This is to certify that the

thesis entitled

METHODS FOR MAPPING SCALABLE VIDEO OVER DIFFERENTIATED SERVICES NETWORKS

presented by

QAZI MUHAMMAD RASHID UL HAQ

has been accepted towards fulfillment of the requirements for

M.S. degree in Electrical Engineering

Major professor

Date Dec. 2, 2002

METHODS FOR MAPPING SCALABLE VIDEO OVER DIFFERENTIATED SERVICES NETWORKS

By

Qazi Muhammad Rashid Ul Haq

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

2002

ABSTRACT

METHODS FOR MAPPING SCALABLE VIDEO OVER DIFFERENTIATED SERVICES NETWORKS

By

Qazi Muhammad Rashid Ul Haq

Differentiated Services (DiffServ), which are Quality-of-Service (QoS) based networks for the Internet, provide diverse classes of reliable delivery of Internet packets at a paid price. In this thesis, we propose methods for mapping scalable video streams over the Assured Forwarding (AF) class of DiffServ. A key objective of the proposed methods is to improve the overall video quality as measured by the packet loss ratio (PLR) of the video layers in a scalable video stream. Furthermore, a ratio of the weighted throughput of video packets to the associated cost is employed as a metric to evaluate the performance of the proposed mapping methods. Two high-level mapping strategies are described: a *one-to-one* strategy for mapping each video layer to a distinct AF priority level, and a *multiple-to-one* strategy for mapping multiple video layers to one AF priority level. In order to achieve the desired weighted throughput of video packets in the later strategy, we propose a *linear pricing model* (LPM) and an *exponential pricing model* (EPM) for network bandwidth commitments. The aforementioned mapping strategies were simulated using the Network Simulator (NS). These strategies were also analyzed for various network overload conditions, which result as a consequence of bandwidth over-commitment. Some of the key conclusions of this work include: (1) the EPM provides the most desirable (weighted) PLR performance under zero over-commitment conditions; (2) the LPM provides a desirable overall performance under a variety of conditions.

To my Abbu *Qazi Masroor Ul Haq* and Ammi *Sanjidah Masroor*

~ and ~

their immeasurable prayers for every success in my life

~ and ~

their perpetual belief in my abilities

ACKNOWLEDGEMENTS

I was very fortunate to have an able researcher and excellent teacher, Dr. Hayder Radha as my advisor. I am most grateful to him for his scientific insights and guiding strategies. His advice, criticism, encouragement, and support were invaluable throughout the course of this research.

In addition, I would like to thank Dr. Percy Pierre and Dr. Michael A. Shanblatt for their time and effort in being part of my committee. I would also like to extend special thanks to Dr. Dimitri Loguinov for his interest, guidance as well as the insight he was able to provide for this work during his short stay at Michigan State.

My gratitude is also extended to all my talented colleagues at the WAVES lab, including, but not limited to, *Irtiza*, *Shirish*, *Ali*, *Shahradha*, and *YongYing*. *Aparna R. Gurijala* deserves special thanks for guiding me during the final preparation of this thesis. Moreover, I will always owe a special thanks to *Khurram Bhai* for providing me invaluable advice throughout my graduate studies.

Words cannot express my thankfulness to *Hashsham Bhai* and *Nihala Baji*, for their support, guidance and encouragement. They not only gave me a place in their home, but also gave me a special place in their lives. I will not be able to forget and return their love and care for the rest of my life. Their cute little sons *Ahmad*, *Abdullah* and *Anser* gave us millions of moments to enjoy together.

I would also like to express my deepest appreciation to my family members back home for their love, emotional support and sacrifice. Besides my parents, I owe special thanks to *Khalid Bhai*, *Lubna Baji*, *Tahir Bhai*, *Husna Baji*, *Anwar* and my nephews

Haseeb, *Habib, Faraz* and specially *Zoheb* (arrived during this work, whom I never met). Their support and encouragement helped make my graduate studies possible. I will never be able to return what I have taken from all of them, family time!

Finally, my fiancée *Bushra* who came in my life at the beginning of this work, and always remained their with an encouraging smile and love for me, all the time. I deeply appreciate her love, care and patience for every decision I took during this whole time. I will remain indebted to her, forever!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Motivation for This Work

Since its introduction as a packet network in the early seventies, the *Internet* [1] has undergone a dramatic increase in its usage and enormous changes in its character. These changes included the support of real-time and high-speed applications such as audio/video streaming in addition to the traditional services like web browsing, email, FTP and telnet.

The *Internet Protocol* (IP) [2] promises a *best effort* approach of data delivery [3][4]. Best effort means that IP finds the best possible path from source to destination for data packet delivery without providing 100% guarantees that the data packets will actually be delivered. In other words, and in general, IP routing protocol selects the shortest number of hops to develop a path from source to destination without any delivery assurance.

The primary task of an IP router is to *receive* an incoming packet from one of its interfaces and to forward the received packet to the next router. This routing is based on a simple *forwarding table lookup* operation in conjunction with a set of routing tables that are stored and updated periodically at each router. Therefore, depending on the packet's destination and port address, an IP router *forwards* the packet to the next hop router according to the forwarding (routing) table. IP routers perform this task as fast as possible

on every packet they receive. When a large number of packets converge on the same router within a very short time interval, and when the outbound links cannot service these packets at the rate of their arrival, the packets get buffered at the router. This condition, which results in excessive delays in packet delivery, is known as *congestion*. Congestion could lead to packet loss events since routers start to discard incoming packets due to limited buffer size.

Under ideal conditions, a packet network, such as the Internet, should guarantee the delivery of every packet (i.e., high reliability) while supporting the desired sending rate of any application. To achieve the first goal (i.e., packet delivery guarantees and high reliability), transport-layer protocols, such as the well-known *Transmission Control Protocol* (TCP) [7], was designed. However, TCP increases the reliability factor while decreasing the sending data rate. Consequently, when TCP establishes an end-to-end connection between the sender and the receiver, it requires the transmission of acknowledgement for every packet from the receiver to deliver another set of packet(s). This causes an application to reduce its sending rate. Delay-insensitive data like email, ftp and WWW can utilize TCP easily and with high reliability factor rather than streaming real time applications.

Another transport layer protocol known as *User Datagram Protocol* (UDP) [5] gives a provision to send data packets at the desired speed that is required by most interactive real time streaming applications without any end-to-end connection establishment. Consequently, relative to TCP, UDP reduces the transmission reliability while maintaining a desired sending rate. *Real time Protocol* (RTP) [6], which was designed for real time media applications, also recommends the use of UDP instead of TCP in

order to meet the transmission rate constraints required by real-time streaming applications.

The employment of UDP over the Internet implies the occurrence of network impairments such as packet loss events. These events are normally characterized through the parameter *packet loss ratio* (PLR). PLR is ratio between the number of packets lost (dropped) during transmission and the total number of packets sent by the sender. Other network impairments, such as delay jitter and high roundtrip delays (also known as RTT) occur over the Internet due to variation in the level of congestion. Since end-to-end unreliable data delivery with jitter, high latency, and packet losses negatively affect the user perceived media quality [10], *Quality of Service* (QoS) tools are being proposed. QoS tools are expected to bring improvement in the network performance by providing different PLR, latency and jitter characteristics to different types of packets [8][9].

While new QoS tools have been proposed and investigated, scalable video coding techniques have been developed for transmission over networks that do not provide QoS guarantees (such as the Internet). Both MPEG-2 [11][25] and MPEG-4 [26] [55]support a variation of scalable compression methods. In such schemes, a video stream is composed of a single *base layer* (BL) and one or more *enhancement layers* (ELs). The base layer contains data required to produce a minimum acceptable quality video, where as every enhancement layer, like the name states, enhances the video quality produced by the base layer. Thus, packets containing the base layer always need a higher priority for transmission than enhancement layer packets. For a layered video packet stream with n layers, containing a base layer (i = 1) and n-1 enhancement layers, the probability P_i of losing a packet in layer i should ideally be:

$$P_i < P_{i+1} \quad i = 1,2,3....n \tag{1.1}$$

or in terms of Packet Loss Ratio (PLR), the minimum requirement should be:

$$PLR_i < PLR_{i+1} \tag{1.2}$$

Best effort Internet does not promise to satisfy the conditions mentioned in equations (1.1) and (1.2) for streaming video applications. In order to ensure delivery of quality video, every layer in a scalable video stream needs a distinct service behavior depending upon its importance and priority.

1.2 Problem Description

Differentiated Services (DiffServ), which is one of the QoS routing protocols suggested by the *Internet Engineering Task Force* (IETF) provides quantitative classification of packets. Four major types of packet forwarding services (also known as *Per Hop Behaviors* (PHB) *Expedited Forwarding* (EF) [12], *Assured Forwarding* (AF) [13] and *Best Effort* (BE) are defined in the DiffServ charter. Depending upon the user-service provider agreement known as *Service Level Agreement* (SLA), a packet can be treated with one of these available packet-forwarding services.

This work aims to analyze different strategies for the transmission of layered scalable video streams, which can be used for popular *streaming applications,* over a DiffServ network node. We consider strategies that meet the PLR constraint expressed in equation (1.2). We also analyze the performance of these strategies in terms of the weighted throughput over some "cost" parameter. This parameter provides a measure of the "cost" paid for utilizing differential treatment of layered packet video streams like the ones supported by the MPEG-4 Fine-Granularity-Scalable (FGS) video standard [31]. Although the EF is the premium service that provides a low jitter and low latency service

and is considered ideal for real-time interactive applications (e.g., voice or video telephony), the potential high cost associated with this type of service does not justify its use for popular streaming applications since these applications do not require very stringent low-delay requirements. Consequently, in this thesis, we focused on utilizing the AF service as a suitable framework for streaming applications. We believe that AF strikes a good balance between requiring reasonable "cost" (when compared to EF) and providing some level of PLR guarantees (when compared with best effort Internet).

The AF service class offers three levels of packet drop precedences (i.e., three priority levels). Thus scalable video layers (say n) in a stream can utilize an AF class either by allocating packets of every distinct layer to a distinct priority level as shown in Figure 1-1(a), or by providing service priority within one or two priority levels as shown in Figure 1-1(b). The first approach (i.e., associating each video layer with a corresponding AF priority level) is not suitable in all cases since DiffServ does not allow more than three-drop priorities within a single class. However, this approach is applicable when a video stream contains three or less number of layers. For the video streams containing more than three layers (which is common in today's popular video compression techniques) we preferred the second approach. In this approach we utilize the two priority levels (Level 1 and Level 2) within a single AF class to provide differential treatment for a scalable video stream in order to achieve a desired packet loss behavior as defined by (1.2). In this thesis we propose two mapping strategies for scalable video over AF (as described later).

Moreover, DiffServ is a commitment based service network; hence it provides differential forwarding treatments to the packets according to a "paid price". The

differential treatment of a packet is based on a commitment between the service provider and the user. This commitment allows a user (or sender) to utilize a *committed* amount of network bandwidth for a reliable delivery of its packets. Under normal conditions, a service provider does not allow a user (or users) to exceed the available network bandwidth. If a service provider allows a user (or users) to exceed the available network bandwidth, such phenomenon is called as *over-commitment* (or overbooking). This work also analyzes the effect of such over-commitments on the PLR performance of scalable video transmission. The performance of video transmission is analyzed under several network overload levels, which are applied as over-commitments of network bandwidth, ranging from 0% to 100%.

The paid price for the user-service provider commitments can be represented by a corresponding set of "pricing parameters". We used such pricing parameters to define approaches to improve the user video throughput by controlling PLR in a DiffServ domain, at several network overload levels, according to equation (1.2). To achieve this goal, in addition to the proposed mapping strategies we also proposed two pricing models besides the traditional flat rate pricing technique.

We present results obtained by performing a number of simulations on the *Network Simulator* (NS) [14] (version ns-2.1b8a) for each proposed mapping strategy and pricing model. After evaluating the video performance, we compared the associated overall "cost" of each strategy. Finally, we proposed a *low-cost* framework to attain a desired video quality transmission and a sustained performance for several network overload levels.

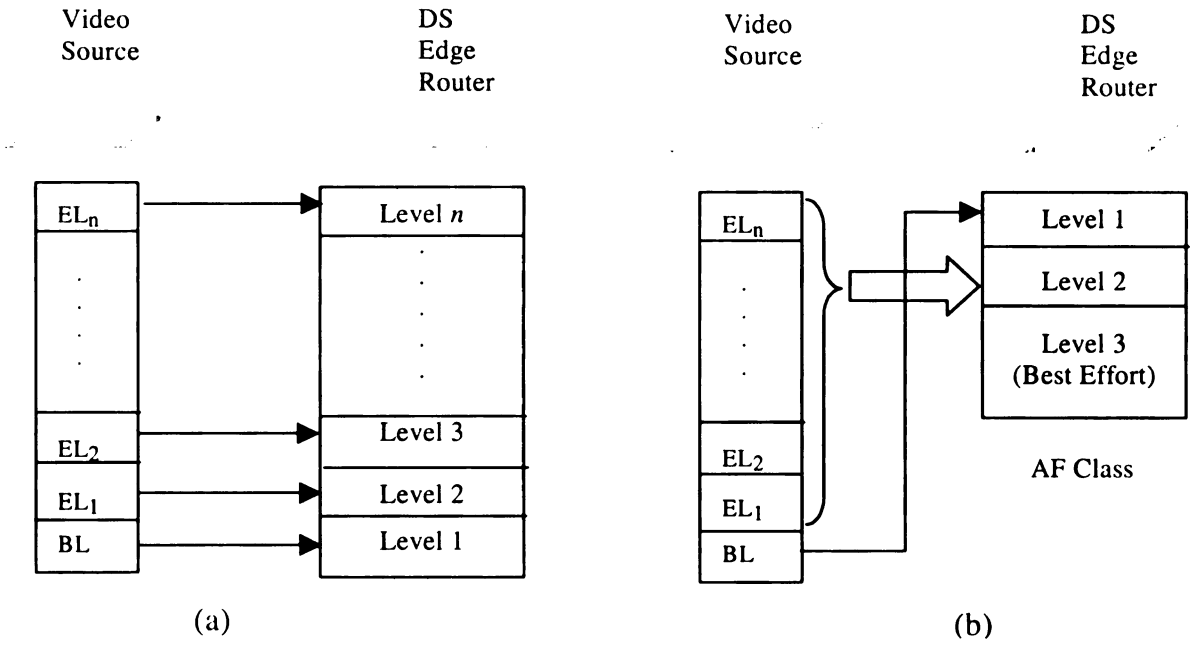

Figure 1-1 Possible mappings of streaming video over DiffServ AF Class.

1.3 Organization of the Thesis

The next chapter provides a brief overview of layered video over IP and their coding techniques, and work done in recent years to bring improvement in video quality and performance. Chapter 2 also includes a high-level overview of the architecture, elements and working details of Differentiated Services for the Internet. Chapter 3 covers the proposed mapping strategies and method of performance evaluation. The setup and description of simulation work is covered in Chapter 4. Chapter 5 includes the simulation results for network PLR behavior and performance analysis of the proposed mapping strategies and pricing models for scalable video streams over DiffServ. Finally, Chapter 6

concludes this thesis with an outline of the key findings of this work and suggestions for future work.

2 Background and Literature Review

This chapter provides some background material for three basic elements of this thesis. It starts with a brief overview of streaming video over the Internet and some description of scalable video transmissions over IP in the first section. The second section provides a detailed description of the architecture, standards and working of Differentiated Services (DiffServ) networks. It also includes a detailed overview of the assured forwarding (AF) class of DiffServ. The Weighted Random Early Detection (WRED) mechanism, an extension of Random Early Detection (RED) mechanism, is employed as the packet traffic shaping (queuing) mechanism for the simulations of DiffServ network. Hence, a brief overview of the basic WRED mechanism is also included in this section. As this thesis also focuses on the performance analysis, in which cost and pricing play a significant role, the traditional flat rate pricing technique and cost evaluation methods for DiffServ and its assured forwarding (AF) class are discussed in last section.

2.1 Streaming Video over the Internet

Streaming is a technique of transmitting real-time multimedia data, commonly audio and video, from a *host* to a *client* where the client "plays" or decodes the data as it is

received. Streaming differs from downloading in that streaming lets the viewer play-back the content in real-time after a short buffering delay, which could be up to few seconds. This buffering allows the application to maintain a continuous playback even during minor network congestions. UDP is the transport protocol used for interactive streaming video applications over the Internet. The Real Time Protocol (RTP) [6] and Internet Multicast Backbone (Mbone) [21] also recommends the use of UDP for video transmissions. While giving a provision of sending data at any desired bitrate, UDP does not specify any error recovery mechanism; therefore it is an unreliable transport protocol. During congestion, the unreliable approach of UDP results in an increased packet loss ratio (PLR), which consequently reduces the user-perceived video quality.

A number of approaches for UDP transmission improvement using adaptive rate control methods were presented in [27]. In these approaches, the sending bitrate of transmission can be reduced in response to a "packet loss" notification reported by the receiver. For this purpose, use of real time control protocol (RTCP) is recommended. Such adaptive rate control methods are helpful in improving the overall losses, but do not provide any assurance to maintain the video quality and transmission performance during multiple levels, ranging from 0% to 100% network congestion. Error concealment methods, for example Forward Error Correction (FEC), and retransmission techniques for multicast streaming [28] were also suggested to bring improvement in UDP transmissions. The error concealment methods help to improve the user perceived video quality, but are only applicable after loss notifications.

Focusing over the available approaches of video packet transmissions, *motion JPEG* and certain wavelet-based schemes use the Intra-frame coding techniques. Whereas

popular video standards like H.261 [22], H.263 [23], MPEG1 [24], MPEG2 [25] and MPEG4 [26] use inter-frame coding technique. Inter-frame coded video packet streams are more sensitive to error and packet losses during their transmission. Every packet loss during transmission affects the corresponding frame and error(s) in one frame propagates to the rest. Mechanisms like *High Priority Protection method* (HiPP) [29] and *Fine-Grained Loss Protection* (FGLP) [30] are suggested to bring improvement in the packet loss resilience of MPEG video streams.

2.1.1 Scalable Video over IP

Typical scalable video applications (using standard like MPEG-4) send video streams composed of one *base layer* (BL) and one or multiple *enhancement layers* (ELs). For this purpose, a scalable video encoder generates two bit streams; one for the base layer and the other(s) for the enhancement layer(s). These streams are encoded for transmission at some bitrate r_b for the base layer and r_e for the enhancement layer. Thus total bitrate (R_{max}) of a video packet stream is the sum of individual bitrates of base layer and enhancement layer:

$$R_{\max} = r_b + r_e \tag{2.1}$$

Under normal conditions, the total bitrate remains less then the available network link capacity (C):

$$R_{\max} < C \tag{2.2}$$

Subsequently, the video server packetizes the video bit streams in packets of specified size, and transmits packets over network link according to a corresponding bitrates. This sequence of processes at the video server is illustrated by Figure 2-1.

At the receiver end, the data packets are received at their respective bitrates. The *decoder* then reconstructs the original bit streams from such packets. Base layer packets contain the data required to produce a minimum acceptable quality video. Whereas enhancement layer provides additional information, which enhances the quality of video produced by the base layer.

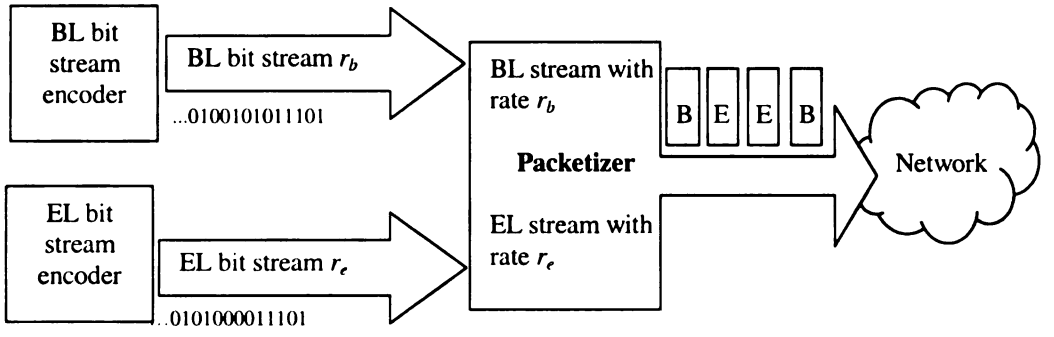

Figure 2-1 Schematic diagram of streaming video encoder.

A typical scheme of encoding base layer and enhancement layer of a video frame is shown in Figure 2-1. The video layers are encoded at their corresponding bitrates r_b and r_e. A single enhancement layer with a high sending bitrate r_e, can be divided into multiple enhancement layers with lower bitrates r_i, such that $r_i < r_e$. A typical method is partitioning of the high sending bitrate r_e of one enhancement layer into equally divided (say n-1) small enhancement layers with lower sending bitrate r_i:

$$r_i = \frac{r_e}{n-1} \quad where\ i = 1...(n-1)$$
$$\Rightarrow r_e = \sum_{i=1}^{n-1} r_i \tag{2.3}$$

Such partitioning of a high sending bitrate r_e of one enhancement layer into (n-1) enhancement layers is shown in Figure 2-2.

The base layer and the enhancement layer(s) are inter-related with each other for a particular frame as shown by the dotted box in Figure 2-2 (a), and Figure 2-2(b). The occurrence of any packet loss during the transmission of base layer makes the whole sequence of enhancement layers (for that particular frame) useless. Such dependence scheme of video layers clearly requires the following simple priority scheme in terms of PLR, during transmission as mentioned by equation (1.2):

$$PLR_i < PLR_{i+1} \tag{2.4}$$

where i is the number of corresponding video layer with base (i =1) as the first layer in the sequence.

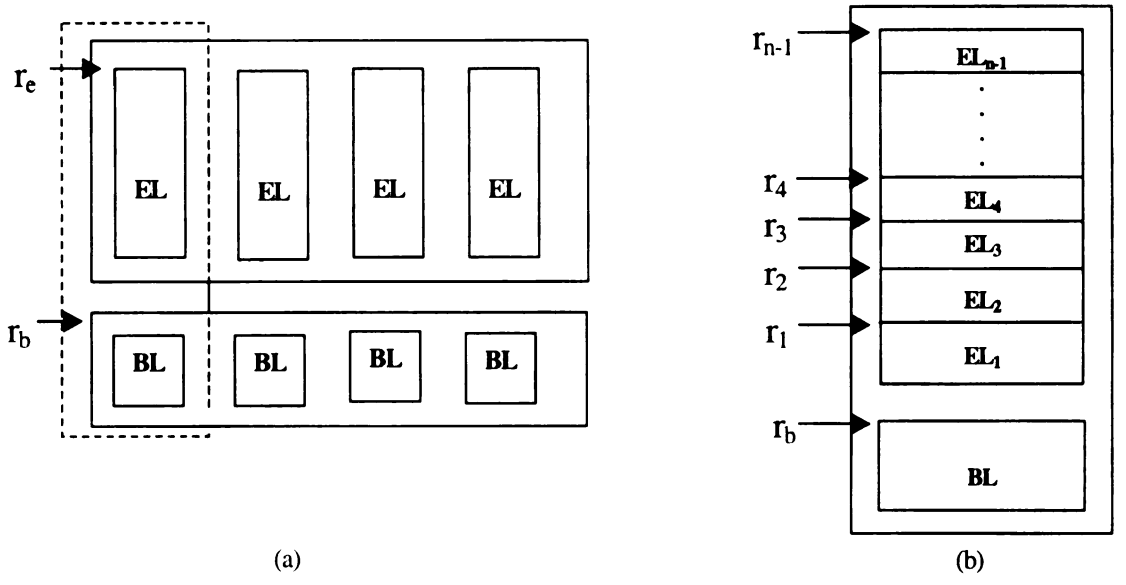

Figure 2-2 Layering of video stream (a) one BL and one EL (b) one BL and n-1 ELs.

2.1.2 Weight Distribution for Video Layers

As described later in this thesis, identifying the overall "cost" of a scalable-video mapping-strategy over DiffServ networks requires a quantitative measure for the *level-of-importance* associated with the different video layers. It is clear from the above description of scalable video that the base-layer, for example, should encounter the minimum (or a virtually null) PLR value. The first enhancement layer should encounter a PLR smaller than the second enhancement layer PLR, and so on. This, however, does not provide a direct quantitative measure for the relative importance of the different layers.

In this thesis, *weight* is a term used for indicating the comparative importance of one video layer to another. The higher the *weight measure* of a particular video layer is, the higher the "performance cost" (or performance penalty) that is associated with loosing data from that layer.

Assigning a quantitative measure for the different layers of a scalable video stream is a very challenging task. This is due to the variety of scalable video coding schemes, and more importantly, is due to the wide range of possible video sequences, each of which could have a different *weight measure* model [55]. Here, we resort to a generic weight measure that captures a variety of prioritization models for scalable video layers.

In a scalable video stream containing one base layer and (n-1) enhancement layers, one possible approach for assigning weight (w) is based on the following exponential equation:

$$w_j = ae^{-(j-1)b} \quad where\ j = 1,2,3..(n-1),\ 0 \le a < 1,\ b \ge 0 \tag{2.5}$$

where j is the corresponding enhancement layer.

In the above equation, the parameter a denotes the weight assigned to the first enhancement layer, while variable b represents the exponential decay in the weight of higher (less important) video layers. Since the base layer usually carries the highest priority (or maximum weight), then the weight w_0 used for the base-layer should be higher than the parameter a. In this thesis, we normalize this measure by using $w_0 = 1 > a$.

As an example, a weight distribution curve for a scalable video stream containing one base layer and 9 enhancement layers is shown in Figure 2-3. This curve is obtained by using equation (2.5) with parameters a = 0.4 and b = 0.7. The same weight distribution curve is used for the analysis later in this thesis.

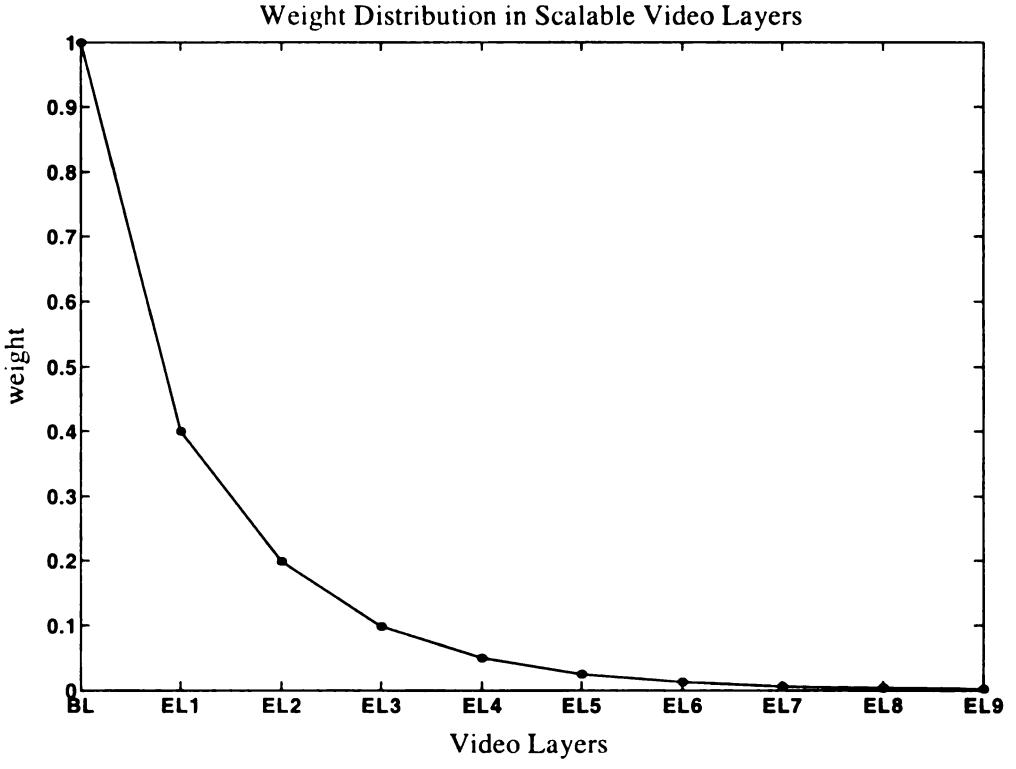

Figure 2-3 Weight distribution curve for video layers with $a = 0.4$, $b = 0.7$.

2.2 Differentiated Services for the Internet

Proposals that have been put forward by the IETF for improving the Internet QoS are based on two types of approaches. The first approach is *Fine Grained,* which provides QoS for individual applications and packet flows, and the second approach is *Coarse Grained*, which provides QoS to the larger classes of data or aggregated packet traffic. The proposals that belong to these two approaches are considered as deployable QoS models in the form of Integrated Services (IntServ) [32] and DiffServ [33] respectively. Due to the complexity associated with providing QoS guarantees to individual flows (i.e., IntServ), DiffServ has been receiving a great deal of attention as a realistic and practical alternative for providing some form of QoS services over the global Internet. Consequently, in this thesis, we limit our discussion to DiffServ-based scalable video services.

Implementation of DiffServ routing over the Internet needs: (i) a proper definition of the packet classes, (ii) method(s) of mapping each packet to a corresponding class, and (iii) a suitable allocation of network resources to each class. These steps are referred to as the *standardization* processes. From the arrival of a packet at the DiffServ domain boundary router, its classification, sorting, queuing and forwarding towards its destination are performed under a number of "standardized" mechanisms [34]. IETF DiffServ working group defined the directions to be taken for the implementation of such mechanisms in the form of Internet drafts and requests for comments (RFCs), which are easily accessible on the Internet [51].

2.2.1 DiffServ Architectural Overview

DiffServ network is a simple, well-defined set of building blocks [35]. Figure 2-4 shows a typical DiffServ domain. In addition to the links and the switches, a DiffServ domain contains two types of network nodes that are referred to as, *Edge Routers* (boundary nodes) and *Core Routers*.

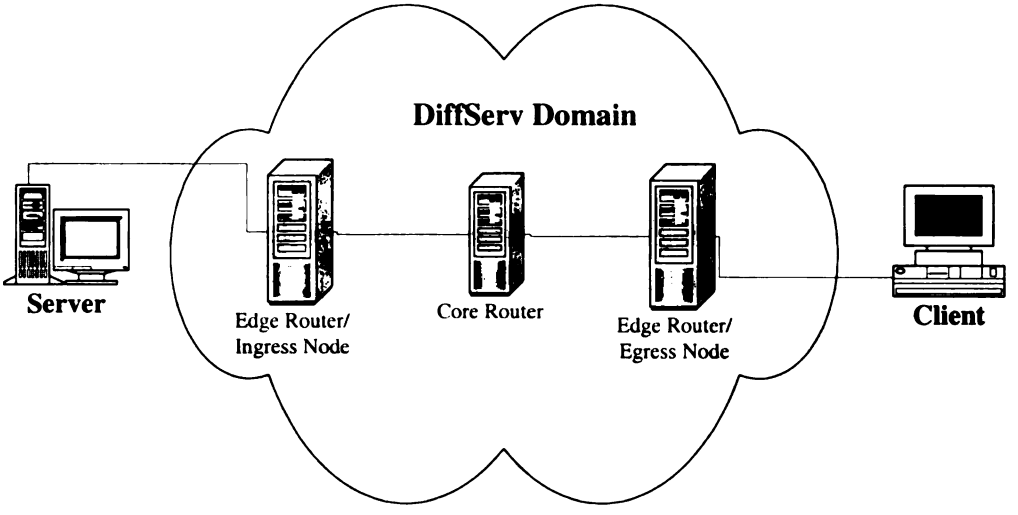

Figure 2-4 A typical DiffServ Framework.

The edge routers are the DiffServ domain ingress/ egress nodes. This implies that every packet entering or leaving the DiffServ domain needs to pass through an edge router. An edge router is responsible for *Traffic Conditioning*, which includes the classification and appropriate forwarding of packets according to their corresponding *Service Level Agreement* (SLA). The SLA is a commitment that is established between the service provider and the user (normally a sender). The primary task of an edge router is to classify and shape the incoming packet stream from a user according to its predefined service profile in the SLA. The SLA service profile contains the committed rate and packet burst size values. The edge routers compare the incoming packet

parameters (for example bitrate) with a corresponding specified value in the SLA profile. The packet(s) exceeding these profile values are classified as *out of profile* and are eligible to get downgraded or even discarded at the DiffServ domain boundary. The *in profile* packets are eligible to be forwarded towards their destinations according to their given priorities. If required, core routers can also perform an additional traffic classification.

Figure 2-5 shows the elements of an edge router, which are required to perform traffic conditioning. These elements are: (i) *Classifier*, (ii) *Meter*, (iii) *Marker*, and (iv) *Shaper/ Dropper*. The function of each element is described in the subsequent sections.

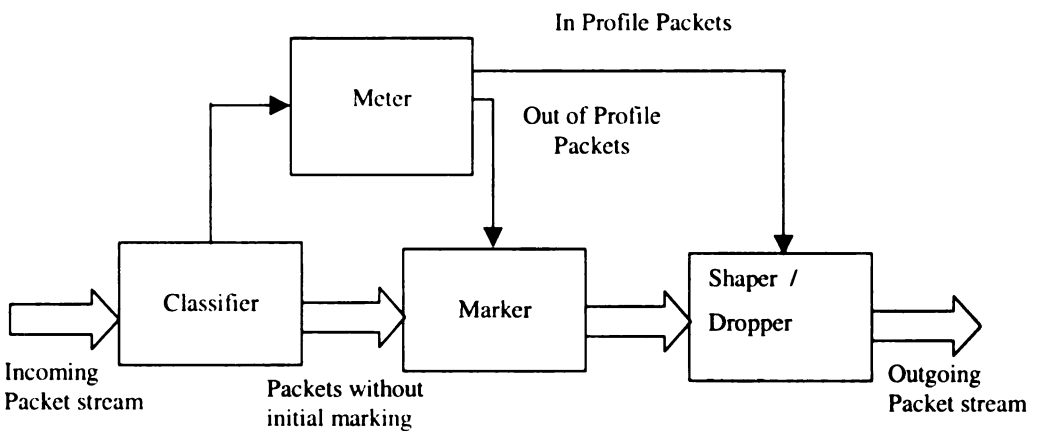

Figure 2-5 DiffServ Edge Router Components.

2.2.2 Traffic Classification

According to the DiffServ standard specifications [33][34], the classifications of every incoming packet in a DiffServ domain can be performed either by using a Behavioral Aggregate (BA) Classification [33] method, or by a Multi Field (MF) Classification [49] method. A set of packets sharing the same level of network resources

across a DiffServ domain in one direction is referred to as *behavioral aggregate* (BA). BA classification requires a proper definition of a *DiffServ Code Point* (DSCP) value in the packet header [33]. Such code point helps an edge router classifier to identify and allocate the corresponding service level to the packet.

The other method is multi field (MF) classification, which uses the existing fields in the packet header (for example source address and destination address) rather than using any specific codes for packet classification.

In this thesis, we focus on the BA classification method. This method gives a simple one to one correspondence between DSCPs and the service level allocation. The DiffServ module in the simulator used in this thesis (i.e., NS) also supports the same method of packet classification. The details about the DSCPs and their corresponding service levels, also known as *Per Hop Behaviors* (PHBs) [48], are discussed in the following section.

2.2.2.1 DSCP and PHBs

A DSCP is a 6-bit binary code in the packet header. The DSCP value [48] of a given packet carries the information about the class and the level of service that should be provided to that packet. Figure 2-6 shows an 8-bit DiffServ field used for the DiffServ code point where six bits are defined for DSCP values and two bits are reserved for future use, and consequently are labeled as *currently unused* (CU). In order to provide a compatibility with the current IPv4, DiffServ uses the *Type of service* (ToS) byte in the IP packet header for DSCP markings, whereas *Class of Service* (CoS) field in IP header of the future IPv6 is proposed for DSCP.

Every defined DSCP value corresponds to a service level aggregate or a PHB. A PHB is a service behavior of a DiffServ node, which refers to the appropriate packet

scheduling and forwarding of each packet sharing the same BA [49]. A service provider is the one who decides the network resource allocation to a PHB for each class. Depending upon the level of packet forwarding service provide by a DiffServ node, the PHBs are divided into four categories. Each category is defined below, whereas a complete PHB tree is shown in Figure 2-7.

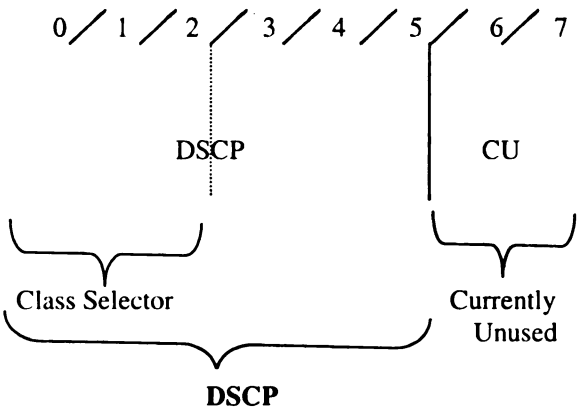

Figure 2-6 Differentiated Services Code Point (DSCP) Field.

Default PHB: A code point '000000' is assigned to default PHB, which gets the traditional best effort service[34]. Also this default PHB is assigned to every packet arriving at a DiffServ domain edge router without any code point in its header.

Class Selector PHB: A class selector PHB provides backward compatibility of DiffServ with previous IP precedence schemes [33]. DSCP values in the form of 'xxx000' are termed as class selector PHBs. Thus, according to such format of DSCP values for class selector PHB, the default PHB can also be classified as a class selector PHB [49].

Expedited Forwarding (EF) PHB: The highest prioritized service within the DiffServ PHB suite is expedited forwarding (EF) [12]. It can be termed as the "premium" service class of DiffServ[34] because it provides a low latency-low jitter service, which is considered ideal for voice and video transmissions over IP. Such premium service level that is provided by the EF class also makes it the highest priced class in the DiffServ PHB suite. Since EF has no further sub-classes defined, thus all packets carrying a DSCP value for EF i.e. 101110 are treated equally[34][37]. Hence, no service differentiation is expected between the packets utilizing the EF service.

In this thesis we focused on utilizing the assured forwarding (AF) class of services. A detailed discussion regarding AF classes and services is presented in section 2.3.

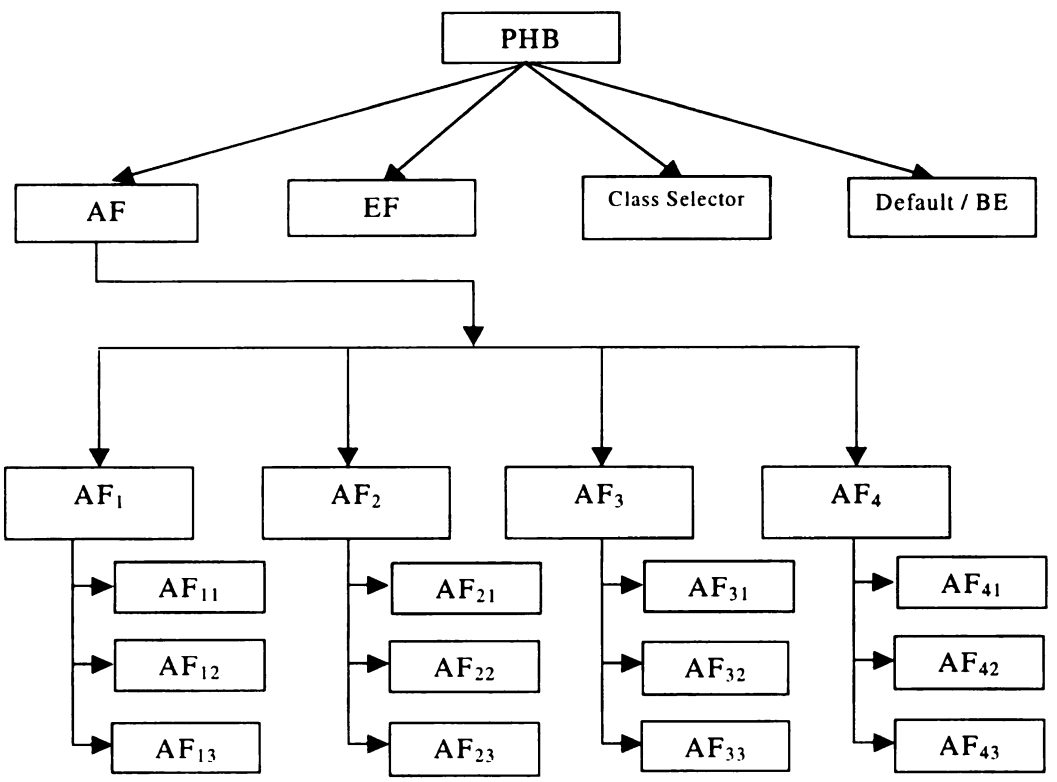

Figure 2-7 DiffServ Service Classes (PHB Tree).

2.2.3 Traffic Metering and Marking

When a packet enters in a DS domain with a corresponding DSCP in its header, an edge router *meters* such packet by using a metering-marking algorithm. Metering is the method of comparing (or testing) the packet parameters (such as bitrate, burst size etc.) with the predefined SLA profile values. A SLA profile depicts the threshold values for such parameters. As a result of metering, a packet may be categorized as an *in-profile* or an *out-of-profile* packet. If the packet's parameters are found exceeding the given threshold limits, the corresponding packet is categorized as out of profile, otherwise it becomes an in-profile packet. As mentioned earlier, an out-of-profile packet is eligible to be downgraded to a lower priority level within an AF class, where it might get discarded.

Metering and marking are two sequel packet processes at an edge router. In other words, marking of a packet "within" a DiffServ domain depends upon the result of packet metering. Moreover, marking of a packet within a DiffServ domain is considered as '*re-marking*' unless a packet arrives at an ingress node without any DSCP in its header.

There are three popular metering-marking algorithms designed for DiffServ:

1. Single Rate Three Color marker (srTCM) [38],
2. Two Rate Three Color marker (trTCM)[39], and
3. Time Sliding Window Three Color Marker (TSWTCM) [40].

TSWTCM [40][41] was specifically developed to support TCP traffic. The other two algorithms srTCM and trTCM can be used for metering of UDP traffic. In this thesis we used srTCM and trTCM as our marking algorithm. These algorithms are described in Appendix at the end of this thesis.

There are two bitrate thresholds that are used in metering-marking algorithms: *Committed Information Rate* (CIR) and *Peak Information Rate* (PIR)[1]. CIR is the minimum network bandwidth, which a service provider is committed to provide a user for a certain flow of packets, for a guaranteed delivery of packets. Basically, CIR is the throughput rate that a user negotiates with a service provider, and the service provider will guarantee that rate. One way the service provider guarantees CIR is by dropping non-CIR traffic.

PIR is the peak rate of a packet flow that is allowed within a profile. Typically PIR is not less than CIR, and not greater than the available link capacity *C*. CIR does not limit the user to send data at the committed bitrate, but the user should not violate the predefined threshold limits of CIR in order to keep the packets in-profile.

Threshold parameters for limiting the burstiness of incoming data streams are: *Committed Burst Size* (CBS) and *Peak Burst Size* (PBS). Burstiness is the maximum amount of bits that a network agrees to support during a certain time interval. These burst sizes are related to their corresponding rates by:

$$CIR = \frac{CBS}{T_c}$$
$$PIR = \frac{PBS}{T_p} \tag{2.6}$$

where T_c and T_p are the corresponding time intervals to evaluate CIR and PIR.

The two bitrates (CIR and PIR) along with their corresponding burst sizes (CBS and PBS) work as two thresholds limits for incoming packet traffic in a DiffServ domain.

[1] srTCM algorithm does not employ PIR as a parameter, however the other two algorithms use CIR and PIR both.

2.2.4 Traffic Shaping

Traffic Shapers are the packet queuing mechanisms at the edge router. After metering and marking, packets are allowed in the buffer queues according to their respective marked or re-marked DSCP. During this temporary storage period, service differentiation is provided by means of a packet queuing algorithm.

Random Early Detection (RED) [15] is one of the congestion avoidance mechanisms that has been successfully deployed in the Internet. It was initially proposed to provide scalability in TCP transmissions. RED gives a provision to control the packet losses during congestion by means of its parameters. While focusing over the AF class of DiffServ, which defines three levels of packet drop precedences, RED can be utilized to establish multiple queues. According to the predefined priorities of each queue, RED parameters can be selected. Such scheme of deploying multiple RED queues with different set of parameters for each queue is called *multiple RED* (MRED) or *weighted RED* (WRED)[50][52]. WRED is a successfully deployed queuing mechanism in Cisco routing software IOS™ release 12.2 [53] to support DiffServ. According to the WRED mechanism, one physical RED queue is divided into multiple virtual RED queues; each virtual RED queue use different set of RED parameter values and follows the basic RED algorithm. The outline of RED algorithm is as follows:

RED Algorithm: The RED algorithm is based on continuous calculations performed for every incoming packet to measure the average length (*avgLn*) of a packet queue in the router. Typically, RED requires two queue length thresholds min_{th} and max_{th} and a drop probability max_p. The RED algorithm for packet queuing and discard can be written as:

i. If the average length of the queue drops below the minimum threshold value min_{th}, RED allows the packet in the buffer, else

ii. If the average length of the queue becomes in between min_{th} and max_{th}, RED discards incoming packets randomly with a drop probability of max_p , as shown in Figure 2-8, else

iii. If the average queue length exceeds the maximum threshold max_{th}, every incoming packet gets discarded, until the average queue length drops below the max_{th}.

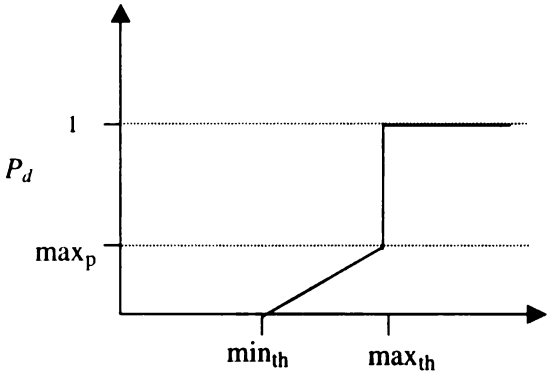

Figure 2-8 A typical RED Algorithm.

We used WRED to establish three packet queues, each dedicated to an AF priority level, by applying a distinct set of RED parameters for each packet queue. *Jacobson et al.* in [15] provided the guidelines as thumb rules for selection of the RED queue length thresholds (i.e. min_{th} and max_{th}). In a multiple queue environment, a given queue can be assigned with a higher priority by using packet drop probability parameter max_p. A packet queue assigned with the highest value of max_p should be the lowest priority queue.

In our case where we need to establish three priority queues, the parameter max_p can be adjusted by using the following relation.

$$\begin{aligned} \max_p(q\_1) &= \frac{1}{\Delta}\max_p(q\_2) \\ \max_p(q\_1) &= \frac{1}{\delta}\max_p(q\_3);\ \Delta > 1,\ \delta > \Delta \end{aligned} \tag{2.8}$$

In the above given relation, the values for parameters Δ, δ and an initial value of parameter max_p of any queue can be selected in such a way that a highest priority will be assigned to q_1, medium for q_2 and lowest for q_3. The setup and configuration of WRED parameters employed for simulations is described in section 4.4.2.

Lyles et al. in [16] suggested that the deployment of RED is not a good choice over the best effort Internet. However, the scalability features of RED (specially in case of WRED) make it a good choice for its use in multiple queue networks, such as an AF service class of DiffServ. Several other extensions in RED algorithm were proposed such as *RED with IN/OUT* (RIO) [17], RED+ [18], Adaptive RED (ARED) [19] and Flow RED (FRED) [20]. Most of the work is done to bring improvements in the fair network resource utilization during transmission, by using fair queuing methods like weighted fair queuing (WFQ) [42][43][44]. Whereas IETF RFC 2963 [45] proposed rate-adaptive shapers by using First In First Out (FIFO) queuing methods.

2.3 Assured Forwarding PHB

Assured Forwarding (AF) is a medium price service as compared to the EF and BE services offered by DiffServ. There are four independent AF classes that are defined in the DiffServ standard, where each AF class is in each DiffServ node allocated a certain

amount of forwarding resources (buffer space and bandwidth). Each AF class provides three levels of packet drop precedences (or packet forwarding priorities) [13]. An AF class can be represented as AF_{xy}, where *x* denotes the number of AF class ranging from 1 to 4, and *y* denotes the packet priority (or drop precedence) level ranging from 1 to 3. Priority level 1 in each AF class carries the highest priority (or lowest packet drop precedence). Since each AF class is independent of the other classes, it may receive an equal share of network resources, which is further distributed among three priority levels. Therefore, for the remainder of the thesis, we focus on mapping scalable video onto a single AF class and its corresponding three priority queues. The priority levels and their corresponding standard DSCPs are given in Table 2.1.

Table 2.1 Assured Forwarding PHB code points.

Drop Precedence	**AF 1**	**AF 2**	**AF 3**	**AF 4**
1) Low (Green)	AF_{11} 001010	AF_{21} 010010	AF_{31} 011010	AF_{41} 100010
2) Medium (Yellow)	AF_{12} 001100	AF_{22} 010100	AF_{32} 011100	AF_{42} 100100
3) High (Red)	AF_{13} 001110	AF_{23} 010110	AF_{33} 011110	AF_{43} 100110

A packet is termed as a colored packet if it is marked with an AF service class DSCP in its header. A packet is called as a *green* packet if it is marked with the DSCP of the highest priority level AF_{x1}, a *yellow* packet if marked for the medium priority level AF_{x2} and a *red* packet if marked for the lowest priority AF_{x3}. If a packet initially marked as a green packet becomes out-of-profile as a result of metering, it can be downgraded as a yellow packet by *re-marking* a new DSCP at the ingress router. No priority level

upgrades are defined within a single DiffServ domain. Throughout this thesis we will use these priority levels in terms of colors, as used by the standard DiffServ marking algorithms. The effect of the number of drop precedences within an AF service class is discussed in detail by *Goyal et al* in [36] and [42].

As mentioned earlier, the highest priority level within a given class is level 1 (green), the medium priority level is 2 (yellow), and the lowest priority level is 3 (red). Thus the probability P_d of loosing a colored packet during transmission can be expressed as:

$$P_d(green) < P_d(yellow) < P_d(red) \tag{2.9}$$

In this thesis we analyzed performance as a function of the "cost" associated with the services provided by each priority level of an AF class during transmission. The cost is a measure of the price paid by the user for utilizing an AF class. The associated cost of each priority level can be defined as a cost function $c(x)$, where x is the corresponding priority level utilized by a packet flow within a single AF class, then:

$$c(green) > c(yellow) > c(red) \tag{2.10}$$

The pricing and cost calculation for an AF class is described in the next section.

2.4 Pricing and Cost Calculation for Assured Forwarding Class

A DiffServ network provides service differentiation at its ingress nodes. This makes *sender-to-pay* type pricing as the most appropriate pricing approach for the network services [47]. This type of service pricing is a departure from the traditional Internet receiver-paid flat rate model. The SLA between the sender and service provider adjust the profile values for the network resource allocations.

As introduced earlier in Chapter 1, an important issue related with the pricing of these services is the *overbooking* of channel bandwidth by the service providers. Service providers usually overbook the capacity of their network channels, hoping that users (customers) will not make excessive demands on the network at the same time. But if the service provider miscalculates, traffic will be dropped, even if it is guaranteed, although the traffic with no guaranteed CIR will be dropped first. This phenomenon implies that, for n packet flows over a channel of maximum capacity C (bps), the committed bandwidth i.e. CIR should follows:

$$\sum_{i=1}^{n} CIR_i < C \tag{2.11}$$

If a DiffServ node starts operating over or very close to the available channel (or link) capacity, the drop precedence in an AF class becomes redundant. Consequently, three-drop precedence levels starts giving the same performance as two. We used this phenomenon of overbooking by sending overload traffic (as committed traffic) to analyze its effect on the transmission performance.

DiffServ follows a simple direct proportion of pricing and their related services. While focusing on an AF class, it has three price classes; one belongs to each priority level. These pricing classes can be implemented in two ways, either using *flat rate* pricing scheme, which is the extreme capacity usage pricing where the available capacity always equals to a fixed fraction of the link capacity, or by *usage based pricing scheme*. A pricing scheme between these two extreme types was describe by *Semret et al* [46] as *explicit pricing of resources*. The cost affecting parameters in a DiffServ network are related to the bandwidth and buffer management. CIR and PIR are considered as the major pricing and cost-affecting constraints in an AF class of DiffServ, when a certain

cost is associated with every increase in CIR value (whereas PIR is used as an upper threshold which equals to the available bandwidth).

Let K_g, K_y, and K_r are the prices (as some \$) associated with every unit-committed rate (bps) of the corresponding G number of green, Y-yellow and, R-red packet flows utilizing an AF class. The corresponding sending bitrates of each green, yellow and red packet flow are r_g, r_y, *and* r_r, along with their corresponding committed information rates CIR_g, CIR_y and CIR_r respectively. Then the individual cost c_g, c_y and c_r of each colored packet flow over an AF class can be calculated as follows:

$$\begin{aligned} &\text{Green packets} && c_g = K_g \sum_{g=1}^{G} \min(r_g, CIR_g) \\ &\text{Yellow packets} && c_y = K_y \sum_{y=1}^{Y} \min(r_y, CIR_y) \\ &\text{Red packets} && c_r = K_r \sum_{r=1}^{R} \min(r_r, CIR_r) \end{aligned} \tag{2.12}$$

Thus the total cost (c_{AF}) for the AF service utilized by the user can be given by the following equation:

$$c_{AF} = c_g + c_y + c_r \tag{2.13}$$

By inserting the corresponding values of c_g, c_y and c_r from equation (2.12) in (2.13) we get,

$$c_{AF} = K_g \sum_{g=1}^{G} CIR_g + K_y \sum_{y=1}^{Y} CIR_y + k_r \sum_{r=1}^{R} CIR_r \tag{2.14}$$

Every incoming colored packet needs to satisfy the upper and lower bound conditions during the metering process, in order to maintain its status as in-profile. As mentioned by [38][39] and [40] the marking (or re-marking) of an incoming packet is done according to

its predefined SLA profile values. The metering-marking algorithms at the edge router use these values. According to the suggested DiffServ metering-marking algorithms (like srTCM and trTCM) the upper and lower bound conditions for every incoming packet with rate r_i can be given as:

$$\text{Green packet flows} \quad \begin{cases} r_i \leq CIR_i \\ r_i < PIR_i \end{cases} \tag{2.15}$$

$$\text{Yellow packet flows} \quad CIR_i < r_i \leq PIR_i \tag{2.16}$$

$$\text{Red packet flows} \quad \begin{cases} r_i > CIR_i \\ r_i > PIR_i \end{cases} \tag{2.17}$$

2.4.1 Flat Rate Pricing Technique

The most common pricing method for commitment based networks like DiffServ is the flat rate pricing. Flat rate pricing technique estimates CIR of any packet flow i as a fixed fraction of its bitrate r_i. The general equation of this model for a video stream containing n video layers (or packet flows) can be expressed as:

$$CIR_i = pr_i \qquad i = 1,2,3...n \text{ and } p \geq 0 \tag{2.18}$$

Any selection of p in the above equation can be adjusted according to the desired priority level of a packet flow. For instance, a simple approach for selecting p for different colored packet flows utilizing an AF class according to the thresholds mentioned in equations (2.15) to (2.17) should be:

$$\text{Green packet flows} \quad p \geq 1 \tag{2.19}$$

$$\text{Yellow packet flows} \quad 0 \leq p < 1 \tag{2.20}$$

Red packet flows	$0 \leq p < 1$	(2.21)

3 Mapping Strategies and Method of Analysis

In this chapter we present different strategies developed to improve video quality during the transmission of scalable video streams over an AF class of DiffServ networks. The method for mapping video layers over any desired AF class is described in the first section. Sections 3.2 and 3.3 describe the proposed strategies aimed to achieve service differentiation for scalable video layers over an AF class. The same section also includes a description of the proposed pricing models. These pricing models are developed to control the packet loss ratio (PLR) of multiple video layers, utilizing a single priority level of an AF class. Section 3.4 describes the method of performance evaluation for scalable video streams over an AF class.

3.1 Packet Mapping

Mapping is a term, which we use to illustrate the selection of a DiffServ class that is employed for transmitting a packet. A packet is mapped to a DiffServ class by marking a DiffServ code point (DSCP) value in its packet header "before" its transmission. By marking all the packets belonging to a given video layer with a similar DSCP, the

complete video layer can be mapped to a corresponding DiffServ class. This work focuses on mapping of the video layers to the priority levels of an AF class.

The mapping of various (say n) video layers in a scalable video stream over an AF class can be divided into two categories. The first category is *One-to-One mapping*, which is applicable to video streams containing up to three video layers i.e. $n \leq 3$. The second category is *Multiple-to-One mapping*, which is applicable to video streams containing more than three layers i.e. $n > 3$. Under each of the above two *mapping categories*, one can define a variety of approaches for mapping scalable video layers onto AF priority levels. We refer to these approaches as *mapping strategies* (as explained below). Moreover, it is important to note that some scalable video coding methods have the flexibility of generating any desired number of enhancement layers. Consequently, these methods can employ strategies from the one-to-one or multiple-to-one mapping category. An example of such scalable video streams is the ones generated by the MPEG-4 FGS coding method.

As described earlier, in order to improve the quality of a scalable video stream containing upto n video layers that are transmitted over a lossy packet network, the PLR and throughput (T) associated with a video layer i, should meet the following constraints:

$$\begin{aligned} PLR_i < PLR_{i+1} \quad i = 1,2,3...n \\ T_i > T_{i+1} \end{aligned} \tag{3.1}$$

All mapping strategies in this thesis are developed with an objective of achieving the network PLR constraint that is expressed in equation (3.1). For each mapping strategy, the *Committed Information Rate* (CIR) is used as a "commitment parameter" as well as a "cost parameter" for the service agreement. In other words, we consider CIR as a measure of cost since the higher the value of CIR that is stipulated by the service level

agreement (SLA), the higher the cost that is associated with this agreement. Meanwhile, the higher the CIR, the lower the PLR that can be achieved. Consequently, the *mapping strategies*, which are defined as functions of the CIR parameter, represent a form of *pricing models*. Hence, below, we use the two expressions: *mapping strategy* and *pricing model* interchangeably. More importantly, CIR plays a key role in expressing and defining the proposed strategies for mapping scalable video over AF DiffServ classes. As described later in this chapter, by defining different functions of CIR for the different video layers, one can achieve different levels of treatment (by a DiffServ node) for these video layers. This will provide the differentiation in PLR performance that is expressed in equation (3.1).

In this thesis, we consider three strategies for mapping scalable video layers to the AF priority levels. We refer to these mapping strategies as *flat rate*, *linear*, and *exponential*. (These strategies are described in detail below). For the one-to-one mapping category, we only employ the flat rate based technique. This is due to two reasons: (1) The small number of layers associated with each AF priority level (only one) in the one-to-one mapping category significantly limits the number of mapping strategies that one can use; and (2) The flat-rate based one-to-one mapping represents a benchmark for the simplest approach for mapping scalable video layers over DiffServ. For multiple-to-one mapping, *linear* and *exponential* pricing models (mapping strategies) are also used in addition to the flat rate pricing technique. After the above high-level description of the different mapping strategies and categories, for the remainder of the thesis, we will use the two expressions *mapping strategy* and *mapping category* interchangeably.

Before proceeding further, we highlight one more item regarding the identification of the different packet flows of scalable video. In order to provide identification for the packets belonging to different video layers during transmission, the use of a unique *flow-id* for each video layer is proposed. A flow-id is a set of 16-bit address field (containing source and destination IP addresses) in an IP packet header and a 32-bit address field (containing the corresponding source and destination port addresses) in the UDP packet header. Thus all packets belonging to a single video layer are assigned the same flow-id. Use of a distinct flow-id helps in identifying each video layer as a distinct packet flow. Hence, a unique set of profile values (containing rate thresholds like CIR and PIR) can be defined for each layer. As described earlier in section 2.2.3, these profile values are used by the metering-marking algorithm at the DiffServ ingress router for traffic conditioning.

3.2 One-to-One Mapping

This mapping category is applicable to video streams that are composed of one base layer (BL) and a maximum of two enhancement layers (EL1 and EL2). The base layer is transmitted with a bitrate r_b, whereas the corresponding bitrates of the two enhancement layers are r_1 and r_2, respectively. Each video layer is mapped to one of the three priority levels of an AF class as shown in Figure 3-1. The packets belonging to the base layer are mapped to the highest priority level AF_{x1} as green packets, and the packets belonging to EL1 and EL2 are mapped to AF_{x2} and AF_{x3} as yellow and red packets respectively. By applying this strategy, we intend to obtain a distinct network packet forwarding behavior for each layer governed by equation (3.1).

For this mapping category, the pricing model of choice is flat rate based pricing model. As we described earlier, a pricing model deals with the network bandwidth commitment between the user and the service provider. Thus, for maximum utilization of the available bandwidth allocated to a single layer mapped onto a single priority level, no other bandwidth commitment model could perform better than flat rate based pricing model. Also, the limited number of one video layer per priority level does not allow using linear or exponential model, as they will reduce the possible throughput while leaving some of the available bandwidth unused. Furthermore, we will use same approach for the calculating CIR of base layer in every strategy. This is because the base layer is the only layer utilizing the highest priority level in both mapping categories in the remaining thesis.

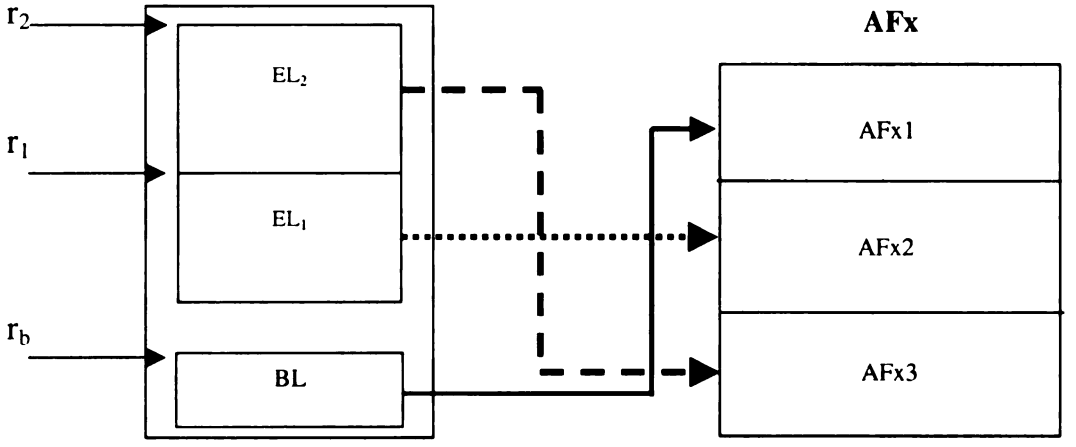

Figure 3-1 One-to-one Mapping.

3.3 Multiple-to-One Mapping

Under this category, the number of video layers (i.e. n) in a stream may be greater than the available AF priority levels (i.e. $n > 3$). This mapping category can be considered as a general case, which is applicable for any number of video layers in a

stream. Under this mapping category, a video stream containing one base layer and n-1 enhancement layers is mapped onto two priority levels of an AF class. As shown in Figure 3-2, the packets belonging to the base layer are mapped to the AF_{x1} priority level and packets belonging to all enhancement layers are mapped to the AF_{x2} priority level. This mapping scheme enables all base layer packets to be recognized as green packets and packets belonging to all enhancement layer as yellow packets.

With this mapping strategy, the lowest possible PLR is expected in the base layer. Therefore, during congestion the green (BL) packets would retain the lowest probability of being discarded from the buffer queues. Hence, this strategy minimizes the loss of green packets at the cost of losing more yellow packets.

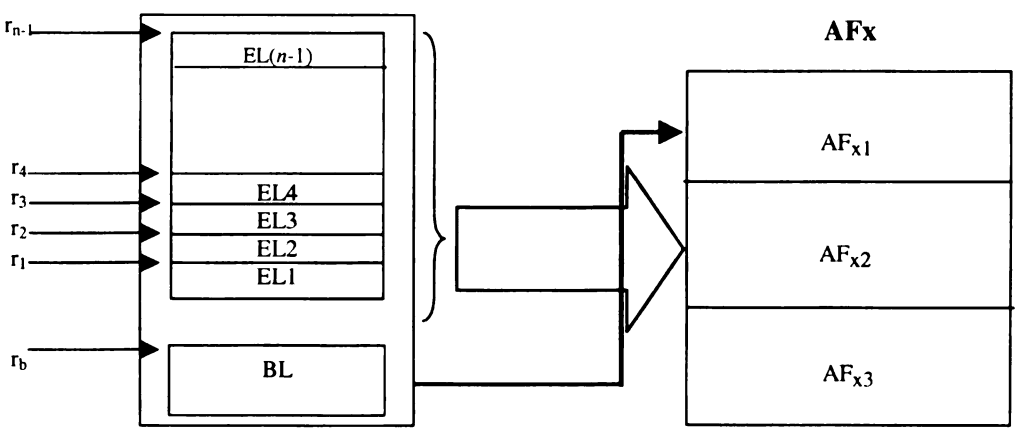

Figure 3-2 Multiple to One Mapping.

All enhancement layers in this mapping category are using a single AF priority level. This may allow the network to treat every enhancement layer equally regardless of their corresponding weight. Thus, an equal PLR for each enhancement layer may ensue, which is undesirable. In order to avoid this situation and achieve the PLR constraints for each

enhancement layer according to the inequality in equation (3.1), we propose two pricing models as strategies for calculating the CIR for each uniformly mapped enhancement layer. As mentioned earlier, these pricing models are developed as a function of CIR, while keeping in mind the inversely proportional relationship between the applied CIR for network service agreement and the PLR of the corresponding packet flow. Thus by using a higher CIR value for a given enhancement layer will result in a lower PLR. In order to obtain a higher PLR for every increasing enhancement layer, the corresponding CIR should be decreased monotonically. To achieve such a CIR scheme, we propose two strategies: *linear* and *exponential*. These approaches are described as two possible pricing models in the subsequent sections.

3.3.1 Exponential Pricing Model (EPM)

Using this model, we aim to evaluate CIR values for the given (n-1) enhancement layers in an exponentially decreasing manner, starting from the highest value for the first enhancement layer to the lowest value for the last enhancement layer in a scalable video stream. Therefore, using this approach an exponentially increasing PLR response is expected for an increasing number of enhancement layers, which satisfy equation 3.1.

As mentioned in equation (2.16), the CIR of a yellow packet flow, which constitutes an enhancement layer in this case, should be less than its corresponding bitrate in order to remain in-profile. We used the following equation to estimate a corresponding CIR value, for a given enhancement layer i with sending bitrate r_i. This CIR estimate will always remain less than the given threshold r_i for (n-1) enhancement layers:

$$CIR_i = r_i - (r_i)^{\alpha_i};\ i = 1,2,3...(n-1) \tag{3.2}$$

In the above equation, α is used as an *exponential pricing factor*, which we define as a ratio of the i^{th} enhancement layer to the total number of enhancement layers in the video stream, i.e.,

$$\alpha_i = \frac{i}{n-1};\ i = 1,2,3...(n-1) \tag{3.3}$$

As i will approach (n-1), the CIR will decrease exponentially, which consequently will increase the corresponding PLR.

It is important to note that the above exponential mapping strategy will lead to a differentiation among the enhancement layers' CIR values even if the rate $r_i = r$ is constant for all of these layers.

3.3.2 Linear Pricing Model (LPM)

The second strategy we propose is a linear decay for estimation of the corresponding smaller CIR value for each increasing enhancement layer. We assume that all enhancement layers have the same sending bitrate i.e. $r_i = r_{i+1} = r$, according to the typical method of rate distribution as discussed in section 2.1.1. In this approach, a CIR value for a given enhancement layer is evaluated by subtracting the product of a constant parameter β and the enhancement layer number i from its bitrate r. Similar to the EPM, the general threshold condition for the yellow packet flows (as given by equation 2.13) is also followed in this approach. The CIR value for a given enhancement layer i with bitrate r can be expressed as follows:

$$CIR_i = r - i\beta,\quad i = 1,2,3...(n-1) \tag{3.4}$$

The constant β in equation (3.4) is the *linear pricing factor,* which can be defined as a ratio of the common bitrate r and the total number of the enhancement layers. According

to this definition, the estimated range of linear gradient β for a video stream containing (n-1) enhancement layers is:

$$0 < \beta \leq \frac{r}{n-1} \tag{3.5}$$

Any selection of β in the above-defined range leads to a linearly decreasing CIR with increasing number of an enhancement layer in the video stream. The LPM provides an equal difference between the CIR values for any two consecutive enhancement layers in a scalable video stream, i.e.,

$$CIR_{n+1} = CIR_n - \beta \tag{3.6}$$

3.4 Performance Evaluation

The mapping categories described in the previous two sections (3.2 and 3.3) are developed with an objective to improve the delivered quality of scalable video streams over the Internet. This improvement in quality is achieved by attaining a desirably lower PLR for a given layer in a video stream, according to its predefined weight. Since DiffServ is a commitment-based network, this improvement in quality is also associated with some price. On the other hand, improvement in the quality of received video over a lossy packet network typically depends on successful reception of packets belonging to the video layers with higher associated weights as compared to others with relatively lower weights. This has compelled us to consider the performance of the proposed strategies as, (1) not only a function of the user perceived throughput but also as a function of the associated weight of every received layer, and as (2) a function of the price paid for Diffserv. Thus, the performance of a certain mapping strategy can be assessed collectively as a measure of the user perceived quality and the price associated

with it. In this thesis, we measure the performance of a given mapping strategy in terms of the *Performance Factor* (PF), which can be defined as the ratio of the *weighted throughput* (T_w) of a given scalable video stream and the *total cost* (c_{AF}) associated with its transmission over an AF class of DiffServ. Mathematically, PF (P_f) can be expressed as follows:

$$P_f = \frac{T_w}{c_{AF}} \tag{3.7}$$

Hence, we evaluate the PF for each of the above mapping strategies to validate their performance and effectiveness. The above equation shows that for a given mapping technique the PF is maximized by increasing the weighted throughput of the video stream while minimizing the cost associated with it. We will evaluate the PF for each mapping strategy as a comparison parameter in Chapter 5. Furthermore, this measure is also used in analyzing the affect of the channel over-commitment on the performance of each mapping strategy.

The calculation of the weighted throughput and the total cost measures for the evaluation of PF are described below:

Weighted Throughput Measure: The mapping strategies discussed in the previous sections were developed with an objective of achieving a desirable PLR for different video layers according to their predefined weights in the video stream. The PLR of a packet flow is related to the channel throughput (T) according to the following relationship:

$$T = (1 - PLR) \tag{3.8}.$$

The *weighted throughput* of a given video layer can be expressed as a product of its throughput, bitrate, and its predefined weight. Thus for a scalable video stream composed

of one base layer and (n-1) enhancement layers, the weighted throughput of the video stream can be determined by using the following relation.

$$T_w = \sum_{i=1}^{n} (T_i r_i w_i), \tag{3.9}$$

where, T_i, r_i and w_i are the corresponding throughput, bitrate and pre-defined weight of the i^{th} video layer respectively.

Total Cost Measure: In order to measure the total associated cost, we measured the cost for each video layer utilizing an AF class of DiffServ in terms of its CIR. Thus, the cost associated with green (c_g), yellow (c_y) and red (c_r) packet flows mapped onto an AF class can be estimated as:

$$\begin{aligned} c_g &= K_g \sum_{i=1}^{G} CIR_i \\ c_y &= K_y \sum_{i=1}^{Y} CIR_i \\ c_r &= K_r \sum_{i=1}^{R} CIR_i \end{aligned} \tag{3.10}$$

where K_x is the bandwidth price (as some $) associated with a unit CIR (bps) for a given AF priority level x. G, Y and R being the numbers of (mapped) green, yellow and red layers respectively.

Hence, the sum of individual costs for all packet flows (i.e., green, yellow and red video layers) gives the total cost c_{AF} for the entire packet stream (in our case, the scalable video stream) utilizing an AF class.

$$c_{AF} = c_g + c_y + c_r \tag{3.11}$$

Thus if a packet stream contains G packet flows mapped as green, Y packet flows mapped as yellow and R packet flows mapped as red, then the total cost (c_{AF}) can be calculated as:

$$c_{AF} = K_g \sum_{i=1}^{G} CIR_i^g + K_y \sum_{i=1}^{Y} CIR_i^y + K_r \sum_{i=1}^{R} CIR_i^r \tag{3.12}$$

where CIR_i^x is the CIR for the i^{th} layer mapped to an AF priority level x.

In equation (3.12), the unknown parameters are the prices of the service K_x for the x^{th} AF class. Since the price associated with an AF class may vary with the service provider, no specific values can be assigned. One can decide a price range for providing such a priority packet-forwarding service. Typically only a relationship between the prices for each AF priority level can be determined based on their assigned priorities:

$$K_g > K_y > K_r \tag{3.13}$$

However, the above relationship is not helpful in deriving any meaningful values, which are needed for subsequent analysis. Due to the absence of any specific relationship between the prices of AF priority levels, we express both K_g and K_r in terms of K_y as described below.

Similar to the technique used in Section 2.2.4 to implement the three WRED priority queues for AF priority levels, we can inter-relate the WRED packet drop probability parameter (max_p) for each AF priority queue (i.e. green, yellow, and red) as follows:

$$\begin{aligned} \max_p(green) &= \frac{1}{\Delta} \max_p(yellow) \\ \max_p(green) &= \frac{1}{\delta} \max_p(red);\ \Delta > 1,\ \delta > \Delta \end{aligned} \tag{3.14}$$

Equation (3.14) shows that during transmission, a green packet receives Δ times more importance than a yellow packet and δ times more importance than a red packet. Thus, every unit K_g for transmission of green packets is equivalent to Δ unit K_y and δ unit K_r. So the prices K_g and K_r can be expressed in terms of K_y as:

$$\begin{aligned} K_g &= \Delta K_y \\ K_r &= \frac{\Delta}{\delta} K_y \end{aligned} \tag{3.15}$$

Thus, the total cost (c_{AF}) of a packet stream utilizing AF services can be obtained by substituting (3.15) in (3.12):

$$c_{AF} = \Delta K_y \sum_{i=1}^{G} CIR_i^g + K_y \sum_{i=1}^{Y} CIR_i^y + \frac{\Delta}{\delta} K_y \sum_{i=1}^{R} CIR_i^r \tag{3.16}$$

or,

$$c_{AF} = K_y \left\{ \Delta \sum_{i=1}^{G} CIR_i^g + \sum_{i=1}^{Y} CIR_i^y + \frac{\Delta}{\delta} \sum_{i=1}^{R} CIR_i^r \right\} \tag{3.17}$$

For convenience, we use K_y=1 for all calculations. As a result equation (3.17) simplifies to:

$$c_{AF} = \Delta \sum_{i=1}^{G} CIR_i^g + \sum_{i=1}^{Y} CIR_i^y + \frac{\Delta}{\delta} \sum_{i=1}^{R} CIR_i^r \tag{3.18}$$

Hence, by inserting the corresponding values of weighted throughput and total cost in equation (3.7), for a packet stream containing G green, Y yellow and R red packet flows, the PF can be evaluated as follows:

$$P_f = \frac{\sum_{i=1}^{n} (T_i r_i w_i)}{\Delta \sum_{i=1}^{G} CIR_i^g + \sum_{i=1}^{Y} CIR_i^y + \frac{\Delta}{\delta} \sum_{i=1}^{R} CIR_i^r} \tag{3.19}$$

where n is the total number of packet flows (i.e. $n=G+Y+R$) and CIR_i^x is the CIR for the i^{th} layer mapped in AF priority level x.

For given video-layers weighting values w_i and bitrates r_i, one can evaluate the performance factor P_f of the desired mapping strategy if the PLR_i values (for the different video layers) are known for that strategy. In this work, we employed the well-known network simulator (NS), which is the most popular software simulation tool that are used for Internet-based performance evaluation, to estimate the PLR_i values of the different video layers and for the different mapping strategies described above.

4 Simulation Setup

In this chapter we present the setup that is used for the simulation of the proposed mapping categories and their associated pricing models. A high-level description of the simulation scenarios are provided in section 4.1. The second section describes the DiffServ network topology and the configuration of video sources used in the simulations. Bitrate calculations for each packet stream transmitted over the DiffServ network is described in section 4.3. An objective of this thesis was the analysis of the proposed strategies at various levels of network over-commitment, where the available network bandwidth is intentionally over-booked by the service provider. This situation is implemented in the simulations by overloading the network links. The details of the network link overloading are also described in the same section. Section 4.4 describes the parameters used for traffic conditioning of video streams at the DiffServ network's edge router. Before conducting the extensive DiffServ simulations for the different strategies described above, we validated the employed network topology by running two randomly chosen simulation scenarios. These validation results are presented in section 4.6. This chapter concludes with a short summary of the simulation steps that are used in this work.

4.1 Simulation Scenarios

The proposed mapping categories in Chapter 3 are used as the two basic simulation scenarios namely: *Scenario I* for one-to-one mapping category, and *Scenario II* for multiple-to-one mapping category. A description of these simulation scenarios is listed in Table 4.1.

Table 4.1 Statistics of Simulation Scenarios.

Simulation Scenario	*n* (number of video layers)	Mapping Strategy	
		Video Layer	AF Priority Level
I	3	BL	AF_{11}
		EL1	AF_{12}
		EL2	AF_{13}
II	10	BL	AF_{11}
		EL1-EL9	AF_{12}

Simulation scenario II is further divided into three sub-scenarios based on the calculation method of the enhancement layer CIRs. Sub-scenario II(a) is used as a reference model, in which the flat rate based technique is used. The other two sub-scenarios are: scenario II(b) for the exponential pricing model (EPM), and scenario II(c) for the linear pricing model (LPM).

4.2 Network Topology and Statistics

We used Network Simulator (NS) version ns-2.1b8a [14] to simulate the proposed mapping strategies. NS is a popular network simulation tool that is extensively employed for network related research, in general, and Internet-based simulation in particular. Only NS ns-2.1b8a and later versions are equipped with the DiffServ module [54] and with

standard traffic conditioning algorithms (e.g. metering-marking algorithms: srTCM, trTCM and TSWTCM) and traffic shaping (or queuing) mechanisms (e.g. FIFO, RED and WRED etc.). Except for a few modifications made in the back-end code of NS, that were specific to our requirements, we mostly utilized the standard built-in sub-routines and algorithms for our simulations. Thus, a configuration of the built-in sub-routines by calculated parameter values are used for most of the simulations.

The network topology used for the simulations is shown in Figure 4-1. All network links in the figure are T1 links, which can support transmission of data up to 1.5 Mbps. A DiffServ domain is established by using one ingress edge router, one core router, and one egress edge router. Five video sources, attached with the ingress router, are configured to send video packet streams toward their respective destinations (or clients), which are attached to the egress edge router. Each video source is configured to send equal size (i.e. 512 bytes) UDP packets at some *constant bitrate* (CBR). A separate packet stream is generated for each video layer by using a separate UDP agent in the source. In this thesis we use CBR packet streams as a limiting case, due to the prior knowledge of available bandwidth. However, the scalable video coding methods like MPEG-4 FGS are capable to send packet streams (or video layers) at any desirable bitrate, depending upon the availability of network bandwidth.

Each packet stream from a video source contains two types of packets: the base layer (BL) packets and the enhancement layer (EL) packets. The method employed for the calculation of bitrates for each video stream and their subsequent video layers is described in the next section.

Table 4.2 Simulation Statistics.

Simulation Parameter	Value
Simulation Time	60 sec
Links between Network nodes	1.5 Mbps, 5ms delay
Packet Size	512 bytes
Maximum number of ELs	9
Number of Video sources	5
Buffer queue length	70 packets

Before transmission over the network link, every video source is configured to mark each packet with: i) a flow-id (to identify the video layer), and ii) a corresponding DSCP of an AF priority level (i.e. AF_{x1} (green), AF_{x2} (yellow) or AF_{x3} (red)) depending on the applied mapping strategy.

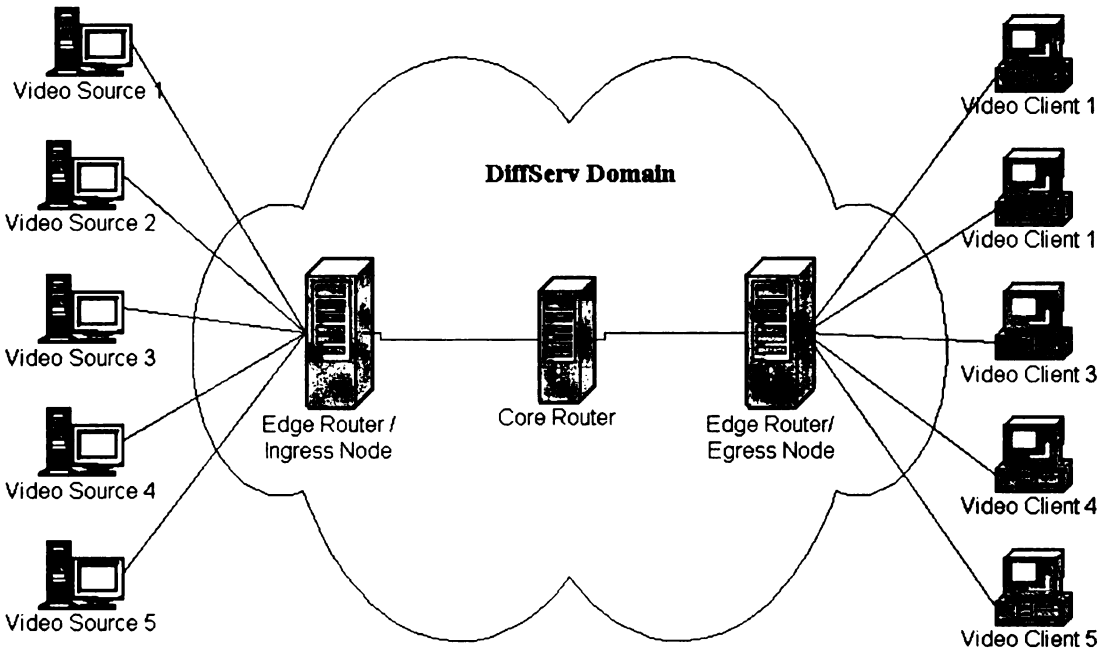

Figure 4-1 Network topology for the simulations.

4.3 Bitrate and Network Overload Configurations

The network topology discussed in the previous section has been simulated under two network conditions: (1) when the bandwidth of the network links is only committed to their full capacity i.e. no overload traffic, and (2) when the network link bandwidth is over-committed (ranging from 10% to 100% overload traffic). These two network conditions are referred to as *normal* and *overload* conditions.

Each over-commitment scenario has been simulated by sending overload video traffic towards the DiffServ domain. For this purpose, the bitrate of each video stream (and its subsequent video layers) is calculated and applied to the network in order to create a given level of network overload. The same bitrates are then used in the committed information rate (CIR) computations using the corresponding pricing model (i.e., flat rate, EPM or LPM) according to the mapping category in use.

In the first phase of experiments, the previously described DiffServ domain is simulated under normal condition, without any overload, to obtain the network PLR response for the proposed mapping strategies. In the second phase of experiments, the network is simulated under three distinct overload levels of 10%, 50% and 100% respectively. In order to estimate the individual bitrates of each video layer at every network overload level, the maximum bitrate (R_{max}) of the video stream sent by each video source is calculated first as a function of the ingress edge router's outbound link capacity C as follows:

$$R_{\max} = \frac{C}{s}(1+v) \qquad (4.1)$$

where s denotes the number of video sources attached to the edge router, and v is the network overload as a fraction that ranges between zero and one.

For instance, in order to overload the edge router by 10%, we used $v = 0.1$ in equation (4.1). Hence, for a given link capacity of 1.5 Mbps and five video sources (i.e. $s = 5$), the bitrate R_{max} is calculated to be 330 kbps for each video source. Similarly, at no overload the available link capacity is equally committed (or allocated) to each video source with $R_{max} = 300$ kbps.

As described in the previous section, each video source is configured to send two types of packet streams, i.e. base layer and enhancement layer packet streams at bitrates r_b and r_e respectively. However, depending on the number of enhancement layers, the number of packet streams from each video source may vary from 1 to n. Thus, the maximum bitrate (R_{max}) of a video source equals the sum of the individual packet streams' bitrates that are generated from the video source.

Under normal conditions, the available bandwidth of edge router's outbound link is distributed equally to each video source. Hence, the default value for R_{max} at no-overload network condition is 300 kbps for $C = 1.5$ Mbps. The bitrate of the base layer (r_b) is kept constant at 64 kbps in each simulation scenario at every network overload level, while the remaining allocated bandwidth is used for the enhancement layer(s). By using a constant bitrate for the base layer, the overloading of network is achieved by a variation of the enhancement layer(s) bitrate(s). The distribution of bitrates r_b, and r_e in R_{max} is listed in Table 4.3 for each network overload condition.

The bitrate r_e in Table 4.3 represents the total bitrate of the enhancement layer packet stream(s). During simulations when more than one (say n) enhancement layers are used, the bitrate r_i of any i^{th} enhancement layer is kept fixed at:

$$r_i = \frac{r_e}{n}, \quad where\ 1 < i \leq n \tag{4.2}$$

Therefore, the rate distribution of the enhancement layers is according to a simple uniform allocation method (as discussed in section 2.1.1).

Table 4.3 Distribution of bitrates (kbps) for video layers.

Network Overload	**0%**	**10%**	**50%**	**100%**
R_{max}	300	330	450	600
r_b	64	64	64	64
r_e	236	266	386	536

As mentioned earlier in section 3.2, we used the flat rate technique for calculating the CIR when a single layer is mapped to a single priority level. Since the base layer is always mapped to the highest priority level as green packet flow in each scenario, the base layer CIR (CIR_b) is always calculated using the flat rate pricing equation (2.18) with p = 1.01 (this value of p conforms to the general threshold condition for green packet flows as indicated in equation (2.15)) as follows:

$$CIR_b = 1.01\ r_b \tag{4.3}$$

Similarly in scenario I, the two enhancement layer(s) are also mapped to the two distinct priority levels as yellow and red packet flows, and thus the corresponding CIRs are calculated using the flat rate pricing equation (2.18) with $p = 0.99$ as follows:

$$CIR_e = 0.99\ r_e \tag{4.4}$$

where r_e is the rate of the corresponding enhancement layer.

The equations (4.3) and (4.4) satisfy the necessary conditions mentioned in equations (2.15) to (2.17) for green, yellow and red packet flows, respectively. Equation (4.4) is also used for calculating the CIR of each enhancement layer in simulation scenario II (a).

4.4 Traffic Conditioning Setup

As described in sections 2.2.2 to 2.2.4, the traffic conditioning is a set of sequel processes of metering, marking, and shaping of the incoming packet streams at the DiffServ boundary node (i.e. ingress edge router). The selected simulation parameters and their statistics for these traffic-conditioning processes are described in the subsequent sections.

4.4.1 Traffic Metering and Marking

A metering-marking algorithm at the edge router is used to "meter" every incoming packet with a given set of service level agreement (SLA) profile values. DiffServ module in NS supports both UDP/CBR compatible standard metering-marking algorithms: srTCM and trTCM [54]. Due to the similarity of the parameters in both algorithms [see Appendix], similar results were expected. This similar expected behavior is also due the fact that the proposed mapping strategies are simulated by varying only the CIR parameter values, while keeping the other parameters constant. Thus, two sets of simulations were performed. Results showed that the two sets of PLR values that were generated from using both markers were similar. Therefore, results generated using only trTCM are included in this thesis.

As described in Chapter 3, the bitrate r_i of each incoming colored packet i is metered using a given set of bitrate threshold values, CIR and PIR:

$$\text{Green packets} \quad \begin{matrix} r_i \le CIR_i \\ r_i < PIR_i \end{matrix} \tag{4.5}$$

$$\text{Yellow packets} \quad CIR_i < r_i \le PIR_i \tag{4.6}$$

$$\text{Red packets} \quad \begin{matrix} r_i > CIR_i \\ r_i > PIR_i \end{matrix} \tag{4.7}$$

Recall from chapter 2 that as a result of metering, an incoming colored packet (i.e. marked initially with a DSCP of an AF priority level) is categorized either as an *in-profile* or an *out-of-profile* packet. If the rate of an incoming packet is found conformal to the given threshold limits of CIR and PIR, it is categorized as an in-profile packet. Such packets are forwarded directly to their respective buffer queues. Whereas, if an incoming packet is found exceeding the given rate threshold, it will be categorized as an out-of-profile packet. Each out-of-profile packet is eligible to be re-marked with the DSCP of a lower AF priority level before being forwarded towards the buffer queues. In other words, the drop probability of an out-of-profile packet increases due to re-marking. A summary of metering-marking algorithm parameters selected for the simulations are listed in Table 4.4.

Table 4.4 Selected values for trTCM parameters.

trTCM parameter	**Comments/ Value**
CIR	Evaluated by using pricing models (flat, exponential or linear)
CBS	2 packets (1024 bytes)
PIR	Green and Yellow packets: 1.5 Mbps Red packets: 0.99 x (bitrate of red packet flow)
PBS	2 packets (1024 bytes)

The CBS and PBS parameters used in trTCM and srTCM algorithms are defined to be the size of token bucket at the ingress router. The use of these token buckets is described in Appendix. For the T1 links used in our simulations, we choose a suitable token bucket size of 2 packets i.e. 1024 bytes. These parameter values are also kept constant throughout the simulations.

4.4.2 Traffic Shaping

As introduced in Chapter 2, we preferred WRED as traffic shaping mechanism. The reason for selecting WRED is the provision of establishing multiple RED queues and control over the packet drop probability in each queue, which allows an assignment of priorities to each buffer queue with respect to the given AF priority level. The employed version of NS for our simulations also supports the WRED queuing mechanism.

According to this mechanism, one physical RED queue is divided into three virtual RED queues, each dedicated to the packets belonging to a given AF priority level; green, yellow, or red. Length of the physical RED queue, for the given T1 links, is kept constant at 70 packets throughout the simulations, while the queue length thresholds (i.e. min_{th} and max_{th}) for each virtual RED queue is selected according to the "rule of thumb" suggested by *Jacobson et al.* in [15]. In order to provide service differentiation and priority, a maximum length of 40 packets is selected for green packet queue, a maximum length of 20 packets for yellow packet queue and a maximum length of 10 packets for red packet

queue[1]. Similarly a different packet drop probability (max_p) is allocated to each virtual queue with respect to its priority level. Thus, by using the relationship (2.8) and its modification in equation (3.14) for AF priority queues:

$$\begin{aligned} \max_p(green) &= \frac{1}{\Delta}\max_p(yellow) \\ \max_p(green) &== \frac{1}{\delta}\max_p(red);\ \Delta \geq 1,\ \delta \geq \Delta \end{aligned} \tag{4.8}$$

For our simulations, we selected $\Delta = 5$, and $\delta = 10$. By selecting the highest value of $max_p = 0.2$ for red packets, we obtained a medium max_p (= 0.1) for yellow packets, and the lowest max_p (= 0.02) for the green packets. The selected WRED parameter values for all simulations are listed in Table 4.5, and are shown in Figure 4-2. However, one can select other parameter values within the specified range, which may ensue different values for PLR. However, the behavioral trend (in terms of PLR variations among the different video layers) will remain the same if the queue priorities will assign in the same proportion.

Table 4.5 Weighted-RED parameters for the simulations.

Queue	min_{th}	max_{th}	max_p
Green	20	40	0.02
Yellow	10	20	0.1
Red	5	10	0.2

[1] This technique is adopted from Cisco IOS™ release 12.2 specifications, which implement service differentiation in RED queues by varying the lower threshold (min_{th}) while keeping the upper threshold (max_{th}) constant.

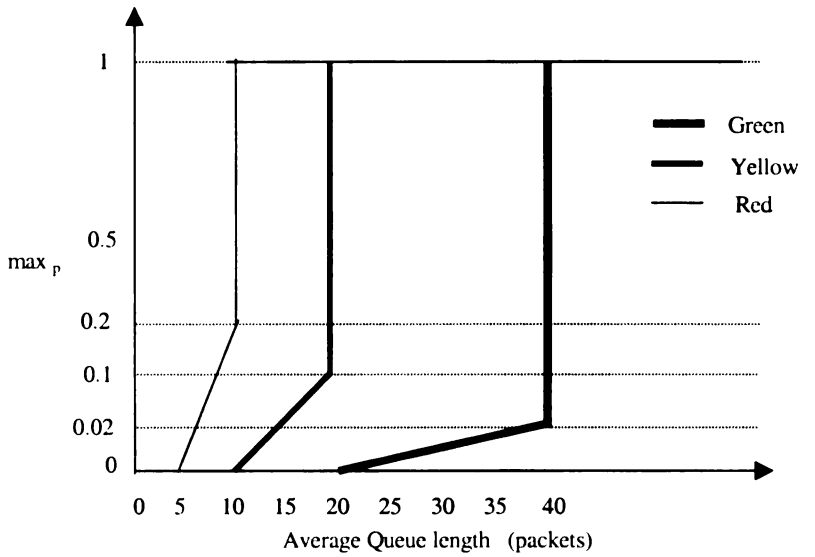

Figure 4-2 WRED parameters for priority queues.

4.5 Weight Distribution for Video Layers

Due to the absence of any typical weight distribution curve definition for video layers in scalable video coding standards [24][25][26], we used an exponential weight distribution curve (see Section 2.1.2) for the performance factor (PF) evaluations. The weight distribution curve (shown in Figure 4-3) is obtained by using $a = 0.4$, $b = 0.7$ in equation (2.5). However, one can use other combinations for the parameters a and b in equation (2.5) to obtain different weight distribution curves for the video layers in a scalable video stream. The selection of $a = 0.4$ denotes that the first enhancement layer (EL1) is responsible for bringing 40% improvement in the video quality produced by the base layer. The curve in Figure 4-3 represents the weight distribution for a video stream containing one base layer and nine enhancement layers i.e. $n = 10$.

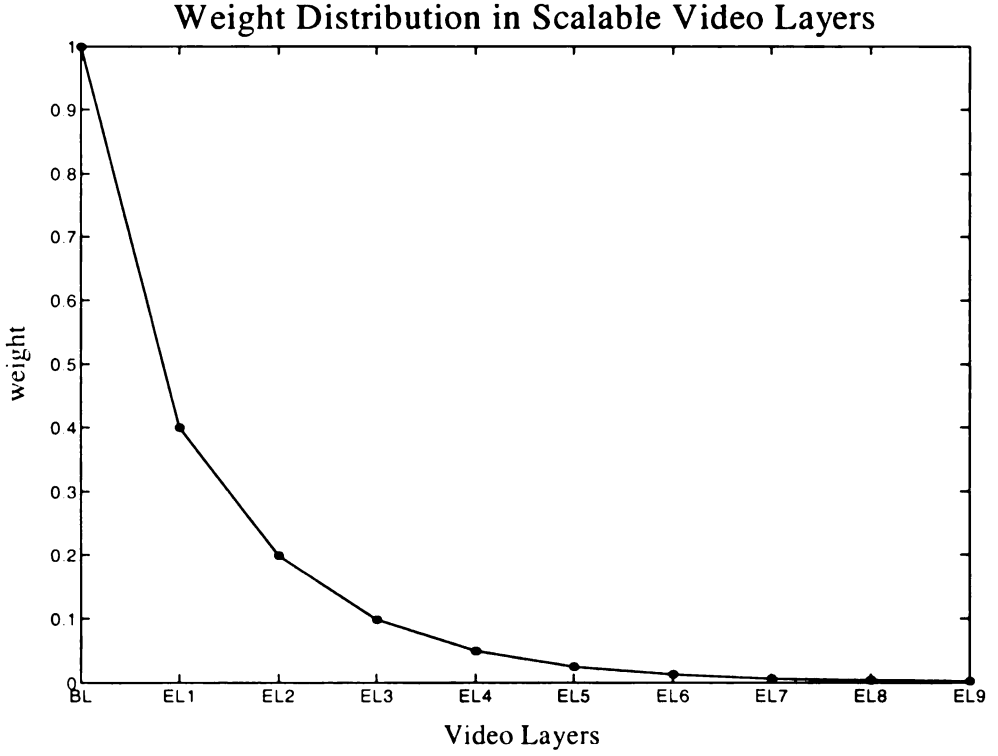

Figure 4-3 Employed weight distribution of scalable video layers.

4.6 Network Fairness Criteria

As discussed in section 4.3, the available bandwidth (and also the over committed bandwidth) is equally distributed to each attached video source. Thus, as a pre-simulation test of the adopted network topology, an experiment was performed to ascertain the *fair* distribution of network resources to all video sources. By a fair distribution we mean that the allocated network resources (like bandwidth and buffer) are equally shared by each video source. With a fair distribution of network resources, each video source is expected to suffer similar PLR. For this purpose, we simulated the network for two randomly chosen scenarios, at different network overload levels. The results in Figure 4-4 show

that the video source location does not significantly affect the obtained PLR response. Hence we will use the data obtained from only one source for all subsequent analyses.

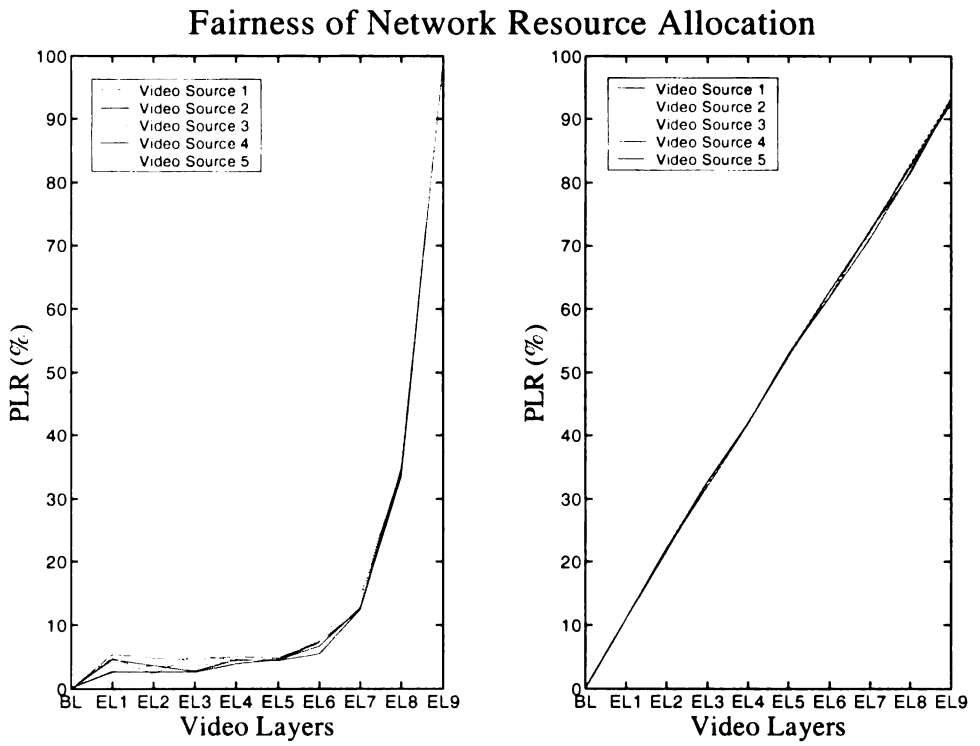

Figure 4-4 PLR of video sources at (a) scenario II(b) at 10% overload, and (b) scenario II(c) at 50% overload.

4.7 Summary of Simulation Steps

Each scenario has been simulated using the following sequence of steps.

STEP 1: Calculation of R_{max} for each video source and for every network overload level.

STEP 2: Calculation of the base layer and the enhancement layer(s) bitrates r_b and r_e, for each network overload level.

STEP 3: CIR calculation for the corresponding base layer and enhancement layer bitrates, for the simulated pricing model at various network overload levels.

STEP 4: NS code configuration: Assignment of a flow-id for each video layer.

STEP 5: NS code configuration: Assignment of a corresponding DSCP value for each video layer, in accordance with the simulation scenario.

STEP 6: Running NS simulation for exactly 60 seconds, by application of the computed bitrates and CIRs for each scenario.

STEP 7: Calculation of PLR for each video layer.

STEP 8: PF evaluation using the network PLRs determined in step 7.

5 Simulation Results and Analysis

This chapter describes the results obtained from the simulations of the proposed mapping categories. The results of each simulated category are presented in the first two sections. In each section, the network packet loss ratio (PLR) response for the simulated video stream is presented first, and the evaluated performance factor (PF) as a measure of the performance of the mapping strategy is presented next.

Section 5.1 contains the simulation results for scenario I (i.e. one-to-one mapping). The next section (i.e.,5.2) contains the simulation results of scenario II (i.e. multiple-to-one mapping). Further three subsections (i.e. 5.2.1 to 5.2.3) illustrate the results obtain by applying the proposed three pricing models (i.e., flat, exponential, and linear). A comparison of the PLR results obtained from using the different pricing models of the multiple-to-one strategy is presented in section 5.3. The last section concludes the simulation results with an analysis of the evaluated performance and cost for each simulated scenario.

5.1 Simulation Scenario I: One-to-One Mapping

The PLR response, obtained by the simulations of scenario I at various network overload condition ranging from 0% to 100% overload, is shown in Figure 5-1. Under normal conditions, i.e. at 0% overload, the PLR response in Figure 5-1 shows similar

packet forwarding behaviors for three distinctively mapped packet flows i.e. one base layer (BL) and two enhancement layers (EL1 and EL2). All BL packets received the highest forwarding priority and suffer no losses during transmission as expected. Similar PLR behavior is also observed for EL1. Being the lowest prioritized packet flow, the maximum packet losses occurred in EL2. The results for this network condition show a fulfillment of the desired PLR constraints as depicted in equation (3.1).

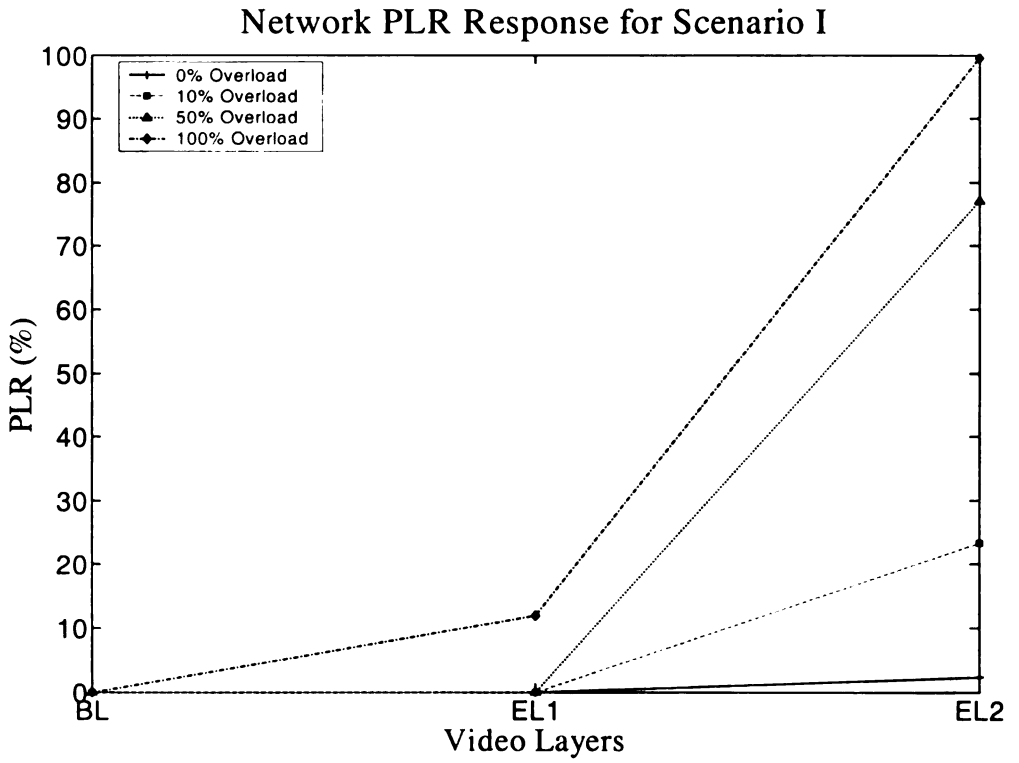

Figure 5-1 PLR for Simulation Scenario I.

As shown in Figure 5-1, an increase in overload does not affect the transmission of the base layer. This behavior indicates that during severe network overload conditions, BL packets will retain the highest priority of being transmitted without any losses. For EL1, the PLR increased significantly (i.e. upto 12%) at 100% overload. Notice that no packet losses appeared in EL1 up to 50% overload. As expected, the maximum packet

losses occurred during the transmission of EL2, which was mapped as the lowest priority red packet flow. The PLR of EL2 increased from 2.3% to 99.5% with a corresponding increase in the network overload from 0% to 100%. Hence, equation (3.1) is satisfied at every network overload level.

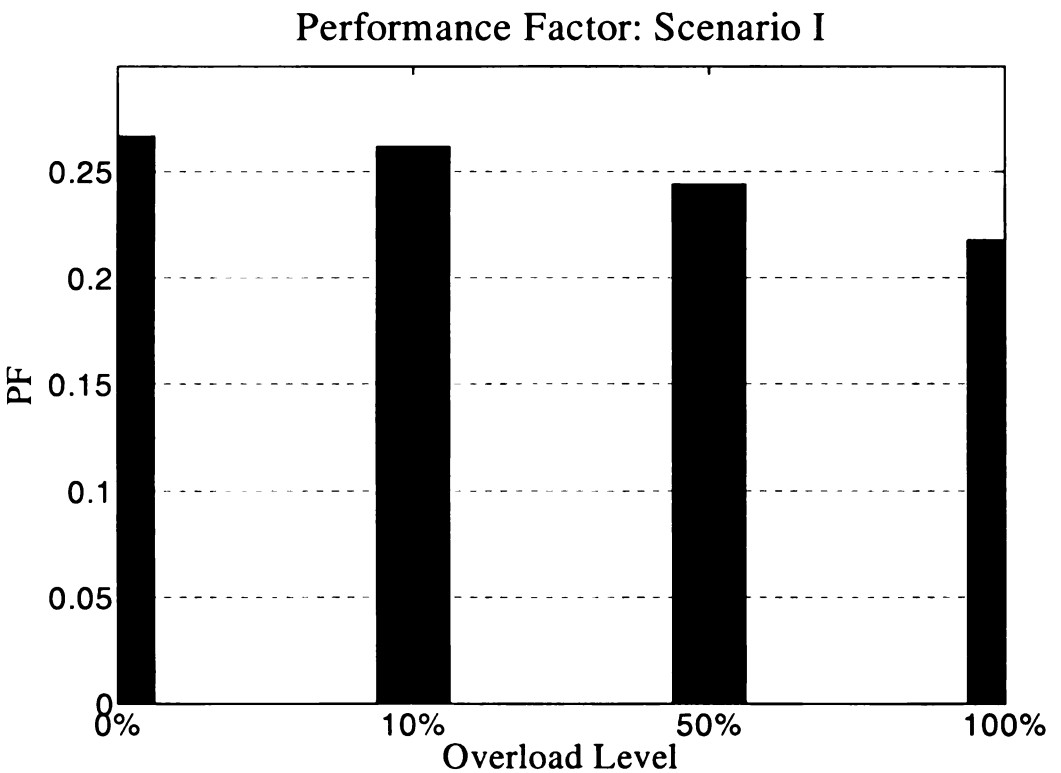

Figure 5-2 Evaluated PF for Simulation Scenario I.

The PF calculated using the above PLR results for this scenario is shown in Figure 5-2. An expected decay in the PF value is observed during the transition of network overload from 0% to 50%, and from 50% to 100% overload. However, the PF value degraded more during the second transition due to an increased number of packet losses in EL1, which is the most important enhancement layer due to its maximum weight.

5.2 Simulation Scenario II: Multiple-to-One Mapping

5.2.1 Simulation Scenario II (a): Multiple-to-One Mapping using Flat Rate Pricing Model

The PLR response obtained by simulating the multiple-to-one mapping using the flat rate pricing is shown in Figure 5-3. The base layer (BL), being the highest prioritized packet flow, suffered no packet losses at any network overload state. Under normal network conditions, all equally prioritized enhancement layers (EL1-EL9) suffered some packet losses. Since, in this case, the network commitments were based on the flat rate pricing model, the price for each enhancement layer is paid to transmit 99% of the delivered packets, and thus, not more than 1% packet losses were expected during the transmission of each enhancement layer. The higher-than expected PLR values shown in the figure, indicates *queue overflow* at the ingress router. Since every packet belonging to any of the enhancement layers (from every video source) is buffered in the same queue (i.e., same yellow packet queue due to the flat pricing strategy), and because (virtually) all of these packets remained "in-profile", then these packets created random *queue congestion*. Therefore, the queuing mechanism (WRED) dropped a large number of packets randomly, according to the specified drop probability parameter max_p for the queue. Consequently, every enhancement layer suffered more than 1% packet losses. However, a closer to flat response is visible for the enhancement layers only at 0% overload, which does not comply with the required transmission quality as given by $PLR_i < PLR_{i+1}$.

For an increasing overload from 10% and above, an undesirable and random PLR behavior is obtained as shown in Figure 5-3. Neither the obtained PLR for all equally mapped (and equally paid) enhancement layers in the video stream met the required relationship i.e. $PLR_i < PLR_{i+1}$, nor a flat PLR response is obtained at these overload levels during the simulation of this scenario.

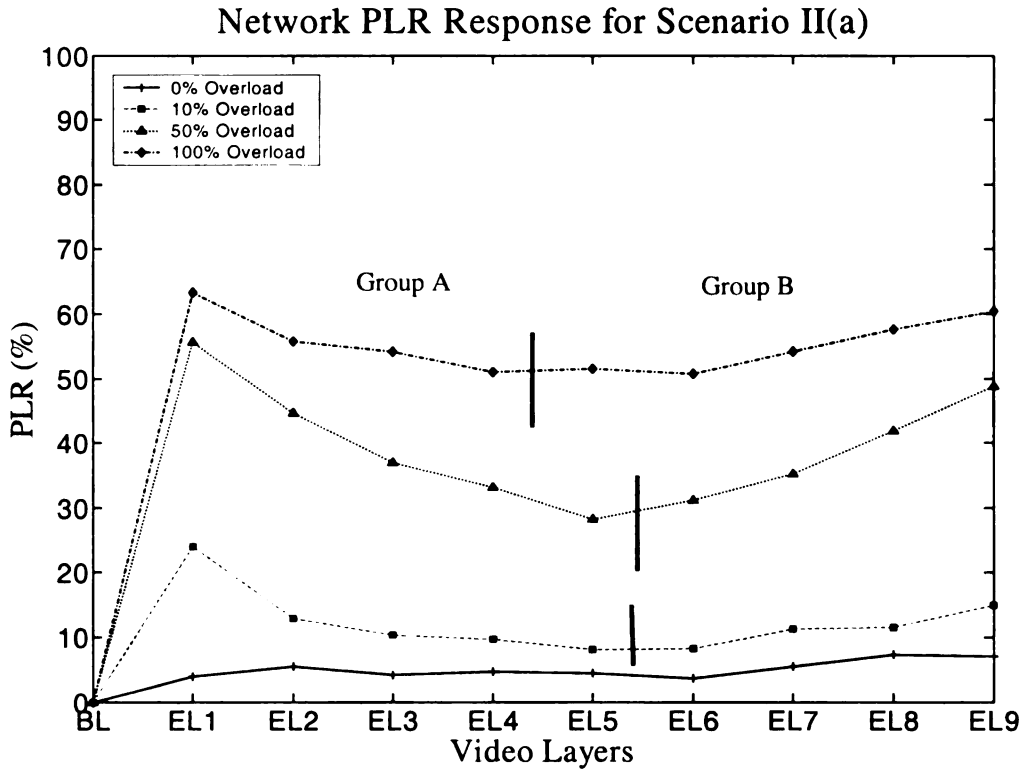

Figure 5-3 PLR for Simulation Scenario II (a).

Under overload conditions, the PLR behavior of enhancement layers shows an anomalous behavior, which divides the enhancement layers into two groups. First group, say 'group A' contains lower enhancement layers starting from EL1, and the second group of layers, say 'group B' contains the upper enhancement layers that starts from the

next layer above 'group A' up to the last enhancement layer EL9. The vertical lines in Figure 5-3 bifurcate the curves into these groups. The PLR of enhancement layers in 'group A' are found to be monotonically decreasing with a maximum PLR in EL1. This behavior shows that PLR of enhancement layers in this group did not fulfill the requirement for quality transmission of video as given by equation (3.1), i.e. $PLR_i < PLR_{(i+1)}$, instead it shows a contradictory behavior i.e. $PLR_i \geq PLR_{(i+1)}$. According to Figure 5-3, 'group A' includes enhancement layers EL1 to EL5 at 10% and 50% overloads, and EL1 to EL4 at 100% overload. On the other hand, the PLR values for the enhancement layers in the second group, i.e. 'group B' satisfy the PLR constraints $PLR_i < PLR_{(i+1)}$.

The evaluated PF values for this scenario at four network overload levels are shown in Figure 5-4. In this scenario, the video streams contained nine enhancement layers, instead of two enhancement layers as in scenario I. However, the total bitrate of the enhancement layers is the same in both scenarios, but the ratio of individual enhancement layer bitrates in scenario I and II is 4.5:1 (i.e., 9:2). A similar ratio (i.e. 4.5:1) is true for the number of packets belonging to each enhancement layer. Thus, the reduced number of packets sent for EL1 and EL2 (with same weights) in this scenario (i.e., scenario IIa), affects the evaluated PF less significantly (i.e., reduced by almost half as compared to the evaluated PF values in scenario I). This explanation is true for all simulation results of scenario II since the same number of enhancement layers and same weight distribution is used in subsequent two scenarios.

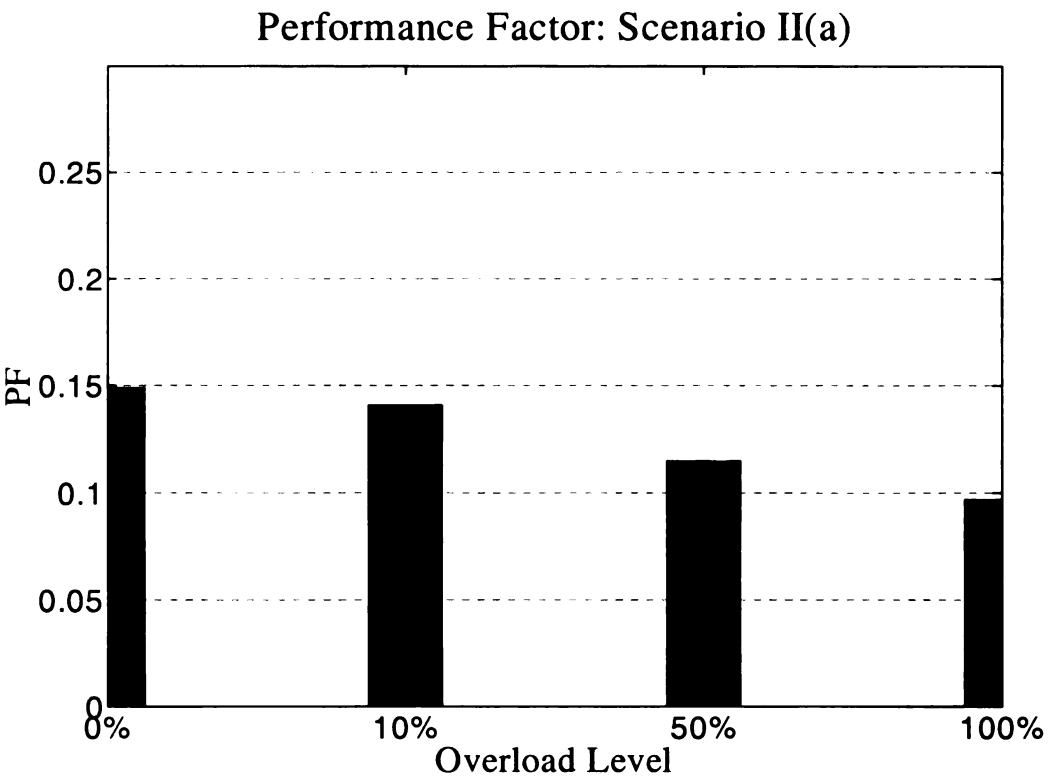

Figure 5-4 Evaluated PF for Simulation Scenario II (a).

Except at 0% overload, the maximum number of packet losses occurred in EL1 during transmission at every network overload state. With an increase in network overload, the PLR for EL1 increased from 3.9% at 0% overload up to 63.3% at 100% overload. Similarly, the PLR of EL2 also increased from 5.4% to 55.7% as network was overloaded from 0% up to 100%. The effect of such high PLR in the two highest priority enhancement layers is clearly visible in Figure 5-4, where the PF value degraded from 0.15 at 0% overload to 0.09 at 100% overload.

5.2.2 Simulation Scenario II (b): Multiple-to-One Mapping using Exponential Pricing Model

The EPM was proposed to achieve an exponentially increasing PLR behavior for the nine equally mapped enhancement layers, and consequently meet the PLR constraints: $PLR_i < PLR_{(i+1)}$. The PLR response obtained by the simulations of this scenario is shown in Figure 5-5. The exponentially increasing PLR curve for all video layers is obtained at 0% network overload, which shows an improved quality of delivered video as compared to the previous scenario II (a). The base layer received highest forwarding priority and suffered no losses during transmission at any network overload level.

Similar to the PLR behavior for the previous scenario II (a) at network overload conditions, the enhancement layers can be categorized into two groups according to their PLR response in this scenario. 'Group A' contains the enhancement layers which shows a monotonically decreasing PLR behavior starting from a maximum value in EL1. This group includes enhancement layers EL1 to EL3 at 10% overload, EL1 to EL5 at 50% overload, and EL1 to EL6 at 100% overload. The desired relationship in PLR of the video layers, i.e. $PLR_i < PLR_{(i+1)}$, is observed in 'group B' during every network overload condition.

At 0% and 10% overload conditions, i.e. minimal overload levels, maximum packet losses are seem to be pushed towards the upper most layer(s). However, this model is found to be unsuccessful in maintaining the over all exponentially increasing PLR during higher network overload states i.e. 10%, 50% and 100% overload.

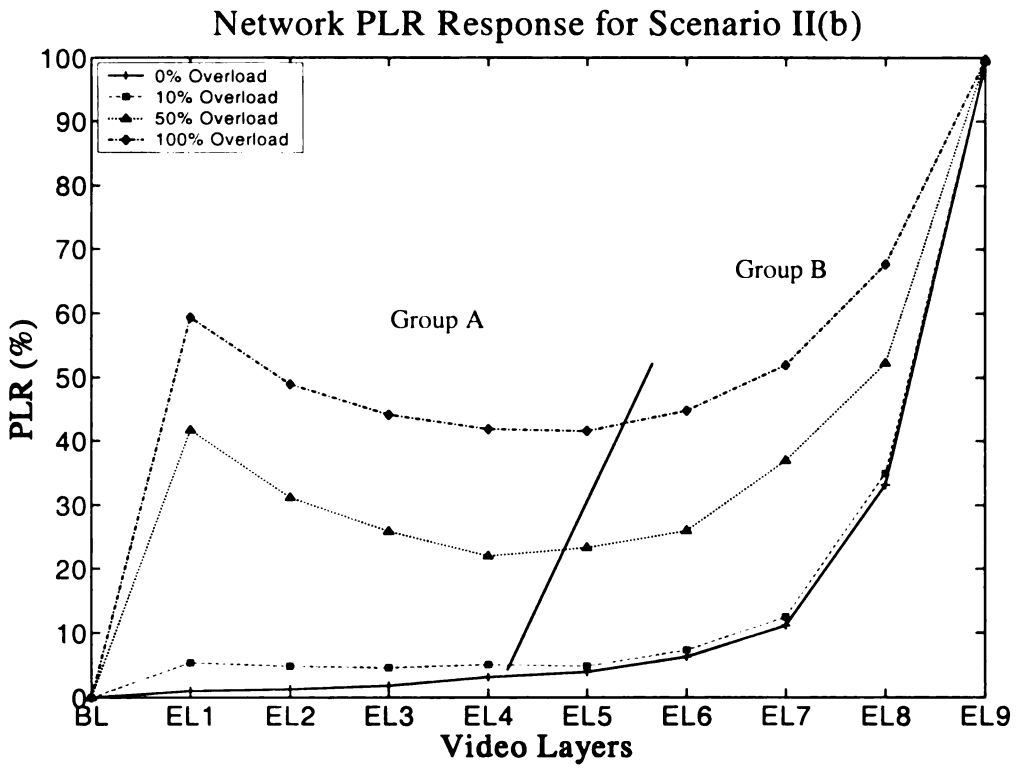

Figure 5-5 PLR for Simulation Scenario II (b).

The PF evaluated by using the PLR results of this scenario is shown in Figure 5-6. The PF value decreased nominally, with increasing network overload from 0% to 10%. With further increase in network overload, the PF value decreased from 0.16 (at 0% overload) to 0.09 (at 100% overload). By using the EPM, the maximum PLR is observed in EL9, at every network overload level. However, with an increase in network overload, the PLR in EL1 also increased from 1.04% at 0% overload up to 59.3% at 100% overload. Also, the PLR in EL2 increased from 1.3% to 48.8% with the corresponding increase in network overload from 0% to 100%. Consequently, the expected qualitative performance behavior is observed only up to a nominal overload of 10%, but the performance degraded significantly at higher network overloads.

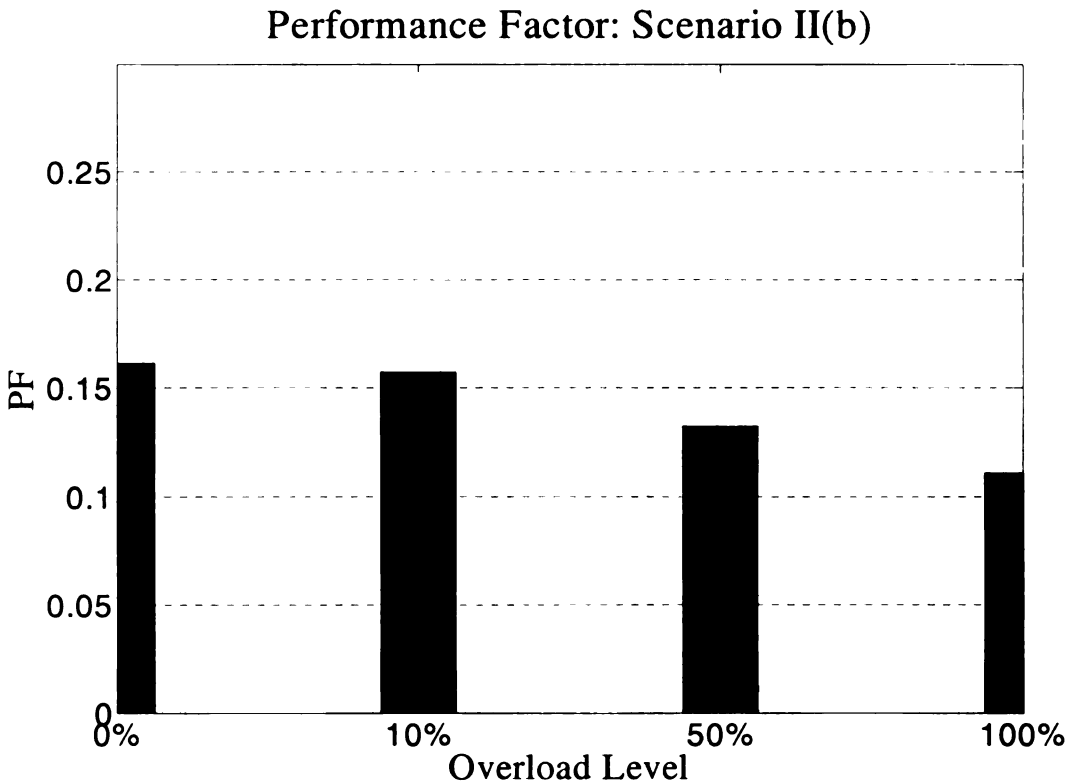

Figure 5-6 Evaluated PF for Simulation Scenario II (b).

5.2.3 Simulation Scenario II (c): Multiple-to-One Mapping using Linear Pricing Model

The network PLR response obtained by using the LPM for multiple-to-one mapping strategy is shown in Figure 5-7. With an increase in the network overload, the linearity manifested itself in the PLR of the enhancement layers. The PLR response became linear, as the network was overloaded up to 100%. As shown in Figure 5-7, the base layer does not suffer any packet losses during its transmission at any network overload condition. Whereas, the desired PLR constraints as given by equation (3.1), i.e. $PLR_i < PLR_{i+1}$, are satisfied by all video layers at every network overload level, ranging from 0% to 100%.

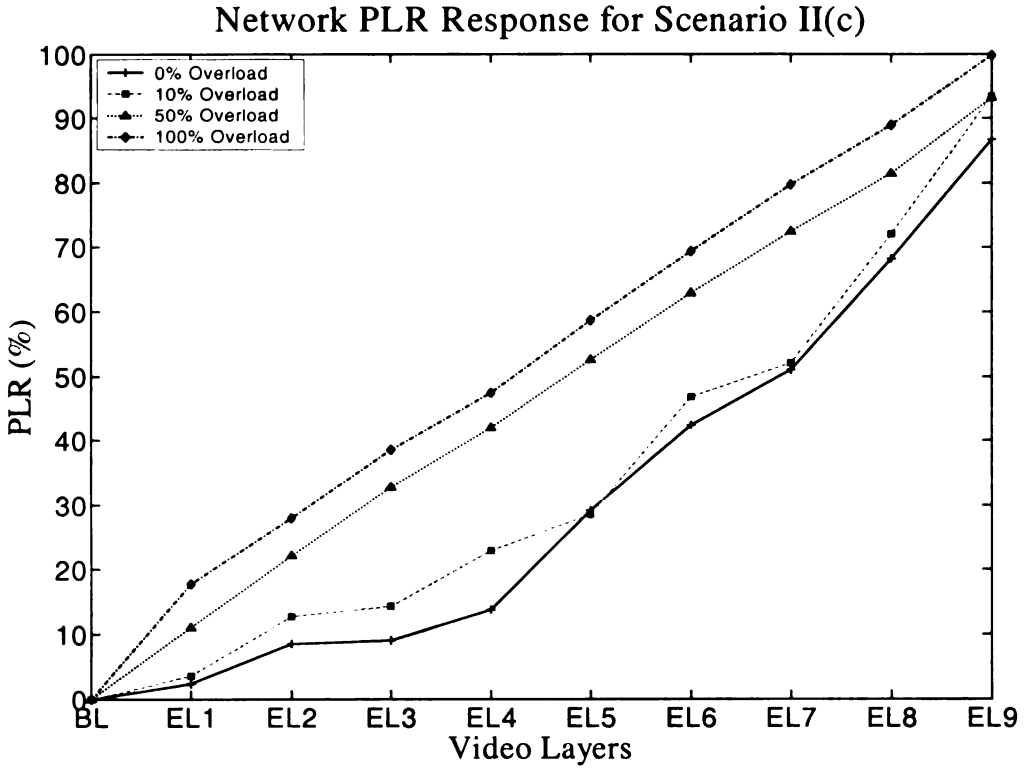

Figure 5-7 PLR for Simulation Scenario II (c).

As shown in Figure 5-7, higher PLR values appeared in the enhancement layers EL2 to EL8 even during nominal network overloads of 0% and 10%. Such high PLR values were not observed in these layers when applying the flat rate and the EPM. Meanwhile, EL1 did not suffer any significant packet losses as compared to the corresponding packet losses for the prior discussed pricing models during 10%, 50% and 100% overloads conditions.

For the resulting PLR, the corresponding PF in Figure 5-8 shows a similar behavior to the other scenarios i.e. a corresponding decrease in PF with increasing network overload. However, the PF value did not significantly degrade during the transition of the network overload from 0% to 100% overload. Best performance in this scenario is observed at 0% overload, which degraded slightly from 0.19 down to 0.17 at 100% overload. This shows

that the performance and quality of transmitted video can be better sustained by applying network bandwidth commitments based on the LPM, as compared to the other pricing techniques used in this thesis.

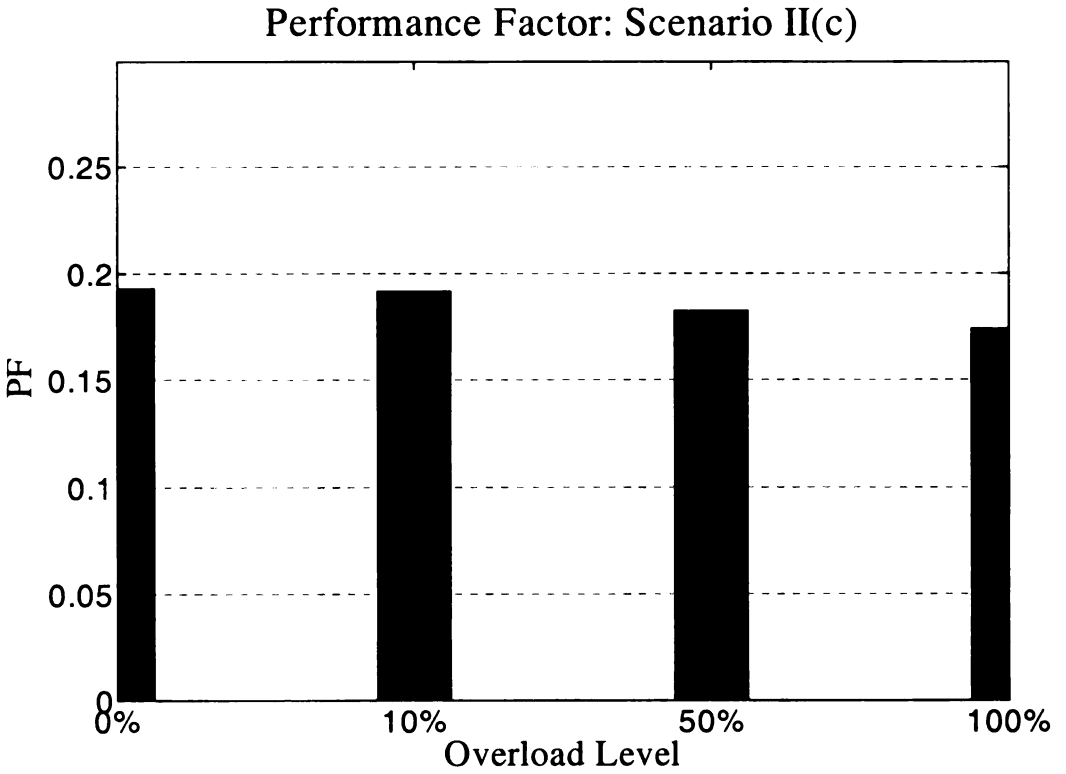

Figure 5-8 Evaluated PF for Simulation Scenario II(c).

The results in Figure 5-7 and Figure 5-8 were obtained by using the maximum possible value of linear pricing factor i.e., $\beta = \frac{r}{n-1}$, in equation (3.4). The PLR results show a desired PLR behavior for all video layers in the stream as depicted by the equation (3.1). Additionally a sustained performance is observed during all the simulated network overload conditions. Moreover, we analyze the network behavior and performance using two different fractional values of β, i.e. 0.1 β and 0.5 β, for the same simulation setup, to explore any possibility for a further reduced cost when using the LPM. The results obtained by the simulations are shown in Figure 5-9.

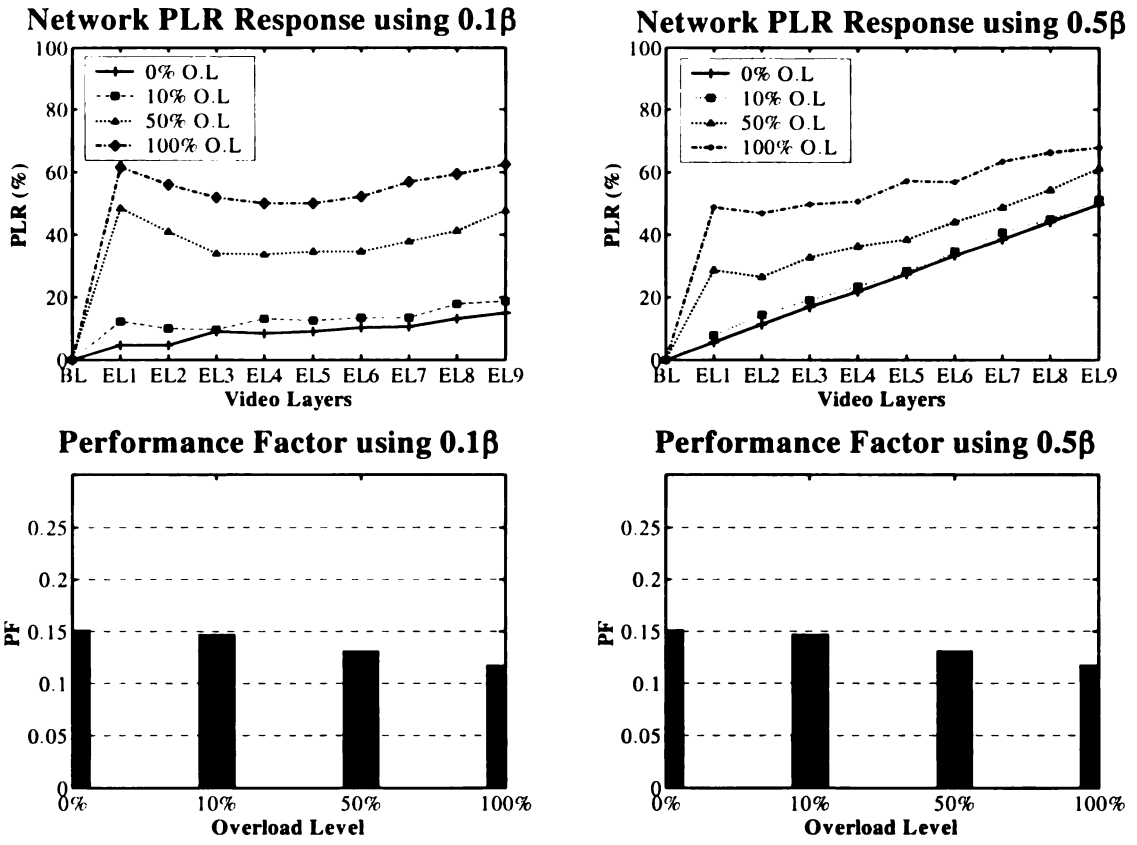

Figure 5-9 PLR and evaluated PF using two different fractional values of β for Scenario II(c).

As shown in Figure 5-9, while using 0.1 β, relatively lower PLR is obtained for enhancement layers at 0% and 10% overload as compared to the previous case. As we increase the network overload to 50% and 100%, the undesirable PLR curve characteristics set in, similar to the prior cases of the flat rate and EPM. Increase in the number of packet losses in the lower enhancement layers (EL1 to EL3) also affects the evaluated PF value, which degraded from 0.15 at 0% overload to 0.11 at 100% overload.

In the second case, using 0.5 β, we obtained approximately a linearly increasing PLR for video layers at nominal network overloads of 0% and 10%. During higher network overloads of 50% and 100%, again EL1 and EL2 suffered an undesirably large number of

packet losses. As a result, the evaluated PF degraded from 0.16 at 0% overload down to 0.13 at 100% overload. The PF values obtained under normal network condition, i.e. at 0% overload, in both cases (i.e. 0.15 by using 0.1 β and 0.16 by using 0.5 β) are also significantly smaller than the PF value (0.19) for the same network overload condition using maximum possible β.

5.3 Comparison of the Pricing Models

The criterion for the estimation of the delivered video quality was based on the relation $PLR_i < PLR_{i+1}$. Under normal conditions, i.e. at no network overload, only exponential and linear pricing models achieved this criterion. On the other hand, flat rate model showed higher PLR values for lower (important) enhancement layers during similar network overload condition. As the network was heavily overloaded to 50% or 100%, lower enhancement layers suffered from a high PLR. The PLR curve of three pricing models (i.e. flat rate, EPM and LPM) at two extreme network overload conditions of 0% and 100% are shown in Figure 5-10 and Figure 5-11.

As shown in Figure 5-10, the PLR response for EPM in the upper layers increased exponentially with a very low change in PLR values for the lower enhancement layers. This behavior depicts an ideal transmission of scalable video streams over lossy packet networks such as the Internet, where all possible packet losses are pushed towards the layer(s) with minimum weight. Whereas, a comparison of the PLR in the most important video layers, in the video stream, is shown by a circle in Figure 5-10. On the other hand, in case of LPM, the PLR started increasing from EL2, whereas a medium PLR value is appeared in EL1 as compared to the other two pricing models.

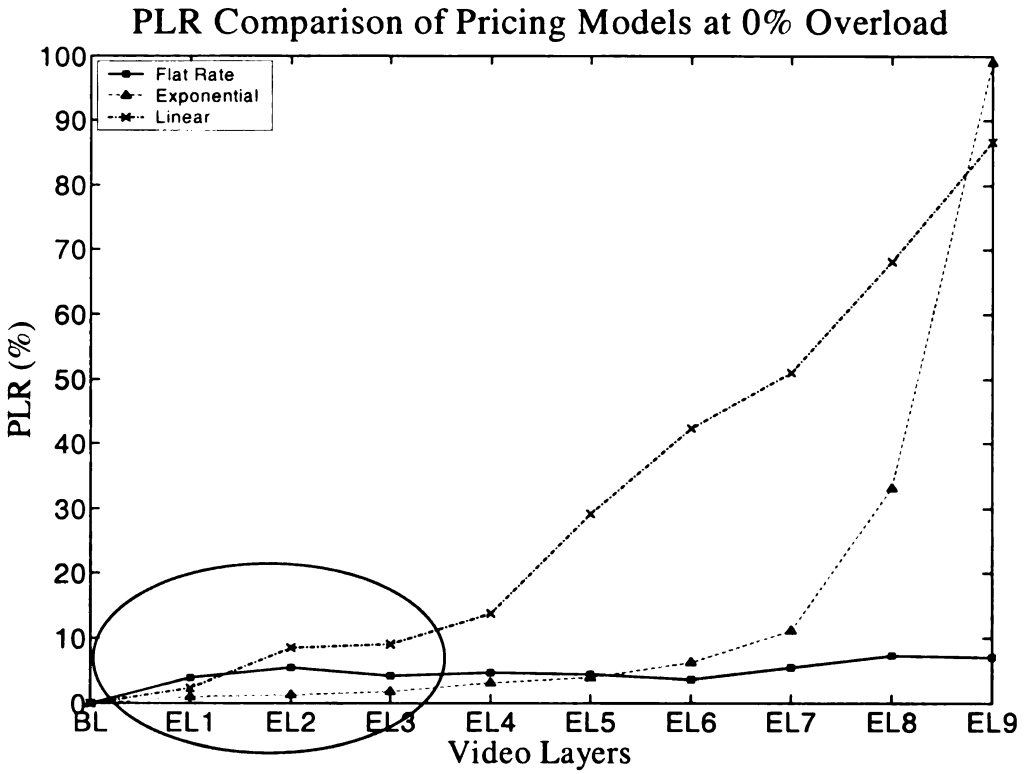

Figure 5-10 PLR comparison of pricing models at 0% overload.

At 100% network overload, as depicted in Figure 5-11, an apparently low PLR is observed using LPM for enhancement layers EL1 to EL4 as compared to the other two pricing models. A circle in Figure 5-11 shows this variation in PLR. Consequently the upper enhancement layers EL5 to EL9 suffered a high PLR for this scenario.

Hence, the results obtained from each simulated scenario of multiple-to-one mapping strategy illustrates that the SLA based on LPM shows a low PLR in the lowest enhancement layer EL1, which is the next most important layer in the video stream after the base layer. This behavior was observed even during heavy network overload conditions. EPM demonstrates the desired PLR response with each increasing video layer only during normal overload conditions. This condition is supposed to be ideal for

scalable video transmissions, where every expected packet loss is pushed towards the last video layer of the stream.

We analyze the network behavior in terms of its performance factor, which in turn is affected by the cost. Due to the high cost paid for lower enhancement layers, EPM does not show any performance advantage over the LPM. The cost comparison of each model is presented in the next section.

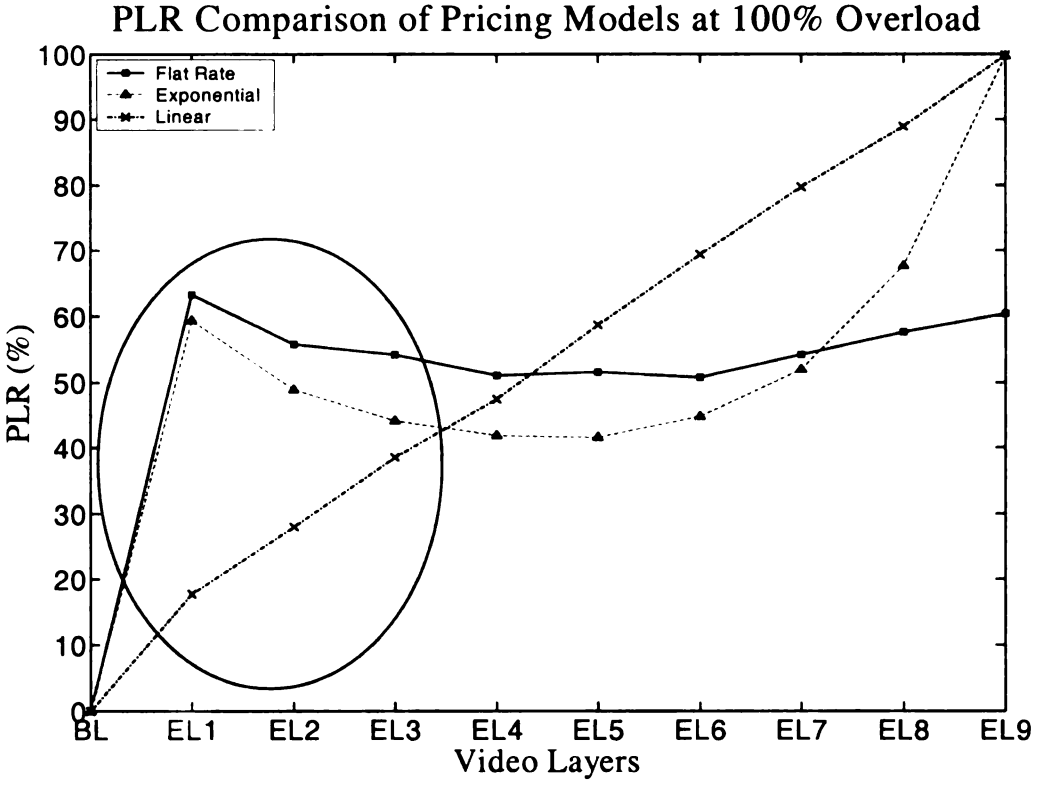

Figure 5-11 PLR comparison of pricing models at 100% overload.

5.4 A Comparative Performance Analysis

The performance analysis presented here is based on two key factors: firstly, the PF *loss* occurred in every strategy when the network is overloaded from 0% to 100%, and secondly the total cost associated with each scenario. The PF loss provides a measure for

the change in overall performance as the network conditions change. Based on this measure, it is desirable that the PF changes gracefully as the network conditions change. Below, we also show the (absolute measured) cost in terms of the bitrate commitment of network (i.e. CIR) according to Equation (3.16).

In order to evaluate the performance loss (P_{fL}) during the transition of network overload from 0% to 100%, we used the following equation:

$$P_{fL} = 1 - \frac{P_f(100)}{P_f(0)} \quad (\%) \tag{4.9}$$

where $P_f(0)$ and $P_f(100)$ are the corresponding PF values at 0% and 100% network overload, respectively. The estimated PF loss and total cost (c_l) for each simulated mapping strategy are listed in Table 5.1 and Table 5.2 respectively.

Table 5.1 Comparison of PF loss in Simulation Scenarios.

Simulation Scenario	$Pf(0)$	$Pf(100)$	$P_{fL}(\%)$
I	**0.2671**	**0.2181**	**18.34**
II(a)	0.1495	0.0969	35.18
II(b)	0.1616	0.1110	31.31
II(c)	**0.1935**	**0.1743**	**9.92**

Table 5.2 Cost comparison of Simulation Scenarios.

Simulation Scenario	**c_{AF} (bps) at 0% overload**	**c_{AF} (bps) at 10% overload**	**c_{AF} (bps) at 50% overload**	**c_{AF} (bps) at 100% overload**
I	**501982**	**524480**	**614471**	**726959**
II(a)	560971	590968	710956	860941
II(b)	522272	547621	649224	776553
II(c)	**429885**	**443216**	**496543**	**563218**

On the basis of the data in Table 5.1 and Table 5.2, we recommend LPM as a framework to attain (1) better quality delivery of scalable video stream at minimum cost, and (2) sustained performance, with a performance loss of only 9.92 % at 100% network overload.

As described in previous section, the EPM shows an ideal PLR behavior for scalable video streaming at normal network condition. This pricing model, however, did not show any consistency in performance with an increasing network overload. Also the cost associated (from Table 5.2) with the performance and quality delivered by the EPM is highest among all the strategies in consideration, even at normal network condition. As compared to the LPM the mean ratio of the total costs for both strategies was also found to be 1: 0.779. By a comparison of the PLR obtained at no overload, the EPM along with the given weight distribution and a high cost provides the best quality of video to the receiver(s). However, this pricing model is also not capable of preserving the desired PLR behavior when the network bandwidth is over-committed.

One-to-one mapping was used in this thesis as a specific case where the generated scalable video streams contained only three layers. In order to utilize the available bandwidth of the network links we used high bitrate video layers in this mapping category. Therefore an overall better performance at every network overload (by using a high bitrate video stream) was obtained, but the performance was not consistent when the network overload was increased from 0% to 100%. A loss of 18.34% in the performance was observed.

6 Conclusions

The main objective of this thesis was to develop methods that provide high quality video transmission over DiffServ. One-to-one and multiple-to-one mapping strategies and related pricing models were proposed and simulated to realize this goal. DiffServ is a commitment-based network, where cost is associated with every transmitted packet. For this reason we analyzed the performance of each mapping strategy and the corresponding pricing models in terms of the cost and the delivered quality i.e., weighted throughput. Another main objective of this thesis was to analyze the effect of network bandwidth over-commitments on the two mapping categories.

The mapping schemes for streaming video were developed for a medium priced class of assured forwarding (AF) service. For this purpose "flow based" video transmission was proposed to apply multiple *service level agreements* (SLAs) to a single video stream. Since performance of every packet stream over DiffServ is also affected by the associated cost of the network transmission services, we used the bandwidth commitment parameter *committed information rate* (CIR), to control the packet losses (or throughput).

During this work we used four network overload conditions to illustrate the over-commitments by the service provider(s) of DiffServ. The simulations performed at 0% overload represent the normal condition where a service provider is committing the

channel bandwidth without exceeding its maximum capacity. The two mapping categories were mainly developed to improve the video quality at normal network conditions. Exponential pricing model (EPM) and Linear pricing model (LPM) were designed for network bandwidth commitments, according to the given weight distribution of video layers. These pricing models demonstrated desired throughput results and thus implied an improvement in delivered video quality when compared to the current best-effort Internet. The associated cost for the application of LPM and EPM was also found to be less than that obtained using the traditional flat rate technique, which proves that a better quality video is achievable without paying a higher cost.

One main reason for analyzing the effect of over-commitments on the delivered quality and performance was to present a framework in which a network can guarantee a stable quality even when the network bandwidth exceeds the available channel capacity. The suggested EPM was able to provide the desired performance and quality during only low over-commitment levels (i.e. up to 10% overload), whereas the LPM showed results that provide assured quality and better performance upto 100% of over-commitment levels.

The simulation results of the proposed low-cost framework, i.e. LPM for multiple-to-one mapping strategy, is capable of preserving the video quality (with minimal performance loss of 9.92%), even as the network link is over-committed upto 100% (which is unusual in practice). Based on the parameter values used, our simulations show that the LPM could provide graceful (and relatively minor) degradation in overall quality while achieving higher utilization of the link capacity. The multiple-to-one mapping scheme is applicable to video streams containing several video layers. On the other hand,

the one-to-one mapping strategy can be reference to as a *high-cost* framework to obtain a consistent quality and performance at least up to 50% over-committed channels.

Future Work Suggestions for future work involve the utilization of improved traffic shaping mechanisms and metering-marking algorithms, to control the PLR within a AF class. A single tradeoff pricing structure to provide the benefits of EPM and LPM during severe network congestion levels could also provide further extension in this work.

In this thesis constant bit rate (CBR) video stream are used in our simulations. Scalable mapping strategies and bandwidth commitment models could be designed to accommodate variable bit rate (VBR) video streams to attain a high quality video transmission and performance for previously known and for emerging video coding standards and methods.

References

[1] Holzmann, G.J., B. Pehrson. *The Early history of Data Networks.* IEEE Computer Society Press, Los Alamitos, CA, 1995.

[2] IETF RFC 791, *Internet Protocol, DARPA Internet Program Protocol Specification*, Sep 1981.

[3] Stevens W. Richard *TCP/IP Illustrated. Volume 1: The Protocols.* Addison Wesley, Reading, MA, September 1998.

[4] Comer, D.E. *Internetworking with TCP/IP. Volume 1: Principles, Protocols, and Architecture.* Prentice Hall, Eaglewood Cliffs, NJ, 3rd Edition, 1995.

[5] IETF RFC 768, *User Datagram Protocol*, Aug 1980

[6] IETF RFC 1889, *RTP: A transport Protocol for Real time Applications*, Jan 1996

[7] Jacobson V. *Congestion Avoidance and Control.* Proceedings of the SIGCOMM'88 pg. 314-329, Aug. 1988.

[8] Ferguson, Paul, and Huston, Geoff. *Quality of Service: Delivering QoS on the Internet and in Corporate Networks.* New York: John Wiley & Sons, 1998.

[9] IETF RFC 2386, "A Framework for QoS-Based Routing in the Internet."

[10] J. Boyce, R. Gaglianello, *Packet Loss effects on MPEG video sent over the public Internet*, Proceedings of ACM Multimedia Conference, UK, 1998.

[11] Draft International Standards, *Generic Coding of Moving Pictures and Associated Audio Information: Video,* ISO/IEC 13818-2, Nov 1994

[12] IETF RFC 3246, *An Expedited Forwarding PHB*, March 2002.

[13] IETF RFC 2597, *Assured Forwarding PHB Group*, June 1999.

[14] *The Network Simulator NS2*, www.isi.edu/nsnam/dist

[15] Floyd, S. and V. Jacobson. *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, vol. 1 No. 4, Aug 1993.

[16] Martin May, Jean Bolot, Christopher Diot, and Bryan Lyles. *Reasons not to deploy RED.* In Proceedings of the IEEE/IFIP International Workshop on Quality of Service (IWQoS'99), June 1999

[17] D. Clark and W. Fang. *"Explicit allocation of best-effort packet delivery service"*. IEEE/ACM Transactions on Networking, 6(4):362--373, August 1998.

[18] T. Ziegler, S. Fdida, U. Hofmann, *"RED+ Gateways for Identification and Discrimination of unfriendly best-effort Flows in the Internet"*, Proceedings of IFIP Broadband Communications 99, November 1999

[19] Feng, W., D. Kandlur, D. Saha, and K. Shin *A Self-Configuring RED Gateway*, Proceedings from INFOCOMM 99, March 1999.

[20] Lin D., R. Morris, *Dynamics of Random Early Detection*, Proceedings from ACM SIGCOMM 97, Oct 1997.

[21] H.Eriksson, *MBONE: The Multicast Backbone*, Comm ACM 37 (8) Aug 1994.

[22] ITU-T Recommendation H.261,*Video Codec for Audio-visual services at Px 64 Kbits*, Mar 1993.

[23] ITU-T Recommendation H.263, *Video Coding for Low bitrate Communication*, March 1996.

[24] International Organization for Standardization JTC1/SC29/WG11 *CODING OF MOVING PICTURES AND AUDIO MPEG* 96/June 1996

[25] International Organization for Standardization JTC1/SC29/WG11 *CODING OF MOVING PICTURES AND AUDIO MPEG* 00/October 2000

[26] International Organization for Standardization JTC1/SC29/WG11 *CODING OF MOVING PICTURES AND AUDIO MPEG* March 2002

[27] I. Busse, B. Deffner, H. Schulzrinne, *Dynamic QoS control of multimedia applications based on RTP*, Computer Communications 19 pgs. 49-68, Jan 1996.

[28] Y. Wang, Q.F. Zhu, *Error Control and concealement for video communication- a review*, Proc. IEEE 86 (5) May 1998.

[29] J. M. Boyce, *Packet Loss Resielient transmission of MPEG Video over the Internet*, Signal Processing: Image Communication 15, pgs 7-24, 1999.

[30] M. Schaar, H. Radha, *Unequal Packet Loss Resielience for Fine-Granular-Scalibility Video*, IEEE Transactions on Multimedia, 3 (4) ,Dec 2001.

[31] H. Radha et. al, *Scalable Internet Video using MPEG-4*, Signal Processing: Image Communication 15(1999) pg. 95-126

[32] IETF RFC 1633, *Integrated Services in the Internet Architecture: an Overview*, June 1994.

[33] IETF RFC 2474,*Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 headers*, Dec 1998.

[34] IETF RFC 2475, *An Architecture for Differentiated Services*, December 1998.

[35] IETF RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*, October 1998.

[36] M. Goyal, A. durresi, R. Jain, *Performance Analysis of Assured Forwarding*, IETF Internet Draft draft-goyal-diffserv-afstudy-00

[37] Xipeng Xiao, Lionel M. Ni, *Internet QoS: the Big Picture*, IEEE Network, 13(2):8-18, Mar-April 1999.

[38] IETF RFC 2697, *A Single Rate Three Color Marker,* September 1999.

[39] IETF RFC 2698, *A Two Rate Three Color Marker,* September 1999.

[40] IETF RFC 2859, *A Time Sliding Window Three Colour Marker (TSWTCM),* June 2000.

[41] Mohammed Atiquzzaman, Hongjun Su, *ItswTCM: A New Aggregate Marker to Improve Fairness in DiffServ*, presented in IPS 2001, (a part of Globecom 2001), November 2001.

[42] M.Goyal, A. Durresi, R. Jain, Chunlie Liu, *Effect of Number of Drop Precedences in Assured Forwarding*, IETF draft-goyal-dpstdy-diffserv-01.txt, June 1999.

[43] Mohamed El-Gendy, Kang G. Shin *Equation-Based Packet Marking for Assured Forwarding Services*, selected for INFOCOMM 2002.

[44] I. Andrikopoulos, L. Wood and G. Pavlou, *A Fair Traffic Conditioner for the Assured Service in a Differentiated Services Internet*, IEEE ICC

[45] IETF RFC 2963, *A Rate Adaptive Shaper for Differentiated Services*, October 2000

[46] N. Semret, R. Liao, A. Campbell, and A. Lazar, "*Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections*", IEEE Journal on Selected Areas of Communications, Vol. 18, No. 12, pp. 2499-2513, December 2001.

[47] M. S. Borella, V. Upadhyay and I. Sidhu. *Pricing Framework for a Differential Services Internet*. European Transactions on Telecommunications, Vol. 10(2), March/April 1999.

[48] IETF RFC 3140, *Per Hop Behavior Identification Codes*, June 2001

[49] IETF RFC 3086, *Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification*, April 2001

[50] U. Bodin, S. Schalen and S. Pink, "*Load Tolerant Differentiation with Active Queue management*", SIGCOMM Computer Communications vol. 30 n3 , July 2000

[51] IETF website, http://www.ietf.org

[52] S.Sahu, P. Nain, D. Towsley, C. Diot and V. Firoiu, "*On Achievable Service Differentiation With Token Bucket Marking for TCP*", In proceedings of ACM SIGMETRICS, June 2000.

[53] Cisco official web site, http://www.cisco.com

[54] NS Manual, http://www.isi.edu/nsnam/ns/ns-documentation.html

[55] H. Radha, M. van der Schaar, Y. Chen, "*The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming over IP*," IEEE Transactions on Multimedia, March 2001.

Appendix

DiffServ Metering-Marking Algorithms

The three types of markers suggested by IETF belongs to two different categories:

1. Token Bucket Markers, includes
    i. Single Rate Three Color Marker (srTCM) [38]
    ii. Two Rate Three Color Marker (trTCM)[39]
2. Time Slide Window Markers
    iii. Time Slide Window Three Color Marker (TSWTCM) [40]

We used srTCM and trTCM marking algorithms for AF packet metering and marking during the simulations. The standard specification of these algorithms is described below:

Single Rate Three Color Marker (srTCM)

Two token buckets C and P with their sizes CBS (bytes) and PBS (bytes) respectively are used in this algorithm. An incoming packet with rate B (bytes per second) is marked as a green or yellow packets depending on the number of available tokens. The srTCM algorithm can be described by the following figure.

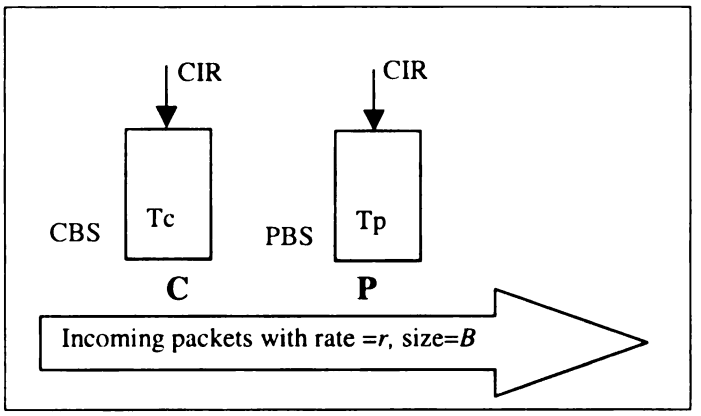

Figure A-1 srTCM Algorithm logic diagram.

The token buckets are initially full at time 0, i.e Token Count Tc(0) =CBS and Token count Tp(0)=PBS. Thereafter the token count Tp is incremented by one CIR times per second upto PBS, and token count Tc is incremented by one CIR times per second upto CBS. For every incoming packet of size B and rate *r*, the srTCM algorithm can be summarized as:

STEP 1. If Tc(t) – B >=0, the packet is *Green*, and Tc is decremented by B, else

STEP 2. If Te(t) –B>=0, the packet is *Yellow*, and Te is decremented by B, else

STEP 3. The packet is Red and neither Tc nor Te is decremented.

The srTCM works in both color blind as well as color aware modes. Marking of a packet with a given color requires that there be enough tokens of that color available to accommodate the entire packet. Committed Information Rate (CIR) is rate used to refill the token bucket in bytes per second.

Two Rate Three Color Marker (trTCM)

The only difference between trTCM and srTCM is that this marker use two rates and two token bucket sizes to meter the incoming packet. Two token bucket C and P are used with sizes CBS (bytes) and PBS (bytes) respectively. Tc and Tp are their respective token count belongs to each token bucket.

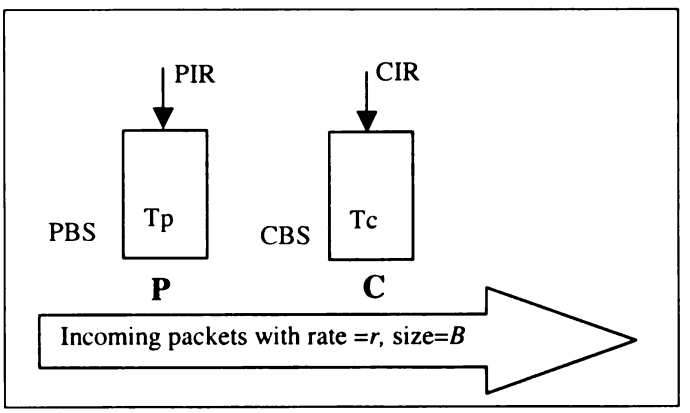

Figure A-2 trTCM Algorithm logic diagram.

The token buckets P and C are initially full at time 0, i.e. the Token Count Tc(0)=CBS and token count Tp(0)=PBS. Thereafter the token count Tp is incremented by one PIR times per second upto PBS, and token count Tc is incremented by one CIR times per second upto CBS. An incoming packet with rate *r* (bytes per second) is marked as a yellow or red packet depending on the number of available tokens.

For every incoming packet of size B and rate *r*, the trTCM algorithms can be given as:

STEP 1. If Tp(t) –B < 0, the packet is Red, else

STEP 2. If Tc(t) –B < 0, the packet is Yellow and Tp is decremented by B, else

STEP 3. The packet is Green and both Tp and Tc are decremented by B.

Two rates i.e. Committed Information Rate (CIR) and Peak Information Rate (PIR) are used to re-fill their respective token buckets. The PBS and CBS are measured in bytes and they always configured to be greater than 0. On the other hand PIR must not be less than CIR.