Effective Fair Share Resource Management Algorithms in Support of Quality-of-Service

Qi Allen Wang, Chung-Horng Lung, Shikharesh Majumdar
Dept. of Systems and Computer Engineering
Carleton University,
Ottawa, Canada
Email: {chlung | majumdar}@sce.carleton.ca

ABSTRACT

This paper concerns the attainment of QoS on a network and focuses on the management of two resources on routers: the link bandwidth and the CPU for achieving fair share resource management. A comparative analysis of four different schedulers is performed. In addition to CPU scheduling, this paper also investigates three different packet-dropping policies. Based on measurements made on a performance prototype of routers, insights into system behaviour and performance are presented.

Keywords: Quality of service, DiffServ, queuing, scheduling, dropping, Chorus systems

1. INTRODUCTION

Traditional Internet only supports one type of service for all applications. Under this so-called best effort concept, all packets are treated identically on a First Come First Served basis. The approach works well for non-real time applications, such as FTP. However, using this approach, there is no way to differentiate among various services or applications. With the increasing demands of supporting multimedia and real time applications, such as Internet telephony and video applications, services with a higher quality of service (QoS) guarantee are required.

Best effort service can not satisfy diverse QoS requirements of applications. In other words, delay sensitive data, such as voice and video applications, must be processed faster than normal data packets. To support different treatment of packets based on QoS requirements, multiple queues with different priorities and a scheduling policy as well as a packet dropping strategy are needed, so that packets from applications with similar QoS requirements are assigned the same service queue. Packets that require higher QoS are marked with higher priority will have higher precedence over those with lower priorities in terms of processing time and dropping.

The DiffServ [1] model was proposed to support QoS in the Internet. In this model, QoS is classified into three main classes: EF (Expedited Forwarding), AF (Assured Forwarding) and BE (Best Effort). EF is used to support premium services which require low loss rate, low delay and low delay jitter. AF provides low loss rate without guarantee on delay. BE, on the other hand, provides no service guarantee.

Due to dynamic traffic flows and random destinations on the Internet, it is difficult to perform connection admission control and resource reservation statically. As a result, it is also difficult to perform scheduling based on static setting of weights. Approaches using dynamic adjustment have been proposed to mitigate the problem. Dynamic adjustments provided by these approaches are based primarily on bandwidth allocation. This paper concerns the management of two resources the link and the CPU for achieving fair share resource management. On such a system a designated portion of the link bandwidth and CPU time is given to each of the three packet groups such that a differentiated quality of service is provided to the packet groups. Quality of service is characterized by two performance metrics: the packet drop rate and the mean packet roundtrip time. This research presents dynamic and adaptive scheduling and packet dropping algorithms based on both CPU share as well as bandwidth allocation, and investigates their performance.

The performance analysis was conducted on a ChorusOS-based real-time performance router prototype running on a network of Pentium PC's. ChorusOS was understood to be an attractive real-time operating system for running telecommunication applications. ChrousOS provides high performance and high availability with a simple, flexible configuration mechanism [2]. In addition, it also provides a complete host-target development environment for its users. This enables the user to develop an application on a host system and execute it on a reference target board where ChorusOS is running [3, 4]. This research was motivated by Nortel Networks, which was interested in resource management for telecommunication applications such as switches and routers running on top of ChorusOS. Note that the conclusions derived are of general nature and are expected to be valid in the context of other systems and hardware platforms.

The main contributions of this paper are briefly summarized.

- This paper introduces and integrates several packet dropping and scheduling policies that aim at providing differentiated service among packet groups.

- Based on a synthetic workload and a performance prototype of the router the performances of these policies are investigated under various combinations of system and workload parameters.
- Insights into system behaviour and performance that include the different characteristics of the resource management policies are presented.
- The performance results indicate that a combination of the Dynamic-Link packet dropping and the Adaptive scheduling policies proposed in this paper can lead to both a lower packet drop rate and mean round trip time.
- Two different factors that can be used to tune the degree of service differentiation provided by these strategies are discussed.

The rest of the paper is organized as follows. Section 2 presents an overview of related work. Section 3 describes the experimental environment including the performance prototype and discusses the algorithms that are investigated. Section 4 presents the results. Our conclusions are presented in Section 5.

2. RELATED WORK

QoS-based scheduling and bandwidth allocation has been studied intensively, e.g., [5]-[16]. Typically scheduling strategies include First Come First Served (FCFS), Priority Queuing, Class-Based Queuing (CBQ), and Weighted Fair Queuing. FCFS may cause significant delay. Priority Queuing supports multiple levels of priorities to serve different packets. On the other hand, this flexibility could lead to starvation for the lower priority group of packets.

Class-Based Queuing specifies the rules for processing various types of packets in order to achieve the desired preference for different groups of packets. CBQ is considered to be a simple method for providing link sharing for various classes of services [8]. However, the computational overhead limits its usefulness for providing differentiated classes of service on high-speed links.

Weighted Fair Queuing (WFQ) sorts the incoming packets into separate flow queues and sends out a fixed portion for each flow at a time. The bandwidth is distributed into shares in accordance with the weights specified for the different active flows, and the lowest volume flow finishes the process first. In this way, WFQ prevents longer flows from consuming network resources that could starve shorter flows. Similar to CBQ, the computation overhead of WFQ is the major obstacle to its scalability.

Scheduling based on statically allocated bandwidth is conservative and will result in low link utilization. For service class with strict delay requirement such as the EF class in the DiffServ model, to allow some burstiness, the allocated bandwidth rate can be much larger than its average rate. As a result, if static bandwidth allocation is used in the scheduling policy, over-provisioning is usually needed, which causes low link utilization. On the other hand, scheduling based on dynamic bandwidth allocation adjusts the bandwidth allocation based on the current QoS measurement. The main problem with this kind of approach is to determine how often the QoS should be measured so that bandwidth allocations can be adjusted accordingly.

Packet dropping policies for IP networks have been studied (see [17] for example). The main design issues in this area include network utilization and application throughput, fairness, simplicity, global synchronization, and scalability. As pointed out in [17], no single approach addresses all of the design issues. Each policy is designed for specific objectives. Random Early Detection (RED) and its variations are commonly discussed. RED was designed to work in collaboration with a transport layer congestion control protocol, such as TCP. This paper investigates tail dropping policies that focus on fairness among different traffic types and are not dependent on the transport layer protocol. We mainly address the effect of dropping policies together with the scheduling policy from the QoS perspective.

3. EXPERIMENTAL ENVIRONMENT

A ChorusOS-based performance prototype of a network router is constructed. This prototype contains only the necessary components for investigating the performances of the resource management strategies. This performance prototype consists of three PCs and one Sun workstation. These three PCs are connected to each other through a switch, while the PCs connect to the Solaris workstation through a private 10 Mbps LAN. The workstation is the host, on which the ChorusOS development environment has been installed. The three PCs are the targets on which the ChorusOS runtime environment is installed. The applications running on the targets are remotely controlled by the host. Each of these three targets acts as a router.

The performance prototype of a network router is based on the CG-Net system [18], which is a software-based network router developed by Nortel Networks. It is designed for the performance improvement study of an IP router subjected to different routing and forwarding protocols. CG-Net consists of a set of emulated core routers; each in turn is composed of the following processes: *generator*, *node*, *sink*, *traffic controller*, and *routing* and *forwarding tables*.

A generator process randomly generates packets that will be forwarded to other router processes. A sink process consumes packets received from other routers or its own generator. A traffic controller process automatically sends the appropriate commands to the network to formulate the necessary network changes required to improve network status. These commands are based on network status from real-time statistics. A node process forwards traffic (control and data) towards the destination sink in accordance with the routing and forwarding tables which are populated by different protocols. Inside of a node process, there are three priority queues that are used for each packet group. In addition, there are destination queues and destination threads. A destination queue and a destination thread are used to emulate an outgoing interface.

The operational behavior of the prototype follows what real routers do to handle packets. Packets are created by the generator processes and sent to their corresponding node process. The node process receives packets from its generator or other node processes and puts the packets in the appropriate priority queues based on the QoS level. The node process subsequently examines the routing table or forwarding table depending on the protocols and find out the outgoing interface. Each packet will then be put into the appropriate destination queue and sent to the next hop or node process by the destination thread.

For this prototype, we are concerned with three packets groups, each of which has a different priority. The highest priority packets correspond to the *Gold* class. Those packets are put into the highest priority queue. The middle priority packets correspond to the *Silver* service class and they go into the middle priority queue. The lowest priority packets correspond to the *Bronze* service class. These packets go to the lowest priority queue. The mapping of these different classes to packet groups depends on the system. In the context of the Diffserv model the EF, AF and BE classes map to the Gold, Silver and Bronze classes, respectively.

3.1. Performance Metrics and Parameters

A number of metrics has been used in network performance analysis and measurement. This research adopts three of the most common ones: packet drop rate, mean roundtrip time and CPU utilization.

Performance Metrics:

- Packet Drop Rate (%)

Each link has a specified capacity and packets are dropped when the link capacity is exceeded. Packet drop rate is defined as:

*PacketDropRate =[(No. of PacketsGenerated – No. of PacketsSent)] / No. of PacketsGenerated * 100*

A packet can be dropped from the priority queue when its waiting time exceeds a pre-defined delay or from the destination queue when there is no bandwidth available. In all the experiments reported in this paper, the pre-defined delay parameter was fixed at 20 ms.

Note that the PacketDropRate can be measured for a given group as well as the overall router. In the first case the number of packets generated and number of packets sent correspond to a specific group. In the second case the number of packets generated and the number of packets sent used in the computation of an *overall drop rate* correspond to the entire router including all the packet groups.

- Mean Roundtrip Time

The roundtrip time for a packet is the sum of the times used by the packet to travel from the source router to the destination router including the time for the acknowledgement from the destination to reach the source. It is used to study the impact of network traffic and scheduling policies on performance.

- Group CPU Utilization (GCU[i], i = 1, 2, 3)

A packet group's GCU is the actual CPU share consumed by the group during a specific period. 1 refers to gold, 2 to Silver and 3 to Bronze. For instance, if the Gold group consumes CPU for 40 seconds out of 100 seconds, GCU[1] is 40%.

This prototype involves various parameters that characterize the workload and the system. They are:

System and Workload Parameters:

- Share [i] (i = 1, 2, 3)

This parameter is used in the Dynamic-Priority and Adaptive scheduling policies that are described in a following subsection. Share [i] specifies the proportion of CPU times that are to be allocated to the different groups of packets. For example, if the CPU Share for the Gold packet group is 30%, the scheduler attempts to provide the Gold packets 30% of the total CPU time.

- Packet Arrival Rate (P/s)

This parameter specifies the rate (packets/sec) at which packets are generated by the packet generator.

- Packet Group Ratio (p1:p2: p3)

This parameter specifies the ratio of the number of packets generated in each group. In an experiment, for a set of (p1 + p2 + p3) packets, there are p1 Gold packets, p2 Silver packets and p3 Bronze packets.

- Packet Group Proportions [i] [i = 1,2,3]

This parameter specifies the proportion (%) of packets generated for each group. The Proportion is related to the Packet Group Ratio as follows:

*Packet Group Proportion [i] = pi/(p1+p2+p3)*100 where (i = 1,2, 3).*

- Link-Share[i] [i = 1,2,3]

This parameter is used by the Fixed-Link policy described in Section 3.2.1. Link-Share[i] specifies the proportion of the link capacity for each packet group. The system allocates Link-Share[1] of the total link to the Gold packet group; Link-Share[3] of the total link capacity is given to the Bronze packet group; and the remaining capacity (Link-Share[2]) is allocated to the Silver packet group.

- Packet Processing Time (Ptime)

This parameter specifies in μs the CPU time required to process a packet. For each experiment it is held at a fixed value, 300 μ sec.

3.2. Resource Management Policies

The typical resources to be managed are the output buffer and the link bandwidth for routers that have dedicated hardware to process packets. However, more and more high-end computers, e.g., Linux machines, which do not have those hardware devices, are used as routers. In this case, CPU also becomes an important resource that needs to be managed. The paper focuses on link bandwidth and CPU time. The packet dropping policies described in the next subsection concern the strategy for discarding packets based on link bandwidth whereas the scheduling polices described in the following subsection are used to determine the order in which the packets in the different groups are executed on the CPU and forwarded to the destination queue.

3.2.1 Packet Dropping Policies

Three dropping policies are described next.

- Default-Link

With this dropping policy, all packets have the same priority. Packets are processed on a FCFS basis. Irrespective of the packet class, if the available bandwidth is not large enough to accommodate the incoming packet, the packet is dropped. Obviously, this dropping policy does not inherently provide any kind of differentiated service based on priority and is used as a basis for comparison with other dropping strategies

- Fixed-Link

Based on the Link-Share vector, the capacity of each link is split into a fixed number of portions for the packet groups. The packet transmission decision is based on the available bandwidth for its group. That is, if there is not enough bandwidth available for its group to transmit this packet, instead of checking the available bandwidth for the overall link, the destination thread, used to emulate the line interface, drops the packet.

The Fixed-Link policy can provide differentiated service to different classes of customers. A higher portion of link capacity is reserved for the higher priority packets, and lower priority packets are offered a lower portion of the link capacity.

- Dynamic-Link

This policy is a combination of the above two policies. The principle of the Dynamic-Link policy is to reserve only a fixed but relatively small portion of link capacity for the higher priority packets, and leave the rest to be shared by all the groups. A more detailed discussion is provided in the following paragraphs.

The Dynamic-Link policy emerges from the following observation. With the Fixed-Link dropping policy, the link capacity is split into three portions: 50% goes to the Gold, 30% to the Silver group and 20% to the BE group, for example. If the packets generated in the Gold group can use only 20% of the total link capacity, the other 30% that is allocated to the Gold group will be wasted. With the Dynamic-Link policy, the other groups can share the unused portion of the link capacity.

In Dynamic-Link, the total link capacity is divided into three portions: the first portion is reserved for Gold, the second portion is reserved for both Gold and Silver, and the third portion is used for all the groups. A drop threshold is used for each of the lower priority packet groups. The threshold value n1 is for the Silver group and n2 is for the Bronze group. As long as the available bandwidth is lower than the drop threshold n2, the packet in the Bronze group will be dropped if necessary. Similarly, when the available bandwidth is lower than the drop threshold n1, packets in both the Silver and Bronze groups will be dropped if necessary.

3.2.2 Scheduling Policies

This section describes two algorithms that dynamically adjust the services of the different queues or packet classes based on the processor share. These two algorithms are compared with two static scheduling policies: *Equal-Priority* (FCFS) and *Fixed-Priority* (Priority Queuing). For Equal Priority there is no differentiated service provided by the scheduler and the packets are processed in a First Come First Served manner. In case of Fixed priority the packets in the Gold class are processed first followed by the packets in the Silver class. The packets in the Bronze class are given the lowest priority. This type of scheduler ensures that higher-priority queues will always be emptied before packets are emptied from lower-priority queues. Bronze packets, for instance, will only be processed when there are no Gold or Silver packets. It is equivalent to the Priority Queuing strategy.

For both the *Dynamic-Priority* and the *Adaptive* scheduling policies introduced in this paper, the scheduler uses Share [i] to make the scheduling decision. This is the difference between them and WFQ which calculates the "share" based on the number of bytes of each priority flow. The two algorithms are described in detail in the following sub-sections.

3.2.2.1 The Dynamic-Priority Scheduling Policy

The initial setting of the priority is the same as that of the Fixed Priority, i.e., the packets are processed in accordance with their group priorities. At a given point in time, if the Packet Group Proportion of a packet group is much lower than its group's Share, it is likely that this group's queue does not have a packet to process. With dynamic-priority, if there is no packet waiting for processing, the CPU share is given to another queue that has packets waiting to be processed.

Figure 1 shows the complete pseudo code for the Dynamic-Priority scheduler. A more detailed description of priority settings of the implementation can be found in [19]. Note that with ChorusOS, a lower priority number implies a higher priority.

Consider an example in which the Packet Group Proportions for the three groups are 20% (Gold), 30% (Silver) and 50% (Bronze), and the Shares are: 50% (Gold), 30% (Silver) and 20% (Bronze). Because the

Share of Gold is the highest, so queue 1 gets a greater chance of being processed at the highest priority.

However, the Packet Group Proportion of the Gold group is the lowest of the three groups, and Gold's queue (queue 1) has a greater chance of being idle when it has been set at the highest priority. Because of this idle period within the Gold group, the scheduler gives the remaining CPU share of Gold to either queue 2 (for Silver) or queue 3 (for Bronze) and let them process the packets in their groups (Silver or Bronze).

3.2.2.2 The Adaptive Scheduling Policy

The initial setting for this policy is the same as that of Fixed Priority. When the scheduler that runs periodically wakes up, it first updates the CPU usage for each queue. The scheduler then compares the CPU usage of a class i to its Share[i]. The queue with the smallest ratio of CPU usage and Share is assigned the highest priority; the queue with the greatest ratio of CPU usage and Share is assigned the lowest priority; and the third queue is set to the middle priority. The pseudo code for the scheduler with Adaptive scheduling policy is presented in Figure 2. Note that in ChrousOS a higher priority number implies a lower priority.

Use the same example as the one in the previous section. Queue 1 has the smallest ratio of CPU usage and CPU Share, it will be assigned the highest priority. Queue 3 exhibits the highest ratio of CPU usage and Share, and it will be assigned at the lowest priority. Queue 2 will be set assigned the medium priority level.

```
Wakeup
  Set the priority of all the scheduled queue back to the lowest
  Update the CPU usage for each queue
  IF the scheduled queue consumes some CPU time (there were
     packets in the corresponding group waiting to be processed)
     Compare the CPU usage for each queue
     Find the group with the smallest ratio of CPU usage and Share
     Choose this group's queue as next scheduling queue
  ELSE (There was no waiting packet for the group)
     IF only one queue did not consume CPU at previous period
        Choose the queue as next scheduling queue
     ELSE IF both the queues did not consume any CPU time
        Compare the CPU usage for each queue
        Find the queue with smallest ratio of CPU usage and Share
        Choose this queue as next scheduling queue
        Reset the idle scheduling counters for each queue
     ELSE (all the remaining queues did consume CPU time)
        Compare the CPU usage for the other two queues
        Find the queue with next smaller ratio of CPU usage / Share
        Choose this queue as next scheduling queue
     ENDIF
  ENDIF
  Raise the priority of the queue that is chosen as next scheduling
     queue (to 152)
  IF this queue has already consumed CPU > its group's Share
     Decrease the scheduling time interval by 1/2
     Compare the CPU usages of the other two queues
     Find the queue with smaller ratio of CPU usage and Share
     Raise the priority of this queue as well to 151
  ENDIF
Sleep for the scheduled time interval
```

Figure 1. **Dynamic-Priority Scheduling Algorithm**

```
Wake up
  Update CPU usage of each queue
  Compare each group's CPU usage with its Share
  Find the queue with the smallest ratio of CPU usage and Share
  Raise the priority of this queue to the highest level (152)
  Find the queue with the highest ratio of CPU usage and Share
  Change the priority of this queue to the lowest level (158)
  Change the priority of the third queue to the middle level (155)
  {Adjust schedule-interval}
  IF the ratio of GCU to Share for the highest priority group is < 0.9
       Multiply the scheduling time interval by 2
  ELSE IF the ratio is greater than 1.1
       Multiply the scheduling time interval by 0.5
  ENDIF
Sleep for the scheduled time interval
```

Figure 2. Adaptive Scheduling Algorithm

4. EXPERIMENTAL RESULTS

The experiments described here are intended to study the impact of different factors on the packet drop rate and roundtrip times. The primary focus is on packet-dropping and scheduling policies. The impact of four parameters on performance is investigated. These factors include packet arrival rate, packet dropping policy, scheduling policy, and packet processing time.

We conducted a number of experiments based on these four factors. A representative set of results are presented here. More data is available in [19]. The effect of packet dropping policies on drop rate is presented first. This is followed by a discussion of the impact of the scheduling policies on performance. For the experiments described in this paper, the total link capacity between any two links is fixed at 2Mbps. In a number of cases we have analyzed the performance of the resource management strategies at high arrival rates that give rise to large drop rates for example. Although the engineered arrival rates to which some routers are subjected to may be lower, the bursty nature of the internet traffic or the occurrence of an unexpected event can give rise to such temporary increases in load. The performance results observed with high arrival rates will be useful in understanding the relative performances of the resource management strategies during the occurrence of such temporary high loads.

4.1. Effect of Packet Dropping Policies

The packet dropping policies affect the packet dropping rate and not the mean roundtrip times that are computed from the roundtrip times of the packets that were not dropped. Thus, only the packet drop rate is used for comparing the performances of the packet dropping policies.

A comparison between the Fixed-Link policy with the Default-Link policy is presented in Figure 3. The packet group proportions of 33.33%, 33.33% and 33.33% are used in the experiment. The link shares for Gold, Silver and Bronze used by the Fixed-Link policy are held at 50%, 30% and 20% respectively. As the arrival rate increases more and more packets arrive per unit time on the system. As expected an increase in packet arrival rate leads to a higher dropping rate. Since the Default-Link policy does not support a differentiated QoS for the three customer classes, and the packet group proportions are equal to one another, the dropping rates achieved by the Gold, Silver and Bronze classes are equal to one another.

For any given arrival rate, the performance of Gold is superior to that of Silver followed by Bronze. Thus, the Fixed Link policy does provide the desired difference in QoS among the three classes. The overall packet drop rate achieved by Fixed Link seems to be higher than that of Default that does not differentiate among the three classes and treat them equally. Thus the differentiated QoS provided by Fixed Link is achieved at a cost of an increase in the overall drop rate.

This deterioration in performance is caused by a poor link utilization demonstrated by Fixed Link at lower arrival rates shown in Figure 3. Since the arrival rates for packets in each class is the same but the link share for Gold is much higher than the two other classes, it is possible that the a portion of the bandwidth allocated to Gold remains unused while the shares for the two other classes are fully utilized leading to the dropping of packets in these classes. As the packet arrival rate increases, the unused portion of the Gold share disappears and the overall dropping rates for Default-Link and Fixed-Link become equal to one another.

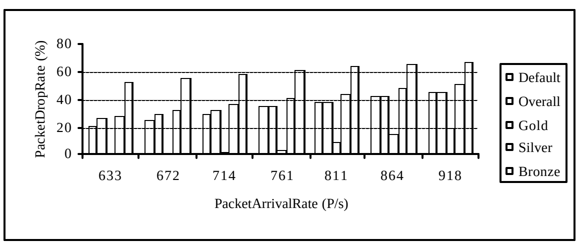

Figure 3. Comparison of the Default-Link and the Fixed-Link Policies

The performance of the Dynamic-Link policy is compared with that of Default-Link and Fixed Link in Figure 4. The arrival rate is fixed at 633 P/s while the packet group proportions are held at the same values used in the pervious experiments. Various values of n1 and n2 are experimented with: these values are indicated along the X-axis for the graph displayed in Figure 4. The last set of values correspond to n1 = 2%, n2 = 0.5%. Thus, 2% of the link capacity is reserved for Gold, 2.5 for both Gold and Silver whereas 97.5% of the link capacity is shared by all the classes. With these values, the Gold class achieves a near zero drop rate whereas the drop rates achieved by both Silver and Bronze are lower than those achieved with the Fixed Link strategy. By dynamically utilizing the spare capacity left by one class, Dynamic-Link shows a superior performance in comparison to Fixed-Link that associates a fixed link share with each class. The drop rates achieved by all three classes are equal for Default that does not support a differentiated QoS. The drop rates for both Gold and Silver achieved with Dynamic-Link are lower than those achieved with Default-Link while the drop rate for Bronze is higher than that achieved with Default-Link.

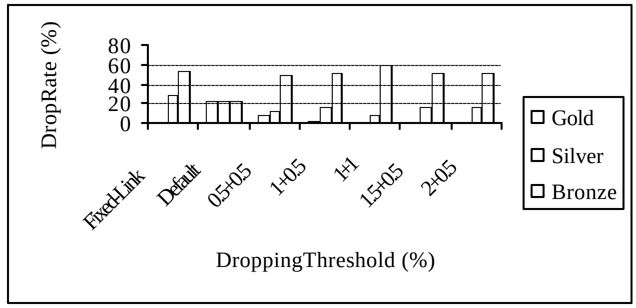

Figure 4. Performance of the Three Packet Dropping Policies

4.2 The Effect of Scheduling Policies on Drop Rate

The packet drop rate achieved with different scheduling and packet dropping policies is discussed next. The relative performances of the scheduling policies are comparable when used in conjunction with the Default-Link and the Dynamic-Link scheduling policies. So, we have included a discussion of the scheduling policies in the context of the Default-Link and Fixed-Link dropping policies only.

4.2.1 The Effect of Scheduling Policies When Used with the Default-Link Dropping Policy

Although the overall packet drop rates are close to each other, the drop rate of every single group of packets varies when different scheduling policies are used. The workload and system parameters used in the experiment that demonstrates the influence of scheduling policies on system behaviour and performance are presented in Table 1.

Table 1. Input Data for Experiment Investigating the Effect of Scheduling Policies

	Gold	Silver	Bronze
Packet Group Proportion (%)	40	30	30
Share (%)	40	30	30
Packet arrival rate (P/s)	918		

Figure 5 shows the packet drop rates for the groups using different scheduling policies and the Default-Link dropping policy. The Fixed-Priority scheduling policy starves the lowest priority packets, which leads to the highest drop rate for the Bronze group. The Equal-Priority (FCFS) scheduling policy serves all groups of packets with a similar drop rate for each group. Both the Dynamic-Priority and Adaptive scheduling policies have a drop rate close to that of Equal-Priority, because the CPU shares allocated to the groups are the same as the Packet Group Proportions (see Table 1).

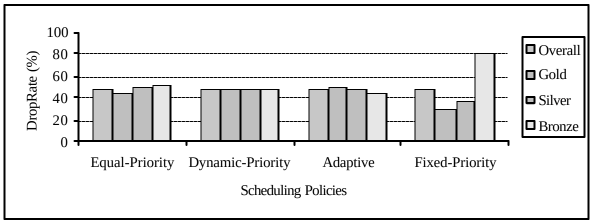

Figure 5. The Effect of Scheduling Policy on Drop Rates with the Default-Link Packet Dropping Policy

The impact of the scheduling interval used in the Adaptive scheduling on performance is captured in Figure 6. It shows the changes in drop rate when the scheduling interval is varied. When the interval increases, the differences in drop rates exhibited by the different packet classes tend to become higher.

The differences in dropping rates are small except for the last value of interval used. When the scheduling interval is larger than or equal to the total run time of the experiment (Max Time), there is essentially no re-scheduling and the Adaptive policy then degenerates to the Fixed Priority scheduling policy (see Figure 6).

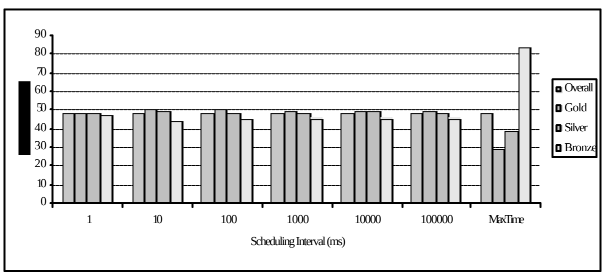

Figure 6. The Impact of Scheduling Interval for the Adaptive Scheduling Policy on Drop Rate

4.2.2 The Effect of Scheduling Policies when Used with the Fixed-Link Dropping Policy

When the packet arrival rate is low compared to the rate that achieves the full link capacity of the physical link, the scheduling policies demonstrate a comparable drop rate. Table 2 provides a summary of the input data for the experiment that investigates this effect. The results are presented in Figure 7.

Table 2. Parameters for the Experiment in which Link-Share is Equal to Packet Group Proportion

	Gold	Silver	Bronze
Packet Group Proportion (%)	Varies		
Share (%)	Same as Packet Group Proportion		
Link-Share (%)	Same as Share		
Packet arrival rate (P/s)	633		

Figure 7 shows the overall drop rate for each scheduling policy when Share is the same as the Packet Group Proportion. The Link-Shares are the same as the CPU shares allocated to the groups that are captured in the Share vector.

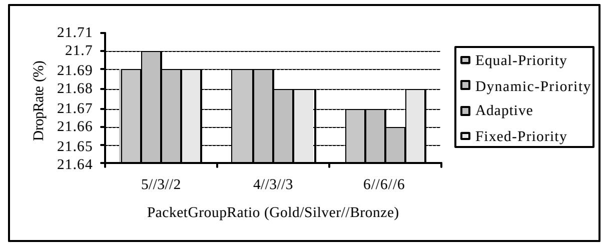

Figure 7. Overall Drop Rates Achieved with Proportional Shares

Figure 7 shows that the overall drop rates are close to each other. The actual differences are less than 0.2%. A similar observation is made to systems using non-proportional Shares. Thus the drop rates achieved by the scheduling policies seem to be fairly insensitive to the change in Packet Group Proportions.

However, as shown in Figure 8, the Fixed Priority scheduling policy can cause bandwidth to be wasted at higher packet arrival rates. Because of starvation of the queue that processes the Bronze packets, the bandwidth assigned to the Bronze group may not be fully utilized. The input parameters for the experiment investigating this effect are shown in Table 3.

Table 3. Input Parameters for the Experiment Investigating Starvation

	Gold	Silver	Bronze
Packet Group Ratio	Varies	Varies	Varies
Link-Share (%)	Same as Packet Group Proportion		
Packet arrival rate (P/s)	811		

Figure 8 shows the effect of Packet Group Ratio on the overall drop rate observed at high arrival rates for the Fixed-Priority policy. Using this scheduling policy, the packets in higher priority groups are processed earlier than the packets in the lower priority groups. When the available CPU time is not large enough to handle all the incoming traffic, packets are dropped. Since the Bronze packets are given the lowest priority, a large proportion of the Bronze packets are dropped. As the packet arrival rate is held at 811 P/s and the Packet Group Ratio for Bronze is reduced, the ratio of the number of Bronze packets that are dropped and the total number of Bronze packets that arrive on the system increases. When the Packet Group Proportion for the Bronze group drops to about 5%, its drop rate increases significantly due to starvation of the queue (the last set of bars in Figure 8). The other scheduling policies in which the queues are processed more equally do not exhibit such a behaviour.

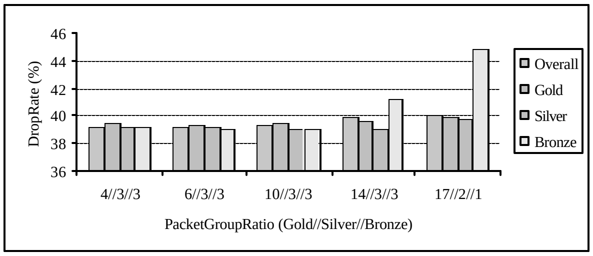

Figure 8. The Effect of Packet Group Ratio on the Drop Rate for Fixed-Priority Scheduling

The experimental results show that the scheduling policies have a small effect on the overall drop rate when integrated with the Default-Link dropping policy. A similar behaviour is observed when the scheduling policies are used in conjunction with the Dynamic-Link dropping policy [19]. This is because all the packet groups share the overall bandwidth. However, with a Fixed-Link dropping policy, the drop rate achieved by the Bronze group can become large at high arrival rates when the Packet Group Ratio for Bronze is small in comparison to the other two groups.

4.2.3 The effect of packet processing times

An investigation of the relationship between packet processing time, Ptime, with the dropping rates achieved by the different scheduling policies is presented in [19]. For conservation of space, only the conclusions are summarized in this paper. As the packet processing time is increased, the time each packet stays in the router becomes longer, and the packet queuing time increases as well. This means that with a higher Ptime the buffer becomes full earlier and more packets are dropped. Thus, in most cases with an increase in Ptime, the proportion of packets dropped at the processing queues become larger in comparison to the packets dropped due to a lack of bandwidth.

4.3 The Effect of Scheduling Policies on the Mean Roundtrip Time

The two following sub-sections present the experimental results that demonstrate the effect of the different scheduling policies on the mean round trip when various CPU shares are assigned to the packet group. Section 4.3.1 corresponds to a system in which the Share of each group is the same as its Packet Group Proportion whereas in Section 4.3.2 the Shares of the groups are different from their Packet Group Proportions. A larger difference in the drop rates achieved by the scheduling policies was observed with the Fixed-Link dropping policy (see Section 4.2). This section presents an investigation of the mean response times achieved by the scheduling policies in the context of the Fixed-Link dropping policy. The analysis of mean response times in the context of other packet dropping policies forms an important direction for further research.

4.3.1 Share Equal to Packet Group Proportion

The input data for the experiment that captures the results of the experiment when the CPU shares allocated to the groups are equal to the Group Proportions are shown in Table 4. Figure 9 presents the results of the experiment.

Table 4. Input Data for the Experiment in which Packet Group Proportion is Equal to Share

	Gold	Silver	Bronze
Packet Group Proportion (%)	40	30	30
Link-Share (%)	40	30	30
Share (%)	40	30	30
Packet arrival rate (P/s)	633		

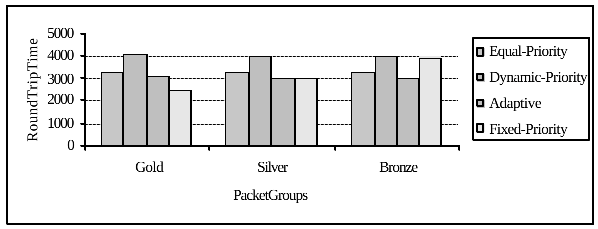

Figure 9. Round Trip Times with Packet Group Proportion Equal to Share

As illustrated in Figure 9, the Fixed-Priority scheduling policy produces the sharpest distinction among the roundtrip times achieved by the different groups. The EF or the Gold group has the shortest roundtrip time followed by the Silver group and the Bronze group. For the other policies the mean roundtrip Times are almost the same across the three packet groups. For any packet group, the highest roundtrip time is produced by the Dynamic-Priority scheduling policy. One of the contributing factors to its performance is the higher computational overhead resulting from running the scheduler frequently.

4.3.2 Share Unequal to Packet Group Proportion

When the CPU Share of a group is not the same as its Packet Group Proportion, the scheduling policies perform very differently. The input data for this experiment is presented in Table 5. The results for the Gold group are presented in Figure 10.Note that the decimal part of the packet group proportions indicated on the X-axis of Figure 10 are truncated – the actual values used sum up to 100%

Table 5. Input Data for the Experiment in which Packet Group Proportion is Unequal to Share

	Gold	Silver	Bronze
Packet Group Proportion (%)	40	30	30
Link Distribution (%)	40	30	30
Share (%)	varies	varies	varies
Packet arrival rate (P/s)	633		

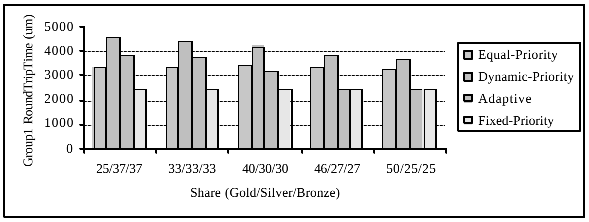

Figure 10. Round Trip Times for the Gold Class with Packet Group Proportion Unequal to Share

Both the Dynamic-Priority and Adaptive scheduling policies distribute the CPU resources based on the Share of each packet group. If a packet group has a higher Share compared to its Packet Group Proportion, then most of its packets will be processed at a higher priority, thus shortening the roundtrip time. In contrast, if a packet group has a lower Share than its Packet Group Proportion; its packets will most likely be processed at a lower priority, so the roundtrip time is longer. The Share[i] vector thus provides the ability to tune the performance of these strategies. As shown in Figure 10 the mean roundtrip time for the Gold group achieved by the Dynamic-Priority and Adaptive strategies improve as Share [1] is increased. Due to its high overhead, the performance of the Dynamic-Priority strategy is the worst. But the Adaptive scheduling policy demonstrates a slightly better performance compared to Fixed Priority when a 50/25/25 Share is used.

The Adaptive scheduling policy is further analyzed in an experiment. The input data is presented in Table 6.

Table 6. Input Data for the Experiment with the Adaptive Scheduler

	Gold	Silver	Bronze
Packet Group Proportion (%)	40	30	30
Link-Share (%)	Equal to the Share		
Share (%)	varies	varies	varies
Packet arrival rate (P/s)	811		

The results in Figure 11 show the variation of the roundtrip time when the Share is changed. Figure 11 demonstrates how Share[i] can be used to tune the differential round trip times for the different groups. From left to right in the graph of Figure 11, the Share for Silver and Bronze are increased, and the differences among the mean roundtrip times achieved by the different groups are reduced.

Based on the results presented in Section 4.3.1 and Section 4.3.2, we observe that out of all the different schedulers investigated, the Adaptive scheduler is flexible and if tuned appropriately can achieve the shortest mean roundtrip time. It can also effectively control the starvation issue that can occur when Fixed-Priority-based scheduling is used.

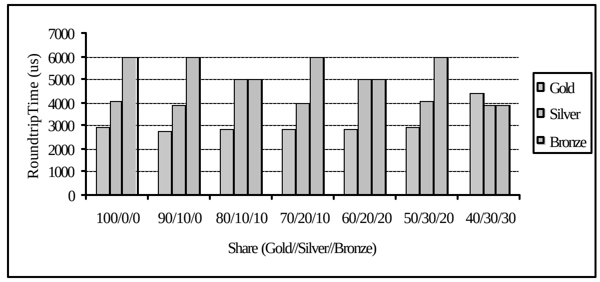

Figure 11. Group Roundtrip Times for the Adaptive Scheduler

Due to frequent scheduling, the Dynamic-Priority scheduling policy results in the longest mean roundtrip time. The Equal Priority scheduler produces approximately the same roundtrip times for the different packet groups and the factor dominating its performance at a given arrival rate is the Packet Group Ratio. The main concern of the Fixed-Priority scheduling policy is producing the lowest roundtrip times for the higher priority groups that some times lead to the starvation of the Bronze packets.

5. SUMMARY AND CONCLUSIONS

In this paper, we have proposed several different dropping and scheduling policies for routers. A number of experiments were conducted to investigate their performance. Insights gained into the system behaviour and performance are presented in this section.

The characteristics of the different dropping policies are summarized first:

- Default-Link does not provide any differentiated service for clients. The router starts to drop packets when the rate of outgoing packets exceeds the link capacity. As the network link is freely shared by all the groups of packets, an increase in packet arrival rate for each packet group not only affects its own group drop rate but also the drop rates of the other two groups.
- Fixed-Link can provide different classes of service by precisely splitting the network link capacity for the different packet groups. Because the bandwidth allocated to each group of packets is fixed, the drop rate of one packet group is strictly related to its own packet arrival rate. To increase the level of service for one group of packets, we can simply assign more bandwidth to it. However, over-provisioning of bandwidth to a group may result in bandwidth wastage resulting in large drop rates for other groups.
- The performance results presented in this paper (see Figure 3 for example) show that the differentiated service provided by Fixed-Link is achieved at a cost of a higher overall drop rate in comparison to the Default-Link policy. This is because the higher link shares divested to a higher priority group may not be utilized fully whereas a lower priority group may be subject to

starvation leading to a higher drop rate. The differences in the drop rate between the two policies decrease at higher arrival rates (see Figure 3).

- Dynamic-Link provides the flexibility to split the network resource more accurately to fit each group's needs. Because only a small portion of the resource is reserved for the higher priority groups, bandwidth wastage is effectively controlled. Adjusting the reserved portion for a high priority group can provide the requested level of service to the group. This ability to control the degree of differentiated service makes Dynamic-Link the most attractive packet dropping policies investigated in this paper.

The impact of scheduling policies on performance is discussed. For a given set of parameters, the overall drop rate is approximately the same for all the scheduling policies. The Fixed-Priority policy provides differentiated drop rates but can give rise to a large drop rate for lower priority groups.

Distributing the CPU resource to three groups in different ways can result in different routing delays for each group of packets. These differences are summarized below:

- The Equal-Priority (FCFS) scheduling policy does not differentiate among packet groups. All packet groups are treated the same, so they have the same mean roundtrip times.
- Similar to Priority Scheduling, the Fixed Priority scheduling policy gives rise to the shortest delay for the highest priority group of packets. However, this could cause starvation of the lowest priority packet group and increase its group drop rate.
- Both Dynamic-Priority and Adaptive scheduling policies can give each packet group a reasonable delay by adjusting the priorities of different queues that handle each packet group. But the Adaptive scheduling policy produces the shortest overall delay, as it results in a smaller scheduling overhead.
- Both the Fixed-Priority and the Adaptive scheduling policies can provide differentiated roundtrip times to different classes. The degree can be controlled by tuning the Share vector associated with the groups.

Overall, the results indicate that different classes of service for packet groups can be effectively achieved by a combination of the Dynamic-Link dropping policy and the Adaptive scheduling policy. The Fixed-Link dropping policy could also be used because of its simplicity, but bandwidth wastage should be avoided by carefully monitoring and managing the network traffic demands. Evaluating the performance of these strategies and comparing with WFQ in a real environment with a large number of routers forms an interesting direction for further research.

ACKNOWLEDGMENTS

This work was partially supported by Nortel Networks.

REFERENCES

[1] Blake, S., et al.., "An Architecture for Differentiated Services", *IETF RFC 2475*, 1998.

[2] Gien, M. "From Operating Systems to Cooperative Operating Environments", *Chorus Sys.*, Oct 1992, http: //www.jaluna.com/developer/papers/CS-TR-92-82.pdf.

[3] Sun Microsystems, "ChorusOS 4.0 Introduction", December 1999, http://docs.sun.com/db/doc/806-0610.

[4] Sun Microsystems, "ChorusOS 5.0 Application Developer's Guide", December 2001, http://docs.sun.com/db/doc/806-6894.

[5] Parekh, A.K., Gallager, R.G., "A generalized processor sharing approach to flow control in integrated service networks: the single node case", *IEEE/ACM Trans. on Networking*, 1 (3), 1993, pp. 344–357.

[6] Parekh, A.K., Gallager, R.G., "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case", *IEEE/ACM Trans. on Networking*, 1994, 2 (2), pp.137 – 150.

[7] Stiliadis, D., Varma, A., "Rate-proportional servers: a design methodology for fair queueing algorithms", *IEEE/ACM Trans. on Net.*, 1998, 6 (2), pp.164-174.

[8] Stiliadis, D., Varma, A., "Efficient fair queueing algorithms for packet-switched networks". *IEEE/ACM Trans. on Networking*, 1998, Vol.6, Iss.2, pp.175 – 185.

[9] Mao, J., Moh, W.M., Wei, B., "PQWRR scheduling algorithm in supporting of DiffServ", *ICC* 2001, pp.679-684.

[10] Wang, H., "Adaptive-weighted packet scheduling for premium service", *ICC* 2001, pp.1846 – 1850.

[11] Oottamakorn, C., Bushmitch, D., "A Diffserv measurement-based admission control utilizing effective envelopes and service curves", *ICC* 2001, pp.1187 – 1195.

[12] Bianchi, G., "Admission control over assured forwarding PHBs: a way to provide service accuracy in a DiffServ framework", *Global Conf.,* 2001, Vol. 4, pp.2561-2565.

[13] Kelly, F., "Notes on effective bandwidths", *Stochastic Networks: Theory and Applications*, Clarendon Press, Oxford, 1996, pp.141-168.

[14] Norros, I., "On the use of fractional Brownian motion in the theory of connectionless networks", *J. on Selected Areas in Comm.*, 1995, 13 (6), pp. 953-962.

[15] Elwalid, A., Mitra, D., "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks", *IEEE/ACM Trans. on Networking*, 1993, Vol. 1, No.3, pp. 329-343.

[16] Chang, C.S., Thomas, J.A., "Effective bandwidth in high speed digital networks", *J. on Selected Areas in Comm.*, 1995, 13 (6), pp. 1091-1010.

[17] Labrador, M.A. and Banerjee, S., "Packet dropping policies for ATM and IP Networks", *IEEE Communications* Survey, 2 (3), 1999, pp. 1-13.

[18] Nortel Networks, "CGNet: A users guide & designers manual", June 2001.

[19] Wang, Q., *Fair-Share Scheduler for Computing and Telecommunication Systems*, Master Thesis, School of Comp. Sci., Carleton U., Ottawa, Canada, Dec 2003.