the packet loss ratio. The reason behind this is that as the number of ports increases in a switch, so does the number of arriving packets, and as the buffer size remains the same and incoming traffic increments the buffer overflows more easily, thus resulting in a higher packet loss ratio.

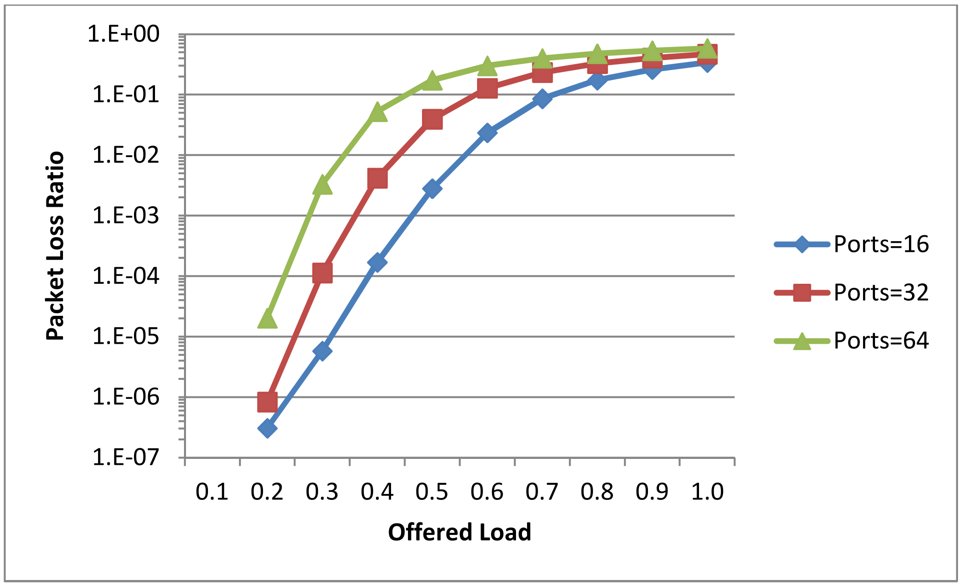


Figure 4.2-CS; Buffer=4096; ABL=256

4.1.3 Effect of Buffer Size

As with increasing number of ports, it can be observed on figure 4.3 that as the size of the buffer decreases the packet loss ratio increases. The reason behind this is that as the buffer available on a switch decreases the buffer overflows more easily, resulting in an increase in packet loss ratio.

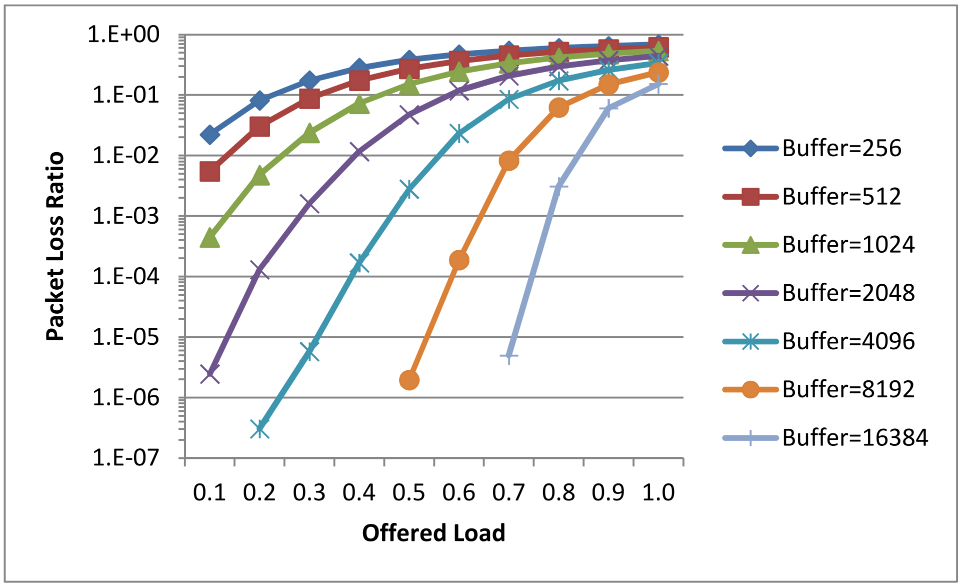


Figure 4.3-CS; Ports=16; ABL=256

4.2 Optimal Configurations

In this section we will determine the best configuration for the schemes SMA, SMXQ and DT under the different tested ABLs, number of ports, and buffer sizes.

4.2.1 Optimal Configuration for Different Average Burst Length

4.2.1.1 SMA. Under all tested scenarios this scheme performed consistently, with bigger Minimum Allocation (MA) presenting less packet loss under high loads and smaller MA presenting less packet loss under low loads, as can be appreciated in figures 4.4, 4.5 and 4.6. SMA with MA of 1/8 presents the best performance out of the three tested configurations under loads of up to 90%, presenting a packet loss ratio decrease of up to 3.15E-02, a 55.95% improvement over the other two configurations; while SMA with MA of 1/2 presented the best performance on loads of 100%, presenting a packet loss ratio decrease of up to 6.03E-03, a 2.39% improvement.

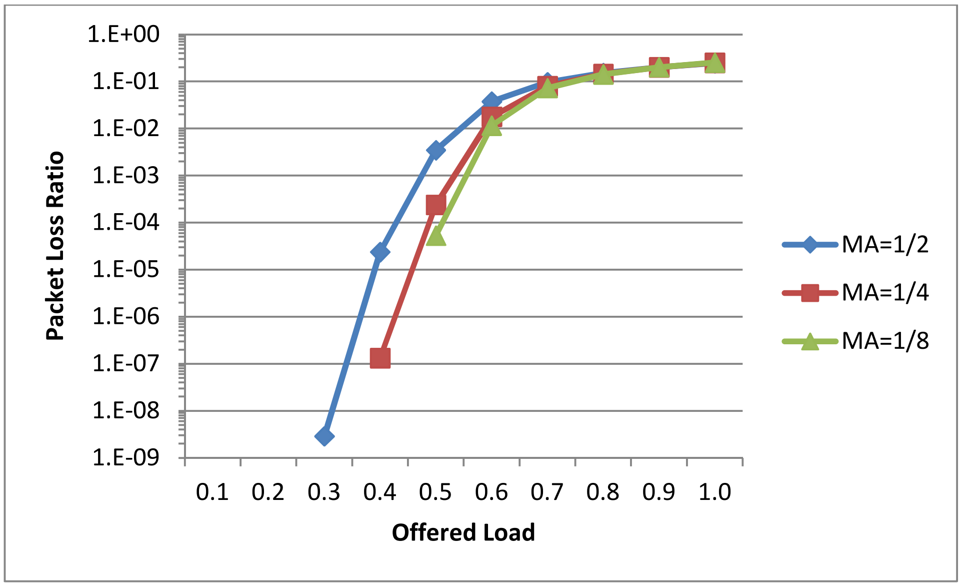


Figure 4.4-SMA; Ports= 64; Buffer=4096; ABL=64

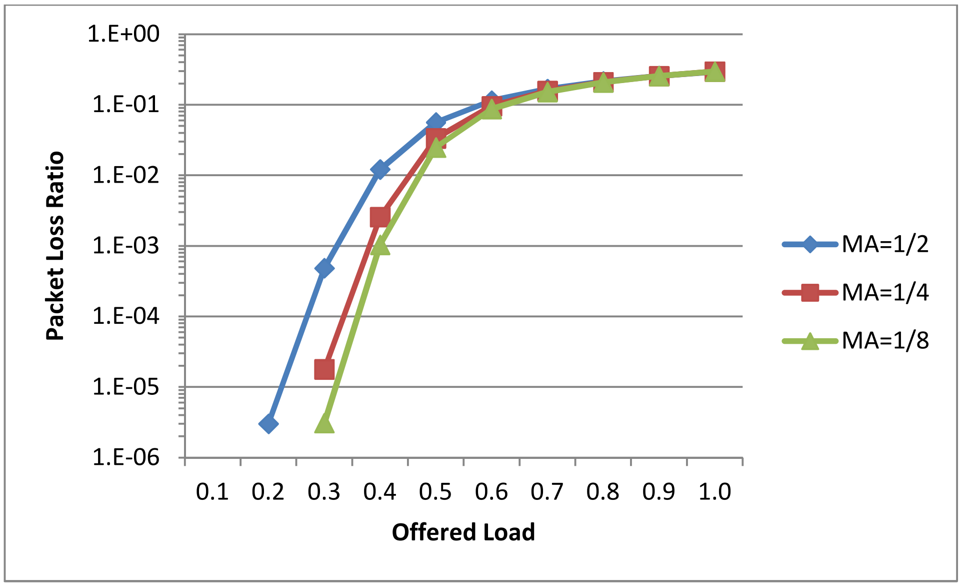


Figure 4.5-SMA; Ports=64; Buffer=4096; ABL=128

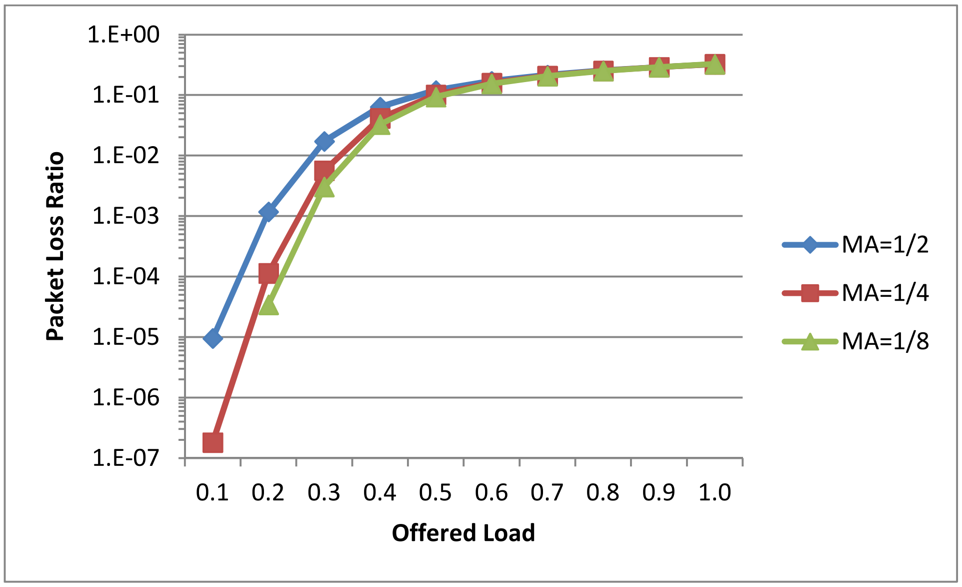


Figure 4.6-SMA; Ports=64; Buffer=4096; ABL=256

4.2.1.2 SMXQ. Under all tested scenarios this scheme performed consistently, with smaller Maximum Queue length (MXQ) presenting a lower packet loss ratio under high loads and bigger MXQ presenting a lower packet loss ratio under low loads, as can be appreciated in figures 4.7, 4.8 and 4.9. SMXQ with MXQ of 1/2 presents the best performance out of the three tested configurations under loads of up to 50%, presenting a packet loss ratio decrease of up to 1.45E-

02, a 82.30% improvement over the other two configurations; while SMXQ with MXQ of 1/8 presented the best performance on loads of 40% to 100%, presenting a packet loss ratio decrease of up to 9.85E-02, a 32.92% improvement.

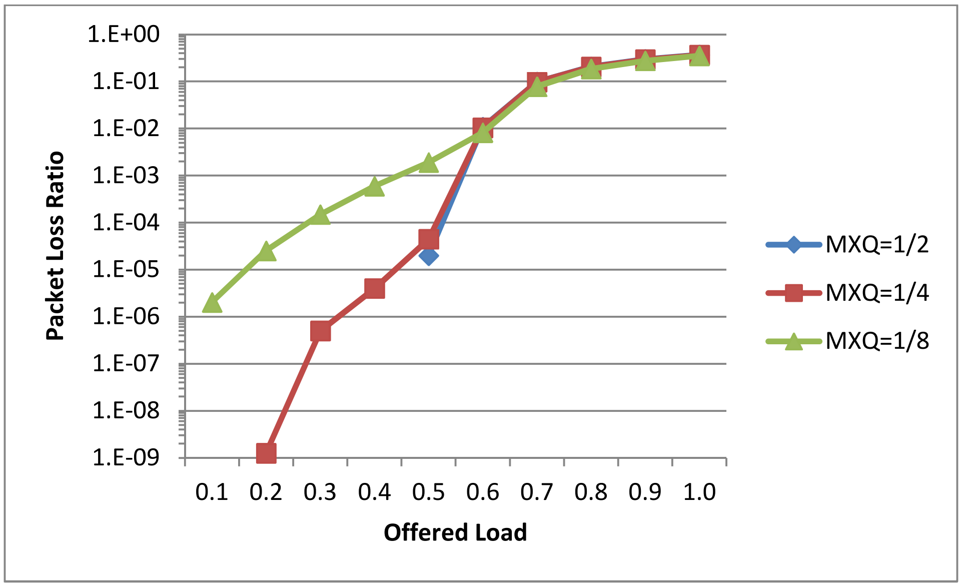


Figure 4.7-SMXQ; Ports=64; Buffer=4096; ABL=64

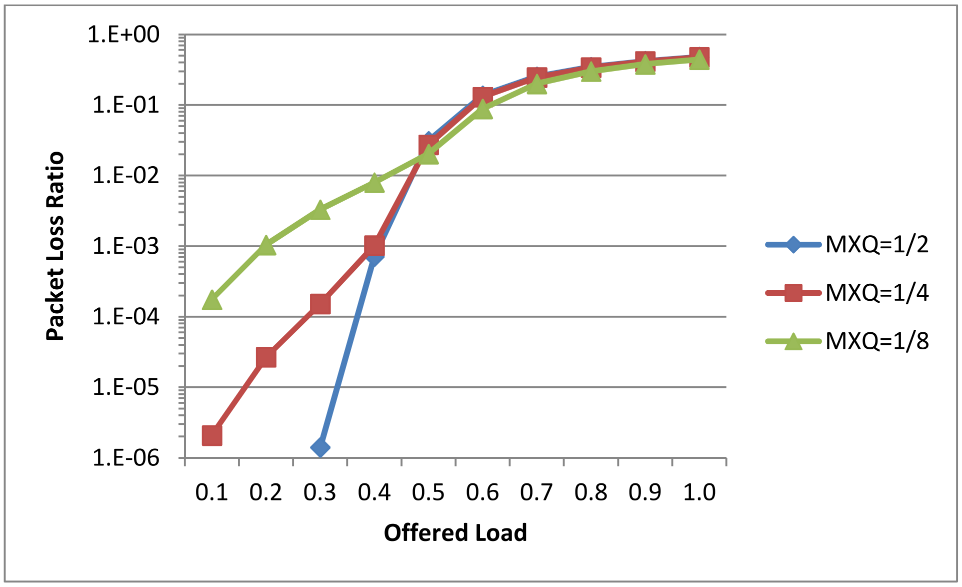


Figure 4.8-SMXQ; Ports=64; Buffer=4096; ABL=128

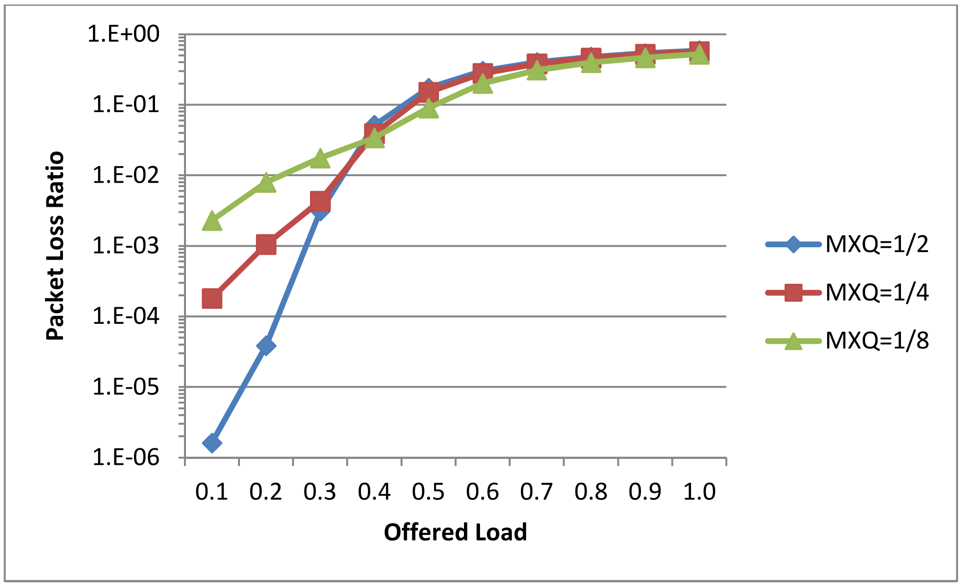


Figure 4.9-SMXQ; Ports=64; Buffer=4096; ABL=256

4.2.1.3 DT. Under all tested scenarios this scheme performed consistently, with larger Alpha presenting a lower packet loss ratio under low loads; under high loads Alpha equal to 2 performs better than the other configurations when the ABL is 64 and 128; Alpha equal to 1 performs better when the ABL is 256, as can be appreciated in figures 4.10, 4.11 and 4.12. DT with an Alpha of 4 presents the best performance out of the three tested configurations under loads of up to 80%, presenting a packet loss ratio decrease of up to 5.77E-03, a 22.13% improvement. DT with an Alpha of 2 presented the best performance on loads of 80% to 100% with an ABL of 64 and 128, presenting a packet loss ratio decrease of up to 4.45E-02, a 18.83% improvement; while DT with an Alpha of 1 presented the best performance on loads of 90% and 100% with an ABL of 256, presenting a packet loss ratio decrease of up to 2.90E-02, a 8.96% improvement.

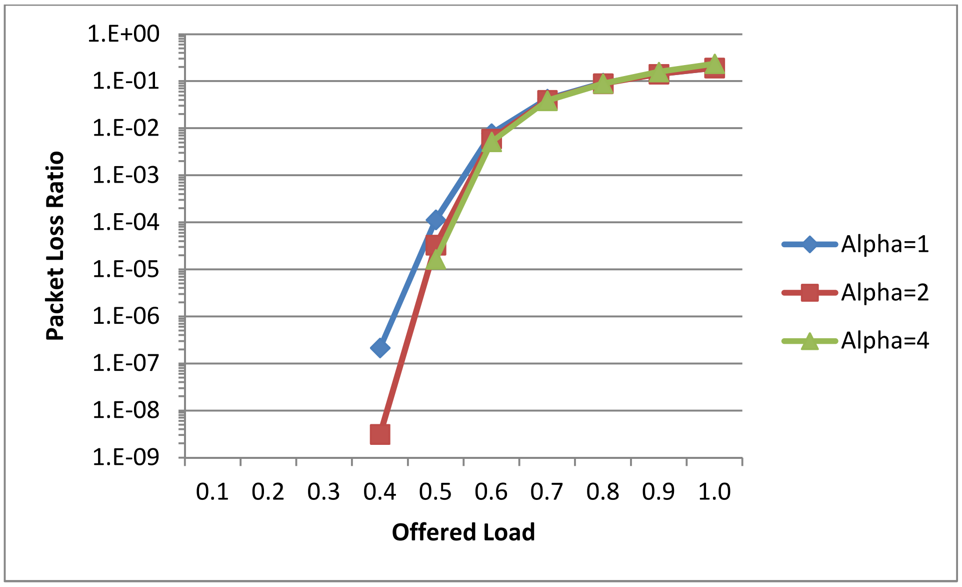


Figure 4.10-DT; Ports=64; Buffer=4096; ABL=64

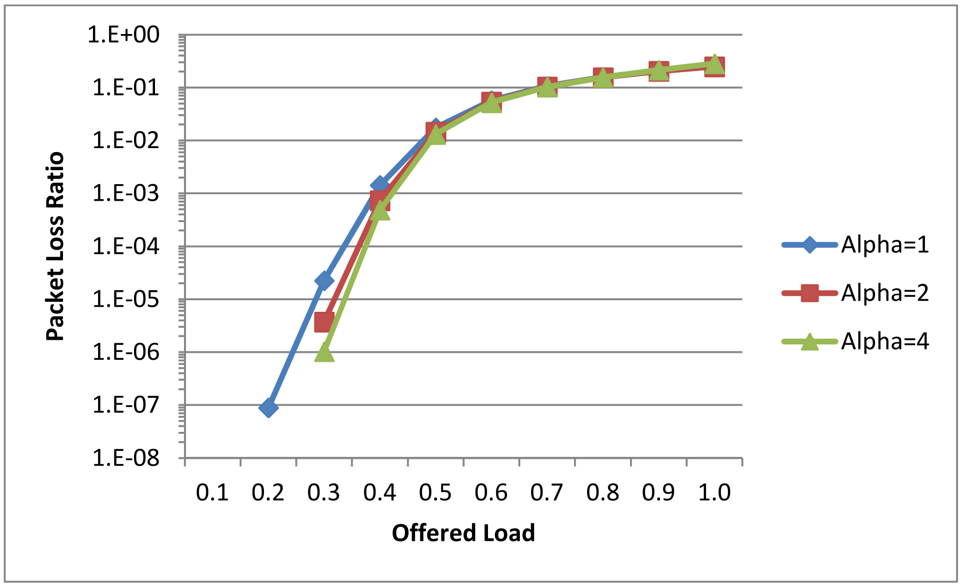


Figure 4.11-DT; Ports=64; Buffer=4096; ABL=128

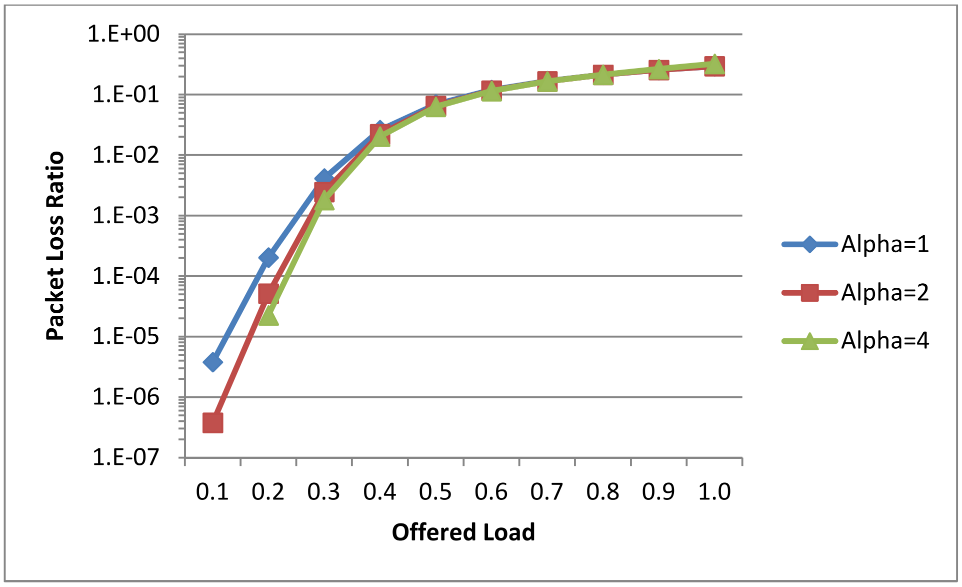


Figure 4.12-DT; Ports=64; Buffer=4096; ABL=256

4.2.2 Optimal Configuration for Different Number of Ports

4.2.2.1 SMA. Under all tested scenarios this scheme performed consistently, with bigger Minimum Allocation (MA) presenting less packet loss under high loads and smaller MA presenting less packet loss under low loads, as can be appreciated in figures 4.13, 4.14 and 4.15. SMA with MA of 1/8 presents the best performance out of the three tested configurations under loads of up to 90%, presenting a packet loss ratio decrease of up to 3.09E-02, a 48.51% improvement; while SMA with MA of 1/2 presented the best performance on loads of 100%, presenting a packet loss ratio decrease of up to 4.03E-03, a 1.70% improvement.

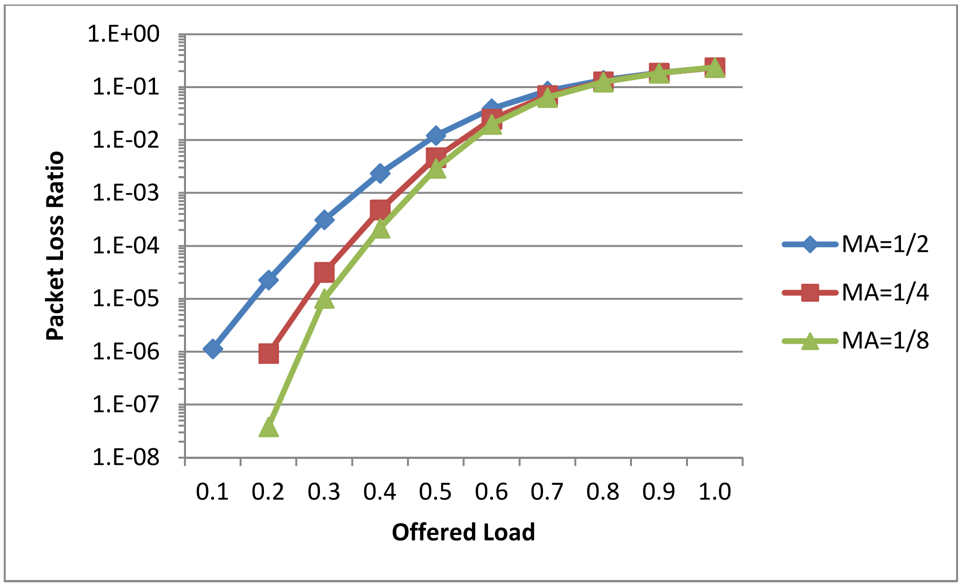


Figure 4.13-SMA; Ports=16; Buffer=4096; ABL=256

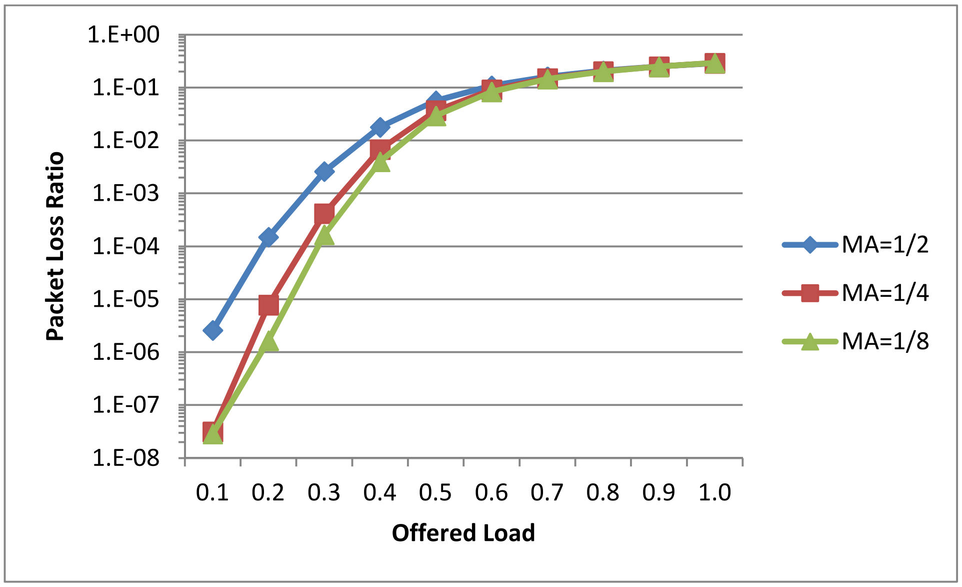


Figure 4.14-SMA; Ports=32; Buffer=4096; ABL=256

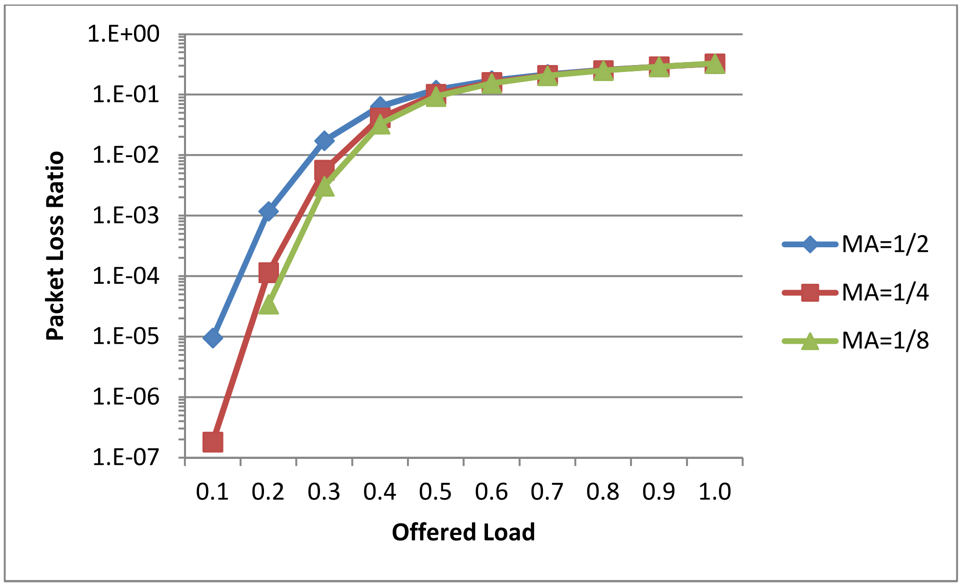


Figure 4.15-SMA; Ports=64; Buffer=4096; ABL=256

4.2.2.2 SMXQ. Under all tested scenarios this scheme performed consistently, with smaller MXQ presenting lower packet loss ratio under high load and bigger MXQ presenting less packet loss under low loads, as can be appreciated in figures 4.16, 4.17 and 4.18. For switches consisting of 16 ports SMXQ with MXQ of 1/2 presents the best performance out of the three tested configurations under loads of up to 60%, presenting a packet loss ratio decrease of up to 1.12E-02, an 80.11% improvement; SMXQ with MXQ of 1/4 presents the lowest packet loss ratio under loads of 70% and 80%, with a maximum packet loss ratio decrease of up to 5.14E-02, a 31.70% improvement; while SMXQ with MXQ of 1/8 presented the best performance on loads of 90% and 100%, presenting a packet loss ratio decrease of up to 1.33E-01, a 40.41% improvement.

For switches with 32 ports an MXQ of 1/2 exhibited the best performance while SMXQ was subjected to loads of up to 40%, with a maximum packet loss ratio decrease of up to 2.71E-02, an 88% improvement; MXQ of 1/4 performed the best only when the scheme was subjected to loads of 50% with a maximum drop in packet loss ratio of 2.26E-02, a 46.16% improvement;

lastly, SMXQ with an MXQ of 1/8 presented the best performance while subjected to loads ranging from 60% to 100%, with an improvement on packet loss ratio over the other configurations of up to 2.71E-02, an improvement of 88%.

For switches of 64 ports the optimal configuration required the use of an MXQ of 1/2 for loads of no more than 30%, showing an improvement in terms of packet loss ratio of up to 1.45E-02, an 82.30% improvement; while an MXQ of 1/8 was required for loads ranging from 40% to 100%, showing an improvement of up to 9.85E-02, a decrease of 32.92%.

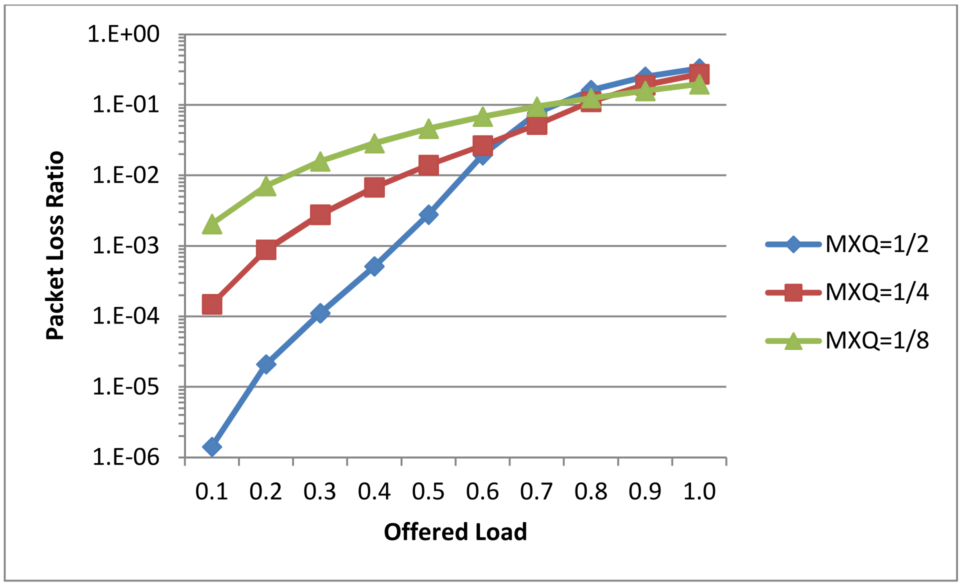


Figure 4.16-SMXQ; Ports=16; Buffer=4096; ABL=256

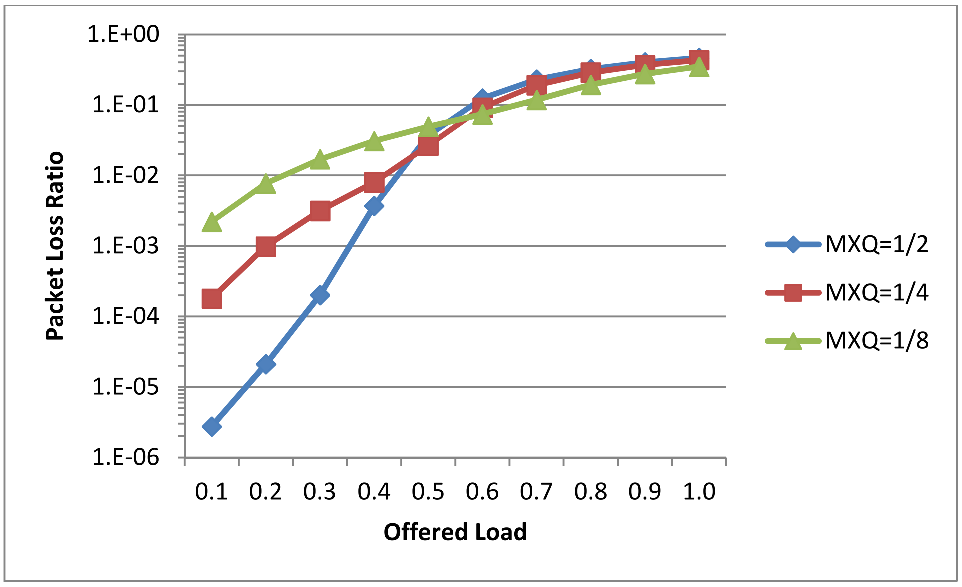


Figure 4.17-SMXQ; Ports=32; Buffer=4096; ABL=256

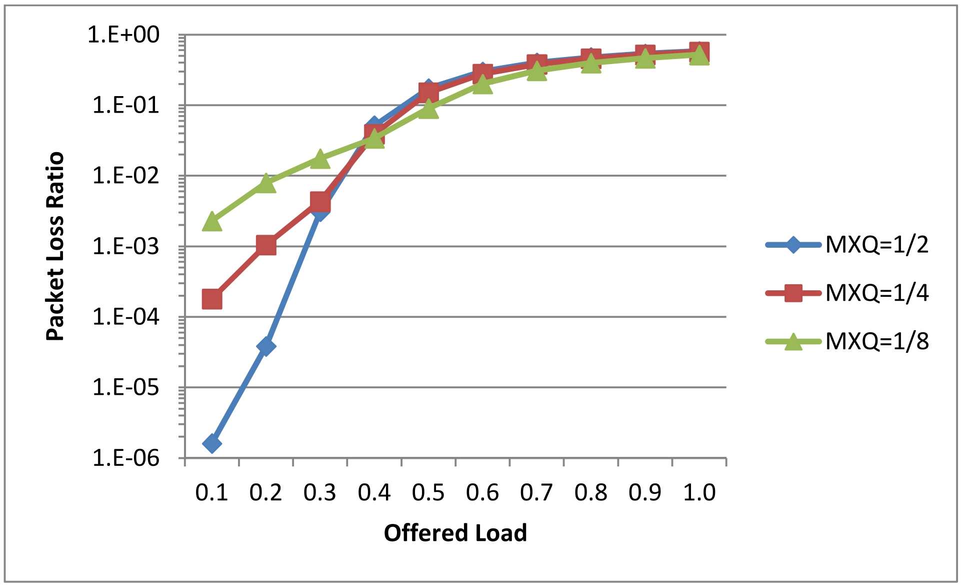


Figure 4.18-SMXQ; Ports=64; ABL=4096; ABL=256

4.2.2.3 DT. Under all tested scenarios this scheme performed consistently, with DT with an Alpha of 4 having the best performance on every scenario except for the case when the simulated switch consists of 64 ports and is subjected to loads of upwards of 90%, as can be appreciated in figures 4.19, 4.20, 4.21. DT with an Alpha of 4 presents the best performance out of the three tested configurations for switches with 16 and 32 ports, presenting a packet loss ratio decrease of up to 1.24E-02, a 21.45% improvement. For switches with a number of ports of 64 DT with an Alpha of 4 is the configuration with the lowest packet loss ratio for loads of up to 80%, presenting a packet loss ratio decrease of up to 5.77E-03, a 22.13% improvement; while DT with an Alpha of 1 presented the best performance on loads of 90% and 100%, presenting a packet loss ratio decrease of up to 2.90E-02, a 8.96% improvement.

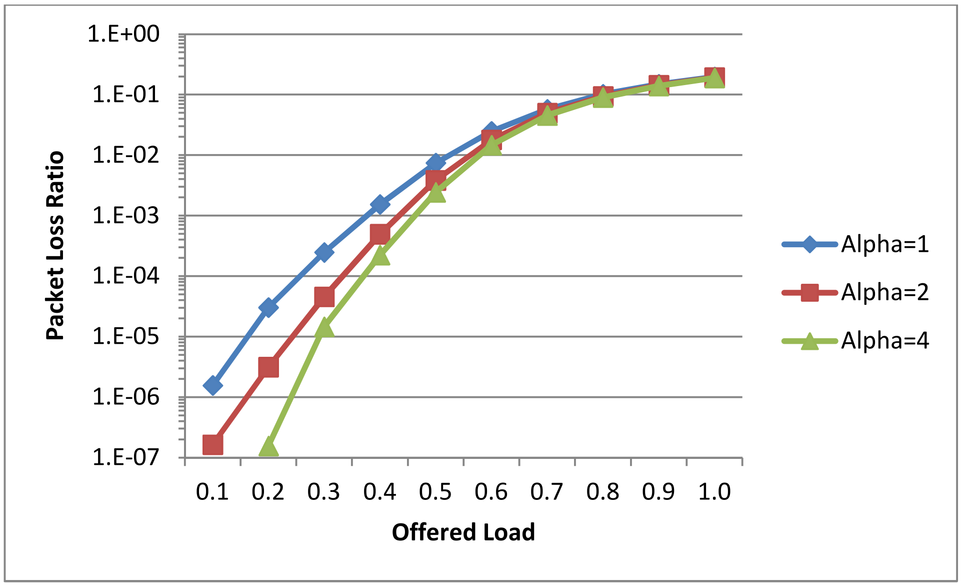


Figure 4.19-DT; Ports=16; Buffer=4096; ABL=256

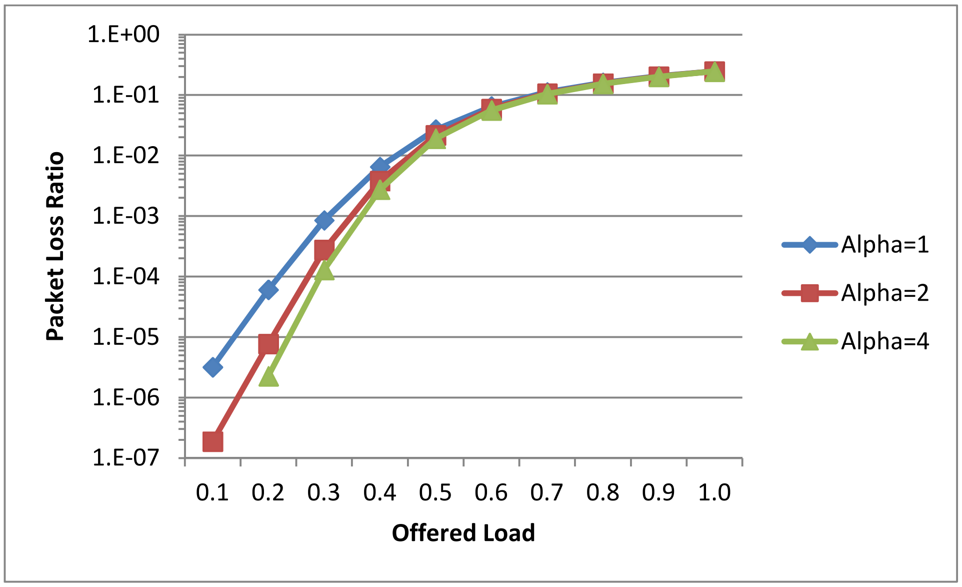


Figure 4.20-DT; Ports=32; Buffer=4096; ABL=256

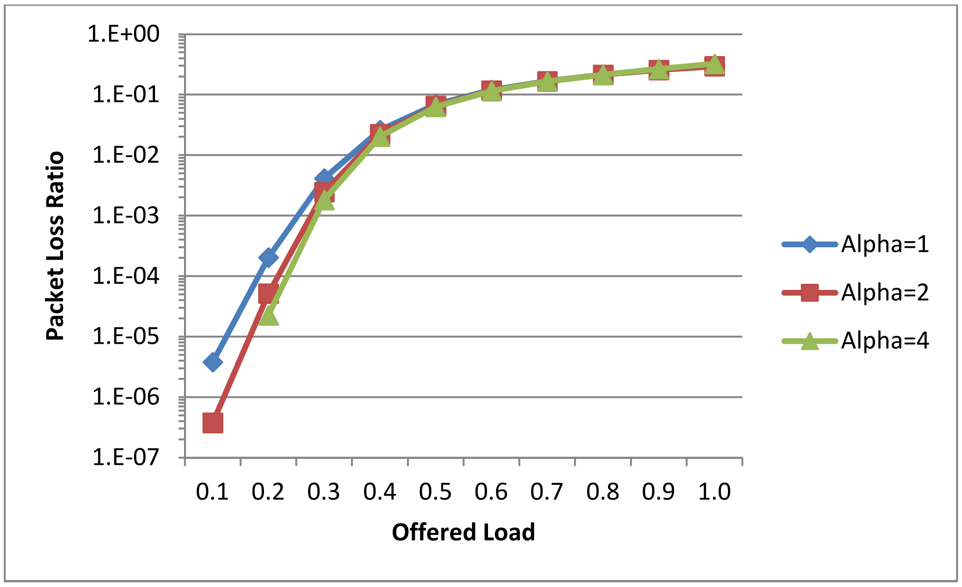


Figure 4.21-DT; Ports=64; Buffer=4096; ABL=256

4.2.3 Optimal Configuration for Different Buffer Size

4.2.3.1 SMA. Under all tested scenarios this scheme performed consistently, with bigger Minimum Allocation (MA) presenting less packet loss under high loads and smaller MA presenting less packet loss under low loads, as it may be observed in figure 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28. SMA with MA of 1/8 performs better than the other configurations under loads of up to 90%, presenting a packet loss ratio decrease of up to 2.34E-02, a 40.72% improvement; while SMA with MA of 1/2 presented the best performance on loads of 100%, with a packet loss ratio decrease of up to 6.82E-03, a 3.77% improvement.

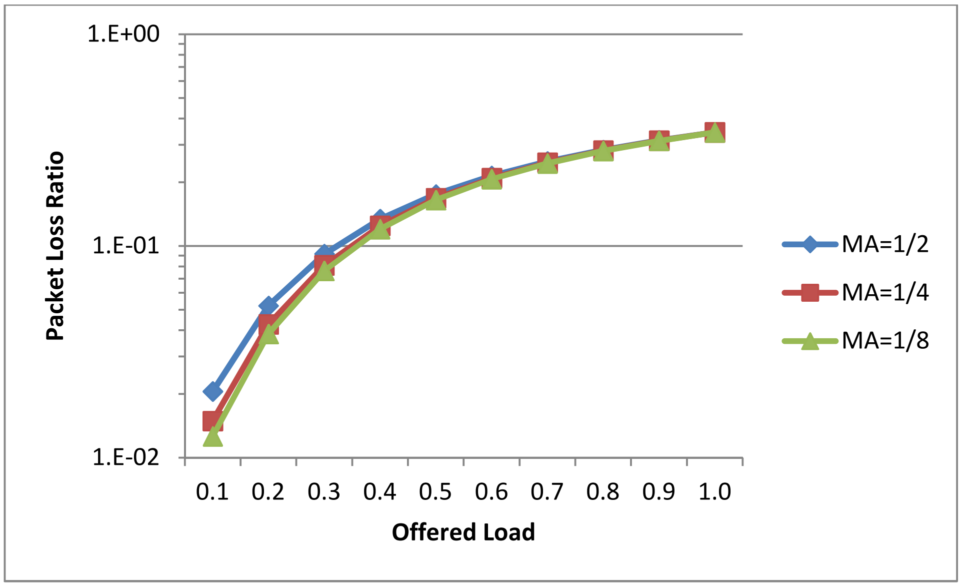


Figure 4.22-SMA; Ports=16; Buffer=256; ABL=256

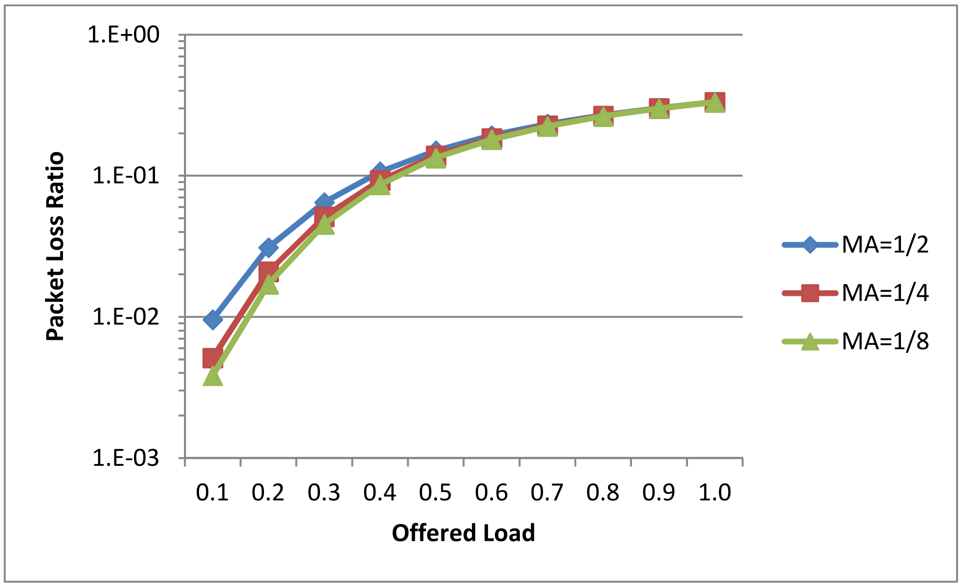


Figure 4.23-SMA; Ports=16; Buffer=512; ABL=256

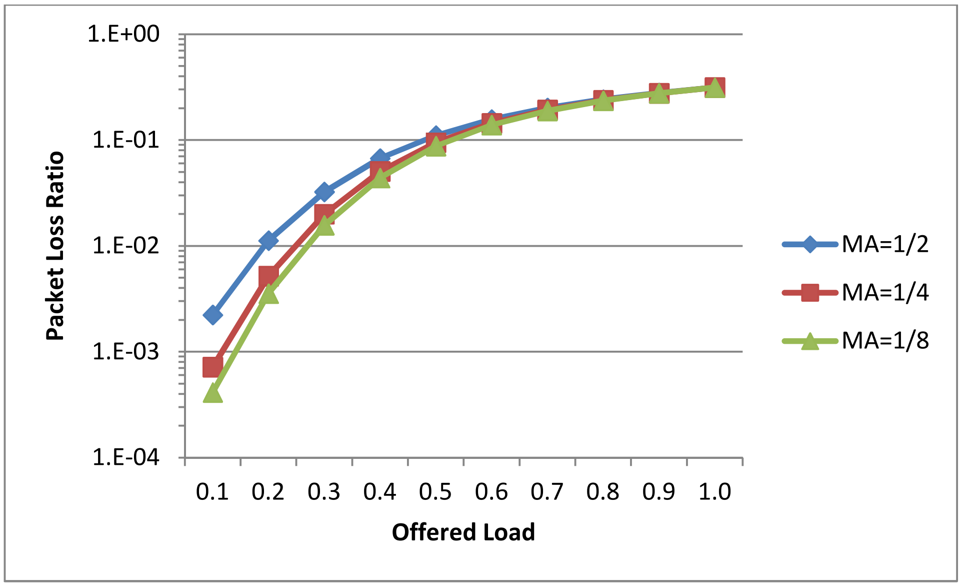


Figure 4.24-SMA; Ports=16; Buffer=1024; ABL=256

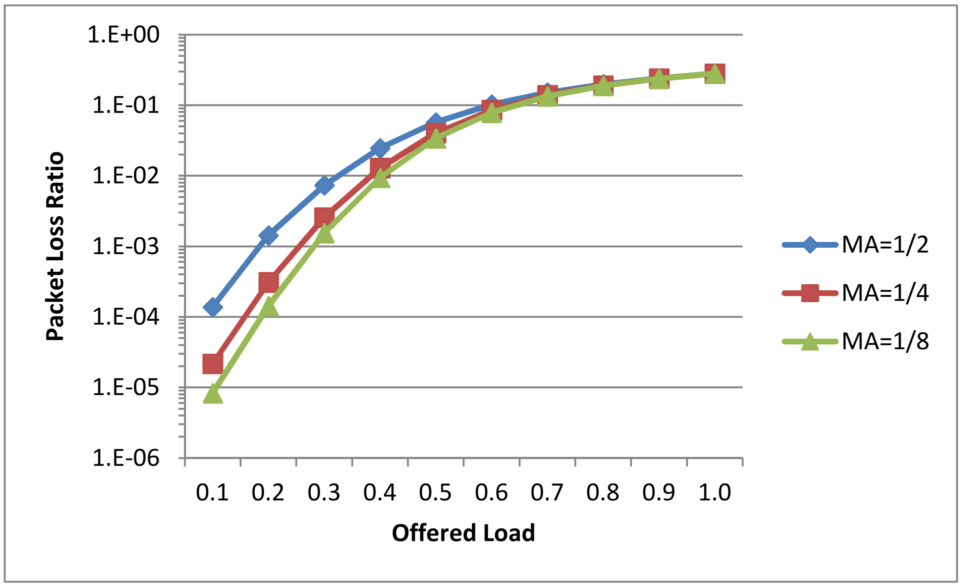


Figure 4.25-SMA; Ports=16; Buffer=2048; ABL=256

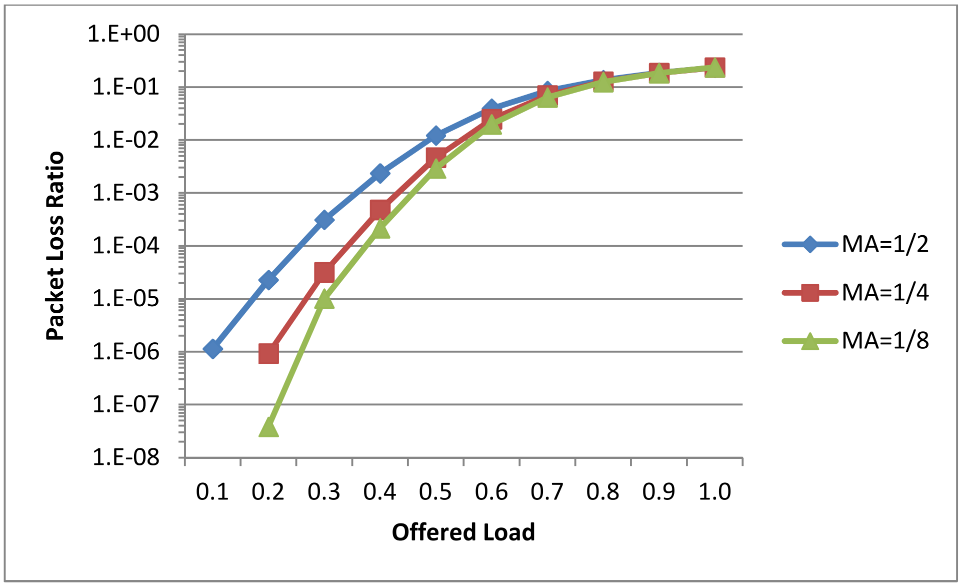


Figure 4.26-SMA; Ports=16; Buffer=4096; ABL=256

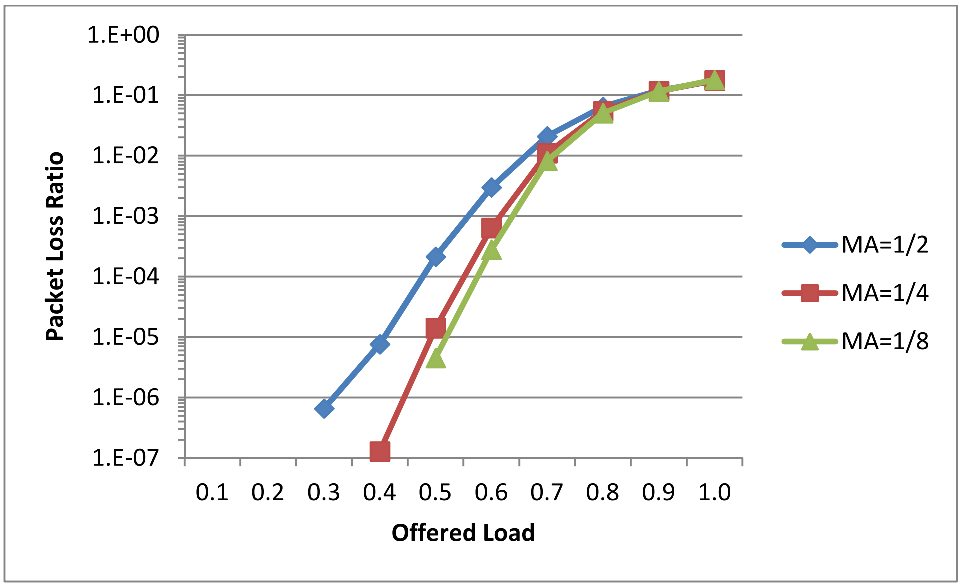


Figure 4.27-SMA; Ports=16; Buffer=8192; ABL=256

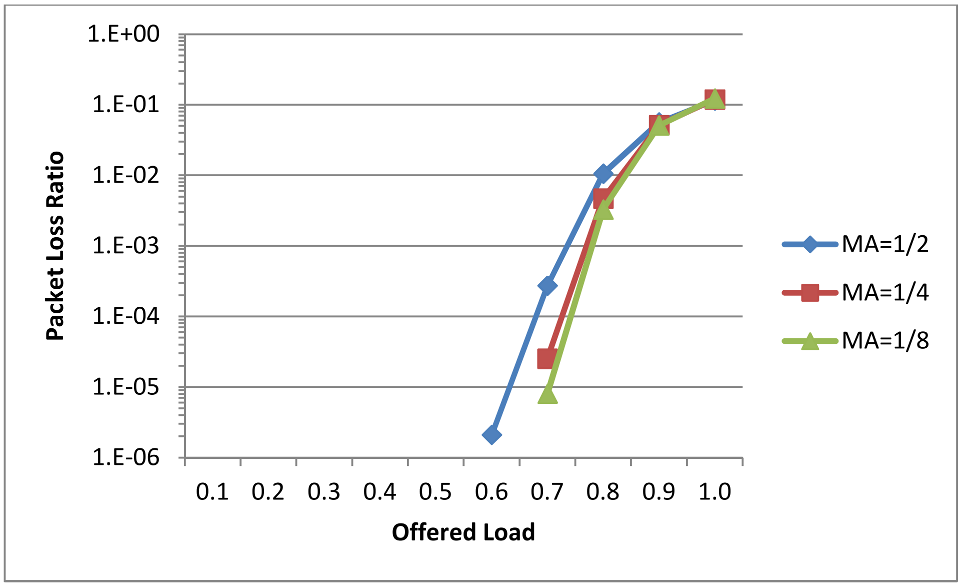


Figure 4.28-SMA; Ports=16; Buffer=16384; ABL=256

4.2.3.2 SMXQ. This sharing memory scheme presents once more a very changing optimal configuration depending on the size of the available buffer, as can be observed on figures 4.29, 4.30, 4.31, 4.32, 4.33, 4.34 and 4.35. It can be appreciated that with a buffer size on 256 SMXQ with a maximum queue length of 1/2 is only the best performing configuration for loads of 10% and 20%, however this range sees an increment of 10% as the available buffer doubles, making the configuration with an MXQ of 1/2 the best one for ranges of up to 30% when the buffer available is 512, up to 40% when 1024, 50% when 2048, 60% when 4096, 70% when 8182, and 80% when 16384. SMXQ with an MXQ of 1/4 performs the best only in the two or one 10% steps following the optimal range for an MXQ of 1/2, while SMXQ with an MXQ of 1/8 performs the best under the remaining offered loads.

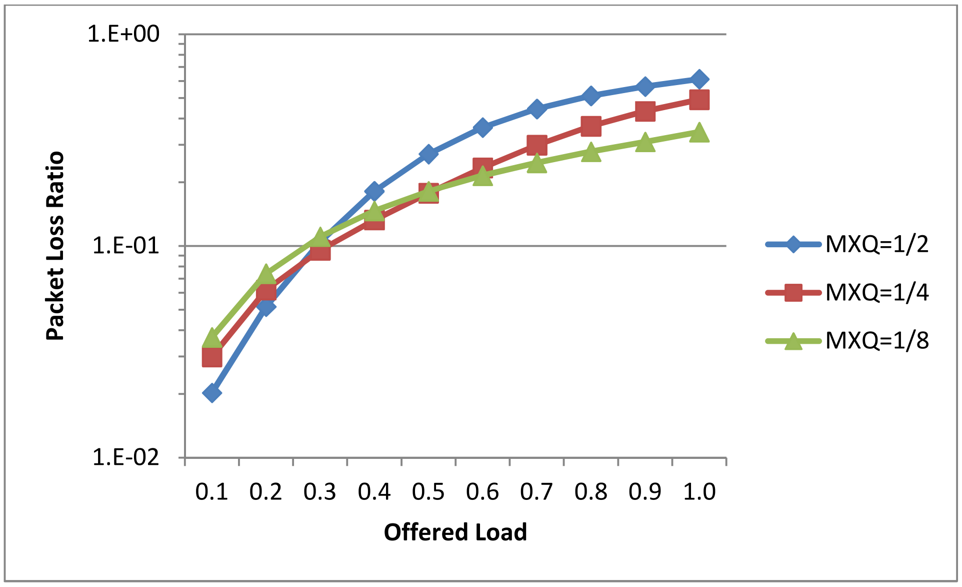


Figure 4.29-SMXQ; Ports=16; Buffer=256; ABL=256

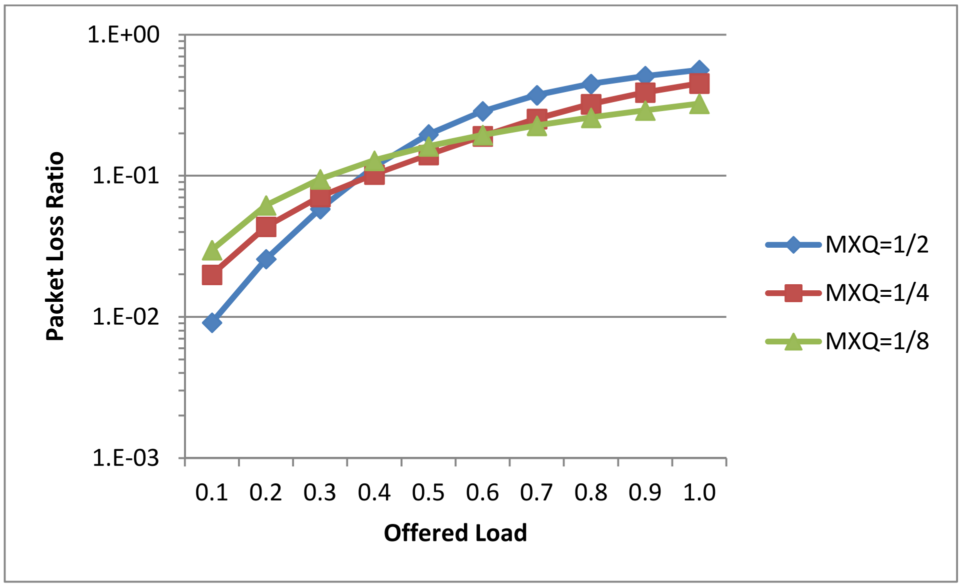


Figure 4.30-SMXQ; Ports=16; Buffer=512; ABL=256

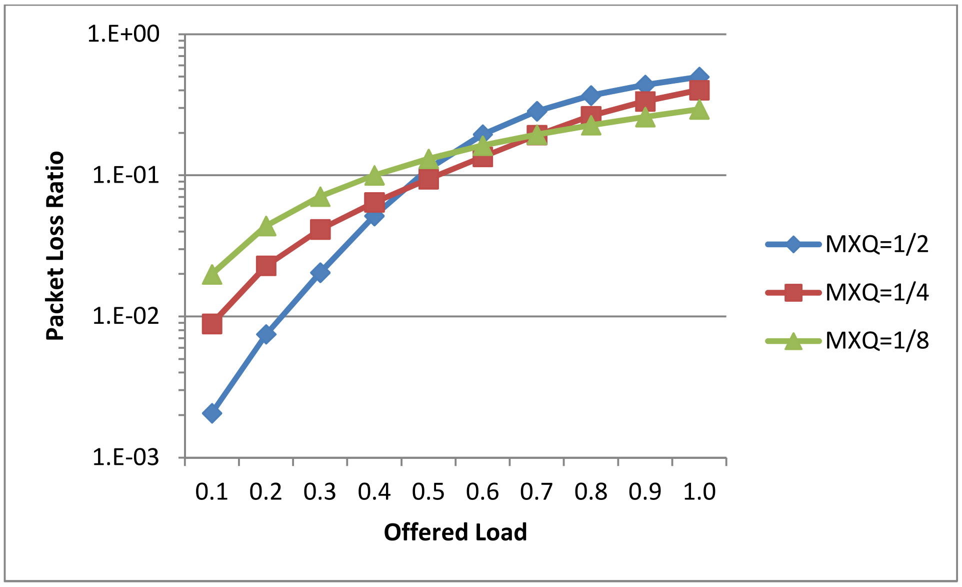


Figure 4.31-SMXQ; Ports=16; Buffer=1024; ABL=256

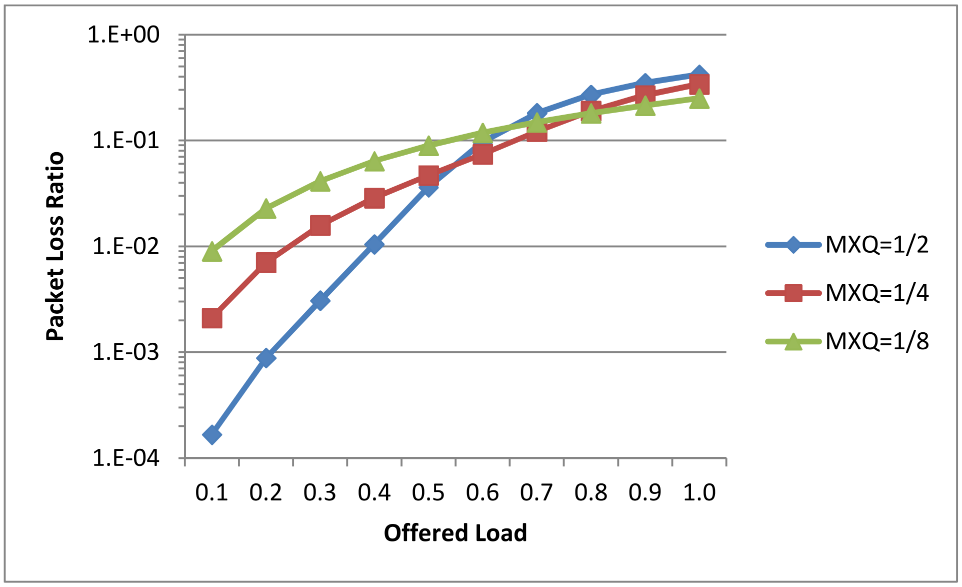


Figure 4.32-SMXQ; Ports=16; Buffer=2048; ABL=256

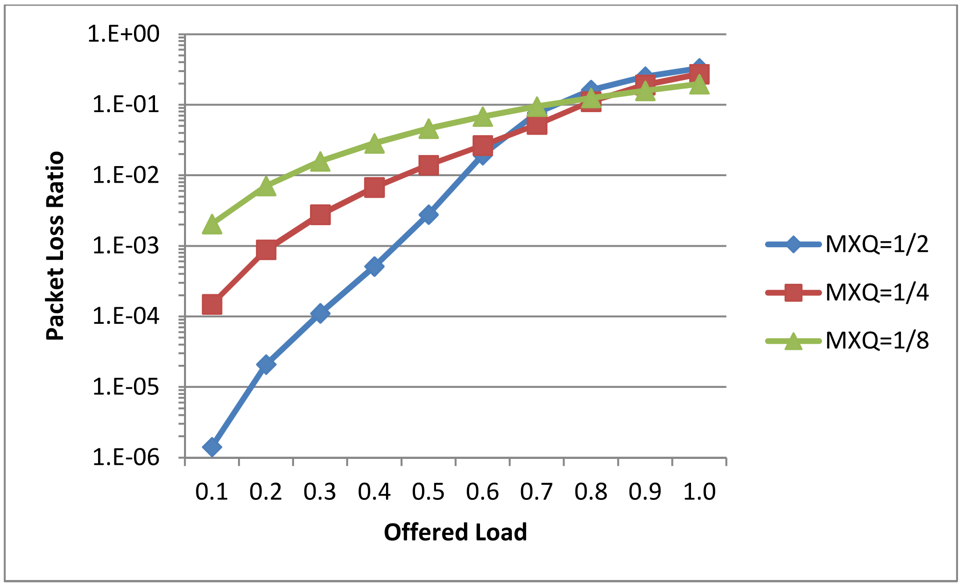


Figure 4.33-SMXQ; Ports=16; Buffer=4096; ABL=256

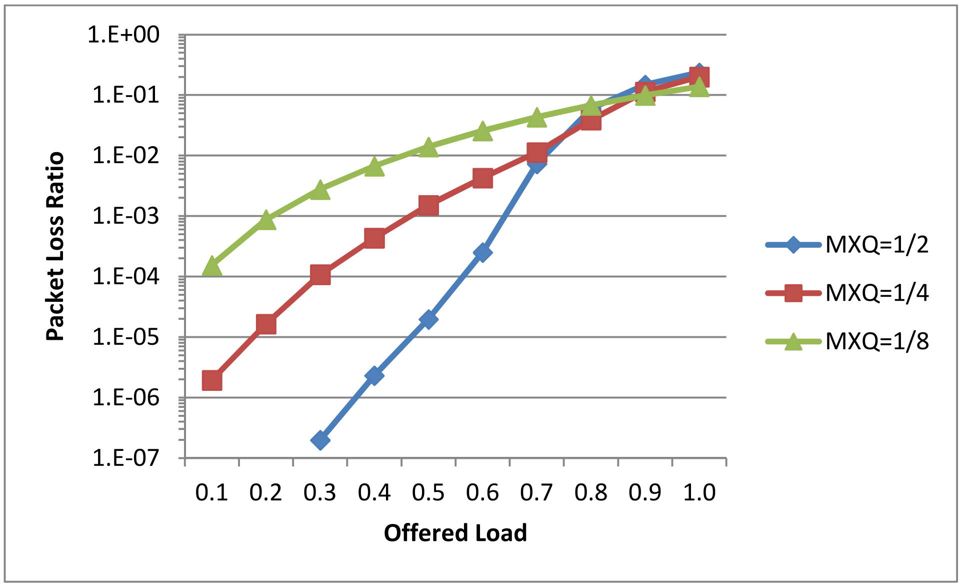


Figure 4.34-SMXQ; Ports=16; Buffer=8192; ABL=256

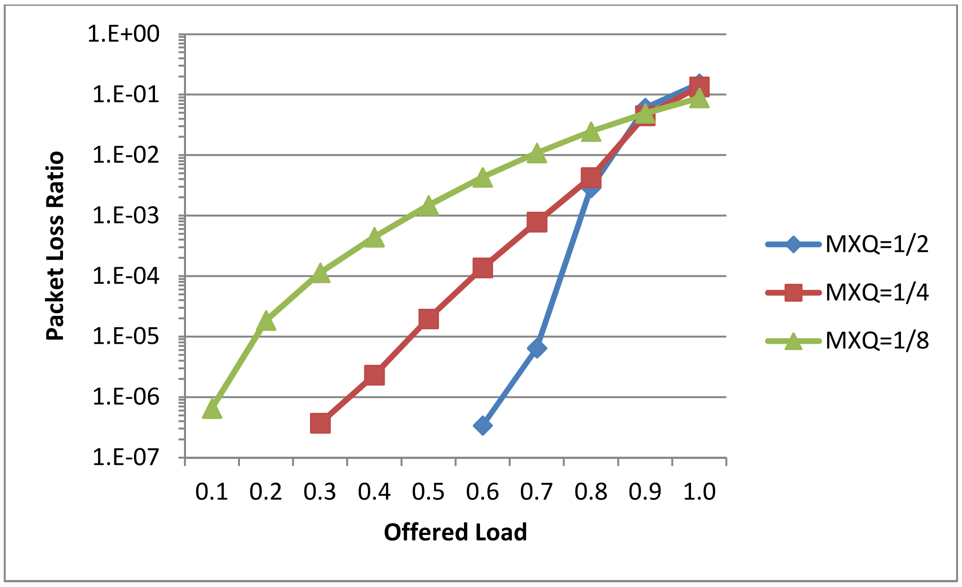


Figure 4.35-SMXQ; Ports=16; Buffer=16384; ABL=256

4.2.3.3 DT. Under all scenarios tested this scheme performed consistently. As it may be observed in figures 4.38, 4.39, 4.40, 4.41 and 4.42, for buffer sizes 1024, 2048, 4096, 8192 and 16384, DT with Alpha equal to 4 performs better than the other configurations, presenting a packet loss ratio decrease of up to 1.54E-02, a 16.95% improvement. For a buffer size of 256, DT with an Alpha of 4 performs the best under loads of up to 60%, presenting a packet loss ratio decrease of up to 1.16E-02, a 13.44% improvement; while DT with an Alpha of 2 performs the best for loads 70% to 100%, presenting a packet loss ratio decrease of up to 3.13E-02, an 8.46% improvement, as it may be observed in figure 4.36. For a buffer size of 512, DT with an Alpha of 4 performed the best under loads of up to 90%, presenting a packet loss ratio decrease of up to 1.44E-02, a 24.72% improvement; while DT with Alpha equal to 2 performs the best for a load of 100%, presenting a packet loss ratio decrease of up to 9.02E-03, a 15.53% improvement, as it is shown in figure 4.37.

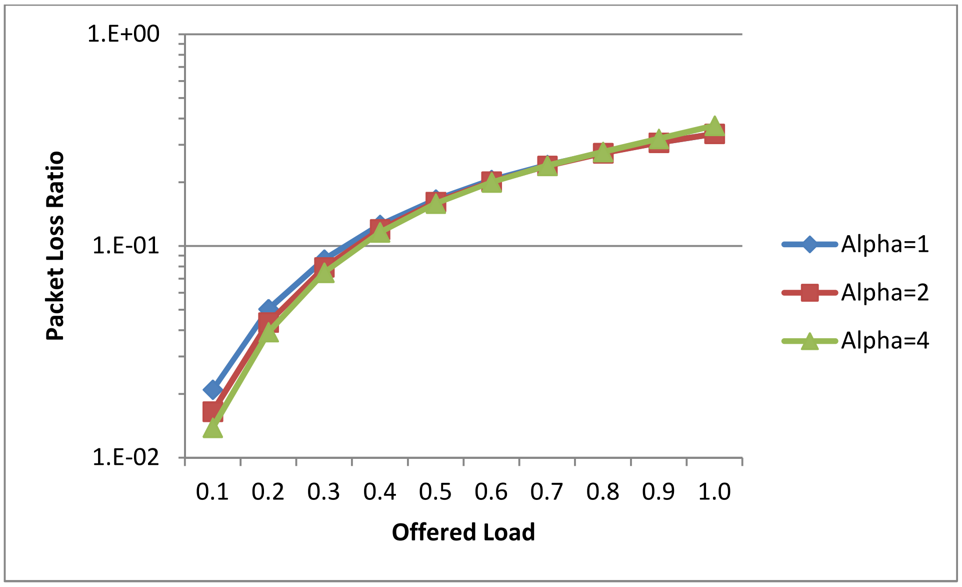


Figure 4.36-DT; Ports=16; Buffer=256; ABL=256

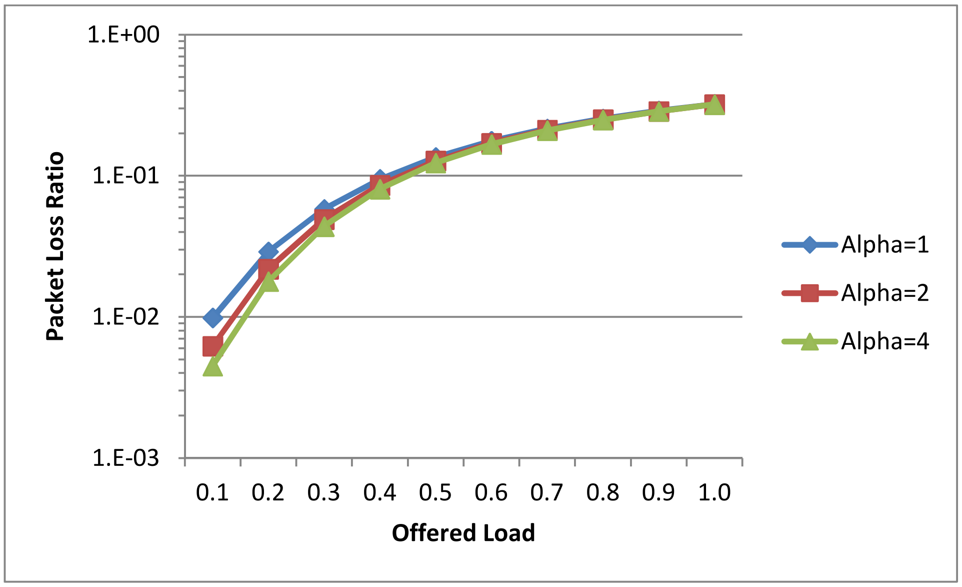


Figure 4.37-DT; Ports=16; Buffer=512; ABL=256

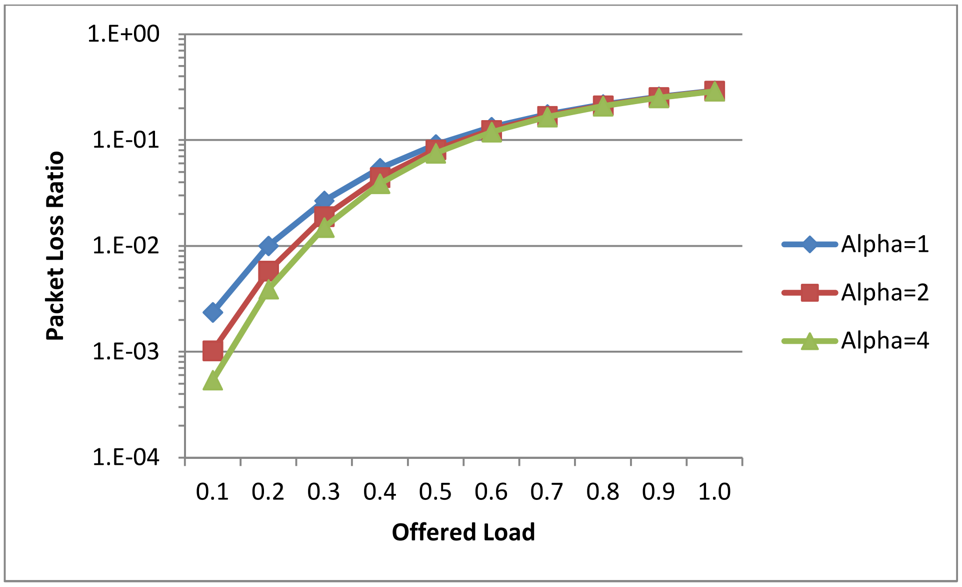


Figure 4.38-DT; Ports=16; Buffer=1024; ABL=256

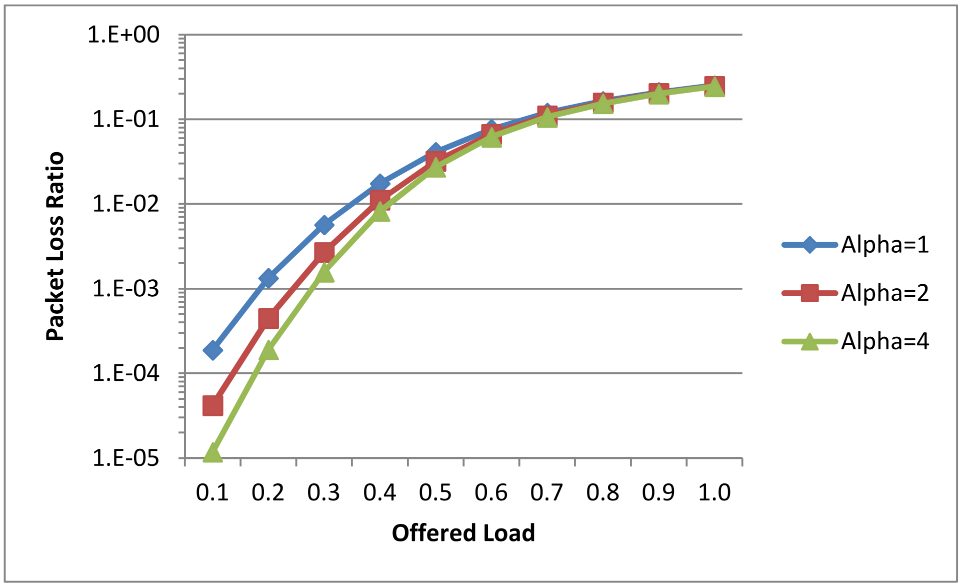


Figure 4.39-DT; Ports=16; Buffer=2048; ABL=256

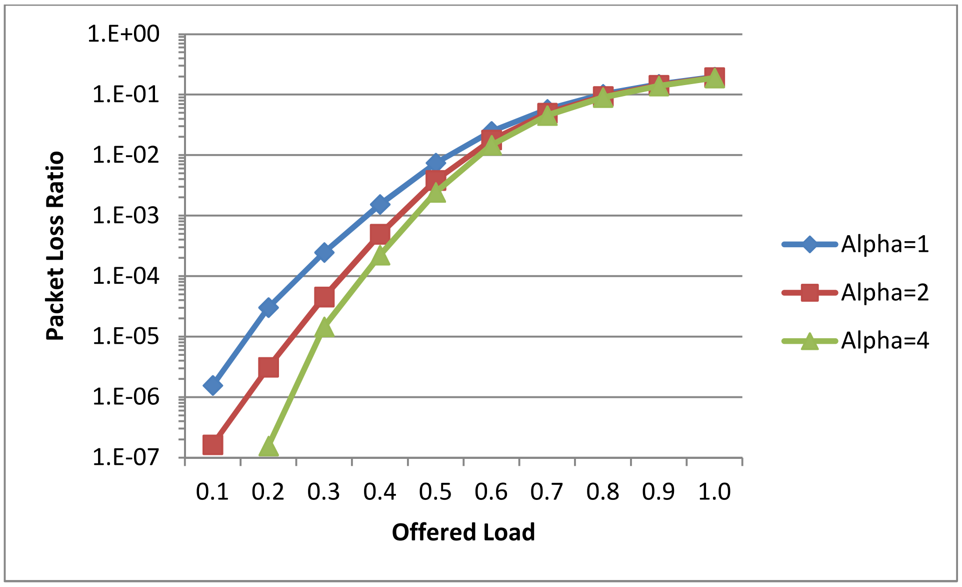


Figure 4.40-DT; Ports=16; Buffer=4096; ABL=256

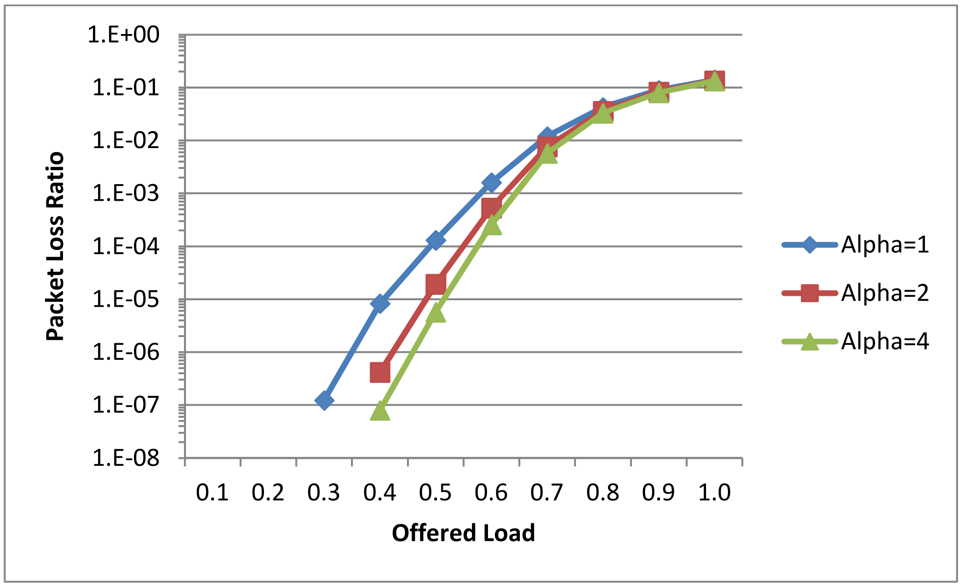


Figure 4.41-DT; Ports=16; Buffer=8192; ABL=256

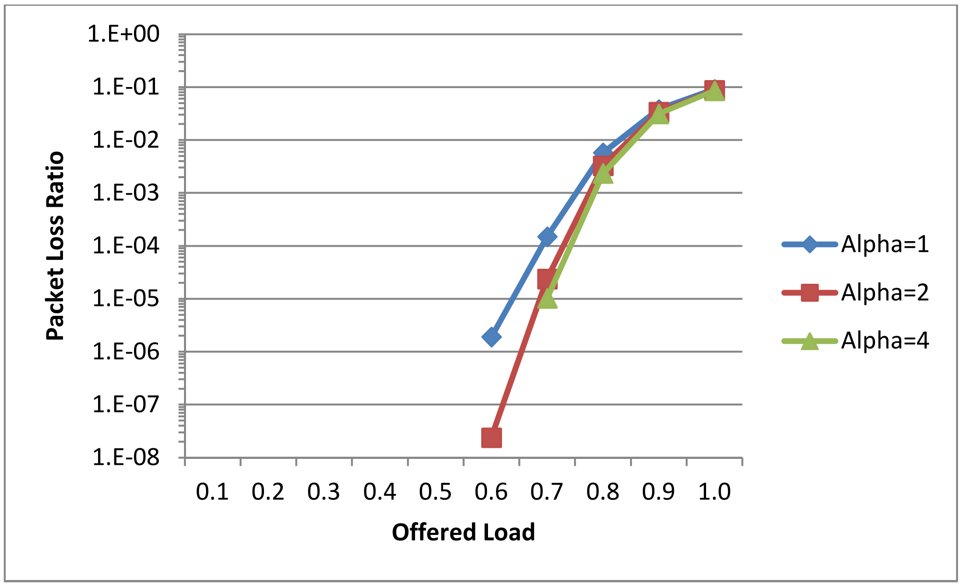


Figure 4.42-DT; Ports=16; Buffer=16384; ABL=256

4.3 Summary of Results

Tables 4.1, 4.2 and 4.3 provide a condensed view of all optimal configurations under the different simulated scenarios from which the following observations are derived.

SMA is the easiest shared memory scheme to configure, because regardless of the ABL, number of ports, or size of the buffer, the way it performs is always consistent. For loads ranging from 10% to 90% the best configuration in terms of packet loss is when SMA has a minimum allocation of 1/8 of the total available memory. Only when SMA is subjected to loads of 100% the minimum allocation required for an optimal performance changes to 1/2.

Out of the simulated shared memory schemes, SMXQ is the most difficult to configure. It is observed that as the number of ports increases, so does the range of loads under which a set maximum queue length of 1/8 presents the lowest possible packet loss ratio. The same can be said for ABL, where a higher ABL also expands the range of high loads in which a maximum queue length of 1/8 exhibits the best performance. Again, the same applies to decreasing available memory, which also widens the range of high loads under which a maximum queue length provides the best possible performance.

DT is also somewhat easy to configure optimally, since for the most part the best possible configuration is achieved utilizing an Alpha equal to four, except when the available memory is either for 256 or 512 cells, in which case an Alpha of two yields the best results when DT is subjected to loads ranging from 70% to 100% for the former case, and loads of 100% for the latter. The other exception is when the number of ports equals 64; under ALBs of 64 and 128 an Alpha equals to two delivers the best possible performance when the scheme is being subjected to loads in between 80% and 100%; when the ABL equals 256 and the offered load is either 90% or 100% then this is the only case where an Alpha of one is required in order to achieve minimum packet loss ratio.

ABL	Scheme	Variable	Load%
64	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-50
		1/8	60-100
	DT	4	10-70
		2	80-100
128	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-40
		1/8	50-100
	DT	4	10-70
		2	80-100
256	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-30
		1/8	40-100
	DT	4	10-80
		1	90-100

Table 4.1-Optimal Configurations for Different ABLs. NxN=64x64; Buffer=4096

Ports	Scheme	Variable	Load%
16	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-60
		1/4	70-80
		1/8	90-100
	DT	4	10-100
32	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-40
		1/4	50
		1/8	60-100
	DT	4	10-100
64	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-30
		1/8	40-100
	DT	4	10-80
		1	90-100

Table 4.2-Optimal Configurations for Different Number of Ports. ABL=256; Buffer=4096

Buffer	**Scheme**	**Variable**	**Load%**
256	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-20
		1/4	30-50
		1/8	60-100
	DT	4	10-60
		2	70-100
512	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-30
		1/4	40-60
		1/8	70-100
	DT	4	10-90
		2	100
1024	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-40
		1/4	50-70
		1/8	80-100
	DT	4	10-100
2048	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-50
		1/4	60-70
		1/8	80-100
	DT	4	10-100
4096	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-60
		1/4	70-80
		1/8	90-100
	DT	4	10-100
8192	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-70
		1/4	80
		1/8	90-100
	DT	4	10-100
16384	SMA	1/8	10-90
		1/2	100
	SMXQ	1/2	10-80
		1/4	90
		1/8	100
	DT	4	10-100

Table 4.3-Optimal Configurations for Different Buffer Sizes. ABL=256; NxN=16

CHAPTER V

CONVENTIONAL SHARING MEMORY SCHEMES COMPARED

Because comparative performance evaluations between Complete Sharing (CS), Sharing with a Minimum Allocation (SMA) and Sharing with Maximum Queue lengths (SMXQ) have already been carried out in the past [27] [28] [55] [56], in this chapter we will focus on determining whether or not the sharing memory scheme Dynamic Threshold (DT) has a better performance in terms of packet loss ratio than its static threshold counterpart SMXQ as well as the hybrid scheme SMA under each of the tested scenarios.

5.1 SMXQ vs. DT

5.1.1 Performance Evaluation under Different Average Burst Length

As it can be observed in figures 5.1 and 5.2, for an ABL equal to 64, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 1.58E-01, or a 45.24% improvement of DT with Alpha of 2 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to 1.37E-01, or a 46.53% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

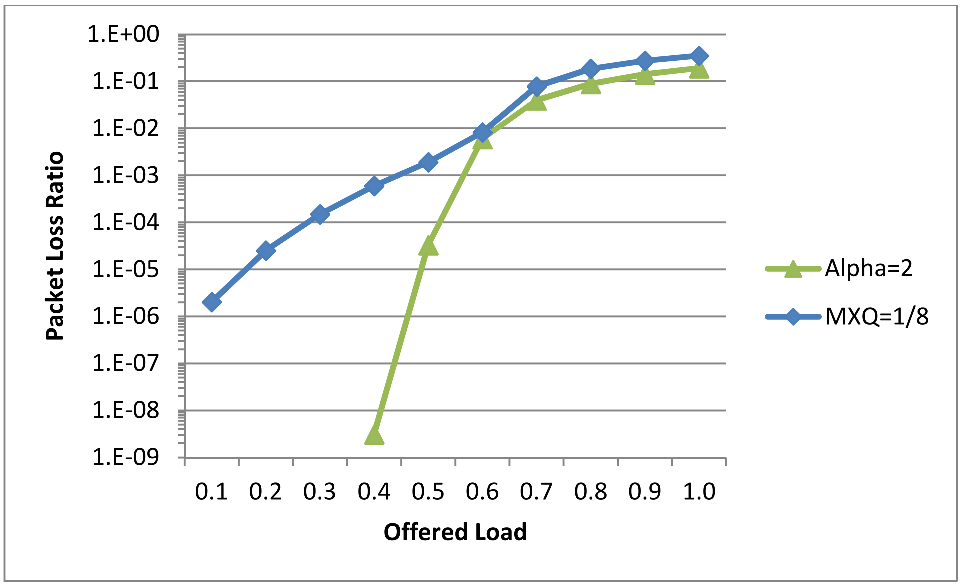


Figure 5.1-SMXQvsDT; Hi; Ports=64; Buffer=4096; ABL=64

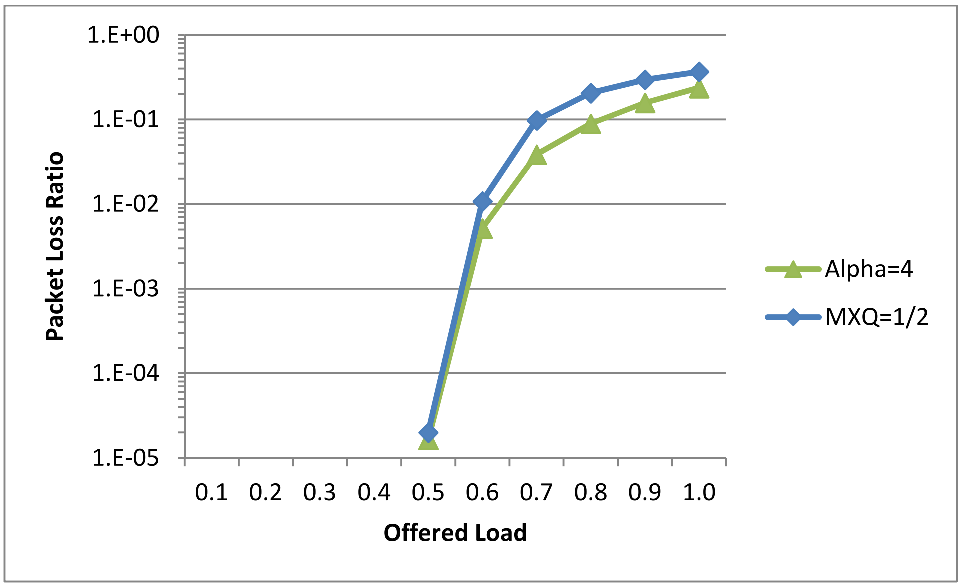


Figure 5.2-SMXQvsDT; Lo; Ports=64; Buffer=4096; ABL=64

As it can be observed in figures 5.3 and 5.4, for an ABL equal to 128, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 1.94E-01, or a 43.76% improvement of DT with Alpha of 2 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to 2.04E-01, or a 48.69% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

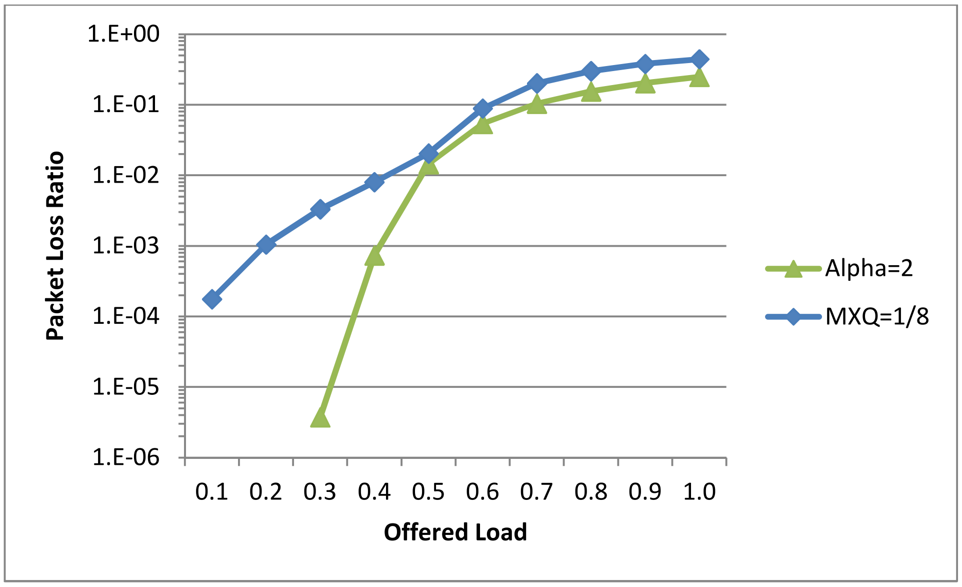


Figure 5.3-SMXQvsDT; Hi; Ports=64; Buffer=4096; ABL=128

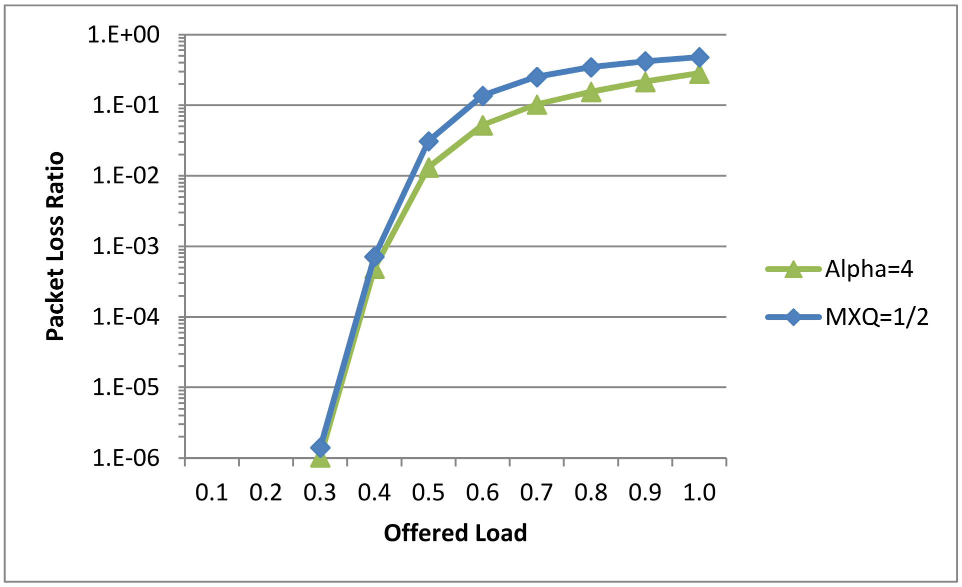


Figure 5.4-SMXQvsDT; Lo; Ports=64; Buffer=4096; ABL=128

From figures 5.5 and 5.6 we can see that for an ABL equal to 256, DT performs better under loads covering the complete range from 10% to 100%, presenting a decrease in packet loss ratio of up to 2.24E-01, or a 43.23% improvement of DT with Alpha of 1 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to 2.69E-01, or a 50.36% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

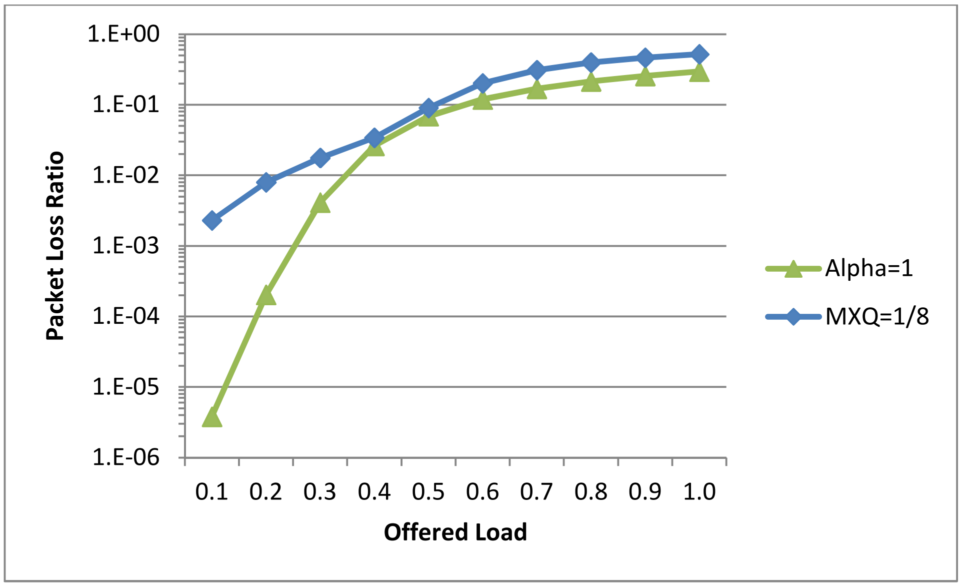


Figure 5.5-SMXQvsDT; Hi; Ports=64; Buffer=4096; ABL=256

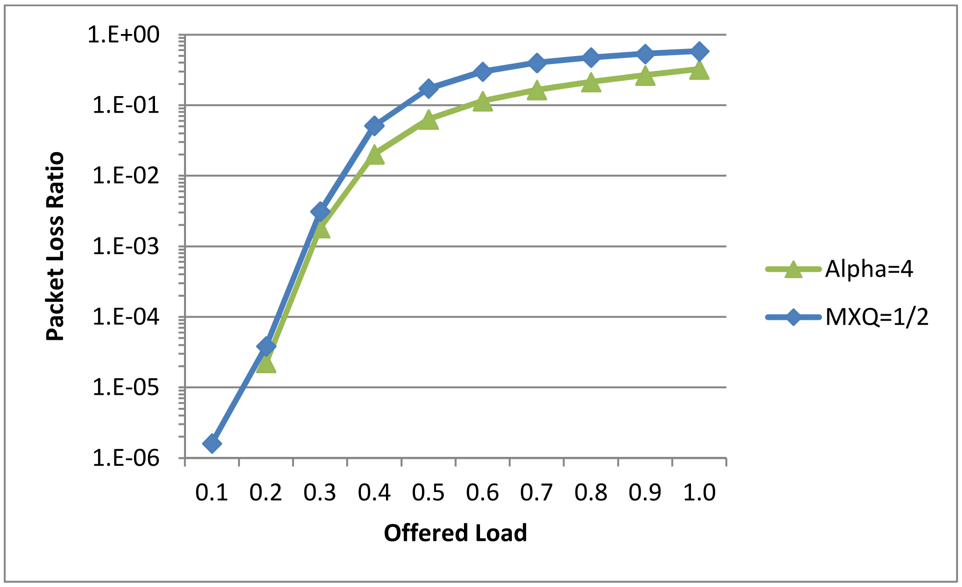


Figure 5.6-SMXQvsDT; Lo; Ports=64; Buffer=4096; ABL=256

5.1.2 Performance Evaluation under Different Number of Ports

In figures 5.7 and 5.8 it can be observed it can be observed that for a number of ports equal to 16, DT performs better for loads ranging from 10% to 100%, presenting a decrease in packet loss ratio of up to 5.35E-02, or a 78.52% improvement of DT with Alpha of 4 over

SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to 1.40E-01, or a 42.50% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

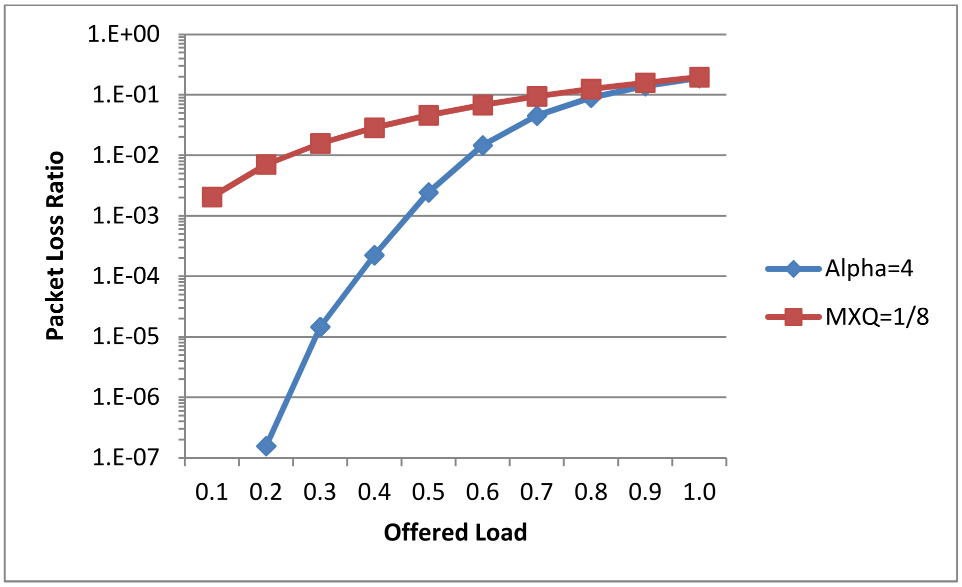


Figure 5.7-SMXQvsDT; Hi; Ports=16; Buffer=4096; ABL=256

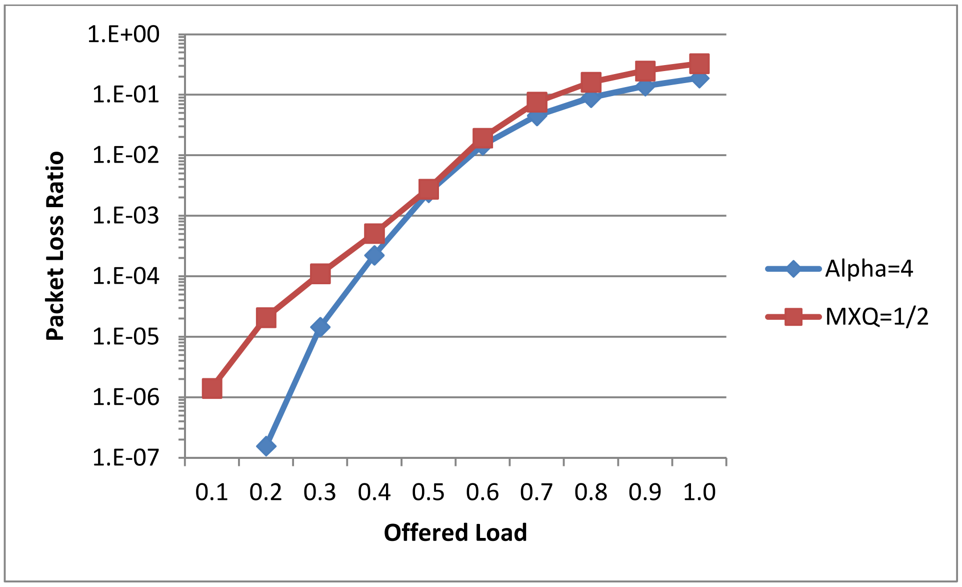


Figure 5.8-SMXQvsDT; Lo; Ports=16; Buffer=4096; ABL=256

As it can be observed in figures 5.9 and 5.10, for a number of ports equal to 32, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 1.02E-01, or a 29.32% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/8, and a

decrease in packet loss ratio of up to 2.17E-01, or a 46.78% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

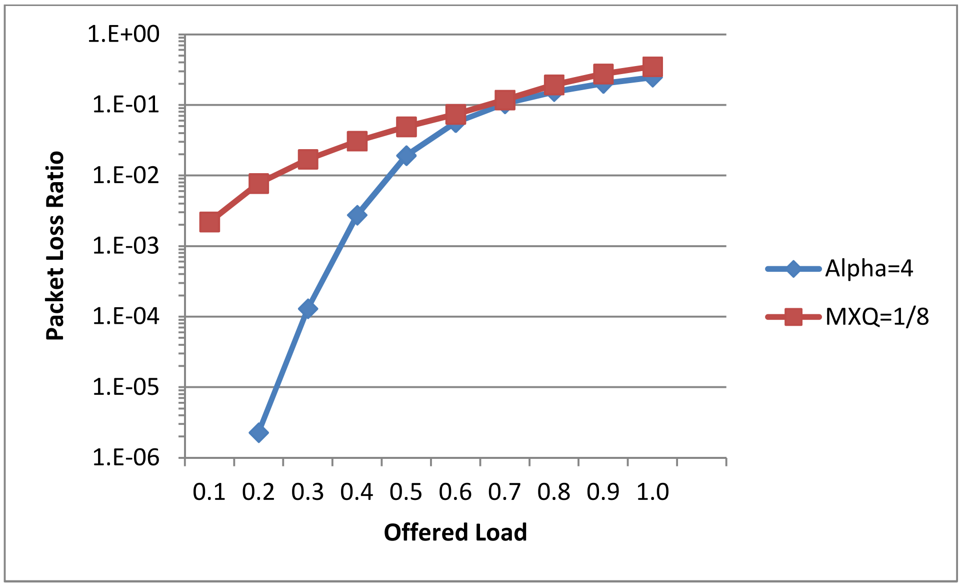


Figure 5.9-SMXQvsDT; Hi; Ports=32; Buffer=4096; ABL=256

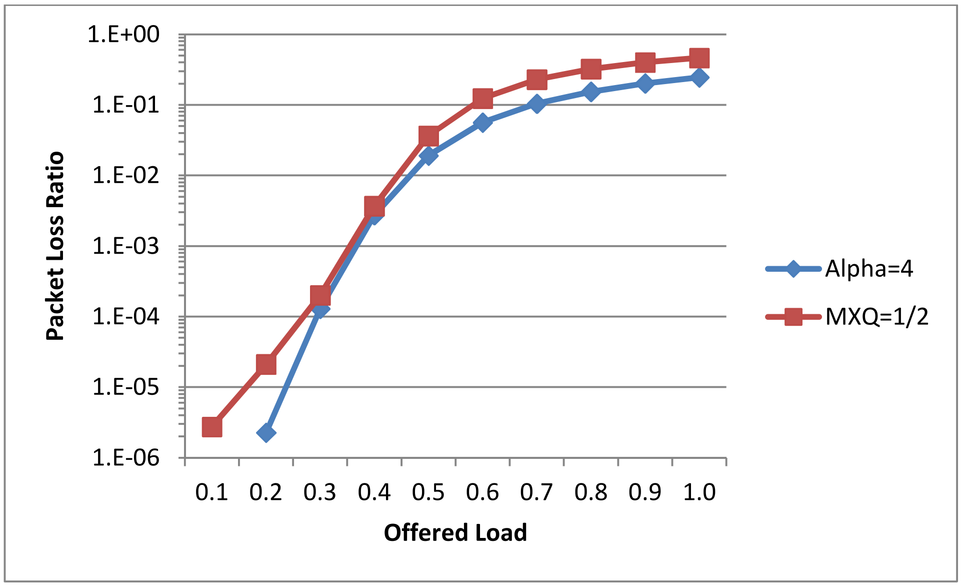


Figure 5.10-SMXQvsDT; Lo; Ports=32; Buffer=4096; ABL=256

From figures 5.11 and 5.12 it can be observed that for a number of ports equal to 64, DT performs better under the complete range of offered loads, presenting a decrease in packet loss ratio of up to 2.24E-01, or a 43.23% improvement of DT with Alpha of 1 over SMXQ with

MXQ of 1/8, and a decrease in packet loss ratio of up to 2.24E-01, or a 50.35% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

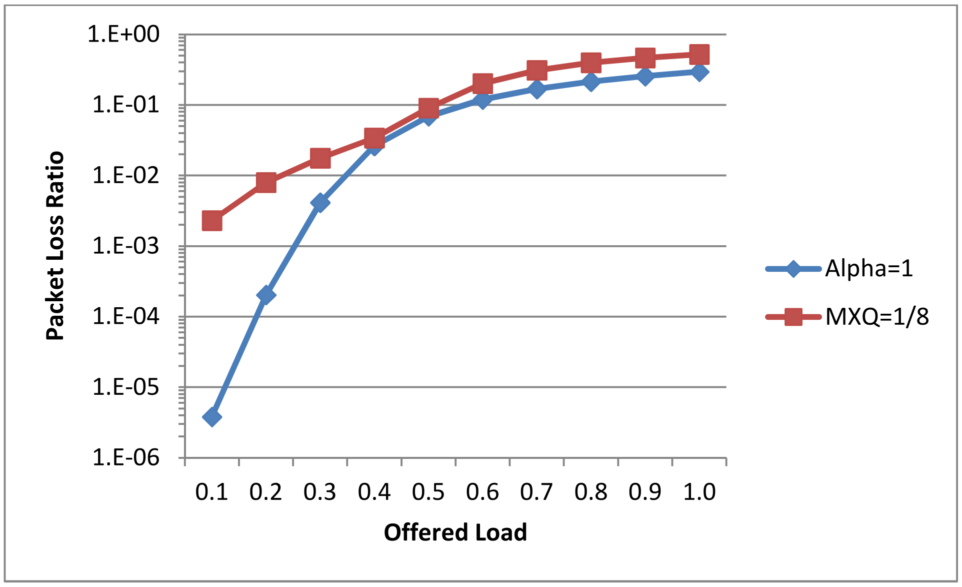


Figure 5.11-SMXQvsDT; Hi; Ports=64; Buffer=4096; ABL=256

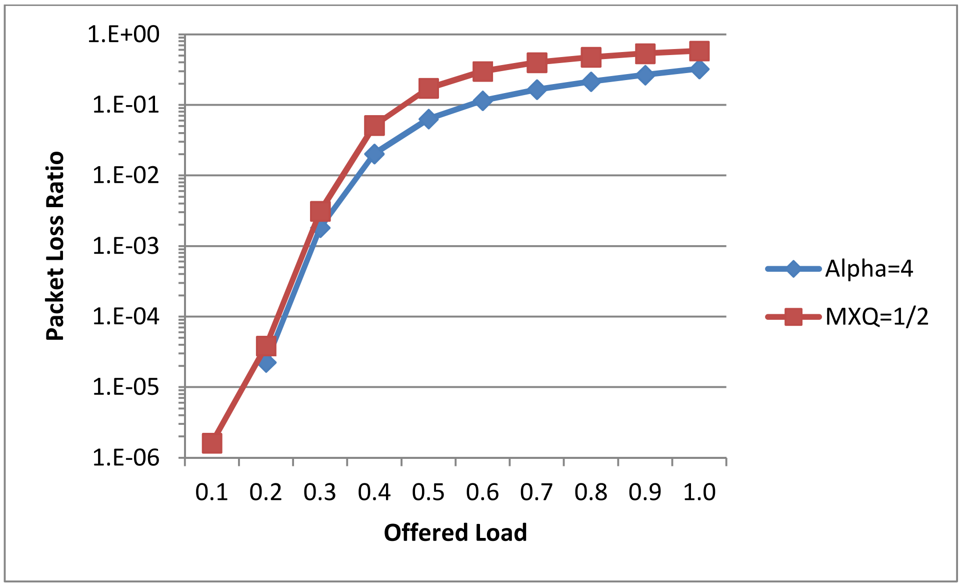


Figure 5.12-SMXQvsDT; Lo; Ports=64; Buffer=4096; ABL=256

5.1.3 Performance Evaluation under Different Buffer Size

Figures 5.13 and 5.14 show that for a buffer size equal to 256, DT performs better under all loads tested, presenting a decrease in packet loss ratio of up to 3.14E-02, or a 28.42%

improvement of DT with Alpha of 2 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to 2.46E-01, or a 43.36% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

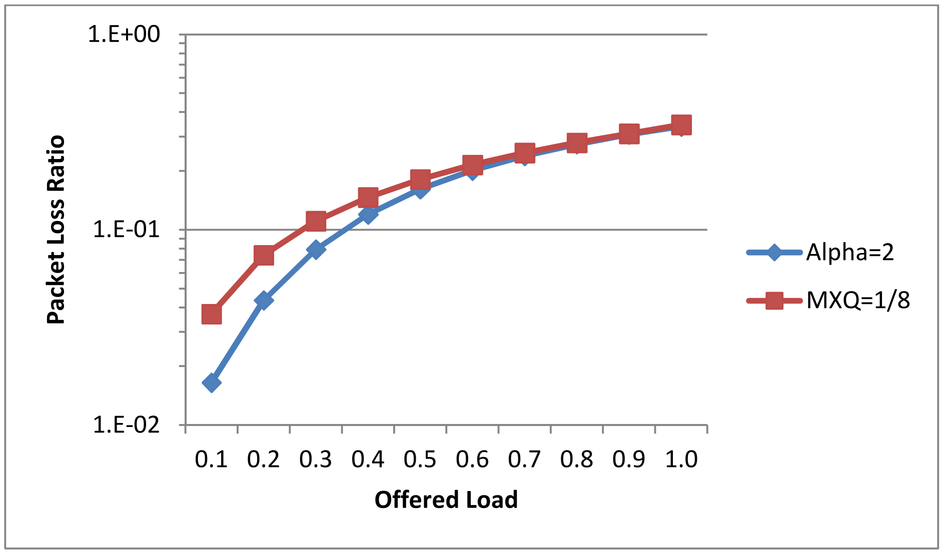


Figure 5.13-SMXQvsDT; Hi; Ports=16; Buffer=256; ABL=256

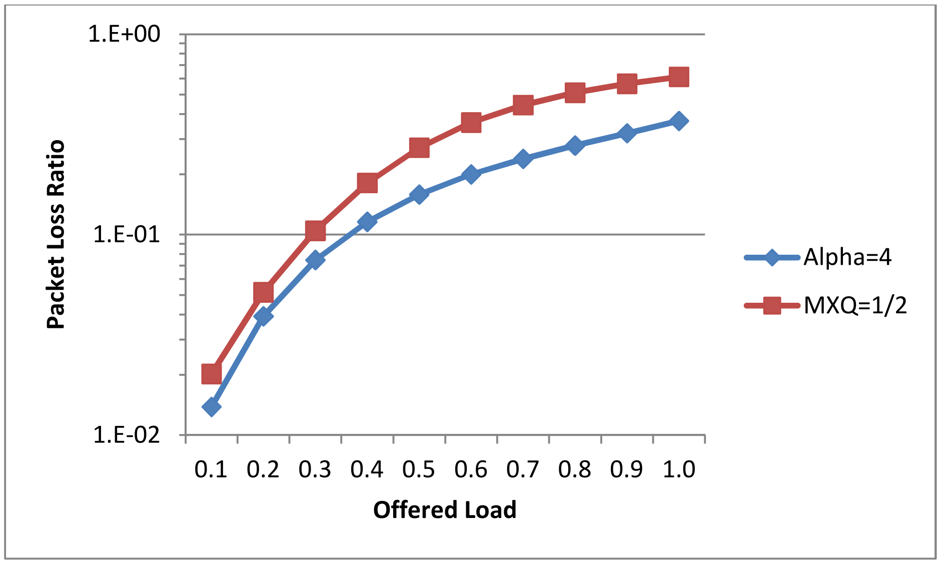


Figure 5.14-SMXQvsDT; Lo; Ports=16; Buffer=256; ABL=256

As it can be observed in figures 5.15 and 5.16, for a buffer size equal to 512, DT performs better all offered loads, presenting a decrease in packet loss ratio of up to 4.57E-02, or a 48.20% improvement of DT with Alpha of 2 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to 2.43E-01, or a 43.09% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

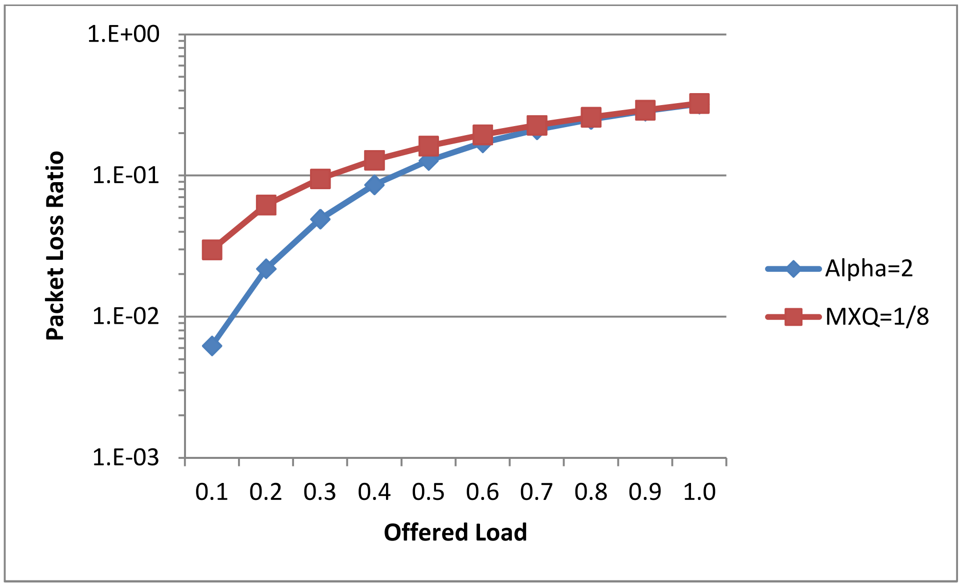


Figure 5.15-SMXQvsDT; Hi; Ports=16; Buffer=512; ABL=256

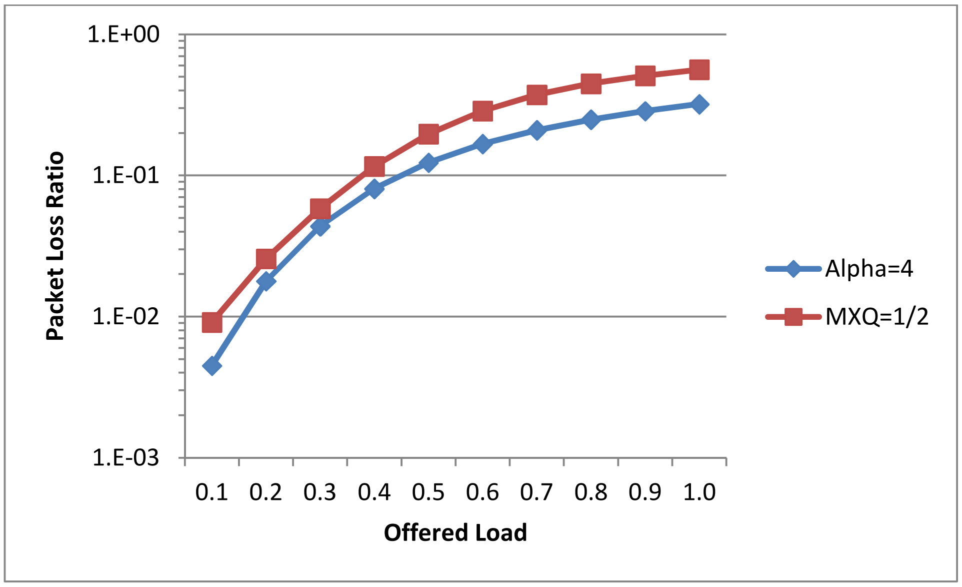


Figure 5.16-SMXQvsDT; Lo; Ports=16; Buffer=512; ABL=256

Figures 5.17 and 5.18 show that for a buffer size equal to 1024, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 6.13E-02, or a 61.23% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to t2.09E-01, or a 41.97% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

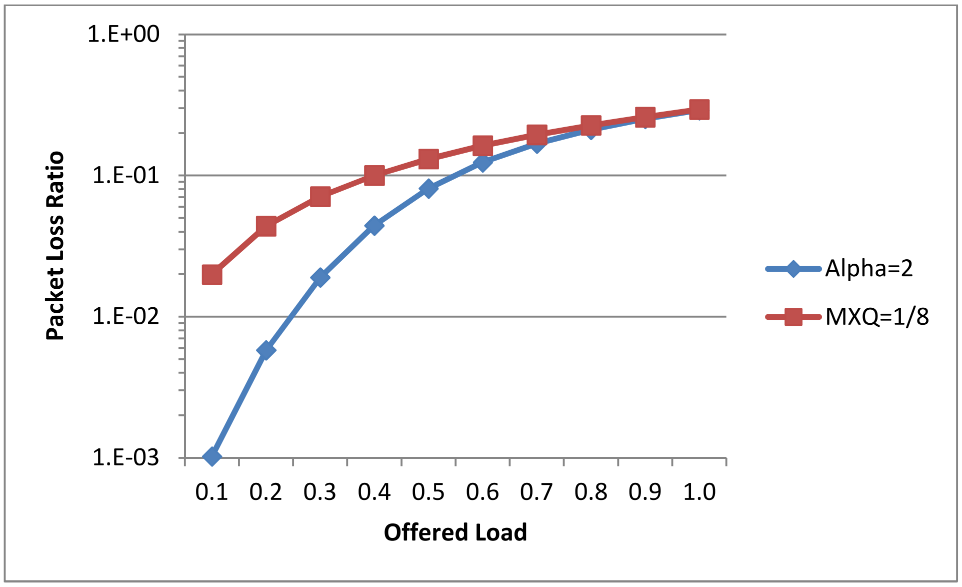


Figure 5.17-SMXQvsDT; Hi; Ports=16; Buffer=1024; ABL=256

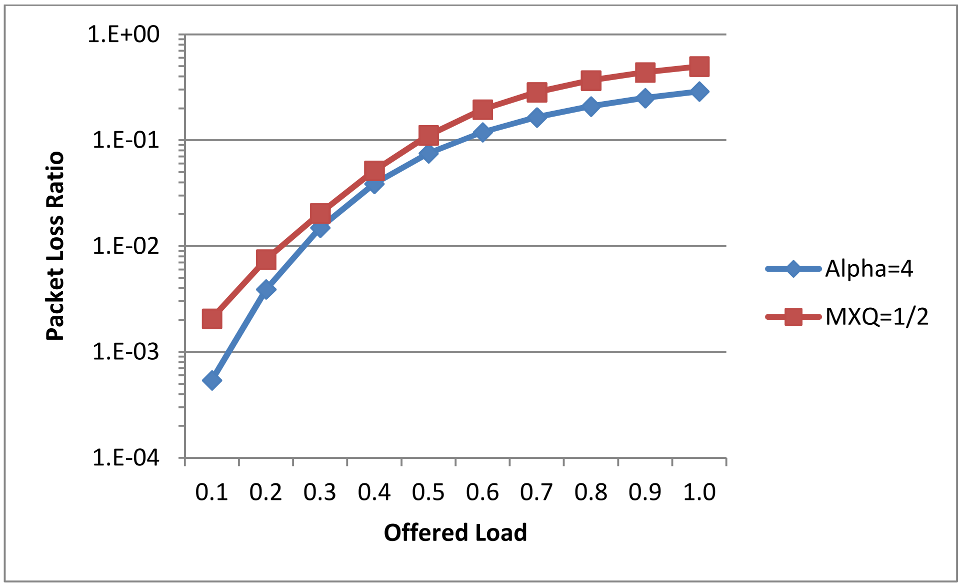


Figure 5.18-SMXQvsDT; Lo; Ports=16; Buffer=1024; ABL=256

From figures 5.19 and 5.20 it can be observed that for a buffer size equal to 2048, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 6.24E-02, or a 69.59% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/8, and a

decrease in packet loss ratio of up to 1.76E-01, or a 41.83% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

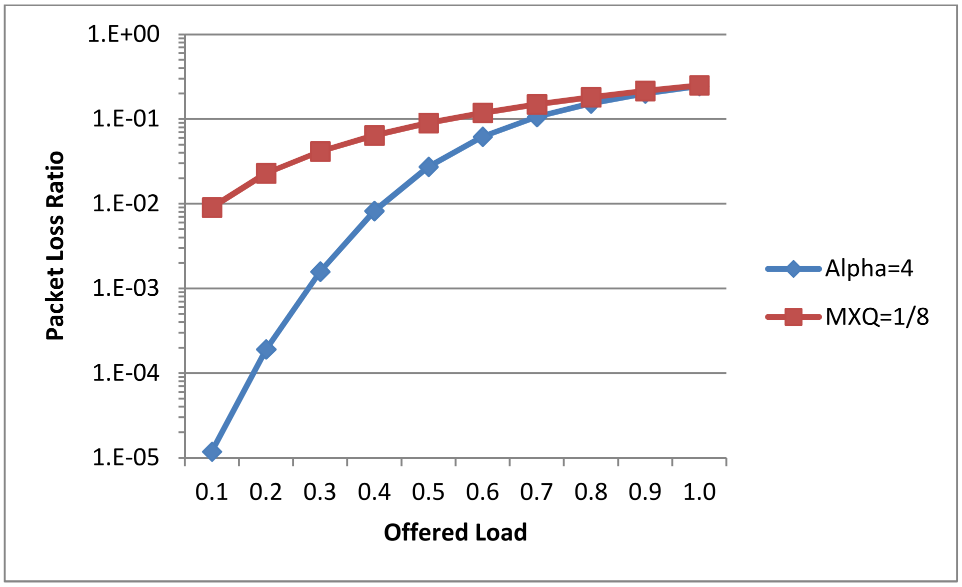


Figure 5.19-SMXQvsDT; Hi; Ports=16; Buffer=2048; ABL=256

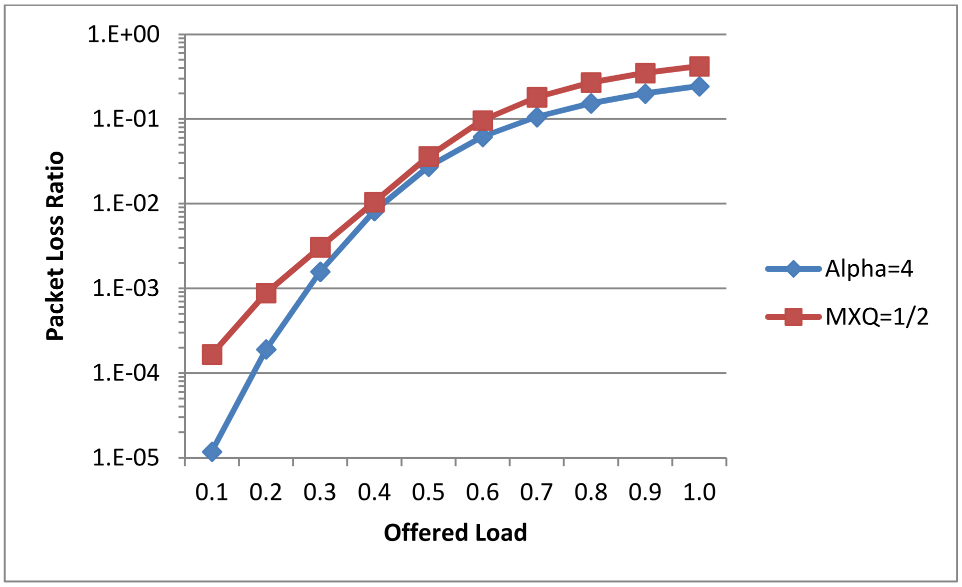


Figure 5.20-SMXQvsDT; Lo; Ports=16; Buffer=2048; ABL=256

As it can be observed in figures 5.21 and 5.22, for a buffer size equal to 4096, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to5.35E-02, or a 78.53% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/8, and a

decrease in packet loss ratio of up to 1.40E-01, or a 42.50% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

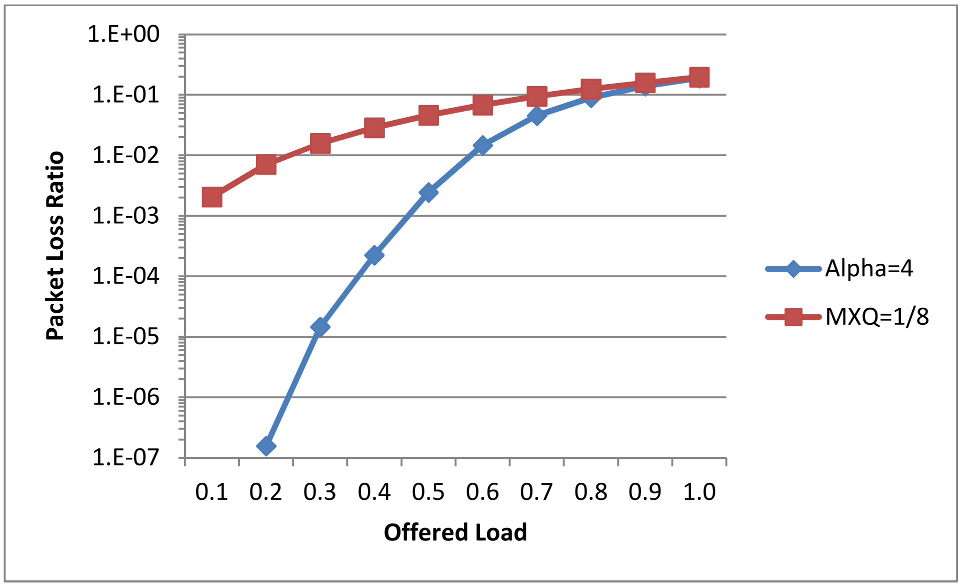


Figure 5.21-SMXQvsDT; Hi; Ports=16; Buffer=4096; ABL=256

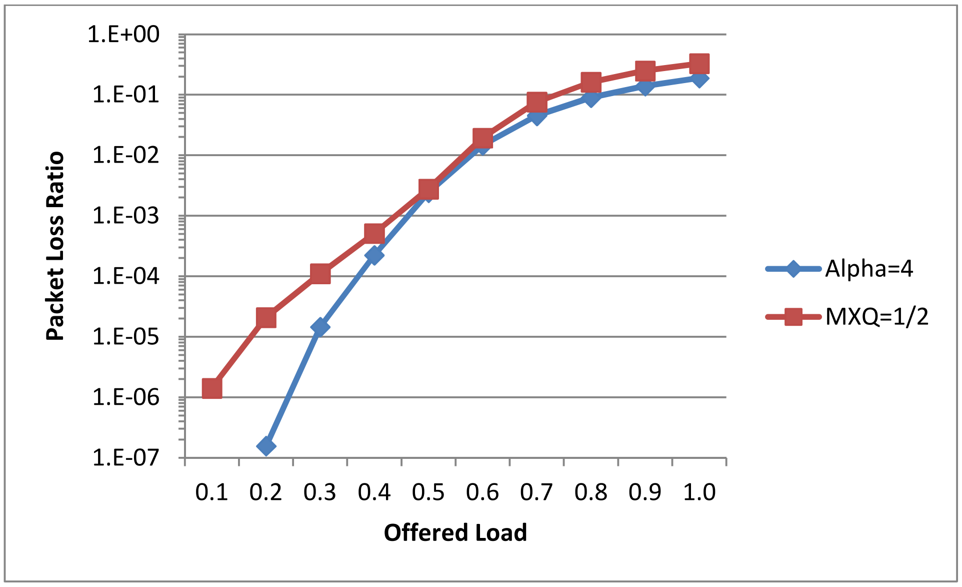


Figure 5.22-SMXQvsDT; Lo; Ports=16; Buffer=4096; ABL=256

As it can be observed in figures 5.23 and 5.24, for a buffer size equal to 8192, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to3.78E-02, or a 86.84% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/8, and a

decrease in packet loss ratio of up to 1.02E-01, or a 43.35% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

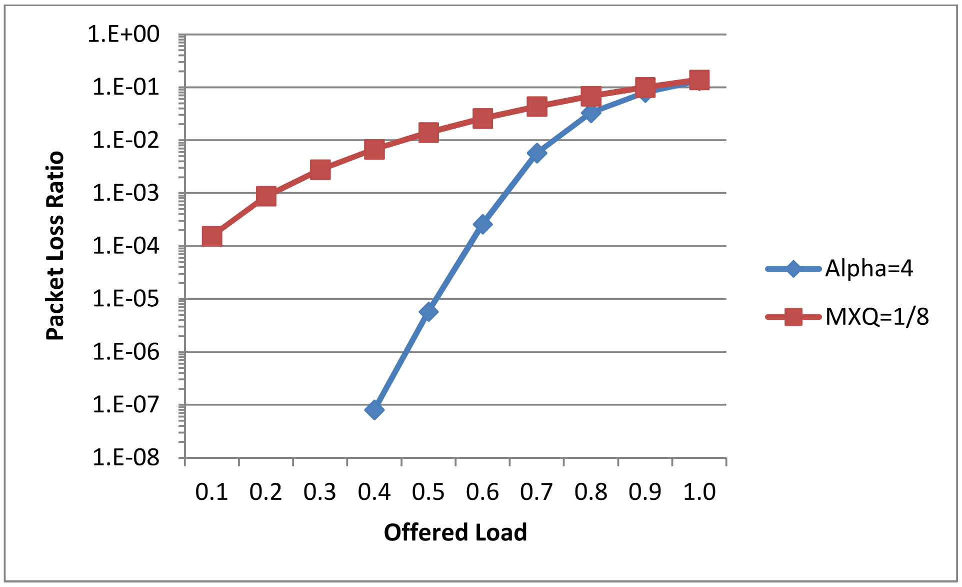


Figure 5.23-SMXQvsDT; Hi; Ports=16; Buffer=8192; ABL=256

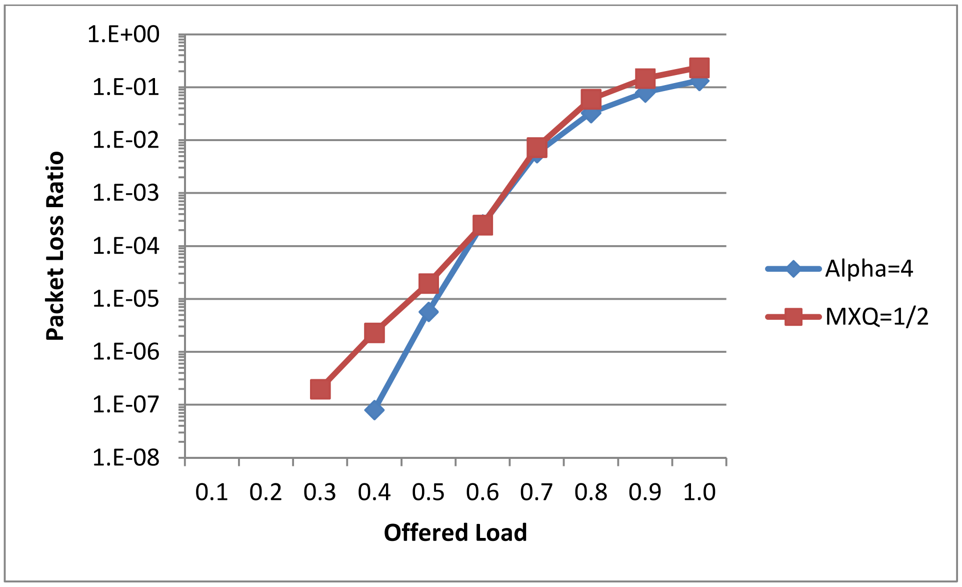


Figure 5.24-SMXQvsDT; Lo; Ports=16; Buffer=8192; ABL=256

Figures 5.25 and 5.26 show that for a buffer size equal to 16384, DT performs better under both high and low loads, presenting a decrease in packet loss ratio of up to 2.23E-02, or a 90.48% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/8, and a decrease in packet loss ratio of up to t6.70E-02, or a 44.06% improvement of DT with Alpha of 4 over SMXQ with MXQ of 1/2.

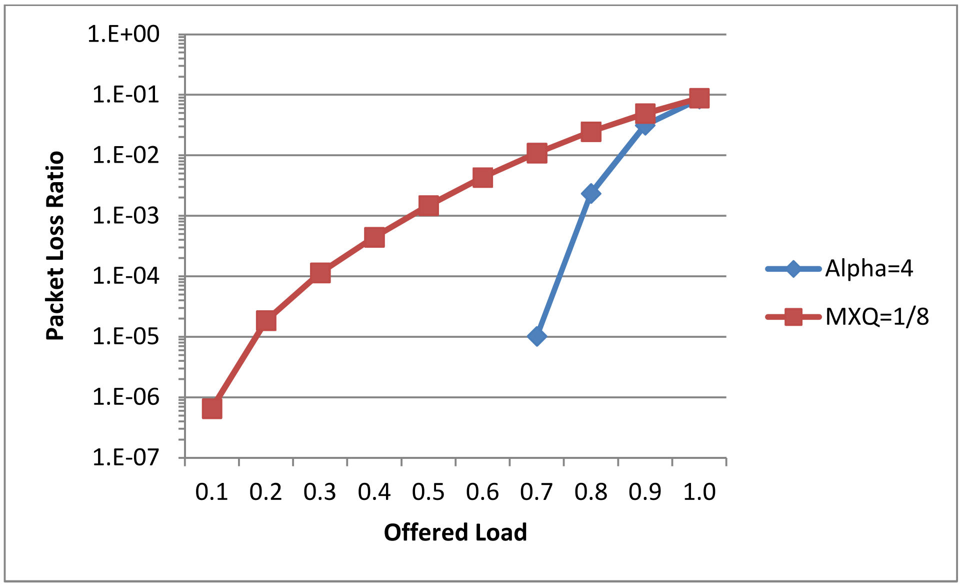


Figure 5.25-SMXQvsDT; Hi; Ports=16; Buffer=16384; ABL=256

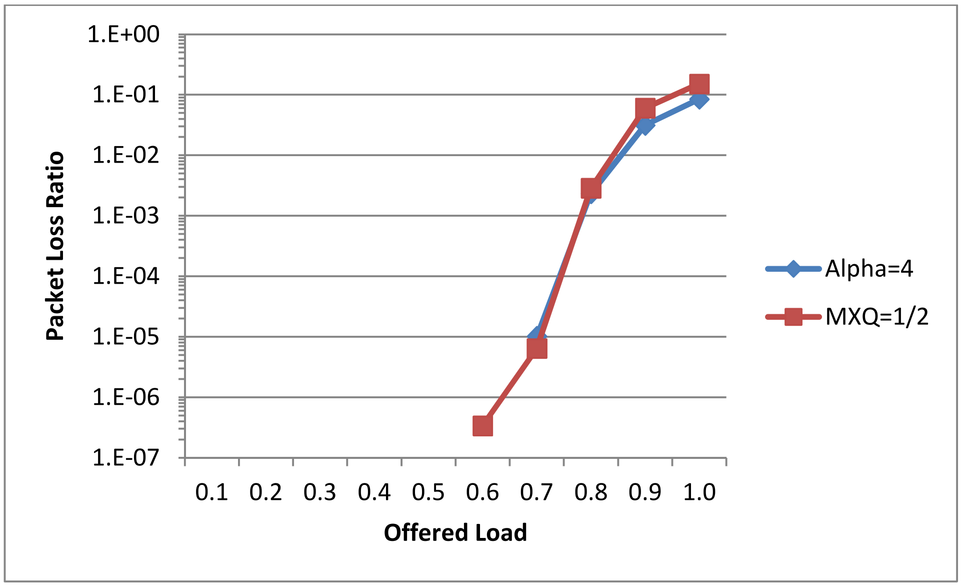


Figure 5.26-SMXQvsDT; Lo; Ports=16; Buffer=16384; ABL=256

5.2 SMA vs. DT

5.2.1 Performance Evaluation under Different Average Burst Length

As it can be observed in figures 5.27 and 5.28, for an ABL equal to 64, DT performs better under every offered load, presenting a decrease in packet loss ratio of up to 6.26E-02, or a 41.30% improvement of DT with Alpha of 2 over SMA with MA of 1/2, and a decrease in packet loss ratio of up to t5.25E-02, or a 37.07% improvement of DT with Alpha of 4 over SMA with MA of 1/8.

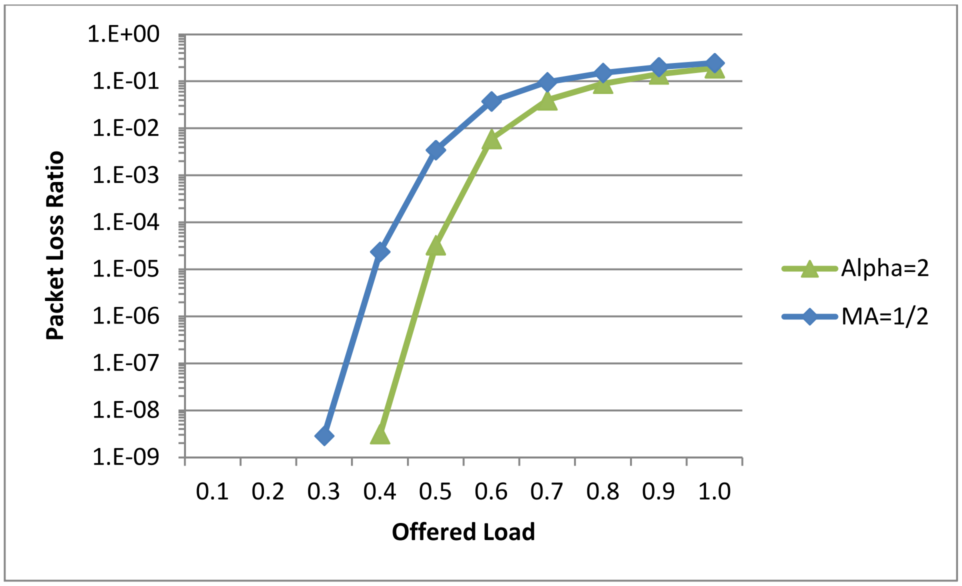


Figure 5.27-SMAvsDT; Hi; Ports=64; Buffer=4096; ABL=64

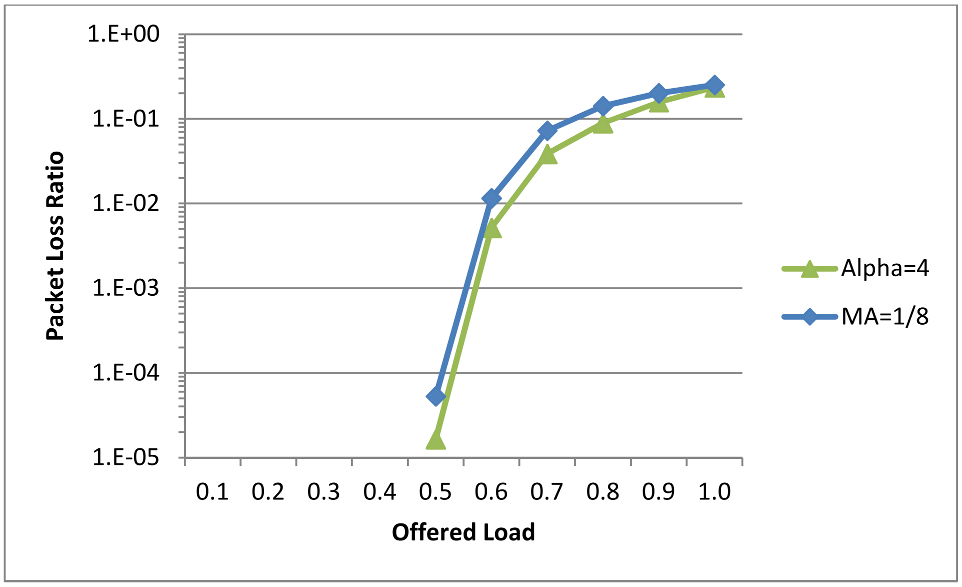


Figure 5.28-SMAvsDT; Lo; Ports=64; Buffer=4096; ABL=64

From figures 5.29 and 5.30 it can be observed that for an ABL equal to 128, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 6.33E-02, or a 37.86% improvement of DT with Alpha of 2 over SMA with MA of 1/2, and a decrease in packet loss ratio of up to 5.20E-02, or a 25.07% improvement of DT with Alpha of 4 over SMA with MA of 1/8.

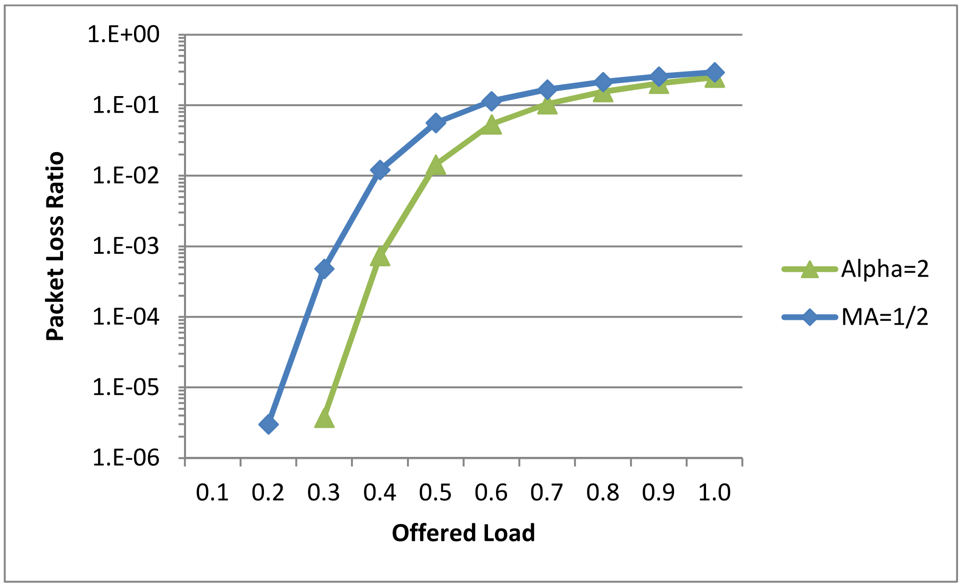


Figure 5.29-SMAvsDT; Hi; Ports=64; Buffer=4096; ABL=128

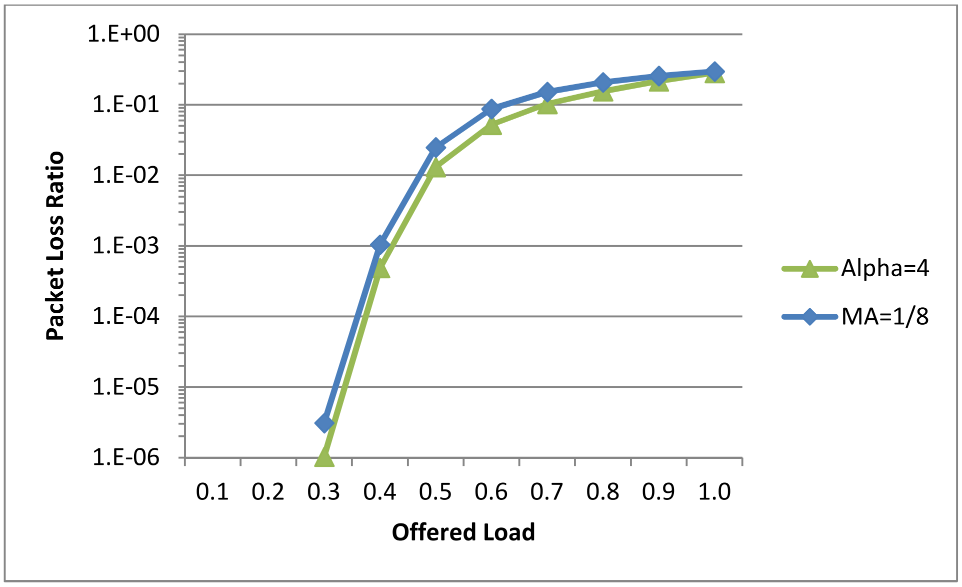


Figure 5.30-SMAvsDT; Lo; Ports=64; Buffer=4096; ABL=128

Figures 5.31 and 5.32 show that for an ABL equal to 256, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 5.20E-02, or a 30.44% improvement of DT with Alpha of 1 over SMA with MA of 1/2, and a decrease in packet loss ratio of up to 4.03E-02, or a 19.58% improvement of DT with Alpha of 4 over SMA with MA of 1/8.

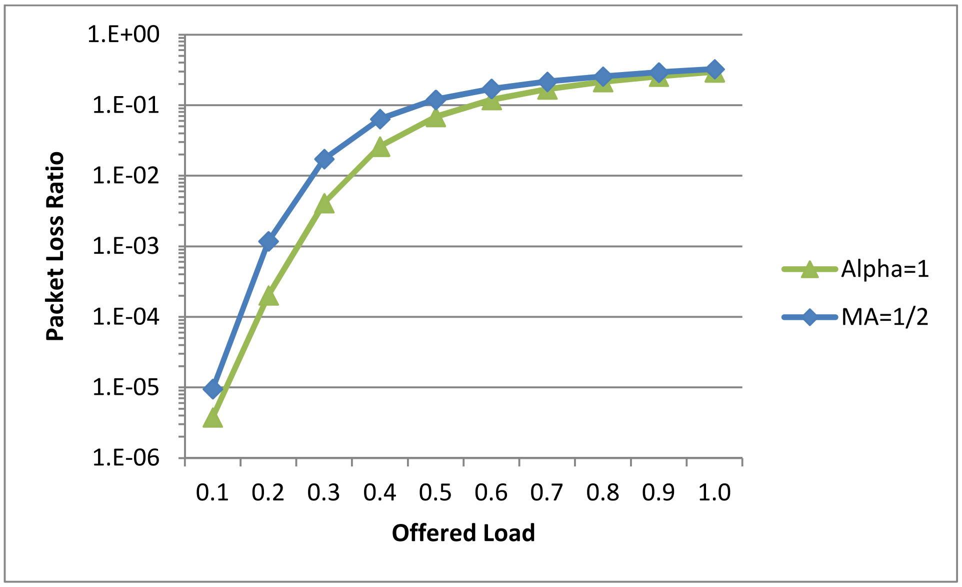


Figure 5.31-SMAvsDT; Hi; Ports=64; Buffer=4096; ABL=256

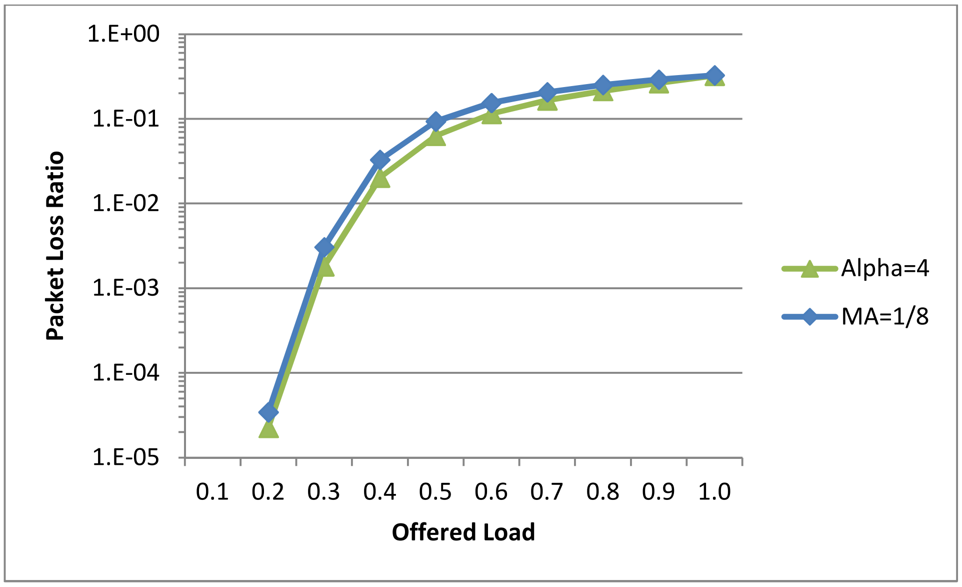


Figure 5.32-SMAvsDT; Lo; Ports=64; Buffer=4096; ABL=256

5.2.2 Performance Evaluation under Different Number of Ports

As it can be observed in figures 5.33 and 5.34, for a number of ports equal to 16, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 4.67E-02, or a 25.07% improvement of DT with Alpha of 4 over SMA with MA of 1/2. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 50%, it is instead SMA with a MA of 1/8, with a decrease in packet loss ratio of up to 6.25E-06, or a 2.80% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

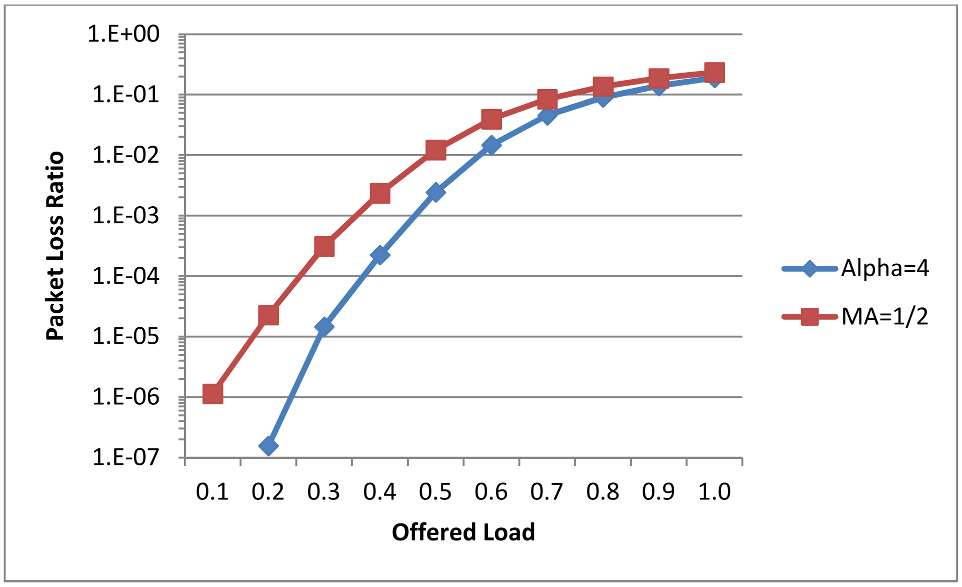


Figure 5.33-SMAvsDT; Hi; Ports=16; Buffer=4096; ABL=256

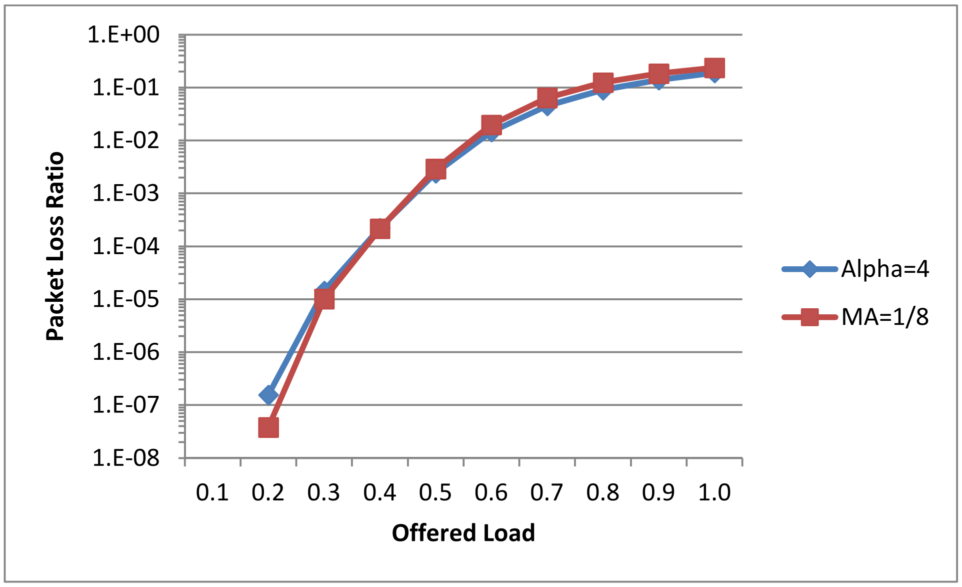


Figure 5.34-SMAvsDT; Lo; Ports=16; Buffer=4096; ABL=256

As it can be observed in figures 5.35 and 5.36, for a number of ports equal to 32, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 5.67E-02, or a 35.25% improvement of DT with Alpha of 4 over SMA with MA of 1/2. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 30%, it is instead

SMA with a MA of 1/8, with decrease in packet loss ratio of up to 6.05E-07, or a 26.63% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

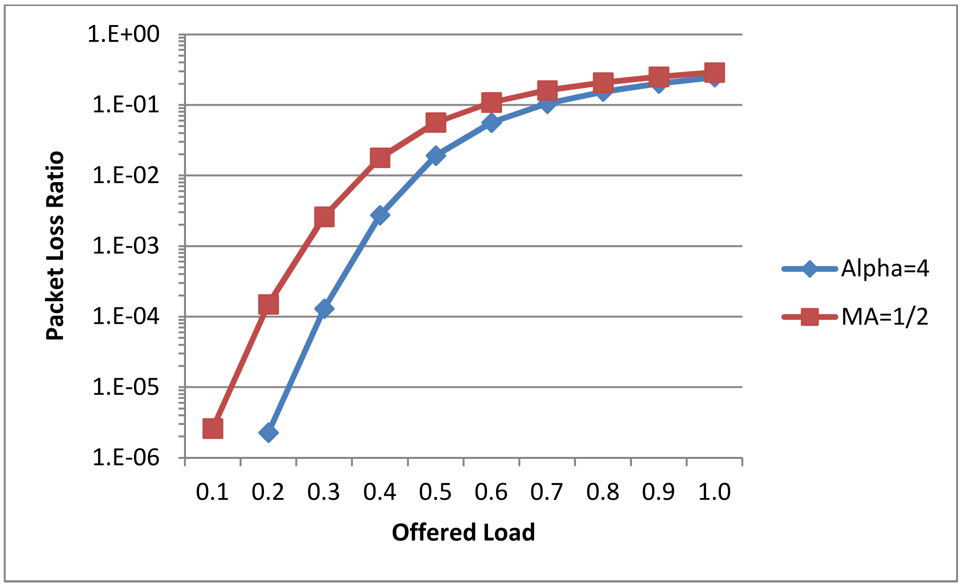


Figure 5.35-SMAvsDT; Hi; Ports=32; Buffer=4096; ABL=256

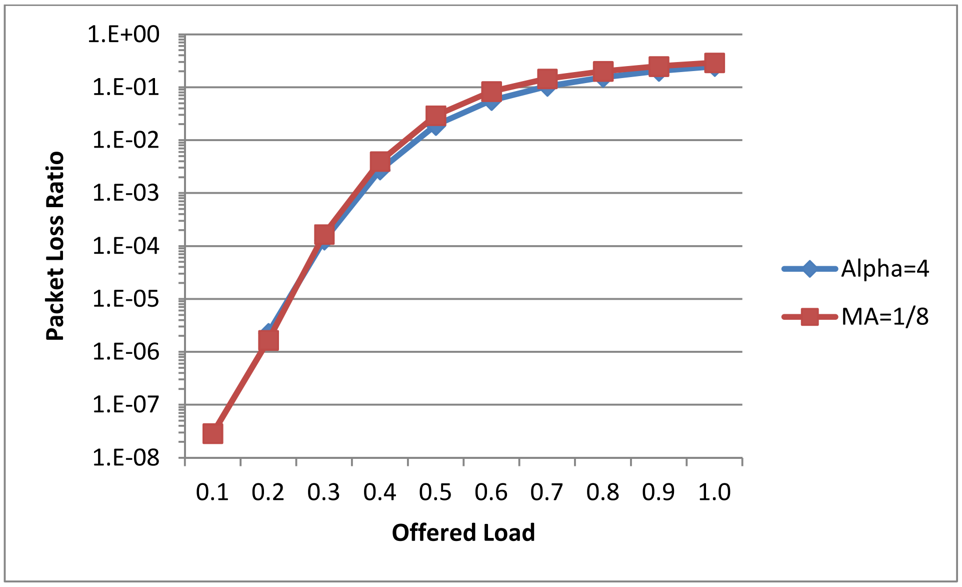


Figure 5.36-SMAvsDT; Lo; Ports=32; Buffer=4096; ABL=256

Figures 5.37 and 5.38 show that for a number of ports equal to 64, DT performs better under all offered loads, presenting a decrease in packet loss ratio of up to 5.20E-02, or a 30.44% improvement of DT with Alpha of 1 over SMA with MA of 1/2, and a decrease in packet loss ratio of up to 4.03E-02, or a 19.58% improvement of DT with Alpha of 4 over SMA with MA of 1/8.

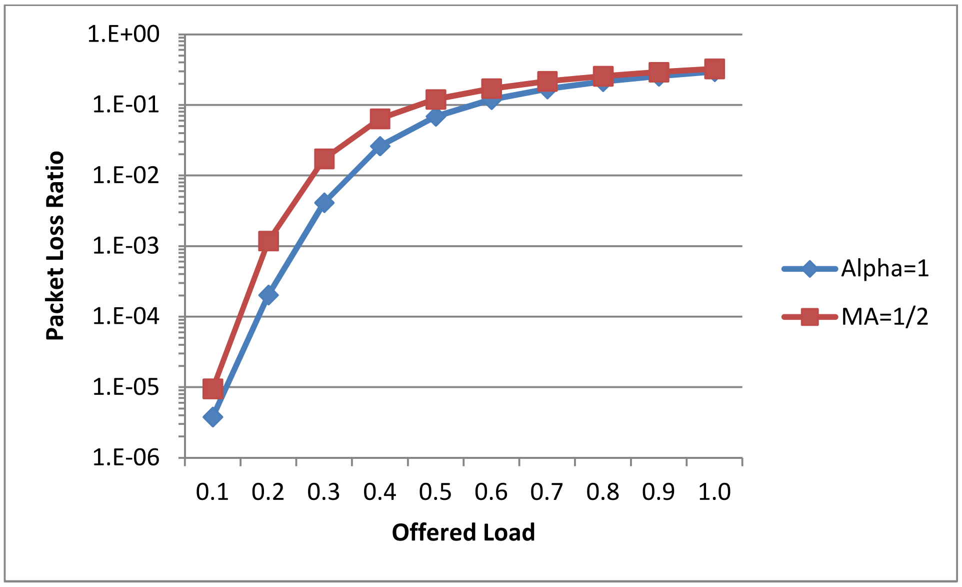


Figure 5.37-SMAvsDT; Hi; Ports=64; Buffer=4096; ABL=256

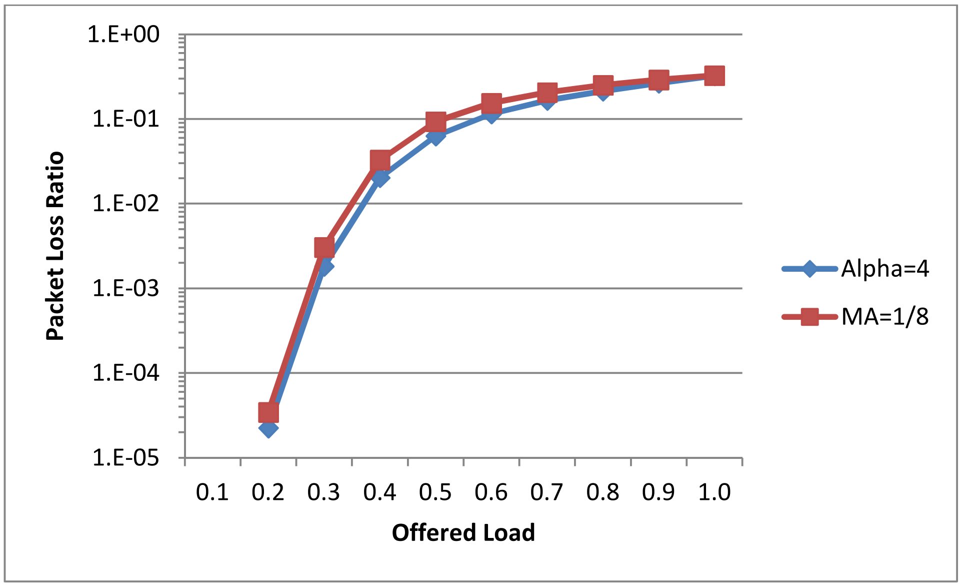


Figure 5.38-SMAvsDT; Lo; Ports=64; Buffer=4096; ABL=256

5.2.3 Performance Evaluation under Different Buffer Size

As it can be observed in figures 5.39and 5.40, for buffer size equal to 256 DT performs better under high loads, presenting a decrease in packet loss ratio of up to 6.80E-03, or a 2.42%

improvement of DT with Alpha of 2 over SMA with MA of 1/8. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 30%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 2.60E-02, or a 7.05% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

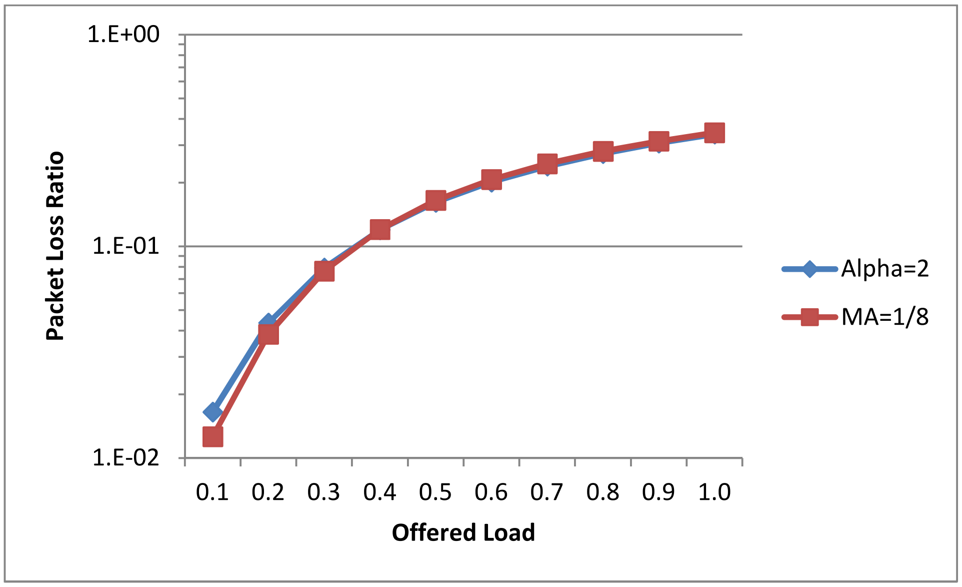


Figure 5.39-SMAvsDT; Hi; Ports=16; Buffer=256; ABL=256

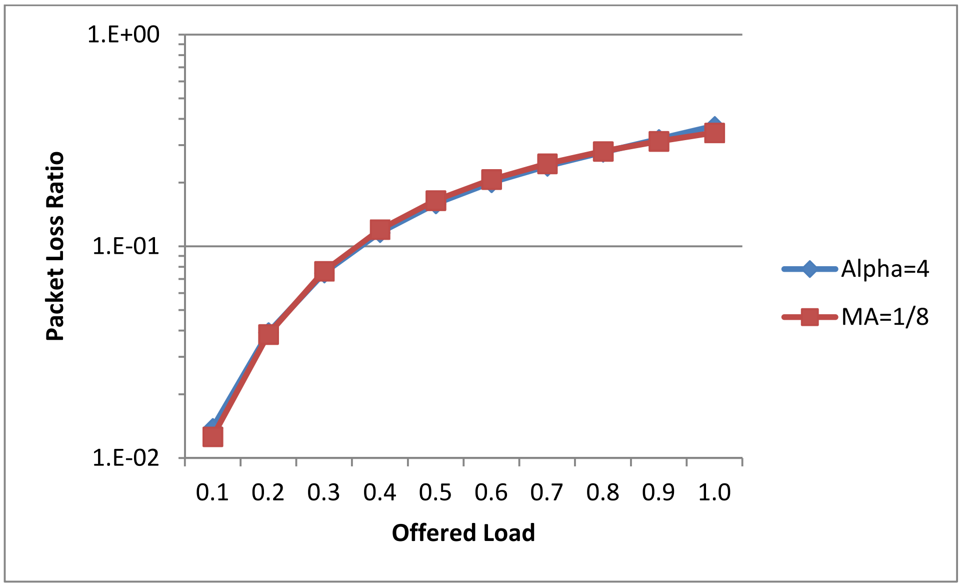


Figure 5.40-SMAvsDT; Lo; Ports=16; Buffer=256; ABL=256

Figures 5.41 and 5.42 show that for buffer size equal to 512, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 1.37E-02, or a 5.21% improvement of DT with Alpha of 2 over SMA with MA of 1/8. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 30%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 7.54E-04, or a 4.23% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

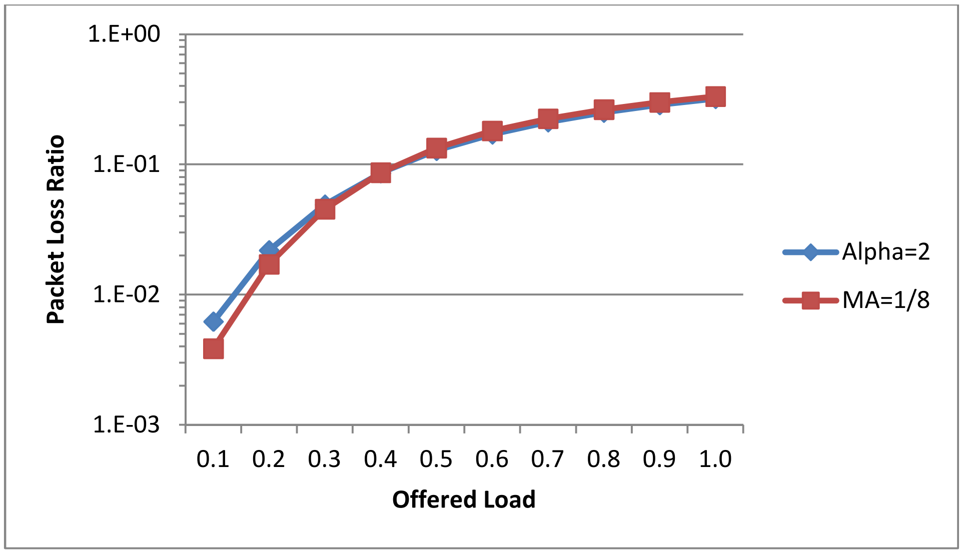


Figure 5.41-SMAvsDT; Hi; Ports=16; Buffer=512; ABL=256

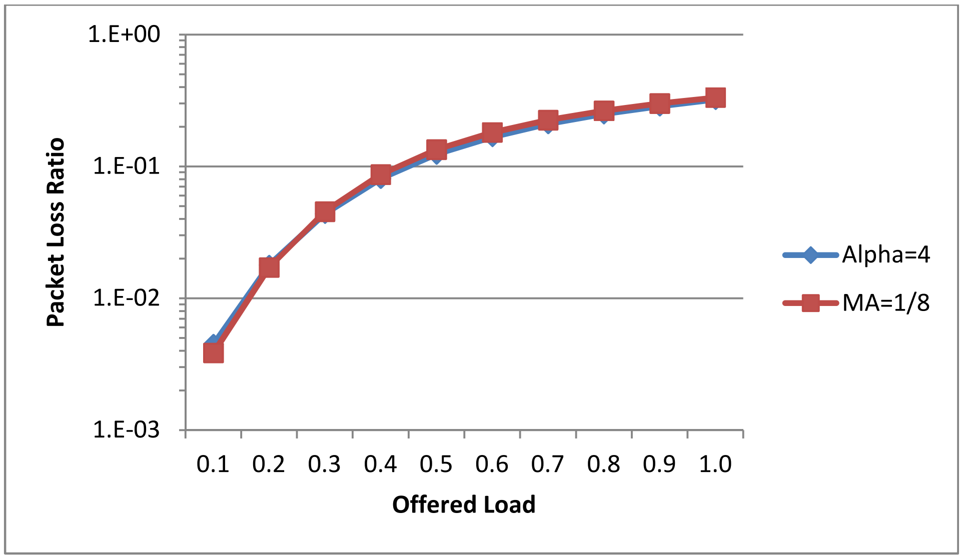


Figure 5.42-SMAvsDT; Lo; Ports=16; Buffer=512; ABL=256

From figures 5.43 and 5.44 it can be observed that for a buffer size equal to 1024, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 2.60E-02, or a 11.08% improvement of DT with Alpha of 4 over SMA with MA of 1/8. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 30%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 3.53E-04, or a 9.06% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

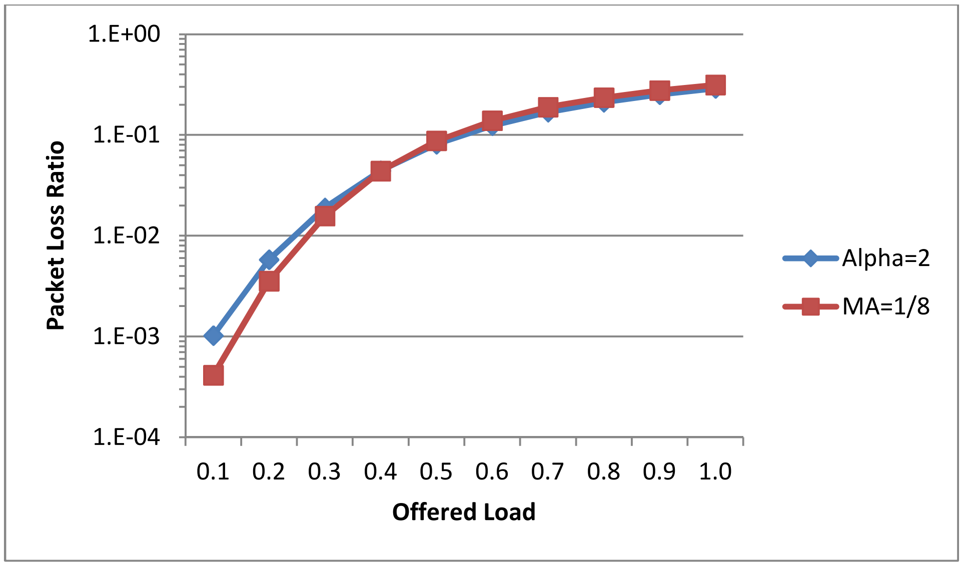


Figure 5.43-SMAvsDT; Hi; Ports=16; Buffer=1024; ABL=256

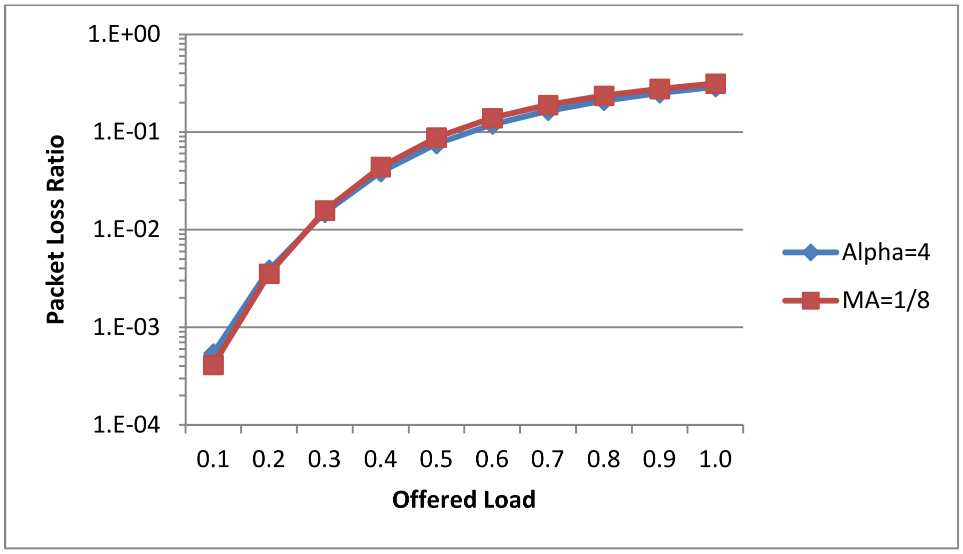


Figure 5.44-SMAvsDT; Lo; Ports=16; Buffer=1024; ABL=256

As it can be observed in figure 5.45 and 5.46, for buffer size equal to 2048, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 4.48E-02, or a 29.75% improvement of DT with Alpha of 4 over SMA with MA of 1/2. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 40%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 4.82E-05, or a 25.28% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

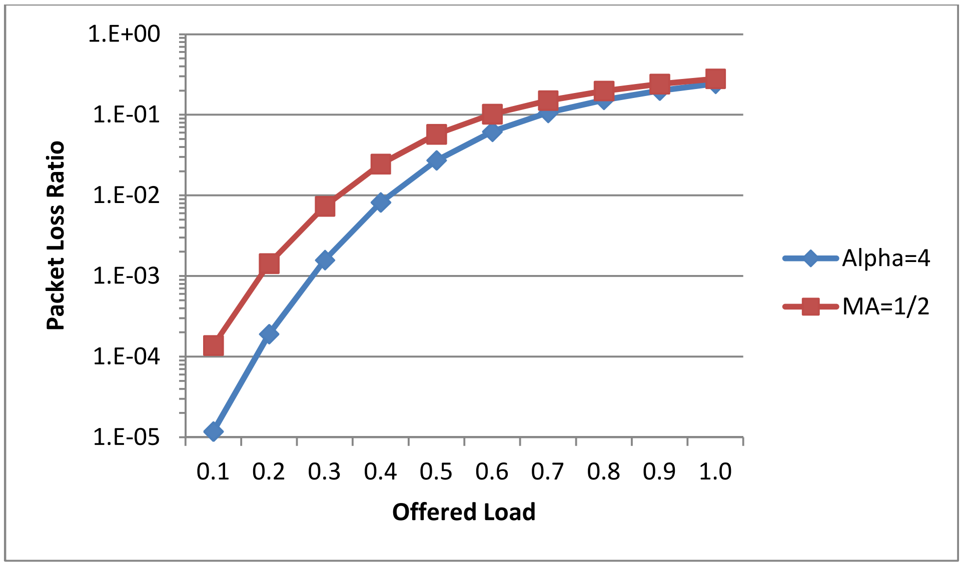


Figure 5.45-SMAvsDT; Hi; Ports=16; Buffer=2048; ABL=256

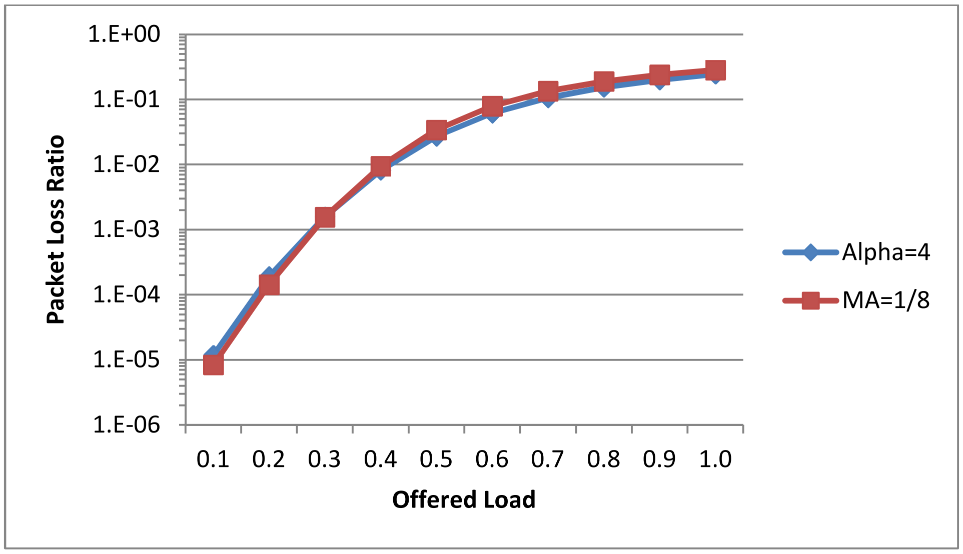


Figure 5.46-SMAvsDT; Lo; Ports=16; Buffer=2048; ABL=256

Figures 5.47 and 5.48 show that for a buffer size equal to 4096, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 4.67E-02, or a 25.07% improvement of DT with Alpha of 4 over SMA with MA of 1/2. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 50%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 6.25E-06, or a 2.80% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

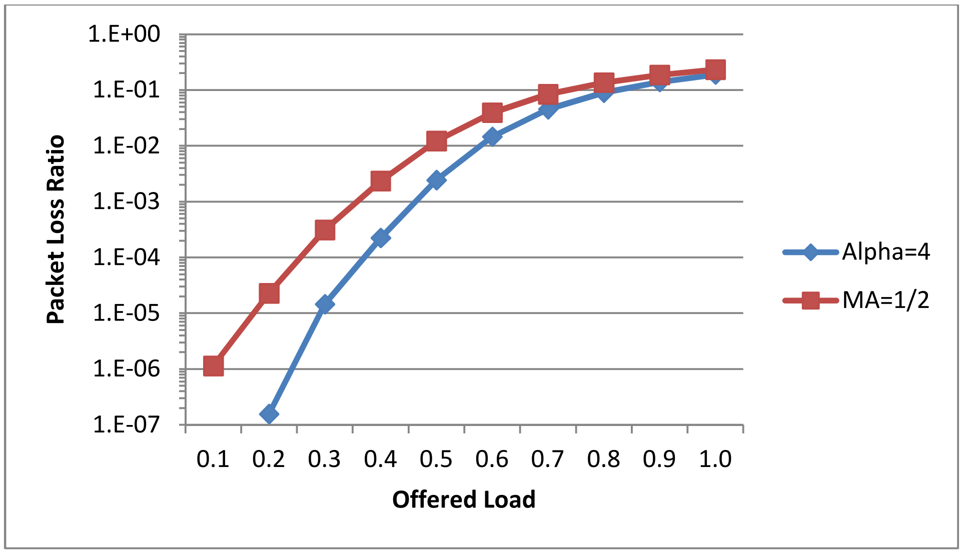


Figure 5.47-SMAvsDT; Hi; Ports=16; Buffer=4096; ABL=256

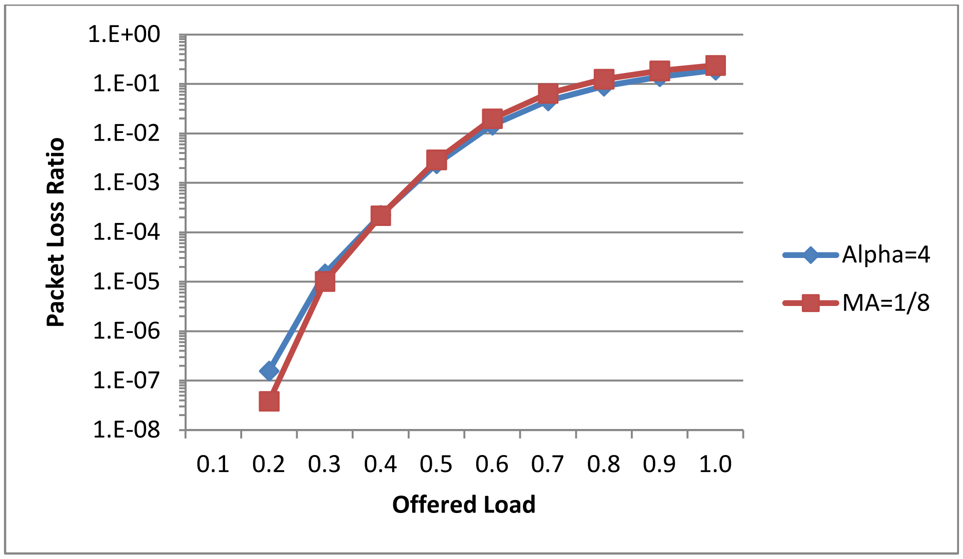


Figure 5.48-SMAvsDT; Lo; Ports=16; Buffer=4096; ABL=256

As it can be observed in figures 5.49 and 5.50, for a buffer size equal to 8192, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 4.06E-02, or a 23.35% improvement of DT with Alpha of 4 over SMA with MA of 1/2. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 60%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 1.23E-06, or a 21.44% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

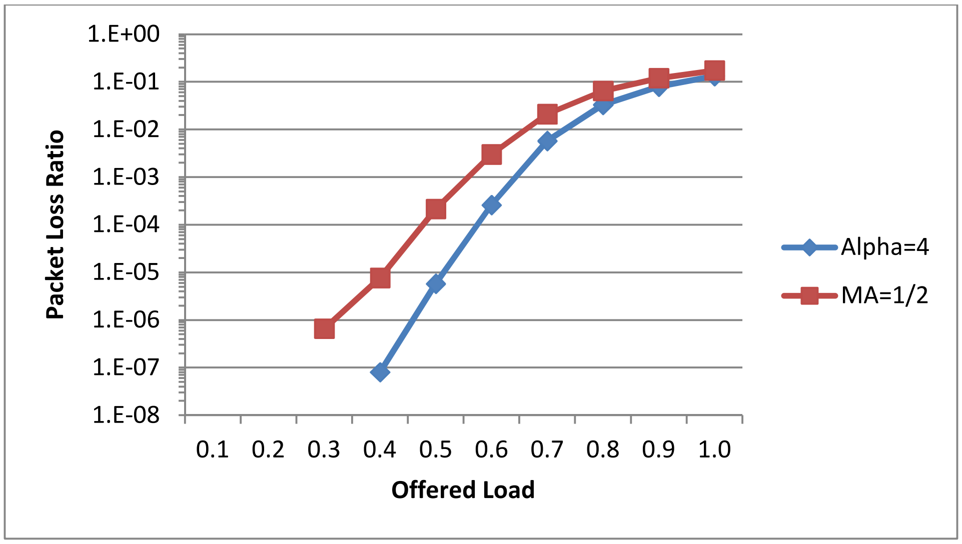


Figure 5.49-SMAvsDT; Hi; Ports=16; Buffer=8192; ABL=256

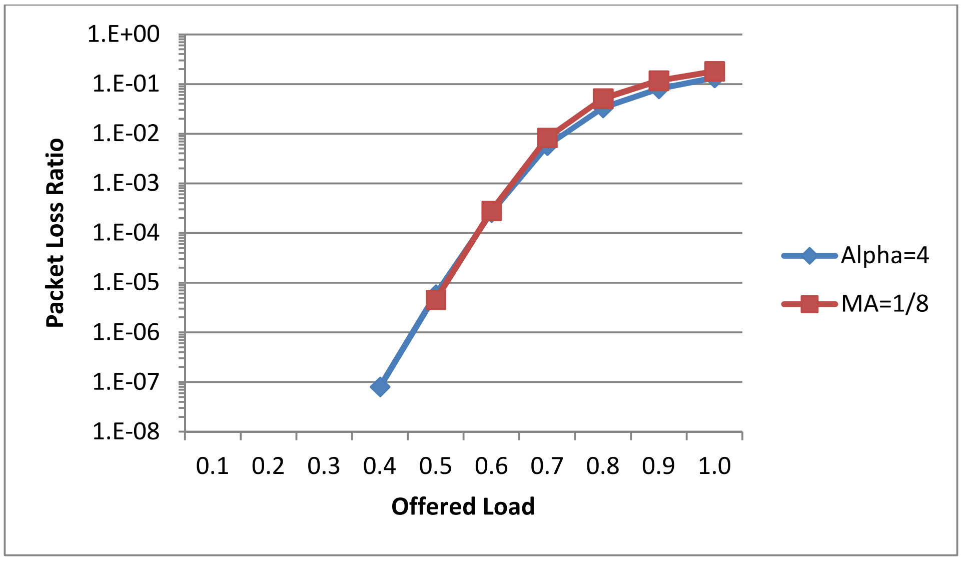


Figure 5.50-SMAvsDT; Lo; Ports=16; Buffer=8192; ABL=256

As it can be observed in figures 5.51 and 5.52, for a buffer size equal to 16384, DT performs better under high loads, presenting a decrease in packet loss ratio of up to 3.04E-02, or a 26.30% improvement of DT with Alpha of 4 over SMA with MA of 1/2. DT however is not the scheme with the lowest packet loss ratio when subjected to loads lower than 80%, it is instead SMA with a MA of 1/8, with decrease in packet loss ratio of up to 2.16E-06, or a 21.07% improvement of SMA with MA of 1/8 over DT with Alpha of 4.

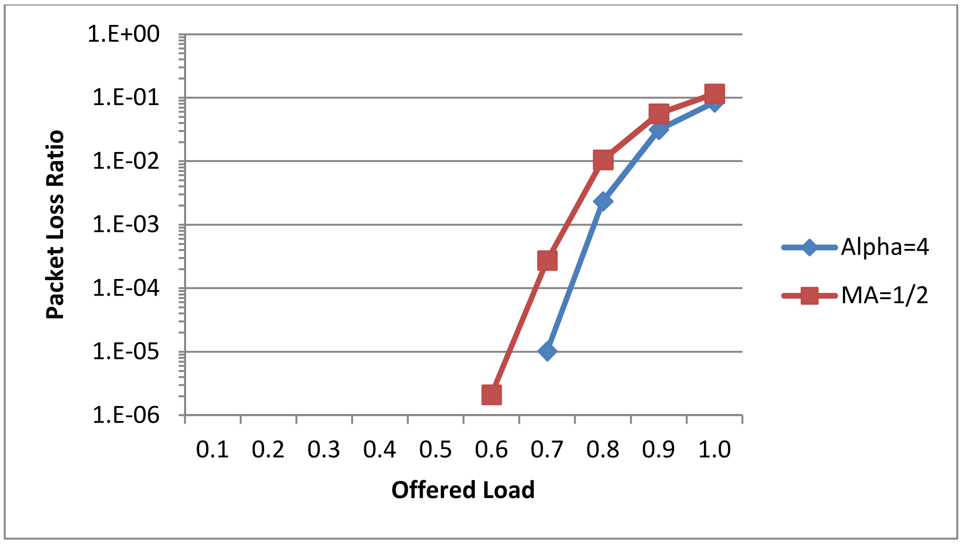


Figure 5.51-SMAvsDT; Hi; Ports=16; Buffer=16384; ABL=256

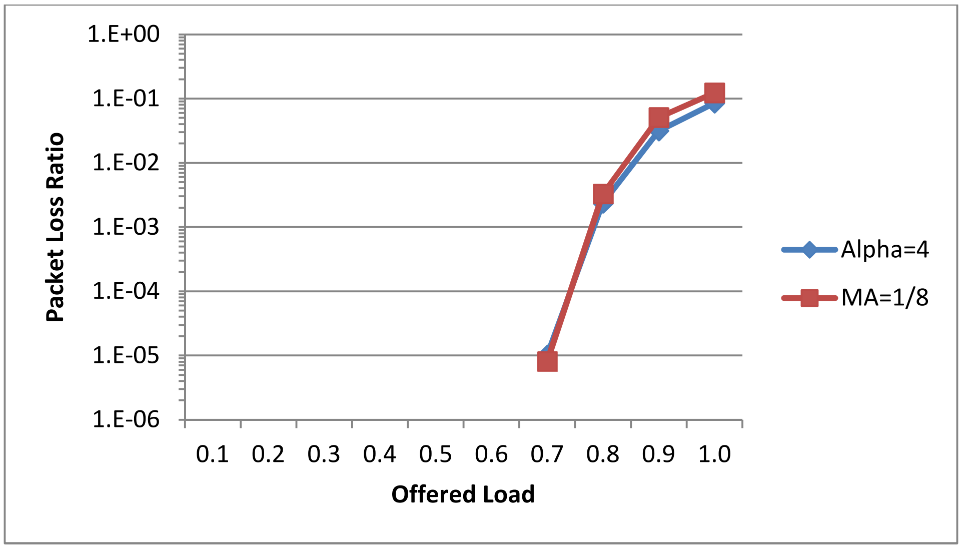


Figure 5.52-SMAvsDT; Lo; Ports=16; Buffer=16384; ABL=256

5.3 Summary of Results

As it can be observed in tables 5.1, 5.2 and 5.3, the scheme Dynamic Threshold always has a lower packet loss ratio than its static counterpart Sharing with Maximum Queue Lengths. However, DT is not always the best, under the tested scenarios it was found that a smaller number of ports, as well as a bigger available buffer increase the range of lower-end loads for which SMA with a minimum allocation of 1/8 trumps over DT with any of the tested Alphas, making it the best of the conventional schemes under loads of up 70% when the number of ports is 16 and the available buffer is 16384.

ABL	Scheme	Variable	Load%
64	DT	4	10-70
	DT	2	80-100
128	DT	4	10-70
	DT	2	80-100
256	DT	4	10-80
	DT	1	90-100

Table 5.1-Best Conventional Scheme for Different ABLs. NxN=64x64; Buffer=4096

Ports	Scheme	Variable	Load%
16	SMA	1/8	10-40
	DT	4	50-100
32	SMA	1/8	10-20
	DT	4	30-100
64	DT	4	10-80
	DT	1	90-100

Table 5.2-Best Conventional Scheme for Different Number of Ports. ABL=256; Buffer=4096

Buffer	Scheme	Variable	Load%
256	SMA	1/8	10-20
	DT	4	30-60
	DT	2	70-100
512	SMA	1/8	10-20
	DT	4	30-90
	DT	2	100
1024	SMA	1/8	10-20
	DT	4	30-100
2048	SMA	1/8	10-30
	DT	4	40-100
4096	SMA	1/8	10-40
	DT	4	50-100
8192	SMA	1/8	10-50
	DT	4	60-100
16384	SMA	1/8	10-70
	DT	4	80-100

Table 5.3-Best Conventional Scheme for Different Buffer Sizes. ABL=256; NxN=16

CHAPTER VI

SQF VS BEST CONVENTIONAL SHARING MEMORY SCHEME

In this chapter we test and evaluate the proposed scheme Shortest Queue First (SQF) under the same simulated scenarios as the best conventional scheme defined in the previous chapter in order to measure the extent to which SQF reduces packet loss ratio.

6.1 Performance Evaluation under Different Average Burst Length

As it can be observed in figure 6.1, for an ABL equal to 64, SQF performs the best under every offered load, presenting a decrease in packet loss ratio of up to 1.66E-03, or a 0.86% improvement of SQF over DT with an Alpha of 2, and a decrease in packet loss ratio of up to 4.61E-02, or a 19.54% improvement of SQF over DT with Alpha of 4.

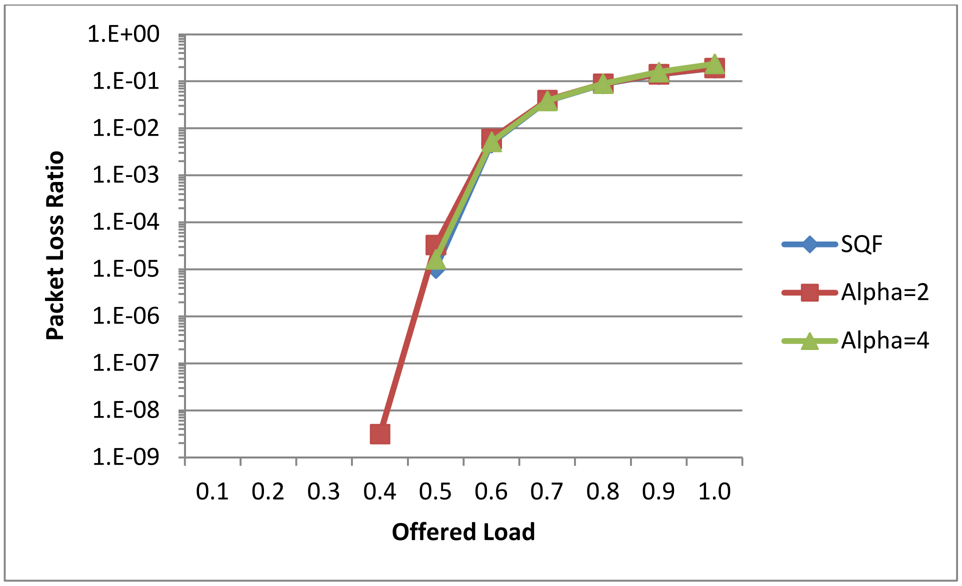


Figure 6.1-SQFvsBest; Ports=64; Buffer=4096; ABL=64

From figure 6.2 it can be observed that for an ABL equal to 128, SQF performs better every tested load, presenting a decrease in packet loss ratio of up to 2.88E-03, or a 1.16% improvement of SQF over DT with an Alpha of 2, and a decrease in packet loss ratio of up to 3.74E-02, or a 13.19% improvement of SQF over DT with Alpha of 4.

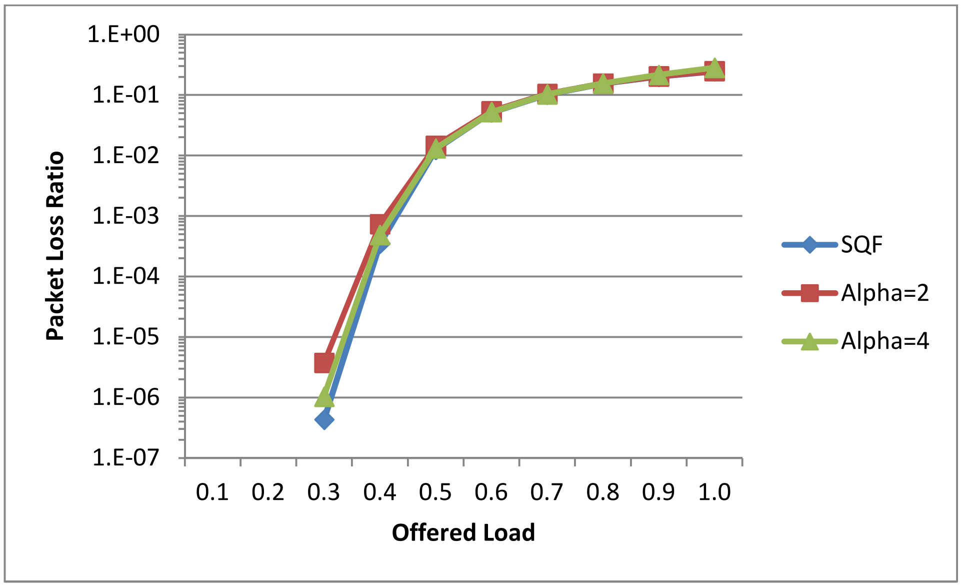


Figure 6.2-SQFvsBest; Ports=64; Buffer=4096; ABL=128

Figure 6.3 shows that for an ABL equal to 256, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 7.01E-03, or a 26.88% improvement of SQF over DT with an Alpha of 1, and a decrease in packet loss ratio of up to 3.21E-02, or a 9.93% improvement of SQF over DT with Alpha of 4.

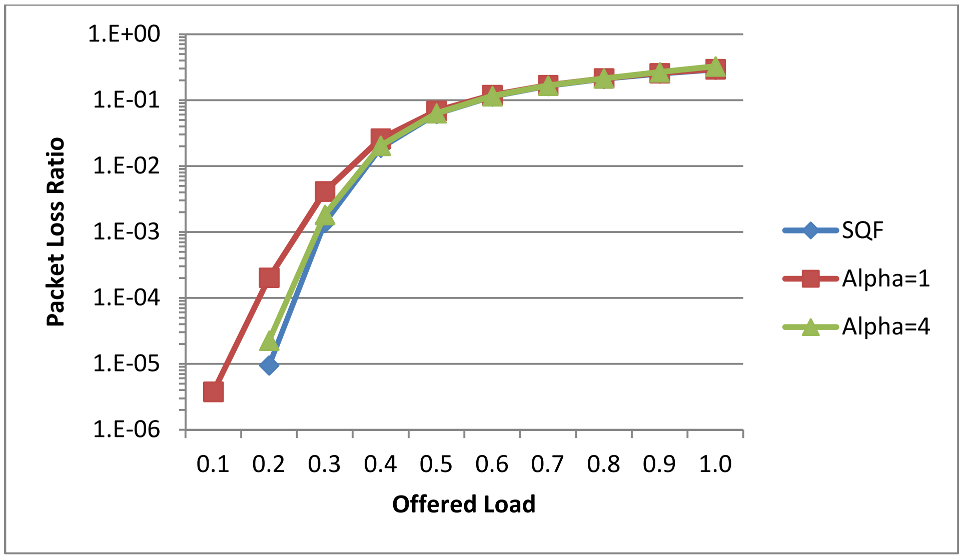


Figure 6.3-SQFvsBest; Ports=64; Buffer=4096; ABL=256

6.2 Performance Evaluation under Different Number of Ports

As it can be observed in figure 6.4, for a number of ports equal to 16, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to3.05E-03, or a 6.73% improvement of SQF over DT with an Alpha of 4, and a decrease in packet loss ratio of up to 5.04E-02, or a 21.26% improvement of SQF over SMA with MA equal to 1/8.

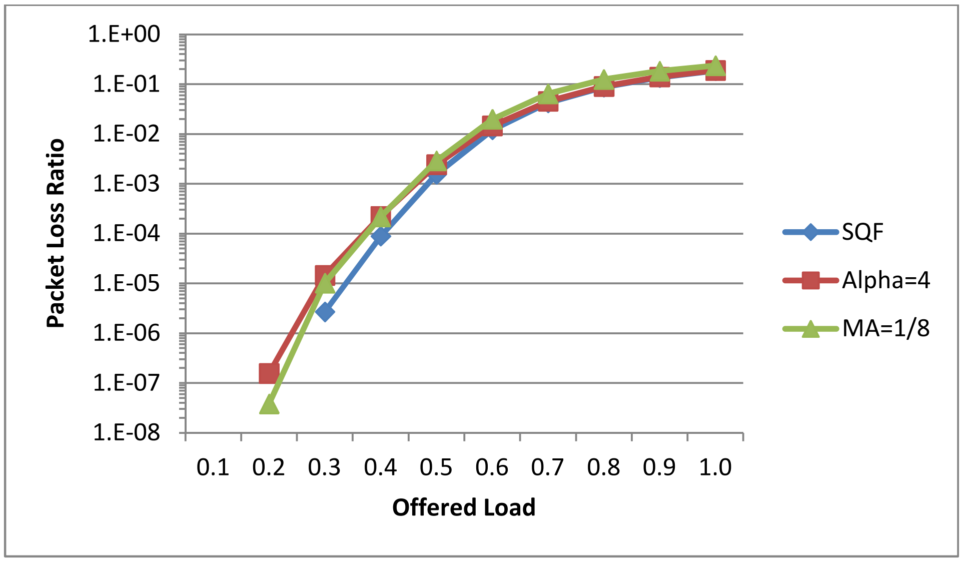


Figure 6.4-SQFvsBest; Ports=16; Buffer=4096; ABL=256

Figure 6.5 shows that for a number of ports equal to 32, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 2.11E-03, or a 0.86% improvement of SQF over DT with an Alpha of 4.

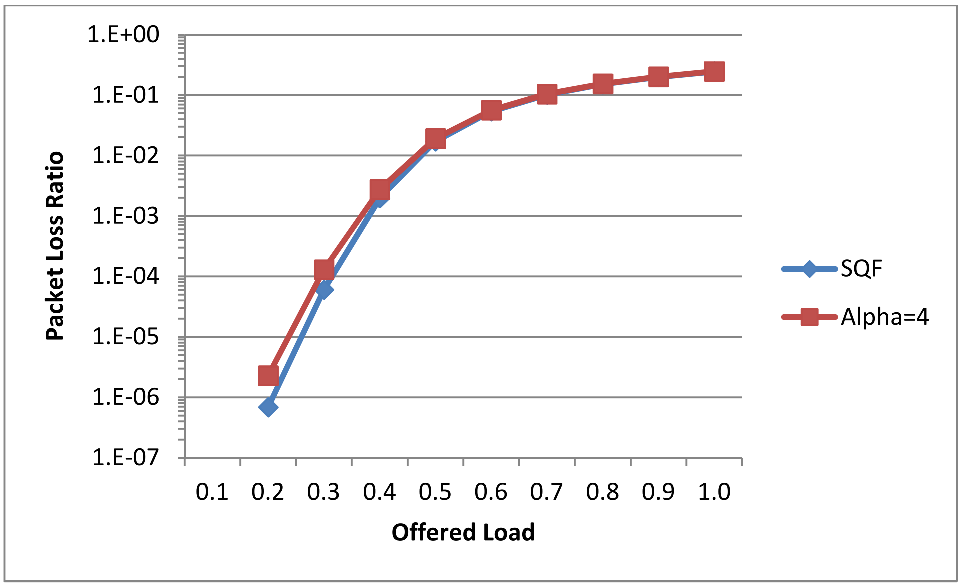


Figure 6.5-SQFvsBest; Ports=32; Buffer=4096; ABL=256

From figure 6.6 it can be observed that for a number of ports equal to 64, SQF performs better under both high and low loads, presenting a decrease in packet loss ratio of up to 7.01E-03, or a 26.88% improvement of SQF over DT with an Alpha of 1, and a decrease in packet loss ratio of up to 3.21E-02, or a 9.93% improvement of SQF over DT with Alpha of 4.

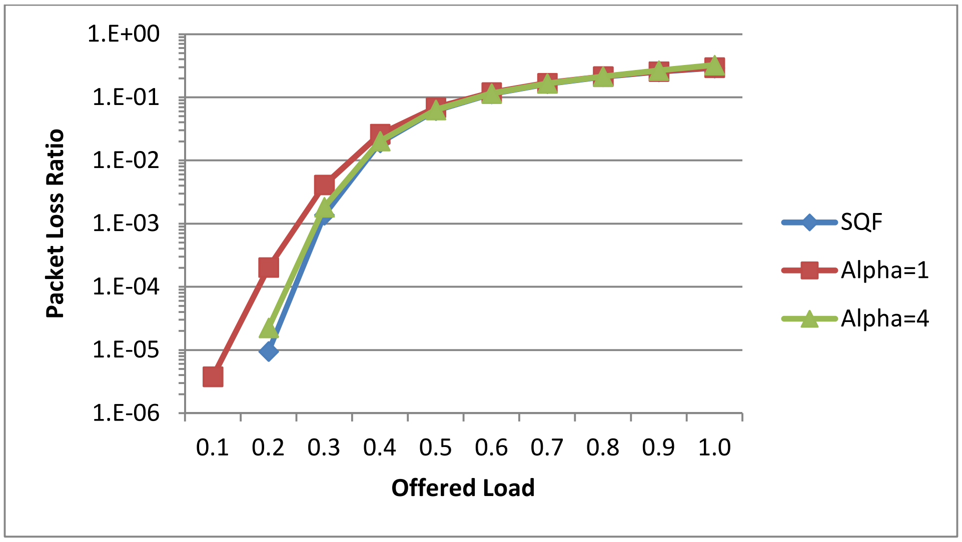


Figure 6.6-SQFvsBest; Ports=64; Buffer=4096; ABL=256

6.3 Performance Evaluation under Different Buffer Size

As it can be observed in figure 6.7, for a buffer size of 256, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 9.17E-03, or a 11.58% improvement of SQF over DT with an Alpha of 2, and a decrease in packet loss ratio of up to 9.44E-03, or a 4.56% improvement of SQF over SMA with MA equal to 1/8.

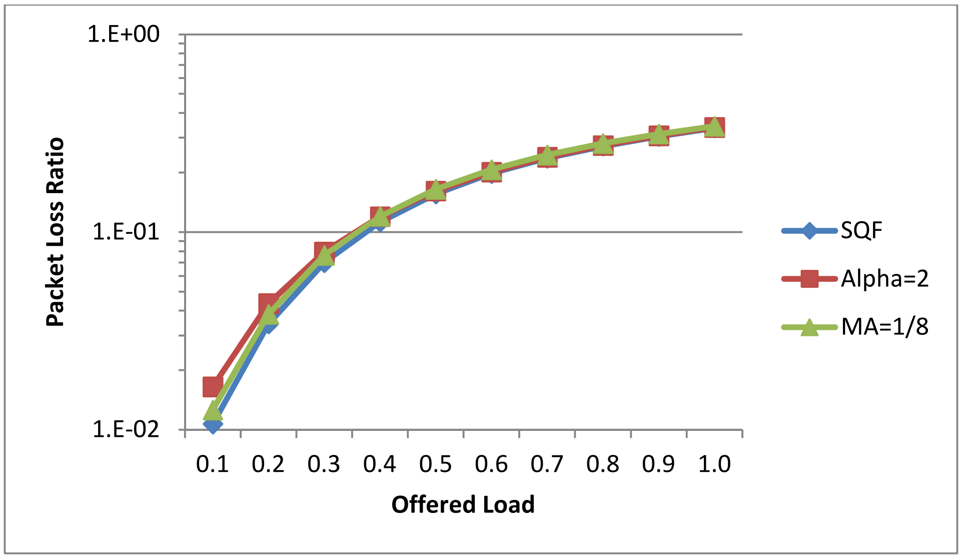


Figure 6.7-SQFvsBest; Ports=16; Buffer=256; ABL=256

Figure 6.8 shows that for a buffer size of 512, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 1.07E-02, or a 21.90% improvement of SQF over DT with an Alpha of 2, and a decrease in packet loss ratio of up to 1.75E-02, or a 7.81% improvement of SQF over SMA with MA equal to 1/8.

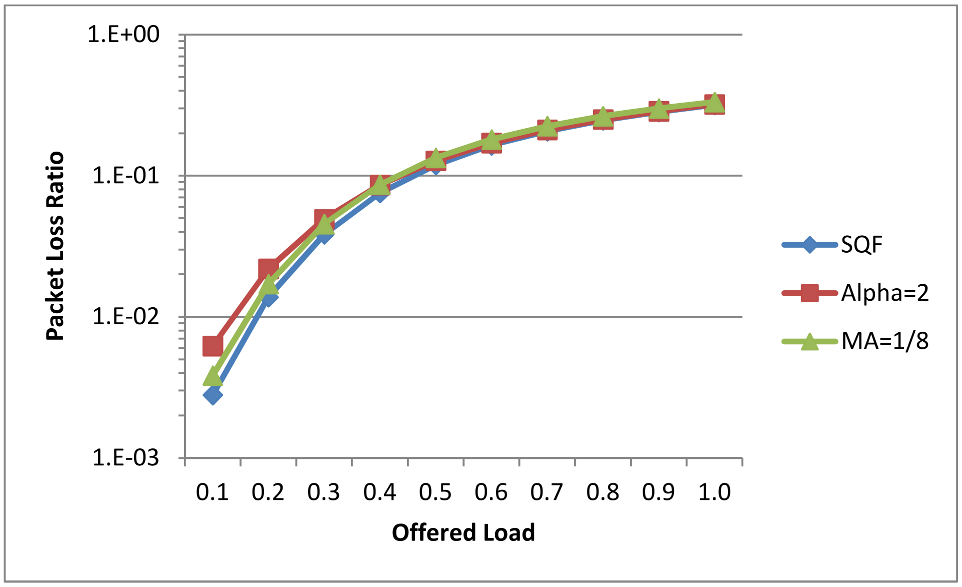


Figure 6.8-SQFvsBest; Ports=16; Buffer=512; ABL=256

From figure 6.9 it can be observed that for a buffer size of 1024, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 6.24E-02, or a 37.90% improvement of SQF over DT with an Alpha of 4, and a decrease in packet loss ratio of up to 8.66E-02, or a 45.86% improvement of SQF over SMA with MA equal to 1/8.

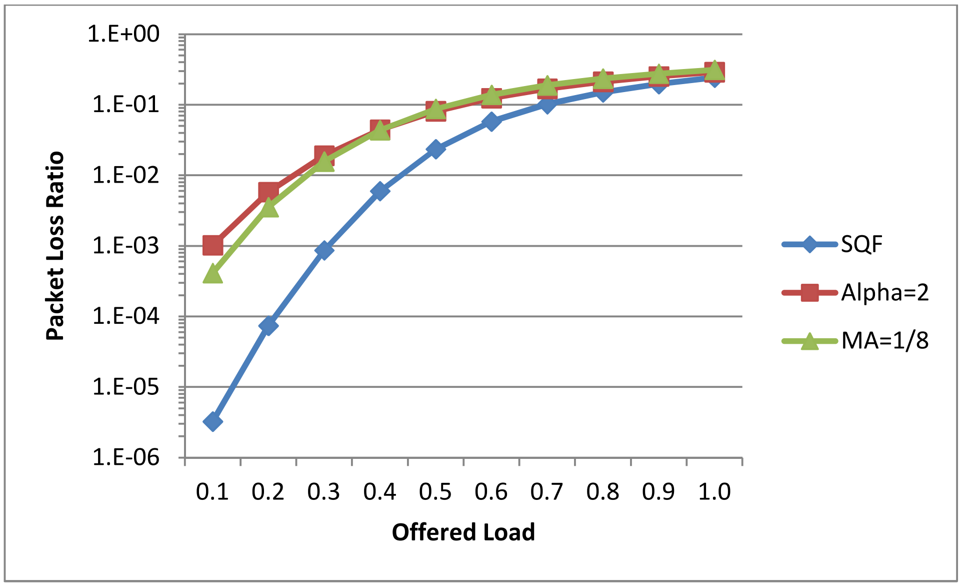


Figure 6.9-SQFvsBest; Ports=16; Buffer=1024; ABL=256

As it can be observed in figure 6.10, for a buffer size of 2048, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 6.64E-02, or a 43.24% improvement of SQF over DT with an Alpha of 4, and a decrease in packet loss ratio of up to 1.02E-01, or a 53.94% improvement of SQF over SMA with MA equal to 1/8.

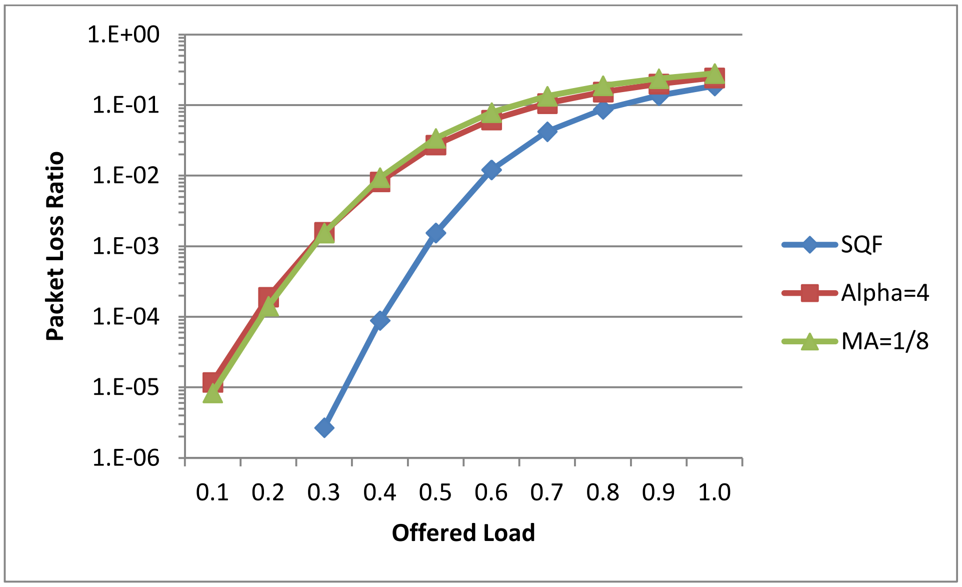


Figure 6.10-SQFvsBest; Ports=16; Buffer=2048; ABL=256

Figure 6.11 shows that for a buffer size of 4096, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 6.20E-02, or a 44.39% improvement of SQF over DT with an Alpha of 4, and a decrease in packet loss ratio of up to1.06E-01 or a 57.79% improvement of SQF over SMA with MA equal to 1/8.

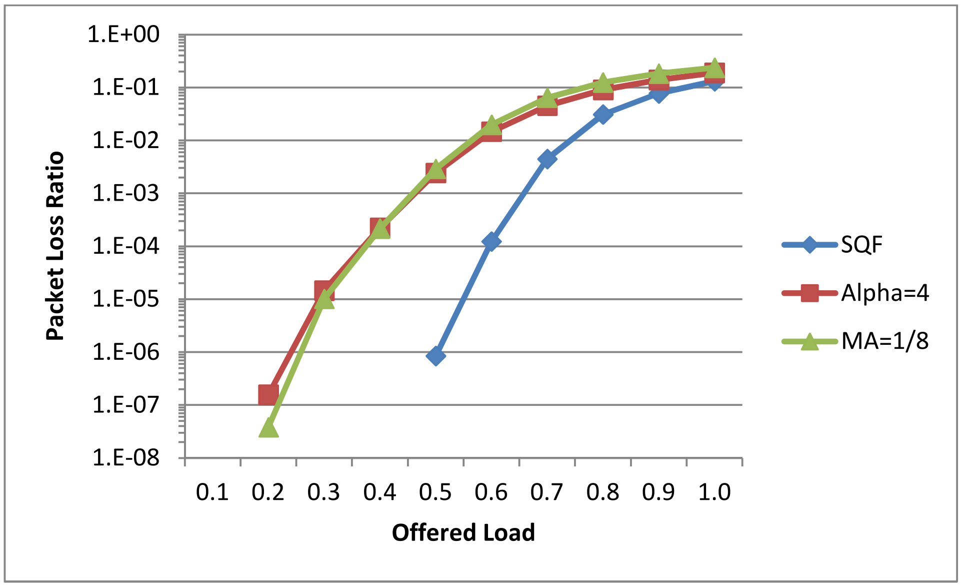


Figure 6.11-SQFvsBest; Ports=16; Buffer=4096; ABL=256

From figure 6.12 it can be observed that for a buffer size of 8192, SQF performs better under both high and low loads, presenting a decrease in packet loss ratio of up to 2.26E-03, or a 2.83% improvement of SQF over DT with an Alpha of 4, and a decrease in packet loss ratio of up to 4.96E-02, or a 27.49% improvement of SQF over SMA with MA equal to 1/8.

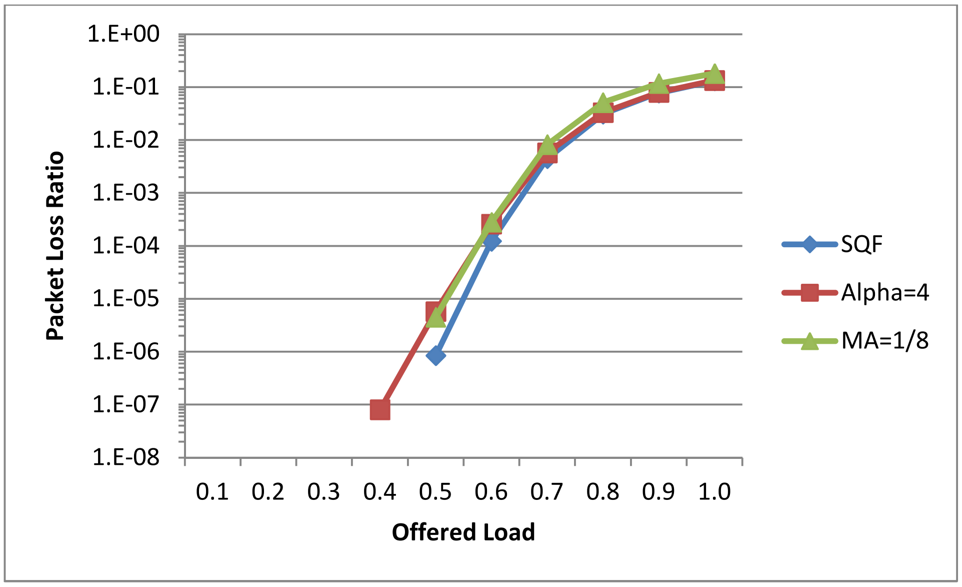


Figure 6.12-SQFvsBest; Ports=16; Buffer=8192; ABL=256

As it can be observed in figure 6.13, for a buffer size of 16384, SQF performs better under all offered loads, presenting a decrease in packet loss ratio of up to 1.61E-03, or a 1.89% improvement of SQF over DT with an Alpha of 4, and a decrease in packet loss ratio of up to 3.94E-02, or a 32.05% improvement of SQF over SMA with MA equal to 1/8.

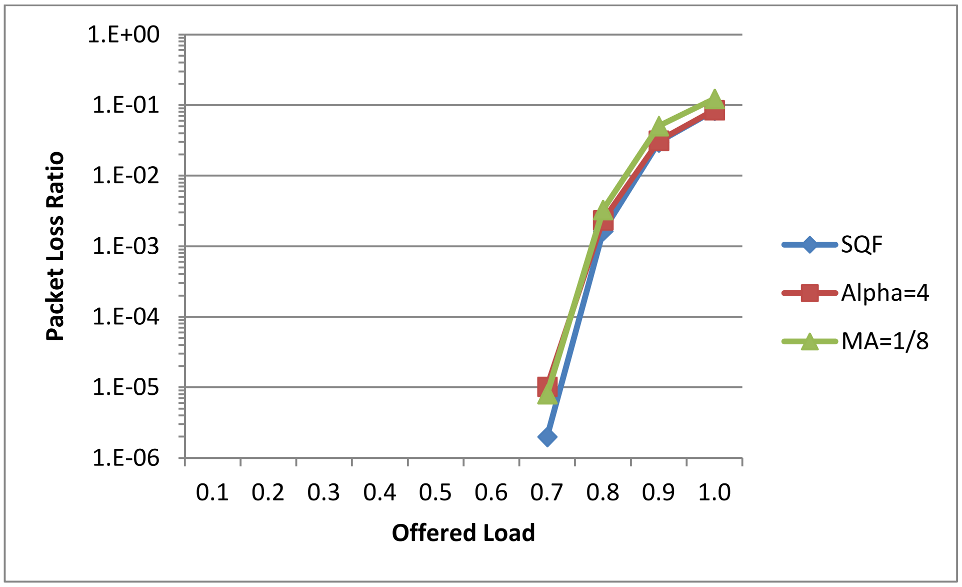


Figure 6.13-SQFvsBest; Ports=16; Buffer=16384; ABL=256

6.4 Summary of Results

When the proposed scheme SQF was compared to the conventional scheme with the best performance in terms of packet loss ratio, it was found that under each and all tested scenarios SQF presented an even lower packet loss ratio, as shown in tables 6.1, 6.2 and 6.3. Proving that achieving the highest possible fairness in the distribution of memory is the key for achieving a higher throughput, and thus minimizing packet loss ratio.

ABL	Scheme	Load%
64	SQF	10-100
128	SQF	10-100
256	SQF	10-100

Table 6.1-Best Scheme for Different ABLs. NxN=64x64; Buffer=4096

Ports	Scheme	Load%
16	SQF	10-100
32	SQF	10-100
64	SQF	10-100

Table 6.2-Best Scheme for Different Number of Ports. ABL=256; Buffer=4096

Buffer	Scheme	Load%
256	SQF	10-100
512	SQF	10-100
1024	SQF	10-100
2048	SQF	10-100
4096	SQF	10-100
8192	SQF	10-100
16384	SQF	10-100

Table 6.3-Best Scheme for Different Buffer Sizes. ABL=256; NxN=16

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

As memory in packet switches remains the bottleneck in terms of the speed and efficiency with which its services are provided, it becomes increasingly important to ensure the more efficient use of available resources in order to mitigate packet loss ratio and avoid the unnecessary retransmission of data that each dropped packet represents, which only adds to the problem by worsening traffic conditions.

Having as primary goal the improvement of fairness with which memory is allocated to incoming packets, in Chapter II we proposed a new sharing memory scheme which we called Shortest Queue First. This scheme prioritizes incoming packets in terms of the lengths of their respective destination output port queues, providing a fair allocation of resources.

In Chapter IV it was found that when the ABL increases so does packet loss; when the number of ports increases so does packet loss; and when buffer size increases packet loss decreases. In this chapter we were also able to determine the optimal configurations for SMA, SMXQ and DT under each of the tested scenarios.

In Chapter V we used the data obtained in Chapter IV to compare the three conventional sharing memory schemes to determine which is the best one. In this chapter we were able to demonstrate that Dynamic Threshold is always better than its static counterpart Sharing with Maximum Queue lengths. However, DT is not always better than SMA, the latter performs better under the lower spectrum of loads, and the range of loads increases with both decreasing number of ports, as well as increasing available buffer.

In Chapter VI we evaluated the performance of SQF under the same scenarios as the conventional schemes, and a comparative performance evaluation was drawn in which it was determined that under each of the tested scenarios SQF presented a lower packet loss ratio than the best of the conventional sharing memory schemes, proving that our initial theory that fairness is key to improving throughput and thus lowering packet loss ratio to be true.

As future work we can suggest the testing and comparison of the conventional schemes as well as the proposed scheme SQF in Multistage Interconnection Networks (MINs), as they are becoming an increasingly common way to build bigger switching fabrics. We can also suggest the implementation as well as testing of the scheme under real world traffic conditions.

REFERENCES

[1] W. Stallings, Data and Computer Communications, Prentice Hall, Tenth Edition, 2013

[2] Cisco Global Cloud Index: Forecast and Methodology, 2012–2017

[3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017

[4] D. Comer, Computer Networks and Internets, Prentice Hall, Fifth Edition, 2008

[5] Ahmad, K.; Begen, A.C., "IPTV and video networks in the 2015 timeframe: The evolution to medianets," Communications Magazine, IEEE , vol.47, no.12, pp.68,74, Dec. 2009

[6] Lihua Wu; Yuan Li; Chen Zhou; Hao Jiang; Xin Qian, "Statistic analysis of data access behavior in the mobile Internet," Communications in China (ICCC), 2013 IEEE/CIC International Conference on , vol., no., pp.89,93, 12-14 Aug. 2013

[7] R. Ramaswami, and K.N. Sivarajan, Optical Networks, A Practical Perspective, Third Edition, M.K., 2009

[8] P. Molinero-Fernandez, Circuit Switching in the Internet, Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, June 2003

[9] Partha P. Mitra, and Jason B. Stark, Nonlinear limits to the information capacity of optical fiber communications, Nature, 411:1027-1030, June 2001

[10] H.J. Chao, C.H. Lam, E.Oki, Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers, John Wiley & Sons, New York, 2001

[11] Youngmi Joo; McKeown, N., "Doubling memory bandwidth for network buffers," INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and

Communications Societies. Proceedings. IEEE , vol.2, no., pp.808,815 vol.2, 29 Mar-2 Apr 1998

[12] Chao, H.J., "Next generation routers," Proceedings of the IEEE , vol.90, no.9, pp.1518,1558, Sep 2002

[13] Karol, M.J.; Hluchyj, M.G.; Morgan, S.P., "Input Versus Output Queueing on a Space-Division Packet Switch," Communications, IEEE Transactions on , vol.35, no.12, pp.1347,1356, Dec 1987

[14] Hluchyj, M.G.; Karol, M.J., "Queueing in high-performance packet switching," Selected Areas in Communications, IEEE Journal on , vol.6, no.9, pp.1587,1597, Dec. 1988

[15] Chen, M.; Georganas, N.D.; Yang, O. W W, "A fast algorithm for multi-channel/port traffic assignment," Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on , vol., no., pp.96,100 vol.1, 1-5 May 1994

[16] Nachiondo, T.; Flich, J.; Duato, J., "Buffer Management Strategies to Reduce HoL Blocking," Parallel and Distributed Systems, IEEE Transactions on , vol.21, no.6, pp.739,753, June 2010

[17] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, “High speed switch scheduling for local area networks,” ACM Transaction on Computer Systems, Nov. 1993, pp. 319-352

[18] McKeown, N.; Mekkittikul, A.; Anantharam, V.; Walrand, J., "Achieving 100% throughput in an input-queued switch," Communications, IEEE Transactions on , vol.47, no.8, pp.1260,1267, Aug 1999

[19] Suzuki, H.; Nagano, H.; Suzuki, T.; Takeuchi, T.; Iwasaki, S., "Output-buffer switch architecture for asynchronous transfer mode," Communications, 1989. ICC '89,

BOSTONICC/89. Conference record. 'World Prosperity Through Communications', IEEE International Conference on , vol., no., pp.99,103 vol.1, 11-14 Jun 1989

[20] Tobagi, F.A., "Fast packet switch architectures for broadband integrated services digital networks," Proceedings of the IEEE , vol.78, no.1, pp.133,167, Jan 1990

[21] Kesidis, G.; McKeown, N., "Output-buffer ATM packet switching for integrated-services communication networks," Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on , vol.3, no., pp.1684,1688 vol.3, 8-12 Jun 1997

[22] Liu Yashe; Liu Zengji, "Performance analysis of ATM switch fabric with combined-input/output buffering," Communication Technology Proceedings, 1996. ICCT'96., 1996 International Conference on , vol., no., pp.508,512 vol.1, 5-7 May 1996

[23] Minkenberg, C.; Engbersen, T., "A combined input and output queued packet switched system based on PRIZMA switch on a chip technology," Communications Magazine, IEEE , vol.38, no.12, pp.70,77, Dec. 2000

[24] Chao, H.J.; Li-Sheng Chen, "Delay-bound guarantee in combined input-output buffered switches," Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE , vol.1, no., pp.515,524 vol.1, 2000

[25] Garcia-Haro, J.; Jajszczyk, A., "ATM shared-memory switching architectures," Network, IEEE , vol.8, no.4, pp.18,26, July-Aug. 1994

[26] Kozaki, T.; Sakurai, Y.; Matsubara, O.; Mizukami, M.; Uchida, M.; Sato, Y.; Asano, K., "32×32 shared buffer type ATM switch VLSIs for B-ISDN," Communications, 1991. ICC '91, Conference Record. IEEE International Conference on , vol., no., pp.711,715 vol.2, 23-26 Jun 1991

[27] Arpaci, M.; Copeland, J.A., "Buffer management for shared-memory ATM switches," Communications Surveys & Tutorials, IEEE , vol.3, no.1, pp.2,10, First Quarter 2000

[28] Kamoun, F.; Kleinrock, L., "Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions," Communications, IEEE Transactions on , vol.28, no.7, pp.992,1003, Jul 1980

[29] Choudhury, A.; Hahne, E.L., "Dynamic queue length thresholds in a shared memory ATM switch," INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE , vol.2, no., pp.679,687 vol.2, 24-28 Mar 1996

[30] Pustisek, M.; Kos, A.; Bester, J., "Buffer management in packet switching networks," EUROCON 2003. Computer as a Tool. The IEEE Region 8 , vol.1, no., pp.276,280 vol.1, 22-24 Sept. 2003

[31] Irland, M., "Buffer Management in a Packet Switch," Communications, IEEE Transactions on , vol.26, no.3, pp.328,337, Mar 1978

[32] Iyer, S.; McKeown, N., "Making parallel packet switches practical," INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.3, no., pp.1680,1687 vol.3, 2001

[33] Iyer, S.; McKeown, N.W., "Analysis of the parallel packet switch architecture," Networking, IEEE/ACM Transactions on , vol.11, no.2, pp.314,324, Apr 2003

[34] Sakurai, Y.; Ido, N.; Gohara, S.; Endo, N., "Large-scale ATM multistage switching network with shared buffer memory switches," Communications Magazine, IEEE , vol.29, no.1, pp.90,96, Jan. 1991

[35] Basak, D.; Choudhury, A.; Hahne, E.L., "Sharing memory in banyan-based ATM switches," Selected Areas in Communications, IEEE Journal on , vol.15, no.5, pp.881,891, Jun 1997

[36] Kumar, S., "The sliding-window packet switch: a new class of packet switch architecture with plural memory modules and decentralized control," Selected Areas in Communications, IEEE Journal on , vol.21, no.4, pp.656,673, May 2003

[37] Yamanaka, H.; Saito, H.; Kondoh, H.; Sasaki, Y.; Yamada, H.; Tsuzuki, M.; Nishio, S.; Notani, H.; Iwabu, Atsushi; Ishiwaki, M.; Kohama, S.; Matsuda, Y.; Oshima, K., "Scalable shared-buffering ATM switch with a versatile searchable queue," Selected Areas in Communications, IEEE Journal on , vol.15, no.5, pp.773,784, Jun 1997

[38] JongIck Lee; JongMoo Sohn; MoonKey Lee, "Design of a shared multi-buffer ATM switch with enhanced throughput in multicast environments," ATM Workshop, 1999. IEEE Proceedings , vol., no., pp.455,461, 1999

[39] Kumar, S.; Agrawal, D.P., "THE SLIDING WINDOW APPROACH TO HIGH PERFORMANCE ATM SWITCHING FOR BROADBAND NETWORKS," Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE , vol.1, no., pp.315,, 14-16 Nov 1995

[40] Ahmad, K.; Begen, A.C., "IPTV and video networks in the 2015 timeframe: The evolution to medianets," Communications Magazine, IEEE , vol.47, no.12, pp.68,74, Dec. 2009

[41] Jiang, X., "Progress and challenges for high speed optical access networks," Wireless and Optical Communications Conference (WOCC), 2011 20th Annual , vol., no., pp.1,3, 15-16 April 2011

[42] Agrawal, B.; Sherwood, T., "High-Bandwidth Network Memory System Through Virtual Pipelines," Networking, IEEE/ACM Transactions on , vol.17, no.4, pp.1029,1041, Aug. 2009

[43] Hluchyj, M.G.; Karol, M.J., "Queueing in high-performance packet switching," Selected Areas in Communications, IEEE Journal on , vol.6, no.9, pp.1587,1597, Dec. 1988

[44] Hahne, E.L.; Choudhury, A., "Dynamic queue length thresholds for multiple loss priorities," Networking, IEEE/ACM Transactions on , vol.10, no.3, pp.368,380, Jun 2002

[45] Naoaki Yamanaka, High-Performance Backbone Network Technology, CRC Press, 2004

[46] Naoaki Yamanaka, GMPLS Technologies: Broadband Backbone Networks and Systems, CRC Press, 2010

[47] N. McKeown, “Scheduling Algorithms for Input-Queued Cell Switches,” PhD thesis, Univ. of California at Berkeley, 1995.

[48] Dally, W.J., "Virtual-channel flow control," Parallel and Distributed Systems, IEEE Transactions on , vol.3, no.2, pp.194,205, Mar 1992

[49] Duato, J.; Johnson, I; Flich, J.; Naven, F.; Garcia, P.; Nachiondo, T., "A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks," High-Performance Computer Architecture, 2005. HPCA-11. 11th International Symposium on , vol., no., pp.108,119, 12-16 Feb. 2005

[50] P.J. Garca, J. Flich, J. Duao, I. Johnson, F.J. Quiles, and F. Naven, “Efficient, Scalable Congestion Management for Inter- connection Networks,” IEEE Micro, vol. 26, no. 5, pp. 52-66, Sept./Oct. 2006.

[51] Duato, J.; Flich, J.; Nachiondo, T., "A cost-effective technique to reduce HOL blocking in single-stage and multistage switch fabrics," Parallel, Distributed and Network-Based

Processing, 2004. Proceedings. 12th Euromicro Conference on , vol., no., pp.48,53, 11-13 Feb. 2004

[52] Martinez, A; Garcia, P.J.; Alfaro, F.J.; Sanchez, J.L.; Flich, J.; Quiles, F.J.; Duato, J., "A Switch Architecture Guaranteeing QoS Provision and HOL Blocking Elimination," Parallel and Distributed Systems, IEEE Transactions on , vol.20, no.1, pp.13,24, Jan. 2009

[53] “g++(1) - Linux man page”. http://linux.die.net/man/1/g++. Last Access: 7/29/2014

[54] http://gridscheduler.sourceforge.net/htmlman/htmlman1/qsub.html. Last Access: 7/29/14

[55] Collier, B.R.; Kim, H.S., "Efficient analysis of shared buffer management strategies in ATM networks under non-uniform bursty traffic," INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE , vol.2, no., pp.671,678 vol.2, 24-28 Mar 1996

[56] Bhagwat, S.; Tipper, D.; Balakrishnan, K.; Mahapatra, A, "Comparative evaluation of output buffer management schemes in ATM networks," Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on , vol., no., pp.1174,1178 vol.2, 1-5 May 1994

BIOGRAPHICAL SKETCH

Javier Castillo was born on August 11, 1987. He finished his undergraduate studies at The University of Texas-Pan American in May, 2010. He obtained the degree of Bachelor of Science in Electrical Engineering. He finished his Masters of Science in Electrical Engineering also at The University of Texas-Pan American on August, 2014. His current mailing address is,

1117 W. Wisconsin Rd.

Edinburg TX 78539