An Enhanced Algorithm for Fair Traffic Conditioning in IP Differentiated Services

by

ADEGBOYEGA Abiola Olumide

**A Thesis Submitted to the
Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF APPLIED SCIENCE**

**Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
Ottawa Ontario Canada**

January 28, 2004

© 2004, ADEGBOYEGA Abiola Olumide

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-93931-6

Our file Notre référence

ISBN: 0-612-93931-6

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Fairness guarantees provided by Traffic conditioners to ensure Quality of Service (QoS) for Assured Forwarding (AF) Classes in the Differentiated Services (Diffserv) framework are investigated. An existing traffic-marking algorithm, developed to ensure fairness was analyzed for effectiveness. Equation Based Marking (EBM) was introduced to enforce fairness among dissimilar AF TCP flows by monitoring existent network conditions, which is used in marking decisions. The estimation of losses at the network edge forms the basis of marking decisions. The loss rates of different connections were demonstrated to converge hence enforcing a fair marking regardless of the metrics of individual connections. EBM was analyzed for fairness and further enhanced by the implementation of a more efficient technique for loss rate estimation. Comparison was made between EBM and the enhanced technique with results showing appreciable improvements in the maintenance of fairness and stability about an operating point. Furthermore, a service definition required by QoS standards was met with the implementation of the additional algorithm to EBM.

Acknowledgments

My heartfelt thanks go to Mum and Dad and my siblings for their support and love from home. My gratitude goes to Professor Lambadaris for his guidance and support. My thanks to Rupinder Makkar whose invaluable suggestions helped in no small way towards the success of this research. And all praise to the Father of Light whose glorious countenance sustains and shines on us all.

Table of Contents

ABSTRACT	III
ACKNOWLEDGMENTS	IV
TABLE OF CONTENTS	V
LIST OF TABLES	VII
LIST OF FIGURES	VIII
LIST OF ACRONYMS	XI
CHAPTER 1 INTRODUCTION.....	1
1.1 BACKGROUND	1
1.2 MOTIVATION	3
1.3 THESIS CONTRIBUTION	5
1.4 ORGANIZATION	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 QUALITY OF SERVICE AND THE INTERNET	7
2.2 SERVICE DIFFERENTIATION: PACKET MARKING	8
2.2.1 <i>AF: Packet Color, Grade of Service and Target Rates</i>	12
2.3 DIFFSERV ARCHITECTURE: THE DOMAIN, THE EDGE AND THE CORE	13
2.3.1 <i>The Diffserv Edge</i>	14
2.3.2 <i>The Diffserv Core</i>	16
2.4 AGREEMENT BETWEEN CLIENTS AND SERVICE PROVIDERS	18
CHAPTER 3 THE EQUATION-BASED TRAFFIC CONDITIONER	19
3.1 TRAFFIC CONDITIONING	19
3.2 TCP DYNAMICS	20
3.3 EBM PHILOSOPHY	24
3.4 EBM MODULES	25
3.4.1 <i>Calculation of Target Loss Probability</i>	26
3.4.2 <i>Estimation of RTT and Timeout</i>	28
3.4.3 <i>Loss Rate Estimation</i>	31
3.4.4 <i>Marking Function</i>	34
3.5 EBM OPERATIONAL ANALYSIS	36
3.6 CONVERGENCE OF ITERATIONS FOR EBM	40
CHAPTER 4 A CRITICAL ANALYSIS OF LOSS RATE ESTIMATION.....	42
4.1 LOSS HISTORY DISCOUNTING	43
4.2 IMPLEMENTATION OF LOSS HISTORY DISCOUNTING	46

CHAPTER 5 EVALUATION.....	47
5.1 PACKET SIZE EXPERIMENT	52
5.2 RTT EXPERIMENT	61
5.2.1 <i>RTT Experiments with EBM</i>	62
5.2.2 <i>RTT Experiments with RBM</i>	71
5.3 TARGET RATE EXPERIMENTS	75
CHAPTER 6 CONCLUSION AND FUTURE WORK	78
APPENDIX A	80
APPENDIX B	84
REFERENCES.....	87

List of Tables

Table 1: Codepoint Assignment for Diffserv.....	10
Table 2: Parameters for RIO with three-drop precedences.....	48
Table 3: RIO parameters for packet size experiment.....	53
Table 4: Ranges chosen for RTT experiment.....	62
Table 5: RIO parameters for RTT experiment.....	63
Table 6: RIO parameters for target rate investigation.....	76

List of Figures

Figure 1: QoS Definitions for Diffserv and the undifferentiated Internet	8
Figure 2: Diffserv Octet Fields in IPv4 and IPv6 Headers	9
Figure 3: A generic Diffserv Domain	13
Figure 4: Generic Traffic Conditioner	14
Figure 5: Traffic Conditioner employing a Token Bucket Marker	15
Figure 6: RIO Active Queue Management for Diffserv	17
Figure 7: TCP Window evolution and Loss Indication by Triple Duplicate ACKs	21
Figure 8: EBM Feedback loop	24
Figure 9: EBM Modules	26
Figure 10: Duality between loss probability and throughput	27
Figure 11: Congestion window evolution for connections with long and short RTTs	29
Figure 12: Throughput achieved for connections with long and short RTTs	29
Figure 13: RTT and T_0 estimation algorithm	31
Figure 14: Weighted Intervals between loss events for loss interval estimation	32
Figure 15: Three-Color Marking Function for EBM	35
Figure 16: TCP-modeled feedback control system	36
Figure 17: EBM TCP-feedback control system	37
Figure 18: Throughput plotted against RTT for analytical proof	39
Figure 19: Loss-probability versus throughput for variable RTT connections	41
Figure 20: Weight distribution for average loss interval	42
Figure 21: Loss history without discounting	44
Figure 22: Loss history with discounted weights	45
Figure 23: Simulation Network	47

Figure 24: Table 2 represented as a multi-level RED graph	48
Figure 25: TSWTCM marking algorithm	50
Figure 26: Scenario for packet size experiment	52
Figure 27: Goodput vs. packet size with TSWTCM marker	54
Figure 28: Goodput vs. packet size with EBM marker	54
Figure 29: Goodput vs. packet size with EBM-HD marker	55
Figure 30: Loss rate per flow (100-500 bytes) vs. time for EBM marker	57
Figure 31: Loss rate per flow (100-500 bytes) vs. time for EBM-HD marker	57
Figure 32: Loss rate per flow (600-1000 bytes) vs. time for EBM marker	58
Figure 33: Loss rate per flow (600-1000 bytes) vs. time for EBM-HD marker	58
Figure 34: Total Loss rate for all connections for EBM and EBM-HD	59
Figure 35: Fairness Index vs. subscription level for all three markers	60
Figure 36: Scenario for RTT experiment	63
Figure 37: Goodput vs. RTT for TSWTCM	64
Figure 38: Goodput vs. RTT for EBM	64
Figure 39: Goodput vs. RTT for EBM-HD	65
Figure 40: Loss rate per flow (40-93 ms) vs. time for EBM marker	66
Figure 41: Loss rate per flow (106-160 ms) vs. time for EBM marker	66
Figure 42: Goodput vs. RTT (40-400 ms) for TSWTCM marker	67
Figure 43: Goodput vs. RTT (40-400 ms) for EBM marker	68
Figure 44: Goodput vs. RTT (40-400 ms) for EBM-HD marker	68
Figure 45: Fairness Index vs. RTT Range	70
Figure 46: Assurance Level vs. RTT Range	70
Figure 47: Goodput vs. RTT for RBM (40-160 ms)	71
Figure 48: Goodput vs. RTT for RBM-HD (40-160 ms)	72
Figure 49: Goodput vs. RTT for RBM (40-400 ms)	73

Figure 50: Goodput vs. RTT for RBM-HD	73
Figure 51: Assurance Level with increasing RTT	74
Figure 52: Fairness Index with increasing RTT	74
Figure 53: Network for target rate investigation	75
Figure 54: Goodput vs. CIR for EBM, EBM-HD and TSWTCM	77

List of Acronyms

QoS	Quality of Service
Diffserv	Differentiated Services
Intserv	Integrated Services
AF	Assured Forwarding
EF	Expedited Forwarding
TCP	Transmission Control Protocol
EBM	Equation Based Marking
EBM-HD	Equation Based Marking with History Discounting
IETF	Internet Engineering Task Force
BE	Best Effort
PHB	Per Hop Behavior
RTT	Round Trip Time
RTO	Retransmit Timeout
FTP	File Transfer Protocol
IP	Internet Protocol
SLA	Service Level Agreement
BA	Behavior Aggregate
RED	Random Early Detection
RIO	RED with In and Out
ACK	Acknowledgment
TDP	Triple Duplicate ACK Period
CIR	Committed Information Rate

PIR Peak Information Rate

TSWTCM Time Sliding Window Three Color Marker

Chapter 1 Introduction

1.1 Background

The emergence of Internet applications with different delivery requirements has fueled the research in network service provisioning to meet the disparate network transport needs. Standards have been developed to enable efficient service delivery over the Internet network layer, which delivered traffic regardless of application-specific requirements. Differentiated Services (Diffserv) and Integrated Services (Intserv) were defined by the Internet Engineering Task Force (IETF) to provide guidelines for the development of architectures to meet these requirements. In both, a tiered Quality of Service (QoS) framework allocates network resources based on application requirements.

Diffserv has emerged as the preferred choice due to scalability issues plaguing Intserv. Diffserv employs bits in the IP packet header to define a grade of service specific to a particular traffic aggregate. Packets belonging to the aggregate are given a unique 'marking' that would determine the service it receives across the transport network. Various levels of service are available within a single implementation. Based on transport requirements, aggregates with more stringent constraints are marked with a higher grade of service than applications with lower delivery constraints.

Diffserv philosophy moves architectural complexity to the 'edge' of the transport network while the 'core' forwards traffic based on packet marking applied at the edge. The Diffserv edge is responsible for packet marking and ingress traffic conditioning.

Traffic conditioning brings incoming traffic streams into conformance with the forwarding conditions specific to the Diffserv enabled network from 'edge-to-edge'. The Diffserv edge may re-mark traffic streams if they are not already marked from the client. Penalties are also imposed when agreements are violated. If ingress traffic is injected at a rate above the agreed value, conditioners at the Diffserv edge could either reduce incoming traffic rates or drop packets from the traffic stream.

The Diffserv core forwards traffic that has already being marked at the edge. The core also enforces service-provisioning agreements during periods of network congestion by dropping packets on a preferential scale based on applied packet marking. The DiffServ core is usually implemented with router queues. Based on marking, packets can be forwarded to separate service-specific queues or a single queue that allocates queuing resources based on grade of service. QoS is applied during congestion when the packets from aggregates subscribed to a higher service profile are protected over aggregates with a lower service profile. When core router queues overflow packets are dropped first from traffic streams subscribed to a lower service offering. In most implementations, if congestion persists, all packets subscribed to a lower grade of service are dropped before traffic streams with a higher grade of service.

QoS offerings in a Diffserv enabled network such as the one described comes in three standard service definitions. All three services can be implemented in the same network at the same time. The oldest method for traffic delivery on the Internet, which is classless and non-differentiated, is known as “best effort” service in Diffserv. This describes traffic streams that are either unmarked with no service definitions whatsoever or traffic marked with the lowest service offering. Besides best effort in Diffserv architecture exist two other services named “Assured Forwarding” (AF) service [2] and “Expedited Forwarding (EF)” [1]. Both AF and EF (or Premium) services are implemented as a “forwarding” behavior on a per-hop basis between adjacent nodes.

1.2 Motivation

Various practical implementations that provide Diffserv QoS according to IETF guidelines have been developed. However issues of unfairness are existent in these implementations when heterogeneous traffic streams with the same policy share a single Diffserv network. Fairness among TCP flows in a Diffserv network can be defined as the achievement of specified target rates of individual connections regardless of their metrics or characteristics. Unfairness occurs when some TCP connections in a shared link occupy more than their fair share of the available bandwidth to the detriment of other connections due to the characteristics that give them an unfair advantage. TCP connections are governed by a throughput equation that is directly proportional to packet size and inversely proportional to the Round Trip Time (RTT).

Thus, connections with large packet sizes that share the same network with those with small packet sizes obtain more than their fair share of bandwidth. In the same vein,

connections with a shorter RTT through the network hog the bandwidth over connections with a longer RTT.

While Diffserv allocates network resources based on their individual requirements, unfair bandwidth and resource allocation occurs when connections subscribed to the same Diffserv network that implements a single policy for all flows are different based on the metrics discussed above. Fairness needs to be enforced in a Diffserv network in order that all connections meet their specified target rates regardless of network and flow metrics.

The service definitions that govern the AF Per-Hop-Behavior (PHB) provide QoS across a wide variety of applications with less stringent demands in terms of delay and loss required by EF. AF philosophy guarantees that user traffic will be delivered with a “high degree of certainty” provided that end hosts do not exceed the agreed subscription rate. The degree of certainty depends on the resources available on the network [1]. If the subscription rate is exceeded and surplus bandwidth is available, user traffic *may* be delivered but not with the same assurance as traffic at or below the subscription rate. A penalty may also be incurred.

The algorithm that is studied employs the classical TCP rate equation that governs packet transmission in IP networks. The TCP equation is used to predict the bandwidth share required by AF sending hosts. The algorithm integrates this bandwidth estimation with an efficient estimation of packet loss rates at the network edge. This enables feedback to the sending hosts to “back off” when their sending rates approach or exceed

prescribed levels. This technique affords some degree of efficiency in enforcing fairness beyond the traditional approaches that will also be discussed.

1.3 Thesis Contribution

This thesis presents an improved scheme for loss rate estimation and its implementation in edge traffic conditioning to improve fairness. It was discovered during experimentation that the QoS delivered by the initial algorithm degrades when the total subscription rate of all user TCP flows is equal to or more than the bottleneck link rate in a variety of scenarios. The implementation of a better marking scheme based on the more efficient mechanism for the estimation of loss rates has lead conclusively to significant improvements. The contributions are as follows:

- Enhancement of the sensitivity of traffic marking when TCP flows have small packet sizes. The original algorithm was discovered to be insensitive to flows with small packets sizes, which lead to inadequate packet marking for such flows, and hence unfairness with regards to achieving the specified target rates for such flows. The implementation of the more efficient loss rate estimation scheme remedied this problem
- Stabilization of the algorithm at an operating point with respect to TCP flows with different RTTs. The original algorithm achieved fairness to *an acceptable degree* with regard to RTT over a given range. The drawback however was that flows

with shorter RTTs were the only ones able to meet target rates while flows with longer RTTs only reached 90% of their target rates. The new implementation offers a true fairness measure that stabilizes the throughput of all connections at the given target rate.

- With the implementation of the enhanced algorithm, an acceptable definition of QoS in agreement with standards required for Diffserv AF was achieved.

1.4 Organization

The thesis is divided into six chapters. Chapter 2 gives an overview of Diffserv with focus on AF. Chapter 3 deals with Traffic conditioning and introduces the Equation Based Marker developed by [3] and its mechanisms of enforcing fairness. Chapter 4 introduces “History Discounting” to the loss rate estimation algorithm integral to Traffic conditioning in [3]. Chapter 5 reports the experiments conducted with various scenarios with a full comparison of the original algorithm, the improved algorithm and one of the traditional Diffserv traffic conditioners. Conclusions are made in Chapter 6.

Chapter 2 Literature Review

2.1 Quality of Service and the Internet

The data delivery at the OSI network layer was performed without differentiation between the types of application with respect to their bounds in terms of loss, delay and jitter. Furthermore, robustness was built into the succeeding (transport) layer, which considers its underlying IP datagram network as being unreliable. In this regard, recovery mechanisms at the transport layer mitigated losses at the network layer. While the IP layer forms the backbone of the Internet, the recovery mechanisms of the TCP layer were adequate for its original applications: Email is delivered via the Simple Mail Transfer Protocol (SMTP) which has no real time requirements while File Transfer Protocol (FTP) recovers from lost packets with TCP's retransmit mechanisms. While the robustness of TCP is indispensable for efficient data delivery on the Internet today, the flat transport scheme of all IP packets across the network layer regardless of application has proved inadequate. With the emergence of voice and video as Internet applications, it was observed that an upper bound of delay was required in terms of packet delivery time from end to end. Furthermore, an upper bound of packet loss existed beyond which voice communications over the Internet is rendered unintelligible. The transport of video packets required even more stringent demands both in terms of delay and jitter.

Diffserv enables service differentiation through packet marking at the IP layer to meet application specific QoS. Figure 1 compares the undifferentiated Internet with the service definitions given by Diffserv.

Figure 1: QoS Definitions for Diffserv and the undifferentiated Internet

2.2 Service Differentiation: Packet Marking

Within the IP packet header, a six-bit Diffserv field is defined [4]. This bears compatibility with the now obsolete IP version 4 Type of Service (TOS) octet and the future IP version 6 Traffic Class. The Diffserv field employs six of its eight bits to assign the packet to a PHB that would determine its treatment across two adjacent network nodes. The remaining two bits are unused in a Diffserv enabled network in determining the PHB of a packet [4]. Figure 2 [4][5] gives the Diffserv field in the IPv4 and IPv6 headers.

Version 4	IHL	Type of Service	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
IP Options			

Version 6	Traffic Class	Flow Label
Payload Length	Next Header	
Source Address		
Destination Address		

0	1	2	3	4	5	6	7

0-5: Diffserv Codepoints

6-7: Used for Error Congestion Notification (ECN)

Figure 2: Diffserv Octet Fields in IPv4 and IPv6 Headers

All standard PHBs (AF, EF, Default and Class Selector) possess a specific Diffserv Codepoint (DSCP) mapping the standard PHBs on the first six bits correspondingly. Once a packet is assigned a DSCP, all packets bearing the same DSCP are grouped into a Behavior Aggregate (BA), which will have a unique forwarding treatment in a Diffserv enabled network.

The Codepoints are illustrated in Table 1 below. The EF service is reserved for applications that require low loss and low latency as well as low jitter. It is chosen only

for the most critical applications (VoIP, online-trading etc) with the avoidance of preferential dropping as done in Diffserv when congestion occurs [4].

Table 1: Codepoint Assignment for Diffserv

Expedited Forwarding (EF)				
1 Assigned Codepoint.	101110			
Assured Forwarding (AF)				
Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low Drop Precedence	(AF11)	(AF21)	(AF31)	(AF41)
	001010	010010	011010	100010
Medium Drop Precedence	(AF12)	(AF22)	(AF32)	(AF41)
	001100	010100	011100	100100
High Drop Precedence	(AF13)	(AF23)	(AF33)	(AF43)
	001110	010110	011110	100110
Best Effort (BE)				
Default	000000			

Four classes AF1 to AF4 are defined in Assured Forwarding. Each class is given an amount of buffer space with different forwarding assurances. This allows the bandwidth resource to be partitioned based on application requirement. Within each class AF1x for example, the drop precedence dp , increases in the order:

$$AF11 \leq AF12 < AF13$$

Thus, when congestion occurs, packets marked AF13 will be dropped before packets marked AF12. Subsequently AF11 packets are dropped. Among AF classes, a level of precedence can be configured that would determine which has the highest level of precedence. This has been configured in practical networks as the Olympic service giving Gold, Silver and Bronze service offerings. Subscribers to the Gold service are given precedence over the Silver service with Bronze service subscribers having the lowest precedence. From Table 1 above, AF1x can be assigned the Gold service and given 50 percent of the available bandwidth while AF2x is given the Silver service with 30 percent and AF3x Bronze with 20 percent [4]. In this configuration the fourth standard AF4x class is unspecified.

Packets marked as '000000' are offered the Best Effort service by default. There are no guarantees for this service.

2.2.1 AF: Packet Color, Grade of Service and Target Rates

In practical AF implementations within one class, packets are marked GREEN, YELLOW and RED with the drop precedence decreasing in the same order. When queues approach congestion, RED packets are dropped first. With sustained congestion, YELLOW packets are dropped only after *all* RED packets have been dropped followed by the drop of green packets. A different approach to marking is also the differentiation of packets as 'in-profile' and 'out-of-profile'. In-profile packets have the same protection profile as GREEN packets while 'out-of-profile' packets correspond to RED in the three-color marking scheme.

For most marking schemes, comparison is made between the current rate of the BA and two target rates. The Committed Information Rate (CIR) is the bandwidth guaranteed by the provider while the Peak Information Rate (PIR) is the surplus bandwidth, which may be available beyond the CIR. If the subscription rate is below or equal to the CIR, packets are marked as green. If the rate exceeds the CIR but is less than the PIR, yellow marking commences. Rates in excess of the PIR results in red marking.

2.3 Diffserv Architecture: The Domain, the Edge and the Core

A collection of contiguous nodes that implement a given set of Diffserv policies with coordinated administration across the network is termed a Diffserv domain [4]. The complex functionalities are moved to the domain's edge while the core forwards user traffic and manages congestion. Figure 3 gives a schematic representation of a Diffserv Domain.

Figure 3: A generic Diffserv Domain

2.3.1 The Diffserv Edge

The edge device (router) is responsible for bringing ingress traffic (which may be pre-marked at the host) into compliance with the policies of the domain. The edge router performs traffic conditioning functions, which are applicable to a behavior aggregate, an application flow or a partition of traffic [4]. The functionality of the Diffserv edge is captured in Figure 4 [6].

Figure 4: Generic Traffic Conditioner

Arriving packets are selected by the classifier based on the contents of one or more of the packet headers [5]. As required, a meter measures the ingress traffic against a profile specified in a conditioning agreement (between client and service provider or between domains). This measurement triggers the subsequent action performed on the packet, which determines it to be either “in-profile” or “out-of-profile” [6]. Markers set the Diffserv field of an incoming packet to a given codepoint or re-mark the packet by changing an existing codepoint if the packet is already marked. The shaper “shapes” ingress traffic usually by delaying some or more packets in order to bring the traffic

stream into conformity with a specific traffic profile. The inner workings of the traffic conditioner that employs a token bucket marker are given in Figure 5 [1].

Figure 5: Traffic Conditioner employing a Token Bucket Marker

For traffic ingress to the edge router’s traffic conditioner, a token bucket “fills” with tokens at a flow rate specified by a usage profile [1]. Each token’s “worth” is a number of bytes also determined by the usage profile. With the presence of tokens in the bucket, ingress packets are marked as “in-profile”, a conformance marking (in which case the token bucket acts as a *shaper*). If a token is not present when packets arrive, depending on the configuration, the packets could be buffered until the token bucket fills up again or else packets could be dropped (in which case it now acts as a *dropper*). The ingress traffic peak rate cannot be more than the token bucket rate and ingress traffic is permitted a “burst size”, a traffic injection rate that does not exceed the token bucket depth.

2.3.2 The Diffserv Core

Once traffic streams are conditioned from the edge router, core routers enforce the domain PHB by forwarding packets based on priority (grade of service/markings). In addition they implement congestion management at the router queues based on the marking applied at the domain edge. Forwarding is achieved through priority queuing while a queue management algorithm implements dropping during congestion. The two are implemented as one mechanism. A typical implementation is done with the Random Early Detection (RED) congestion management algorithm [7]. This mechanism anticipates congestion by active measurement of the queue length using a moving average, a value that is compared with two thresholds, a minimum and a maximum set according to network requirements.

With the active queue length below the minimum threshold, traffic streams pass through queues without packets being dropped. With the active queue length between the thresholds, packets are dropped probabilistically with the probability of drop increasing as the queue length approaches the upper threshold. Beyond the upper threshold packets are dropped with a probability of one. Clark and Fang [8] implemented a set of thresholds in RED one each for “in-profile” and “out-of-profile” traffic for Diffserv. The mechanism ensures protection of packet streams adherent to traffic profiles during periods of congestion. Figure 6 illustrates RED with IN and OUT (RIO). Two sets of thresholds one for each in-profile and out-of-profile traffic. For out-of-profile packets, when the average queue size reaches a value x (say) that lies between the minimum threshold $\min_th(OUT)$

and maximum threshold max_th(OUT) , the packet is dropped with a probability $dprob$ calculated as:

$$dprob = \frac{x - \text{min_th(OUT)}}{\text{max_th(OUT)} - \text{min_th(OUT)}} \text{max } p_{out} \quad (1)$$

Figure 6: RIO Active Queue Management for Diffserv

The drop probability is calculated for in-profile packets the same way. In Figure 6 the maximum threshold for OUT packets overlaps the minimum threshold for IN packets. Other implementations could be non-overlapped or have the same thresholds (for IN and OUT) but differ in the maximum drop probabilities.

2.4 Agreement between Clients and Service Providers

For the delivery of end-to-end services, an agreement exists between clients and service providers and also among different providers. This is defined as a Service Level Agreement (SLA). This covers the forwarding treatment of a subscriber's traffic across the providers network and it includes all traffic conditioning agreements along with the properties of the traffic stream within the domain such as rate and burst size [6]. As a QoS example, a provider offering Gold service to a customer in connecting two of its sites across the provider network will have in the SLA that no more than 5% of all user traffic will be dropped from end-to-end provided the subscriber adheres to a subscription rate of 4 000 kbps and a burst size of 4 kbps. A delay of 10 ms is guaranteed with the SLA. If the provider fails to provide the minimum agreed service conditions, the service charge for the utility reduces accordingly. Generally the SLA agreements are based on statistical bounds and percentiles.

Chapter 3 The Equation-Based Traffic Conditioner

3.1 Traffic Conditioning

The Equation Based Marker [3] was developed to enforce fairness among disparate AF traffic flows based on the classical TCP rate equation. The conditioning function of EBM was modeled on the inverse of the TCP rate equation. TCP flows make up the bulk of traffic flows on the Internet [9]. Thus the implementation of its transmission model in edge conditioning functions enables the accurate prediction of the required bandwidth share of heterogeneous connections.

Various solutions have been devised to combat the unfairness that is inherent when TCP connections sharing the same network are different based on RTT, packet size, target rates and number of microflows. These solutions are based on the TCP transmission model. The Adaptive Packet Marking APM algorithm proposed in [10] monitors TCP mechanisms and preserves target throughput rates. The implementation of APM would need to be deployed in the TCP sender and receiver, which would require redesign of the whole network. The intelligent traffic conditioner [11] utilizes a simple TCP model used in [12] in a manner similar to EBM. This approach works with the precondition that all subscribers cooperate and also requires the supply of external inputs, which makes implementation complex. The Round-Trip-Time Retransmit Time-Out (RTT-RTO) aware conditioner developed in [13] captures RTT dynamics also based on the TCP model. It ensures fairness with regard to long and short RTT connections and ensures small window protection. However it has not been investigated with respect to the other factors that cause unfairness in Diffserv. Finally the Fair Marker introduced in

[14] and the TCP-friendly marker [15] while addressing unfairness introduces complexities without a thorough analysis being done yet. The uniqueness of EBM lies in its ability to capture the integral factors that affect unfairness as given in the complete TCP model.

3.2 TCP Dynamics

TCP is a connection-oriented end-to-end transmission protocol that assures packet delivery and implements its own recovery mechanisms. TCP is composed of four algorithms: slow start, congestion avoidance, fast retransmit and fast recovery [16]. Following the establishment of a connection between end hosts [17], the first phase of the TCP algorithm begins with slow start. Here, the sender injects packets into the network at a rate below a specified “congestion window” which is the limiting threshold on senders congesting the network. In essence it is a phase that observes the rate at which acknowledgements are returned for sent packets and the rate at which the sending rate should be advanced given the limiting congestion window.

The sender begins by transmitting one segment of data (with a unique sequence number) and then waits for its acknowledgement (ACK). Upon receipt of this, the congestion window is incremented from one to two followed by the transmission of two segments. The sending rate follows this exponential increase until it approaches the congestion window or when packets are dropped the flow control phase is initiated [16].

The congestion avoidance phase deals with recovery after dropped packets signal congestion. Loss is indicated either as a Timeout or by the receipt of Triple duplicate ACKs (TD) for lost packets, i.e., the reception of four ACKs with the same sequence

number without any interleaving packets. The congestion avoidance phase reduces the sending rate (cutting the congestion window in half) and a subsequent advancement by one segment every RTT upon every ACK. Figure 7 illustrates the slow-start and congestion avoidance phases for a TCP sender in which loss is indicated by the reception of Triple Duplicate ACKs (TD).

Figure 7: TCP Window evolution and Loss Indication by Triple Duplicate ACKs

Transmission starts with the congestion window advanced from zero until the current size of the congestion window W is reached [18]. With the transmission of all packets falling into this window, the sender waits for an ACK whose reception marks the end of the round, which is equal to the RTT. If b is the number of packets acknowledged

by an ACK, most current implementations of TCP send just one cumulative ACK for two consecutive packets received making b typically 2. With the transmission of W packets in one RTT, the sender receives W/b acknowledgements. In Figure 7 the evolution of the congestion window lies between two TD loss indications. A TD period (TDP) is defined as the period between two loss indications. Following the arguments in [18], for the i th TDP, the number of packets Y_i is sent in a period A_i the duration between two TDPs. With W_i denoting the window and $\{W_i\}_i$ considered to be a Markov regenerative process (Appendix B) with rewards $\{Y_i\}_i$, it can be demonstrated that:

$$B = \frac{E[Y]}{E[A]} \quad (2)$$

Where B is the long-term steady-state sending rate of a TCP connection. Based on succeeding arguments [18], the complete analysis gives the steady state equation as a function of loss indication p :

$$B(p) = M \frac{\frac{1-p}{p} + W(p) + \frac{Q(p, w)}{1-p}}{RTT \left(\frac{b}{2} W(p) + 1 \right) + \frac{Q(p, w)F(p)T_o}{1-p}} \quad \text{If } W(p) < W_{max} \quad (3)$$

$$B(p) = M \frac{\frac{1-p}{p} + W_{max} + \frac{Q(p, w_{max})}{1-p}}{RTT \left(\frac{b}{8} W_{max} + \frac{1-p}{pW_{max}} + 2 \right) + \frac{Q(p, w_{max})F(p)T_o}{1-p}} \quad \text{Otherwise}$$

The parameters in Equation (3) above are explained as follows:

M is the maximum segments size in bits

W_{max} is the maximum congestion window in packets

T_0 is a Timeout

b is the number of packets acknowledged by an ACK

p is the loss probability

RTT is the Round-Trip-Time of the connection

$W(p)$ is the average window size

$$W(p) = \frac{2+b}{3b} + \sqrt{\left(\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2\right)}$$

$Q(p, w)$ is the probability that the loss in a congestion window is a Timeout.

$$Q(p, w) = \min\left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{(w-3)}))}{1-(1-p)^w}\right)$$

Here w , for convenience denotes the current congestion window $W(p)$

$Q(p, w_{max})$ is the probability that the loss when the congestion window is constrained to the maximum window W_{max} is due to a Timeout

$$Q(p, w_{max}) = \min\left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{(w_{max}-3)}))}{1-(1-p)^{w_{max}}}\right)$$

Here again, w_{max} , for convenience denotes the maximum congestion window W_{max}

$F(p)$ is the average duration of a timeout sequence

$$F(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6$$

The inverse of this steady-state equation does not have a closed form solution. The iterative method employed in EBM to compute the numerical solution is given in Appendix A. The bandwidth B is replaced by the CIR and PIR of the AF subscriber.

3.3 EBM Philosophy

All the factors in Equation (3) are encoded in the traffic conditioner at the edge router, which works in the inmost feedback loop given in Figure 8.

Figure 8: EBM Feedback loop

The effectiveness of EBM lies in its ability to estimate the loss rates of TCP connections at the network edge, which is normally sensed by TCP sender and receiver only. With packet loss events sensed and employed as a basic metric at the traffic conditioner, efficient packet marking is done based on the bandwidth requirements of all competing connections. Through the inmost feedback loop in figure 8, the conditioner uses the TCP equation to estimate the sending rates and thus the required packet marking.

Floyd et al implemented equation based congestion control [19] for unicast TCP applications. The marking function in EBM is based on the loss rate estimation developed in their TCP-friendly rate control algorithm and used in the context of AF. The reaction to packet losses is used to estimate target loss probabilities for the CIR and PIR of individual connections for provision of fairness and QoS.

3.4 EBM Modules

The operation of EBM is illustrated in Figure 9 below. Ingress traffic at the edge are conditioned and marked based on the parameters of RTT, CIR, PIR, Loss rate and maximum window size. Loss rate and RTT are estimated in dedicated modules. Each module senses the current RTT and Loss rate based on a measurement window that can be adjusted to improve performance. Each module is discussed in the following sections.

Figure 9: EBM Modules

3.4.1 Calculation of Target Loss Probability

For a given maximum congestion window size, RTT and packet size, Equation (3) gives a one-to-one mapping between the sending rate and the loss rate. This is employed to compute the target loss probabilities that correspond to the CIR and PIR respectively. The inverse of Equation (3) with respect to p the loss rate, denoted as T_p^{-1} ; is a function of four variables: the RTT, Retransmit Timeout, the maximum congestion window and the CIR (or PIR as required). It gives the target probabilities as:

$$\begin{aligned}
p_{CIR} &= T_p^{-1}(CIR, RTT, To, W_{\max}) \\
p_{PIR} &= T_p^{-1}(PIR, RTT, To, W_{\max})
\end{aligned} \tag{4}$$

From equation 4, the inverse of the TCP equation is required for this calculation. There is however no closed form existent for the solution. Numerical iterations are used to make sure that calculated values converge to an accurate value [3]. The convergence criterion will be discussed. The duality between the target loss probabilities and the sending rates is illustrated in Figure 10 below.

Figure 10: Duality between loss probability and throughput

The target loss probability p_{cir} maps to the corresponding target throughput rate CIR while the target loss probability p_{pir} maps to the corresponding target throughput rate PIR. This duality enables EBM to compute the required marking probability that determines whether the packet should be marked GREEN, YELLOW or RED. Fairness is

ensured when target loss probabilities as well as target throughput rates are computed for flows with disparate metrics.

3.4.2 Estimation of RTT and Timeout

TCP transmission is advanced by ACKs for received data as illustrated in the sections above. The advancement of the congestion window is done according to the feedback across the network; hence connections with a shorter network path update the TCP senders faster than connections with a longer network path. This adversely affects the bandwidth share of connections with longer RTTs. Figure 11 show the congestion window evolution for four TCP connections sharing a common network link. Two have an RTT of 240 ms while the other two have an RTT of 600 ms. As shown, the shorter RTT connections update their senders in a shorter time and thus are able to advance their congestion windows to higher values. Figure 12 in turn shows the throughput achieved with 10 connections sharing the same link, five each having RTTs of 240 ms and 600 ms respectively. The shorter connections thus achieve the higher throughput to the detriment of the connections with a longer network path. This affects the achievable throughput in Diffserv AF as well.

Figure 11: Congestion window evolution for connections with long and short RTTs

Figure 12: Throughput achieved for connections with long and short RTTs

There are situations in which packets (or ACKs) are lost and less than three duplicate ACKs are received at the sender. In such situations the sender waits for a period of time referred to as a Timeout denoted by T_0 . At the expiration of the Timeout the congestion window is reduced to one followed by the transmission of a single packet [18].

The RTT estimation is the same as the module written into available implementations of TCP. TCP senders and receivers both use sequence numbers to measure RTT. Every time the receiver sends an ACK to the sender, it includes the sequence number from the most recent data packet. This enables the sender to measure the RTT through the network. Subsequently, the measured RTT is smoothened using an exponentially weighted moving average. This weight is used to determine the responsiveness of the transmission rate to changes in the round trip time [19]. This functionality is encoded as a module in the edge marker for EBM.

From the sender perspective, the value of the retransmit timeout T_0 (t_{RTO} below) is determined from the TCP algorithm with the formula:

$$t_{RTO} = SRTT + 4 * RTT_{var} \quad (5)$$

RTT_{var} is the variance of RTT and $SRTT$ is the estimate of the RTT. In practice, t_{RTO} only significantly affects the allowed sending rate when packet loss rate is extremely high. In practice, different TCP implementations use different clock granularities to calculate

retransmit timeout [19]. This makes it difficult to define an accurate model for all implementations but it is claimed in [19] that equation 5 above works reasonably well. The estimation algorithm used in EBM is given in Figure 13 below.

```

With a given estimation window
Upon receipt of an ACK, record and timestamp a TCP packet
/*Check if timestamp is for previously recorded packet*/
If (seqno of ACK  $\geq$  recorded seqno)
    current rtt := now - recorded timestamp
    /*Calculate the Weighted Moving Average (WMA) of the measured RTT values*/
    srtt =  $w * rtt + (1-w) * srtt$ 
    delta =  $| rtt - srtt |$ 
    rttvar =  $w_{rto} * delta + (1 - w_{rto}) * rtt_{var}$ 
    T0 =  $srtt + 4 * rtt_{var}$ 
end If

```

Figure 13: RTT and T₀ estimation algorithm

3.4.3 Loss Rate Estimation

The estimation of loss rate is pivotal to the efficiency of the EBM marker. The estimated loss rate measures the loss event rate and not the packet loss rate i.e. the interval between two loss events. The loss event consists of several packets lost within a RTT. Various methods have been devised for calculating loss events and as discussed in

[19], the most efficient is the average loss interval method. A *loss interval* is defined as the number of packets between loss events. This is illustrated in Figure 14 below.

Figure 14: Weighted Intervals between loss events for loss interval estimation

The average loss interval $\hat{s}_{(l,n)}$ is calculated as the weighted average of the last n intervals.

This is given as:

$$\hat{s}_{(1,n)} = \frac{\sum_{i=1}^n w_i s_i}{\sum_{i=1}^n w_i} \quad (6)$$

for weights w_i :

$$w_i = 1, 1 \leq i \leq n/2,$$

$$w_i = 1 - \frac{i - n/2}{n/2 + 1}, \quad n/2 < i \leq n$$

For $n = 8$, the weights are given as 1, 1, 1, 1, 0.8, 0.6, 0.4 and 0.2 (from w_1 to w_8). In practice and with the analysis made in [19], $n = 8$ is an accepted value which is the lower bound that achieves stability and strikes a reasonable balance between resilience and noise. The calculation of loss event rates over a number of loss intervals requires a means to deal with the interval since the most recent loss, that is the last interval that is not terminated by a loss event. If s_i is the number of packets in the i -th most recent loss interval, then let s_0 be the interval containing packets that have arrived since the last loss. Since this interval is not terminated by a loss, s_0 becomes s_1 with the most recent packet loss and all intervals are shifted down by one accordingly. The given loss event rate is then given as $1/\hat{s}$.

3.4.4 Marking Function

Marking is based on proportionality to the target loss probabilities p_{cir} and p_{pir} derived in Equation (4) above. With the computation of p_{cir} and p_{pir} , a linear function is used to calculate P_{yellow} (probability of marking a packet as yellow) and P_{red} (probability of marking a packet as red) when a three-color marking scheme is used. The marking scheme is similar to the scheme employed in the Time-Sliding Window Three Color Marker TSWTCM [25]. Here the CIR and PIR are replaced by the p_{cir} and p_{pir} . This is given in Figure 14 below. The $YScale$ and $RScale$ are design parameters based on current network conditions. Referring to Figure 15, the computation of the loss rate is a measure of the sending rate relative to the CIR and PIR with inverse duality between the loss rate and the sending rate. As given below if the $loss\ rate \geq p_{cir}$ (i.e. if $sending\ rate \leq CIR$), then the sender's subscription rate is below the CIR and therefore packets are marked GREEN. With $p_{cir} > loss\ rate > p_{pir}$ ($CIR < sending\ rate < PIR$), the sender has exceeded the subscribed information rate but still less than the peak information rate so the packet is marked as YELLOW which has a lower guarantee of delivery than a packet marked as GREEN. When $loss\ rate < p_{pir}$, the sending rate has exceeded the PIR and thus packets are marked as RED.

```

If (loss_rate) > pcir
    mark packet as GREEN

Else if ( pcir > loss_rate > ppir )
    Pyellow = (pcir - loss_rate)/pcir x (pcir x YScale)
    with probability Pyellow mark packet as YELLOW and
    with probability (1 - Pyellow) mark packet as GREEN

Else if (loss_rate < ppir )
    Pred = (ppir - loss_rate)/ppir x (ppir x RScale)
    Pyellow = (pcir - ppir)/ppir x (ppir x YScale)
    with probability Pred mark packet as RED and
    with probability Pyellow mark packet as YELLOW and
    with probability (1 - (Pyellow + Pred)) mark packet as GREEN

```

Figure 15: Three-Color Marking Function for EBM

3.5 EBM Operational Analysis

The steady-state analysis of EBM is given. The proof of EBM's effectiveness in enforcing fairness which is found in [3] follows the approach used in [20] in which the system is modeled as a feedback control system. This is given in Figure 16 below.

Figure 16: TCP-modeled feedback control system

In this control system, the TCP senders are being controlled. The controlling element is the drop module B (any active queue management technique like RIO already discussed). This has an average queue length, which is a function of the drop rate p , which is also the feedback signal. This is different from a classical control system in that there are several simultaneously controlled elements (the TCP senders) and not one [20].

The system employed for the analysis is shown in figure 17 below. There are n TCP flows passing through a common link l with a capacity C over a single hop. The results are valid for a multi-hop scenario. All access links $(r_l - r_n, d_l - d_n)$ have sufficient

bandwidth so that link l is the only bottleneck in the network. The following assumptions are made concerning the system:

Figure 17: EBM TCP-feedback control system

- All TCP connections (flows) have the same parameters but differ only in RTT (The same could be done with the packet size as the only variable parameter).
- The system is assumed to be in steady state i.e. TCP sources are not in slow-start and they have unlimited data to send. There exists surplus bandwidth in the bottleneck link making the network *undersubscribed*.
- A two-color RIO gateway is employed for active queue management with a fixed loss probability for IN and OUT packets

With the foregoing assumptions, the proof of EBM's effectiveness with regards to fairness is iterated below and a numerical example is given.

For the set of TCP flows f_i , $1 \leq i \leq n$, each with a unique RTT denoted as RTT_i :

1) Calculate the throughput $r_i(Loss_i, RTT_i)$ from the classical TCP (Equation (3)) where $Loss_i$ is the loss probability seen by each flow f_i .

2) $Loss_i$ is calculated from the RIO loss probability for each flow with its packet marking probability given as:

$$Loss_i = P_i * loss_{out} + (1 - P_i) * loss_{in} \quad (7)$$

3) The marking probability P_i is calculated for each flow with a marking function similar to Figure 15 with two colors, one for IN packets and one for OUT packets.

$$P_i = \frac{p_{cir_i} - Loss_i}{p_{cir_i}} * (p_{cir_i} * Scale) \quad (8)$$

Where $Scale$ is based on network conditions

4) The value of p_{cir_i} is calculated for each flow using its RTT_i from the inverse of the TCP Equation (3) with respect to p as:

$$p_{cir_i} = T_p^{-1}(CIR_i, RTT_i) \quad (9)$$

5) Solving for $Loss_i$ using (7) and (8) above:

$$Loss_i = \frac{p_{cir_i} * (loss_{out} - loss_{in}) * Scale + loss_{in}}{1 + (loss_{out} - loss_{in}) * Scale} \quad (10)$$

With the numerical values given in [3] and for RTTs distributed between 200 and 900 ms, Figure 18 below shows the fairness achieved by EBM.

Figure 18: Throughput plotted against RTT for analytical proof

3.6 Convergence of iterations for EBM

There is no closed form solution for the inverse of the classical TCP equation (Equation (3)); therefore iterations are used in the calculation of the requisite values for the target throughput marking probabilities p_{pir} and p_{cir} . Following the analysis in [3], the bisection method in [21] is used for the numerical solutions. Since all probability is bound in the interval [0,1], the numerical iterations start with a value of $p=0$ and ends with $p=1$. However the inverse TCP equation with respect to p the loss probability is multivariate as given in Equation (4) repeated below:

$$P_{CIR / PIR} = T_p^{-1} (CIR / PIR, RTT, To, W_{max})$$

With these variables, there exist a limiting value for p for which a maximum throughput r can be found. Since the iterations begin from $p=0$, with all variables factored in, this maximum value of throughput is deduced from the iteration which occurs at a low value of drop probability p i.e. the higher the loss indication (packet loss) the lower the congestion window of the sender and hence the throughput and vice-versa. Figure 19 shows the results of the iterative procedure. Here throughput is plotted against loss probability p for a given set of flows with RTTs in the range [0.07s, 0.7s]. The packet size is 576 bytes while the maximum congestion window is 256 packets. With a low drop probability, the throughput is high for all connections regardless of RTT. As the drop probability increases the throughput is throttled by the conditioner and all connections approach a stable throughput with fairness enforced regardless of RTT.

Figure 19: Loss-probability versus throughput for variable RTT connections

Chapter 4 A Critical Analysis of Loss Rate Estimation

In [19] an extensive research on the effectiveness of loss rate estimation in TCP rate control was performed by the analysis of three different techniques of loss rate estimation. The average loss interval method was implemented for EBM. However as discussed in [19], this technique has its limitations given certain network conditions that could exacerbate marking functions for EBM. The average loss interval technique averages over a number of loss intervals as opposed to the total number of lost packets divided by the total number of transmitted packets to determine the loss rate. The weight distribution follows an *s* shape [22] as illustrated below.

Figure 20: Weight distribution for average loss interval

The recent loss intervals receive similar weights (intervals 0-3) because they represent the current congestion state of the network. Then follows a smooth decline in weighting over the next four intervals (4-8). This is to ensure that older loss intervals and loss events that leave the loss history do not affect the loss estimate. All older intervals are weighted with zero [22].

This weight distribution ensures stability and response when the network is subjected to a sudden increase in congestion. It is slow in response to a sudden decrease in loss rate when there is a large interval between the current loss event and the last loss event. To this end *loss history discounting* (discussed below) is implemented to adapt the weights for loss rate estimation to this kind of scenario. This allows a more timely response to a sudden absence in congestion [19].

4.1 Loss history discounting

History discounting also known as proportional deweighting identifies a particularly long interval since the last drop and smoothly discounts the weights given to the older loss intervals [23]. Figure 21 [22] shows the weight distribution for a session in a TCP friendly rate control implementation. The width of the loss intervals corresponds to their size while the height is equivalent to their weights.

Figure 21: Loss history without discounting

The current loss interval (ppl_i above) is considered to be unrepresentative of the loss event rate when it exceeds a multiple m of the average size ppl_{i-1}^{avg} . In proportional deweighting a factor d_i is determined for the unrepresentative interval to mitigate the impact on the estimation of the loss rate.

$$d_i = \begin{cases} 1 & \text{for } ppl_i \leq m \cdot ppl_{i-1}^{avg} \\ \frac{m \cdot ppl_{i-1}^{avg}}{ppl_i} & \text{for } ppl_i > m \cdot ppl_{i-1}^{avg} \end{cases} \quad (11)$$

With a greater discrepancy between the size of the average loss interval and the current loss interval, the smaller is the deweighting factor and thus the weight assigned to the older loss intervals. This is given as:

$$ppl_{i-1}^{avg} = \frac{w_0 \cdot ppl_i + d_i \sum_{j=1}^{n-1} w_j \cdot ppl_{i-j}}{w_0 + d_i \sum_{j=1}^{n-1} w_j} \quad (12)$$

The weights after discounting of the loss history are given in Figure 22 below.

Figure 22: Loss history with discounted weights

While history discounting adapts to long intervals unrepresentative of the loss rate in the network, it also needs to adapt to a sudden increase in loss rate *after deweighting*. To this end, a lower bound can be imposed on the deweighting factor. This prevents a shift in all the weight to the current loss interval.

4.2 Implementation of Loss history discounting

The discounting mechanism that was added to EBM is given as follows:

If the current loss interval $s_0 > 2\hat{s}_{(i \geq 1)}$, then the current loss interval s_0 is considerably longer than the recent calculated average. The weights for the older intervals are discounted by using the discount factor:

$$d_i = \max(0.5, \frac{2\hat{s}_{(i \geq 1)}}{s_0}), \quad \text{for } i > 0, \quad (13)$$

$$d_0 = 1.$$

The lower bound of 0.5 ensures that past losses are not completely discarded in the calculation of the loss rate and improves adaptability in the face of a sudden increase in losses after deweighting is applied. The new calculation of the loss estimate with discounting is given as:

$$\hat{s} = \frac{\sum_{i=0}^{n-1} d_i w_{i+1} s_i}{\sum_{i=1}^n d_{i-1} w_i} \quad (14)$$

When a loss occurs, the old interval s_0 is shifted to s_1 . The corresponding discount factors are shifted so that once an interval is discounted it is not un-discounted. Without discounting, $d_i = 1$ for all values of i .

Chapter 5 Evaluation

The effectiveness of EBM in enforcing fairness in Diffserv AF was investigated by simulation using the simulation tool ns-2 [24]. The simulation network setup is given in Figure 23.

Figure 23: Simulation Network

There are ten sources (S1-S10) that send TCP traffic to the corresponding destinations (D1-D10). An EBM marker is inserted between each source and the network edge router E1. The bottleneck link spans three core devices, with a capacity of 10 Mbps and a latency of 10 ms between each device as shown above. The access links (between each source and the network edge E1 as well as E2 and each destinations) have a capacity of 10 Mbps and a latency of 10 ms. The core routers implement the AF PHB realized as

RIO with three drop precedences. The RIO setting in non-overlapped mode is given in Figure 24. The threshold parameter values are given in Table 2.

Figure 24: Table 2 represented as a multi-level RED graph

Table 2: Parameters for RIO with three-drop precedences

Parameters for	Green	Yellow	Red
Queue Length(Packets)	L	L	L
$\max_{th}$ (Packets)	0.875L	0.625L	0.3125L
$\min_{th}$ (Packets)	0.625L	0.3125L	0.025L
$\max_p$	0.02	0.05	0.1
w_q	0.002	0.002	0.002

A TCP agent at each source generates unlimited FTP bulk data. Based on network conditions a CIR and PIR common to all sources is specified. This topology presents two departures from the simulations in [3]. There are no UDP sources (background traffic) connected and the primary investigation is done solely with TCP sources.

The second departure concerns the definition of subscription level. In [3], the subscription level is based on the sum of two rates: Half of the bottleneck bandwidth is filled by the total sum of the CIRs of all assured sources (TCP) while the other half is the bit-rate of the background UDP sources. The latter is varied by either incrementing beyond the one-half the bottleneck bandwidth as a definition of *oversubscription* or decremented as a definition of *undersubscription*. It was reasoned that the true test of enforced fairness is better guaranteed by making the sum of all CIRs equal to the bottleneck bandwidth as a definition of capacity allocation with no surplus bandwidth.

With this as a starting point, the sum of all CIRs was increased over the bottleneck bandwidth as a definition of oversubscription while its decrease below the same was defined as undersubscription. The goodput (the total packets successfully received at the destinations as opposed to the total packets sent at the sources) was measured. The average goodput is calculated over one simulation run while the experiment was repeated 20 times for a confidence interval of 95 %. Three traffic conditioners are compared:

- Time-Sliding-Window Three-Color marker denoted as TSWTCM
- Traditional EBM without history discounting denoted as EBM
- EBM with history discounting denoted as EBM-HD

The TSWTCM [25] estimates the rate of competing connections and compares with the CIR and PIR in order to determine packet marking. The marking algorithm is given below in Figure 25 below.

```

If (estimated_rate) < CIR
    mark packet as GREEN
Else if( CIR < estimated_rate < PIR )
    mark packet as YELLOW with
     $P_{\text{yellow}} = (\text{estimated\_rate} - \text{CIR}) / \text{estimated\_rate}$ 
Else if(estimated_rate > PIR )
     $P_{\text{red}} = (\text{estimated\_rate} - \text{PIR}) / \text{estimated\_rate}$ 
     $P_{\text{yellow}} = (\text{PIR} - \text{CIR}) / \text{estimated\_rate}$ 
    Choose n from a uniform distribution
    mark packet RED if  $n \leq P_{\text{red}}$ 
    mark packet as YELLOW if  $n \leq P_{\text{yellow}}$ 

```

Figure 25: TSWTCM marking algorithm

The investigations are based on three basic traffic metrics that affect fairness: Packet Size, RTT and Target rate. The metric of fairness fi was computed according to [26] which gives the index of fairness among n competing flows, each with a bandwidth share of x_i , as:

$$fi = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (15)$$

The simulations and the results are described in the following sections.

5.1 Packet Size Experiment

This section explores the fairness among different TCP flows with disparate packet sizes. The scenario given in Figure 26 below was simulated. Packet sizes vary from 100 bytes (S1) to 1000 bytes (S10). Table 3 gives the RIO queue parameters. Each source is allocated a CIR of 1 Mbps and a PIR of 1.2 Mbps at the network edge. All core links have a capacity l . Capacity allocation (100% subscription) is defined when the core link, l , has a total rate equal to the total of all source CIRs (10 Mbps). Oversubscription (125%) was achieved by constricting the core link rate to 8 Mbps with an expected fair share of 0.8 Mbps for each source. Undersubscription (75%) was achieved by increasing the core link rate to 13.33 Mbps with an expected fair share of 1.33 Mbps per source. Table 3 gives the RIO parameters for the experiment.

Figure 26: Scenario for packet size experiment

Table 3: RIO parameters for packet size experiment

Parameters for	Green	Yellow	Red
Queue Length (Packets)	1000	1000	1000
$\max_{th}(\text{Packets})$	875	625	312
$\min_{th}(\text{Packets})$	625	312	25
$\max_p$	0.02	0.05	0.1
w_q	0.002	0.002	0.002

Simulation run was for 800 seconds and the goodput achieved by each source was measured with 95 % confidence interval. The results are shown in Figures 26 to 28 for TSWTCM, EBM and EBM-HD.

Figure 27: Goodput vs. packet size with TSWTCM marker

Figure 28: Goodput vs. packet size with EBM marker

Figure 29: Goodput vs. packet size with EBM-HD marker

From Figure 27, the disparity in achieved goodput with the TSWTCM marker increases with packet size. The marking function in TSWTCM estimates the rate and determines the marking accordingly [25]. Since there is no detection of packet sizes at the edge router given that TCP bandwidth is directly proportional to the packet size, the connections with the larger packet sizes hog the available bandwidth. EBM achieves a skewed fairness for flows with packet sizes varying from 400 bytes to 1 kilobyte. From Figure 29 above and using 100% subscription for explanation, it can be seen that EBM-HD is able to enforce fairness across the given range with an average goodput of 900 kbps for each flow at capacity allocation. The effectiveness of history discounting in enforcing fairness can be explained by the convergence of the loss rate estimate over time. Figures 30 and 31 respectively compare the loss rate estimate for EBM and EBM-

HD. In Figure 30, whereas loss rates converge to a given range, it is observed that EBM is not sensitive enough to calculate the loss rate for the first connection (100 bytes) with insufficient estimation for the next two connections that have packet sizes of 200 and 300 bytes respectively. History Discounting enhances this sensitivity and pushes the all loss rates into the appropriate range for the marking required to sustain the goodput across all sources.

The skewed fairness of EBM is further explained with the loss rate for packet sizes ranging from 600 bytes to 1 kilobyte. The loss rate estimate converges rapidly in this range and enables the correct marking. EBM-HD's convergence covers the range totally from 100 bytes to 1 kilobyte. Finally the total loss rate for all connections is illustrated in Figure 34. History discounting results in a higher loss rate because it monitors each flow more aggressively than traditional EBM.

Figure 30: Loss rate per flow (100-500 bytes) vs. time for EBM marker

Figure 31: Loss rate per flow (100-500 bytes) vs. time for EBM-HD marker

Figure 32: Loss rate per flow (600-1000 bytes) vs. time for EBM marker

Figure 33: Loss rate per flow (600-1000 bytes) vs. time for EBM-HD marker

Figure 34: Total Loss rate for all connections for EBM and EBM-HD

From Figure 29, an accepted definition of a service level agreement can be achieved with the implementation of EBM-HD for edge traffic conditioning. While the agreed CIRs are not met even at capacity allocation (1 Mbps specified but 0.9 Mbps achieved), SLAs in real world networks are in the form of agreed bounds. With this in mind an SLA for EBM-HD at capacity allocation will assure subscribers that the subscription rate will have a lower bound of 90% of the agreed CIR. This SLA holds even at oversubscription while with surplus bandwidth, users are assured at least their CIRs and PIRs with respect to the surplus bandwidth. Finally, a comparison of the fairness index of all three markers with subscription level is given in Figure 35.

Figure 35: Fairness Index vs. subscription level for all three markers

5.2 RTT Experiment

RTT investigations have been the subject of varied and diverse research. The range of testing is in the order of milliseconds with values chosen with respect to the latency existent in LANs (tens of milliseconds), to satellite networks (hundreds of milliseconds). For a complete investigation, a range of RTTs was chosen starting from 40 ms to 400 ms. Low-earth orbit satellite networks have RTTs up to 400 ms. In related work in [13], tests were conducted up to 200 ms and in [3], the range was up to 1500 ms. For the investigations conducted here, the range from 40 to 400 ms was reasoned to be wide enough for real life Diffserv networks that can span at most a satellite network and hence the upper bound of 400 ms.

Furthermore, a marking scheme defined as Rate-Based Marking (RBM) was also implemented in [3] but remained uninvestigated. This scheme estimates the throughput of each source using the TCP equation (Equation (3)), which is then compared with the CIR and PIR of the source to make marking decisions. The marking scheme is the same employed in the TSWTCM marker (page 50). This scheme is made efficient since the calculation of the loss rate is used in estimating the throughput according to the TCP equation. From investigations, it was found that the Rate Based Marker (RBM) provides better stability and fairness in terms of RTT over EBM while the implementation of history discounting with RBM further enhances this stability.

5.2.1 RTT Experiments with EBM

For the experiments in this section the three algorithms, TSWTCM, EBM and EBM-HD are compared. The scenario explored is given in Figure 36. Based on the range of RTTs used by other researchers as mentioned, for each algorithm, four ranges were experimented as given in Table 4.

Table 4: Ranges chosen for RTT experiment

RTT Range (ms)	Ratio (Shortest RTT to Longest RTT)
40-160	4
40-240	6
40-320	8
40-400	10

The simulation network is given in Figure 36 below for all four ranges of RTT experimented. Table 5 gives the RIO queue parameters. Each source has a CIR of 500 kbps and a PIR of 700 kbps allocated as the edge of the network. In Figure 36, the access links (from marker to core) have a capacity of 10 Mbps with their latencies varied to cover the range of RTTs for each experiment; i.e., 40 ms (S1) to the highest value (160 ms up to 400 ms at S10). All core links have a capacity l and a latency of 2 ms. Capacity allocation (100% subscription) is defined when the core link, l , has a total rate equal to the total of all source CIRs (5 Mbps). Oversubscription (125%) was achieved by

constricting the core link rate to 4 Mbps with an expected fair share of 400 kbps for each source. Undersubscription (75%) was achieved by increasing the core link rate to 6.67 Mbps with an expected fair share of 667 kbps per source. The results follow.

Figure 36: Scenario for RTT experiment

Table 5: RIO parameters for RTT experiment

Parameters for	Green	Yellow	Red
Queue Length (Packets)	800	800	800
$\max_{th}(\text{Packets})$	700	500	250
$\min_{th}(\text{Packets})$	500	250	20
$\max_p$	0.02	0.05	0.1
w_q	0.002	0.002	0.002

Figure 37: Goodput vs. RTT for TSWTCM

Figure 38: Goodput vs. RTT for EBM

Figure 39: Goodput vs. RTT for EBM-HD

Figure 37 shows again the inability of TSWTCM to maintain fairness with disparity in RTT given that it estimates the arrival rate of each connection and subsequently applies three-color marking without throttling the bandwidth for the low RTT connections. EBM and EBM-HD are able to achieve optimum fairness at oversubscription and undersubscription. At 100% subscription (with an expected goodput of 500 kbps for all sources), the first three connections with the lowest RTTs exceed the CIR of 500 kbps while the other seven connections maintain a bandwidth that is approximately 460 kbps. This allows an SLA definition of a lower bound of 92% of customer CIR. History Discounting provides no enhancement beyond the regular method of loss rate estimation. The loss rate graphs of Figures 40 and 41 cover the same range for both EBM and EBM-HD.

Figure 40: Loss rate per flow (40-93 ms) vs. time for EBM marker

Figure 41: Loss rate per flow (106-160 ms) vs. time for EBM marker

EBM calculates a higher loss rate for the low RTT connections (Figure 40) and a lower loss rate for the high RTT connections (Figure 41). While this sufficient to throttle the low RTT connections, a solution that would enhance the level of fairness which can still be investigated would be the estimation of the RTT to improve the convergence of loss rates and hence an improved level of fairness.

The subsequent experiments increased the ratio of RTTs using the values in Table 3. The performance of all three marking algorithms was compared by measurement of the level of assurance and fairness as the ratio increased. Figure 42 gives the goodput for TSWTCM with the final range of 40 ms to 400 ms while Figures 43 and 44 illustrate the performance of EBM and EBM-HD under the same conditions.

Figure 42: Goodput vs. RTT (40-400 ms) for TSWTCM marker

Figure 43: Goodput vs. RTT (40-400 ms) for EBM marker

Figure 44: Goodput vs. RTT (40-400 ms) for EBM-HD marker

With the longer range of 40 to 400 ms, the disparity in achieved goodput increases with the TSWTCM while both EBM and EBM-HD are still able to maintain fairness, which has reduced from the 92% of the subscriber CIR guaranteed with the shorter range to 90%. EBM-HD is only able to improve fairness slightly at capacity allocation. The performance of all three markers over the range of RTTs is given in Figures 45 and 46 in terms of fairness and assurance level respectively.

Figure 45: Fairness Index vs. RTT Range

Figure 46: Assurance Level vs. RTT Range

5.2.2 RTT Experiments with RBM

The experiments were repeated with the same network and over the same RTT range. The results using the marking algorithm employed by the TSWTCM are displayed below. The comparison is made between RBM and RBM-HD.

Figure 47: Goodput vs. RTT for RBM (40-160 ms)

From Figure 47, it can be seen that RBM provides optimum fairness at capacity allocation and offers a higher level of assurance up to 96% over the 92% offered by EBM with the same scenario. The convergence of loss rates for all connections with different RTTs as done in EBM when coupled with the estimation of arrival rate as done in the TSWTCM marker proves efficient in the guarantee of fairness. Figure 48 shows that discounting should be eschewed at capacity allocation for low RTT connections.

Figure 48: Goodput vs. RTT for RBM-HD (40-160 ms)

Subsequently, the range of RTT was increased as before and the performance measured. Figures 49 and 50 show the goodput achieved per connection when the RTTs span 40 to 400 ms for RBM and RBM-HD respectively. The overall performance of all markers is compared in Figures 51 and 52 in terms of Assurance Level and Fairness. The best performance in terms of providing fairness with respect to RTT is RBM-HD. This needs further research as it fails to provide fairness with respect to packet size.

Figure 49: Goodput vs. RTT for RBM (40-400 ms)

Figure 50: Goodput vs. RTT for RBM-HD

Figure 51: Assurance Level with increasing RTT

Figure 52: Fairness Index with increasing RTT

5.3 Target Rate Experiments

Different sources subscribed to Diffserv sometimes differ only in the size of their individual target rates. In [27] it was discovered that connections with a larger target rate hog the available bandwidth to the detriment of connections with a lower target rate. To explore fairness according to target rates, the scenario given in Figure 53 below was simulated. Target rates vary from 500 kbps (S1) to 2300 kbps (S10). Table 6 gives the RIO queue parameters. Each source has a PIR that is twice the CIR. All core links have a capacity of 14 Mbps, which is the sum of the CIRs of all connections. Testing was done only at capacity allocation (100% subscription).

Figure 53: Network for target rate investigation

Table 6: RIO parameters for target rate investigation.

Parameters for	Green	Yellow	Red
Queue Length (Packets)	1000	1000	1000
$\max_{th}(\text{Packets})$	875	625	312
$\min_{th}(\text{Packets})$	625	312	25
$\max_p$	0.02	0.05	0.1
w_q	0.002	0.002	0.002

Figure 54 below shows the goodput achieved by each source. EBM offers the required protection based on the target loss rates determined according to both the PIR and CIR. EBM with history discounting however works to overprotect the connections with low CIRs with the effect that the higher CIRs are not met. Discounting is therefore discouraged in ensuring fairness according to target rates.

Figure 54: Goodput vs. CIR for EBM, EBM-HD and TSWTCM

Chapter 6 Conclusion and future work

History discounting has demonstrated its ability to enhance the determination of loss rates for the appropriate traffic marking for Diffserv edge conditioning. The contributions of history discounting can be listed as follows:

- When connections that share a common network differ in packet size, history discounting enhances the sensitivity of EBM to loss rates for low packet sizes in ensuring an acceptable definition of fairness
- History discounting only slightly enhances fairness when connections differ in terms of RTT
- History Discounting promotes true fairness when implemented with the RBM that measures subscriber arrival rates to determine marking. The measurement of the loss rates of the individual connections and the convergence of all loss rates promotes this fairness and stabilizes achieved goodput at the CIR

EBM-HD however still has its drawbacks. The overprotection of connections with low target rates over subscribers with higher target rates still needs to be investigated. The oscillations in throughput calculation when used with the TSWTCM need to stabilize. History discounting however is one of several methods that enhance the computation of the loss rate estimate. While it was chosen over the other methods

mentioned in [23], their performance in a Diffserv network can still be explored. While EBM and EBM-HD can enforce fairness according to packet size, RTT and target rate, the algorithm cannot detect the number of microflows within a flow and throttle the bandwidth accordingly. This could be implemented as a separate module to be integrated into the EBM algorithm. The protection of short-lived flows is also the subject of future research.

Appendix A

Steady State rate Equation for a TCP sender

The long-term steady state rate B of a TCP connection [18] with respect to the steady state loss rate or indication p is given as:

$$B(p) = M \frac{\frac{1-p}{p} + W(p) + \frac{Q(p, w)}{1-p}}{RTT \left(\frac{b}{2} W(p) + 1 \right) + \frac{Q(p, w) F(p) T_o}{1-p}} \quad \text{If } W(p) < W_{max}$$

$$B(p) = M \frac{\frac{1-p}{p} + W_{max} + \frac{Q(p, w_{max})}{1-p}}{RTT \left(\frac{b}{8} W_{max} + \frac{1-p}{p W_{max}} + 2 \right) + \frac{Q(p, w_{max}) F(p) T_o}{1-p}} \quad \text{Otherwise}$$

The parameters are as follows:

M is the maximum segments size in bits

W_{max} is the maximum congestion window

T_o is a Timeout

b is the number of packets acknowledged by an ACK

p is the loss probability

RTT is the Round-Trip-Time of the connection

$W(p)$ is the average window size

$$W(p) = \frac{2+b}{3b} + \sqrt{\left(\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b} \right)^2 \right)}$$

$Q(p, w)$ is the probability that the loss in a congestion window is a Timeout.

$$Q(p, w) = \min \left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{(w-3)}))}{1-(1-p)^w} \right)$$

Here w , for convenience denotes the current congestion window $W(p)$

$Q(p, w_{max})$ is the probability that the loss when the congestion window is constrained to the maximum window W_{max} is due to a Timeout

$$Q(p, w_{max}) = \min \left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{(w_{max}-3)}))}{1-(1-p)^{w_{max}}} \right)$$

Here again, w_{max} , for convenience denotes the maximum congestion window W_{max}

$F(p)$ is the average duration of a timeout sequence

$$F(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6$$

There is no closed form solution for this equation; therefore, iterations are employed to compute the values of p that correspond to the target throughput rates CIR and PIR respectively. A first calculation to determine the success of iteration [3] is done as given below.

Each Periodic interval

1. Use current estimate of RTT and Timeout respectively
 2. Use estimate of current loss probability, p_{curr}
 3. Use Value of maximum TCP Window, W_{max}
 4. Calculate maximum achievable throughput which occurs when the loss is nil at $p = 0$ i.e. $r_{max} = T(0, RTT, T_0, W_{max})$
 5. if ($CIR > r_{max}$)
 6. Exit with error "Not able to achieve CIR and PIR"
 7. if($PIR > r_{max}$)
 8. Warning "Not able to achieve PIR"
 9. end if
 10. end if
 11. Calculate $p_{cir} = T_p^{-1}(CIR, RTT, T_0, W_{max})$
 12. Calculate $p_{pir} = T_p^{-1}(PIR, RTT, T_0, W_{max})$
 13. mark packets
-

The complete iterative procedure equivalent to the inversion of Equation (3) is given below.

```

 $p = T_p^{-1}(BW, RTT, T_0, W_{\max}) \{$ 

// BW is CIR or PIR as required, set accuracy used to be 10, 100, ; set resolution used to be 0.00001
// set iteration limit set at 100000

Begin {
count = 0;

Calculate current loss probability  $p$  from using average loss rate interval in Equation (6)
Calculate Bandwidth  $B = \text{Calculate\_BW}(p, W_{\max}, b, RTT, T_0, MSS, \text{type})$ ; //from Equation (3)

    if( $B > BW$ ) //if the estimated bandwidth is greater than the CIR (or PIR as required)

        diff =  $B - BW$ ;
        else
        diff =  $BW - B$ ;
        if(diff <= accuracy) // reach a good value
        break;

    if( $B < BW$ ) {
         $p = p - \text{resolution}$ ;
        if( $p <= 0$ )
         $p = 0.0$ ;
        break;}
    }else if( $B > BW$ )
         $p = p + \text{resolution}$ ;
        if( $p >= 1.0$ ) {
         $p = 1.0$ ;
        break;}
        else if ( $B == BW$ )
        break;
        count++; //
        if(count >= iteration Limit ||  $p == 0.0$  ||  $p == 1.0$ ) {
         $p = p0$ ; //Iteration not complete at this accuracy, resolution
        accuracy = accuracy * 2.0;
        if(number++ < 3)
        goto begin; }else
        puts("success");
        return  $p$ ; }

```


Appendix B

Markov Processes

A random process which is also referred to as a stochastic process is defined formally as a set of random variables, $\{X_t\}$ that are indexed with a parameter t . the index is usually interpreted as time. While in general the future behavior of a system depends on its past, there exists a set of stochastic processes that exhibit a limited but quite useful form of state dependency [28]. These processes satisfy the Markov property that states that:

Given the current state of a process X_t , when the process is observed at a given moment, t , the distribution of any future state, X_y , does not depend on the past history, $\{X_u: u < t\}$

The set of values taken by the random variables X_t is referred to as the state space. Of interest are Markov processes that have a discrete state space. If in addition the time parameter is discrete, the process is referred to as a Markov chain and when it occurs in continuous time, it is termed a Markov process.

Given a Markov chain $X = \{X_n : n = 0, 1, 2, \dots\}$ whose state space $\{0, 1, \dots\}$, may be infinite or finite, the Markov property says, with regards to this case, that given the state of X at time n , its state at time $n+1$ is independent of the states at times $0, 1, \dots, n-1$:

$$P(X_{n+1} = j | X_0, X_1, \dots, X_n) = P(X_{n+1} = j | X_n)$$

Thus, the evolution of the Markov chain is completely described by the one-step transition probabilities, $q_{i,j}(n)$, that the chain moves to state j at time $n+1$, given that it is in state i at time n :

$$q_{i,j}(n) = P(X_{n+1} = j | X_n = i) ; i, j, n = 0, 1, \dots$$

As an example, a machine can be in one of two possible states: broken or operational (0 or 1 respectively). If it is broken at time n , it will be broken or operational at time $n+1$ with the probabilities 0.2 and 0.8 respectively. If it is operational at time n , it will be broken at time $n+1$ with probabilities 0.1 and 0.9 respectively [28]. This behavior satisfies the Markov property as illustrated in Figure B.1 below and given in the transition probability matrix Q below.

Figure B.1: States of an unreliable machine

$$Q = \begin{bmatrix} 0.2 & 0.8 \\ 0.1 & 0.9 \end{bmatrix}$$

One step transition probability for the unreliable machine

The Markov process in view of TCP connections can be applied as was done in [28] to a discrete communication buffer with a finite size N whose behavior is explained. During the n th time slot, either a new packet arrives in the buffer, with a probability α or one packet is transmitted with probability β , or there is no change in the buffer's state. The buffer can be modeled as a Markov chain which is in state i when there are i messages in the buffer with a state space $\{0,1,\dots,N\}$. From state i , there are transitions to state $i-1$ (with the exception of the limiting state $i = 0$), i , and state $i+1$ (with the exception of the limiting state when $i = N$) with probabilities β , $1-\alpha-\beta$ and α respectively. The state diagram is given in Figure B.2.

Figure B.2: One-step transition diagram for a buffer of capacity N

When the Markov chain above is indexed in continuous time, it becomes a Markov process. This model thus describes the TCP process in slow start up to fast recovery and since the window is incremented again starting from 1, it is described, as been a regenerative Markov process.

References

- [1] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", RFC 2638, July 1999
- [2] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding Per-Hop-Behavior Group", RFC 2597, June 1999
- [3] El-Gendy, M.A., Shin, K.G. "Equation Based Marking for Assured Forwarding Services", IEEE INFOCOM, Vol. 2, June 2002
- [4] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPV4 and IPV6", RFC 2474, December 1998
- [5] http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/difse_wp.pdf
- [6] S. Blake et al., "An Architecture for Differentiated Services", RFC 2475, December 1998
- [7] Floyd, S.; Jacobson, V., "Random early detection gateways for congestion avoidance", IEEE/ACM Transactions on Networking, Vol. 2 No. 4, Aug. 1993
- [8] Clark, D.D.; Wenjia Fang, "Explicit allocation of best-effort packet delivery service", IEEE/ACM Transactions on Networking, Vol. 6 No. 4, Aug. 1998
- [9] K. Thompson, G. Miller and M. Wilder, "Wide-area Internet traffic patterns and characteristics," IEEE Network, Vol. 11, Nov./Dec. 1997
- [10] W. Feng, D. Kandlur, D. Saha and K. Shin, "Adaptive packet marking for maintaining end-to-end throughput in a differentiated services Internet," IEEE/ACM Transactions on Networking, Vol. 7, No. 5, October 1999.
- [11] B. Nandy, N. Seddigh, P. Piedad and J. Ethridge, "Intelligent traffic conditioners for assured forwarding based on differentiated services networks," Proc of High Performance Networking 2000 Conference, Paris, France, May 2000
- [12] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," Computer Communications Review, ACM SIGCOMM, Vol. 27, No. 3, July 1997
- [13] A. Habib, B. Bhargava, S. Fahmy, "A round trip time and timeout aware traffic conditioner for differentiated services networks," IEEE international conference on communications ICC 2002, Vol. 2, May 2002
- [14] H. Kim, "A fair marker" Internet-draft, work in progress, draft-kim-fairmarker-diffserv-00.txt, IETF, April 1999

- [15] A. Feroz, A. Rao and S Kalyanaraman, "A TCP-friendly traffic marker for IP differentiated services," Proc. Of IWQoS, 2000
- [16] W Stevens, "TCP Slow start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC 2001, January 1997
- [17] Jon Postel, "Transmission Control Protocol", RFC 793, September 1981.
- [18] Padhye, J., Firoiu, V., Towsley, D.F., Kurose, J.F., "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation", IEEE/ACM transactions on Networking, Vol. 6 Issue 2, April 2000
- [19] S. Floyd, J. Padhye and J. Widmer, "Equation-based control for unicast applications", Proc. Of ACM SIGCOMM '00, 2000.
- [20] V. Firoiu and M. Borden, "A study of active queue management for congestion control," Proc of IEEE INFOCOM 2000.
- [21] Kendall E. Atkinson, An introduction to numerical analysis, John Wiley and sons, New York, 2nd ed., 1989.
- [22] Widmer, J. "Equation Based Congestion Control" Diploma Thesis, Department of Mathematics and Computer Science, University of Mannheim, February 2000.
- [23] S. Floyd, J. Padhye and J. Widmer, "Equation-based control for unicast applications: the Extended Version", International Computer Science Institute technical report, tr-00-03, March 2000.
- [24] Network Simulator, ns-2, University of California at Berkeley, version 2.1b6, www.isi.edu/nsnam/ns with the EBM implementation suite by El-Gendy available at http://kabru.eecs.umich.edu/qos_network/diffserv/EBM/ebm.html
- [25] W. Fang, N. Seddigh and B. Nandy, "A Time Sliding Window three Color Marker (TSWTCM)," RFC 2859, June 2000
- [26] Raj Jain, "The Art of Computer Systems Performance Analysis", John Wiley and Sons Inc., 1991
- [27] Seddigh, N., Nandy, B., Pineda, P., "Bandwidth assurance issues for TCP flows in a differentiated services network" IEEE Globecom vol. 3, 1999
- [28] Isi Mittrani, "Probabilistic Modeling", Cambridge University Press, 1998