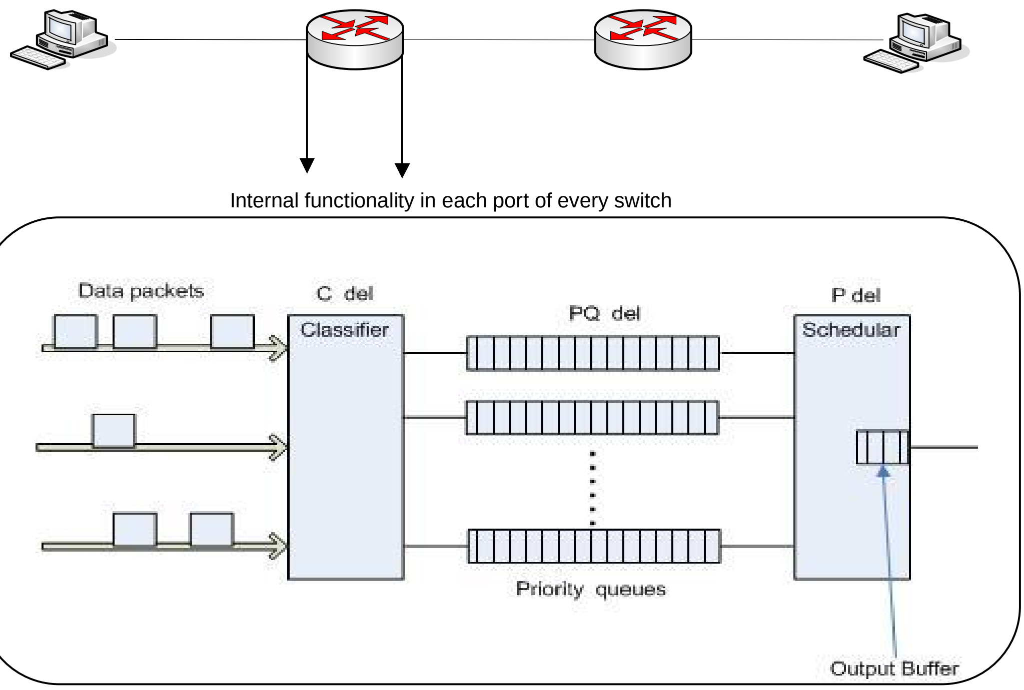
Technical Report, IDE1044, June 2010

Switched Multi-hop Priority Queued Networks Influence of priority levels on Soft Real-time Performance

Master's Programme in Computer Systems Engineering

Submitted by:

Iftikhar Ahmed and Muhammad Farooq



School of Information Science, Computer and Electrical Engineering
Halmstad University, Sweden

Switched Multi-hop Priority Queued Networks

Influence of priority levels on Soft Real-time Performance

Master Thesis in Computer Systems Engineering

School of Information Science, Computer and Electrical Engineering

Halmstad University

Box 823, S-301 18 Halmstad, Sweden

June 2010

Description of cover page picture/figure:

Model diagram of packet traversing at each link in the switch.

Preface

This thesis project is submitted to fulfil the requirement of the master degree program in Computer System Engineering at Halmstad University Sweden in IDE (School of Information Science, Computer and Electrical Engineering) department.

First of all, we are extremely thankful to our supervisor, Mattias Wecksten for his guidance and feedback. We are also thankful to all the friends who morally helped us.

Special thanks to our parents for their kind cooperation and encouragement at every stage during the whole study program and we want to express extra gratitude to all the teachers with whom we have contacted in any way during this study of master program.

Muhammad Farooq

Iftikhar Ahmed

Halmstad University, June 2010

Influence of priority levels on Soft Real-time Performance

Abstract

In the last few years, the number of real-time applications has increased. These applications are sensitive and require the methods to utilize existing network capacity efficiently to meet performance requirements and achieve the maximum throughput to overcome delay, jitter and packet loss. In such cases, when the network needs to support highly interactive traffic like packet-switched voice, the network congestion is an issue that can lead to various problems. If the level of congestion is high enough, the users may not be able to complete their calls and have existing calls dropped or may experience a variety of delays that make it difficult to participate smooth conversation.

In this paper, we investigate the effect of priority levels on soft real-time performance. We use the priority queues to help us manage the congestion, handle the interactive traffic and improve the over all performance of the system. We consider switched multi-hop network with priority queues. All the switches and end-nodes control the real-time traffic with "Earlier Deadline First" scheduling. The performance of the network is characterized in terms of the average delay, the deadline missing ratio and the throughput.

We will analyze these parameters with both the bursty traffic and evenly distributed traffic. We will analyze different priority levels and will see how the increase in priority level increases the performance of the soft real-time system.

Index Terms: Soft real-time, multi-hop, packet switched, priority queues, bursty and evenly distributed traffic.

Table of Contents

1 Introduction

Real-time systems demand the correct results in the given time span. Real-time systems should be efficient in the sense of guaranteed quality of service. For guaranteed quality of service (QoS) in packet switched networks, a number of service disciplines are available [4] [5] [6]. There are several disciplines that are based on the prioritization of the packets. In these techniques the priority value is assigned to packets and their transmission becomes possible on the basis of the values of the priority. The “Priority Queues” are managed for the implementation of the priority based technique and real-time sorting.

There are two main categories for real-time systems:

1) Hard real-time systems

2) Soft real-time systems.

In hard real-time systems the deadline is guaranteed and the task must be completed before deadline. If the system could not perform its task in the given time and it misses the deadline, then catastrophe can occur [8].

On the other hand, the soft real-time system is less restrictive. In this system if the task misses the deadlines then only performance of the system is affected and no catastrophe occurs.

In this project, with the help of simulation results, we have demonstrated the influence of priority queues in switched multi-hop networks with respect to real-time performance. All the traffic in the proposed network is periodic. Every packet has its own priority number, the priority to the packets is assigned on the bases of earlier deadline first (EDF).

In simulation the comparisons are shown for both types of traffics (i.e. bursty traffic and evenly distributed traffic). In bursty traffic all the traffic flow starts at the same time. In evenly distributed traffic the starting time of the traffic flow is different. Bursty traffic and evenly distributed traffic are differentiated by adding some offset values.

Consequently the average delay, deadline ratio and throughput are determined for both types of traffic flows. The results of different priority levels have been compared to get the conclusion. The priority queues have been implemented at each port in all the switches in the network. Finally, we have provided the result of the research work about the influence of different priority levels on the performance of soft real-time systems.

1.1 Motivation

The packet-switched integrated-service networks need to support large types of real-time applications with different quality of service (QoS) requirements. These real-time applications have strict performance requirements in terms of end-to-end delay, average throughput and packet-loss rate. These requirements must be satisfied, to achieve really high speed, without compromising on network utilization.

To achieve the QoS guarantees in packet-switched networks, many different service disciplines have been presented. Many of these service disciplines base on prioritization.

In priority-based service policy, packets are assigned priority values and then transmitted according to highest-priority-first order. So priority queues can be used to maintain a real-time sorting of the queue elements in a decreasing order of priorities. Thus, the task of highest-priority can be serviced first.

Due to this, we are going to use the priority queues at each port at each switch to handle the traffic in such a way that it fulfils the QoS requirements.

The priority queues enable to provide better service to certain flows, and ensure the efficient use of network resources. When the network is facing a heavy load of traffic, then the priority queues handle the data in a very efficient way and, as a result, the performance of soft real-time system gets better, because priority queues help to reduce the deadline miss ratio, average delay and packet loss and increase the throughput.

1.2 Related work

The objective of the given thesis work is to make a scheduling algorithm by using EDF approximation that manages the priority queues of different priority levels for switched multi-hop network. There are many techniques to prioritize the queues. The priority queues technique can be managed on the basis of:

1) Shortest Message Preempt Packet Scheduling (SMP-PS) [1].
2) Shortest Remaining Transmission Time (SRTT-PS) [1].

In SMP-PS, the priority queues are managed on the basis of message length [1]. The packets belonging to the messages of shorter length are processed first. It interrupts the transmission of the packets of the messages which are longer in length. The longer message's packets again continue their transmission when shorter one finishes. In this technique, it is necessary to know the length of the packet before implementation. All the priority queues can be managed on the basis of the priority of the packets, and priorities are the inverse of the message length of the packet to which these packets belong [1]. In comparison to SMP-PS and FCFS, the SMP-PS is efficient than FCFS [1]. However, the drawback of the SMP-PS priority technique is that, when a short message interrupts the longer one, the longer one stops its transmission, even if it may be near to completion [1].

The packets belonging to the message whose transmission time is shorter have higher priority over those packets which belong to the messages having longer transmission time. Both SMP-PS and SRTT-PS use almost the same strategy. SMP-PS works on the basis of message length and SRTT-PS works on the basis of transmission time. Shortest SRTT-PS is more complicated than SMP-PS because in this scheduling technique it is necessary for each node to have the complete information of every message state which is in buffer [1]. Therefore for SRTT-PS, each node needs the information of the message length and message state. Every packet needs the tag of message state and length, which makes it possible for the nodes to identify the message state. SRTT-PS scheduling is little bit more complex than SMP-PS but it is more efficient than SMP-PS [1].

Similarly, according to the literature so far, many hardware based priority queues have been introduced, like calendar queues, binary tree of comparator based priority queues and systolic array based priority queues [4] [3]. All of these techniques have some limitations. Such as the calendar queues technique can maintain only a small and fixed set of prioritized data and the large priority set requires intensive hardware support. Similarly, the priority queues based on binary tree comparison and systolic array are normally harder to scale because the hardware depends on the worst case queue size binary tree of comparator based priority queues and systolic array based priority queues [4].

In this project, the EDF approximation technique has been used to manage the priority queues [2]. There would be multiple priority queues, all depending upon the nature of the data. Priority queues are managed on the basis of the deadline of the packet, so there would be little experience of starvation. The management of priority queues by using EDF approximation is reliable for real-time systems because it will keep track of the time period and deadline of every packet. We have implemented the soft real-time systems because the hard real-time system is beyond the scope of this thesis project.

1.3 Problem Definition

In a multi-hop packet switched network, we need to evaluate the effect of priority levels upon the performance of soft real-time systems for both kinds of “bursty” and “evenly distributed” traffics.

Soft real-time system is a system in which the overall system remains in working state, but the performance is affected, when the tasks are not completed on time. Like in video conference, if voice data packets are not synchronized with video data packets, then it will irritate the clients, but the system will remain in a working state i.e. it will never crash. The video can be seen and voice can be heard although they are not synchronized. In this proposed project there is a need to implement the priority queues to check the effect of these priority levels on the performance of soft real-time systems.

1.4 Thesis organization

The remaining parts of the thesis are organized as follows:

Chapter 2 contains discussion about the Network Model assumptions and specifications.

Chapter 3 is about the model of the network.

Chapter 4 carries the simulation approach and experiments.

Chapter 5 gives the Conclusion.

2 Network model

For the logical structure of the network model, binary tree topology has been implemented. Figure 1 shows the topology of proposed network model, in which all the ending nodes can send and receive data over full duplex links.

2.1 Assumptions and Specifications

We assume a multi-hop network with end nodes and switches that are connected in a packet switched network. There are many network topologies and configurations which have their own pros and cons but, we are using tree topology having the computation nodes in the leaves and switch nodes in the rest of the tree.

In our case the switches are output buffered and support store and forward message passing technique. As we are using tree topology, so every switch has three ports.

The classification delay, processing delay and propagation delay are fixed. The priority queues are employed at each port in all the switches without any change in the network protocols. The switches are output buffered and handle the flows on per port basis. All the switches and the end nodes have software (Real-time layer) added which shapes the traffic on the real-time channel [15].

A real-time channel with index i is characterized by:

$\{P_i, C_i, d_i\}$,

Where P_i is the period of data, C_i is the amount of data per period, and d_i is the relative deadline [15]. Each flow has a random number of packets from the range 1 to 4 from uniform distribution.

The comparison has been made for the following parameters:

i. Average delay

ii. Deadline miss ratio

iii. Throughput

2.2 Multi-hop network topology

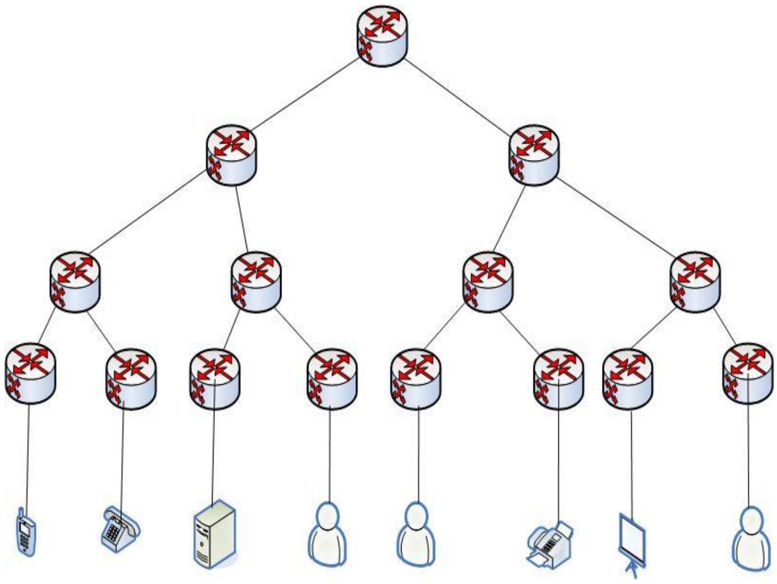


Figure 1: Multi-hop network

It is the general diagram of proposed Multi-hop network, where multiple hops are connected to each other using tree topology. Figure 1 shows the binary tree topology that is being used in the proposed network model. This is a generic model, generic in the sense that we can add and remove the desired number of hops in/from the network. The experiments have been made by making fix number of hops that facilitate a comparison of the results during experiments. For this purpose four levels binary tree topology has been used (15 switches). Figure 2 depicts the internal functionality of the switch. Each switch performs its action according to this methodology.

2.3 Methodology

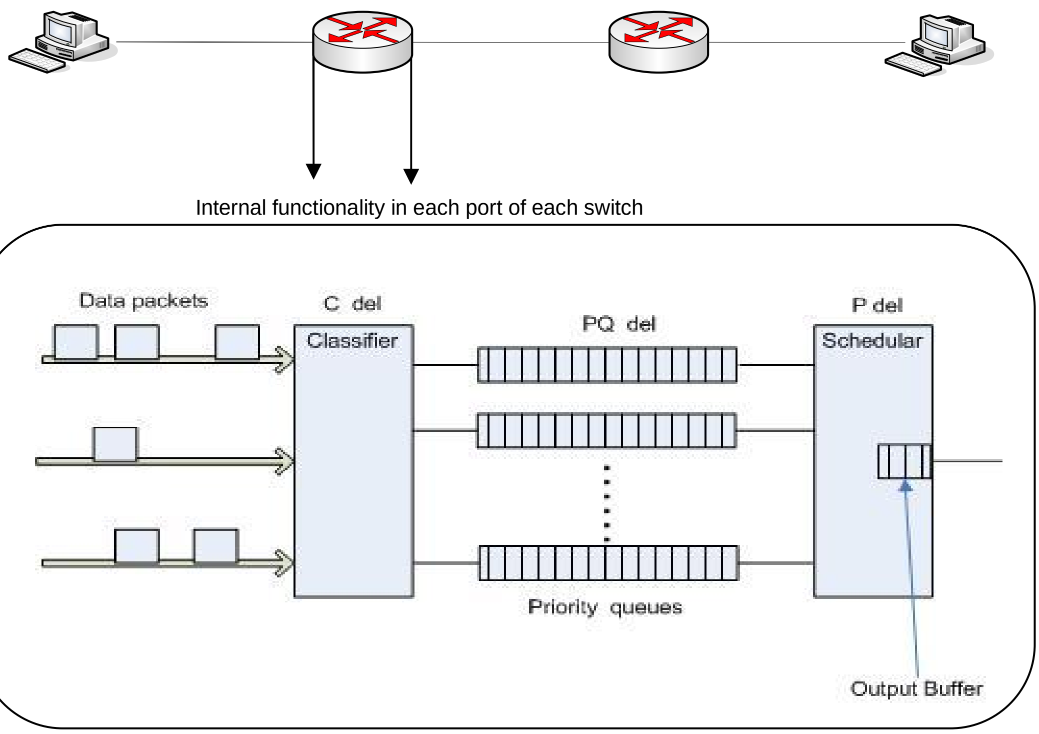


Figure 2: Packet traversing model at each port in each switch

Figure 2 shows the packet traversing using priority queues of each port of every switch in multi-hop network. In the proposed tree topology, each hop has at least three links through which it can transmit and receive data. These processing components, like the set of priority queues and scheduler, will provide the service at each link. So the above processing unit works at each port.

A packet has to pass through different stages before travelling to the next hop in a switched multi-hop network, where a packet is scheduled through priority queues. The packet passes through the stages of classifier, priority queue, and scheduler. Then it is transmitted to next hop.

Classifier classifies the data packets on the basis of packet header information, i.e. deadline and nature of the data etc. Then these packets are forwarded concerned queue. There are three queues in the proposed model. The top queue has high priority containing high priority data like voice packets. The middle queue has medium priority data, such as instant messages data packets. The lower queue has low priority and contains packets of low priority like e-mail data packets.

After passing packets to the concerned queues, the scheduler gets the data from all these queues on the basis of their priorities and sends to the link. If the link is busy, the packet waits in the output buffer from where it is transmitted to the next hop.

When a packet moves from one hop to the concerned hop, it suffers different sorts of delays like classifying delay, priority queuing delay, processing delay, transmission delay and propagation delay.

2.4 Classifier

From the implementation point of view, a classifier is a small function which is used for classification. It fills the queues according to the given pattern of the priority. This classifier receives the packets from the sources and accesses their header information and then on the base of their deadlines it classifies the packet priority queues. The packet which has a shorter deadline has the high priority [7].

There are many techniques to maintain the priority queues. Mostly the priority queues are maintained on the basis of message length, shortest remaining transmission time [1] and on the hardware level. There are also some other techniques to maintain the priority queues such as calendar queue, binary tree of comparator based priority queues and systolic array based priority queues [4] [3]. However, in the proposed project, the recommended one is the EDF approximation. That is why it has been used instead of other techniques. Classifier always remains active and classifies the packet as it receives. Hence no packet remains in a waiting state in the classifier.

2.5 Priority queue

In the priority queues technique, the highest priority packets are serviced first. When there is no packet available in high priority then low-level priority traffic would be served [10]. In this case, the project classifier classifies the data packets and fills the queues according to their priority. Each packet goes into its assigned queue. In our case, there are three priority level queues i.e. high, medium and low level priority queue.

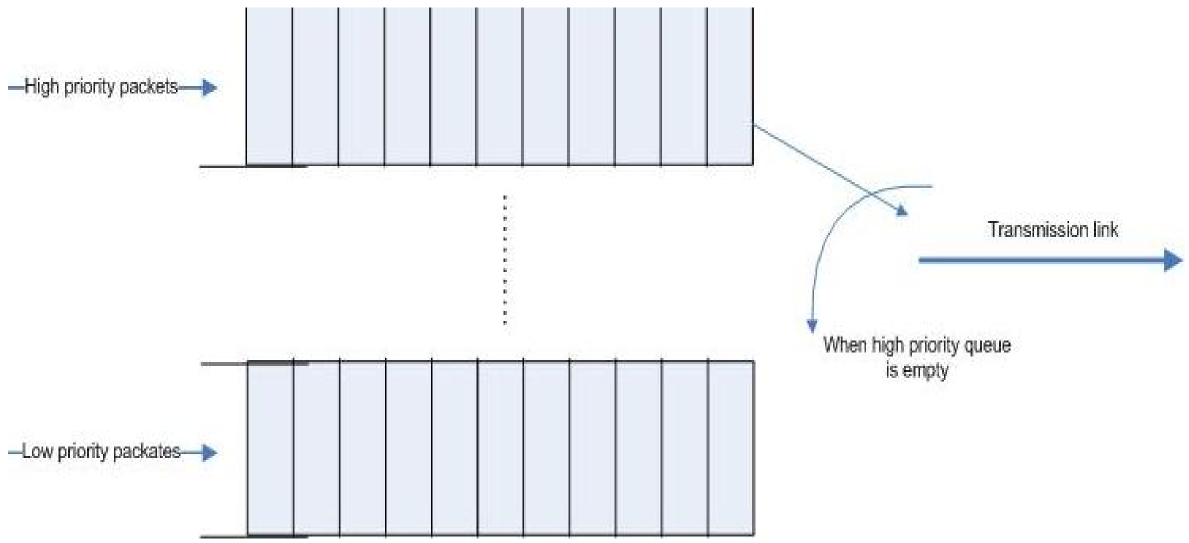


Figure 3: Priority queues

Packets belonging to the high priority go to the level one queue and packets of medium priority go to second level queue and low priority packets would be in third level queue. At this stage, the purpose of the queues is to hold the prioritized data packets to provide the services to scheduler.

Figure 3 depicts the priority wise access method of the priority queues. In this figure, there are number of priority queues which are managed according to the priority of the packets.

High priority packets (for example VoIP) will go in the level one queue, data having low priority (for example data from UDP applications) will go in level two queue and data from TCP applications will be managed in the third level queue. Then the scheduler will access the queues on the basis of their priority. The high priority queue will be served first and when it becomes empty the scheduler will move to the low priority queue and so on.

2.6 Scheduler

The scheduler provides the services to access the priority queues. Scheduler fetches the data packets from the priority queues and sends those packets through the transmission medium to the next hop. In a case when the link is busy, as in our case, only one packet can travel through the link in a single unit of time. So the packet will wait in the buffer.

For fetching the packets from the queues there are many techniques that a scheduler can use like round robin (RR), weighted round robin (WRR) and first come first serve (FCFS).

The RR accesses all the available queues in turns from 1 to n then again starts from 1, and so on. It gives the opportunity to all the queues [11]. In WRR, each queue has some wattage over others [11]. For example, if there are three queues, then four packets from first level and two from second level and one from first level.

We are using FCFS scheduling technique in our implementation. In FCFS, the packet is served first which comes first [2]. The scheduler accesses the packets from the priority queues by using FCFS. We can say that, over all scheduling method is priority queues, and every individual queue is accessed using FCFS.

2.7 Communication Mechanism

As the real-time communication needs strict delay requirements and guaranteed performance, the real-time channels are used to guarantee the performance [13]. So, in the proposed model, the real-time channels have been used for data transmission.

2.7.1 Real-time Channel Establishment

A real-time channel, also called the “logical” channel, is a communication technique which is used for timely delivery of messages in the real-time communication [14]. It is a path between the source and destination having parameters which represent the performance requirements of the client [13]. It is a process of request and acknowledgment in which the source node, destination node and switches take part, and agree to establish a communication channel. Both the end nodes and switches have a software layer also called real-time channel layer, which shapes the traffic on the real-time channel.

The real-time channel is established, when the application task requests the network by sending a “request-frame” to the switch; in its request, the task states all the required constraints. The switch performs the feasibility test, and determines whether the establishment is feasible or not. If it is feasible, the “request-frame” is forwarded to the next hop. In reply the destination node sends back a “response-frame”, which ultimately reaches to the source node. If the request is not feasible the “request-frame” is not sent to the next hop. Instead, “response-frame” is sent to source with the information of rejection [15].

2.7.2 Real-time Traffic Handling

The real-time layer changes the IP header and sets up the outgoing real-time IP datagram [15]. The absolute deadline of the frame is set by the IP source address and 16 most significant bits of IP destination address. The 16 least significant bits become the real-time channel ID. The type of service field is set to value 255. To deliver the frame to final destination, the switch exchanges the source and destination IP addresses and MAC addresses with correct ones (which were stored in the switch at the time of channel establishment). The error checking techniques are applied before putting frames into output queue [15].

3 Simulation approach

For the simulation, binary tree topology has been used; the tree has four levels, containing fifteen switches, so the network is considered as a multi-hop network. Each real-time channel has the inter-arrival time of messages that it carries; each real-time channel can carry a message of maximum length of 4 packets, i.e. each message has random number of packets within the range of 1 to 4 from a uniform distribution. Every real-time channel has a deadline according to which, it is served [9], the shortest the deadline highest the priority [7].

The maximum delay to send a packet at 100 Mbps link with a packet size 1518 bytes, is ≈ 121 μs, but we assumed little bit more, i.e. 125μs, as assumed in [12]. Packets are classified into their proper priority queues through the classifier. Then, the scheduler fetches the packets from the queues on the basis of the FCFS scheduling mechanism [2], and sends to the next hop if the link is free. If the link is not free then the packet has to wait in output buffer until the link is free.

From the real-time point of view, the arrival time of a packet should be less than or equal to the assigned deadline.

For a hard real-time system, it is very important to meet the deadline otherwise catastrophe will occur, but for a soft real-time system, if packet delay increases than its deadline, then definitely performance is affected but the overall system remains in working condition and no catastrophe occurs. The delay that any packet can suffer is the difference of starting time (when a packet enters a network) and delivery time (when a packet reaches its destination).

3.1 Delay comparison between bursty and evenly distributed traffic

When all the traffic starts from same time zero it is called bursty traffic. It means that all the traffic enters in the network at the same time. The evenly distributed traffic is differentiated on the basis of offset. In the following example, we calculate the delay of both types of traffic and see that the total delay of evenly distributed traffic is less than the bursty traffic. This is logical, because when all the traffic enters in the network at the same time then the network would definitely be congested and this congestion will cause the extra delay. We will see the clear difference in the following delay calculations.

Figure 4 shows a very simple binary tree network that has been considered for the experiment. In this diagram, there are three switches and four nodes, these nodes can send and receive the data. In this experiment, node N1 and N2 are source nodes and N4 is destination node.

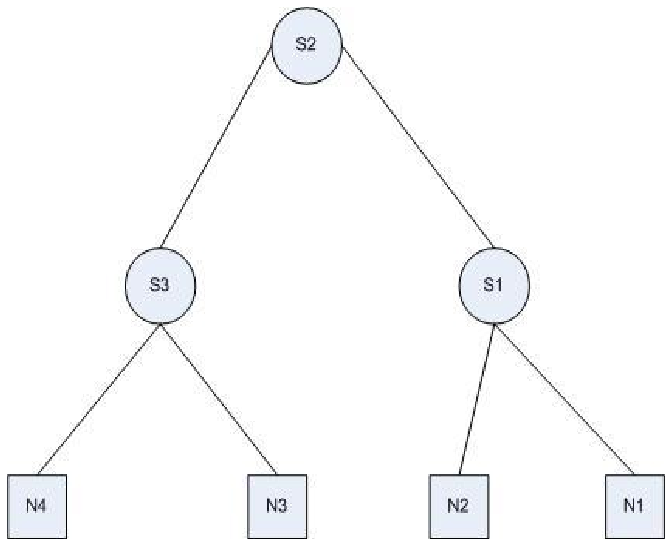


Figure 4: N1 & N2 are source node and N4 are destination nodes.

3.2 Evenly distributed traffic

In this section the delay calculation is discussed for evenly distributed traffic in the network. In this experiment the M1, M2, M3 are three messages and the N1, N2 are source nodes while N4 is the destination node; N1 is generating one message M1 and N2 is generating two messages M2 and M3. The message M1 has offset 25µs, length 10 bytes, and deadline is 250µs. The message M2 has offset 0µs, length 25bytes, deadline 175µs. The message M3 has offset 15µs, length 25bytes and deadline 225µs. Data rate is supposed to be 1 Mbps. So according to the given data rate and length of the messages, the transmission delay of message M1 is 10 µs and both the messages M2 and M3 have 25µs.

At the time zero microsecond, the packet of message M2 arrives with 25µs transmission delay towards switch S1. At time 15µs, the message M3 tried to arrive, but the link was busy so it suffered a 10µs queuing delay because of the message M2. So, the message M3 will reach at the switch S1 at time 50µs, meanwhile the message M2 would be transmitted and it will reach at the switch S2 at time 50µs, and the message M1 will reach at time 35µs at the switch S1, before the message M3.

At the time 50µs, the message M2 will be transmitted to the switch S3 and it will reach at time 75µs. The message M1 will be transmitted now and reaches at the switch S2 at time 60µs and it will suffer 15µs queuing delay because, at that time, the link was busy because of the message M2. After that, the message M3 will be transmitted and reaches at the switch S2 at time 85µs by facing 10µs queuing delay because of the message M1, which kept the link busy.

Meanwhile the message M2 will be transmitted to destination node at time 100µs; that will never face any queuing delay because it is one link ahead from the beginning. When it moves to next link, then other packets come into the previous link. So, when the message M2 transmitted from the switch S3, at the same time the message M1 transmits from the switch S2 to switch S3 and will face 15µs delay because of message M2. Then message M3 will be transmitted; it will reach at the switch S3 by facing 10µs queuing delay because of the message M2, and the message M3 will reach at destination node by facing zero microsecond delay from the switch S3 to the destination node N4. Finally to calculate the delay, we need to subtract the starting time from the total time that a packet has faced to reach the destination. The following table shows the delay at each switch.

The following abbreviations have been used:

DS1 = delay at switch (S1)

DS2 = delay at switch (S2)

DS3 = delay at switch (S3)

DN1 = delay at source node (N1)

DN2 = delay at source node (N2)

DN4 = delay at destination node (N4)

TD = total delay

The delay at each switch that a packet faced is as follows:

Messages	**Offset**	**DS1 (**µs)	**DS2 (**µs)	**DS3 (**µs)	**DN4 (**µs)	**TD (**µs)
M1	25	35	60	85	110	110-25=85
M2	0	25	50	75	100	100-0=100
M3	15	50	85	110	135	135-15=120

Blocking time that each packet faced at each hop:

Messages	**Offset**	**N1-N2** (µs)	**DS1** (µs)	**DS2** (µs)	**DS3** (µs)
M1	25	N1=0	15	15	15
M2	0	N2=0	0	0	0
M3	15	N2=10	10	0	0

3.3 Bursty traffic

For the case of bursty traffic, the same traffic parameters are used which were used for the evenly distributed traffic experiment, but here all the traffic is starting from time zero. So, M1, M2, M3 are three messages, that will traverse through the switches S1, S2 and S3, from source nodes N1 and N2 towards the nodes N4 which is the destination node.

The node N2 is generating two messages M2 and M3. The message M1 has deadline 250µs, M2 has 175µs and M3 has 225µs. The transmission delay is supposed to be 25µs.

So, according to the deadline, the message M2 is transmitted first and it will reach at the switch S1 at the time 25µs, then the message M2 wants to be transmitted but the link is busy so it will face 25µs delay and it will reach at switch S1 in 50µs, because the link will be free after 25µs and 25µs is the transmission delay, so that is why it will reach at the switch S1 in

50µs. The message M1 is from a separate node, so it will reach at the switch S1 in 25µs. Now the message M2 will be transmitted and it will reach at the switch S2 in 50µs, at this time, the message M1 was in the queue of the switch S1, so the message M1 wants to be transmitted but the link is busy so it will face delay of 25µs and it will reach at the switch S2 in 75µs. Meanwhile the message M3 also wanted to be transmitted but the link was busy so it had to face delay of 25µs and it would reach at the switch S2 in 100µs.

Now the message M1 will be transmitted and it will reach at the switch S3 in 100µs, the message M3 will reach at the switch S3 in 125µs.

Now, again the message M2 will be moved first and it will reach at the destination node in 100µs, the message M1 in 125µs and the message M3 in 50µs. It is clearer in the table.

The delay at each switch that a packet faced is as follows:

Messages	**Offset**	**DS1(**µs)	**DS2 (**µs)	**DS3 (**µs)	**DN4 (µs)**	**TD(µs)**
M1	0	25	75	100	125	125-0=125
M2	0	25	50	75	100	100-0=100
M3	0	50	100	125	150	150-0=150

Blocking time (queuing delay) that each packet faced at each hop:

Messages	**Start time**	**N1-N2 (**µs)	**DS1(µs)**	**DS2 (**µs)	**DS3 (**µs)
M1	0	N1=0	25	0	0
M2	0	N2=0	0	0	0
M3	0	N2=25	25	0	0

From this experiment, it is clear that the non-busty traffic has lower delay than busty traffic. In the case of bursty traffic, the total delay is 125+100+150=375µs, while for the evenly distributed traffic, the delay is 85+100+120= 305µs.

3.4 Simulation experiments

The simulation analysis presented here shows the performance difference among usage of different priority levels with the EDF scheduling. The network has been simulated with 100 Mbit/s full-duplex Ethernet switches, source nodes, destination nodes, and sets of real-time channels. Each real-time channel has random deadline from the range 1200µs to 2500µs and has period of 2000µs. Each real-time channel has a random number of packets from the range

1 to 4 from the uniform distribution. Maximum number of priority levels is 3 and the priority has been assigned on the base of deadline.

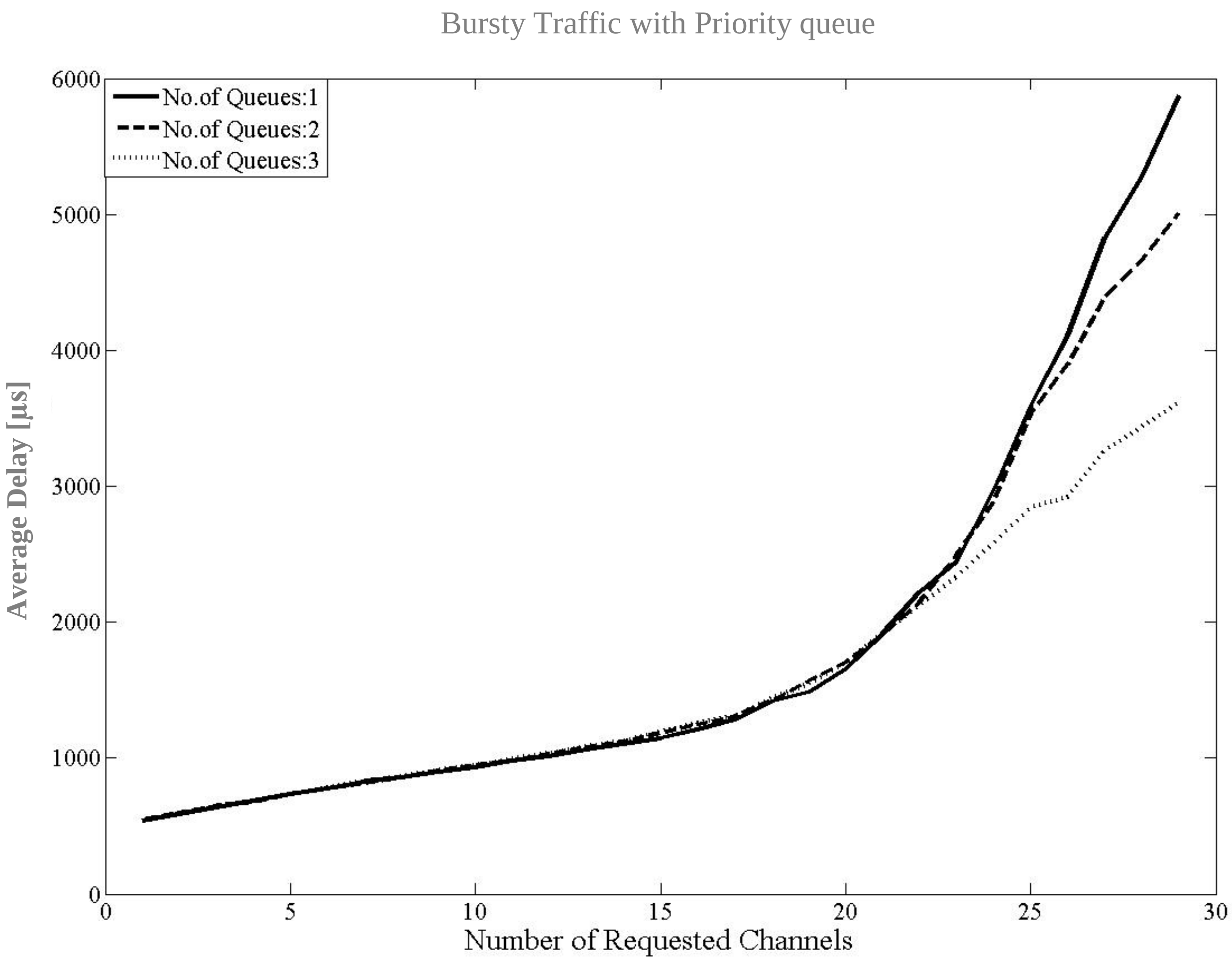


Figure 5: Average delay for bursty traffic

Figure 5 shows the average delay comparison among different priority levels having bursty traffic in the network.

The graph shows three scenarios. The first one is when there is only one queue which is depicted by solid line. The second one when there are two priority levels is depicted by dashed line and finally third scenario when there are thee priority levels is depicted by dotted line.

Up to 22 to 23 real-time channels, the average delay is almost same for all scenarios but, after that the network starts becoming congested and the effect of priority levels gets clear.

The average delay represented by the solid line is above the others. Its last point is near 6000 µs, which is higher than the other two. The dashed line is between the solid and dotted lines. The last point of this line is nearly 5000µs.

The dotted line is below the other two lines. It shows that the average delay decreases when there are three priority levels and its last point is near to 3500µs.

We can see how the average delay decreases with the increase of priority level.

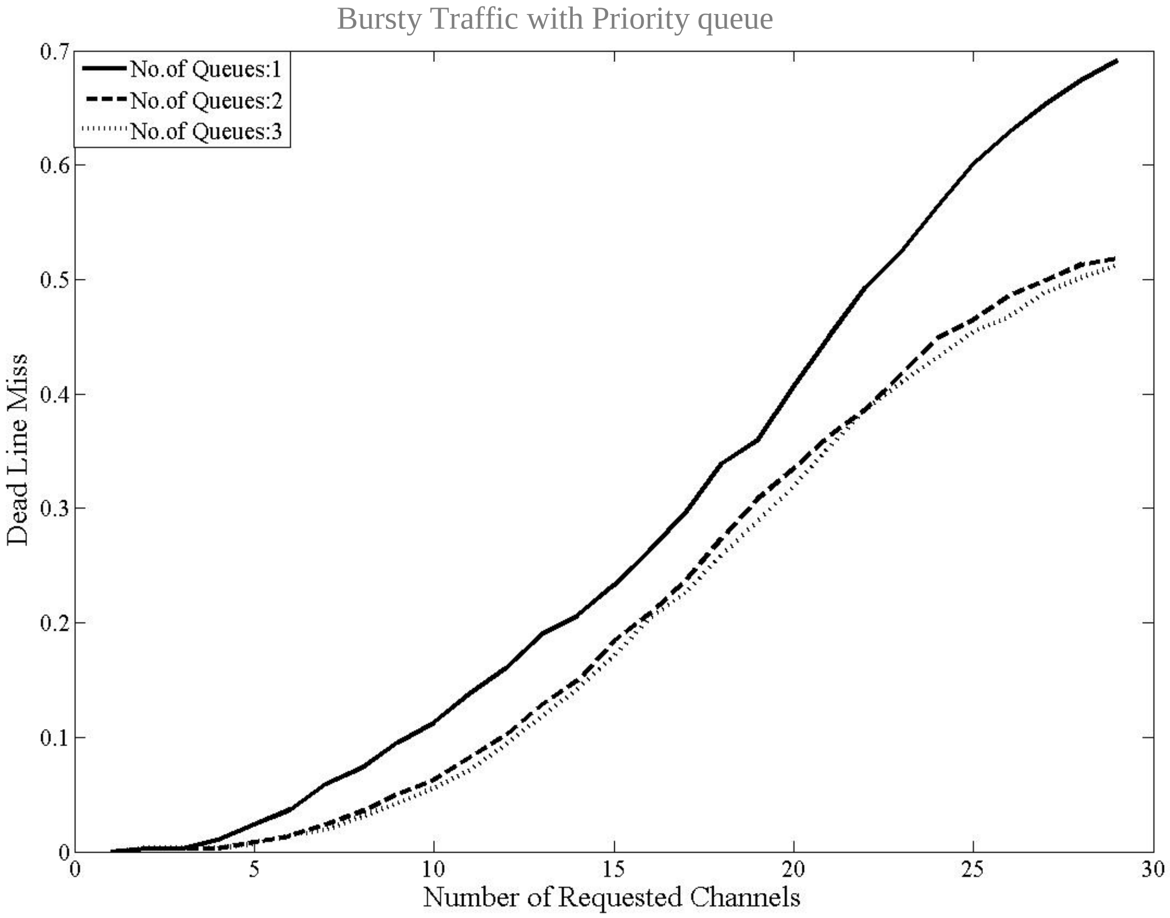


Figure 6: Dead line miss ratio for bursty traffic

Figure 6 shows the deadline miss ratio comparison, when there is bursty traffic in the network. This graph also shows the effect of priority levels very clearly. We can see that as we increase the priority level the deadline miss ratio decreases.

The solid line is higher than the other two, showing that the deadline miss ratio will be high for the case of one queue. All the traffic goes into one queue, so if the packets of low priority are already in the queue and there come high priority packets, then the high priority packets will suffer delay; they will have to wait until the delivery of the low priority packets. Ultimately, performance will be decreased.

The deadline miss ratio for the case of two priority queues is less than the one queue but slightly more than the three priority queues. In the case of three priority queues, the deadline miss ratio even reduces.

Figure 7 below, shows the average delay comparison among different priority levels but, having the evenly distributed traffic in network.

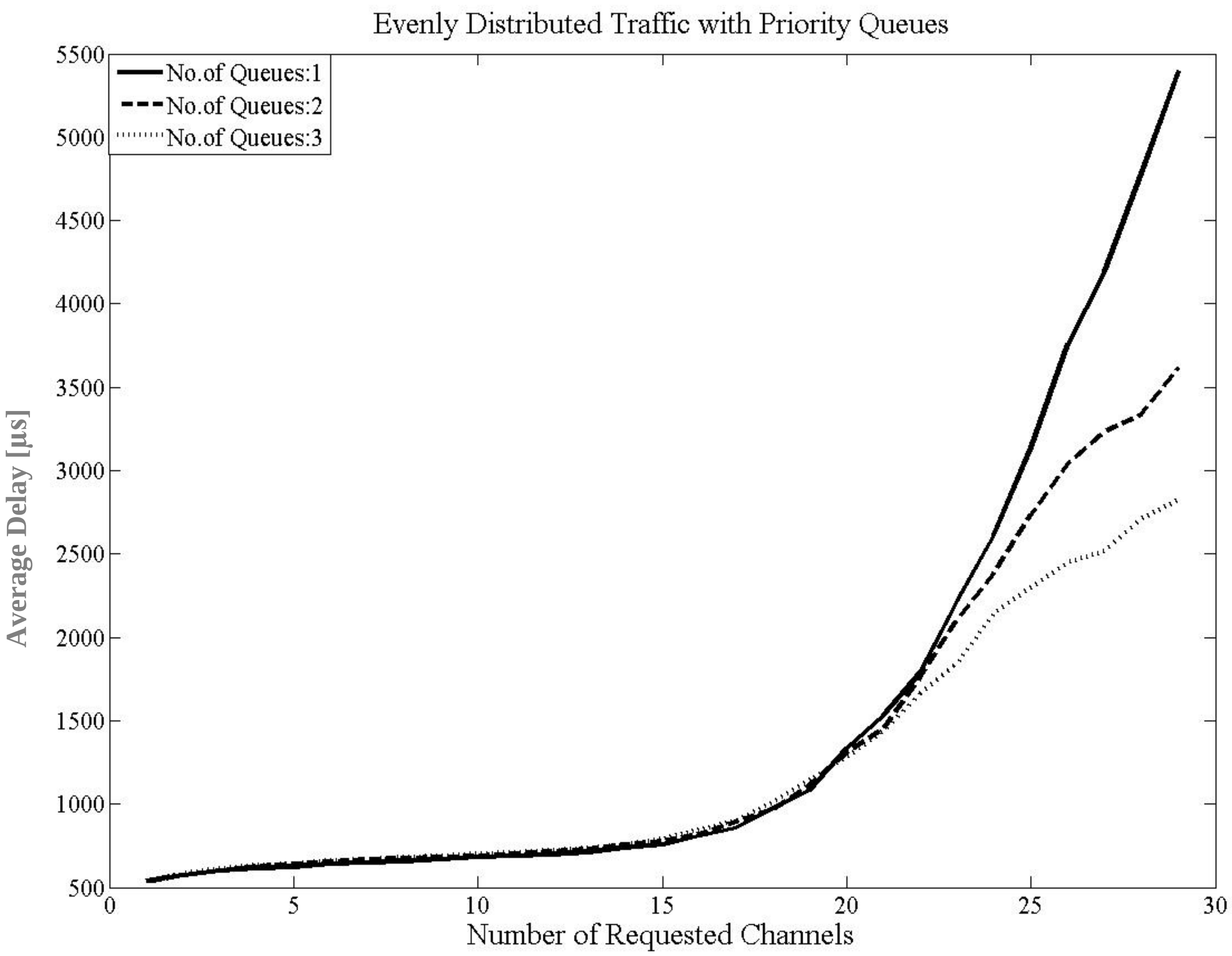


Figure 7: Average delay for evenly distributed traffic

This figure also shows the clear difference among the different priority levels. We can see that the average delay decreases with the increase of priority level.

Like figure 5, in figure 7 there is no difference for low traffic load. The effect becomes clear with the increment of traffic load. When the network is more congested the effect of multiple priority queues becomes significant.

The last point of dotted line is near 3000µs where as the last point of solid line is near 5500µs which is nearly double the dotted line.

Figure 8 below shows the deadline miss ratio comparison among different priority levels but, having evenly distributed traffic in network.

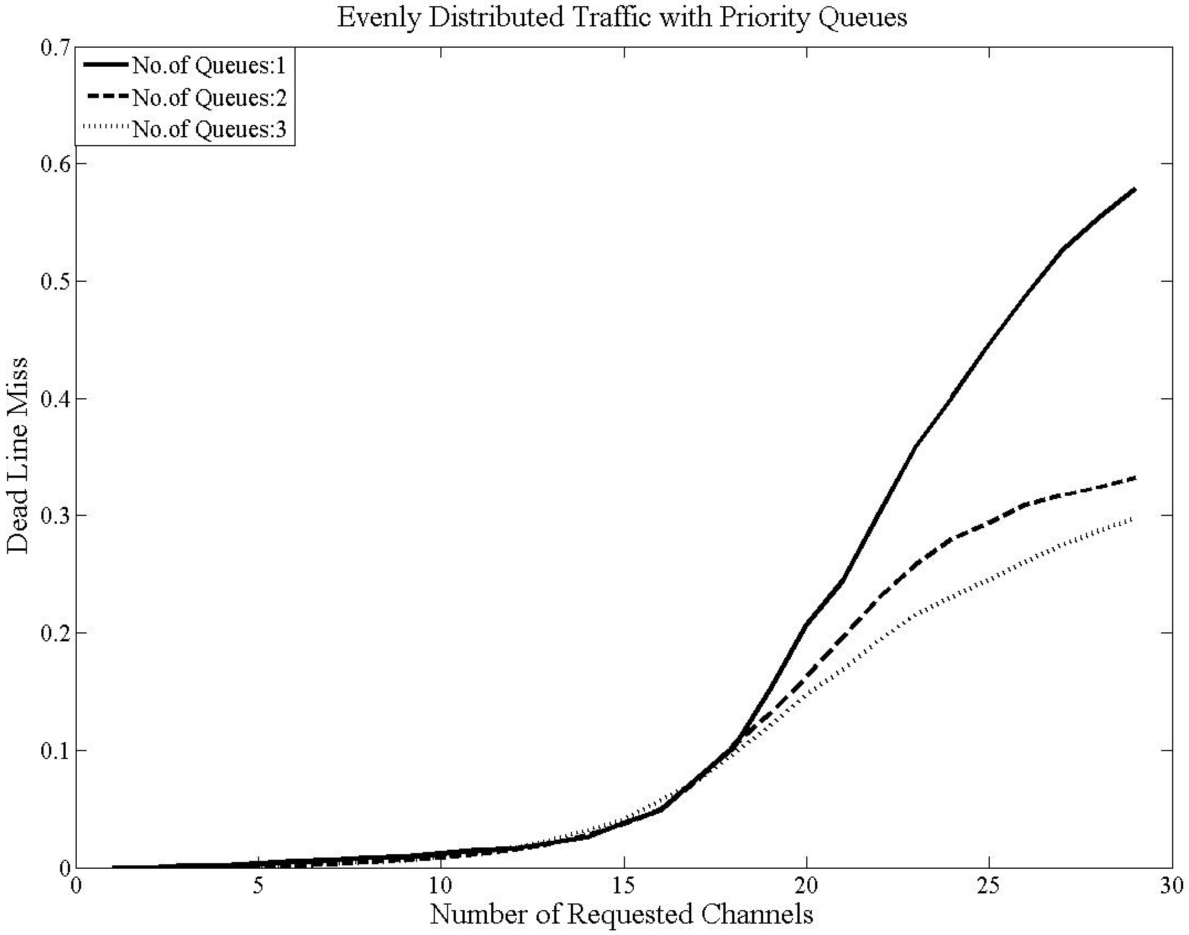


Figure 8: Dead line miss ratio for evenly distributed traffic

Figure 8 shows the graph of deadline miss ratio for evenly distributed traffic according to the same parameters as given above. This graph depicts that the deadline miss ratio decreases with the increase of the number of priority levels. In the graph, the solid line is for a single queue. The dashed line shows two priority levels and the dotted one is for three priority levels.

From Figure 8, we see in the beginning that the deadline miss ratio is the same for all the cases, but soon it depicts the clear effect of multiple priority levels. When the traffic load increases, the solid line gets higher than other two, showing that the deadline miss ratio is high for that case.

For the dashed line, the deadline miss ratio is lower than the solid line and higher than the dotted line. The dotted line shows that the deadline miss ratio is lower than the other two cases. We noticed in this figure that the three priority queues produces the better results and have positive effect on the performance of real-time system.

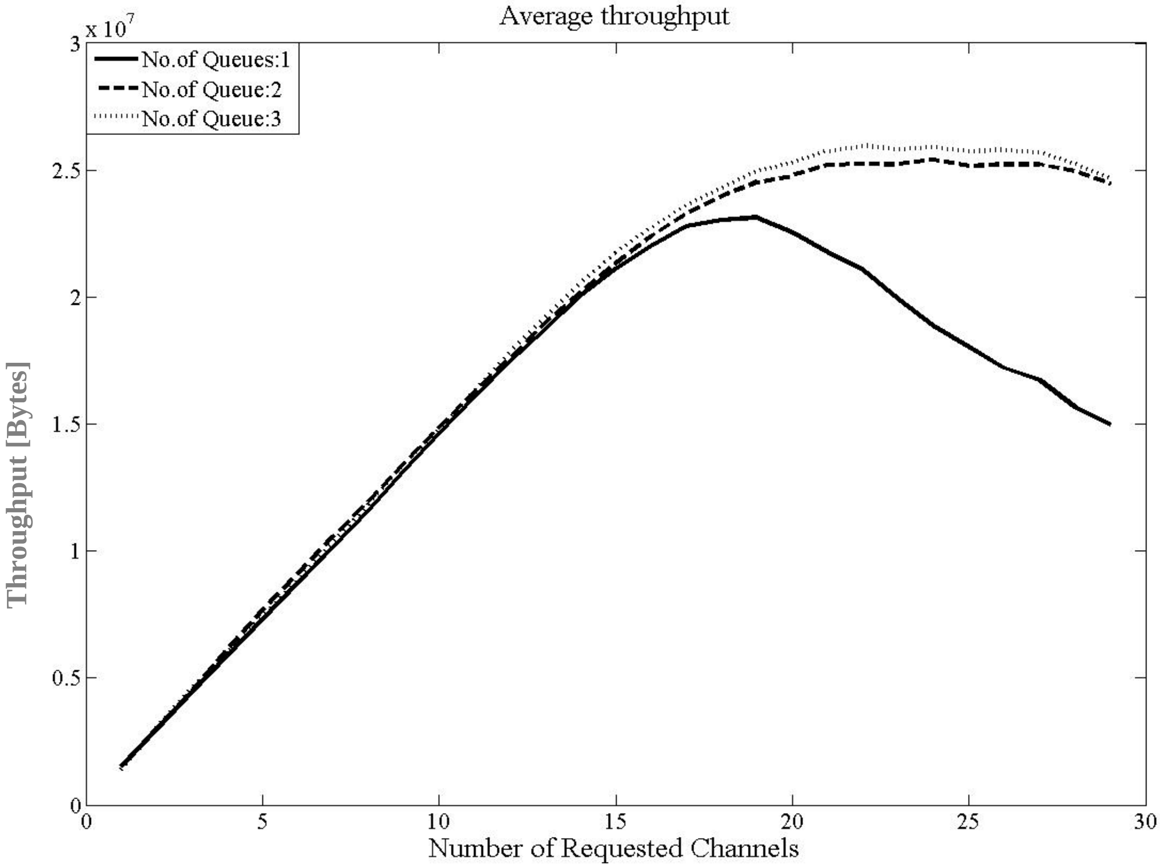


Figure 9: Average throughput

Figure 9 shows the average throughput comparison among different priority levels.

For a system having low network traffic load, the better performance can be achieved using single priority level, but with high traffic load, we need more priority levels to get better performance in the form of high throughput. As it can be seen form the figure 9, when there is more channel request (>15) the throughput of the priority level 3 is better than the priority level 2 which is better than the priority level 1.

It is clear from the figure 9 that the single queue system becomes saturated earlier than the two or three priority levels systems and starts missing deadline more frequently, and ultimately throughput starts decreasing, but the systems of two and three priority levels never get saturated as soon as for the case of one queue, because they handle the traffic more efficiently. The performance gets better with the increment of priority level. We can see that the dotted line is above than the solid and dashed lines, showing the higher throughput and better performance.

On the basis of the above graph, it can be said that the increase in priority level increases the system performance by increasing the throughput, decreasing the average delay and deadline miss ratio.

4 Conclusion

In this thesis project, we have presented a switched multi-hop priority queued network supporting real-time communication. Implementing the priority queues in each switch has a positive impact on the performance of soft real-time systems. From the simulation experiments, it is clear that the increase in priority level increases the system performance by increasing throughput, decreasing delay and decreasing deadline miss ratio.

It is observed that there should not be too many traffic types as high priority. Because the priority queuing is not fair, in the sense that, the high priority packets are always served first; If there is a constant stream of packets of high priority, the packets in the low priority queue are ignored until the high priority queue is empty. So the packets in low priority queue face starvation and miss their deadlines. Moreover, it is also observed that the evenly distributed traffic suffers less delay and less deadline miss ratio as compared to bursty traffic. However, the effect of priority levels for both kinds of traffic is positive. The increase in priority level brings the better performance in soft real-time systems by guaranteeing the quality of service.

5 References

[1] Eytan Modiano, "Scheduling Packet Transmissions in a Multi-hop Packet Switched Network Based on Message Length", 1095-2055/97 IEEE.

[2] Jörg Liebeherr, "Priority Queue Schedulers with Approximate Sorting in Output-Buffered Switches", IEEE Journal on Selected Areas in communication, vol.17, no. 6, June 1999.

[3] Manolis G. H. Katevenis, Aggelos Ioannou, "Pipelined Heap (Priority Queue) Management for Advanced Scheduling in High-Speed Networks", IEEE/ACM Transaction on Networking, vol. 15, no. 2, April 2000.

[4] Ranjita Bhagwan, Bill Lin, "Fast and Scalable Priority Queue Architecture for High-Speed Network Switches" INFOCOM 2000.

[5] D. Ferrari and D. Verma, "A scheme for real-time channel establishmentin wide-area networks" IEEE Journal on Selected Areas in Communications, vol. 8, no. 4, pp 368-379, April 1990.

[6] H. Zhang and D. Ferrari, "Rate-controlled service disciplines," Journal of High Speed Networks, vol. 3, no. 4, pp. 389-412, 1994.

[7] Kai Zhu, Yan Zhuang, Yannis Viniotis, "Achieving End-to-end Delay Bounds by EDF Traffic Shaping" IEEE INFOCOM 2001.

[8] Arshad lqbal, Asia Zafar, Bushra Siddique, "Dynamic Queue Deadline First SchedulingAlgorithm for Soft Real-Time Systems", IEEE, International Conference on Emnerging Technologies September 17-18, 2005.

[9] Dinesh C. Verma, Hui Zhang, Domenico Ferrari, "Delay Jitter Control for Real-Time Communication in a Packet Switching Network", In Proceedings of TriComm,1991.

[10] Priority and weighted fair queuing in IP networks, ISBN 82-423-0574-9.

[11] Hemant M. Chaskar, Upamanyu Madhow, "Fair Scheduling With Tunable Latency:A Round-Robin Approach," IEEE/ACM Transactions on networking, vol. 11, no. 4, August 2003.

[12] H.Hoang, M.Jonsson, A.Kallerdahl and U.Hagström, "Switched Real-Time Ethernet with Earliest Deadline First Scheduling - Protocols, Traffic Handling and Simulation Analysis," 2000, Parallel and Distributed Computing Practices (PDCP), 5(1).

[13] Domenico Ferrari, Fellow, IEEE and Dinesh c,Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks" vol. 8.no. 3. April 1990.

[14] Kai Chiu Kwan, Parameswaran Ramanat han "Multiple Route Real-Time Channels in Packet – Switched Networks", Real-time systems symposium, December 1994.

[15] H.Hoang, M.Jonsson, "Real-time communication for industrial embedded systems using switched Ethernet", in ARTES – A network for Rel-Time research and graduate Education in Sweden 1997-2006, Editor: Hans Hansson, pp. 417-427, Mar. 2006, ISBN 91-506-1859-8.