The University of Kansas

Information and Telecommunication Technology Center

Technical Report

Exploring the Impact of Differentiated Services on Carrier Networks

Ajay Uggirala and Victor Frost

ITTC-FY2001-TR-18836-04

July 2000

Project Sponsor:
Sprint Corporation
Technical Planning & Integration

Table of Contents

List of Figures

List of Tables

Abstract

In this thesis, the impact of differentiated services on carrier networks is explored. Two separate studies were performed to understand different issues related to deployment of differentiated services. In the overbooking study the impact of overloading the service providers link on the end-to-end characteristics of different classes is explored. The impact of assigning the same class to TCP and UDP flows and queuing flows with different traffic characteristics in the same queue are also explored. It was found that the overbooking did not effect higher priority flows, particularly UDP flows. When UDP and TCP flows were assigned same class TCP flows were treated unfairly, it was found that TCP flows could be protected to certain extent by assigning higher drop precedence to UDP flows. It was also found that the performance depended on the length of the packets. In the other study, the impact on the performance of the number of DiffServ Classes was studied. Two architectures, a two-queue and a three-queue system are considered for this study and their relative performance is evaluated. In the two-queue model the better than best effort (BBE) (or assured) packets and best effort (BE) packets are queued in the same queue, the BBE packets are treated better than BE by using Random Early Drop with In and Out (RIO). In the three-queue model the two flows, Assured (BBE) and BE, are queued in separate queues. It was found that the performance of the three-queue system is better than the two-queue system. The impact of scheduling in the customer edge was also studied, and both First In First Out (FIFO) and Deficit Round Robin (DRR) schemes were evaluated. It was observed that service guarantees could be achieved by using DRR scheme instead of FIFO scheme.

Chapter 1

Introduction

Traditionally, network service providers (both enterprise and traditional Internet Service Providers (ISP)) provide all customers with the same level of performance (best-effort service). However, in recent years, increased usage of the Internet has resulted in a demand for voice and other, mission critical applications. At the same time, new applications have emerged which demand much improved service quality.

A "Service" defines some significant characteristics of packet transmission in one direction across a set of one or more paths within a network. These characteristics may be specified in quantitative or statistical terms of throughput, delay, jitter, and/or loss, or may otherwise be specified in terms of some relative priority of access to network resources. Differentiated Services (DiffServ) are intended to provide scalable service discrimination in the Internet without the need for Per-flow State and signaling at every hop [1].

A wide range of services can be provided by a combination of: [2]

- setting bits in the TOS octet at network edges and administrative boundaries,

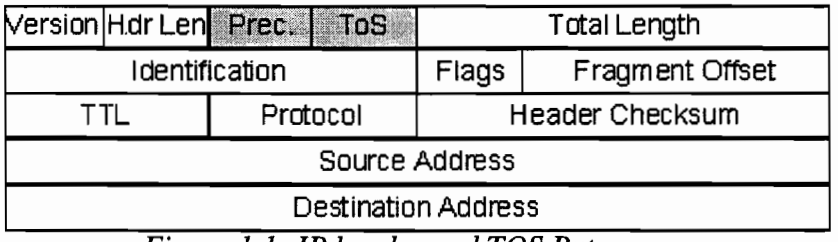


Figure 1.1: IP header and TOS Byte.

- using those bits to determine how packets are treated by the routers inside the network, and
- conditioning the marked packets at network boundaries in accordance with the requirements of each service.

The TOS octet in the IP header, if used for providing differentiated services is called Differentiated Service (DS) byte. The first six bits of DS byte are used as differentiated services code-point (DSCP) to select the Per-Hop Behavior PHB a packet experiences at each node. The last two bits are currently unused (CU).

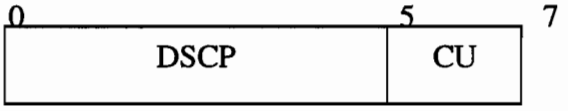


Figure 1.2: DS Byte.

1.1 Services and PHBs

The IETF (Internet Engineering Task Force) has defined a service and per-hop-behavior (PHB) as follows ([1] [2]). A service is the overall treatment of a defined subset of a customer's traffic within a DS domain or end-to-end. The Service Level Agreement (SLA) between the customer and service provider specifies the customer's forwarding service.

A PHB is an externally observable forwarding behavior applied at a DS compliant node to a DS behavior aggregate. A PHB can also be termed as service provided to a traffic aggregate. The DSCP is used to mark packets to select a particular PHB.

The IETF has standardized two type of PHBs, Expedited Forwarding (EF) and Assured Forwarding (AF). Both EF and AF are needed in the network so as to provide service according to the various customer demands. The implementation of various components depends on the service definition.

1) Premium/ Expedited Service: - This is a kind of service which will provide the customer with minimum or no queuing delay, there is always a minimal amount of bandwidth that the user is guaranteed to receive. This kind of service is particularly useful for applications like Voice and Virtual Private Networks (VPNs).

2) Assured Service: - The customer with this service might receive the same delay characteristics as best effort service but during the periods of congestion the service they are going to receive mostly remains unaffected. Different classes can be provided within this service, which have different levels of transparencies with respect to the congestion in the network. Assured Forwarding (AF) PHB group provides forwarding of IP packets in N independent AF classes. Within each AF class, an IP packet is assigned one of M different levels of drop precedence. IETF has suggested four different classes with each class having three levels of drop precedence.

3) Best Effort Service (BE): - Same service as the packets are receiving in the present Internet. Still will remain major type of traffic flowing through the network.

1.2 DiffServ Mechanisms

To realize these services, the routers, where the packets are served should be provisioned with new components (like classifiers and conditioners), the queue management and the scheduling schemes of those queues should changed (this is where the PHB is realized). PHBs are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node to behavior aggregates. The components in the nodes have to be monitored and configured according to the requirements. In other words the bandwidth allocation has to be changed depending on the requirement. These mechanisms are explained in detail in Chapter 2.

1.3 Motivation

The impact of Differentiated services on carrier networks is not well known. The deployment of DiffServ into the networks is still in its preliminary stages. There are

numerous questions that need to be answered like identification of functional elements and their relative merits, the interaction of TCP and UDP, and interaction of different classes.

In this thesis we identified some these components and the target architectures and then answered some these questions that will be useful for ISP's to deploy Differentiated Services.

1.4 Executive Summary

1.4.1 Research Objective

The main objective was to address DiffServ issues prior to the possible delployment of the technology into the network. Questions addressed here include the impact of overbooking the network the resources on the end-to-end performance of different service classes. The interaction between TCP and UDP flows, and the traffic flows with different packet lengths. Two different schemes of queuing customer traffic were evaluated. A two-queue model, which is extensively used to implement Better than Best Effort service was compared to a three-queue model. We also evaluated the impact of the scheduling scheme used in the customer edge.

1.4.2 Results

Differentiated services can used to provide scalable service guarantees to flows by changing some of the network components. The studies conducted here revealed that service guarantees could be provided to high priority flows. It was also learned that the UDP flows were responsible to cause unfairness to TCP flows, when they are queued in the same queue particularly if the queue is overloaded. So it is not advisable to overbook the high priority queue. It was found that TCP flows could be protected to certain extent by assigning higher drop precedence to the UDP flows. The overbooking factor was found to have an effect on the lower service classes while the higher service classes were mostly protected. It was found that its hard to realize service guarantees with TCP flows.

The performance of a two-queue model and three-model was compared. It was found that the three-queue model was able perform better than the two-queue model. The impact of the scheduler in the customer edge was also studied and it was found that harder service guarantees (guaranteed bandwidth and delays) could be provided by bandwidth allocation in the customer edge, especially for overloaded cases. It was again found that queuing UDP and TCP flows in the same queue resulted in unfairness to TCP flows. It was also found that the parameters used to provide service for TCP flows should be configured carefully, so that TCP fragmentation and bursts are taken into account.

1.5 Organization of this Thesis

The rest of the thesis is organized in the following manner. Chapter 2 discusses the background information, this includes a brief description of the existing DiffServ architectures and design issues (i.e., different functional elements and mechanisms) to provide differentiated services. Chapter 3 discusses the work related the studies performed in this thesis. Chapter 4 presents the overbooking study, which explores the impact of overloaded a backbone link, it also addresses the issues such as assigning same class to UDP and TCP flows and queuing packets of different lengths in the same. Chapter 5 discusses the set of simulations conducted to determine the relative performance of a two-queue and a three-queue model, it also addresses the impact of scheduling in the customer edge. Chapter 6 contains the conclusion and future work.

Chapter 2

Background Information and Design

This chapter provides background information on how to realize end-to-end Differentiated Services in a network. Various functional elements required to provide Differentiated Services are discussed. To carry out the proposed studies a network simulator called OPNET [3] is used. The design issues and the algorithms of functional elements that are implemented in OPNET to make it DiffServ capable are also discussed.

2.1 Architecture

To provide an end-to-end Differentiated Service, it should be ensured that all the domains, through which the traffic might flow, are capable of providing differentiated services. A domain has well-defined boundary with (boundary) nodes through which traffic may enter (Ingress Node) or may leave (Egress Node) the domain. A Differentiated Service (DS) capable domain consists of a set of nodes, which operate with a common service provisioning policy and set of PHB groups implemented on each node. Figure 2.1 shows a cascade of Domains through which traffic might flow to reach its

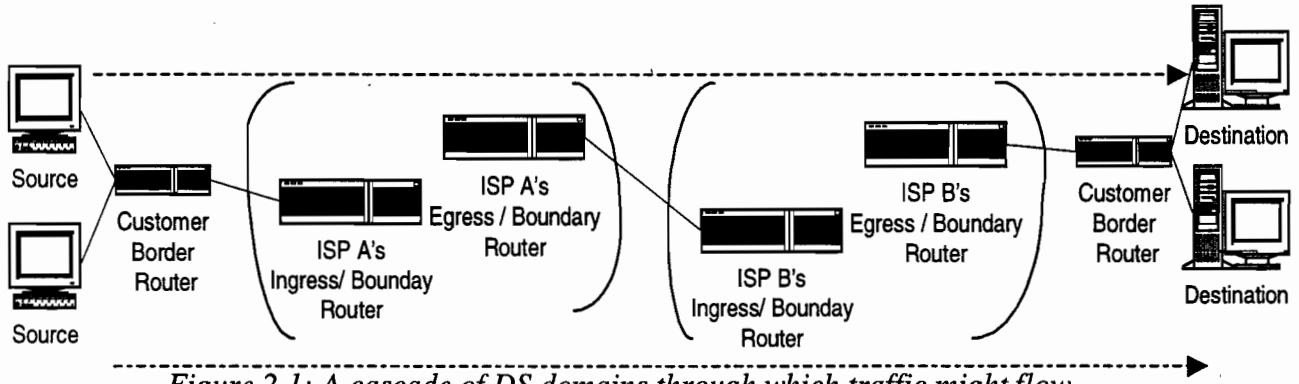


Figure 2.1: A cascade of DS domains through which traffic might flow.

A DS domain has a well-defined boundary consisting of DS boundary nodes, which classify and possibly condition ingress traffic to ensure that the traffic entering the domain conforms to the Traffic Conditioning Agreement (TCA) between its domain and the sending domain. Similarly the boundary node might also be responsible for ensuring

that traffic going out from the domain through it also conforms to the TCA between the domains. To provide such functionality the boundary node or border router needs Classifiers and Traffic Conditioners. The boundary node may also need to check that the incoming packets are appropriately marked to select a Per-Hop-Behavior (PHB) supported in the domain. DS boundary nodes act both as a DS ingress node and as a DS egress node for different directions of traffic.

DS boundary nodes interconnect the DS domain to other DS or non-DS-capable domains, while DS interior nodes only connect to other DS interior or boundary nodes within the same DS domain. The boundary nodes (border routers) in the service provider's domain have responsibilities similar to those described above. The nodes (interior routers) in the core of the domain simply forward packets according to the PHB associated with the marked value of the packet. The marked value of the packet is the DS Byte in the IP header, which was described, in the previous chapter.

Differentiated services are extended across a DS domain boundary by establishing a Service Level Agreement (SLA) between an upstream network and a downstream DS domain. The SLA may specify packet classification and re-marking rules and may also specify traffic profiles and actions to traffic streams, which are in- or out-of-profile. The TCA between the domains is derived (explicitly or implicitly) from this SLA.

A traffic profile specifies the temporal properties of a traffic stream selected by a classifier for example a stream/flow with rate - r and burst - b. It provides rules for determining whether a particular packet is in-profile or out-of-profile. Different conditioning actions may be applied to the in-profile packets and out-of-profile packets. Classifiers are used to steer packets matching some specified rule to an element of a traffic conditioner for further processing.

2.2 Classifiers

To differentiate the traffic flows all the nodes need a classifier [4]. The classifier should be able to classify packets into different output flows based on the some portion of the header. Classifiers can be of different types depending on the basis of their classification. The two types of classifiers which will be needed to provide differentiated services are the Behavior Aggregate (BA) classifier, which classifies packets based on DSCP and the Multi-Field (MF) classifier, which classifies packets based on some portion or a combination of some portions of the IP header such as the source address, destination address, source port etc.

The type of classifier that has to be used in a node depends mainly on the position of the node. All the leaf routers or the nodes connected directly to the client need to have a MF classifier, since they might have to check the traffic that is being sent by clients for admission control, i.e., to see to that the traffic flows are conforming to the SLAs. This might be necessary to prevent illegal use of service by the hosts or for accounting purposes. The boundary nodes may also have a Multi-Field classifier in case of packets being classified based on the destination address or some other portion of the IP header. The interior nodes usually have a behavior aggregate classifier.

Figure 2.2 shows a classifier with two filters and three outputs. A filter consists of a set of conditions on the component values of a classification key. The classification key is one or a combination of the header fields. Filter1 and Filter2 specify exact-match conditions on the value of classification key. The third filter is a wildcard default filter which matches every packet, but which is only selected in the event that no other more specific filter matches.

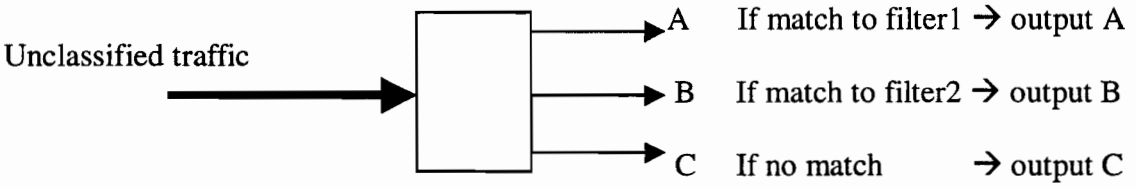


Figure 2-2: Classifier

2.2.1 Behavior Aggregate Classifier

The BA classifier is one of the most essential elements. These classifiers are *parameterized* by a set of *6-bit DSCP* (value, mask) pair; the two filters of BA classifier will be having DSCP as their classification key. Table 2.1 shows an example of a BA classifier with specific values assigned the classification key.

	Filter1	**Filter2**
Type	BA	BA
Value	111000	110000
Mask	111111	111111

Table 2.1 [4]: Filters used for a BA classifier

The filters can be configured to different DSCP based on the PHB behavior that are being offered in the domain and which DSCP is to be mapped to which PHB.

2.2.2 Multi-Field Classifier

A Multi-Field classifier has to classify packets based on some portion of the header or combination of the header fields; it is *parameterized based on the header fields* that the packets are classified. In this case the classification key for the filters will be a single or a combination of header fields. If the classification key were DSCP then the classifier would become a BA classifier. Table 2.2 shows values of filters when the classifier shown in Figure 2.2 acts as MF classifier. The packets are classified based on combination of five fields in the header namely source address, destination address, source port, destination address and the protocol used.

Classifiers are simple and easy to implement. MF classifiers, which examine the contents of transport layer header fields, may incorrectly classify packet fragments. The policy that has to be applied to the packet fragments has not been decided yet by IETF. BA classifier was implemented in OPNET for our studies.

	Filter1	Filter2
Type	MF	MF
IP src. Addr. Value	172.31.8.0	172.31.9.0
Ipdest. Addr. Value	0	0
IP src. Port value	0	0
IP dest. Port value	5003	5003
IP protocol value	0	0
IP src. Addr. Mask	225.225.225.0	225.225.225.0
IP dest. Addr mask	0.0.0.0	0.0.0.0
IP src. Port mask	0	0
IP dest. Port mask	0xFFFF	0Xffff
IP protocol mask	0	0

Table 2-2 [4]: Filters used for MF classifier

2.3 Traffic Conditioners

Traffic Conditioners are the most complex and important components required for the realization of Differentiated Services. The functionality of the traffic conditioner may vary depending on the position of the node, the service or the PHB. For example a particular type of service (Expedited) may need a shaper whereas another kind of service (Assured) may not need a shaper.

The traffic flow that is coming into the conditioner is parameterized by its traffic profile. A traffic profile [1] specifies the temporal properties of a traffic stream selected by a classifier. It provides rules for determining whether a particular packet is in-profile or out-of-profile. For example, a profile based on a token bucket may look like:

codepoint = X, use token-bucket with rate r and burst size b.

The above profile indicates that all packets marked with DS codepoint. X should be measured against a token bucket meter with rate r and burst size b. In this example out-of-profile packets are those packets in the traffic stream, which arrive when insufficient tokens are available in the bucket. The concept of in and out-of-profile can be extended to more than two levels, e.g., multiple levels of conformance with a profile may be defined and enforced. A traffic profile is an optional component of a TCA and its use is dependent on the specifics of the service offering and the domain's service provisioning policy.

A traffic conditioner [5] in general may look as shown below: -

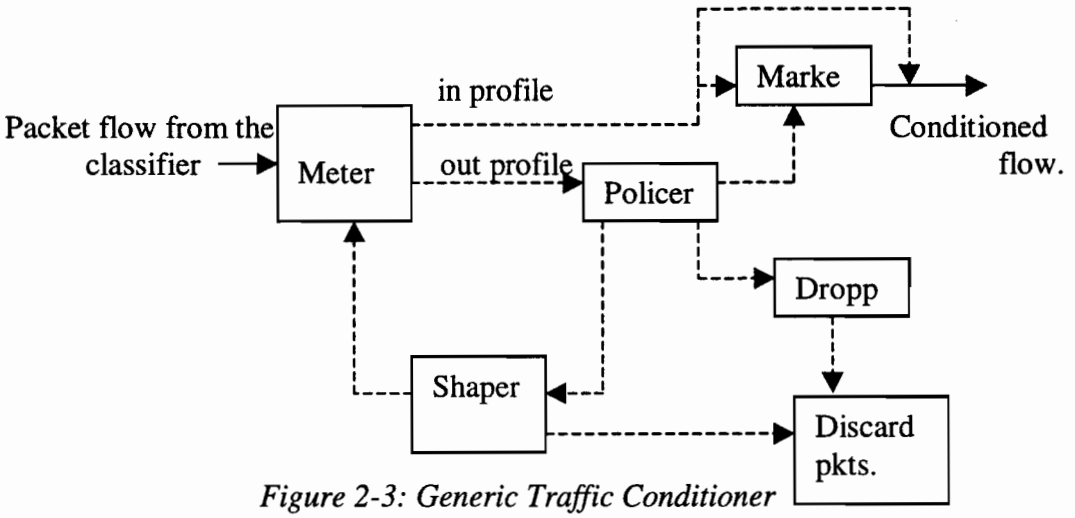


Figure 2-3: Generic Traffic Conditioner

As shown in the above diagram a traffic conditioner may contain a meter, policer, marker, shaper and dropper.

- A traffic flow is directed to a particular traffic conditioner by the classifier. The traffic conditioner will be configured to condition the traffic flow of a *specific profile.*
- A *meter* is used to measure the traffic flow against a traffic profile. Based on the result of the measurement of the flow it is directed out as in or out of profile.
- Different conditioning actions may be applied to the in-profile packets and out-of-profile packets, or different accounting actions may be triggered.

- In-profile packets may be allowed to enter the DS domain or passed on to the next hop without further conditioning; or, alternatively, their DS codepoint may be changed by passing the flow through a *marker.* The latter happens when the DS codepoint is set to a non-Default value for the first time, or when the packets enter a DS domain that uses a different PHB group or codepoint to PHB mapping policy for this traffic stream.
- Out-of-profile packets may be queued until they are in-profile (*shaped*), discarded (*dropped*), marked with a new codepoint (*re-marked*), or forwarded unchanged while triggering some accounting procedure.
- Out-of-profile packets may be mapped to one or more behavior aggregates that are "inferior" in some dimension of forwarding performance to the BA into which in-profile packets are mapped.

Now the various components and their associated parameters involved in a traffic conditioner shall be discussed,

2.3.1 Meters

There are different kinds of meters that can be used and implemented. The three most prominent ones are Average Rate Meter [6], Exponential Weighted Moving Average Meters [4] and Token Bucket Meters [14].

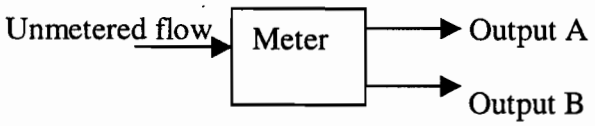


Figure 2.4: Meter

If the flow conforms to the profile send it to output A, if not then send it to output B. The meters can also send the packets as a single packet stream to a marker, where the packets are marked based on whether they are in profile or out profile. An example of this is three color marker, which will be described later.

- An **Average Rate Meter** measure the average rate at which the packets are submitted to it over a specified average time, if it exceeds a certain value than the packet is noted as out of profile. This meter is parameterized by the *average rate of packet arrivals.*

  The average rate estimator can be implemented as a time sliding window (TSW) algorithm as suggested in [8]. TSW estimates the rate at which the packets are arriving, and decays, or forgets the past history over time. TSW maintains three state variables*: win_length,* which is given in units of time, *avg_rate,* is the rate estimate upon each packet arrival and *t_last,* is the time of last packet arrival. As TSW estimates rate upon each packet arrival *avg_rate* and *t_last* are updated each time a packet arrives but *win_length* is pre-configured. The algorithm is given below,

  Initialization
  *win_length = a /*constant*/;*
  avg_rate = 0;
  t_last = 0;

  For each packet arrival
  *bytes_in_TSW = avg_rate * win_length;*
  new_bytes = bytes_in_TSW + pkt_size;
  avg_rate = new_bytes /(now - t_last + win_length);
  t_last = now;

  where *now* is the time of current packet arrival and *pkt_size* is the packet size of the arriving packet. The window length determines the worth of past history the algorithm remembers, or in other words the weight of the past against the present.

  After rate estimation the packet flow can be directed to one of the two outputs as follows,
  if (avg_rate <= target_rate)

*send packet to output A; /*in profile*/*

else

*send packet to output B; /*out profile*/*

- **Exponential Weighted Moving Average (EWMA)** is parameterized as follows; *gain* controls the time constant (e.g. frequency response) of a low pass filter. *actual_rate (n)* and *avg_rate(n)* measure the number of incoming bytes in a small fixed sampling interval, *delta.* If the packet that arrives pushes the *avg_rate* over a predefined rate *target_rate* it is deemed as out of profile. For example a flow can be metered against a profile with *gain = 1/16, delta= 10μs* and *avg_rate = 200 kbps.* The algorithm is given below,

*avg_rate (n+1) = (1-gain) * avg_rate(n) + gain * actual(n+1)*

t (n+1) = t(n) + delta

Now for an arriving packet at time t(m),

if (avg_rate (m) <= target_rate)

*send packet to output A; /*in profile*/*

else

*send packet to output B; /*out profile*/*

- **Token Bucket Meter** is the most important and used meter. The token bucket meters can be described by two parameters a token *rate r* and a *bucket depth B*. In this meter byte tokens keep on accumulating at a *rate r* until a maximum number of tokens is reached, which is the burst size. When a packet arrives and there are more tokens in the bucket than the number of bytes in the packet then the packets are considered to be IN profile. When the tokens present are less than the number of bytes in the packet than the packet is considered to be out of profile.

Note the meter can simply send the packet stream and the result of the metering to marker where the packet can be marked according to meter result. A meter followed by a marker can be viewed as the simplest Traffic Conditioner. This type of Traffic Conditioner is particularly useful in implementation of assured service (to remark the packets to a lower service category if they are out of profile) and is also called as *profile meter*. A token bucket is extensively used as a meter in a profile meter; this will be described later. Studies conducted in [8] revealed that token bucket is superior to rest of the meters, so we implemented token bucket algorithm in OPNET to meter the traffic flows.

2.3.2 Markers

A marker simply set the DSCP in the IP header to a certain value. They are used at a number of places, the client/leaf router can mark the packets going out with a particular DSCP to give them a particular PHB, in border routers to change the DSCP to a value which is recognized in that domain and in a router to change the DSCP of non conforming packets so as to change their drop precedence or class. The *six bit DSCP* parameterizes a Marker. Markers can be intelligent or dumb i.e. they can simply mark all the packets coming in with a particular DSCP or they can mark depending on some parameter as the *result of the metering*.

2.3.3 Shaper

A shaper tries to bring packets, which are out of profile into conformance to a particular profile by delaying those packets. A shaper is similar to a meter however a shaper also contains a queue. Instead of marking the packets of non-conforming flow as *Out* of profile the shaper brings them to conformance by queuing them, if they are *In* profile they are simply forwarded at a configured rate.

When a packet comes to the shaper, the following happens:

If the queue is empty, the traffic shaper processes the arriving packet.

If possible (if tokens are available), the traffic shaper sends the packet.

else, the packet is placed in the queue.

If the queue is not empty, the packet is placed in the queue.

When there are packets in the queue, the traffic shaper removes the number of packets it can transmit from the queue every time interval.

The shaper delays packets until they conform to the profile. In case of a token bucket the packets may be delayed until enough tokens are available to send the packets. In the process of delaying packets if the queue gets filled, the arriving packets will be dropped. The queue depth should not be too large as it might increase the packet delay, which might cause the service level agreement to be violated and at the same time it should not be too small such that packets get dropped frequently. An optimal queue depth should be chosen through experimentation and based on the type of service level agreement. For example a customer might not want his packets be delayed more than certain amount of time but he might tolerate dropping of packets or customers can choose between there flows getting shaped or simply dropped if they are out of profile. A shaper is used in border routers to make the flow coming in and going out to conform to TCA and in clients and leaf router to make traffic conform to SLA.

Since a shaper is similar to meter but with a queue, the parameters of the shaper are similar to those of the meter in addition to the *depth of the queue*. The profile will usually be same as that of meter for example in the case of a token bucket the profile will usually have same avg_*rate* and *token depth* as that of the meter token bucket.

- **Droppers** simply discard the packets and no decision making is involved it simply drops all the arriving packets.

All these elements parameters are configured according to SLAs, TCAs and resources available.

Element	Types of Elements	Parameters
Classifier	Multi-Field	*Header fields*
	Behavior Aggregate	*DSCP (TOS byte)*
Meter	Average Rate	*Target_rate & win_length*
	EWMA	*Target_rate, gain & delta*
	Token Bucket	*Token_rate & buket_depth*
Shaper	Average Rate	*Queue_depth, target_rate & win_length*
	EWMA	*Queue_depth, target_rate, gain & delta*
	Token Bucket	*Queue_depth, token_rate & bucket_depth*
Marker	None	*DSCP & meter_result.*

Table 2-3: Summary of Traffic Classification and Conditioning Elements

2.4 Examples of Traffic Conditioners

2.4.1 Three Color Marker

Below a three-color marker [9] is described. It uses token bucket as the meter. It implements the traffic conditioning functionality, which is particularly useful for conditioning traffic of AF class and in the interior routers,

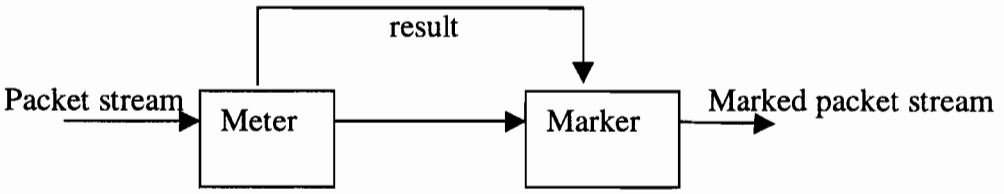


Figure 2.5: A logical representation of a Three-color marker.

The Meter operates in one of the two modes. In the Color-Blind mode, the Meter assumes that the packet stream coming in is uncolored. In the Color-Aware mode the Meter assumes the incoming packet stream is marked as green, yellow, or red. The color is coded in the DS field of the packet in a PHB specific manner i.e. each color represents a DSCP value.

The three-color marker is configured by setting its mode and by assigning values to three traffic parameters: a *Committed Information Rate (CIR), a Committed Burst Size (CBS),* and an *Excess Burst Size (EBS).* The *CIR* is measured in bytes of IP packets per second; it includes the IP header, but not the link specific headers. The *CBS* and the *EBS* are measured in bytes. The Meter has two token buckets, *C* and *E*, both of which share the common rate *CIR*. The maximum size of the token bucket *C* is *CBS* and the maximum size of the token bucket *E* is *EBS*.

The algorithm is as follows,

The token buckets *C* and *E* are initially (at time 0) full, i.e., the token count *Tc (0) = CBS* and the token count *Te (0) = EBS*. Thereafter, the token counts *Tc* and *Te* are updated *CIR* times per second as follows:

if Tc is less than CBS, Tc is incremented by one, else
if Te is less then EBS, Te is incremented by one, else
neither Tc nor Te is incremented.

When a packet of size *B* bytes arrives at time *t*, the following happens if the TCM is configured to operate in the *Color-Blind mode*:

if Tc(t)- B >= 0,
Tc is decremented by B and mark the packet as green,
else if Te(t)- B >= 0
Te is decremented by B and mark the packet as yellow,
else
the packet is red and neither Tc nor Te is decremented.

When a packet of size *B* bytes arrives at time t, the following happens if the TCM is configured to operate in the *Color-Aware mode*:

if the packet has been precolored as green and Tc (t)-B >= 0,

Tc is decremented by B and mark the packet as green,
else if the packet has been precolored as green or yellow and if Te (t)-B >= 0,
Te is decremented by B and mark the packet as yellow,
else
the packet is red and neither Tc nor Te is decremented.

Note that according to the above rules, the marking of a packet with a given color requires that there should be enough tokens of that color to accommodate the entire packet. Other marking policies can also be possible. It can be seen that the volume of green packets is never smaller than what has been determined by the CIR and CBS, i.e., tokens of a given color are always spent on packets of that color, so a deterministic behavior for packet flow of that particular color can be guaranteed. In case this traffic conditioner is used to condition AF PHB then *the colors can be coded as the drop precedence of the packet.*

The Two Rate Three-Color Marker (trTCM) [10] meters a packet stream and marks its packets green, yellow, or red. A packet is marked red if it exceeds the Peak Information Rate (PIR). Otherwise it is marked either yellow or green depending on whether it exceeds or doesn't exceed the Committed Information Rate (CIR). The trTCM is useful, when it is required to enforce two rates on a packet stream. We have implemented the Two Rate Three-Color Marker in OPNET for our studies.

Traffic conditioners differ based on the type of packets they are conditioning. Traffic conditioners that may be used to condition Expedited and Assured PHB packets are described below respectively.

2.4.2 Traffic Conditioner for EF Packet Flow

A traffic conditioner used to condition Expedited/Premium packet flow may look as shown in the following diagram,

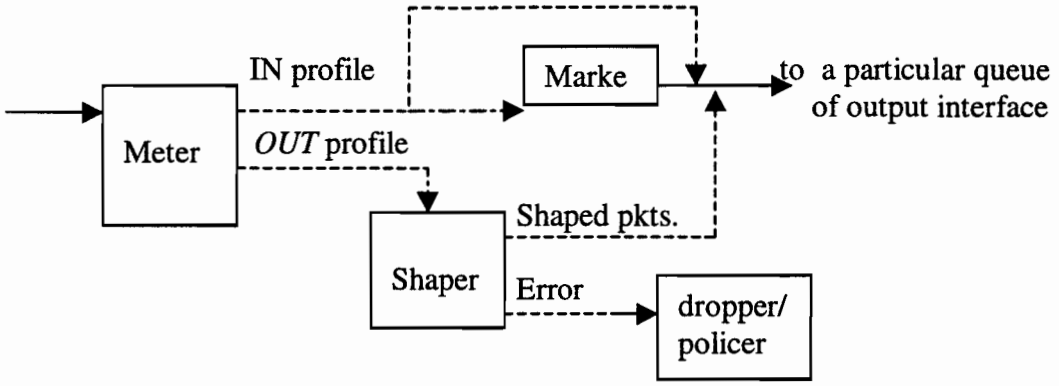


Figure 2.6: A traffic conditioner for EF packets.

The meter checks whether the packet flow is within the specified profile or not. The *IN* profile packets are directly sent to the output interface queue. The *OUT* of profile packets are sent to a shaper where they are shaped, the shaper has a simple FIFO queue where the incoming packets are delayed in order to bring them to conformance with the specified profile. If this queue is filled the packets are dropped (unshaped). The size of this queue should be optimal since the only place where premium packets might experience delay is here. The marker is optional it is used in the ingress router of a domain, if that domain uses different DSCP to PHB mapping then the marker is used to remark the DSCP to a value that is understood in the domain.

If both the meter and shaper use token bucket then the algorithm will be as follows,
Let the rate and bucket size of both the token buckets be r and b respectively i.e.
TRM (Token Rate for Meter) = r, BSM (Bucket Size for Meter) = b and
TRS (Token Rate for Shaper) = r, BSS (Bucket Size for Shaper) = b.

Initialization:

TRM = r; BSM = b;

TRS =r; BSS = b;

When a packet of size B bytes arrives,

If (BSM - B >= 0)

send the packet to the output interface;

BSM = BSM - B;

else

send the packet to the shaper;

When a packet comes to the shaper,

en-queue the packet; /* the queue is a FIFO queue */

if(B > available space in the shaper queue)

drop the packet;

For 1/r second, token buckets are updated and packets are sent from shaper as follows,

If (BSM < b)

BSM++;

If (BSS <b)

BSS ++;

/* BH is size of the packet in bytes at the head of the queue */

If (BSS - BH >= 0)

de-queue the packet from the head of the queue;/ and */*

send it to output interface;

BSS = BSS - BH;

2.4.3 Traffic Conditioner for AF Packet flow

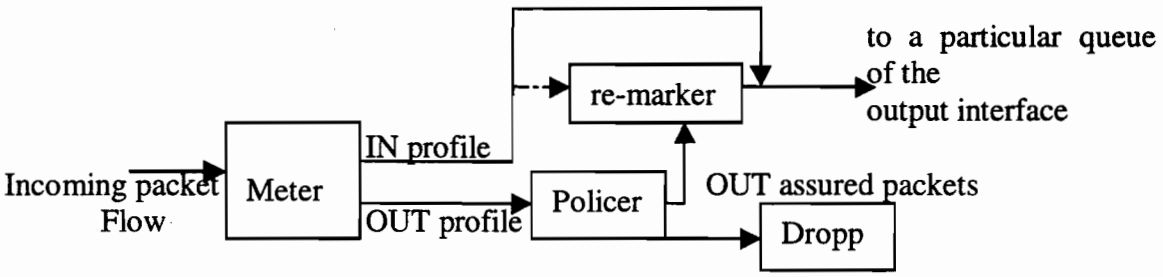


Figure 2.7: A traffic conditioner for AF packets

This traffic conditioner consists of a meter, which checks the incoming flow of Assured service packets from classifier. If the flow is *IN* profile then they are sent directly to the interface queue. The packets may be remarked if this traffic conditioner is in a border router and the domain implements a different DSCP to PHB mapping from its peering domain. The OUT of profile packets are sent to a policer. A policer is similar to a shaper but with zero queue size, it checks the arriving packets. The packets may be sent to the re-marker where they are remarked to higher drop precedence packets or they may be discarded based on some decision like if the flow is OUT of profile by a large amount.

If both the meter and policer use token bucket and if the incoming packets belong to a particular AF class (say class 1) with only two drop precedence AF_{11} and AF_{22} (for AF_{ij} i = class and j = drop precedence) then the algorithm will be as follows,

Let the rate and bucket size of the meter token bucket be
TRM (Token Rate for Meter) = r, BSM (Bucket Size for Meter) = b and
Let the rate and bucket size of the policer token bucket be
TRP (Token Rate for Policer) = 2r, BSP (Bucket Size for Policer) = 2b.

Initialization:
TRM = r; BSM = b;
TRP =2r; BSP =2b; / example values */*
For every 1/r second,

If (BSM < b)

BSM++;

For every 1/2r second,

If (BSP <2b)

BSP ++;

When a packet of size B bytes arrives, the meter does the following,

if (DSCP = AF_{11}) / then */*

    if (BSM - B >= 0)

    send the packet to the output interface;

    BSM = BSM - B;

    else

    send the packet to the policer;

else

    send the packet to the output interface;

When a packet of size B bytes comes to the policer it takes the following actions,

if (BSP - B >= 0)

send the packet to the marker;

else

drop the packet; /* or it can be marked as best effort */

When a packet comes to the marker,

change the DSCP to AF_{12};

In case of congestion the queue in the output interface to which the packets are sent showed drop AF_{12} before dropping AF_{11} packets. RIO (RED with In and Out, described later) with AF_{12} marked as *Out* and AF_{12} marked as *In* will be an ideal choice. To implement TC for an AF class with three-drop precedence, the meter and policer should have two token buckets each.

Type of TC	Parameters
Three Color Marker	*Mode, CIR, CBS and EBS*
EF TC	*TRM, BSM, TRS, BSS and queue_depth*
AF TC	*TRM, BSM, TRP, BSP,* AF_{i1} and AF_{i2}

Table 2-4: Parameters of different TC's described.

2.5 Queue Management

For Differentiated services the choice a queuing scheme depends on the type of packets that are buffered and how the packets should be discarded in times of congestion. For instance the Expedited / Premium service packets are usually buffered in simple FIFO while for buffering the assured service packets there are a variety of options like the RED (Random Early Drop) [11] and different versions of RED like WRED (Weighted RED) [7], and RIO (RED with In & Out). The choice also depends on how many classes are there in AF and whether we want the BE and AF to be buffered in the same queue. Below is a description of RED, WRED, and RIO,

2.5.1 Random Early Drop (RED)

RED takes advantage of the TCP's congestion control mechanism. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. A RED algorithm [13] works as follows,

- The average length of the queue is computed upon each packet arrival and is compared with the minimum and maximum thresholds.
- The packet is queued, dropped or marked based on the result of the comparison.
- Normal mode of operation: if the length of the queue is below the minimum threshold then the incoming packets are queued.
- Congestion avoidance mode of operation: If the average length of the queue exceeds the minimum threshold and is below the maximum threshold then the incoming packets are dropped or marked with a certain probability, the dropping probability

depends on the length of the queue, each drop helps in slowing down the sending rate of sources.

- Congestion control mode of operation: if the average length of the queue exceeds the maximum threshold then the incoming packets are either simply dropped or marked.

The following diagram shows the behavior of RED.

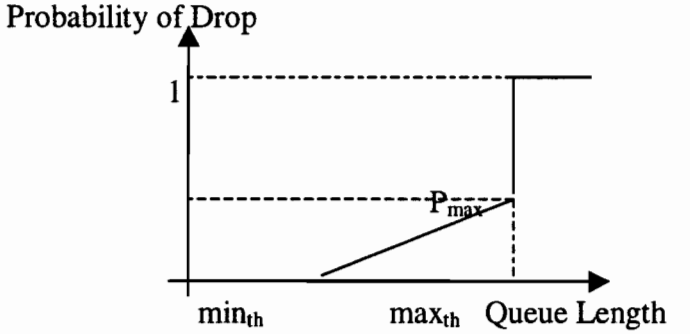


Figure 2.8: Behavior of RED queue.

The formula used for queue length calculation in RED is:

*average queue length = old average queue length * weight*
*+ current queue length * weight*

A RED gateway can be parameterized by the following parameters: *min_th, max_th,* P_{max} and *weight.* P_{max} is the maximum drop probability and *weight* is used in calculating the average length of the queue. The values suggested for *weight* and P_{max} are 0.002 and 0.02, for higher values of weight the average queue length closely tracks current queue length. The *max_th* is suggested to be double that of *min_th* to avoid global synchronization. Global synchronization manifests when multiple TCP hosts reduce their transmission rates in response to packet dropping, then increase their transmission rates once again when the congestion is reduced.

2.5.2 Weighted RED [WRED]

WRED [9] combines the capabilities of the RED algorithm with DSCP / IP Precedence to provide for preferential traffic handling of higher priority packets. WRED can selectively discard lower priority traffic when the interface begins to get congested

and provide differentiated performance characteristics for different classes of service. WRED generally drops packets selectively based on DSCP / IP Precedence. Packets with a higher DSCP / IP Precedence are less likely to be dropped than packets with a lower precedence. Thus, the higher the priority of a packet, the higher the probability that the packet will be delivered.

WRED works in the same way, as RED with the dropping of packets being dependent on their DSCP value, therefore it can be parameterized with the same parameters as RED viz. *min_th, max_th, n* (n is a exponential weight factor) in addition to *DSCP*. WRED was initially implemented in Cisco 7200 series routers. The formula used in queue length calculation is,

$$\textit{average queue length} = (\textit{old average queue length} * (1-2^{-n})) + (\textit{current queue length} * 2^{-n})$$

Where 2^{-n} is equivalent to weight in RED queue length calculations and n is exponential weight factor.

WRED can be used to realize an AF PHB group. The different drop precedence within an AF class can be assigned different values of IP precedence there by obtaining different treatment for each class.

2.5.3 RED with In and Out (RIO)

RIO [6] uses the same mechanism as in RED, but is configured with two set of parameters, one for *In* packets and one for *Out* packets. The two sets of parameters are as follows, *min_in, max_in,* P_{max_in}, $weight_{in}$ and *min_out, max_out,* P_{max_out}, $weight_{total}$. $weight_{in}$ and $weight_{total}$ are used to calculate the average queue length of In packets and the average total queue length of all (*In* and *Out*) packets, i.e., total queue length. Figure 2-9 shows how RIO works,

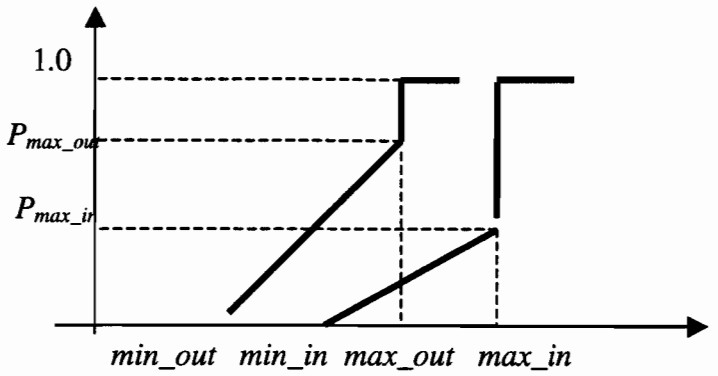


Figure 2.9: Behavior of RIO queue.

Upon each arrival RIO calculates the average total queue length (avg_total) and checks whether the packet is *In* or *Out*, if it is *In* then it calculates the average queue length of in packets (avg_in). The probability of dropping an *In* packet depends on *avg_in* and the probability of dropping an *Out* packet depends on *avg_total.*

RIO is more aggressive in dropping the *Out* packets. This discrimination against *Out* packets in RIO is created by carefully choosing the parameters as shown in the above diagram. In essence RIO drops Out packets when it detects congestion, and drops all out packets if the congestion persists. The In packets are dropped only as a last resort when the gateway is flooded mainly with the *In* packets, if the resources are well provisioned in the network then this is very unlikely to happen.

RIO can be used to easily manage two type classes like both AF and Best Effort packets can be sent into the same queue with AF packets marked as *In* packets and Best Effort packets marked as *Out* packets.

Queue	Parameters
RED	*min_th, max_th, weight* and *P_max*
WRED	*min_th, max_th, n* and *IP precedence values*
RIO	*min_in, max_in,* P_{max_in}, $weight_{in}$ and *min_out, max_out,* P_{max_out}, $weight_{total}$

RED mechanism can be used to buffer only packets belonging to a particular service class. While RIO and WRED can be used to buffer packets belonging to different

service classes. RIO has two levels of drop thresholds one for *In* and the other for *Out* packets. RIO usually distinguishes *In* and *Out* packets based on the DSCP value. So RIO can be used to buffer packets belonging to two PHBs like AF and BE. WRED provides separate thresholds for different DSCP / IP precedence. So it is useful to buffer packets belonging to a particular AF class with different drop precedence. All the queuing mechanisms described are based on RED algorithm with different dropping policies.

2.6 Scheduling

The different scheduling methods that can be used to realize differentiated services are described below.

2.6.1 Strict Priority Scheduling (SPS)

The incoming packets are classified and placed in different queues, which are assigned with different priorities. A packet in a lower priority queue gets served only if all the higher priority queues are empty. Figure 2.10 gives an idea on how this scheme works.

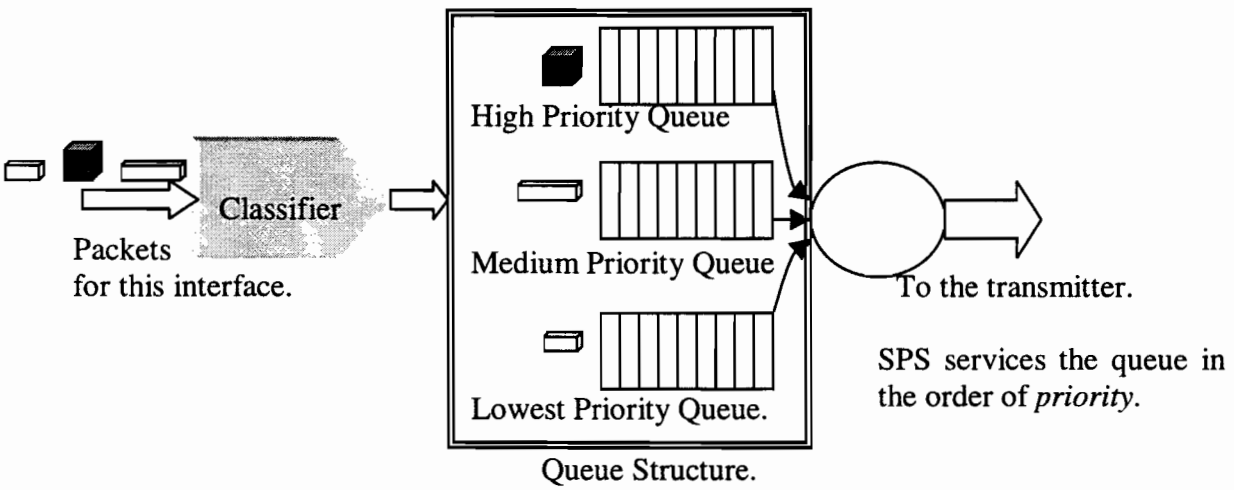


Figure 2.10: Strict priority scheduling.

The highest priority queue will get minimum queuing delay, but all other priority levels may experience resource starvation if the highest priority traffic queue remains

occupied. Strict priority scheduling mechanism is simple to implement but does not scale well especially if there is no limit on the highest priority traffic coming in. More sophisticated scheduling algorithms are proposed to support QoS.

2.6.2 Class Based Queuing (CBQ) and Deficit Weighted Round Robin (DRR)

CBQ is a variation for priority queuing and is designed to prevent complete resource denial for any particular class of packets, which is likely to happen in case of

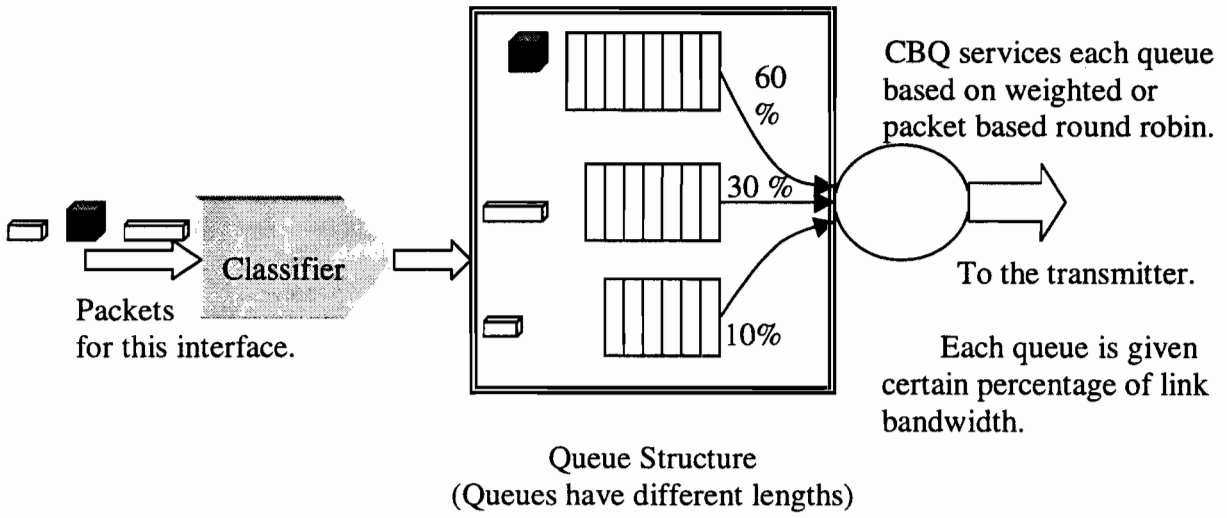


Figure 2.11: Class Based Queuing.

Strict Priority Scheduling. The incoming packets are placed in different queues based on the classification result. These output queues have different queue lengths and are served in a round robin fashion, either packet round robin or weighted round robin. It can also be specified the amount of traffic that can be drained from each queue when it is served. This scheduling scheme provides some fairness by prioritizing queuing service for certain type of traffic, while not allowing any one class of traffic to monopolize the system resources and bandwidth.

Weighted round robin does not take the size of packets into consideration, which works only when all the packets being queued are of the same size. A different version of

weighted round robin called deficit weighted round-robin (DRR) [13] algorithm considers the size of the packets being served. DRR modifies the round-robin algorithm to serve the queues in terms of bits rather in terms of packets. Each queue's weight is used to determine number of bits to be served in a given round, weighted value. A separate deficit counter is also maintained for each queue.

A packet is scheduled from the head queue only if the packet size minus the per-queue deficit counter is less than the weighted value, and the next packet in the queue is tested using a weighted value which has been reduced by the size of the scheduled packet. When the test fails, the remaining weighted quantum size is added to the per-queue deficit counter, and the scheduler moves to the next queue. We have implemented both SPS and DRR in OPNET for our studies.

2.6.3 Weighted Fair Queuing (WFQ) [7][13]

Weighted Fair Queuing (WFQ) is a packet scheduling technique allowing guaranteed bandwidth services. The purpose of WFQ is to let several sessions share the same link. WFQ is an approximation of Generalized Processor Sharing (GPS) which, as the name suggest, is a generalization of Processor Sharing (PS).

In PS each session has a separate FIFO queue. At any given time the N non-empty queues are serviced simultaneously, each at a rate of $1/N^{th}$ of the link speed. Contrary to PS, GPS allows different sessions to have different service shares. GPS have several nice properties. Since each session has its own queue, an ill-behaved session (who is sending a lot of data) will only punish itself and not other sessions. Further, GPS allows sessions (queues) to have different guaranteed bandwidths allocated to them.

The basic idea is as follows, a weight is associated with each flow (queue) and the link capacity is shared among the active flows in direct proportion to their weights. These weights logically specify how many bits to transmit each time the router services that queue, which effectively controls the percentage of link's bandwidth that the flow will get. For example one queue might have a weight of 2, a second queue might have a

weight of 1, and a third queue might have weight of 3, assuming that each queue always contains a packet waiting to be transmitted, the first flow will get $1/3^{rd}$, the second will get $1/6^{th}$ and the third will get 1/2 of the available bandwidth (L). Figure 2.12 shows the working of WFQ,

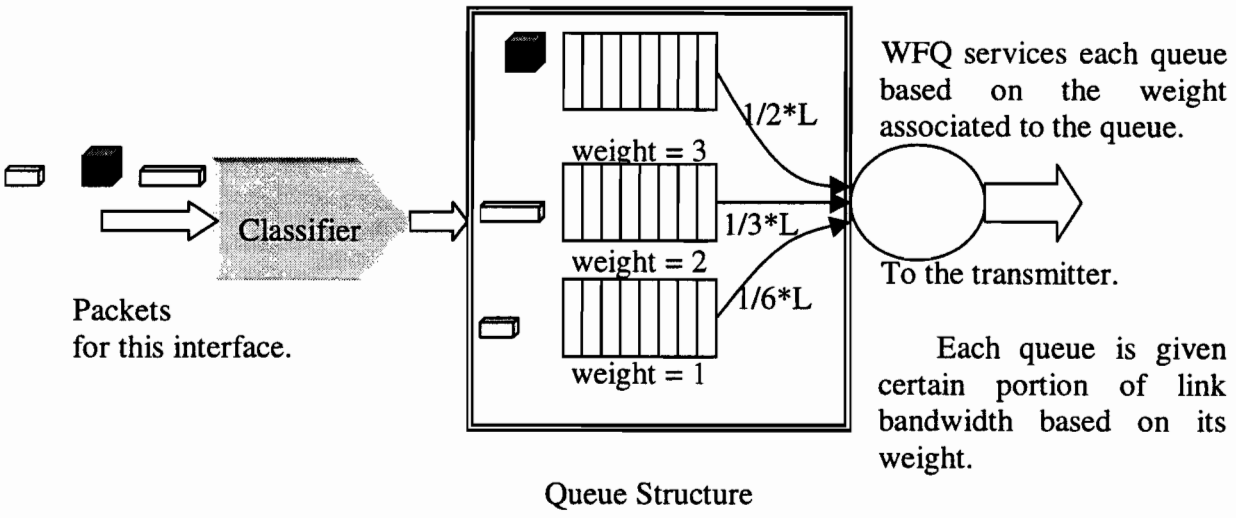


Figure 2.12: Weighted Fair Queuing.

GPS is an idealized model, which is not practically realizable. WFQ, is a packet approximation of GPS. In WFQ a packet at a time is selected and sent to the output among the active sessions. This works as follows, Each arriving packet is given virtual start and finish times. The virtual start time S (k, i) and the virtual finish time F (k, i) of the k^{th} packet in session (queue) i are computed as follows:

S (k, i) = max (F (k-1, i), V (a (k, i)))

F (k, i) = S (k, i) + L (k, i)/r (i)

where F (0,i) = 0, a(k,i) and L(k,i) are the arrival time and the length of the packet respectively. V (t) is the virtual time function representing the progression of virtual time and is defined as follows:

dV (t)/dt = 1/(Sum of active sessions shares at time t)

This means that when there are inactive sessions the virtual time progresses faster. In the corresponding GPS case this can be viewed as that the remaining active sessions get more service. The packet selected for output is the packet with the smallest virtual finish time. DRR is also close approximation of GPS.

The scheduling algorithm that is used depends on the PHBs that are supported. If only AF PHB is supported then CBQ, DRR or WFQ depending on the requirement will be sufficient. If both EF and AF PHBs were supported then a SPQ would be more appropriate. A prioritized WFQ or Deficit Round Robin might also be implemented to avoid EF from depriving bandwidth to AF.

2.7 Design Issues of the Functional Elements

There are various design issues associated particularly the position of the functional elements. The following two sections shows a generalized view of our design approach. The discussion below gives generalized view of the design we intend to use in our studies, specifics will be given later for each study separately.

2.7.1 Input Interface of a DS Node

A generalized picture of the Input Interface of a DS capable node is as shown in Figure 2.13,

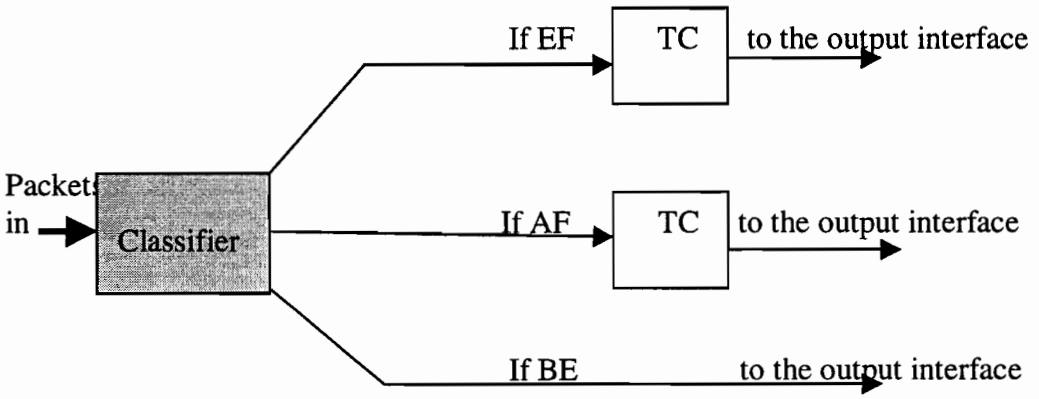


Figure 2.13: Input interface.

Figure 2.13 shows the input interface with all the functional elements it can have. Given a DS node its input interface generally consists of a traffic classifier and traffic

conditioners. Depending on the classifier's output the packets are sent to different traffic conditioners. If the flow is BE then the flow may not be conditioned. The Traffic Conditioner (TC) changes depending on the type of packets being conditioned and also depending on whether it is present in an interior router or in a boundary router. The input interface might not contain any of the DiffServ components if it belongs an interior node. If the TC is used to condition EF packets then it needs a shaper while if it is used to condition AF packets it may not need a shaper. If the TC is present in interior router shaping or policing functionality may not be needed, while if the TC is present in boundary router or leaf router then shaping and policing function are required.

2.7.2 Output Interface of a DS Node

The output interface of a DS capable node may look as shown in Figure 2.14,

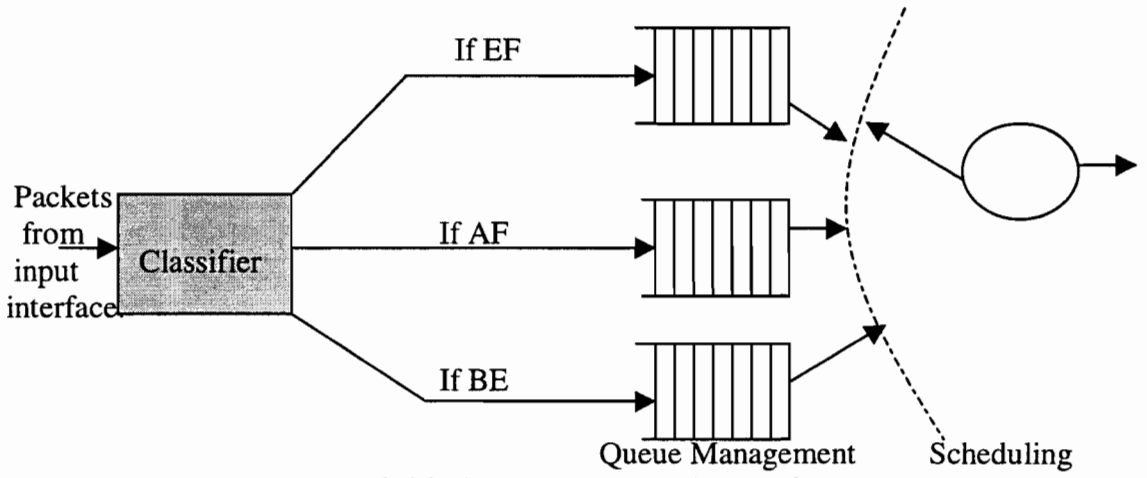


Figure 2.14: Output interface of DS node.

This is not a standard output interface. For example there could be a traffic conditioner before queuing packets rather in the input interface. The classifier output might be different in different cases depending on various implementation issues. If there was more than one AF class then there might have been another queue present to buffer the packets for that particular class. If an AF class has different drop precedence with in it, then the queue in which they are buffered can be chosen to be a WRED (Weighted Random Early Drop queuing discipline described later). If there was only one AF class we might want to buffer AF and BE in one queue using RIO (RED with In and Out

described later) with AF packets marked as *In* and BE packets marked as *Out*. There is also a relation between the scheduling schemes and queue parameters. So depending on the scheduling scheme, the queue parameters should be chosen appropriately.

	Type of node	**Type of interface**	**Functional Elements (FE).**	**Type of FEs.**
DS domain	Boundary Node.	Input Interface.	▪ Traffic classifier. ▪ Traffic conditioners (TC)	▪ BA or MF classifier. ▪ TCs for each PHB offered in domain are required.
		Output Interface.	▪ Classifiers ▪ Queue management ▪ Scheduling ▪ TC (optional)	▪ BA / MF classifier. ▪ FIFO, RED, WRED and RIO the choice mainly depends on the PHBs offered in domain ▪ SPQ, CBQ or WFQ. ▪ TC is used particularly if SPQ is used.
	Interior Node.	Input Interface.	▪ Usually no special components are required.	None.
		Output Interface.	▪ Classifiers ▪ Queue management ▪ Scheduling	▪ Mostly BA classifier. ▪ FIFO, RED, WRED or RIO mainly depends on the PHB that is being realized. ▪ SPQ, CBQ or WFQ.
	Host or Leaf router.	Input Interface.	▪ Traffic classifier. ▪ Traffic conditioners	▪ Mostly MF classifier. ▪ TCs for each service specified in SLA.
		Output Interface.	▪ Classifiers ▪ Queue management ▪ Scheduling	▪ Mostly MF classifier. ▪ FIFO, RED, WRED and RIO the choice mainly depend on the service offered/requested. ▪ SPQ, CBQ or WFQ.

Table 2-6: Summary of functional elements and components and their location.

2.8 Resource Allocation

The two important factors that are critical in providing proper Resource Allocation are monitoring and configuration of components in the network elements. This can be also seen as the management of the network. Monitoring is essential as it keeps track of the current usage of resources and it is needed for accounting purposes like the number of packets dropped in the traffic conditioner.

Resource allocation is one of the crucial aspects for ISP's because once the deployment of differentiated services is done they may have to continuously change the resources allocated according to the number and kind of services sold, and the TCA with the peer domain. For example, at particular instant of time a marked (say EF) packet flow is using up all the resources allocated to its PHB and now if a customer sends the same type of packets (EF) into the network his packets might get dropped which will be a violation of agreement. So the ISP needs to monitor the current usage of resources and change the configuration of network elements like increasing the rate of token bucket shaper to accommodate any excess traffic coming into the network.

Allocation of resources is the process of meeting the traffic commitments made to the customer or the peer domain or the ISP. The allocation ideally should be done immediately when the commitment is made, may be by pre-configuring of usage profile before the commitment is made. There can be two kind of allocations one static allocation in which the customer is given a profile and the appropriate changes will be made in the network to accommodate the customer like resource allocation, so the allocation of resources fairly remains static unless some customer subscribes or unsubscribe to service. The customer will be paying a flat rate for a particular kind of service (i.e. premium or assured service with a particular profile) and the end of service would be only when the customer wishes to unsubscribe this is similar to subscribing to a paid channel for your television. When the ISP sells the service he may have to change the TCA with its peer domain to accommodate the additional customer. The new customer might be accommodated by changing the configuration of network elements.

The other kind of service would be dynamic, which is similar to viewing a pay per view movie on your television, where you will subscribe to a particular service with a particular profile during a particular period of time. To provide this, the ISP's should be capable of dynamic allocation of resources since the customer might be anywhere in the network. The ISP has to talk with its peer domains to check whether it can accommodate the service requested by the customer during that period of time.

Clearly mechanisms are needed to communicate information about the request to the leaf router and peer domain. This configuration information may be the rate, burst, and whether it is Expedited or Assured service that is requested. There should be also a mechanism by which this information is communicated with neighboring peers and a process to communicate right back to the customer informing him that he has been given the service he has requested. A mechanism, which dynamically changes the configuration of network elements, might also be required. A way of doing all this is bandwidth brokers suggested by Van Jacobson [17], which is still in the process of development and RSVP or SNMP can be used by the network elements to pass different monitoring and configuration information i.e. for communication.

For our study purpose a strictly statically allocated scenario was used. Where the network elements are configured and the resources are allocated according to amount of marked traffic that is supposed to flow in the network.

Chapter 3

Related Work

This chapter discusses previous research closely related to our studies. Specifically, each section discusses a published paper directly related to our work.

3.1 Preliminary Simulation of an Assured Service

The main problem that was addressed in [8] was the analysis and evaluation of Assured Service (AS) suggested by IETF [15]. RIO (RED IN/OUT) is investigated, which is intended to provide assured service. The NS (network simulator) [16] was used to model the network. AS and Best Effort traffic were mixed. RIO was used to discriminate between AS and Best Effort (BE) traffic.

3.1.1 Simulation Configuration and Parameters

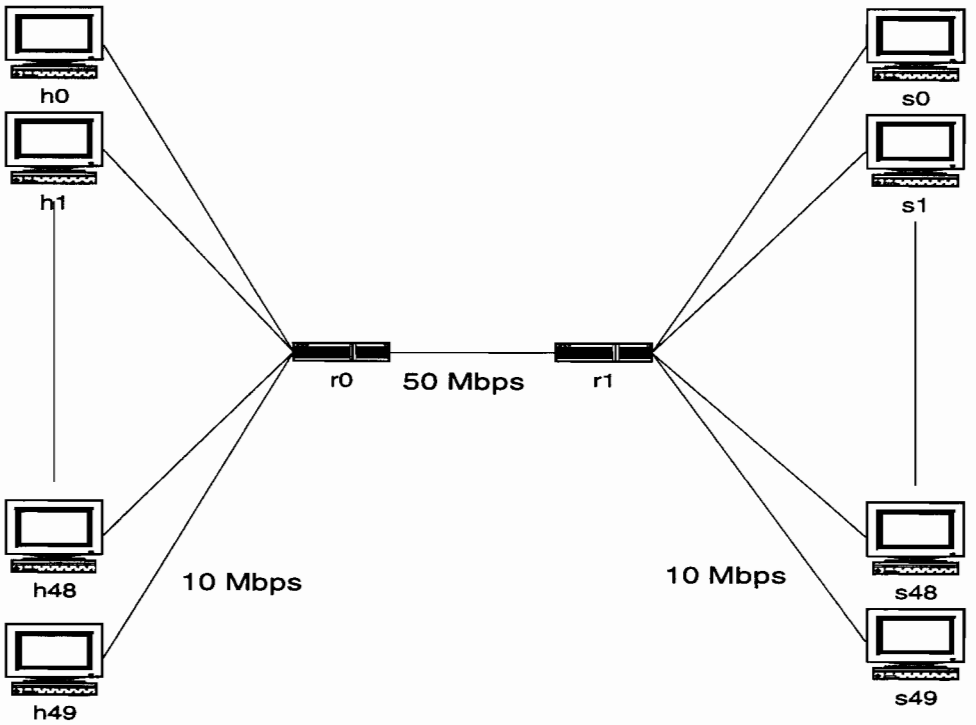


Figure 3.1: Network topology used for simulations.

The network model considered in [8] (as shown in Figure 3.1) consists of fifty sources (ti's) and fifty (si's) destinations. Bottleneck link between r0 (Router0) and r1 (Router1) is 50 Mbps (6.25 MBps). 10 Mbps link were used between sources (ti's) and Router r0. A profiler is attached to DiffServ (or Assured Forwarding) capable source with a Target Rate of 1.0 Mbps (125 KBps). An aggregate policer is attached in Router r0.

A profiler in this [8] refers to a marker, which marks packets IN (AS) or OUT (BE) based on the profile it is configured to and whether the packet belongs to AS or BE. A BE is always marked as OUT while an AS packet can be marked as IN (if the AS packets coming in doesn't exceed certain profile (rate and burst)) or OUT (if the AS packets coming in exceed certain rate and burst).

Packet lengths are 576 bytes (distribution was not given). Information transfers are unidirectional. The sources are configured with infinite FTP connections, which are considered to represent greedy sources. Starting times are randomly distributed within first few seconds of simulation. RTT's are randomly chosen in the range of 50 to150 ms for each connection. It was assumed that ACKs are never lost.

RIO Parameters:

The following notation was used to represent RIO parameter:

weight = 0.002.

Minimum Threshold (minth_(in or out)) / Maximum Threshold (maxth_(in or out))/ Maximum Drop Probability (maxp_(in or out)).

Parameters for OUT packets: 0.40*maxq_size / 0.8*maxq_size / 0.05.

Parameters for IN packets: 0.45*maxq_size / 0.8*maxq_size / 0.02.

It was pointed that setting IN parameters to much better values than OUT parameters may cause trouble if most of the packets arriving are IN packets. It was quoted that enough discrimination can be achieved between IN and OUT packets by choosing IN parameters close to OUT parameters, with any proportion of marked traffic. Therefore, the parameters that were used were, for IN packets are 420/840/0.02 and for OUT packets are 500/840/0.05 with maxq_size as1050 packets.

Profiler/Policer Choice:

Two mechanisms token bucket and average rate estimator with the same network model described above were investigated in [8]. Token bucket was found superior for following reasons.

1) It was observed that AS connections achieved better performance and the discrimination between AS and BE connections was much better, when Token Bucket was used.
2) Token bucket permitted transmission of a deterministic burst of IN packets while average rate estimator probabilistically marks some packets as IN and some as OUT beyond the target rate. Therefore a token bucket allows natural burstiness of TCP by marking all packets as IN within a burst while average rate estimator marks some as OUT. This is true when compared to average rate estimator but not general token bucket does mark some packets as OUT if the burst is large as shown in other papers and in our studies.
3) Average rate estimator allowed source to transmit at a sustained rate higher than the contracted one, token bucket does not allow this.

So, token bucket was used - with a token depth of 20 Kbytes at sources (ti's) and 25 Kbytes at the aggregate policer in Router r0.

3.1.2 Scenarios and Results

3.1.2.1 Influence of RTT

A TCP source takes longer to recover from a packet lose if the RTT is large. The purpose of this scenario was to determine if this effect is reduced by the use of AS. Simulations are run with 10, 25 and 40 AS connections out of the 50, the others being BE connections. Each AS capable source had a target rate of 1 Mbps (125 KBps), i.e., 20%, 50% and 80% of the bottleneck capacity for 10, 25 and 40 AS sources, respectively. Simulations were also run for cases when the sources were all AS sources and they were all BE sources. The achieved rates were plotted vs. RTT's.

Observations:

1) The dependency of achieved rate on RTT is noticeable for AS connections.
2) When AS connections are large in number (40) they showed less deviation to different RTT values. The deviation is also less compared to the deviation shown by BE connections when they are large in number (40).
3) Some connections did not achieve the target rate while others exceed the target rate (especially for low RTT's). It was pointed out that when average rate estimator was used some BE connections got more bandwidth than some AS connections.
4) All AS connections show less RTT unfairness (or deviation of achieved rate with RTT) compared to all BE connections.

Reasons given for the Observed Results:

Since a connection with a small RTT gets more bandwidth by opportunistically exceeding the target rate and sending OUT packets, many of those packets will be dropped, causing the connection to decrease its sending rate. The more OUT packets a source sends, the higher the probability that one or more of its packets will be dropped within a certain time interval. That has the effect of mitigating the gain of connections with a smaller RTT.

It was shown that in 'all BE connections' case the average queue size oscillates substantially compared to AS connections, leaving room for small RTT connections to opportunistically send more packets (reason for observation above).

Having a significant amount of assured traffic not only lowers the dependency on the RTT for AS connections, but also for best-effort connections. The cause is the same as above because connections with a small RTT send more packets than those with a large RTT, thus having more chances to undergo a packet drop in a certain time interval (reason for observation above). The results related packet drops are not provided. The exact reason for substantial oscillation of BE's average queue size was not given.

3.1.2.2 Performance with respect to Target Rates

In this case the authors of [8] investigated how well AS TCP connections achieve their target rates. The simulation used 40 connections 20 of which are AS and the remaining 20 are BE. All the connections had a RTT of 100 ms. The bottleneck link bandwidth was 20 Mbps (2.5 Kbytes/s) and the RIO parameters were 347/694/0.05 390/694/0.02. 76% of the total bandwidth is allocated to the AS connections. If the excess 24% of the total bandwidth is equally allocated among all the 40 connections then each connection would get 120 Kbps of additional bandwidth.

Observations:

1) Only connections with small target rates reach or exceed it.
2) As the target rate increases the achieved rate by the connections also increases but not proportionally.
3) It is observed that the connections were achieving a rate more than the target rates when target rate is small and less than the target rates when the target rate is large.

Reasons for the Observed Results:

The reason for the above observations was given as due to the variation of congestion window. When the window closes down due to a packet lose the connections with lower target rates come back sooner to the original window size faster than the connections with higher target rates. During the time taken for large connections to come back to original window size the smaller connections use the excess bandwidth left over. Hence the achieved rates are higher for smaller target rate connections and lower for higher target rate connections. This is comparable to the opportunism of small RTT connections at the expense of large RTT connections.

3.1.2.3 Effect of a Non-Responsive Source

In this scenario the influence of a non-responsive source was studied. They simulated 20 point-to-point connections with bottleneck link of 20 Mbps. The RIO parameters were 173/347/0.05 and 195/347/0.02. The RTT for all connections was around 100 ms. There

were 10 AS connections and 10 best effort including the non-responsive CBR source connection.

Discussion of Results:

Non-Responsive (CBR) source as Best Effort:

1) As the non-responsive CBR source increases its sending rate all the TCP connections get degraded, with Best Effort connections being pushed towards starvation.
2) Even though the non-responsive source (CBR) experiences increasing drops it manages to capture a lot of bandwidth.
3) It was pointed out that the CBR source reaches a limit. This limit corresponds to situation where no OUT packets are sent by AS connections. This happens when any OUT packet issued gets dropped as the maximum threshold is constantly reached. In this case the AS TCP connections oscillate between slow start and congestion avoidance phase. As a results all IN packet are generated, which will make their way through the congested link stopping the CBR source from grabbing more bandwidth.

Non-Responsive (CBR) source as Assured Service: The results achieved are almost identical to the previous case.

Thus it was observed that a non-responsive source produces same impact when it is AS capable or not. As long as the CBR source sends at a contracted rate, the AS connections use the bandwidth allocated to them. On the other hand even if the non-responsive source breaches the contract it will grab more bandwidth at the expense of TCP connections.

3.1.2.4 Effect of AS on Best-Effort Traffic

When the number of AS connections is increased the BE connections rate is decreased, this is acceptable. In this case it was investigated if the AS and BE connections compete equally for the excess bandwidth.

The simulations were run with 25 AS connections with a target rate of 125 KBps, 25 BE connections and a bottleneck link bandwidth of 6.25 MBps. This leaves a 2.5 MBps of

excess bandwidth. If the excess bandwidth is shared equally among the 50 connections each connection should get 50 KBps.

Discussion of Results:

It was observed that the amount excess bandwidth obtained by the BE connections is 75 KBps (average) and the AS connections were able to get only 30 KBps (average). It is clear that BE connections were able get more than equal share of excess bandwidth while AS connections were getting well below the equal share.

The reason for this is given as whenever AS connections send OUT packets they experience some drops just as BE connections. But AS connections have higher target rates so they take more time to increase their window size back to the original size, meanwhile BE connections grab the unused bandwidth.

3.1.2.5 Effect of Traffic Bursts on AS

In this case a more complex merging topology was simulated to study the effect of traffic bursts.

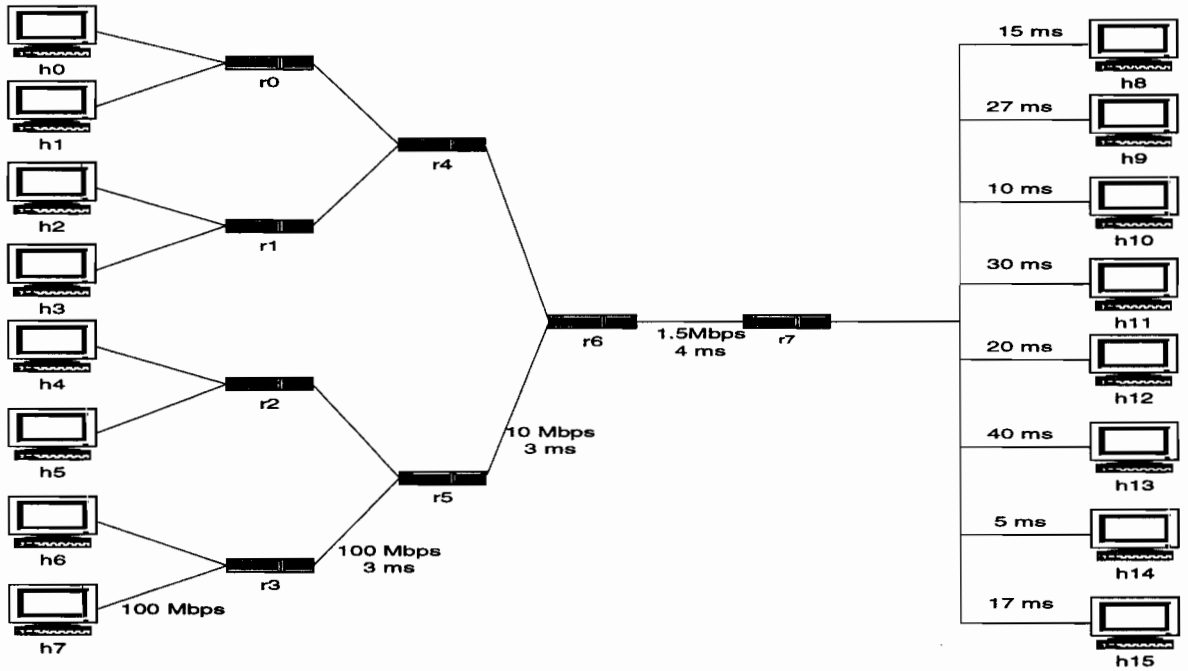


Figure 3.2: Network Topology used to study Traffic Bursts.

Figure 3.2 shows the topology used to evaluate impact of burst size. All the hosts are AS capable with a target rate of 100 Kbps (12.5 KBps). Aggregate IN traffic is policed at each node with the following target rates 200 Kbps for node r0 to r3, 400 Kbps for r4 and r5 and 800 Kbps for node r6. Packet sizes are 1500 bytes including the headers. The topology represents a decreasing bandwidth hierarchy ending in a T1 link. The queue at the T1 link is RIO with a maximum queue size of 24 packets and with parameters 4/9/0.05 5/9/0.02.

Discussion of Results:

It has been observed that some IN packet were dropped but no early drops were seen. Thus marked traffic is dropped even when it conforms to the profile (IN). It was observed that 7.5% of IN packets are demoted to OUT packets. The IN packet drops were observed due to queue overflows due to bursts of the OUT packets.

3.1.3 Conclusions

In this study [8] the Assured Service was evaluated under different scenarios.

- They concluded that AS cannot offer a quantifiable service to TCP traffic.
- In particular the defined service, 'AS' does not allow a strict allocation bandwidth for users.
- It was concluded that due to use of single queue for IN and OUT packets, there is a dependency on each other. As result IN packets were getting poor performance.

Drawbacks:

- It can be seen that the service model, topologies and traffic models used were simple and optimistic.
- It was said that bandwidth was allocated to assured service traffic, but the allocation scheme was not given. In particular as both IN and OUT belonged to the same class the bandwidth allocation seemed confusing.

3.2 Fair Bandwidth Allocation in Differentiated Services

In this section the work done in [18] is discussed, which studied how fairly bandwidth is allocated by three proposed schemes for providing differentiated services viz. the RIO scheme [8], the two-bit scheme [17] and the User-Share Differentiation (USD) scheme [19]. The three schemes were simulated using the same network configuration as shown in Figure 3.3.

3.2.1 Network Configuration and Parameters

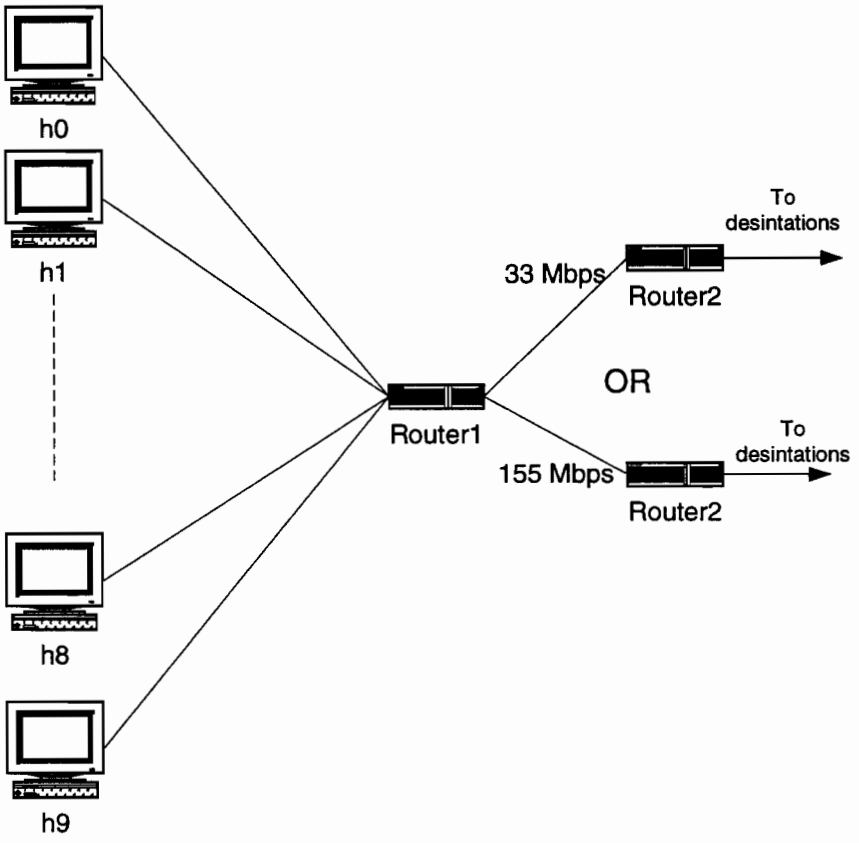


Figure 3.3: Network Topology.

REAL simulator was used in [18] to simulate the network configuration shown in Figure 3.3. Ten sources were used which communicated with one of the two destinations. Links from the sources to the first router are 30 Mbps and links from the first router to

destinations either 33 Mbps or 155 Mbps, i.e., simulations were carried out with bottleneck link capacity of 33Mbps and 155 Mbps separately. The one way propagation delays for the sources from the top are 20, 20, 40, 40, 50, 50, 70, 70, 100 and 100 ms respectively.

All the sources use TCP-Reno and each of them send 500,000 packets to the destination each packet being 500 bytes. The RIO parameters are 400/800/0.02 for IN packets and 100/200/0.5 for OUT packets. Actually the RIO scheme can be compared to the studies performed in [8] and discussed in 3.1. For User Share Differentiation (USD) WFQ is used and for Two-Bit scheme two queue priority scheduler is used to discriminate between premium and assured service packets.

3.2.2 Scenarios and Results

3.2.2.1 Effect of Expected Bandwidth Profiles

In this scenario the effect of expected bandwidth (throughput) profiles and bottleneck bandwidths on actual bandwidth allocation was studied. The same network configuration as in Figure 3.3 was used. In each simulation the odd numbered sources were configured with lower expected bandwidth profile and the even numbered sources were configured with higher bandwidth profiles. The expected lower and higher bandwidth profiles were set to 1Mbps and 5Mbps (aggregate (30Mbps) close to 33Mbps), 3Mbps and 15Mbps (aggregate is close to 94Mbps, average of 33Mbps and 155Mbps) and 5Mbps and 25Mbps (aggregate (150Mps) close to 155Mbps). The simulations with different expected bandwidth profiles were run twice, with 33Mbps and 155 Mbps bottleneck links separately. Notation used: X Mbps (y, z), X is the bottleneck bandwidth (33 or 155 Mbps), y is lower bandwidth profile and z is the higher bandwidth profile.

RIO Scheme:

Load nearly equal to 1.0 (33Mbps(1, 5) and 155Mbps(5, 25)):

In this case the results were obtained maintaining the link almost at full load. From the results obtained it was shown that the RIO scheme performs well in this case, i.e., the

actual bandwidth allocated matched the expected bandwidth profiles. It was also observed that the allocated bandwidth was slightly higher for lower bandwidth profile sources and slightly lower for higher bandwidth profile sources for 33Mbps (1, 5) case, while this was opposite for 155Mbps (5, 25) case. It was suggested that this difference might be due to RED parameters indicating the dependence of RED on bandwidth profiles.

Overloaded (33Mbps(3, 15) and 33Mbps(5, 25):
From the results obtained it was shown that the RIO scheme performs well for the most part if the bottleneck bandwidth is less than the aggregate expected bandwidth.

Underloaded (155Mbps(1, 5) and 155Mbps(3, 15):
The RIO scheme is unable to allocate bandwidth in proportion to user expectations if there is excess bandwidth at the bottleneck. When there was excess bandwidth, the bandwidth was allocated evenly among all the sources giving lower bandwidth profile sources allocations almost equal to higher bandwidth profile sources. The reason for this was given as the excess bandwidth is allocated only to OUT of profile packets.

Two-Bit Scheme:
In this case the higher bandwidth profile traffic was configured as premium and the lower bandwidth profile was configured as assured. The RIO parameters, the network configuration and the simulations were the same.

Load almost Equal to 1.0 (33Mbps(1, 5) and 155Mbps(5, 25)):
It was shown that in this case the two-bit scheme performed well in this case, actually better than the RIO scheme.

Overloaded (33Mbps(3, 15) and 33Mbps(5, 25):
Some of the high-expected bandwidth sources were seen to get no bandwidth at all. It was pointed that the connection setup calls got rejected due to the lack of bandwidth.

Underloaded (155Mbps(1, 5) and 155Mbps(3, 15):

It was observed that the lower bandwidth profile sources got more bandwidth allocation than higher bandwidth profiles. Because it seems that the premium traffic is never allocated excess bandwidth because all the out of profile premium packets are dropped. The performance is poorer than RIO scheme.

USD Scheme:

Using USD it was observed that all the sources were almost exactly getting their share of bandwidth irrespective of the load on the bottleneck. It was also observed the share of the bandwidth is also fair i.e. it was configured and shown that premium (or higher expected bandwidth profile sources) get a higher relative share of the bandwidth.

3.2.2.2 Short Duration Traffic

In this case the effect of short connection life times on three different schemes (RIO, Two-Bit and USD schemes) were studied. The sources were modified to send only 200 packets. The rest of the settings were the same. Simulations were performed for only 33Mbps (1, 5) and 155Mbps (1, 15) cases.

RIO Scheme:

33Mbps (1, 5): The bandwidth allocation does not commensurate with the expected bandwidth profiles. The reason that was given in [18] was that the average queue length determined by RED algorithm and the Time Sliding Window technique used by their profile meter depend on long term averages. They indicated that in short term the Time Sliding Window tends to overestimate the sending rate and mark more packets as out of profile than necessary. It was recommended that the closer the time windows are to the round trip times the more accurate the estimates are.

155Mbps (1, 5): It was observed that the results make sense compared to the previous case, i.e., the higher bandwidth sources received more bandwidth than lower bandwidth sources expect for one source. The bandwidth was allocated al most equally for all the sources for the same case in the previous scenario.

But the reason for the difference in this case compared to previous case (and previous scenario) was given as, higher bandwidth sources were able to grab higher bandwidth share of the excess bandwidth since their congestion windows open up faster (due to lower drop rate at RIO) than those of the lower bandwidth sources. This would be direct contradiction to [8] and section 3.1 if there were some packets of higher bandwidth sources were dropped.

Two-Bit Scheme:

It was observed that for both cases, i.e., 33Mbps (1, 5) and 155 Mbps (1, 5) the higher bandwidth profile sources got lower bandwidth than lower bandwidth profile sources. The reason for this behavior was given as the way the premium traffic is managed. The monitor (profiler) generates premium tokens at a fixed rate. Since TCP initially doubles the size of the congestion window every round trip time, the traffic generated is bursty and the rate can exceed the token generation rate for a short term resulting in drops. This was observed to happen initially resulting in multiple time outs causing the congestion window open up very slowly. It was pointed out this behavior would be absent if the simulations were run longer. The lower bandwidth profile sources were able to grab the unused bandwidth getting higher bandwidths.

USD Scheme:

From the results obtained it was shown that they are almost the same as obtained in the previous scenario and it was observed that the sources get their expected share of bandwidth. For short duration traffic USD scheme is recommended for providing differentiated services.

3.2.2.3 Non-Responsive Sources

In this case the effect of non-responsive sources was studied. Two kinds of non-responsive sources were used CBR sources that send at a fixed rate and malicious sources that flood the network by sending as fast as they can. Only the first two sources were changed the rest of the sources were the same with even numbered sources being high bandwidth profile sources with 5 Mbps while odd numbered sources being low

bandwidth profile sources with 1 Mbps. The simulations were run for both 33 Mbps and 155 Mbps bottleneck links.

CBR Source case: The CBR sources were configured to transmit at 5 Mbps (expected bandwidth) and all their packets were marked as OUT.

RIO Scheme:

For both the cases (i.e. with 33Mbps and 155Mbps bottleneck) the CBR sources were able to get close to their expected bandwidth. The TCP sources for 33 Mbps case got bandwidth close to their expected bandwidth even though the high bandwidth sources got slightly less than expected while the low bandwidth sources got slightly high than expected. For 155 Mbps case all the sources got in excess to their expected bandwidth and it was observed that the bandwidth was allocated equally among the sources.

Two-Bit Scheme:

The CBR sources almost got their expected bandwidth for both the cases. For 33 Mbps case all the TCP sources got their expected bandwidth expect for one low bandwidth source which got almost nothing. It was observed that the bandwidth lost by this source was allocated among the rest of the low bandwidth source. The reason for this was given attributed to Non-Responsive source. For 155 Mbps case all the sources got their expected bandwidth but all the excess bandwidth was allocated to low bandwidth sources. Because the high bandwidth sources (premium) packets were dropped if they are out profile.

USD Scheme:

As expected all the sources got their expected share of bandwidth. Again USD scheme is recommended to provide differentiated services.

Malicious Sources: Again the first two sources were configured as malicious sources, which declared an expected bandwidth of 1 Mbps and then try to flood the network. The rest of the settings are the same as previous case.

RIO Scheme:

For 33 Mbps case all the TCP cases do not do very well. None of the high bandwidth sources got more than 2.5 Mbps, while low bandwidth sources got almost nothing. The reason for this was given as due the continuos flooding of queues by the malicious sources resulting in packet drops of the TCP sources. For 155 Mbps case the malicious sources grabbed a bandwidth equal to their outgoing link to the router, 30 Mbps. The TCP sources got more than their expected as there was enough bandwidth. Again the bandwidth was almost distributed evenly since the excess bandwidth was allocated only to the OUT packets.

Two-Bit Scheme:

For 33 Mbps case the low bandwidth TCP sources got almost nothing. While the malicious sources were able to grab less bandwidth compared to the RIO scheme. Because in Two-Bit scheme the premium traffic was queued in a separate queue so high bandwidth (premium) sources were able get their expected bandwidth resulting in lower bandwidth to non-responsive sources compared to RIO scheme. Since low bandwidth source traffic was queued in the same queue as malicious source traffic they got almost nothing. For 155 Mbps case it was observed that the high bandwidth (premium) sources got a bit less than their expected bandwidth. The reason for this was given as because the malicious source continuously filling up the low priority queues resulting the router spend more time on low priority queues than high priority one. The excess bandwidth was distributed evenly among the low bandwidth sources while the malicious sources got 30 Mbps equal to outgoing link bandwidth. The premium sources got no excess bandwidth, as there OUT packets were dropped (excess bandwidth is allocated only to OUT packets).

USD Scheme:

From the results obtained it was shown that for both 33Mbps(1, 5) and 155Mbps (1, 5) cases the malicious sources were not given any extra bandwidth and actually were

slightly punished. The well-behaved TCP sources got slightly more than their expected bandwidth.

3.2.2.4 Flows with High Expected Bandwidth

In this case all the sources were configured as TCP with expected bandwidth of 5 Mbps over a 33 Mbps bottleneck link. The RIO scheme allocates bandwidth almost evenly to all sources. The Two-Bit Scheme was able to give connections to only 6 out of 10 sources, which got fair bandwidth allocation, but it has to deny connections to four sources. USD scheme allocates equal bandwidth to all the sources.

3.2.3 Conclusions

The following conclusions are inferred from the work done in [18],

- It was shown that RIO and Two-Bit scheme performs well only when there is no mismatch between expected aggregate bandwidth and bottleneck bandwidth.
- USD scheme was shown to perform well with non-responsive sources and short term sources compared to RIO or Two-Bit schemes. RIO performed better than Two-Bit scheme with non-responsive sources.

Drawbacks:

- USD scheme is not as scalable as other two schemes.
- The Two-Bit model implemented is not the best performance model. The model can be changed to give better performance especially dropping of OUT packets for premium packets and excess bandwidth being allocated only to OUT packets.
- Neither the RIO model used was configured as good as in other studies (see [8] and section 3.1) to provide service differentiation.
- The results and scenarios were biased towards USD scheme.
- Proper reasons were not given for the results obtained. Some of the results concerning RIO scheme are contradicting results from [8], as discussed in section 3.1.

3.3 Performance Analysis of Assured Forwarding

This section discusses the work done in [20]. A large number of simulations were run in [20] with a variety of parameters for the two-color and three-color markers and RED under two scenario viz. two-color and three-color precedence.

3.3.1 Simulation Configuration and Parameters

The network configuration as shown in Figure 3.4 consists of ten customer sites nine of them contain five TCP sources while one site contains a single UDP source sending at a date rate of 1.28 Mbps. The parameters are shown in Table 3.1 and 3.2.

Simulation Time	TCP Window	IP Packet Size	UDP Rate	Queue Size
100 sec.	64 packets	576 Bytes	1.28 Mbps	60 packets/576B

Table 3.1: Simulation Configuration Parameters

Link from / Parameters	**CEN* to Sources**	**CEN to Router 1**	**Router1 to Router2**	**Router2 to Router2**
Link bandwidth	10 Mbps	1.5 Mbps	1.5 Mbps	1.5 Mbps
One way delay	1 msec.	5 msec.	30 msec.	5 msec.
Drop policy	Tail Drop	Tail Drop	RED_n	Tail Drop

Table 3.2: Simulation Configuration Parameters (contd..)

CEN* - Customer Edge Node.

All the packets are initially marked as green before being remarked by a traffic conditioner at CENs. The traffic conditioner that is used consists of two leaky buckets (one for green and one for yellow packets), which marked the packets according to their token generation rates (the green token generation rate is considered as reserved rate). In two-color simulation yellow rate is configured as zero. So in two-color simulation both UDP and TCP packets are marked as green or red. In three-color simulation the CEN of UDP site has yellow rate as zero and rest of the CENs have non-zero yellow rates.

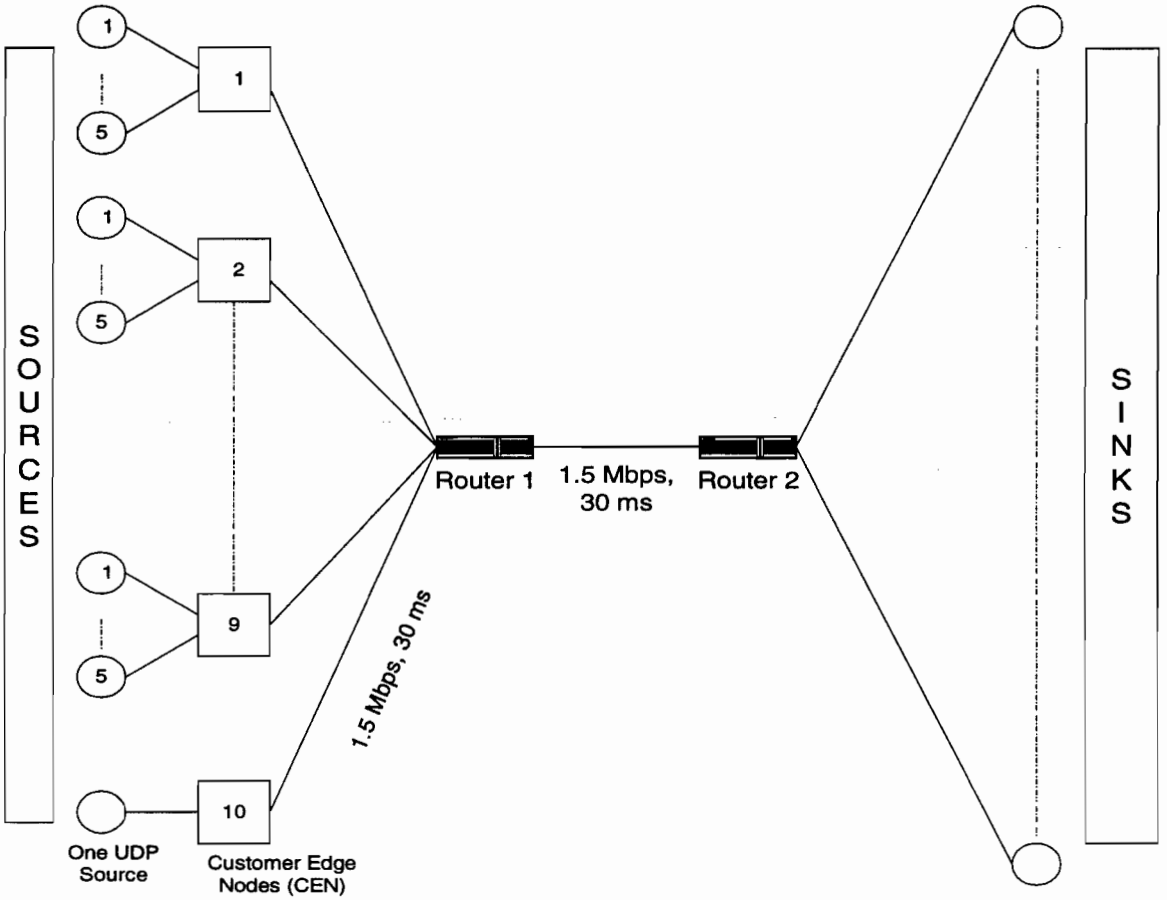


Figure 3.4: Network Topology.

Therefore for three-color simulations TCP packets can be marked as green, yellow or red while UDP packets can be marked as green or red. It has to be noted that a color represents a drop precedence in a AF class. The amount of traffic generated from a site is controlled by the bucket rates. Simulations were carried out for different values of bucket rates for both two and three color simulations. All the traffic entering Router1 passes through RED.

3.3.2 Simulation Results

The simulations were run for a variety of parameters of the traffic conditioner and RED algorithm for both the scenarios. Again different values of rates for buckets were used which determines the amount of traffic generated from a site. In this section those parameters are not discussed but only the results obtained are discussed. It can be said that all sensible parameters were used for two and three color marker and for RIO to

carry out a lot of simulations. The results in this paper are evaluated using two performance metrics the utilization of reserved rates by customers and fairness in allocation of excess bandwidth to the customers.

Reserved Rate Utilization of a customer is measured by the ratio of the green throughput of the customer and the reserved rate. The green throughput of a customer is given as the number of green packets (bits) that are received at the destination. It is also pointed out that the chances of getting a green packet dropped are minimal because of large threshold values.

Fairness in allocation of excess bandwidth among n customers is computed by the following formula, which was taken form [21],

$$\text{Fairness Index} = (\Sigma Xi)^2 / [n * (\Sigma Xi)^2]$$

Where Xi is the excess bandwidth of the i^{th} customer. Excess bandwidth is the number of yellow and red packets received by the destination.

Reserved Rate Utilization:

It was observed that the UDP customer always had good reserved rate utilization for both two and three color simulations. Only in some cases when the network was oversubscribed it was seen that *reserved rate utilization* for UDP was lower than one. In contrast in spite of very low drop probability of green packets TCP customers were not able to fully utilize their reserved rate in all cases. It was pointed out that for TCP customers green bucket size is the main factor in determining the reserved rate utilization, since TCP traffic is bursty in nature it is not able utilize its reserved rate unless the bucket size is sufficiently high. Whereas the UDP traffic sending at uniform rate of 1.28 Mbps was able to fully utilize its reserved rate even when bucket size was low. It is also pointed out that the minimum size of the leaky bucket to fully utilize the token generation rate depends on the burstiness of the traffic.

Fairness:

It was pointed out that with two color the fairness in allocating excess bandwidth is poor. Excess traffic of UDP and TCP are marked as red and hence both get the same treatment in network. The congestion sensitive TCP flows respond to congestion by slowing down whereas the congestion insensitive UDP flows grab the extra bandwidth resulting in unfairness. It was pointed out that with three-color simulation the fairness varied widely with fairness being good in most of the cases. In three-color simulations the excess TCP traffic is marked as yellow in contrast to red as done in two color simulations. It was observed that better fairness was achieved if the sufficiently large yellow bucket rate and size were used, so that all the excess TCP traffic is marked as yellow. Again large bucket sizes were recommended to accommodate bursty TCP traffic.

3.3.3 Conclusions

- It was concluded that the three-color (drop precedence) model performs better than two-color (drop precedence) model especially concerning the fairness.
- It was concluded that for overbooked case the two color and three color model gave the same performance.
- Some recommendations were made in their conclusions viz. the aggregate reserved rate for all customers should be less than network capacity, RED parameters have significant impact on performance.

Drawbacks:

- Results were not presented to support most of the conclusions that were made.
- If the IN UDP packets are green then the interaction of green UDP and TCP packets is not given.

3.4 Voice over Differentiated Services

In this section we discuss the work done in [22], which has studied the Two-Bit differentiated service model to study voice/video traffic models. The premium service

was allocated to CBR traffic sources and assured service was allocated to ON-OFF traffic sources.

3.4.1 Simulation Configuration and Parameters

The network studied in [22] consists of three core routers, each of them consists of two priority queues. The higher priority is assigned to premium traffic (Np - number of premium flows) and the lower one to assured (Na - number of assured flows) and best effort (Nbe - number of best effort flows) traffic. The lower priority queue implements RIO scheme.

The traffic enters the core routers from the leaf routers, which reshape (premium traffic by en queuing) and police (assured traffic by demoting to best effort) using leaky bucket. The delays in leaf routers are ignored. The border consists of token buckets for aggregate premium and assured packets, which discard non-confirming premium packets and remark non-confirming assured packets to best effort. The parameters for each source are shown in Table 3.3. The bucket parameters represent the marker parameters for each source.

Source	Avg. Rate	Peak Rate	POI*	Packet Length	Flow's Token Rate	Flow's Bucket Size
CBR	8000	8000	-	576	8000	576
ON-OFF	3200	8000	10	576	5000	3744

POI*: Average number of packets during on interval.

Table 3.3: The source's traffic descriptor (all units are in Bytes or Bytes/ second).

The core links ran at 45 Mbps of which 20% is assigned to premium, 40% is assigned to assured and the remaining 40% is assigned to best effort traffic. The aggregate rate of the border policers was set according to bandwidth assignment. Premium token rate was set to 1.125 MBps and assured token rate was set to 2.25 MBps. The bucket depth was varied for each simulation; it was set to sum of bucket depth of all flows that reached the border router for e.g. if Np = 140 and Na = 703 then premium bucket size is 140 * 576

bytes and assured bucket size will be 703 * 3744 bytes. The assured bucket parameters were set such that 10% of packets were remarked to best effort. For the base model the premium queue size was set to 30 packets (each 576 bytes) such that the delay was bounded by 10 msec. The RIO queue was set to 650 packets for the base model such that the delay was bounded by 200 msec. when there are no premium packets. Whenever the queue sizes are changed it is seen that the same delay bounds are maintained. The RIO parameters were set to 0.4*max_qsize/0.8*max_qsize/0.05 for OUT packets and 0.45*max_qsize/0.8*max_qsize/0.02 for IN packets and a weight of 0.004 was used.

3.4.2 Simulation Results

In first scenario they changed the number of premium flows (Np - 35, 70, 105, 125 and 140) at each node. The assured and best effort loads are 0.36 and 0.44 instead of 0.4 each, as 10% of assured packets are remarked to best effort. The end-to-end delay at the highest load for the premium packets is 1.15 msec and for assured packets it is 130 msec, which is less than 200 msec bound. It is shown that the assured flows delay stay within a reasonable value as long as the network load remains below 95%. The standard deviation of delay data is shown to increase for assured flows as the load is increased (i.e., as Np increases), while for premium flows it decreases after a certain point. The reason is given as majority of premium packets, arrive late at destinations when the load is high and arrive early when the load is low, so in both cases the standard deviation is low. There were no premium packet loses. While at highest load RIO drop rates were limited to 10^{-06} for assured packets and to $2*10^{-02}$ for best effort packets.

In the second scenario both premium and assured load were varied. The ON-OFF sources, i.e., assured sources were changed to CBR sources with same parameters as premium sources. Results for delay and RIO buffer occupancy were obtained. In first case Premium load was kept constant at 20% and assured load was varied from 25% to 40%. In the second case Assured load was kept constant at 40% and premium load was varied from 5% to 20%. It has to be noted neither of classes is overloaded till now. It was observed that at the same high total load value, high premium load, i.e., 20% created more congestion in the RIO queue than at high assured load, i.e., 40%, while in the rest of

the cases the opposite was found true, which was obvious. It was also observed that assured delays got better values at fixed premium load of 20% than at fixed assured load of 40%, which is again obvious as the assured load in the former one was varied from 25% to a maximum of 40%. At all loads none of premium or assured packets were dropped. The best effort packets were dropped only at high loads (i.e., 20% premium and 40% assured), when the premium load was constant the drop rate was $5.98*10^{-05}$ and when assured load was constant it was $1.54 * 10^{-05}$. Another observation that assured CBR sources in this scenario got better delay performance than ON-OFF sources in the previous scenario particularly for high loads.

In the third scenario in [22] the effect of shaping on premium flows. For a load of 100% Np = 140, Na = 703 and Nbe = 703. The effect of shaping was studied at the leaf router i.e. before the flows are injected into the core router. The following four cases were studied,

1) With no reshaping.
2) With same token bucket parameters but CBR packets had a small initial jitter.
3) This case is same as 2 but the bucket size was increased from 1 packet to 2 packets.
4) This case is same as 2 but the token rate is made 1.04 times the previous value.

The delay results were obtained. It was observed that for case the delay and standard deviation of delay values was higher than other cases. While the delay values for the rest of cases were almost the same. The standard deviation of delay values for cases 3 and 4 were higher than case one because of the initial jitter.

In another case the Premium sources were changed from CBR to ON-OFF sources. The bucket sizes were varied at leaf router to study the effect, accordingly the bucket depth at border router is also changed. It was observed that the delay experienced by premium packets at low bucket values were quite high. The delay values started come down after a bucket size of 10, which is the number of packets sent by the sources during the ON interval. The delays came down to reasonable values only at 300-packet bucket size. The number of premium packets dropped at the policer in the border router increases for decreasing bucket depth values at the leaf router. The reason for this behavior is given as

due to the jitter introduced at the re-shaper in the leaf router and the low bucket depth values in the border router as initially they were configured considering the smooth CBR flows.

In the fourth scenario for the same original model the premium load was increased from 10% to 50% to study the bandwidth reservations. It was observed that both premium and assured got their expected bandwidth while the rest was allocated to best effort traffic. Which was some what contradicting as earlier it was told that both assured and best effort traffic are queued in the same lower priority RIO queue than premium queue, which should result higher bandwidth values for best effort and lower bandwidth values for assured compared to what was given. The delay values were given to increase from 0.93 msec. to 1.18 msec. and 100 msec. to 200 msec. when the load was increased. It should be noted that 200 msec. is the maximum delay bound for assured voice traffic which was reached at 50% premium load.

3.4.3 Conclusions

- One main conclusion that can be obtained is that shaping of premium traffic can result in poor results. So we can conclude to drop premium packets rather than shape them.

Drawbacks

- The results obtained from other scenarios, apart from fourth scenario, were mostly obvious.
- The transport protocol used by sources is not given, it can be assumed that they have used UDP atleast for premium and assured sources from the results obtained.
- Results were not obtained with the links overloaded.

3.5 Impact of Different Classes and Drop Precedence on TCP connections (especially for ACK packets)

The impact of different drop precedence in a class was investigated in [23] and is discussed in this section. To simulate their network they have used the network simulator (ns) [16]. They modified ns to include traffic conditioners, multi-color RED and RIO queues.

3.5.1 Network Topology

As shown in Figure 3.5 the network topology consists of five nodes each node is connected to the router over a 10 Mbps link. All the nodes are connected to 10 TCP

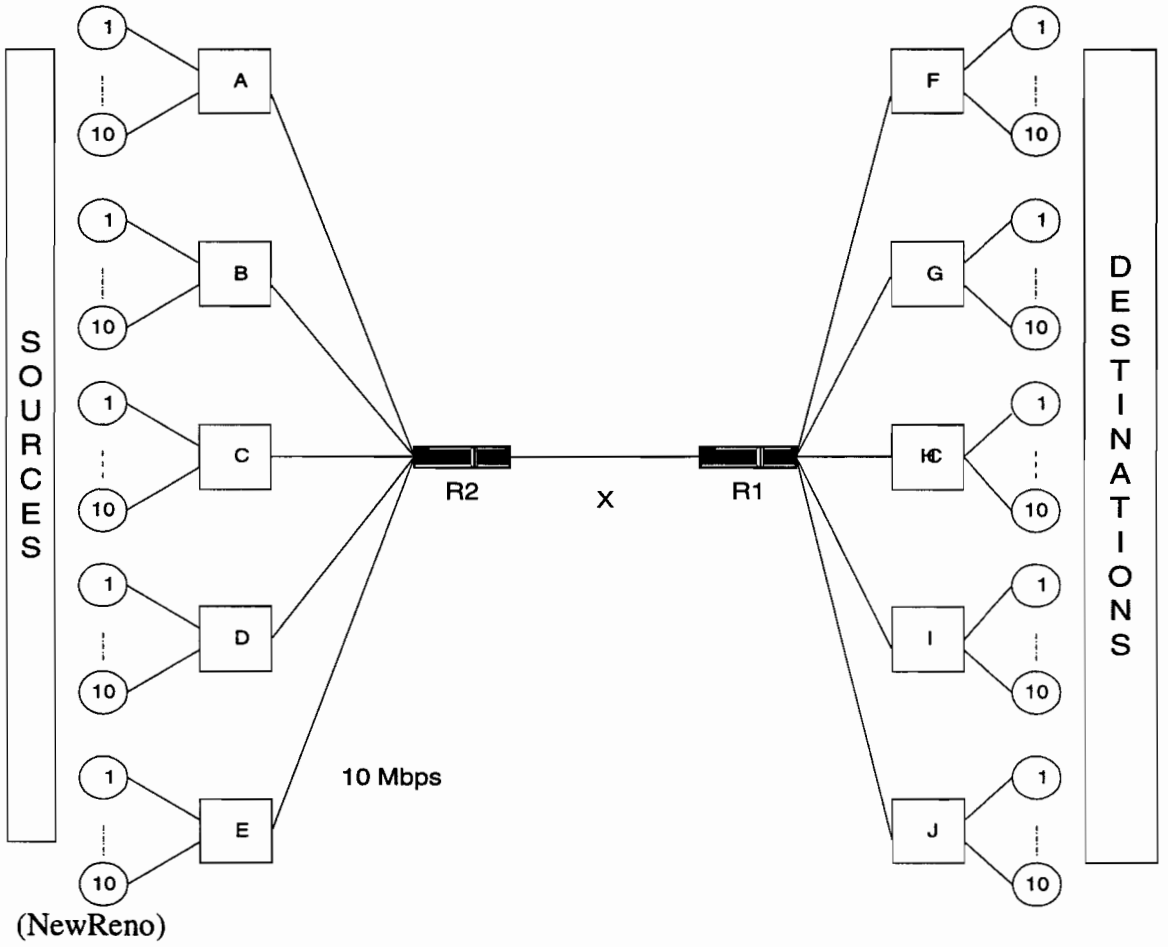


Figure 3.5: Network Topology.

sources sending FTP traffic. Two of the sources send Premium /Green traffic (20%), three of them assured / yellow (30%) and five sources send Best Effort / Red (50%). The link between the routers R1 and R2 is at load 1.0 when the bandwidth is 50 Mbps. The bandwidth of this ink is changed to simulate bottleneck. Only few connections were observed traffic from the rest of the connections was considered as background traffic. The background TCP connections send ACKs of the same class (drop precedence) as the data. Only the examined connections use different classes and drop precedences for data and ACKs.

They have investigated two models:

Model 1: Two-Bit Differentiated Services Architecture [14]:

This model was suggested in paper [14]. It contains three classes Premium, Assured and Best Effort. In this scenario the Impact of different classes for data and ACKs in TCP connections was studied. The nodes were configured with two token buckets to condition Premium and Assured traffic. The premium packets are dropped and assured packets are remarked to best effort if they exceed their contracted rate. The nodes and the routers are equipped with RIO queues to treat Assured and Best Effort traffic.

Model 2: Two rate three color marking with three drop precedence:

In this model all the traffic belongs to same class, the difference is that they belong to different drop precedence. The nodes are configured with a two rate three-color Marker (trTCM) [10] to control the color and n-RED queue (with n different thresholds and drop probability for n different drop precedence). The routers are also configured with an n-RED queue. In this scenario the effect of different drop precedence was studied.

3.5.2 Parameters of Study

Delay and RTT: The simulations were run for different RTT values. Table 3.4 shows all the scenarios that were considered. The simulations were also run for same and three different delay values for the links as shown in Table 3.4. The three different delay values would result in three different RTT values for the sources connected to the nodes. In the

Table 3.4 case 2 represents the case of three different dealy avlues, the link of first node to the router has a delay of 3 ms, the second node to the router has delay of 23 ms, the third node to the router has a delay of 3 ms and so on.

Case	Node to Router (ms)	R1 to R2 (ms)	Router to destination(ms)	Average RTT (ms)
1	3	4	3	20
2	3-23-3-23-48	4	3-23-3-23-48	20
3	10	10	10	60
4	15	20	15	100
5	23-48-3-23-48	4	23-48-3-23-48	
6	30	40	30	200
7	48-23-3-23-48	4	48-23-3-23-48	200

Table 3.4: Simulated delays and RTTs.

Load: The bandwidth of the link from the sources to the nodes is 10 Mbps. The bandwidth of the link between the R1 and R2 is changed as shown in Table 3.5. The bandwidth was changed as it is difficult to define load for TCP connections and because TCP adapts its window size to the available resources.

Load	0.5	1.1	1.11	1.12	1.13	1.14
Bandwidth (Mbps)	100	50	45.5	42	38.5	36

Table 3.5: Simulated loads of the bottleneck link

Traffic Conditioners and Queue Management: The parameters for traffic conditioner s are configured considering the fact that the network sees 20% of the Premium/Green (AF11), 30% of Assured/Yellow (AF12) and 50% of Best Effort/Red traffic. The following tables show the parameters of all the components,

Premium CIR	250 KB/s = 2 Mbps
Premium CBS	50 KB
Assured CIR	375 KB/s = 3 Mbps
Assured CBS	70 KB

Table 3.6: Scenario 1,Token Bucket parameters

AF11 CIR	250 KB/s = 2 Mbps
AF11 CBS	50 KB
AF12 CIR	375 KB/s = 3 Mbps
AF12 CBS	70 KB

Table 3.7: Scenario 2, trTCM parameters

Minout/ASSURED	35
Maxout/ASSURED	50
Pout/ASSURED	0.1
Minin/PREMIUM	55
Maxin/PREMIUM	65
Pin/PREMIUM	0.05
Wq	2.00E-003
Queue limit	90

Table 3.8: Scenario 1, RIO parameters

Minred	35
Maxred	50
Pred	0.3
Minyellow	45
Maxyellow	60
Pyellow	0.2
Mingreen	60
Maxgreen	70
Pgreen	0.1
Wq	2.00E-003
Queue limit	90

Table 3.9: Scenario 2, n - RED parameters

3.5.3 Simulation Results

The results are collected for different Round Trip Times and for different loads as given in Table 3.4 and Table 3.5 respectively for both the scenarios.

3.5.3.1 Two-Bit Differentiated Services

This model is used to see the Impact of different classes for data and ACKs in a TCP connection. Notation: XY connection, where X is data type and Y is ACK type. (e.g. PP represents Premium data and Premium ACK, PA represents Premium data and Assured ACK and PB represents Premium data and Best Effort ACK).

It was seen the PP connections were independent of load. PA and PB connections were seen to have lower throughputs than PP connections because of the higher delay and the probability of ACK getting lost in the RIO queue. Thus even in Premium class the appropriate choice of ACKs is necessary. The choice of ACK was observed to be more evident in case of Assured data. The throughput of AP connections was observed to be 140% higher than AB connections. The Premium and Assured are protected by the mechanisms that were used as pointed out earlier.

It has also been observed that for higher RTT values the influence of classes is reduced. The variation of throughput for Assured and Best Effort at different loads is significantly higher than Premium due the mechanisms provisioned in the network. At low loads it can be seen that the class of ACKs is a significant factor but for higher load it has been observed that the ACK's class get less significant

3.5.3.2 Three Color Marking with Three Drop Precedence (Impact of different drop precedence in a class)

In this case the influence of different drop precedence in an AF class was investigated. The simulation was kept same only the all the classes were changed to drop precedences, so that the whole traffic belonged to a single AF class.

The throughput was observed to depend only on the drop precedence of the data and not on the drop precedence of the ACKs expect at higher loads. For high RTT values the throughput was observed to be independent of drop probabilities expect at higher loads. At higher loads the lower drop probability connections are protected at the expense of higher drop probability connections. It was seen only for low high RTT values and high load the drop precedence of the ACKs influenced the throughput.

3.5.4 Conclusions

- It has been concluded that the throughput of a TCP connection in a DS network does not only depend on the sender but also receiver, the appropriate choice of ACK 'Class' has influence on the throughput.
- It also has been shown that the use of different drop precedence within a class for ACKs has no influence on the performance. The drop precedence of data of a particular class has effect only for high loads and low round trip times.

3.6 Lessons Learned

The following lessons are learned from the studies investigated in the previous sections,

- The Assured Service defined in [8] does not allow a strict allocation of bandwidth for users.
- There is a dependence on RTT's and the window sizes of TCP connections on the performance.
- In [18] it was shown that RIO and Two-Bit schemes have were unfair in some cases in allocating the bandwidth, which were overcome by using, USD scheme to provide differentiated services. But it has to be noted that USD scheme is not scalable.
- The assured service class was analyzed in [20], which showed that three-drop precedences performed better than two-drop precedences especially concerning the fairness.
- It was also learned from [22] shaping the premium flows resulted in poor performance.

- It was learned that the class of ACK packets has considerable effect on the TCP connection's performance.
- The models that are considered in these studies do not represent a carrier network in particular the traffic models.
- The end-to-end characteristics like end-to-end delay and jitter, which were not considered, are considered in the next two chapters.
- Most of the studies investigated only a particular scheme or answered a particular problem. Our studies investigated the problems that can occur and schemes that cab be used in deploying DiffServ into the networks.
- The impact of overbooking the network on service classes was investigated. The impact of assigning same class to TCP and UDP flows was studied. The performance achieved by sources when changed from UDP to TCP is compared. The impact of queuing flows, with different traffic characteristics, in same queue is studied.
- The impact of number of drop precedence in a class was studied but the impact of number of DiffServ classes (that a service provider can offer) on the performance is studied.
- Scheduling schemes were not discussed in the previous work as most of the studies evaluated a single class. Performance of FIFO and DRR schemes is compared when they are used in the customer. Priority and DRR scheduling schemes were used to provide differentiation among classes.

Chapter 4

Overbooking Study

The purpose of this study is to determine the impact of overbooking on throughputs and end-to-end delay characteristics, i.e., average and standard deviation of end-to-end delays in a DiffServ network. Overbooking will be achieved by increasing the number of customer sites served by a single provider edge router. The number of customer sites will be varied from one to five. At the same time we also studied the effect of queuing traffic of two different characteristics in the same queue under different cases.

4.1 Network Architecture

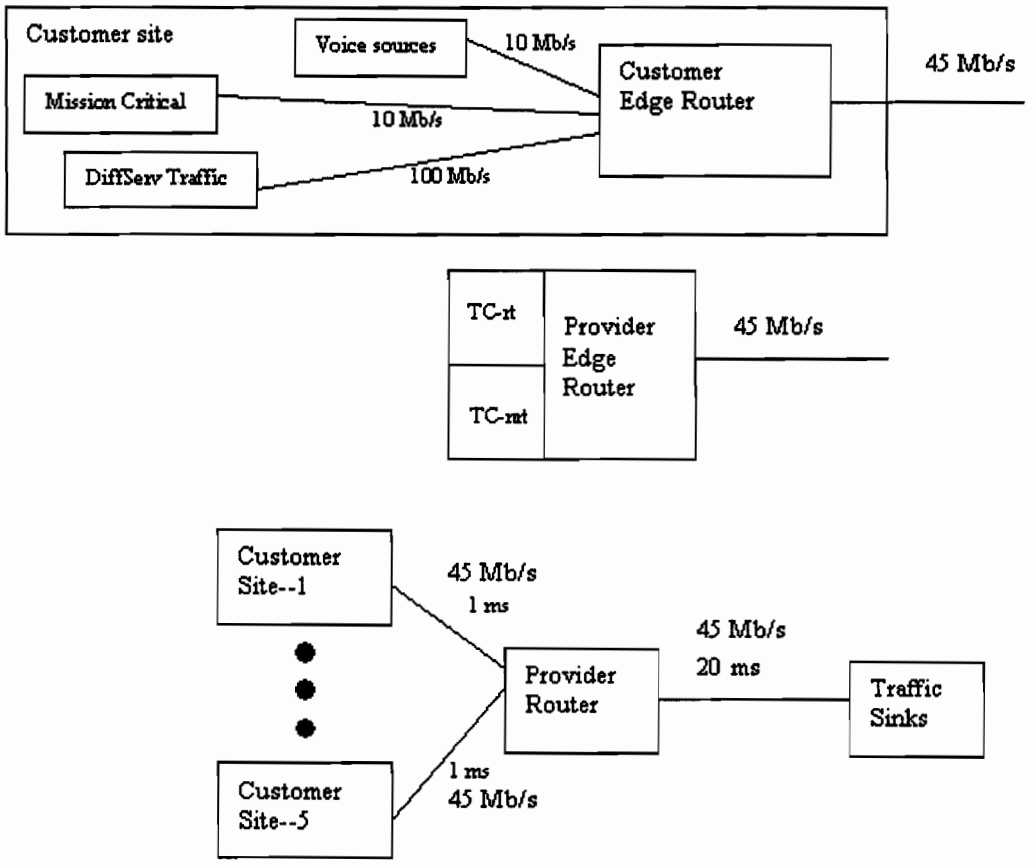


Figure 4.1: Network Architecture

Figure 4.1 shows the network architecture that has been used for the study. Each customer site contains one aggregate source of 80 voice (AF1) sources, one mission critical (AF1) source and one DiffServ (AF2) source. The traffic from the sources reaches

the sinks thorough the customer edge and provider edge router. All the sources in a site are connected to a customer edge router, which is connected to the provider edge router using a DS3 link.

The customer edge routers uses FIFO queuing scheme and Deficit Round Robin (DRR) scheduling scheme to queue and serve the packets respectively at the outgoing link. The provider edge router uses Weighted. Random Early Drop (WRED) queuing scheme and DRR scheduling scheme to queue and serve the packets respectively at its outgoing link.

4.2 Experimental Parameters

4.2.1 Traffic Model

Aggregate of 80 Voice sources - AF1x (x = 1 or 2)

64 bytes/packet - Fixed length

10000 packets/sec - Fixed interarrival times (80 sources)

Protocol UDP

Aggregate rate = 5,120 kb/s

Customer router access rate = 10 Mb/s

Mission Critical (MC) Traffic - AF11

1500 bytes/packet - Exponential length

400 packets/sec - Exponential interarrival times

Protocol TCP

Average rate = 4,800 kb/s

Customer router access rate = 10 Mb/s

DiffServ Class - AF21

1500 bytes/packet - Exponential length

1500 packets/sec - Exponential interarrival times

Protocol TCP

Average rate = 18,000 kb/s

Customer router access rate = 100 Mb/s

Total A1 traffic = 9,920 Kb/s. Total AF2 traffic = 18,000 Kb/s. Total traffic/site = 27,920 Kb/s (62 % of a DS3).

The traffic generated includes the TCP/IP overhead.

4.2.2 TCP parameters

The TCP parameters were configured meticulously to obtain good throughput values for the sources.

Parameter	Value
Fast Retransmit/Recovery	Enabled
Retransmission Timeout	Karns Algorithm
Retransmission Thresholds	15 attempts
Nagle SWS Avoidance	Disabled
Window Scaling Option	Enabled
SACK option	Disabled

Table 4.1: TCP Parameters.

TCP maximum segment size was configured to 64 Bytes for voice sources and 1500 bytes for mission critical and DiffServ sources, these values were choose to preserve the packet size sent by the application layers. The maximum receive buffer size (window size) was configured according to RTT * (Offered Throughput value), so as to obtain good throughput and low drop rate values.

4.2.3 DRR Parameters

DRR is used by both customer and provider edge routers to serve the packets at their outgoing links.

Customer edge router weights
AF1 11,250 kb/s
AF2 33,750 kb/s

Table 4.2: Scheduling weights for the customer edge router.

Table 4.2 shows the weights for each traffic class in customer edge router. These weights are used to serve the traffic outgoing from the CE router to PE router on the DS3 link. The load on this link is 0.62 for all customer sites. Therefore there is not significant discrimination between the classes in the customer edge router.

Provider router weights: 1 customer site	**Provider router weights: 2 customer sites**	**Provider router weights: 3 customer sites**	**Provider router weights: 4 customer sites**	**Provider router weights: 5 customer sites**
AF1 11,250 kb/s	AF1 22,500 kb/s	AF1 33,750 kb/s	AF1 45,000 kb/s	AF1 45,000 kb/s
AF2 33,750 kb/s	AF2 22,500 kb/s	AF2 11,250 kb/s	AF2 0 kb/s	AF2 0 kb/s

Table 4.3: Scheduling weights at the provider edge router.

Table 4.3 shows the weights used in the provider edge router to serve each traffic class. As it can be seen that as the number of the sites increases the weight assigned to AF1 class is increased and the weight assigned to AF2 is decreased proportionally. It has to be noted that the total weight is equal to 45 Mbps. Therefore AF2 weight is decreased so that AF1 does not experience congestion. There is only one case where even AF1 queue is also overloaded, i.e., when the number of sites was increased to five.

Figure 4.2 shows that the weights are assigned in such a way that the load generated by AF1 traffic remains constant 88% till number of sites are four. The total load on the link increases in steps of 0.6 as the number of sites increases. The load experienced by AF2 traffic increases sharply, as the weight assigned to this class is decreased as the number sites or AF2 traffic increases.

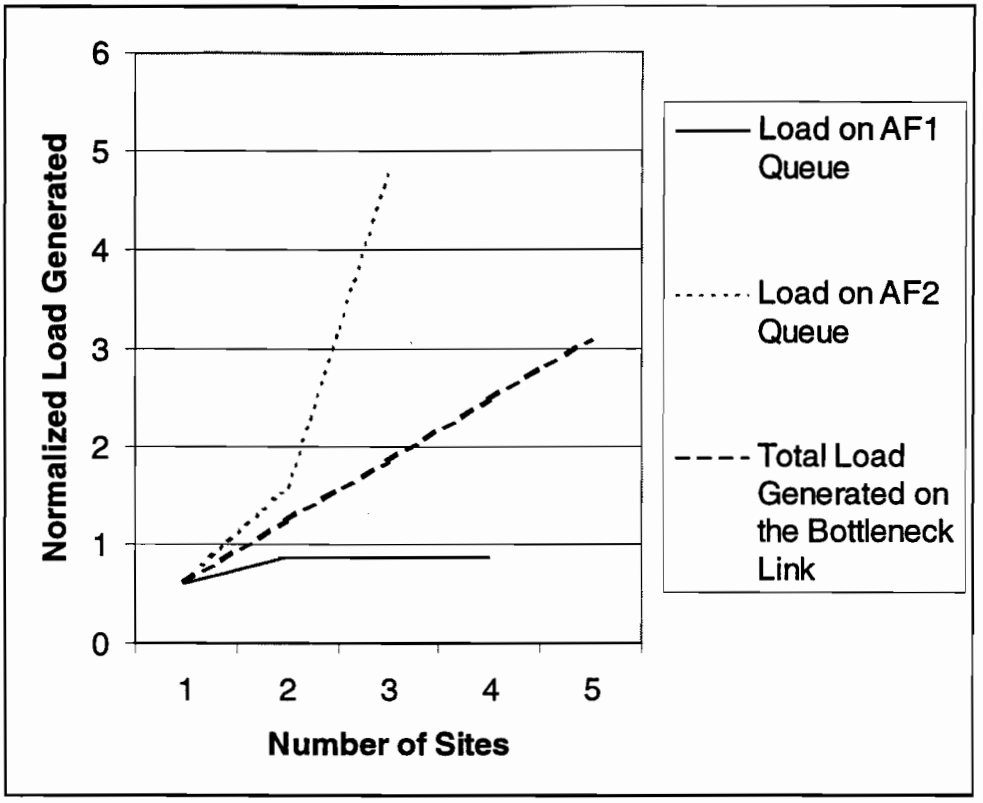


Figure 4.2: Normalized Load Vs Number of Sites.

4.2.4 WRED Parameters

We have two classes AF1 and AF2 each with three-drop precedence classes. For dropping AFk3 (k=1 or 2) packets the average queue length (in packets) will be estimated based on all packets in the queue. For dropping AFk2 (k=1 or 2) packets the average queue length (in packets) will be estimated based on AFk2 and AFk1 (k=1 or 2) packets in the queue. While for dropping AFk1 (k=1 or 2) packets the average queue length (in packets) will be estimated based on only the average number of AFk1 (k=1 or 2) packets in the queue.

The exponential weight, alpha, used in estimating the average length will be set to 0.005 for both AF1 and AF2. Note that Floyd and Jacobson [4] recommend that alpha > 0.001. Setting alpha to a value depends on lot of factor like load on the queue, amount of input traffic coming in etc. [13] [14]. For our network configuration it was found that 0.005 is a good value for alpha, as the average queue size calculated by the algorithm followed more closely to the queue size. Floyd and Jacobson [4] also recommend that

maxth > 2minth, they also observed that the drop probability p is about maxp when the average queue size is halfway between minth and maxth.

We configured the maxth value to three times the minth value because RED drops all the packets coming in, when maximum threshold is reached. Therefore maxth value was set high so that sources can see enough drops before filling the queue to the maximum threshold and also considering the fact the in most simulations the queues were overloaded. Our RED algorithm implementation considers number of bytes instead of number of packets in the queue. Since the size of packets being queued in AF1 queue are not the same it has been observed that using RED in terms of number of packets resulted in unfair behavior for packets of larger lengths. It has been observed that for a given queue length and threshold values, the probability of larger packet being dropped are higher than the probability of smaller packet being dropped. Considering the traffic model and by performing some initial simulations runs, the following WRED parameters were considered selected.

Class	Minimum Threshold(bits)	Maximum Threshold(bits)	Max. Drop Prob.	Traffic Type using this class
AF11	990,769.23	3,821,538	0.02	Voice n MC
AF12	1,486,153	3,821,538	0.05	Voice
AF13	1,910,769	3,821,538	0.1	-
AF21	840,000	2,160,000	0.02	DiffServ
AF22	960,000	2,160,000	0.05	-
AF23	1,080,000	2,160,000	0.1	-

Table 4.4: WRED Parameters

4.3 Scenarios

The basic network architecture is shown in Figure 4.1, with this given architecture the number of customer sites attached to Provider Edge (PE) Router are changed from one to five, to study the effect of overbooking the link on end-to-end characteristics of traffic. So essentially five scenarios were simulated. For each scenario two different possibilities for voice traffic are considered.

The voice source considered here is similar to a CBR (Constant Bit Rate) source while the mission critical source is similar to VBR (Variable Bit Rate) source, traffic from both the sources is being processed in the same queue. So the effect of changing the voice sources transport protocol and drop precedence on mission critical traffic was also studied. Three cases as shown in Table 4.5 were run to see how each one of them effects the performance.

Case	Transport Protocol for Voice	Drop Precedence for Voice
1	UDP	AF11
2	UDP	AF12
3	TCP	AF11

Table 4.5: Different configurations of Voice sources.

So in total fifteen simulation experiments were considered, five scenarios and each scenario consisting of three cases for voice. It has to be noted that when voice is configured, as TCP it can be assumed that traffic belongs to a Real Time application not necessarily voice, as it is hard to see voice-using TCP. It might be noted that the case with voice (Real Time) as TCP source and with drop precedence as AF12 is not considered, the reason for this will become evident once the results for Case 3 are observed for different scenarios.

4.4 Performance Metrics

The performance metrics was measured at AF1 and AF2 queues in Provider Edge routers and at the destination of each source. The performance metrics that are collected as a function of the overbooking factor.

The analyzed metrics mainly represents the end-to-end performance. The metrics are average end-to-end delay per traffic class, i.e., source and DSCP marking, jitter (variance and standard deviation of delay) per traffic, source and DSCP marking and Throughput per traffic class and per source.

Jitter is defined by the following formula,

$$\text{Jitter} = (T_r(n) - T_r(n-1)) - (T_t(n) - T_t(n-1)).$$

Where $T_r(n)$ is time at which nth packet is received and $T_t(n)$ is the time at which nth packet is transmitted. Therefore if the total number of packets sent are t, then we would have (t-1) values for jitter, computing variance or standard deviation on these (t-1) data points would give a measure of jitter.

4.5 Simulation Results

4.5.1 One Site Scenario

In this scenario only one customer site is used. The simulations for one site scenario have used the network configuration shown in Figure 4.3.

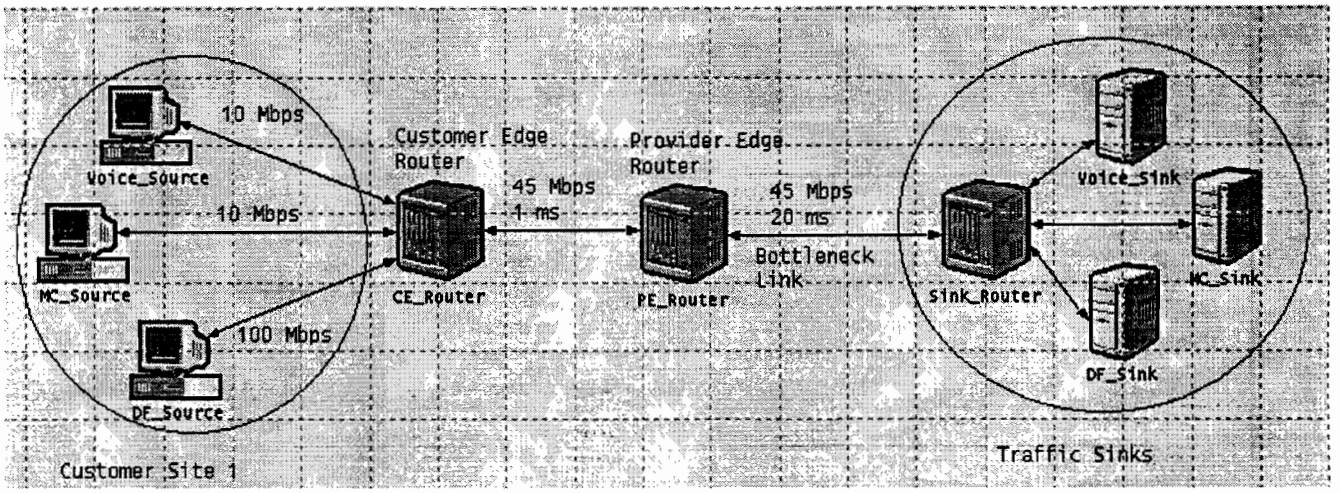


Figure 4.3: Network Architecture for One Customer Site Scenario

Traffic Type	Total Generation Rate	Scheduler weights
AF1	9.92 Mbps	11.25 Mbps
AF2	18.0 Mbps	33.75 Mbps
Voice	5.12 Mbps	-
Mission Critical	4.8 Mbps	-
DiffServ	18.0 Mbps	

Table 4.6: Traffic Generation Rates and Scheduler Weights for One Customer Site Scenario

As it can be seen that the links are not overloaded in this case. The sources should get throughput equivalent to the traffic generation rate. The results for all the three cases

are shown in Tables 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12. The following observations are made on the results obtained.

Observations on the AF1 and AF2 Queue Results: The queue results are the same for all the three cases, which is offered as the load did not change and the queues are not overloaded to treat traffic differently for the three cases under investigation. It has to be noted that when the configuration of voice traffic is changed, the voice packets will be treated differently only by WRED in the provider edge router. The WRED will not be triggered until the queue length exceeds the minimum threshold.

Observations on the End-to-end Results: For cases one and two, the delay values for mission critical traffic is more than voice even though they belong to same class. It has to be noted that the delays collected are between application layers, so the delay collected will also include the delay experienced in the TCP layers. Hence the higher delays for mission critical traffic is due to the delay experienced in TCP layer and also due to the larger packet sizes. By changing the voice sources to TCP in Case three we noticed increase in end-to-end delay because of the delay experienced by packets in the TCP layer. That also explains the lower delays experienced by voice sources when they were using UDP when compared TCP mission critical sources. It can also be seen that each source got its expected throughput.

Case 1: AF11 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	3.77	4836	0.185	0%	9.94 Mbps	9.92 Mbps
AF2	1.32	7458	0.238	0%	17.87 Mbps	18.0 Mbps

Table 4.7: Case 1, Statistics Collected at Output Queue of the Provider Edge Router.

Table 4.7 shows the results collected at AF1 and AF2 queue in the Provider Edge router. It can be seen that both types of aggregates are getting their offered throughput.

Traffic Type	ETE(msec)* Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice	75.92	4.189e-07	1.184e-06	0.00108	5.108	5.12
MC	113.54	0.0001839	3.2e-05	0.00565	5.05	4.8
DiffServ	110.57	9.235e-05	3.95e-06	0.00198	18.16	18.0

ETE (msec)- End-to-end Delay collected between the application layers of Source and Destination.

Table 4.8: Case 1, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. It can be seen that the throughput achieved is almost equivalent to the traffic generation rate or the offered throughput.

Case 2: AF12 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF12.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	4.54	5289	0.189	0%	9.834 Mbps	9.92 Mbps
AF2	0.50	6149	0.273	0%	17.65 Mbps	18.0 Mbps

Table 4.9: Case 2, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. It can be seen that both types of aggregates are getting their offered throughput and that their not much difference from case one as the queues are not overloaded.

Traffic Type	ETE(msec)* Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice	80.38	5.75e-07	2.48e-06	0.00157	5.108	5.12
MC	136.60	0.00029	6.72e-05	0.0082	4.88	4.8
DiffServ	147.93	0.00016	9.68e-06	0.00311	17.625	18.0

Table 4.10: Case 2, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. Changing the marking of voice packets from AF11 to AF12 did not showed much difference, as the network is not overloaded.

Case 3: AF11 TCP Voice Source

In this case all the voice sources were configured to use TCP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	4.073	5317	0.194	0%	10.155 Mbps	9.92 Mbps
AF2	1.332	7482	0.239	0%	18.136 Mbps	18.0 Mbps

Table 4.11: Case 3, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. It can be seen there is no significant difference from the above cases one and two.

Traffic Type	ETE(msec)* Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice	155.50	0.113	4.56e-05	0.00675	5.108	5.12
MC	113.69	0.000185	3.54e-05	0.00595	5.198	4.8
DiffServ	110.30	9.91e-05	4.74e-06	0.00217	18.104	18.0

Table 4.12: Case 3, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. It can be seen that changing the voice source to TCP has increased the end-to-end delay for voice source for the reasons explained in the observations.

4.5.2 Two Sites Scenario

In this scenario two customer sites are considered. The simulations for two sites scenario have the network configuration shown in Figure 4.4.

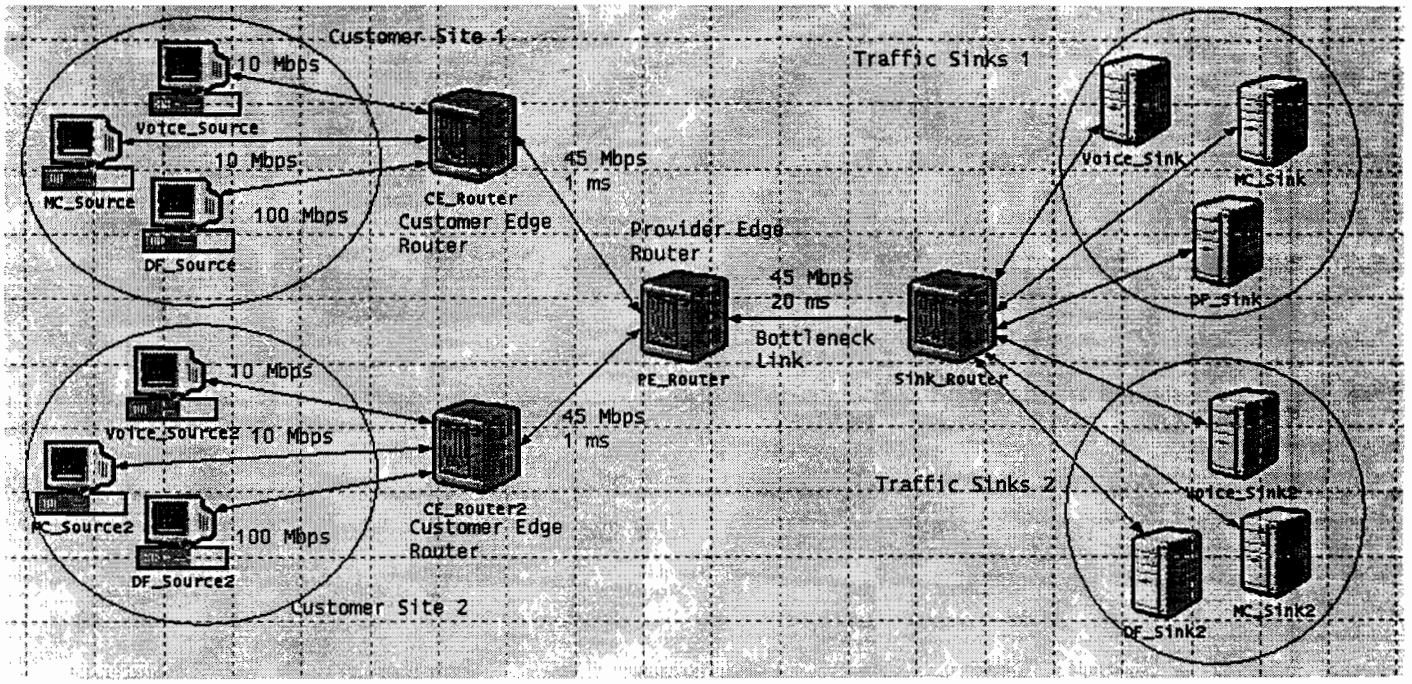


Figure 4.4: Network Architecture for Two-Customer Sites Scenario

Traffic Type	Total Generation Rate	Scheduler weights
AF1	19.84 Mbps	22.50 Mbps
AF2	36.0 Mbps	22.50 Mbps
Voice	2*5.12 = 10.24 Mbps	-
Mission Critical	2*4.8 = 9.6 Mbps	-
DiffServ	2*18.0 = 36.0 Mbps	

Table 4.13: Traffic Generation Rates and Scheduler Weights for Two-Customer Sites Scenario

The bottleneck link as shown in Figure 4.4 is overloaded and it is configured such that the throughput achieved by AF1 traffic will be transparent from the congestion, as the scheduler weights at the bottleneck are configured such that the load experienced by AF1 traffic is 88%. It is configured such that AF1 get a throughput of 19.84 Mbps and AF2 traffic gets the rest of the DS3 link bandwidth. It has to be noted that the scheduler weight for AF1 queue is 22.5 Mbps and AF2 traffic is allocated 22.5 Mbps of the

bottleneck link, so AF2 should get a throughput of 22.5 Mbps plus excess bandwidth allocated to AF1, i.e., (22.5 - 19.84) 2.66 Mbps. So the offered throughput for AF2 queue is 25.16 Mbps. The results for all the three cases are shown in following tables.

Case1: AF11 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	402.15	498312	24.90	0%	20.13 Mbps	19.84 Mbps
AF2	68.79	530308	18.58	0%	23.81 Mbps	25.16 Mbps

Table 4.14: Case 1, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. It can be seen that both types of aggregates are getting their offered throughput.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	97.398	7.164e-07	4.353e-06	0.00208	5.105	5.12
MC-1	165.803	0.0002001	9.472e-05	0.00973	5.0	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	12.04	12.58
Voice - 2	115.868	3.97e-07	4.041e-06	0.00201	5.098	5.12
MC-2	280.615	0.000325	9.637e-05	0.00981	4.89	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	11.73	12.58

N/A* - Queue overloaded.

Table 4.15: Case 1, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. The sources are getting the offered end-to-end throughput. Since the AF2 (DiffServ) traffic experiences overloaded bottleneck the end-to-end delay characteristics for these sources are large. It can also be observed that the throughput achieved by mission critical sources is higher than offered throughputs due to TCP fragmentation.

Case 2: AF12 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF12.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	407.55	505352	22.55	0%	20.23 Mbps	19.84 Mbps
AF2	74.31	587054	19.29	0%	23.3 Mbps	25.16 Mbps

Table 4.16: Case 2, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. In the above table we can see that both the queues are getting their share bandwidth. Again there will not be much difference between this case one and case two as RED is still not triggered in AF1 queue.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	97.15	4.94e-07	4.916e-06	0.00221	5.098	5.12
MC-1	197.00	0.000209	9.614e-05	0.00980	5.047	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	12.0	12.58
Voice - 2	113.44	4.861e-07	4.748e-06	0.00217	5.096	5.12
MC-2	590.82	0.00039	0.000101	0.01006	4.95	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	11.8	12.58

N/A* - Queue overloaded.

Table 4.17: Case 2, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. Even though the DiffServ sources got their offered throughput the delays were high as the generated load is higher than offered load and the TCP windows were configured according to offered load. Therefore packets experienced large TCP delays and hence larger end-to-end delays.

Case 3: AF11 TCP Voice Sources

In this case all the voice sources were configured to use TCP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	183.2	236970	10.13	0	20.43 Mbps	19.84 Mbps
AF2	116.17	325028	10.29	4.16e-3%	16.73 Mbps	25.16 Mbps

Table 4.18: Case 3, Statistics Collected at Output Queue of the Provider Edge Router.

It can be seen that the AF2 traffic was unable to achieve the offered bandwidth, as there were few drops. Because of the drops the AF2 TCP sources had to slow down thereby achieving lower throughputs.

The table below show results collected at traffic sinks dedicated for each source. Both the AF2 sources have experienced packet drops, therefore the low throughput values.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	169.07	4.48e-05	4.014e-06	0.002003	5.109	5
MC-1	179.86	0.00025	4.92e-05	0.00701	5.01	
DiffServ-1	N/A*	N/A*	N/A*	N/A*	6.9	
Voice - 2	198.65	4.14e-06	5.035e-05	0.00709	5.097	
MC-2	342.35	0.000352	5.85e-05	0.00765	5.16	
DiffServ-2	N/A*	N/A*	N/A*	N/A*	9.75	

N/A* - Queue overloaded.

Table 4.19: Case 3, Statistics Collected at Traffic Sinks (End-to-end

4.5.3 Three Sites Scenario

In this scenario two customer sites are considered. The simulations for three sites scenario have used the following network configuration.

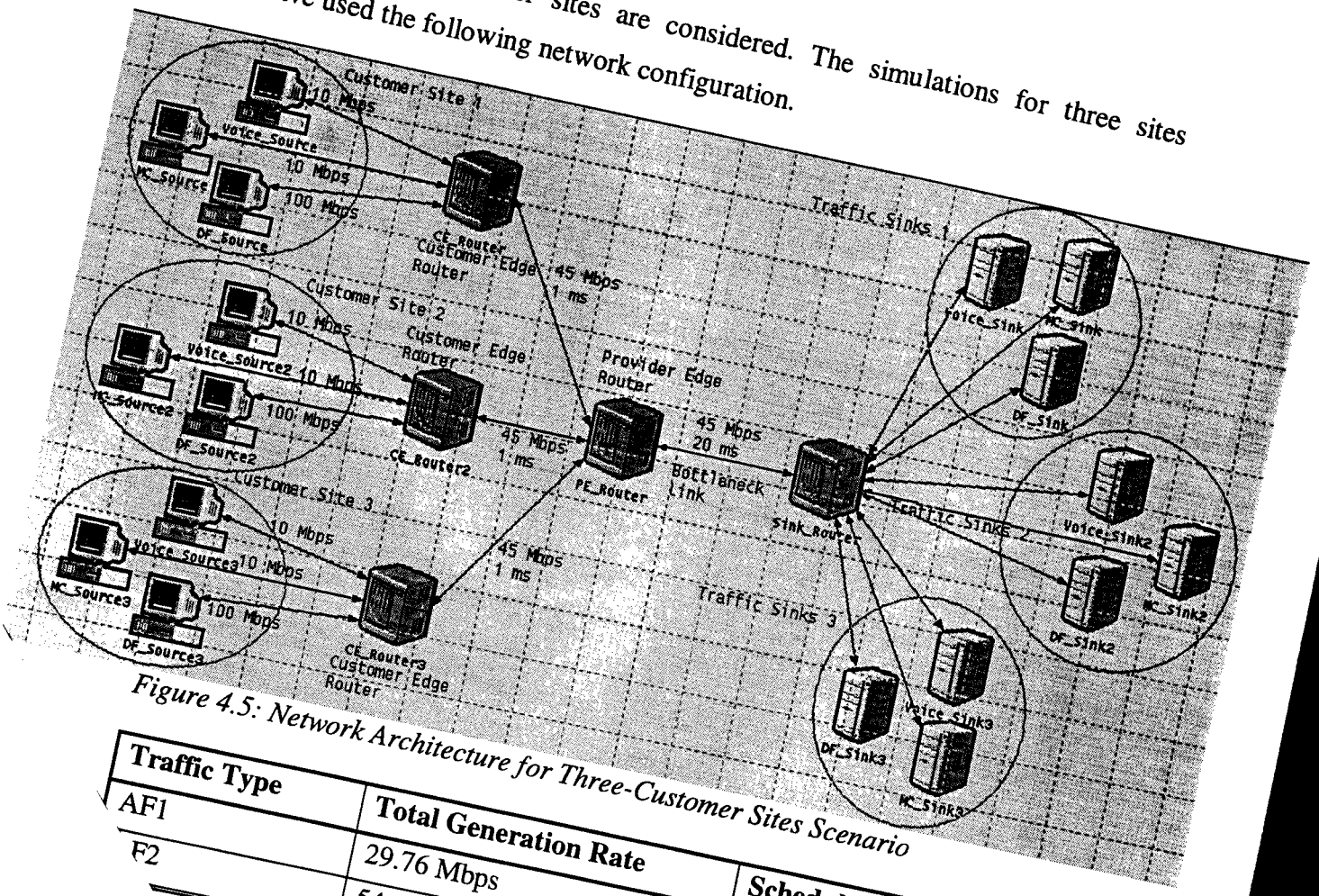


Figure 4.5: Network Architecture for Three-Customer Sites Scenario

Traffic Type	Total Generation Rate	Scheduler weights
AF1	29.76 Mbps	33.75 Mbps
F2	54.0 Mbps	11.25 Mbps
	3*5.12 = 15.36 Mbps	-
T Critical	3*4.8 = 14.4 Mbps	-
	3*18.0 = 54.0 Mbps	

tion Rates and Scheduler Weights for Three-Customer Sites Scenario.

traffic is

throughput

3.99 Mbps.

uler weight for AF1 queue is 33.75 Mbps and AF2

tleneck link, so it is offered that AF2 gets a

idth allocated to AF1 i.e. (33.75 - 29.76)

ue is 15.24 Mbps.

Case 1: AF11 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	375.73	473776	15.81	0	30.09 Mbps	29.76 Mbps
AF2	119.61	570053	29.50	0.0653%	13.25 Mbps	15.24 Mbps

Table 4.21: Case 1, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. The throughputs achieved are almost equivalent to the Offered throughputs. There were few drops in AF2 queue as the queue is overloaded resulting in lower throughputs than expected.

Table 4.22 shows the results collected at traffic sinks dedicated for each source. The AF1 sources results were identical to offered results. The AF2 packets that were lost belonged to DiffServ - 2 and DiffServ - 3 sources, as it is evident from the low throughput values they achieved. It is interesting to note that because of the overloaded situation DiffServ - 3 source was badly effected while DiffServ - 1 source was not effected much. DiffServ -1 source actually did not lose any packets and was able to obtain the excess bandwidth lost by other two DiffServ sources.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	83.65	4.59e-07	1.77e-06	0.00133	5.103	5.12
MC-1	163.70	0.000266	5.59e-05	0.00747	4.98	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	6.14	5.08
Voice - 2	96.58	2.63e-07	1.67e-06	0.00129	5.1006	5.12
MC-2	883.40	0.000518	9.067e-05	0.00952	4.88	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	4.36	5.08
Voice - 3	104.89	-4.88e-07	1.692e-06	0.0013	5.098	5.12
MC-3	368.999	0.000436	8.137e-05	0.00902	5.044	4.8
DiffServ-3	N/A*	N/A*	N/A*	N/A*	2.88	5.08

N/A* - Queue overloaded.

Table 4.22: Case 1, Statistics Collected at Traffic Sinks (End-to-end Statistics).

Case 2: UDP Voice Sources as AF12

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF12.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	285.23	367666	11.8842074	0	30.3 Mbps	29.76 Mbps
AF2	73.79	384320	23.8260878	7.24e-3%	13.25 Mbps	15.24 Mbps

Table 4.23: Case 2, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. The results are almost similar to case one. As the AF1 queue is still not overloaded to see any drops so as to see the effect changing the drop precedence of voice traffic. The only difference is that the number of AF2 packets lost is lesser when compared to the case where UDP sources have the same drop precedence as the TCP sources, the reason for this becomes clear by examining the table below.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	79.52	4.989e-07	1.367e-06	0.00116	5.099	5.12
MC-1	110.84	0.000211	3.468e-05	0.00588	5.01	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	6.52	5.08
Voice - 2	92.12	2.289e-07	1.126e-06	0.00106	5.096	5.12
MC-2	474.07	0.000478	6.58e-05	0.00811	5.02	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	1.49	5.08
Voice - 3	101.05	-2.14e-07	1.31e-06	0.00114	5.093	5.12
MC-3	276.71	0.000352	6.17e-05	0.00785	5.06	4.8
DiffServ-3	N/A*	N/A*	N/A*	N/A*	5.66	5.08

N/A* - Queue overloaded.

Table 4.24: Case 2, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. The results are similar to those of case one, expect that in this case DiffServ - 2 source got lower throughputs as the AF2 packets lost belonged to this source. It is interesting to note that all the AF2 packets lost belonged to DiffServ -2 source i.e. lowers percentage of

drops of AF2 packets compared to case one. In this case DiffServ - 2 source was effected very badly because of the overload situation this gave a chance for DiffServ 1 and 3 sources to achieve better results.

Case 3: AF11 TCP Voice Sources

In this case all the voice sources were configured to use TCP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	362.21	471240	15.29	5.8e-4%	28.47	29.76 Mbps
AF2	100.51	483299	27.86	0.43%	13.85	15.24 Mbps

Table 4.25: Case 3, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. The interesting point to note in the above results is that even some AF1 packets were dropped. When the voice sources were changed to TCP, the burstiness of AF1 traffic was increased (it has to be noted that the voice sources generated small packets at a very high rate). So at some instances these TCP sources made the AF1 queue overloaded making RED to drop few packets, remember that the AF1 queue was configured with 0.88 load.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	163.87	4.81e-05	2.55e-06	0.00159	5.104	5.12
MC-1	N/A*	N/A*	N/A*	N/A*	4.122	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	5.21	5.08
Voice - 2	228.50	5.92e-05	3.25e-06	0.001804	5.0975	5.12
MC-2	N/A*	N/A*	N/A*	N/A*	3.98	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	3.1	5.08
Voice - 3	235.26	6.233e-05	3.555e-06	0.00188	5.0982	5.12
MC-3	164.63	0.000258	5.87e-05	0.00766	4.98	4.8
DiffServ-3	N/A*	N/A*	N/A*	N/A*	5.64	5.08

N/A* - Queue overloaded

Table 4.26: Case 3, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. It can be seen that all the AF1 packets lost belonged to mission critical sources 1 and 2. The throughputs achieved by AF2 sources are better in this case as they were able to capture the bandwidth lost by the Mission Critical sources 1 and 2.

4.5.4 Four Sites Scenario

In this scenario four customer sites are considered. To simulate four site scenario the network configuration shown in Figure 4.6 was used.

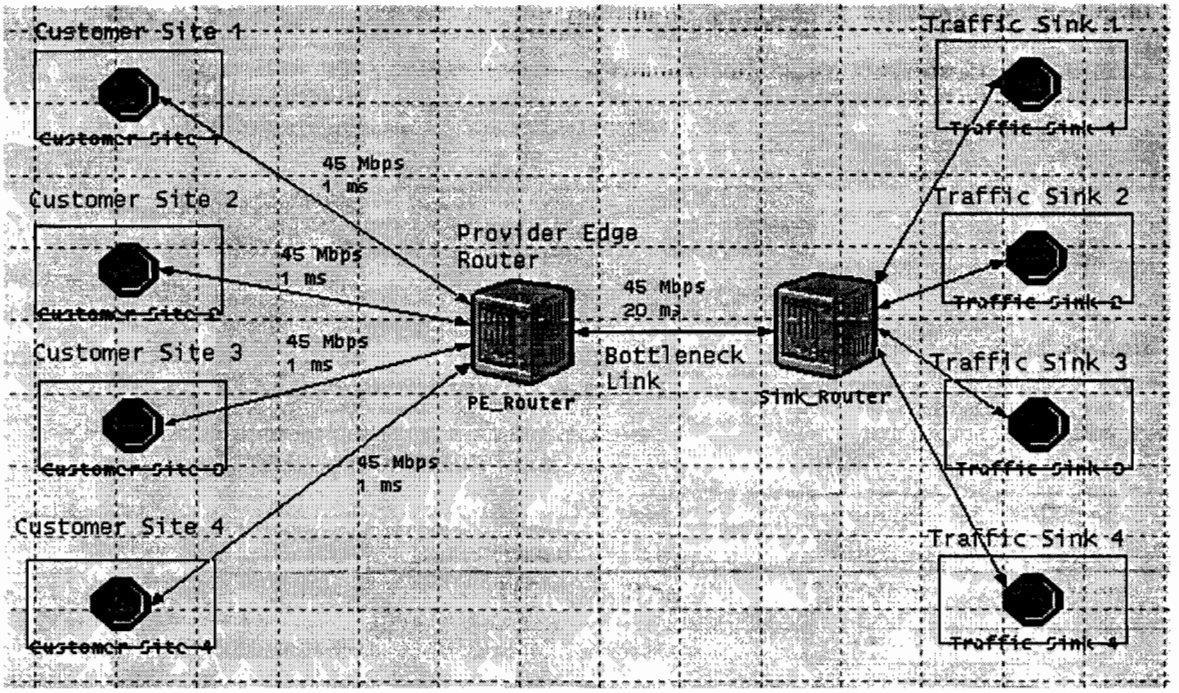


Figure 4.6: Network Architecture for Four-Customer Sites Scenario

Traffic Type	Total Generation Rate	Scheduler weights
AF1	40.68 Mbps	45.0 Mbps
AF2	72.0 Mbps	0.0 Mbps
Voice	4*5.12 = 20.48 Mbps	-
Mission Critical	4*4.8 = 19.2 Mbps	-
DiffServ	4*18.0 = 72.0 Mbps	

Table 4.27: Traffic Generation Rates and Scheduler Weights for Three-Customer Sites Scenario.

It has to be noted that the scheduler weight for AF1 queue is 45.0 Mbps and AF2 traffic is allocated 0 Mbps of the bottleneck link, so it is configured that AF2 gets a throughput of only the excess bandwidth allocated to AF1 i.e. (45.0 - 40.68) 4.32 Mbps. So the offered throughput for AF2 queue is 4.32 Mbps.

Case 1: AF11 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	50.21	68450	1.571	0	40.16 Mbps	39.68Mbps
AF2	237.63	993138	289.0	0.065%	3.64 Mbps	5.32 Mbps

Table 4.28: Case 1, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. The results shown follow the Offered results. The AF2 packets were dropped as the queue was overloaded.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	68.12	4.93e-08	3.418e-08	1.84e-4	5.104	5.12
MC-1	131.23	0.000229	1.766e-05	0.00420	5.0766	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	0.73	1.33
Voice - 2	76.063	3.38e-08	3.505e-08	1.87e-4	5.102	5.12
MC-2	245.58	0.000375	3.130e-05	0.00559	5.03	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	1.053	1.33
Voice - 3	96.35	6.76e-08	3.582e-08	1.89e-4	5.096	5.12
MC-3	470.49	0.000681	4.518e-05	0.00672	4.955	4.8
DiffServ-3	N/A*	N/A*	N/A*	N/A*	0.863	1.33
Voice - 4	82.84	3.98e-08	3.503e-08	1.87e-4	5.1003	5.12
MC-4	170.501	0.00044	3.21e-05	0.00566	5.08	4.8
DiffServ-4	N/A*	N/A*	N/A*	N/A*	0.95	1.33

N/A* - Queue overloaded

Table 4.29: Case 1, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. The achieved throughputs are almost equivalent to those expected.

Case2: UDP Voice Sources as AF12

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF12.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	52.62	71673	1.644	0	40.097 Mbps	39.68Mbps
AF2	205.61	1042240	293.41	0.058%	3.63 Mbps	5.32 Mbps

Table 4.30: Case 2, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. The results shown follow the Offered results. The AF2 packets were dropped as the queue was overloaded.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	67.04	5.02e-08	3.64e-08	1.909e-4	5.104	5.12
MC-1	119.78	0.000331	1.75e-05	0.00418	5.18	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	1.73	1.33
Voice - 2	76.02	3.38e-08	3.505e-08	1.87e-4	5.103	5.12
MC-2	192.92	0.000484	3.101e-05	0.00556	4.93	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	1.053	1.33
Voice - 3	97.82	5.305e-08	3.56e-08	1.88e-4	5.096	5.12
MC-3	387.90	0.0006006	5.52e-05	0.00743	4.91	4.8
DiffServ-3	N/A*	N/A*	N/A*	N/A*	0.51	1.33
Voice - 4	82.91	4.03e-08	3.44e-08	1.85e-4	5.068	5.12
MC-4	155.70	0.000395	2.53e-05	0.00503	5.08	4.8
DiffServ-4	N/A*	N/A*	N/A*	N/A*	1.08	1.33

N/A* - Queue overloaded.

Table 4.31: Case 2, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. The results are similar to case one.

Case 3: TCP Voice Sources

In this case all the voice sources were configured to use TCP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	473.60	399174	9.033	0.4%	11.646 Mbps	39.68Mbps
AF2	5.999	36158	8.983	0.0%	12.04 Mbps	5.32 Mbps

Table 4.32: Case 3, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. AF1 packets are dropped because the voice sources were changed to TCP. The AF1 traffic became burstier and there were instances when the AF1 queue was overloaded causing packet drops. The AF2 queue grabbed the bandwidth lost by AF1 queue and was able get better results than offered.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	968.91	0.0007503	0.01202	0.109	0.555	5.12
MC-1	547.51	0.00194	0.0373	0.331	0.705	4.8
DiffServ-1	N/A*	N/A*	N/A*	N/A*	3.58	1.33
Voice - 2	439.44	0.000312	0.00204	0.0451	1.084	5.12
MC-2	N/A*	N/A*	N/A*	N/A*	1.56	4.8
DiffServ-2	N/A*	N/A*	N/A*	N/A*	2.95	1.33
Voice - 3	187.55	6.77e-05	2.94e-06	0.0017	6.68	5.12
MC-3	1562.4	0.00808	0.1018	0.319	0.29	4.8
DiffServ-3	N/A*	N/A*	N/A*	N/A*	2.7	1.33
Voice - 4	1302.10	0.000877	0.00406	0.0637	0.59	5.12
MC-4	1162.18	0.0108	0.162	0.402	0.188	4.8
DiffServ-4	N/A*	N/A*	N/A*	N/A*	2.94	1.33

N/A* - Queue overloaded.

Table 4.33: Case 3, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. As it can be seen almost all AF1 sources experienced packet loses and as all these sources are TCP they backed off, thereby getting lower throughput and higher delay values. The AF2 sources took advantage of bandwidth lost by AF1 traffic and were able to obtain better throughput values.

4.5.5 Five Sites Scenario

In this scenario two customer sites are considered. The simulations for five sites scenario have used the shown network configuration in Figure 4.7.

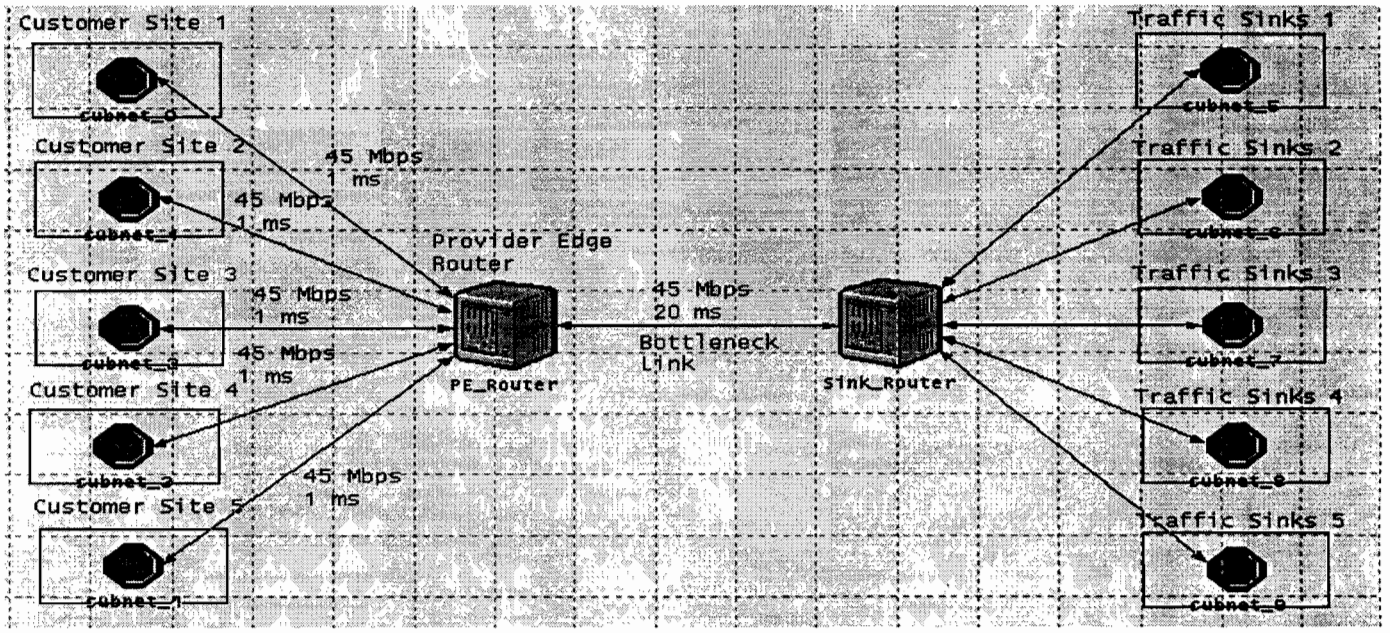


Figure 4.7: Network Architecture for Five-Customer Sites Scenario

Traffic Type	Total Generation Rate	Scheduler weights
AF1	49.60 Mbps	45.0 Mbps
AF2	90.0 Mbps	0.0 Mbps
Voice	5*5.12 = 25.6 Mbps	-
Mission Critical	5*4.8 = 24.0 Mbps	-
DiffServ	5*18.0 = 90.0 Mbps	

Table 4.34: Traffic Generation Rates and Scheduler Weights for Three-Customer Sites Scenario.

It has to be noted that the scheduler weight for AF1 queue is 45.0 Mbps and AF2 traffic is allocated 0 Mbps of the bottleneck link. So both AF1 and AF2 queues are overloaded. AF1 traffic is offered to get a throughput of 45.0 Mbps while AF2 traffic is offered to get nothing.

Case 1: AF11 UDP Voice Sources

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	583.11	677998	15.16	0.002%	40.92 Mbps	45.0 Mbps
AF2	128.22	570590	513.40	0.081%	2.5 Mbps	0.0 Mbps

Table 4.35: Case 1, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. As it can be seen we have first time witnessed AF1 packet drops for case 1. The drops resulted, as AF1 queue is overloaded and crossing of RED threshold values. Therefore AF1 traffic achieved throughput less than offered.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	80.13	5.75e-07	2.67e-08	1.63e-4	5.0986	4.74
MC-1	1779.82	2.67e-08	0.00169	0.0411	4.046	4.34
DiffServ-1	N/A*	N/A*	N/A*	N/A*	0.482	0.0
Voice - 2	83.19	3.62e-07	2.39e-08	1.54e-4	5.0984	4.74
MC-2	1203.49	0.00905	0.0815	0.285	3.6	4.34
DiffServ-2	N/A*	N/A*	N/A*	N/A*	0.86	0.0
Voice - 3	110.09	9.2e-07	2.86e-08	1.69e-4	5.087	4.74
MC-3	N/A*	N/A*	N/A*	N/A*	4.76	4.34
DiffServ-3	N/A*	N/A*	N/A*	N/A*	0.24	0.0
Voice - 4	97.78	4.41e-07	2.66e-08	1.63e-4	5.093	4.74
MC-4	N/A*	N/A*	N/A*	N/A*	3.93	4.34
DiffServ-4	N/A*	N/A*	N/A*	N/A*	0.56	0.0
Voice - 5	87.33	5.78e-07	2.44e-08	1.56e-4	5.096	4.74
MC-5	N/A*	N/A*	N/A*	N/A*	2.8	4.34
DiffServ-5	4350.80	0.0379	0.0195	0.139	0.36	0.0

N/A* - Queue overloaded.

Table 4.36: Case 1, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The above table shows results collected at traffic sinks dedicated for each source. This is a pure overloaded case as both AF1 and AF2 traffic experience congestion. Most of the sources experienced packet drops as a result of WRED. AF1 mission critical source uses TCP as a result they had to back off for each packet drop resulting in higher delays and lower throughputs.

AF1 voice sources used UDP therefore even though RED was dropping voice packets they never backed off. As a result the queue length kept crossing the threshold values resulting in more drops and lower performance values for mission critical traffic. So this can viewed as 'Unfairness ' towards mission critical sources. In case we shall the behavior when voice traffic is given lower drop precedence.

Case2: UDP Voice Sources as AF12

In this case all the voice sources were configured to use UDP as the transport protocol and mark their packets as AF12.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	2353.79	2829394	62.815	0.5e-3%	43.7 Mbps	45.0 Mbps
AF2	45.739	320463	795.21	0.0%	0.3 Mbps	0.0 Mbps

Table 4.37: Case 2, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. It can be seen that the number of AF1 packets dropped came down and throughput achieved by AF1 traffic increased when compared to case 1. The AF2 traffic was not assigned any bandwidth at all, so the TCP was unable setup connections with the destinations, i.e., it reached maximum attempts to setup connections.

The table below shows the results collected at traffic sinks dedicated for each source. It can be seen that by giving the non-responsive UDP voice sources lower drop

precedence we were able achieve better performance values for responsive mission critical sources.

Traffic Type	ETE (msec) Application Layers	Mean Jitter $(sec)^2$	Variance of Jitter $(sec)^2$	Std. Deviation of Jitter	Achieved Throughput (Mbps)	Offered Throughput (Mbps)
Voice - 1	128.09	1.62e-06	2.77e-08	1.66e-4	4.937	4.74
MC-1	297.73	0.000384	2.96e-05	0.00544	4.91	4.34
DiffServ-1	N/A*	N/A*	N/A*	N/A*	0.006	0.0
Voice - 2	128.02	1.19e-06	2.65e-08	1.62e-4	4.947	4.74
MC-2	392.47	0.00194	0.00127	0.0356	3.072	4.34
DiffServ-2	1139.18	0.00943	0.0925	0.304	0.23	0.0
Voice - 3	159.78	2.13e-06	2.695e-08	1.64e-4	4.923	4.74
MC-3	N/A*	N/A*	N/A*	N/A*	3.85	4.34
DiffServ-3	N/A*	N/A*	N/A*	N/A*	0.0018	0.0
Voice - 4	144.14	1.37e-06	2.669e-08	1.63e-4	4.937	4.74
MC-4	N/A*	N/A*	N/A*	N/A*	3.972	4.34
DiffServ-4	865.68	0.0774	0.00541	0.0735	0.027	0.0
Voice - 5	135.32	1.62e-06	2.607e-08	1.61e-4	4.935	4.74
MC-5	N/A*	N/A*	N/A*	N/A*	3.644	4.34
DiffServ-5	N/A*	N/A*	N/A*	N/A*	0.037	0.0

N/A* - Queue overloaded.

Table 4.38: Case 2, Statistics Collected at Traffic Sinks (End-to-end Statistics).

The mission critical traffic is still not getting the Offered throughput. Both voice and mission critical are being queued in the same queue and served at the same rate. There is still a bit 'unfairness' associated as far as serving the packets from AF1 queue is concerned.

Case 3 AF11 TCP Voice Sources

In this case all the voice sources were configured to use TCP as the transport protocol and mark their packets as AF11.

Queue	Packets Queued	Bits Queued	Queuing Delay (msec)	Packets Dropped	Achieved Throughput	Offered Throughput
AF1	258.03	231820	5.31	0.8%	17.57 Mbps	45.0 Mbps

AF2	14.67	78535	10.75	0.0045%	15.23 Mbps	0.0 Mbps

Table 4.39: Case 3, Statistics Collected at Output Queue of the Provider Edge Router.

The above table shows the results collected at AF1 and AF2 queue in the Provider Edge router. AF1 packets are dropped because the voice sources were changed to TCP. The AF1 traffic became burstier and there were instances when the AF1 queue was overloaded causing packet drops. The AF2 queue grabbed the bandwidth lost by AF1 queue and was able get better results than offered.

The table below show the results collected at traffic sinks dedicated for each source. As it can be seen almost all AF1 sources experienced packet loses and as all these sources are TCP they backed off, thereby getting lower throughput and higher delay values. The AF2 sources took advantage of bandwidth lost by AF1 traffic and were able to obtain better throughput values.

Traffic Type	**ETE (msec) Application Layers**	**Mean Jitter $(sec)^2$**	**Variance of Jitter $(sec)^2$**	**Std. Deviation of Jitter**	**Achieved Throughput (Mbps)**	**Offered Throughput (Mbps)**
Voice - 1	1.082141652	0.00157	0.00861	0.0928	0.322	4.74
MC-1	0.255776939	0.0373	0.00167	0.0409	0.85	4.34
DiffServ-1	N/A*	N/A*	N/A*	N/A*	3.54	0.0
Voice - 2	1061.87	0.000666	0.00374	0.0612	0.905	4.74
MC-2	N/A*	N/A*	N/A*	N/A*	0.574	4.34
DiffServ-2	N/A*	N/A*	N/A*	N/A*	2.9	0.0
Voice - 3	249.03	9.94e-5	4.06e-06	0.00201	6.58	4.74
MC-3	1108.01	0.0913	0.0913	0.302	0.375	4.34
DiffServ-3	N/A*	N/A*	N/A*	N/A*	2.7	0.0
Voice - 4	250.25	8.88e-05	3.899e-06	0.00197	6.58	4.74
MC-4	981.26	0.0332	0.119	0.345	0.112	4.34
DiffServ-4	N/A*	N/A*	N/A*	N/A*	2.92	0.0
Voice - 5	1172.10	0.00797	0.0744	0.272	0.061	4.74
MC-5	603.63	0.003005	0.0144	0.1202	0.77	4.34
DiffServ-5	N/A*	N/A*	N/A*	N/A*	3.43	0.0

N/A* - Queue overloaded.

Table 4.40: Case 3, Statistics Collected at Traffic Sinks (End-to-end Statistics).

It is clear changing voice sources to TCP caused AF1 traffic to become lot burstier because of the voice traffic characteristics. It is clear that resources were under provisioned for AF1 traffic for case 3. If a load lower than 0.88 was maintained on AF1 queue then AF1 traffic would have got better results.

4.6 Discussion of Results

In this section the results presented in the above tables is discussed mainly through the help of throughput plots. From Figure 4.2 it can be seen that total traffic increases as number of sites were increased, but the load on AF1 queue was maintained at 0.88 till the number of sites were four. Throughout the discussions Case one and Case two are compared more and case three is treated separately.

4.6.1 AF1 Throughput vs the Number of Sites

Figure 4.9 shows the throughput achieved by AF1 traffic for all the three cases when the number of sites were changed. The throughput is collected at the AF1 queue of the PE Router.

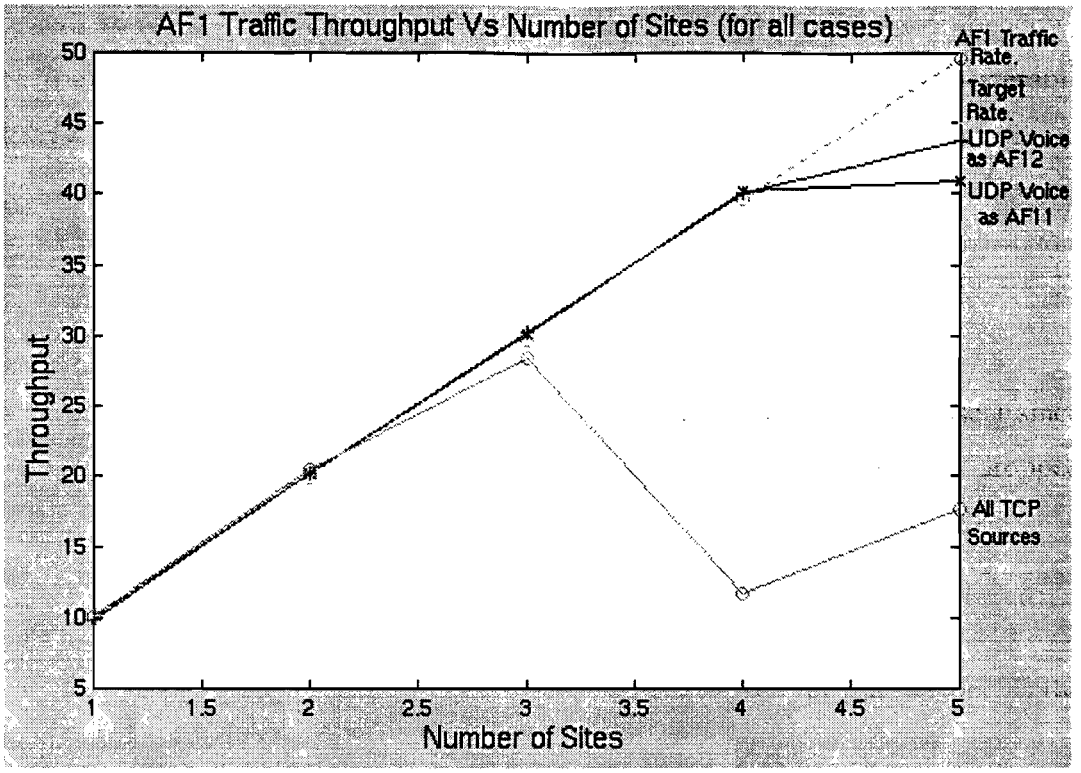


Figure 4.9: AF1 Throughput (Mbps) Vs Number of Sites.

It should be noted that AF1 queue is overloaded for only when the number of sites is five. The following observations were made:

1) It can be seen that for Case two the AF1 throughput closely followed to the target/offered rate than for case one.
2) The AF1 throughput for case three follows closely to the target rate till the number of sites were three and then it considerably decreases for sites 4 and 5.

The only difference between Case one (where UDP and TCP flows have same drop precedence) and Case two (where UDP flows have higher drop precedence than TCP flows) is when the number of sites were 5 till then the achieved throughput for both cases is identical. When the number of sites was 5 the AF1 queue is overloaded resulting in few drops. The MC packets were considerably large packets when compared to voice packets therefore the probability of exceeding the minimum/maximum threshold is more when an MC packet is being queued if both voice and MC packets have the same drop precedence. AF1 voice sources used UDP therefore even though RED was dropping voice packets they never backed off. As a result the queue length kept crossing the

threshold values resulting in packet drops. AF1 mission critical source uses TCP as a result they had to back off for each packet drop. Therefore for Case 1 most of the AF1 packets being dropped belonged to MC. This can be viewed as 'unfairness' to MC traffic.

For Case two, by giving the non-responsive UDP voice sources lower drop precedence we were able achieve better performance values for responsive mission critical sources. This resulted in drops of both voice and MC packets according to their respective threshold values and drop precedence. The mission critical traffic was still not getting the offered throughput. Even though more voice packets were being dropped it did not effect the voice throughput much because the voice was UDP. Both voice and mission critical are being queued in the same queue and served at the same rate. There is still a bit 'unfairness' associated as far as serving the packets from AF1 queue is concerned.

For Case three, the achieved throughput was following close to the target rate till the number of sites were three. When the number of sites were four and five the achieved throughput was considerably less than the target rate. In this case all the sources were TCP including the voice sources. The TCP sources are burtsy in nature and in particular considering the traffic characteristics of voice sources. Therefore, when the number of TCP sources were increased by changing UDP voice sources to TCP, the queue was overloaded sometimes causing RED to drop packets. Therefore the TCP sources had back off reducing the achieved throughput.

It is clear changing voice sources to TCP caused AF1 traffic to become lot burstier because of the voice traffic characteristics. It is clear that resources were under provisioned for AF1 traffic for Case 3. If a load lower than 0.88 was maintained on AF1 queue then AF1 traffic would have got better results. On investigation it was found that a load of 0.6 was suitable for the given traffic model.

4.6.2 AF2 throughput vs Number of Sites

Figure 4.10 shows the throughput achieved by AF2 traffic for all the three cases when the number of sites were changed. The throughput is collected at the AF2 queue of the PE Router.

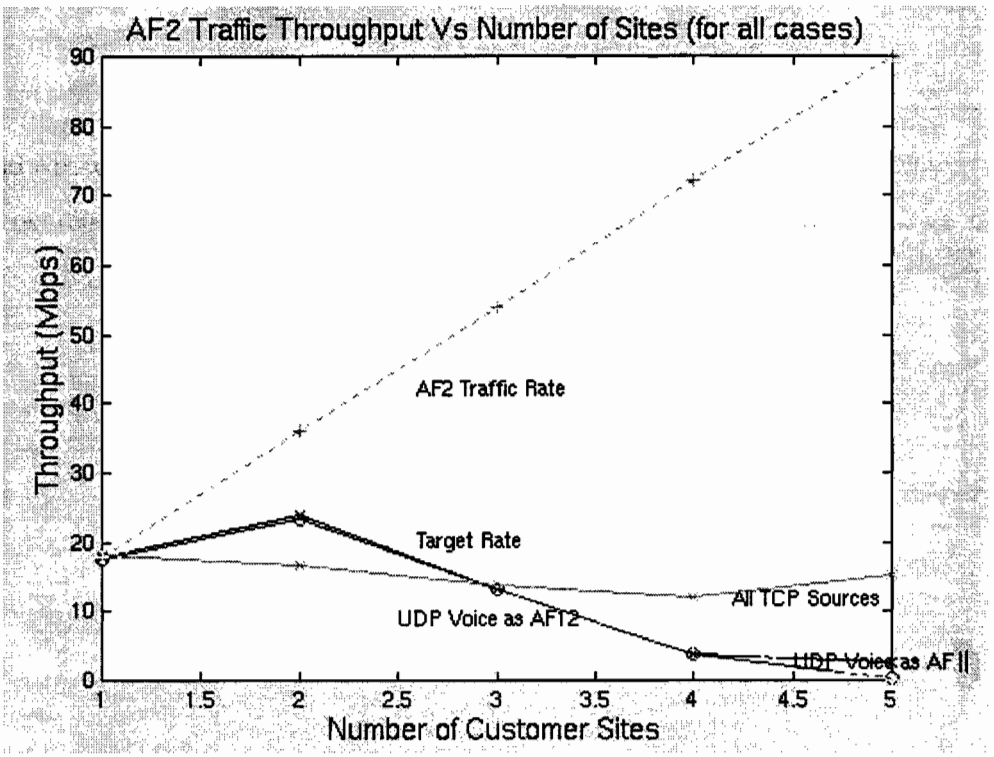


Figure 4.10: AF2 Throughput (Mbps) Vs Number of Sites.

The following observations were made:

1) It can be seen that for Case two (where UDP have higher drop precedence than the TCP flows) the AF2 throughput followed more closely to the target/offered rate than for case one.
2) The AF2 throughput for Case three (where all the sources were only TCP) is follows closely to the target rate till the number of sites were three and then it considerably increases for sites four and five.

It should be noted that AF2 queue is overloaded for all the scenarios except when the number of sites was one.

The throughput achieved for Case one (where UDP and TCP flows have same drop precedence) and Case two (where UDP have higher drop precedence than the TCP flows) is same till the number of sites was 4. When the number of sites were five the AF2 throughput for case two followed more closely to the target rate (i.e. decreased more) than case one. The reason for this is because of the behavior of AF1 traffic. For case one the AF1 throughput decreased more than offered for the reasons explained above. Therefore AF2 throughput achieved for case one is more than Offered since it was allocated the extra bandwidth which was not used by AF1.

For case three AF2 throughput is more than Offered when the sites were 4 and 5. This is again because of the behavior of AF1 traffic for case three. The bandwidth that was not used by AF1 traffic for sites 4 and 5 was allocated to AF2.

4.6.3 Voice Throughput Vs Number of Sites

Figure 4.11 shows the throughput achieved by Voice traffic for all the three cases when the number of sites were changed. The throughput is the average of the throughputs collected at the each of the per source sink/server.

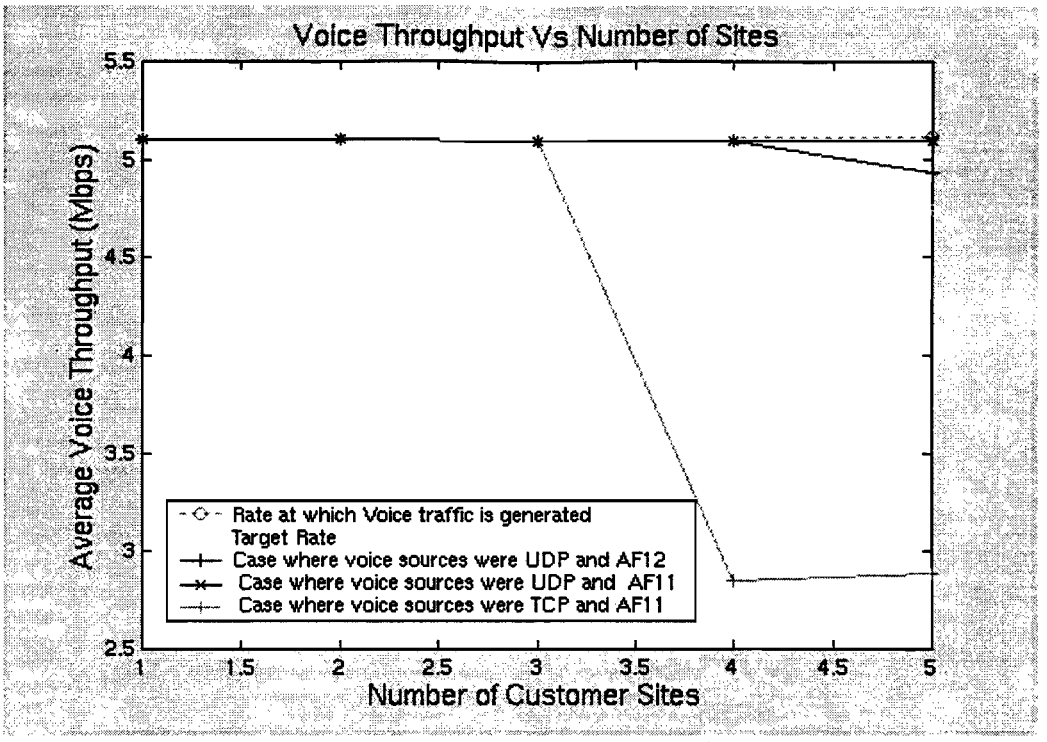


Figure 4.11: Voice Throughput (Mbps) Vs Number of Sites.

The following observations were made:

1) It can be seen that for Case two (where UDP have higher drop precedence than the TCP flows) the voice throughput followed more closely to the target/offered rate than for case one.
2) The voice throughput for case three follows closely to the target rate till the number of sites were three and then it considerably decreases for sites 4 and 5.

The voice throughput achieved is expected to be less than generation rate when the number of sites was 5 because the AF1 queue is overloaded. For case 1 the voice throughput does not decrease as offered while case 2 it decreases.

For Case one since both voice and MC traffic have same drop precedence, most of the packets dropped belonged to MC packets for the reasons explained above. Therefore the MC sources backed off. The UDP voice sources took advantage (unfair to MC sources) of the backing off of the MC sources and achieved throughput equivalent to their traffic generation rate.

For Case two voice packets had lower drop precedence than MC packets. Therefore most of the packets dropped were voice packets thereby the throughput achieved by voice traffic is lower than its generation rate and closer to the Offered rate.

The voice throughput for case three decreased considerably when the number of sites was 4 and 5. The reason for this is same as the reason for decrease in AF1 traffic throughput. The voice sources were working fine for all the cases when UDP was used but when TCP is used the performance degraded for when number of sites four and five. When TCP used the parameters should be configured more careful to obtain good performance values.

4.6.4 Mission Critical Throughput Vs Number of Sites

Figure 4.12 shows the throughput achieved by MC traffic for all the three cases when the number of sites were changed. The throughput is the average of the throughputs collected at the each of the per source sink/server.

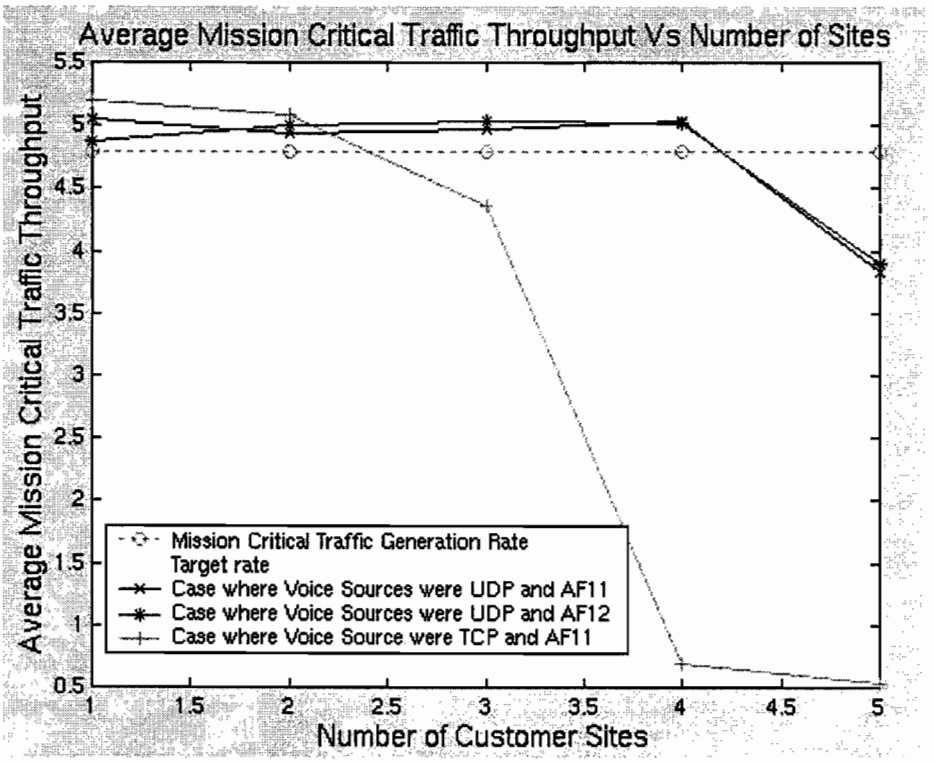


Figure 4.12: Mission Critical Throughput (Mbps) Vs Number of Sites

The following observations were made:

1) It can be seen that for Case 2 (where UDP have higher drop precedence than the TCP flows) the MC throughput followed more closely to the target/Offered rate than for Case1 (where UDP and TCP flows have same drop precedence).

2) The MC throughput for case 3 (where all the sources were TCP sources) follows closely to the target rate till the number of sites were two and then it considerably decreases for sites 3, 4 and 5.

The MC throughput achieved is offered to be less than generation rate. When the number of sites was 5 because the AF1 queue is overloaded.

For Case 2 the MC throughput does not decrease as much as Case 1. For Case 1 since both voice and MC traffic have same drop precedence, most of the packets dropped belonged to MC packets for the reasons explained above. For Case 2 voice packets had lower drop precedence than MC packets. Therefore the number of MC packets dropped decreased thereby achieving better throughput. The MC throughput for Case 3 decreased considerably when the number of sites was 3, 4 and 5. The reason for this is same as the reason for decrease in AF1 traffic throughput.

4.6.5 DiffServ Traffic Throughput vs Number of Sites

Figure 4.13 shown below is similar to AF2 throughput plot shown in Figure 4.10, because AF2 traffic consisted of only DiffServ traffic. The throughput is the average of the throughputs collected at the each of the per source sink/server.

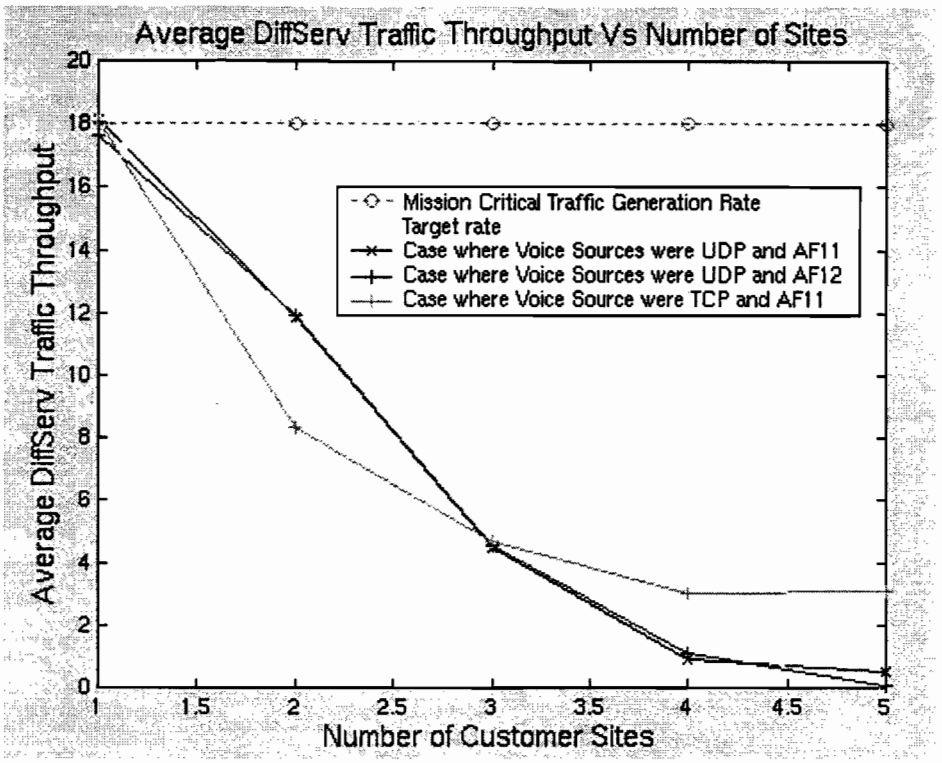


Figure 4.13: DiffServ Throughput (Mbps) Vs Number of Sites.

4.7 Conclusions from Overbooking Study

1) For UDP voice and significant overbooking, it is observed the throughput achieved by AF1 traffic is less than the target rate. The throughput achieved by MC traffic (part of AF1 traffic) is less than its target rate. While the throughput achieved by voice traffic is more than its target rate.
2) For TCP voice, when the number of sites was increased beyond 3 the throughput achieved by AF1 and voice is started decrease compared to the target rate. While the throughput for MC traffic considerably decreased.

The voice sources achieved throughput more than the target rate because UDP was used and they are being queued with TCP (MC) sources. In Case 1 the voice sources almost achieved throughput equivalent to the rate at which they are generating the traffic. The slight loses of throughput in Case 2 is because of change in drop precedence. The MC sources on the other hand achieved a throughput, which is less than Offered. Even though changing voice traffic to AF12 helped it was not enough for MC sources to

achieve the target throughput. Given the fact that AF1 traffic achieved lower throughput than offered it is clear that MC sources were treated unfairly under overloaded conditions.

For case three the throughput achieved by MC traffic is considerably less. Sometimes the AF1 queue is overloaded for the reasons explained above. Almost all the packets dropped belonged to MC sources (note that voice and MC have same drop precedence in this case and MC packets were considerably large than Voice packets) therefore they had to back off decreasing the throughput achieved. As observed from the other cases it can be said that the RED queue discriminates between smaller and larger packets (if the traffic generation rate for both type of packets is comparable) to a certain extent even when it is used in Byte mode. It should also be noted it is observed that when RED queue is used in packet mode the larger packets are treated better than smaller packets (if the traffic generation rate for both types of packets is comparable).

The following conclusions based upon the above study,

- If possible RED should not be used with UDP traffic. As there would not be any response from sources.
- It is clear that if UDP and TCP traffic are queued in the same queue, with both them belonging to the same class, then TCP traffic will be treated unfairly.
- It was shown that in the event where both UDP and TCP belong to the same class, TCP traffic could be protected to certain extent by marking UDP traffic to lower drop precedence.
- If performance results are critical for TCP traffic and congestion is expected, than it is highly desirable to mark UDP traffic to a separate class.
- In the event where both UDP and TCP belong to the same class, fair performance can be obtained by provisioning enough resources for that particular class, so that it does not experience congestion. As seen in the results, mission critical traffic was able achieve good results when the AF1 queue was not overloaded, i.e., when the number of sites were less than five..

- While using RED, length of the packets being queued should be considered. If the packet lengths are not uniform, RED should be used in byte mode. Even in byte mode it has been found that RED showed discrimination based on packet lengths but by choosing optimal parameters based on the traffic characteristics, the sensitivity to packet length factor can be reduced.

It was seen that AF1 traffic achieved good performance values even when the link was overloaded, by punishing lower class AF2 traffic. So useful services can be offered using AF1. When a traffic source uses UDP it achieves better throughput results than when it was TCP and it is also shown that treating UDP traffic is less complex compared to TCP traffic. Therefore providing guarantees to UDP flows is easier than to TCP flows and a service provider can charge more for UDP flows. From the results it is recommended to distinguish traffic based on UDP or TCP to treat them separately. We can conclude that soft service guarantees can be provided to flows using DiffServ model.

The previous work done investigated different schemes and problems, particularly related to assured class. In this chapter we studied the effect of overbooking factor on different service classes. It was shown that traffic flows could be provided service guarantees when they use UDP. RED in byte mode was used, to treat packets of different lengths. It was found the packet length has an impact on the performance of the flow when RED is used. It was found that UDP and TCP flows should be transparent to each other and they can be assigned different drop precedence to achieve the transparency. It was also found it harder to provide service guarantees to TCP flows than UDP flows.

All the factors are inter related from the type and size of packets being queued in the RED queue to the number of classes required. Considering all these factors and based upon the performance it maybe a complex process for a service provider to meet specific requirements.

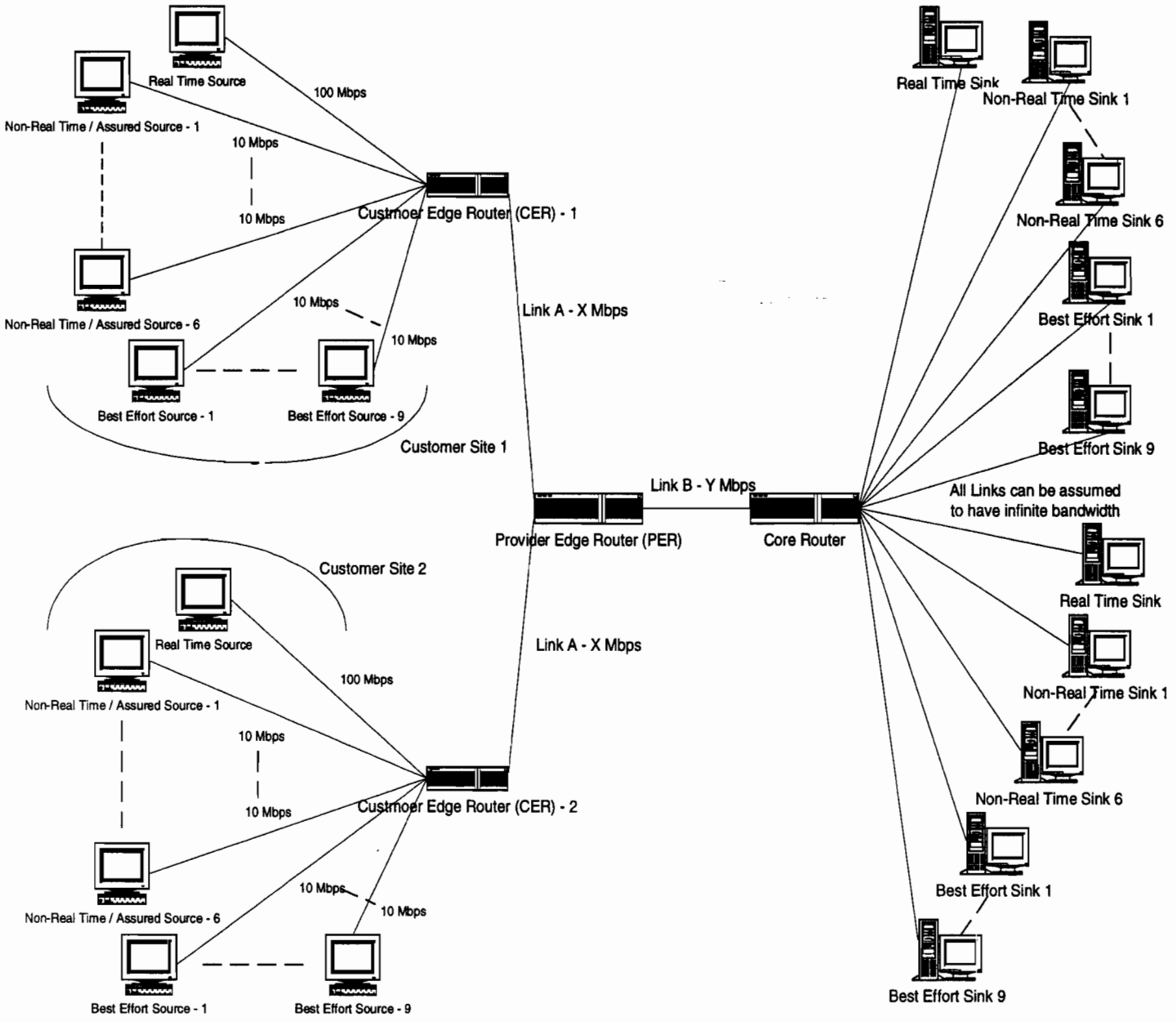
Chapter 5

Evaluation of the Performance Impact of the Number of DiffServ Classes

Most of the studies of DiffServ [14] [18] have investigated and used a two queue model, in which Premium packets are queued in one queue and "Better than Best Effort" (BBE) and Best Effort (BE) packets are queued in the other queue. In which case the OUT of profile BBE packets and BE packets are given the same drop precedence. In this study we compared the performance of the two-queue model and the three-queue model, in which BBE packets and BE packets are queued in separate queues. The performance is compared under different scenarios, to study the effect of the network configuration on the performance. It should be noted that 'BBE' and 'Assured' classes are used interchangeably, even though 'BBE' would be more appropriate for the two-queue model and 'Assured' would be more appropriate for three-queue model.

5.1 The Network Model

The network model shown in Figure 5.1 was used in this investigation. The network model consists of two customer sites. Each of the customer sites contains a Customer Edge Router (CER), which is connected to a Provider Edge Router (PER) in the provider domain. Each site consists of one real time source, six non-real time sources and nine best effort sources. Each source is connected to the Customer Edge Router (CER) using 10 Mbps link except for the real time sources, which are connected using 100 Mbps links. Sources mark their packets such that the real time traffic belongs to premium class, non-real time traffic belongs to "better than best effort" / assured class and best effort traffic belongs to best effort class. The average round trip time of the each TCP sources is 150 ms.



*This a model for simulation, CER and PER have no effect from the sink side. DiffServ is applicable to traffic flows from sources to sinks.

Figure 5.1: Network Topology

The bandwidth of the links connecting CERs to PER is varied. The CER uses FIFO (First In First Out) or DRR (Deficit Round Robin scheme) to transmit the packets. The PER conditions the traffic coming into the network using a two-color marker. The two-color

marker consists of a token bucket, which checks if the flow coming in is in-profile or out of profile, the out of profile packets are remarked as higher drop precedence packets. Only non-real time traffic is conditioned, as the real time sources are configured such that their flows are always in-profile. The PER queues the packets using RIO algorithm and serves them strict priority scheduling. The bandwidth of the link connecting PER to the core router is also varied. Each source is associated with a sink, the core router sends the traffic coming in to their respective sinks. Define link A as the link between the customer edge routers and the provider edge router and link B as the link between the provider edge router and the core router.

5.2 Scenarios

The network configuration shown in Figure 5.1 is used to study and evaluate relative performance of the two-queue and the three-queue models.

Two-Queue Model: In this model the PER consists of only two queues, which are served using priority-scheduling scheme as shown in Figure 5.2. The premium (real time) packets are queued in the higher priority queue using RED algorithm, while the assured (non-real time) and best effort packets are queued in the lower priority queue using RIO algorithm. The two-color marker at the input interface of the PER marks the out of profile assured (non-real time) class packets to best effort packets. Assured class packets are treated better than best effort packets, in the event of congestion RIO drops best effort packets in preference to assured class packets.

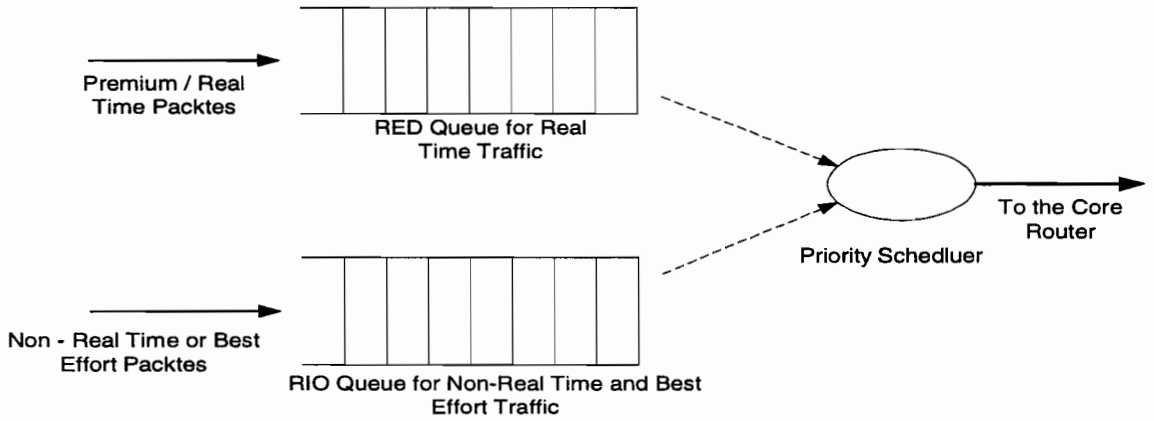


Figure 5.2: The Two-Queue Model.

Three-Queue Model: In this model the PER consists of three queues for premium, assured and best effort classes in that order of priority as shown in Figure 5.3. Packets are queued using RED algorithm in premium and best effort queues, while they queued using RIO algorithm in Assured class queue. The two-color marker marks the out of profile assured class packets such that they have higher drop precedence than in-profile packets. RIO drops the packets with higher drop precedence in preference to the lower drop precedence packets.

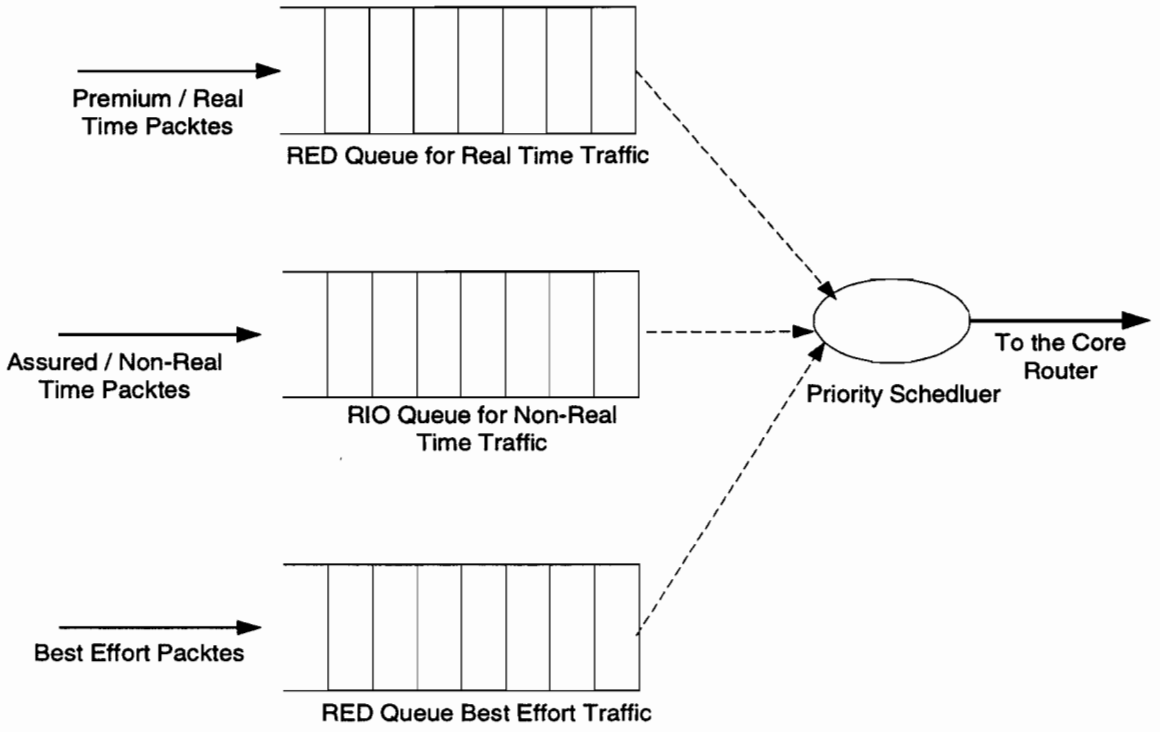


Figure 5.3: The Three-Queue Model.

These two models are studied under different network configuration to evaluate their relative performance. The scheduling scheme in CERs is varied between FIFO and DRR to see the effect on the performance. The experiments were also conducted under different loads by changing the bandwidth of the access links between CER and PER, and the core link between PER and the Core Router. Table 5.1 summarizes the scenarios for each model.

Queuing Scheme in CER	Load on the link between PER and the Core Router.	Load on the link between CER and PER.
FIFO	80%	80%
		90%
		110%
	90%	80%
		90%
		110%
	110%	80%
		90%
		110%
DRR	80%	80%
		90%
		110%
	90%	80%
		90%
		110%
	110%	80%
		90%
		110%

Table 5.1: Different Scenarios for each Model.

5.3 Simulation Parameters

5.3.1 Traffic Model

The parameters used to configure the three sources, i.e., real time, non-real time and best effort sources are shown in Table 5.2. The non-real time and best effort sources can transmit at a rate of 1.068 Mbps, while real time sources can transmit at rate of 10.6875 Mbps. The real time source represents a CBR source and it uses UDP as the transport protocol, as most of the real time traffic uses UDP. It can be seen that non-real time and best effort sources are configured with similar parameters so as to make their

performance comparable, given the fact that they are going to be queued in the same queue.

Parameter	Real Time	Non-Real Time	Best Effort
Pkt Length	64 Bytes	1024 Bytes	1024 bytes
Pkt Length Distribution	Constant	Exponential	Exponential
Pkt Transmission Rate	20,896.1 Pkts/sec	130.4 Pkts/sec.	130.4 Pkts/sec.
PIT* Distribution	Constant	Exponential	Exponential
Transport Protocol	UDP	TCP	TCP
Class	Premium	Assured	Best Effort

PIT* - Packet Interarrival Time.

Table 5.2: Traffic Characteristics.

5.3.2 TCP Parameters

The parameters shown in Table 5.3 are used to configure both non-real time and best effort sources. The TCP parameters are chosen bases on the delay bandwidth product.

Parameter	Value
Fast Retransmit/Recovery	Enabled
Maximum Segment Size	1024 Bytes
Maximum Window Size	64 K Bytes
Retransmission Timeout	Karns Algorithm
SACK option	Disabled
Nagle SWS Avoidance	Disabled
Window Scaling Option	Enabled

Table 5.3: TCP Parameters

5.3.3 Two-Color Marker Parameters

The two-color marker is configured with the parameters shown in Table 5.4. The marking is done on aggregate basis, the token rate is calculated by simply multiplying the total number of non-real time sources and their maximum rate of transmission, i.e., 1.068 Mbps * 12. The bucket size is chosen such that a marking rate of 3 to 5% achieved, when the sources are transmitting at the maximum rate.

Parameter	Value
Token Rate	12.816 Mbps.
Bucket Size	51,200 bytes / 50 Pkts.

Table 5.4: Two-Color Marker Parameters.

5.3.4 Parameters at the Output Interface of the CER

The CER uses FIFO scheme to queue the packets and uses either FIFO or DRR scheduling scheme. The queue sizes in the CER are configured such that they are larger than those in PER. When FIFO scheduling is used the queue size is configured to 430,080 bytes. Tables 5.5 shows the parameters used to configure the queues when DRR is used as the scheduler.

Queue Type	Queue Size	Scheduler Weight
Real Time	102,400 bytes	10.7 Mbps
Non-Real Time	163,840 bytes	6.7 Mbps
Best Effort	163,840 bytes	Variable

Table 5.5: Queue Parameters.

Where is X is calculated by simply subtracting the weights of the real time and the non real time queues from the bandwidth of the outgoing link, if the link is overloaded. If the link is under loaded X is configured to be 10 Mbps and the excess bandwidth is distributed among the three queues. It has to be noted at 100 % or more loads the non-real time and real time queues are almost fully loaded, with the given weights.

5.3.5 Parameters at the Output Interface of PER

The output interface of the PER consists of two or three priority queues. If two-queue model is used the real time traffic is queued in the higher priority queue which uses RED algorithm, while the non-real is queued in the lower priority queue which uses RIO. The parameters of the queues are shown in the Table 5.6 and 5.7 for the two-queue and the three-queue models respectively. For all the RED queues a weight of 0.005 was used.

Queue Type	Marking	Minimum Threshold	Maximum Threshold	Max Drop Probability
Real Time	Premium	25.6 KB / 400 Pkts	51.2 KB / 800 Pkts	0.02
Assured and Best Effort	Assured	46.08 KB / 45 Pkts	81.92 KB / 80 Pkts	0.02
	Best Effort	40.96 KB / 40 Pkts	8.192 KB / 80 Pkts	0.05

Table5.6: Queue Parameters for the Two-Queue Model

Queue Type	Marking	Minimum Threshold	Maximum Threshold	Max Drop Probability
Real Time	Premium	25.6 KB / 400 Pkts	51.2 KB / 800 Pkts	0.02
Assured with 2 DP*	Higher DP	46.08 KB / 45 Pkts	81.92 KB / 80 Pkts	0.02
	Lower DP	40.96 KB / 40 Pkts	8.192 KB / 80 Pkts	0.05
Best Effort	Best Effort	40.96 KB / 40 Pkts	8.192 KB / 80 Pkts	0.05

DP* - Drop Precedence

Table5.6: Queue Parameters for the Three-Queue Model

5.4 Performance Metrics

The performance metrics used here are the throughput obtained by each class at PER's output interface, average end-to-end delay values per source and average throughput values per source. It has to be noted that the end-to-end delay shown in the plots is the delay between the IP layers of the source and the corresponding sink. The end-to-end delays between application layers were also obtained but are not presented in plots as for

overloaded cases those delays were very high. The offered throughput for each class at PER's output interface is 21.375 Mbps for real time (premium) traffic, 12.816 Mbps for non-real time (assured) traffic and 19.224 Mbps for best effort traffic. The delays are shown in milliseconds. The offered throughput for non-real time and best effort sources is 1.068 Mbps, while for real time source it is 10.6875 Mbps. It has to be noted that only two real time sources are present, one in each site. The average throughput per source is plotted only for non-real time and best effort. In all the cases real time sources were able to get their offered throughput.

5.5 System Performance Results

We explored the relative performance of the two-queue model and the three-queue model, and also wanted evaluate the effect of the scheduling scheme in the CER. The results obtained for a given load values are compared for four scenarios, i.e., FIFO scheduling in CER and two-queue model in PER (FIFO/2Q), FIFO scheduling in CER and three-queue model in PER (FIFO/3Q), DRR scheduling in CER and two-queue model in PER (DRR/2Q) and DRR scheduling in CER and three-queue model in PER (DRR/3Q). Note that non-real time and best effort sources are TCP sources so they back off whenever a packet is dropped, while real time source is UDP and it does not back when packet are dropped. Throughout the discussions a packet drop belonging to non-real time or best effort packets means that they back-off freeing bandwidth for sometime.

5.5.1 A Load of 0.8 on the CER to PER Link

The results discussed in this section are obtained by maintaining a load of 0.8 on the link A and varying the load on the link between PER and Core Router from 0.8 to 1.1. Therefore the load on the link from PER to the Core Router is under-loaded, while the link from CER to PER is changed from an under-load to an overloaded link.

5.5.1.1 A Load of 0.8 on the Link between PER and Core Router

The results obtained when a load of 0.8 is maintained on the link A are shown in Figures 5.4, 5.5 and 5.6. It can be seen that all the three sources are getting the offered throughputs. From Figure 5.4 it can seen that for FIFO/3Q and DRR/3Q the non-real time traffic is getting slightly higher throughput, while the best effort traffic is getting slightly lower throughput. This is because in the three-queue model we are using a separate queue for non-real time traffic, which has higher priority than best effort queue. The end-to-end delays were quite low as expected with real time traffic getting the lowest delays.

From Figure 5.6 it has to be noted that the average throughput per source is around 1.12 Mbps, higher than the offered throughput of 1.068 Mbps. This is because we have configured the maximum TCP segment size to be 1024 bytes, the application is configured to have exponentially distributed packet lengths with average length being 1024 bytes. So whenever the application sends a packet, which more than 1024 bytes TCP fragments it thereby adding the unaccounted TCP/IP overhead to it. Three percent of the Assured class packets were marked to lower drop precedence and best effort packets for three-queue and two-queue models respectively. There were no packet drops in PER, while very few real time and best effort packets were dropped in CER when DRR scheme was used. DRR serves the queues in round robin fashion. The real time traffic was getting almost full bandwidth on link B because of the priority scheduling, so real time (UDP) was filling up its queue while the other two queues are being served.

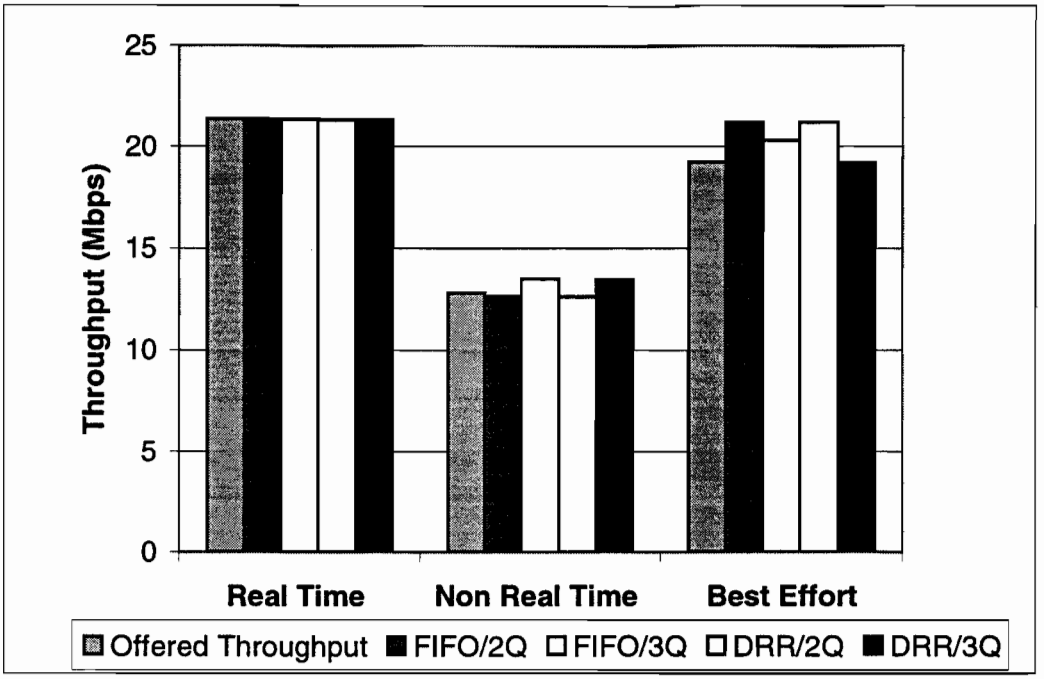


Figure 5.4: Throughput at the Output Interface of the PE Router, 0.8 load on CE to PE Link and 0.8 Load on CE to Core Link.

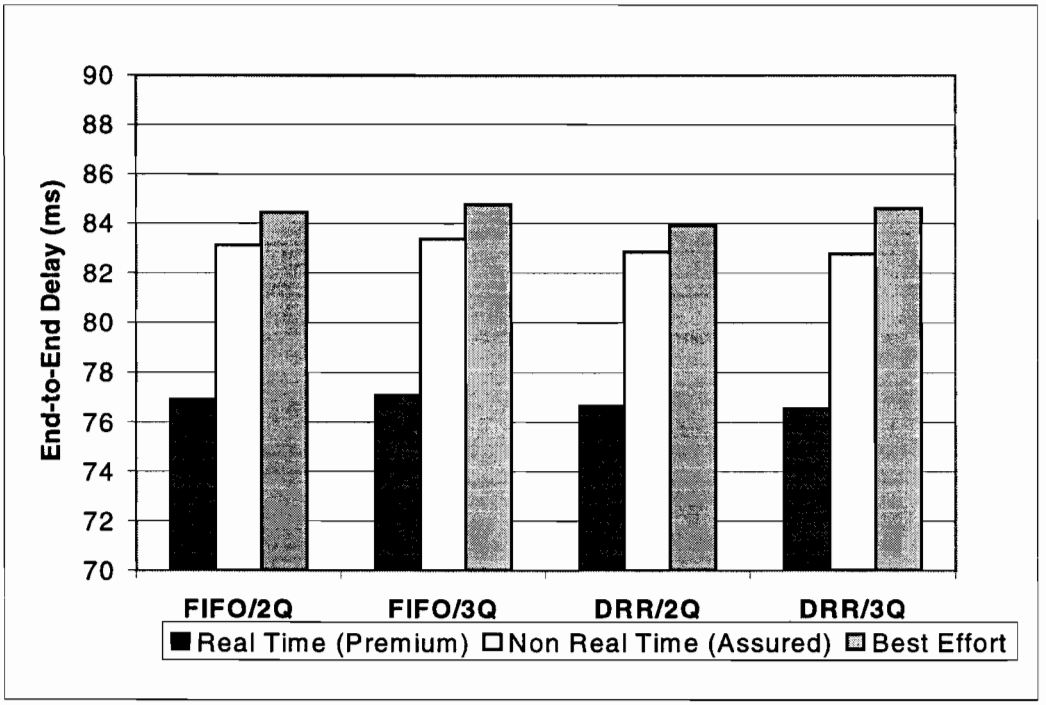


Figure 5.5: Average End-to-End Delay per Source, 0.8 load on CE to PE Link and 0.8 Load on CE to Core Link.

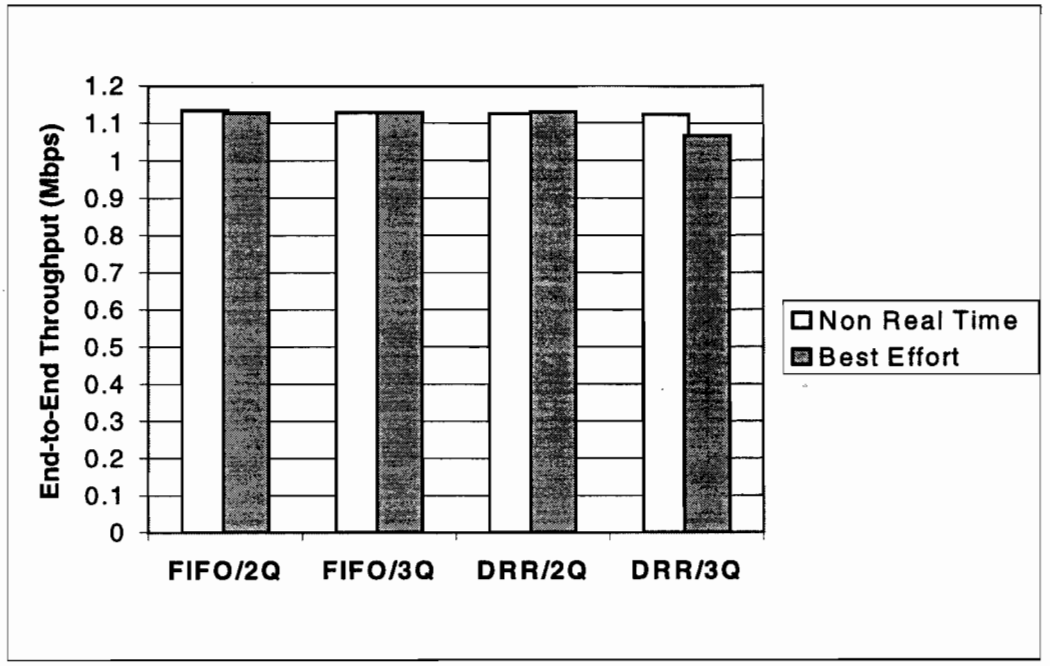


Figure 5.6: Average End-to-End Throughput per Source, 0.8 load on CE to PE Link and 0.8 Load on CE to Core Link.

5.1.1.2 A load of 0.9 on the Link between PER and Core Router

The results obtained with a load of 0.8 on the link from CER to PER and a load of 0.9 on the link from the PER to the Core Router are shown in Figures 5.7, 5.8 and 5.9. As it can be seen from Figure 5.7 real time sources were getting their offered throughput for all the cases. From Figures 5.7 and 5.9 it can be seen the non-real time sources were getting their offered throughput (1.068 Mbps) only for two cases and best effort sources were not getting their offered throughput (1.068 Mbps) for any of the cases.

The packets that were dropped in the PER belonged to best effort class for all the cases. For two-queue model some of the best effort packets that were dropped belonged to assured class that were marked as best effort packets by the two-color marker. Therefore, from Figure 5.9 it can be seen that for two-queue model for both FIFO and DRR case non-real time (assured) sources were unable to get their fair share of the bandwidth. On the other hand best effort sources got better throughput values for the two-queue model as they took advantage of the bandwidth lost by non-real time sources. There were no packet drops in the CER for FIFO case. Few packets were dropped when DRR scheme

most of them belonged to real time sources. From Figure 5.9 it can be noted that the non-real time sources got lesser throughput than the best effort sources for DRR/2Q case, because it lost additional packets in CER.

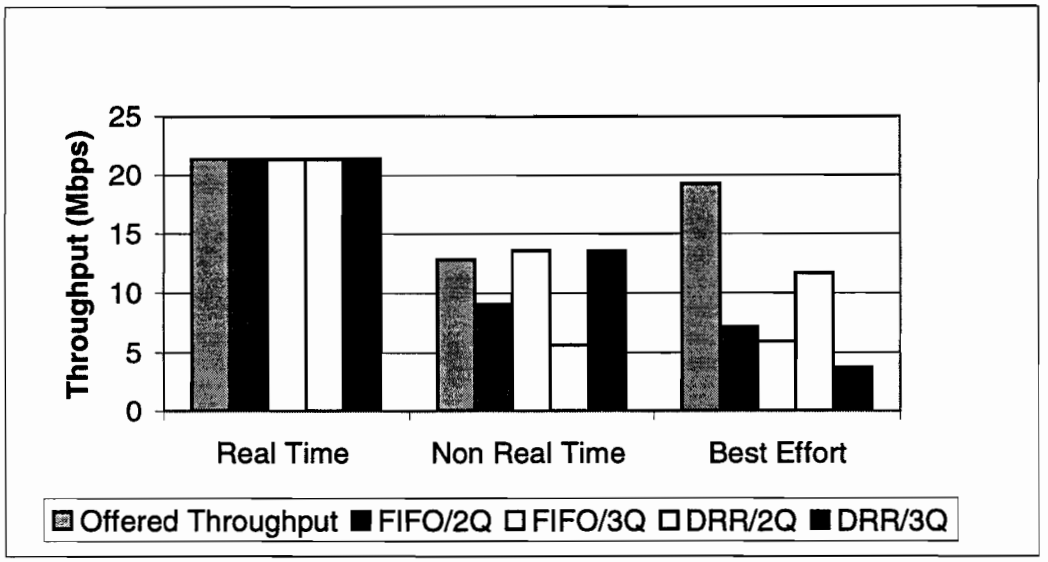


Figure 5.7: Throughput at the Output Interface of the PE Router, 0.8 load on CE to PE Link and 0.9 Load on CE to Core Link.

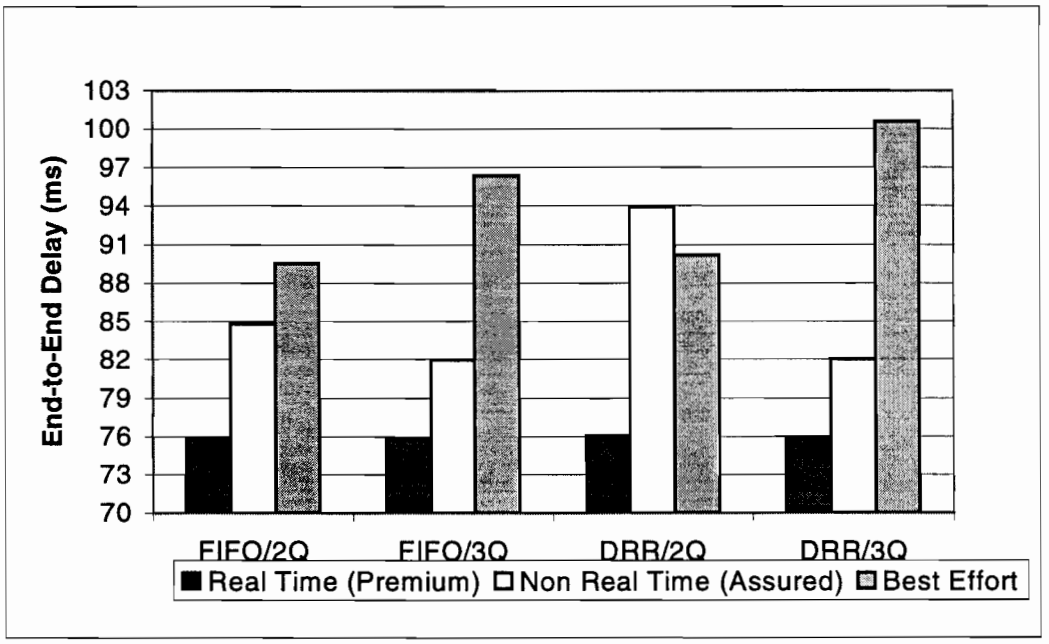


Figure 5.8: Average End-to-End Delay per Source, 0.8 load on CE to PE Link and 0.9 Load on CE to Core Link.

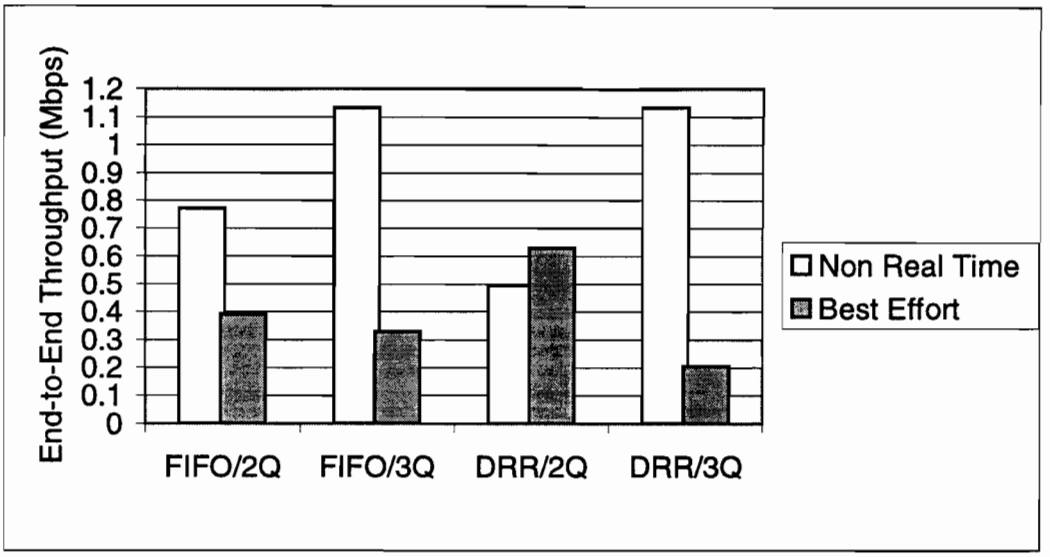


Figure 5.9: Average End-to-End Throughput per Source, 0.8 load on CE to PE Link and 0.9 Load on CE to Core Link.

5.1.1.3 A Load of 1.1 on the Link between PER and Core Router

The results obtained when a load of 1.1 is maintained on the link A are shown in Figures 5.10, 5.11 and 5.12. From Figure 5.10 it can be seen that the real time sources were getting their offered throughput for all the cases. From Figures 5.10 and 5.12 it can be seen the non-real time sources were getting their offered only for two cases and best effort sources were not getting their offered for any of the cases.

As shown in the Figure 5.12 the non-real time sources were able to get their offered throughput (1.068 Mbps) only for the three-queue model for the reasons explained in the previous case. Lot of the best effort packets were dropped as expected in the PER as the link was overloaded. From Figure 5.12 it can be seen that for FIFO/3Q case best effort sources got almost nothing, while for DRR/3Q case DRR was able to protect best effort sources. For two-queue model some of the sources managed to get good throughput values from the bandwidth lost by non-real time sources. No packet drops were observed in CER as sufficient bandwidth became able because non-real time and best effort sources backed off due to packet loses in PER.

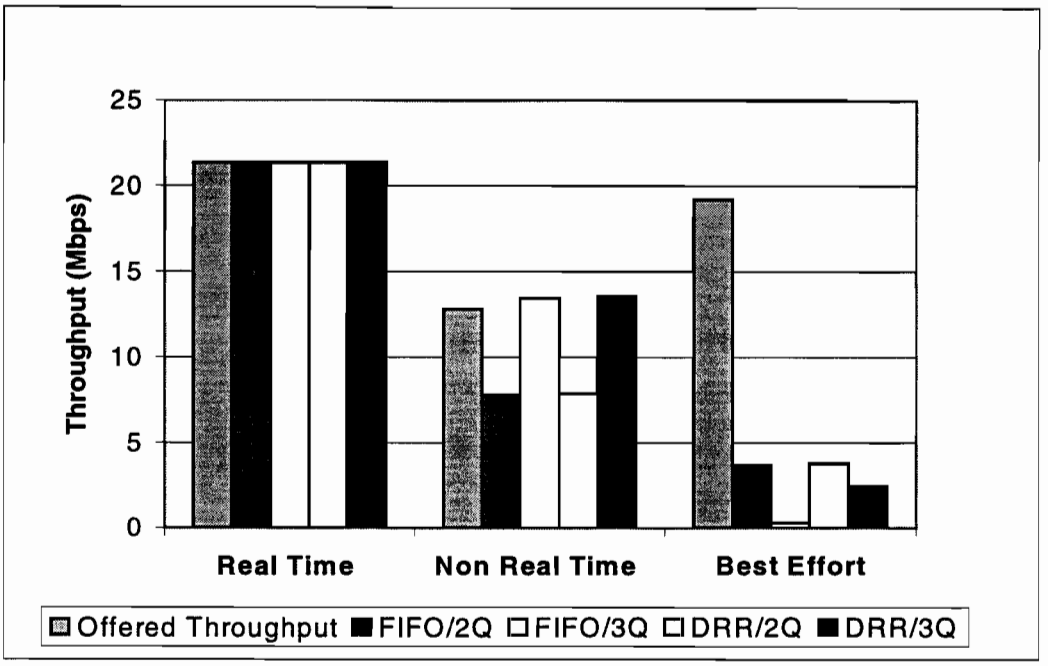


Figure 5.10: Throughput at the Output Interface of the PE Router, 0.8 load on CE to PE Link and 1.1 Load on CE to Core Link.

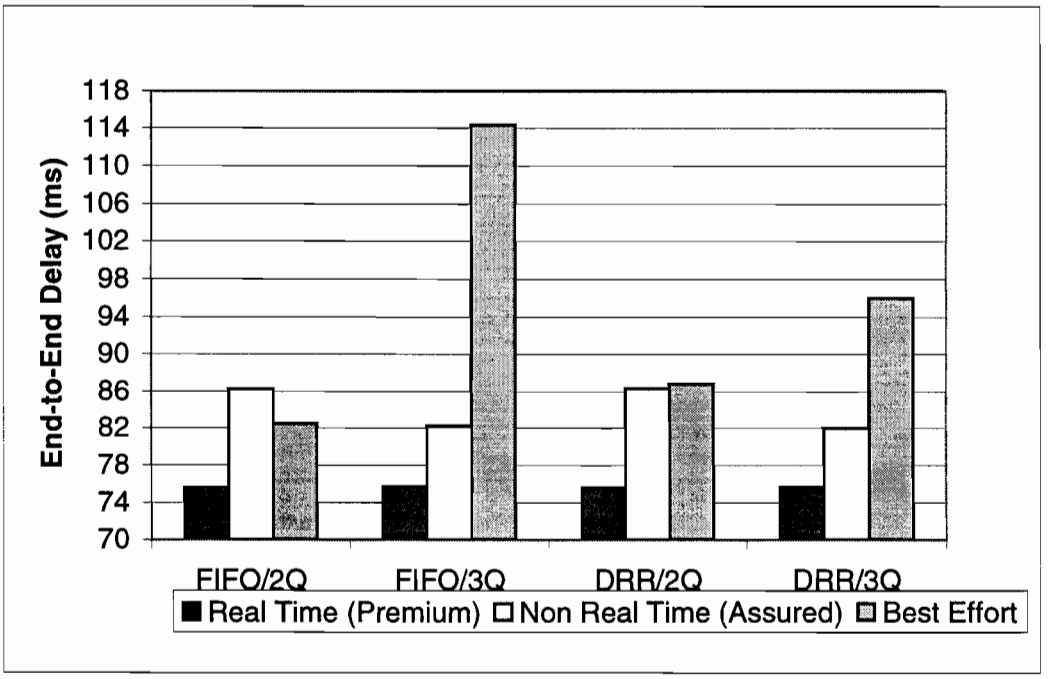


Figure 5.11: Average End-to-End Delay per Source, 0.8 load on CE to PE Link and 1.1 Load on CE to Core Link.

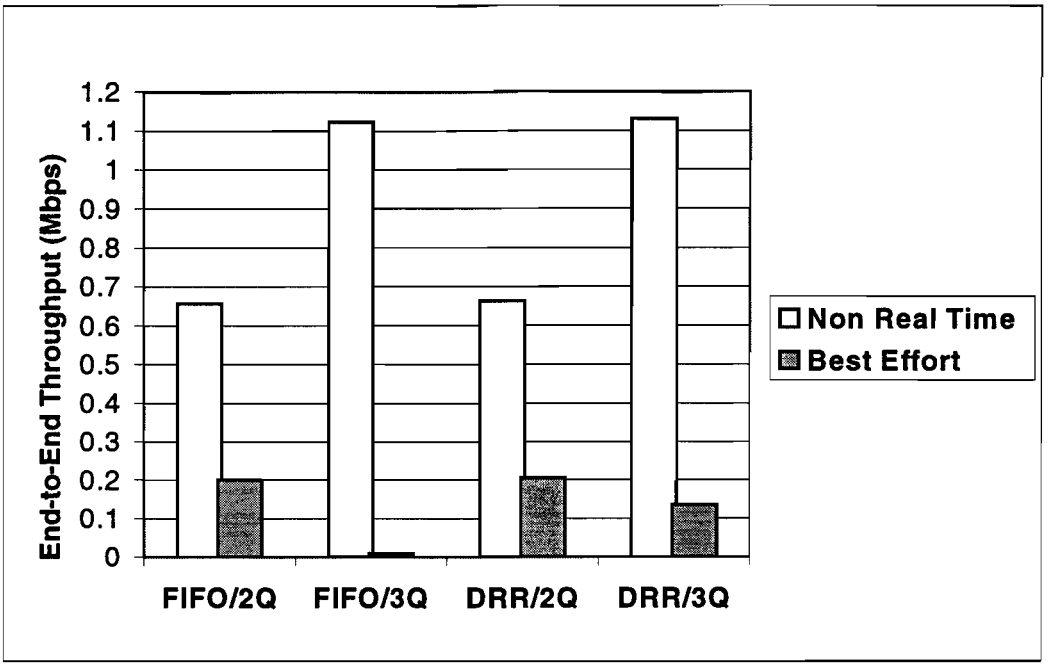


Figure 5.12: Average End-to-End Throughput per Source, 0.8 load on CE to PE Link and 1.1 Load on CE to Core Link.

5.1.2 A Load of 0.9 on the Link between CER and PER

In this section the results are presented are obtained by keeping the load on the link A constant at 0.9 and varying the load on the link B from 0.8 to 1.1. It has to be noted we can expect to see packet drops at CER at 0.9 especially for DRR case considering the fact that the TCP sources are fragmenting their packets. It should also be noted that when FIFO scheme is used in CER UDP (real time) and TCP (non-real time and best effort) traffic are queued together. Therefore if packets are dropped in this queue belonged to TCP sources they would back-off, while if the dropped packets belonged to UDP source it would not back-off and will keep sending at the same rate.

5.1.2.1 A Load of 0.8 on the Link between PER and Core Router

The results obtained with a load of 0.8 on the link between PER and the core router are shown in Figures 5.13, 5.14 and 5.15. From Figure 5.13 it van be seen that all real time sources are getting their offered throughput, the small amount of loss in throughput is due to the drops in the CER. From Figure 5.15 it can be seen that the non-real time sources

got their offered throughput two cases, while for the other two sources they lost around 0.1 Mbps.

Surprisingly the non-real time sources did not get their offered throughput for DRR/3Q. The reason for this is because in this case the load on the non-real time queue in the PER is lower than the load in CER causing the sources to fill up the queue in the CER. This phenomenon did not occurred in FIFO/3Q case because, there is no separate queue for no real time in FIFO case and as explained earlier DRR uses round robin queues to serve queues therefore a given queue should wait for the other two queues to get served. From Figure 5.15, the best effort sources got almost same average throughput as the non-real time sources except for DRR/2Q due the differential treatment by DRR in CER. Best effort sources should have obtained similar throughput values for DRR/3Q as in DRR/2Q case but they grabbed the throughput lost by non-real time sources and managed to get higher throughput values.

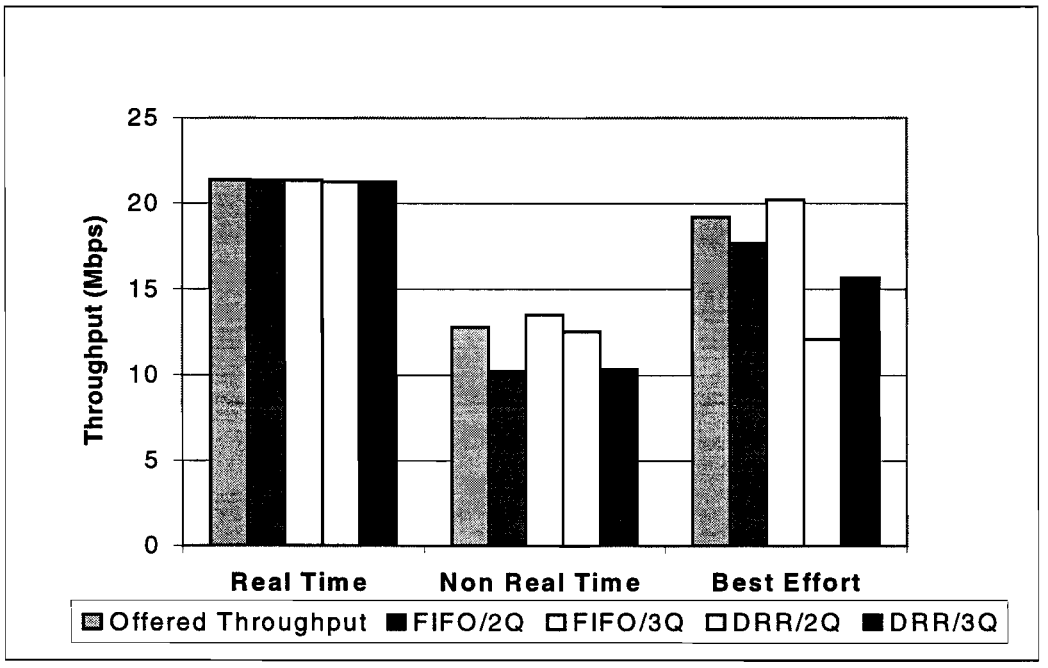


Figure 5.13: Throughput at the Output Interface of the PE Router, 0.9 load on CE to PE Link and 0.8 Load on CE to Core Link.

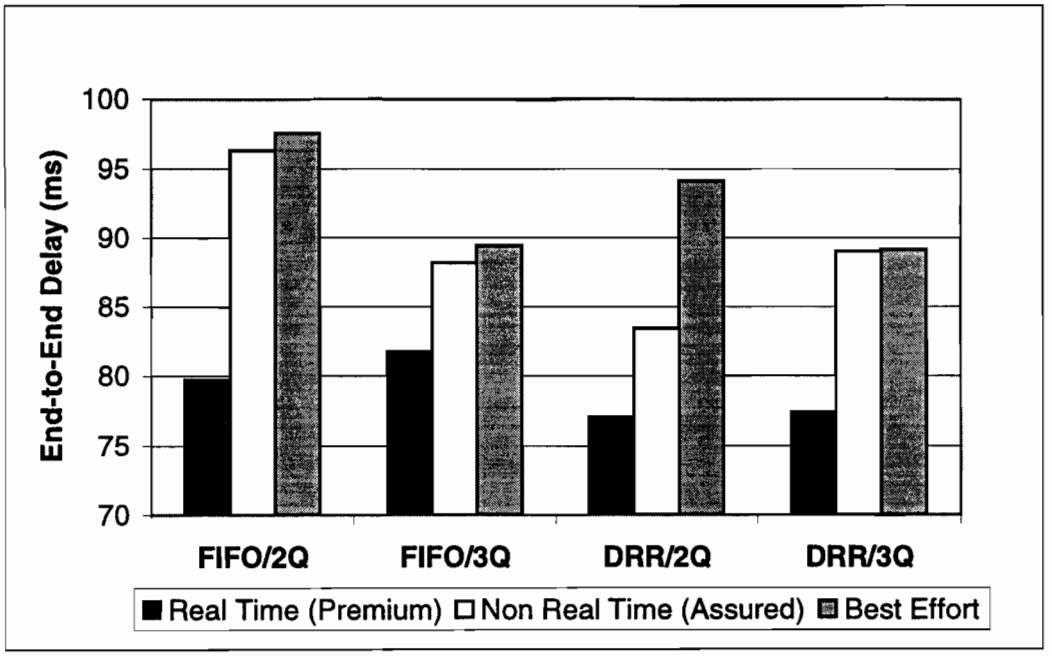


Figure 5.14: Average End-to-End Delay per Source, 0.9 load on CE to PE Link and 0.8 Load on CE to Core Link.

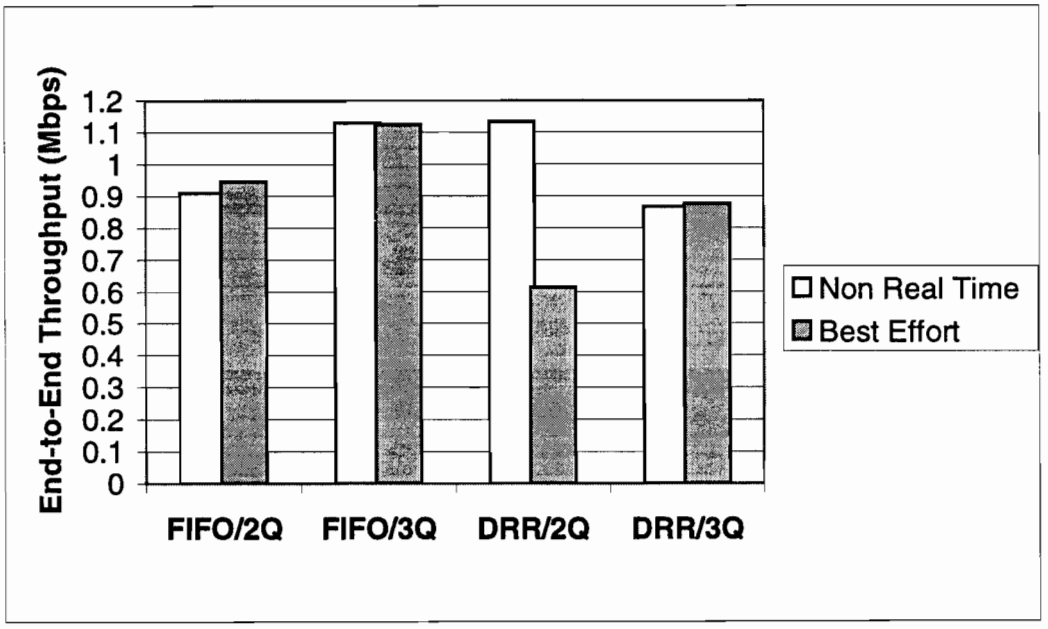


Figure 5.15: Average End-to-End Throughput per Source, 0.9 load on CE to PE Link and 0.8 Load on CE to Core Link.

5.1.2.2 A Load of 0.9 on the Link between PER and Core Router

Figures 5.16, 5.17 and 5.18 shows the results obtained when a load of 0.9 is maintained on the link between PER and the Core Router. It is observed that there were no drops in the PER, as both links A and B are maintained at 90% load. From Figure 5.16 it can be seen that real time sources got their offered throughput for all the cases. From Figure 5.19 it can be seen that the non-real time sources got very good throughput values for DRR/2Q and DRR/3Q cases, because DRR scheme protected non-real time sources from losing any packets that also explains the low end-to-end delays for non-real time sources for these cases. While for FIFO case non-real time sources lost few packets thereby achieving higher delays and lower throughputs. It has to be noted that for all the cases non-real time sources almost managed to get their offered throughput. The best effort sources got good throughput values for FIFO/2Q and FIFO/3Q cases as non-real time sources also shared the loses and therefore both non-real time and best effort sources got almost equal throughput as shown in Figure 5.19.

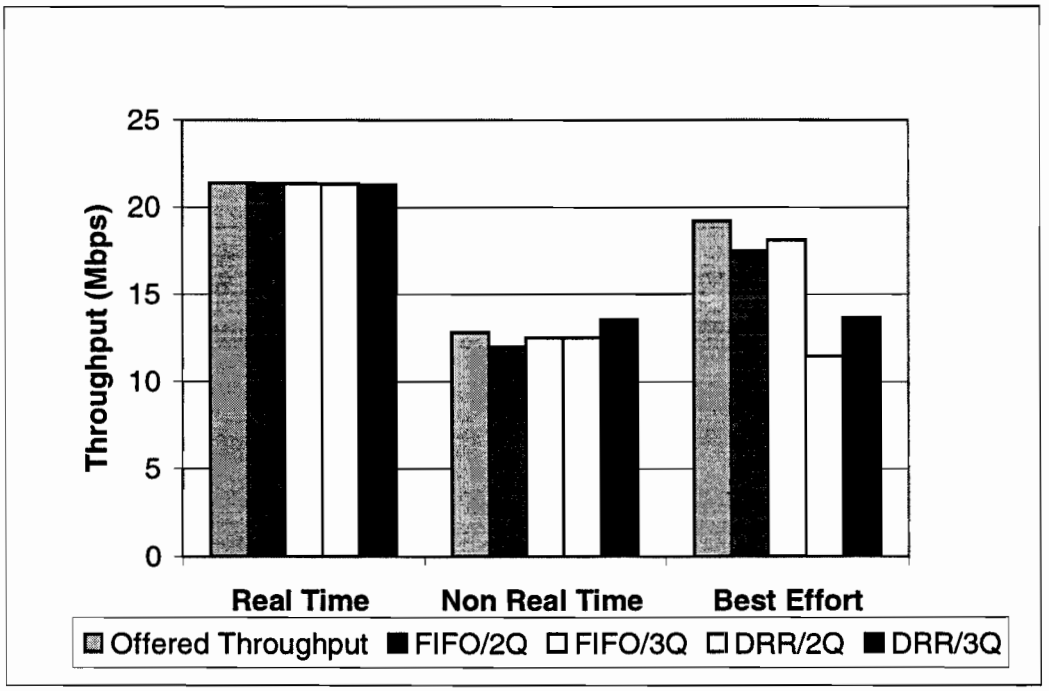


Figure 5.16: Throughput at the Output Interface of the PE Router, 0.9 load on CE to PE Link and 0.9 Load on CE to Core Link.

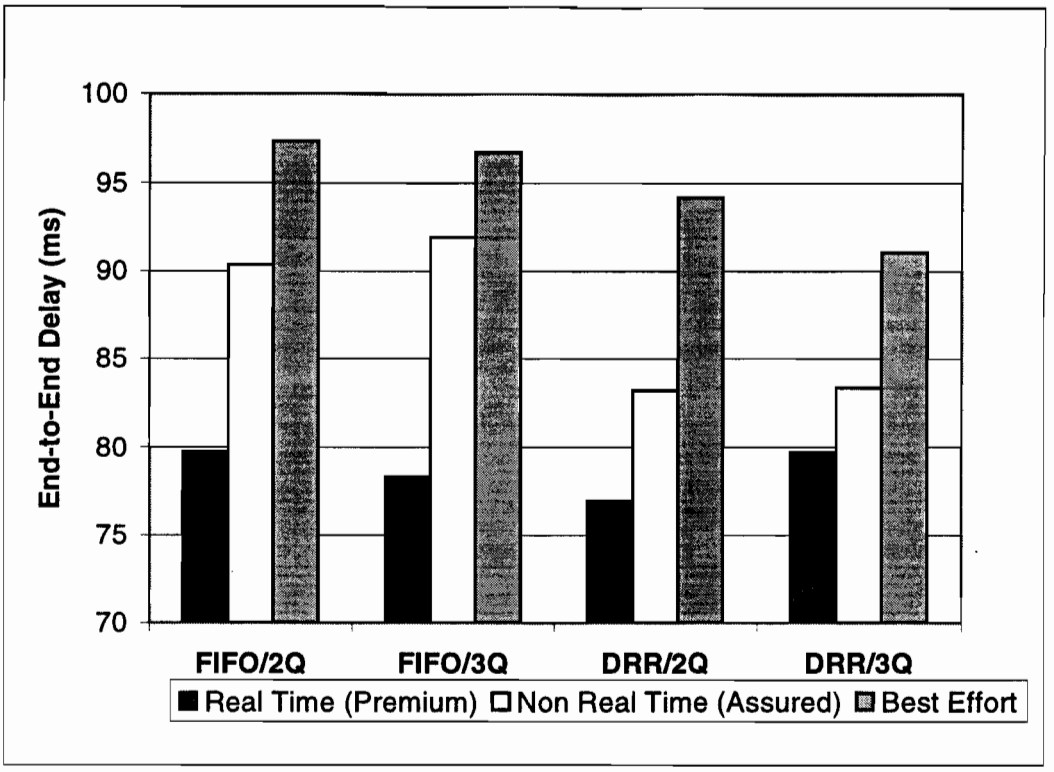


Figure 5.17: Average End-to-End Delay per Source, 0.9 load on CE to PE Link and 0.9 Load on CE to Core Link.

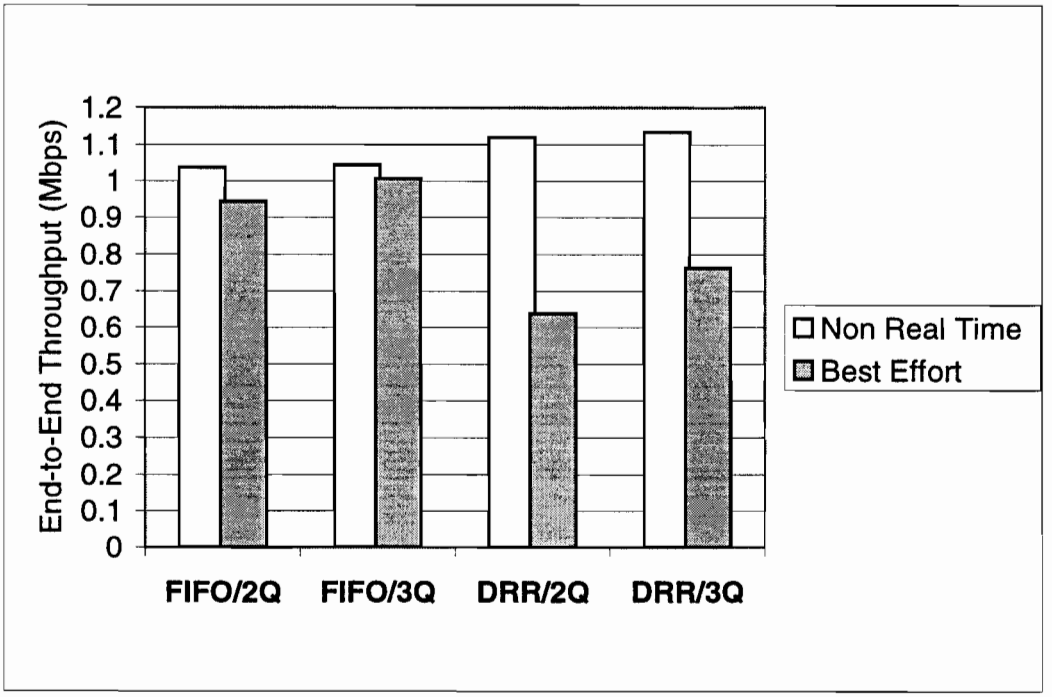


Figure 5.18: Average End-to-End Throughput per Source, 0.9 load on CE to PE Link and 0.9 Load on CE to Core Link.

5.1.2.3 A Load of 1.1 on the Link between PER and Core Router

Figures 5.19, 5.20 and 5.21 shows the results obtained when the link from PER to Core Router was maintained at 110% load. It is observed that no packets are dropped in the CER expect for DRR/2Q case, where few real and non-real time packets are dropped. The real time packets were dropped for the reasons explained earlier, while the non-real time packets were dropped due DRR round robin scheme and the increase in the traffic bursts due to congestion at link B. It can be observed from Figure 5.19 that real time sources got their offered throughput for all the cases. It can be observed that for two-queue model neither non-real time nor the best effort sources got good throughput values due to repeated drops in the PER, all the packets dropped are either best effort packets or remarked non-real time packets. The non-real time packets got low throughput values as their packets, which were marked as best effort by two-color marker, were being dropped. For the three-queue model the non-real time (assured) sources got their offered throughput value (1.068 Mbps). The end-to-end delay of non-real time sources is more than best effort sources for FIFO/2Q case, this is because for this case few best effort packets were successfully transmitted and the end-to-end values recorded were for those packets.

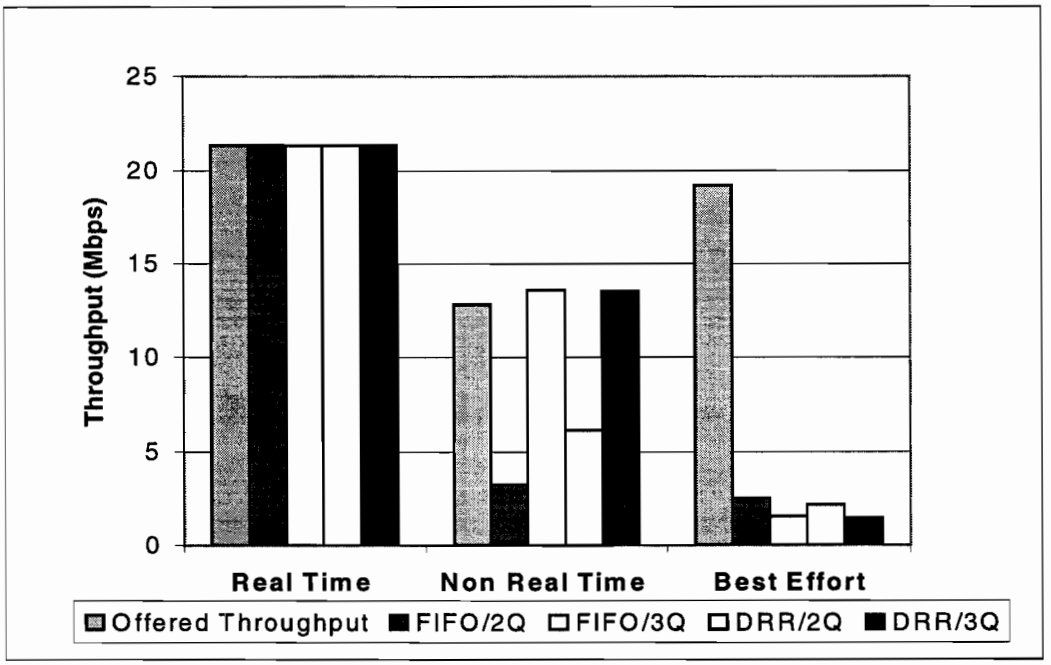


Figure 5.19: Throughput at the Output Interface of the PE Router, 0.9 load on CE to PE Link and 1.1 Load on CE to Core Link.

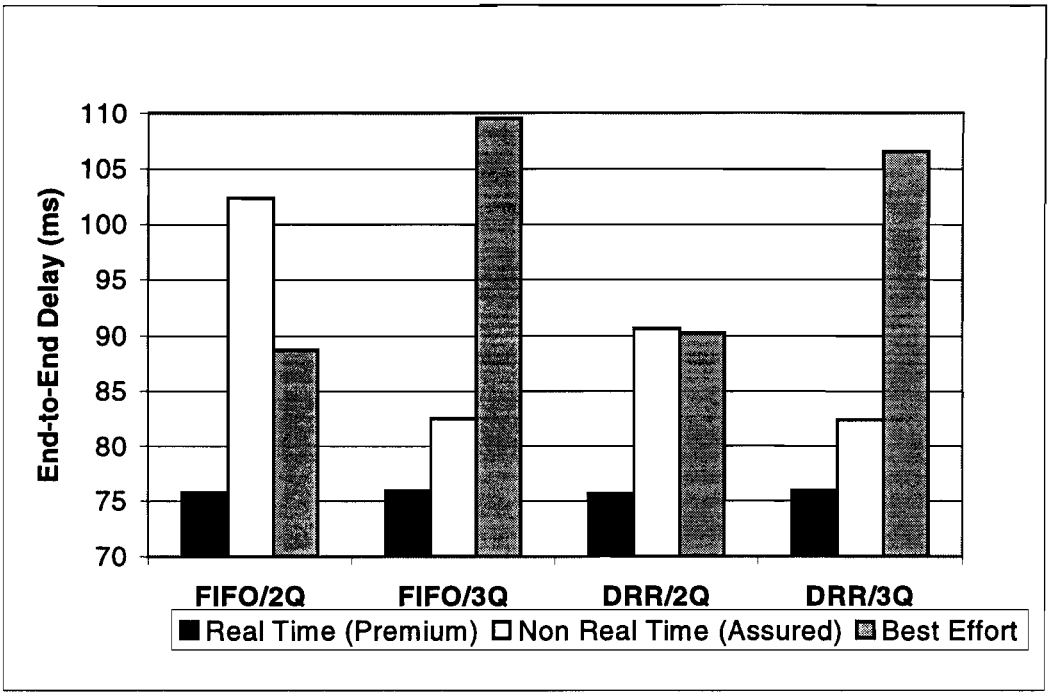


Figure 5.20: Average End-to-End Delay per Source, 0.9 load on CE to PE Link and 1.1 Load on CE to Core Link.

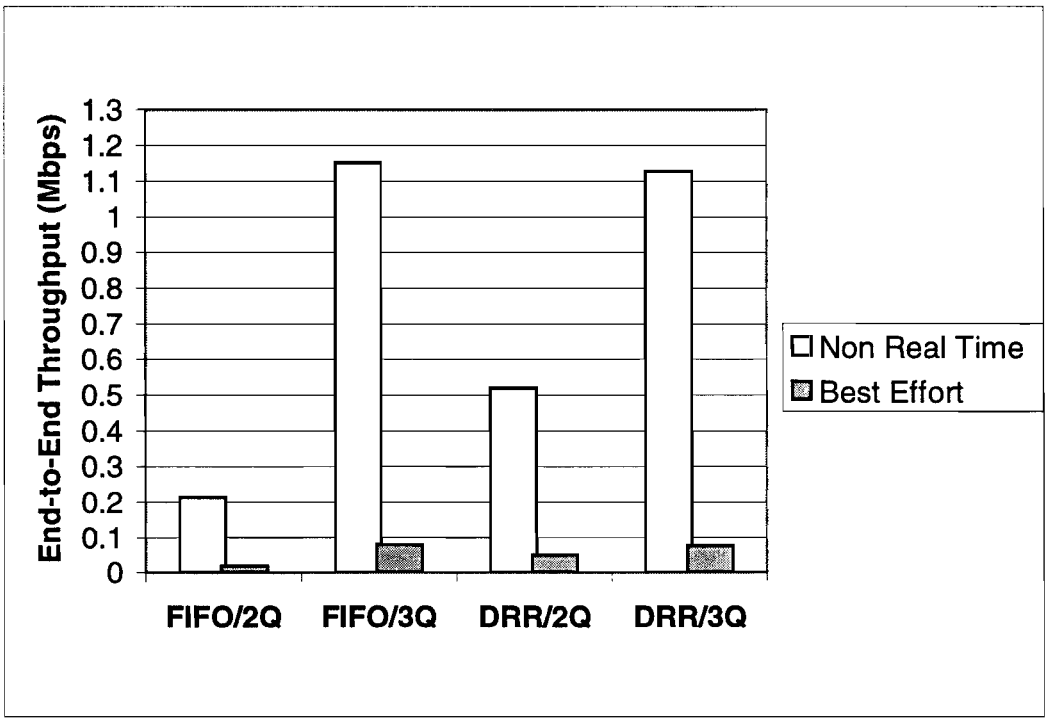


Figure 5.21: Average End-to-End Throughput per Source, 0.9 load on CE to PE Link and 1.1 Load on CE to Core Link.

5.1.3 A Load of 1.1 on the Link between CER and PER

The results presented in this section are presented by keeping the link between CER and PER constant at 110% load and changing the load on the link between the PER and the Core Router between 80% and 110%. It has to be noted that the link from CER to Per is overloaded and the weights of DRR are configured such that a load of 95% is maintained on real time and non-real time queues. It should also be noted that the TCP sources are fragmenting some their segments resulting in TCP/IP overhead, which is not considered while configuring the weights. When FIFO scheme is used in CER it should be noted that both the UDP and TCP traffic are queued in the same queue.

5.1.3.1 A Load of 0.8 on the Link between PER and Core Router

Figures 5.22, 5.23 and 5.24 shows the results obtained when the link between PER and the Core Router is maintained at 80%. It can be seen from Figure 5.22 that the real time sources got their offered throughput for all the cases. From Figure 5.24 it can be seen that both non-real time and best effort sources got very low throughput values for FIFO cases and got higher throughput values for DRR cases.

All the flows were queued in the same queue for FIFO cases, since the link A was overloaded large number of packets were dropped. The real time sources were using UDP so they did not backed off even though their packets were dropped and were able to comfortably grab the unused bandwidth of non-real time and best effort sources, which backed off due to packet loses as they were using TCP.

When DRR was used the non-real time sources were able to obtain a lot better throughput values as they were queued in separate queue with some reserved bandwidth. The best effort sources did not gain as much as non-real time sources as their queue was still overloaded even after assigning some portion of bandwidth. It is also observed that even though the load on the non-real time queue was only 95% they did not got their offered throughput as their packets were dropped due large bursts and the round robin mechanism used by DRR.

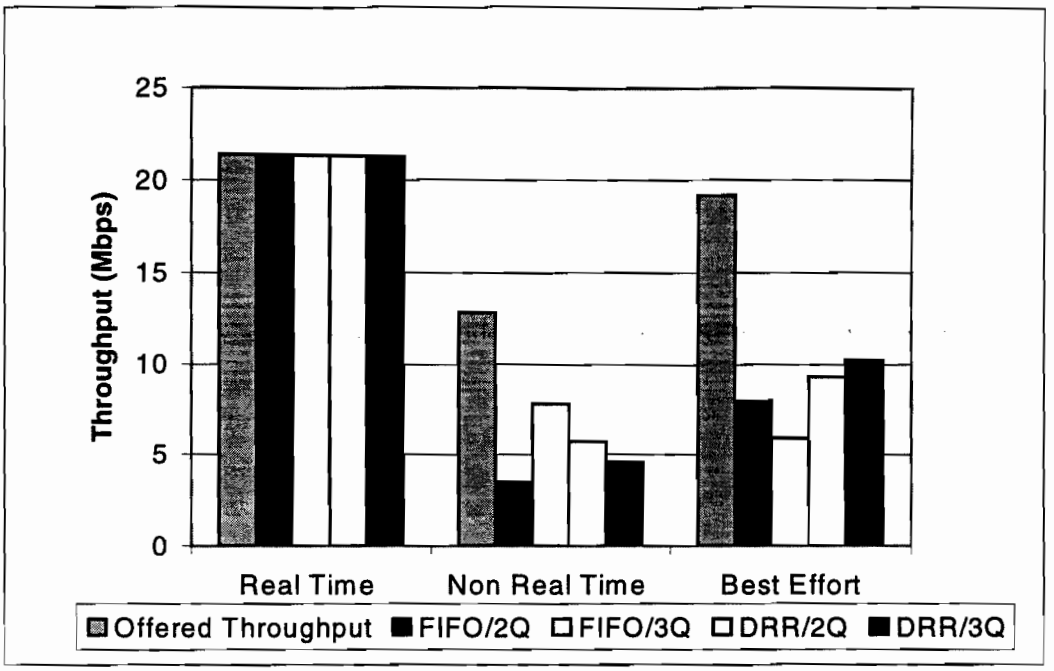


Figure 5.25: Throughput at the Output Interface of the PE Router, 0.9 load on CE to PE Link and 1.1 Load on CE to Core Link.

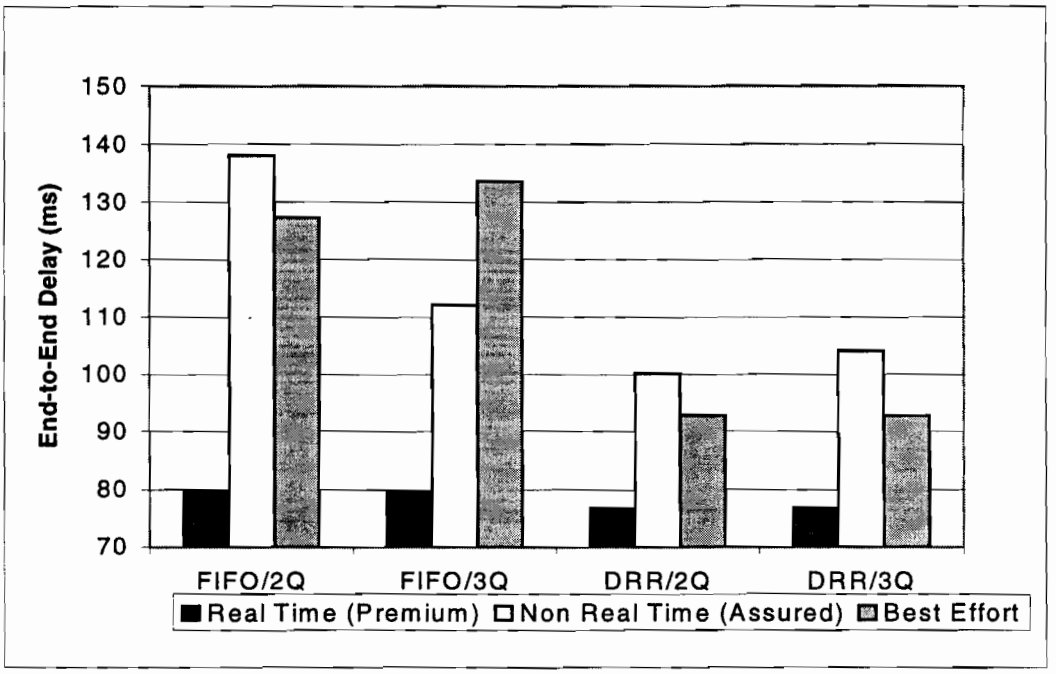


Figure 5.26: Average End-to-End Delay per Source, 0.9 load on CE to PE Link and 1.1 Load on CE to Core Link.

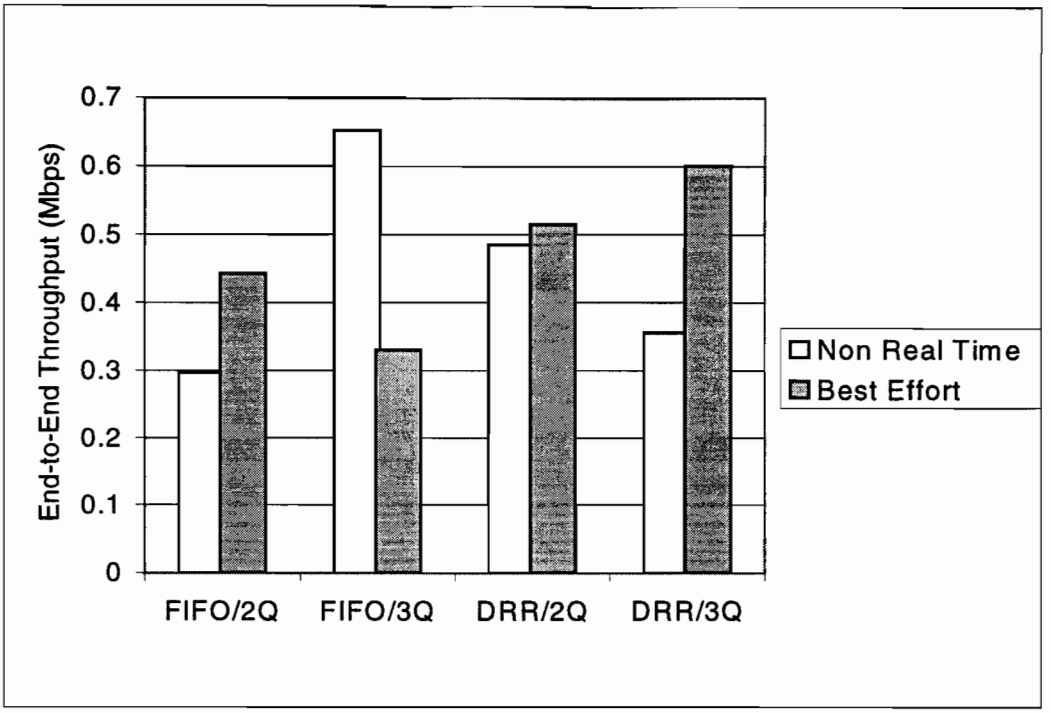


Figure 5.27: Average End-to-End Throughput per Source, 1.1 load on CE to PE Link and 0.9 Load on CE to Core Link.

5.1.3.3 A load of 1.1 on the Link between PER and Core Router

The results obtained are shown in Figures 5.28, 5.29 and 5.30. From Figure 5.30 it can be seen that neither non-real time nor best effort sources got their offered throughput because the link A is over loaded, while real time sources were able to get their offered throughputs as they were using UDP. From Figure 5.28 and 5.30 it can be seen that both non-real time and best effort sources got better throughputs when DRR is used in the CER. As explained earlier, real time flows are UDP flows so they do not back off when packets are dropped, while non-real time and best effort flows use TCP so they back-off when their packets are dropped. When FIFO is used both non- real time and best effort flows were queued together with real time flow, which caused the non-real time and best effort flows to obtain low throughput values. When DRR is used non-real time and best effort flows are queued in separate queue with some reserved bandwidth, which resulted in higher throughput values.

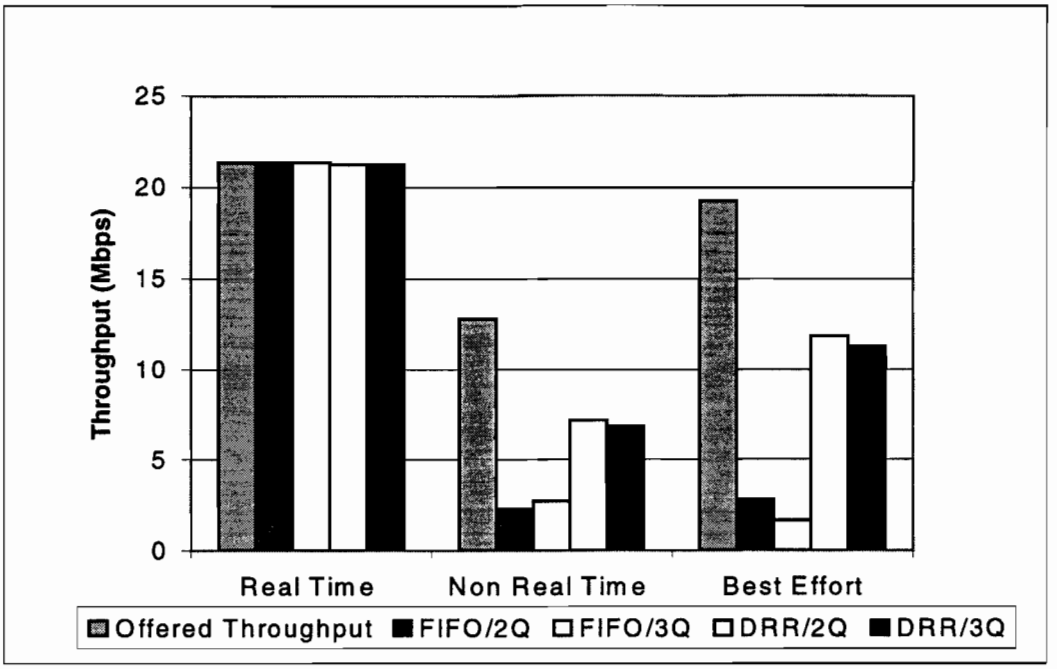


Figure 5.28: Throughput at the Output Interface of the PE Router, 1.1 load on CE to PE Link and 1.1 Load on CE to Core Link.

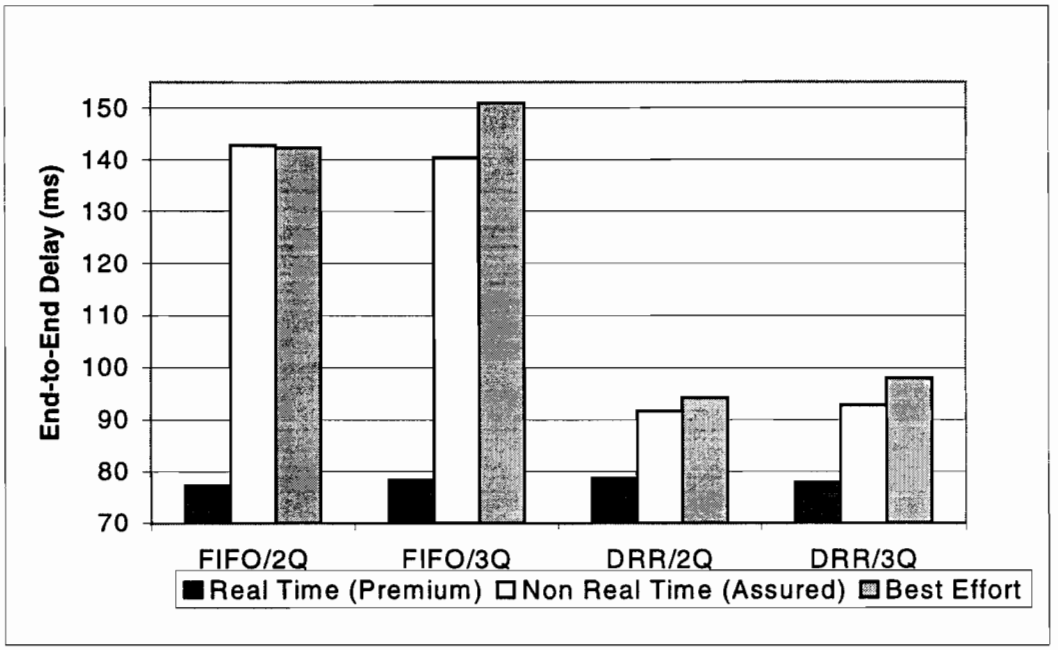


Figure 5.29: Average End-to-End Delay per Source, 1.1 load on CE to PE Link and 1.1 Load on CE to Core Link.

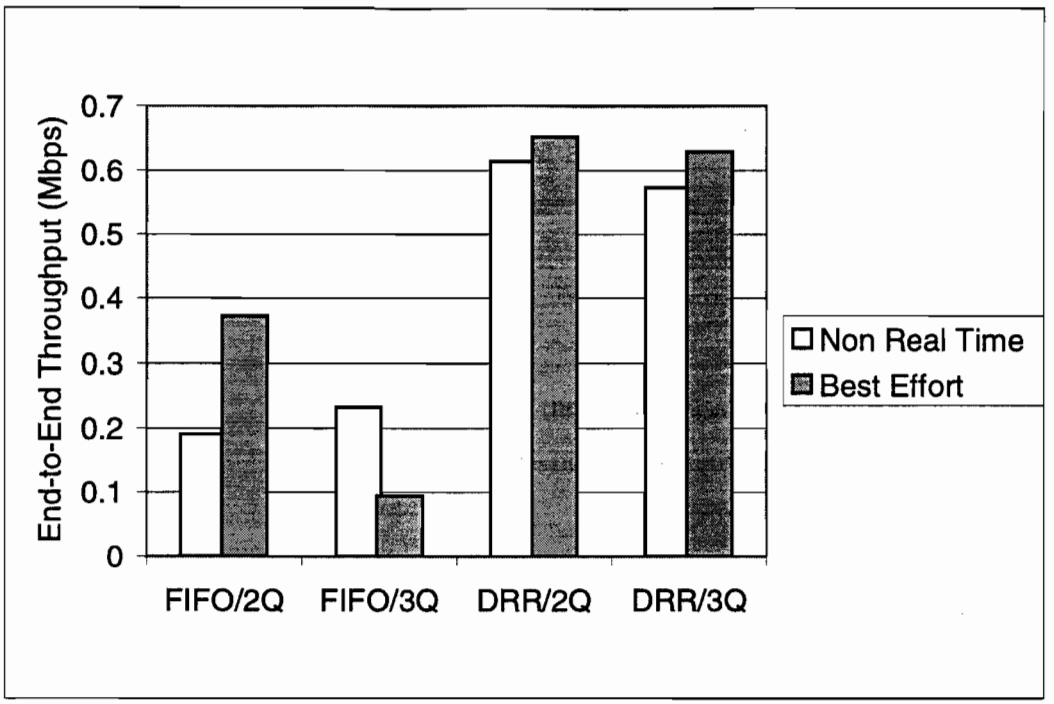


Figure 5.30: Average End-to-End Throughput per Source, 1.1 load on CE to PE Link and 1.1 Load on CE to Core Link.

5.2 Conclusions

In this chapter we tried to determine, which network architecture is better under different combination of loads. Some important lessons were learned through the course of the study, which are discussed in this section. The comparison is done on the basis of the average throughput achieved by each source. The performance comparison of each case is summarized in Table 5.7.

Notation in Table 5.7: In the load column (x, y) represents (load on the link A, load on the link B). G-NRT - Good performance for Non-Real Time, G-BE - Good performance for Best Effort. If instead of G, P is used it means poor performance and if M is used its moderately good performance (e.g. M-NRT - moderately well for Non-Real Time, P-BE - Poor performance for Best Effort). G is used when the sources get 100% of their offered throughout, M is used when the sources get more than 50% of their offered throughput and P is used when the sources get less than 50% of their offered throughput.

The following conclusions are made from the studies performed in this chapter,

- The real time traffic was not effected at all, but for few loses in some cases because it was given the top priority and as it was using UDP as its transport protocol the few loses did not effect the throughput.
- It can be clearly concluded that the three-queue model performs better than the two-queue model particularly when there is congestion on the provider core link.
- In some cases when two-queue model it was observed that the best effort sources got better throughputs than non-real time sources
- The performance of the flows depend firstly on how they were treated in CERs, and then on how they were treated in the providers network. So if there is congestion in customer edge then even if the provider links are under-engineered the performance will be poor.
- In most of the cases when the link A was overloaded, FIFO scheduling gave a poor performance compared to DRR.
- It has also been observed that the performance for TCP flows is badly effected when they are queued with UDP flows.
- DRR scheme provides some guarantees even in the customer edge.
- The weights on the DRR should be configured very careful, even though we took care that a load of 95% is maintained on the non-real time queue, non-real time flows were unable to fully utilize their reserved bandwidth for overloaded situations. The load on the real time queue was also maintained at 0.95%, but the real time flows were able to fully utilize their reserved bandwidth.
- Real time flows were UDP flows so even though there were drops they were able to get their allocated bandwidth while non-real time flows were TCP flows so when their packets were dropped they backed-off which resulted in the under utilization of the reserved bandwidth.
- The UDP (real time) packets were dropped mainly because of the round robin serving mechanism of DRR. The packets in the real time queue had to wait till the scheduler serves the other two queues (non-real and best effort).
- For non-real time queue the round robin scheme had more effect than real time as the non-real time were also a lot burstier. Apart from the above reason the non-real time

packets were because of the TCP fragmentation which added additional overhead which was not accounted for when configuring the weights. This caused the non-real time queue to see a load of around 0.99 to 1.25 at certain times causing the drops.

- The traffic conditioner used is a two-color marker, so whenever a flow is out of profile the packets are marked so that marked packets are preferentially dropped when congestion occurs.
- The TCP flows are very sensitive to marked packets dropped. Because the TCP source backs -off, which in most of the causes under utilization of the bandwidth.
- It has been suggested in IETF and other papers that marker should be used to condition assured class traffic which primarily consists of TCP flows and shaper should be used to condition Expedited / Premium flows which primarily consists of UDP flows. It is interesting to note that if a marker is used to condition a TCP flow their marked packets are dropped in the network if congestion occurs, which results in throughput loses. On the other hand if TCP flows are policed at the edge of the networks better throughputs might be achieved as there will be no OUT packet drops.
- Further research should be done on the marking issue and to find a better way to treat the sensitive TCP flows.
- Most of studies previously done have investigated number of drop precedences and considered a two-queue model to implement assured / BBE service. It is proved in this Chapter that a three-queue model gives better performance results than two-queue model. It is also proved that better service guarantees can be provided if the customer edge allocates bandwidth (DRR scheme) to the aggregate flows especially if congestion is anticipated. It has also been observed that the parameters should be carefully configured for TCP flows to obtain better performance.

parameter. The parameters of the components used to provide service guarantees to TCP flows can also be investigated under different scenarios. Research could be done resource allocation methods to provide harder service guarantees.

Differentiated services can be used to provide service guarantees to mission critical sources like voice, especially if they are assigned a high priority. Soft service guarantees can be provided for TCP flows. The main advantage of differentiated services is that scalable service discrimination can be achieved, through small changes in the network. There is a demand for service guarantees, a service provider can offer these service guarantees thereby increasing the revenues. On the other hand the network has to be designed so that these service guarantees are met. It has been seen throughout these studies that the high priority traffic was always able achieve good performance. So a service provider can offer a premium type of service to customers whose flows are high priority. The service provider has to realize the PHBs that can be implemented in his network and offer services based on that, the service level agreement should offer only those guarantees that can be fulfilled. We have seen that it is hard to provide service guarantees to medium priority service, assured forwarding or the better than best effort type of PHB. If the service provider is confident of provisioning sufficient network resources to these flows then SLAs can be built on this PHB. At present point of time only soft service guarantees can be provided to TCP flows especially if they are not the top priority class. Some dynamic resource allocation to flows can be incorporated in the network to provide more strict service guarantees. Differentiated services can be deployed into network with small number of classes or PHBs with recommendations provided in this work and other similar studies. The network configuration should be tested thoroughly for parameter values and performance results. The SLAs can be built based the PHBs being offered and their expected performance.

References

[1] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, Walter Weiss, *"An Architecture for Differentiated Services,"* RFC 2475, December 1998., ftp://ftp.isi.edu/in-notes/rfc2475.txt

[2] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", ftp://ftp.isi.edu/in-notes/rfc2474.txt.

[3] OPNET Modeler, http://www.opnet.com/products/modeler/home.html.

[4] Bernet, Y., Smith, A., Blake, S., "A Conceptual Model for DiffServ Routers", Internet Draft, December 1999, http://www.ietf.org/internet-drafts/draft-ietf-diffserv-model-00.txt

[5] Lin, L., Lo, J., Ou, F., "A Generic Traffic Conditioner", Internet Draft, April 1999, http://www.ietf.org/internet-drafts/draft-lin-diffserv-qtc-00.txt

[6] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service", November, 1997 http://diffserv.lcs.mit.edu/Papers/exp-alloc-ddc-wf.pdf

[7] Cisco Documentation " Quality of Service (QoS) Networking" http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm

[8] J. Ibanez, and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service", Internet Draft, February 1999, http://search.ietf.org/internet-drafts/draft-ibanez-diffserv-assured-eval-00.txt

[9] Heinanen, J., Guerin, R., "A Single Rate Three Color Marker", Internet Draft, May 1999, http://www.ietf.org/internet-drafts/draft-heinanen-diffserv-srtcm-01.txt

[10] Heinanen, J., Finland, T., "A Two Rate Three Color Marker", Internet Draft, May 1999, http://search.ietf.org/internet-drafts/draft-heinanen-diffserv-trtcm-01.txt

[11] Random Early Detection Gateways for Congestion Avoidance, S. Floyd, V. Jacobson, http://www.aciri.org/floyd/papers/red/red.html

[12] Guerin, R., Peris, V., "Quality-of-Service in packet networks: basic mechanisms and directions",

[13] Cisco Documentation " Policing and Shaping Overview "
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart4/qcpolts.htm#xtocid74401

[14] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", November 1997.
ftp://ftp.ee.lbl.gov/papers/dsarch.pdf.

[15] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, *"Assured Forwarding PHB Group,"* RFC 2597, June 1999.

[16] NS, network simulator, http://www-mash.cs.berkeley.edu/ns/.

[17] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", November 1997.
ftp://ftp.ee.lbl.gov/papers/dsarch.pdf.

[18] Anindya Basu, Zheng Wang, "Fair Bandwidth Allocation For Differentiated Services".

[19] Z. Wang, "USD: Scalable bandwidth allocation for the Internet", Proceedings of HPN'98, Sept. 1998.

[20] Mukul Goyal, Arian Durresi, Raj Jain, Chunlei Liu"Performance Analysis of Assured Forwarding," IETF draft-goyal-diffserv-afstdy-00.txt, February 2000.

[21] Raj Jain, The Art of Computer Systems Performance Analysis, John Wiley and Sons Inc., 1991.

[22] Hassan Naser, Alberto Leon-Garcia, "Voice over Differentiated Services", December 1998.

[23] Stepfan Kohler, Uwe Schafer, " Performance Comparision of Different Class-and-Drop Treatment of Data and Acknowledgements in DiffServ IP Networks".

[24] Martin May, Jean Bolot, Christophe Diot, and Bryan Lyles, "Reasons not to deploy RED".