Congestion Control for Adaptive Satellite Communication Systems with Intelligent Systems

by

Banupriya Vallamsundar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2007

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Banupriya Vallamsundar

Abstract

With the advent of life critical and real-time services such as remote operations over satellite, e-health etc, providing the guaranteed minimum level of services at every ground terminal of the satellite communication system has gained utmost priority. Ground terminals and the hub are not equipped with the required intelligence to predict and react to inclement and dynamic weather conditions on its own. The focus of this thesis is to develop intelligent algorithms that would aid in adaptive management of the quality of service at the ground terminal and the gateway level. This is done to adapt both the ground terminal and gateway to changing weather conditions and to attempt to maintain a steady throughput level and Quality of Service (QoS) requirements on queue delay, jitter, and probability of loss of packets.

The existing satellite system employs the First-In-First-Out routing algorithm to control congestion in their networks. This mechanism is not equipped with adequate ability to contend with changing link capacities, a common result due to bad weather and faults and to provide different levels of prioritized service to the customers that satisfies QoS requirements. This research proposes to use the reported strength of fuzzy logic in controlling highly non-linear and complex system such as the satellite communication network. The proposed fuzzy based model when integrated into the satellite gateway provides the needed robustness to the ground terminals to comprehend with varying levels of traffic and dynamic impacts of weather.

ACKNOWLEDGEMENTS

I would like to offer my first prayers and thanks to the Almighty for having given me the strength and courage to study and survive away from home and family for two long years.

My heartfelt thanks to my second Gods, my ever loving parents, Prof Vallam Sundar and Poornima Sundar who have extended their unfailing love and support to me all through the course of this study. They have been there with me through all the moments of jubilation and frustration in equal amounts and no measure of hard work could ever repay the love they have given me. I am truly blessed to have such wonderful parents.

I owe very sincere thanks to my advisor Prof Kumaraswamy Ponnambalam for his patience in allowing me to explore different avenues of research and for having shared with me his knowledge and time so generously.

I am extremely grateful to my twin sister, Suriyapriya Vallamsundar for having been there for me always during all those best and worst moments one can share in an alien land. She made my life simpler just by being there in it and filled each day of my life with lots of joy.

Very special thanks to the two best friends in my life, Mahesh Sukumar and Farah Ahmed who have filled up my life with so much of love and happiness and continue to work their magic on my everyday life. I could never have survived grad school with their unfailing support. Thanks also to my other best friend, Charanya Chandramouli for all those inspiring and motivating discussions.

Thanks are in order for numerous friends I made in Canada during the course of my study. I offer all my thanks to Bharathi Sambasivan who has been a great friend and guided me through all the tough times. I will always treasure the good memories we shared together. Thanks also to Betty Hou, Jing Ma, Subam Basuthkar, Kamal Harb and Sue Ann Soulliere for having made my life so much nicer and more enjoyable.

I would like to take this opportunity to acknowledge the help extended by my colleagues and friends, Cornelius Rabsch, Jiesheng Zhu, Brian Cheng, Hongqing Zeng, Masoud Mahootchi and Arun Veeramany for all the help they extended and the numerous enriching research related I shared with them. Thanks all!

Dedicated to the loving memory of my grandmother,

Mrs. Sarojini Venkataswamy

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Networks
AQM	Active Queue Management
CBR	Constant Bit Rate Applications
COA	Centre of Area
CWND	Congestion Window
DT	Drop Tail
ECN	Explicit Congestion Notification
FIFO	First In First Out
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
FRED	Fuzzy Random Early Detection
FTP	File Transfer Protocol
GA	Genetic Algorithm
ICMP	Internet Control Message Protocol
IEFT	Internet Engineering Task Force
ISO	International Organization for Standardization
ITU	International Telecommunications Unit
LAN	Local Area Network
MIMO	Multiple Input Multiple Output
ns	Network Simulator
Otcl	Object Oriented Extension on Tcl
PI	Proportional-Integral Controllers
QoS	Quality of Service
RED	Random Early Detection
REM	Random Early Marking
RFC	Request for Comments
RTT	Round Trip Time
RVSP	Resource Reservation Protocol
SLA	Service Level Agreement
SRED	Stabilized RED
TCP/IP	Transport Control Protocol/Internet Protocol
UDP	User Datagram Protocol
VoIP	Voice over IP

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Satellite networks and transmissions play a significant and major role in numerous fields. They find their application in fields of computer communications, telephone communications, television broadcasting, transportation, banking and finance, Supervisory Control and Data Acquisition (SCADA) to name a few. Recent trends in telecommunications indicate that four major growth market/service areas are: messaging and navigation services (wireless and satellite), mobility services (wireless and satellite), video delivery services (cable and satellite), and interactive multimedia services (fiber/cable, satellite). Coupling the characteristics of satellites such as multicast capabilities and wide area coverage with terrestrial networks opens up vast market opportunities. Due to its extensive geographic reach, satellite communications may be the only solution in those areas where terrestrial high-bandwidth communications infrastructure is impractical or non-existent. Also satellite components have been predicted to play an important role in the third generation wireless multimedia services as a result of their wide coverage.

The two major types of satellite systems are the Geo Stationary (GEO) and the Leo Stationary (LEO) satellites. GEO satellites orbit the earth at an altitude of about 36,000 km while the LEO satellites travel at altitudes between 700 and 2,000 km above the earth. GEO satellite system is studied in this work. One of the major drawbacks with GEO satellites is the long delay associated with it and this can have serious impact on the

Quality of Service (QoS) to the end-users. Also, atmospheric conditions can affect the data transmission done through the satellite. Bandwidths associated with satellite systems is a scare resource compared to terrestrial systems and also with the weather conditions being harsher on these systems, resource allocation assumes a significant position.

In the satellite communications family, a new concept based on broad-band satellite networks have been proposed (Jamalipour, 2001). In the last few years, the broad-band satellite network has gained tremendous research interest (Taleb, 2005). The broad-band satellite network is IP-based and provides a ubiquitous means of communications for multimedia and high data rate Internet-based applications, such as audio and video applications.

The major limitation of satellite communications is the high propagation delay, due to their altitude, and the SNR (signal to noise ratio), which can dramatically decrease with adverse atmospheric conditions. The overall performance of satellite communications might be severely degraded due to dynamic weather conditions. Rain channel fading due to precipitation is one of the many conditions that lead to channel degradation. The channel fading most often leads to network congestion. Congestion control is a significant issue in satellite communications networks due to the inherent large channel latency, or high bandwidth-delay product. Consequently, similar to terrestrial IP networks, the network congestion control is critical in broadband satellite-based IP networks, for committing to the Service-Level-Agreement (SLA) with regard to Quality-Of-Service (QoS). Some of the satellite characteristics also can be utilized to arrive at the satellite specific Quality of Service (QoS) classes. The challenge of this research work is to adaptively manage the QoS in ground terminals as their output performance can change without adequate warning due to dynamic weather conditions.

1.2 MOTIVATION

The prime objective of this research is to significantly increase the performance and efficiency of satellite communication under dynamically changing atmospheric conditions (such as rain, snow, sleet, hail etc) which affects the physical channel between the satellite and the ground terminals. Deterioration of the physical channel leads to extremely limited link capacity and reduces the quality of services that is promised.

The current satellite systems employ Drop Tail (DT) routing algorithms in their gateways. These discard arriving packets if the buffer at the gateway is full and follows First-In-First-Out (FIFO) scheduling. Recently, Active Queue Management (AQM) has been proposed to support QoS. AQM is a packet dropping/marking mechanism for router queue management and congestion control. It targets to reduce the average queue length and thereby decrease the packet latency, while reducing the packet loss to ensure efficient network resource utilization. RED algorithm is one of the most popular AQM mechanisms, deployed widely in today's IP routers. Contrary to the DT algorithm, RED algorithms start their packet drop earlier to notify traffic sources of incipient stages of congestion in the gateway. RED has been designed to substitute the DT technique of buffer management. However, RED mechanism is often disabled by the users in the real world due to their drawbacks such as the sensitivity to network configuration parameters and states, queue fluctuation, low throughput. Also tuning of RED parameters has been an inexact science for long and is often characterized by non-linear and chaotic behavior.

This inherent uncertainty associated with tuning can also be considered as fuzzy information with respect to controlling the network congestion. This research proposes to use the reported strength of fuzzy logic to control such kind of non-linear and complex control tuning of the RED. The design of fuzzy logic control system is based on fuzzy logic controlled RED queue, wherein parameters of the RED algorithm are tuned "intelligently" to provide effective traffic/congestion control in the presence of dynamic network state changes.

In this study, a novel fuzzy-based RED (FRED) algorithm is proposed that works on the limitations introduced by the Drop Tail and RED models. Three different priority levels are set up within the FRED model to provide different levels of services to the customers. This model can be implemented as system based intelligent algorithms in satellite ground terminals to contend with the inherent impacts of dynamic weather changes on the system. In a broad sense, these algorithms would use dynamic atmospheric conditions and traffic sources as input to engineer a real-time adaptive QoS that provides the right service levels to the customers.

1.3 CONTRIBUTION OF THE THESIS

There is an increasing worldwide demand for satellite communication primarily due to its extensive geographic reach and wide deployment. Also communications involving satellite networks require relatively less time to develop a communications infrastructure at low costs. These attractive features of satellite networks are opening new markets and creating business opportunities for satellite service providers. The service providers are constantly under pressure to keep improving the quality of services they provide and hence are always on the look out for new and advanced technologies that aid them in achieving this.

Applying intelligent systems in the field of internet satellite communication field under dynamic weather conditions is a novel idea which has never been attempted before. Intelligent solutions will enable monitoring and satisfying the service promised to the customers. Generally Service Level Agreements drawn up between the service providers and customers have strict guidelines with promised measures of packet loss, latency, jitter and bandwidth guarantees. This research proposal attempts to satisfy the SLAs under dynamic weather conditions through intelligent tuning of the RED parameters.

This could lead to service providers opening new application areas with increased and predictable performance to customers leading to customer satisfaction and expanded commercial opportunities.

With additional attractive features such as bandwidth guarantees and predictable performance, new intelligent system application such as remote control operations (like e-health) can now be applied over satellite networks. Coupling extensive geographic reach of the satellite networks to the above features, services such as e-health can be used to reach places all around the globe.

This proposed intelligent system development offers the potential to improve the level of services to the end-users. Also, the deployment of the intelligent system would allow the creation of satellite and terrestrial hybrid networks allowing seamless services to the end-user resulting in an even bigger demand for satellite service. This will result in new business opportunities for the satellite owners and the satellite service providers as well.

1.4 STRUCTURE OF THE THESIS

This thesis is organized into five chapters. Chapter 2 presents the background on QoS in satellite communication systems and a brief review on the techniques to achieve QoS. It is followed by a review of the existing congestion control techniques in TCP/IP. An introduction to AQM followed by different types of AQM mechanisms for congestion avoidance is discussed next. The chapter also discusses RED algorithm, the main buffer management mechanism under analysis in the present study, in detail. Chapter 3 introduces the problem space and follows it with a detailed discussion on the problem methodology with emphasis given to the various components of the proposed fuzzy logic controller. This is followed by a discussion on the evaluation environment under which modeling and simulation of the developed fuzzy model was performed. Chapter 4 present comparisons of performances of Drop Tail and RED algorithms discussed in Chapter 2. It then presents the results obtained by the fuzzy RED algorithm using the analysis model discussed in Chapter 3. Various simulated scenarios are presented at the end of the chapter to establish strength of the proposed fuzzy RED model. Summary and future work is presented in Chapter 5.

CHAPTER 2

THEORITICAL BACKGROUND

This chapter presents reviews of some of the theoretical background of the problem. It is divided into three main sections. Quality of Service in Satellite communications is reviewed in Section 2.1.

TCP congestion control mechanism and different active queue management strategies are briefly discussed in Section 2.2. Also, a detailed discussion on RED algorithm, the prime buffer management scheme under our study is provided. Section 2.3 introduces the different types of soft computing tools with a detailed description of various components of Fuzzy Logic Control system.

2.1 SATELLITE COMMUNICATION SYSTEMS

2.1.1 INTRODUCTION TO SATELLITE COMMUNICATIONS

Qualities of wide area coverage, high bandwidth availabilities coupled with multicast capabilities make satellite communication systems an attractive service. Satellite communications play an important role in situations where terrestrial communication requiring high bandwidth services are non-existent and almost impossible to provide. Due to its vast coverage area, satellite systems are expected to play a significant role in future third-generation wireless and multimedia services. Quality of Service is one of the most important performance measures of any communication service. The following section provides a brief analysis on the quality of service measures and various techniques to achieve the same.

2.1.2 QUALITY OF SERVICE IN SATELLITE SYSTEMS

The most common feature in modern communication systems has been reliable and safe message transfer with error control and notification of non-delivery of the message being incorporated in it. However, in recent times, the ability to perceive the quality and timeliness of data arrival has been given much importance in complex communication systems. To provide predictable and reliable services, service providers of communication services negotiate Service Level Agreement (SLAs) with the customers. The purpose of SLAs is to identify the shared objectives between the service providers and the customers (Leopoldi, 2002)

Service Provider – "We agree to provide you this particular level of service based on agreed-to set of guidelines"

Customer – "We agree to abide by your guidelines in anticipation that you provide us this agreed level of service"

Service providers define the "Quality of Service" (QoS) a customer can expect in the Service Level Agreements. The most important agreements between the provider of service and the user of that service include (Leopoldi, 2002)

1. Availability of the promised level of service to the user
2. Performance measures of various components of the user's workload
3. Bounds of guaranteed performance and availability
4. Cost of the service provided to the customer

More specific parts of an SLA are service level specification (SLS), traffic conditioning agreement (TCA), and traffic conditioning specification (TCS).

The term “service” in the telecommunications context seems to be obvious. A service in an IP based environment is defined by International Telecommunications Unit (ITU) as “a service provided by the service plan to an end user (e.g., a host [end system] or a network element) and which utilizes the IP transfer capabilities and associated control and management functions, for delivery of the user information specified by the service level agreements” (ITU Recommendations, 2001).

The meaning of “quality” is very broad. In telecommunications it is commonly used in assessing whether the service satisfies the user’s expectations. However, the evaluation depends on various criteria related to the party rating the service. Customers assess it on the basis of a personal impression and in comparison to their expectations, while an engineer expresses quality in terms of technical parameters. This discrepancy may sometimes lead to misunderstandings between the service provider and the customer. In this study, we refer to the ISO definition of quality as “the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs” (ISO 8402, 1994).

“Quality of Service (QoS) is defined as “A set of service requirements to be met by the network while transporting a flow” by Hardy (2001). QoS in packet networks is expressed by at least the following set of parameters that are meaningful for most IP-based services:

1. *Bit rate* of transferring user data available for the service or target throughput that may be achieved.

2. *Queuing delay* experienced by packets while passing through the network. It may be considered either in an end-to-end relation or with regard to a particular network element. Whenever the networks are free of congestion, the transmission capability of the network is greater than the amount of traffic flow in the channel. In cases such as this, the latency or queuing delay is very low. Once congestion starts to build up in the channel, the packets are forced to wait and queues are created waiting to be serviced. In these cases, latency can be very high and pose a major problem to the dynamics of the channel. Since the network operation is fast, latency is usually measured in milliseconds (ms). Non-congested channels might have a latency of 50ms. Congested channels have numerous packets waiting to be transmitted and this in turn reduces the transmission capacity of the channel which makes latency values high such as 500ms, 1000ms and sometimes even as high as 5000ms.
3. *Jitter variations* in the IP packet transfer delay. Variance in queue delay corresponds to jitter of flows. When the latency values are more or less constant, design engineering can re-design the applications that are latency sensitive and take into consideration the high latency values attempting to achieve an improved performance of the system. On the other hand, if variation of latency (jitter) is uncertain and high, the problem gets complicated and this makes it almost impossible for re-design of the algorithm to fix the problem of latency. Problem of high values of jitter is worse than large values of latency.
4. *Packet Losses* occur when there is congestion in the transmission channel and these channels are forced to drop the packets since it is almost impossible to

transfer them any forward to their destination nodes. TCP, the protocol studied in our work deals with the problem by asking the sender to re-transmit their lost packets again. Other protocols, such as the UDP are less capable than the TCP and are not equipped with enough intelligence to deal with situations such as lost data. TCP Loss Rate is defined as the ratio of total number of TCP packets that are dropped to the total number of TCP packets that arrive at the same output destination node.

As outlined in Hardy (2000), "QoS requirements define the non-functional characteristics of a system that affect the perceived quality of results".

2.1.3 SURVEY ON EXISTING QoS TECHNIQUES

Both end-to-end QoS and hop-by-hop QoS have been proposed and studied in the research community and industry. The end-to-end QoS is more fundamental and hop-by-hop QoS is only used to indirectly serve the end-to-end QoS. In our study, we focus on the end-to-end QoS. There are three major categories of approaches to ensure end-to-end QoS namely;

1. Integrated Services (IntServ)
2. Differentiated Services (DiffServ)
3. Active Queue Management (AQM)

The first two are based on flows distinction assumption. AQM can fairly regulate network traffic without any flows discrimination. This is particularly prevalent in best effort networks where all streams have the same network access right.

2.1.3.1 INTERGRATED/GUARANTEED SERVICES

Early work used the "IntServ" model to reserve network resources to meet QoS requirements defined in SLAs. In this model, applications used the Resource Reservation Protocol (RSVP) to request and reserve resources through a network. In a broadband network typical of a larger service provider, core routers would be required to accept, maintain, and tear down thousands or possibly tens of thousands of reservations. Thus, a critical problem with IntServ is that many states must be stored in each router. As a result, IntServ works on a small-scale, but as one scales up to a system the size of the Internet, it is difficult to keep track of all of the reservations. It was widely believed that this approach would not scale with the growth of the Internet, and in any event was antithetical to the notion of designing networks so that core routers do little more than simply switch packets at the highest possible rates. As a result, IntServ is not very popular.

2.1.3.2 DIFFERENTIATED SERVICES

In the DiffServ or differentiated services model, packets are marked according to the type of service they need. In response to these markings, routers and switches use various queuing strategies to tailor performance to requirements. Routers supporting DiffServ use multiple queues for packets awaiting transmission from bandwidth constrained (e.g., wide area) interfaces. Router vendors provide different capabilities for configuring this behavior, to include the number of queues supported, the relative priorities of queues, and bandwidth reserved for each queue. In practice, when a packet must be forwarded from an almost packed queue , packets requiring low jitter such as VoIP are given priority over packets in other queues. Typically, some bandwidth is allocated by default to network control packets (e.g., ICMP and routing protocols), while best effort traffic might simply be given whatever bandwidth is left over.

One of the main limitations of the DiffServ model is that the details of how individual routers deal with the type of service field is somewhat arbitrary, and it is difficult to predict end-to-end behavior. This is further complicated if a packet crosses two or more DiffServ clouds before reaching its destination. From a commercial viewpoint, this is a major flaw, as it means that it is impossible to sell different classes of end-to-end connectivity to end users, as one provider's highest-priority packet may be another's low-priority. Internet operators could fix this, by enforcing standardized policies across networks, but are not keen on adding new levels of complexity to their already complex peering agreements. Diffserv operation only works if the boundary hosts honor the policy agreed upon. A host can always tag its own traffic with a higher precedence, even though the traffic doesn't qualify to be handled with that importance.

2.1.3.3 ACTIVE QUEUE MANAGEMENT

Active Queue Management (AQM) is a technique of preventing congestion in packet-switched networks. It attempts to achieve high link utilization with a low queuing delay. Random Early Detection (RED), discussed in detail in the next section is one of the most well-known AQMs and many variants have been proposed since RED was developed. RED is a queue management and a congestion avoidance algorithm. In the traditional tail drop algorithm, a router or other network component buffers as many packets as it can, and simply drops the ones it can't buffer. If buffers are constantly full, network is congested. Tail drop distributes buffer space unfairly among traffic flows. Tail drop can also lead to TCP global synchronization as all TCP connections "hold back" simultaneously, and then step forward simultaneously, networks become under-utilized and flooded by turns. RED addresses these issues. It monitors the average queue size and drops packets based on statistical probabilities. If the buffer is almost empty, all incoming packets are accepted. As the queue grows, the probability for dropping an incoming packet grows too. When the buffer is full, the probability has reached 1 and all incoming packets are dropped. RED is considered more fair than tail drop. The more a host transmits, the more likely it is that its packets are dropped. Early detection helps avoid global synchronization.

2.2 CONGESTION CONTROL MECHANISMS

2.2.1 TCP CONGESTION AVOIDANCE MECHANISMS

Around 90% of today's Internet traffic transmission is supported by the Transmission Control Protocol (TCP) and a thorough understanding of the TCP mechanism is vital to design congestion avoidance mechanisms and to ensure end-to-end QoS. TCP has numerous objectives, a few of which is outlined below;

- Maintain congestion avoidance in the network channel
- Adaptation of transmission rate of packets to the available link capacity
- Create reliable connection by retransmission of lost packets

One of the prime objectives of TCP congestion avoidance mechanism is to avoid high packet loss rates in Internet. TCP implemented window based flow control mechanism in the early 1980s. In this type of window based control, the receiver possessed the ability to control the amount of data sent by the source nodes and this could in turn prevent the buffer space overflow of the receiver for that TCP connection. Sometime in the mid 80s, Jacobson (1988) observed a phenomenon termed as “Internet meltdown” which was later termed as congestion collapse. In 1986, Jacobson proposed congestion avoidance mechanisms to fix the Internet meltdown problem. This mechanism is currently used in TCP implementations. The congestion avoidance mechanism is installed in the end destination hosts. The TCP sources are sensitive to packet losses in the network and automatically back off when an impending congestion situation is detected.

The TCP congestion control mechanism is ‘window based’. To control the transmission rate, the total number of unacknowledged packets is bounded by a ‘congestion window’ (CWND). The source increments the size of the congestion window (value of CWND) and its transmission rate after receiving acknowledgements for data packets that were successfully transmitted. After a certain point in time, the transmission rate of the data packets forwarded by the source is bound to exceed the network’s link capacity to deliver them to the end-destination. Under this situation, buffer starts to overflow and queues start building up. Packet loss and drop starts. At this point the TCP source is warned of packet losses at the buffer and it reduces the size of its congestion window.

The implementation in TCP congestion control algorithm is simple. When the TCP source is ready for data transmission, the algorithm starts to increase the value of its congestion window in an attempt to increase the sending (transmission) rate. The window size is exponentially increased until a certain threshold value is hit (SSTHRESH). The stage is termed as the ‘slow start’ phase. In this phase, the TCP source keeps doubling its congestion window value which in turn increases the data transmission rate for every round-trip time (RTT). Once the window size exceeds the SSTHRESH value, TCP enters into the next phase termed as "congestion avoidance" phase. The window size is increased at a comparatively slower rate at this phase. TCP sources detect immediate packet loss through keeping a check on the total number of duplicate cumulative acknowledgements it receives from the end-destination node (Jacobson, 1990). When the source receives a duplicate acknowledgement for an earlier transmitted packet, it understands that there was a packet loss at the destination and it automatically reduces the congestion window size thereby reducing its data transmission rate. It sets SSTHRESH to a new value based on the newly set congestion window size. The above phases are termed as fast retransmit and fast recovery and TCP sources enter into these stages bypassing the slow start and congestion avoidance phases.

In cases of severe congestion conditions, the TCP sources lose their capability of detecting packet losses. Under such circumstances, TCP relies on 'retransmission timeout mechanism' which triggers retransmission of lost packets. In this case, TCP reduces its window size to one segment per round trip time and attempts to retransmit the lost packet. At times of severe congestion, TCP makes use of an 'exponential back off algorithm' that prevents continual retransmission of lost packets. TCP doubles its retransmission timeout interval when there are sources that continually keep sending the same packets and receive no acknowledgement for their transmitted packet. Fig 2.1 shows the slow start and congestion avoidance phase of the TCP congestion avoidance mechanism.

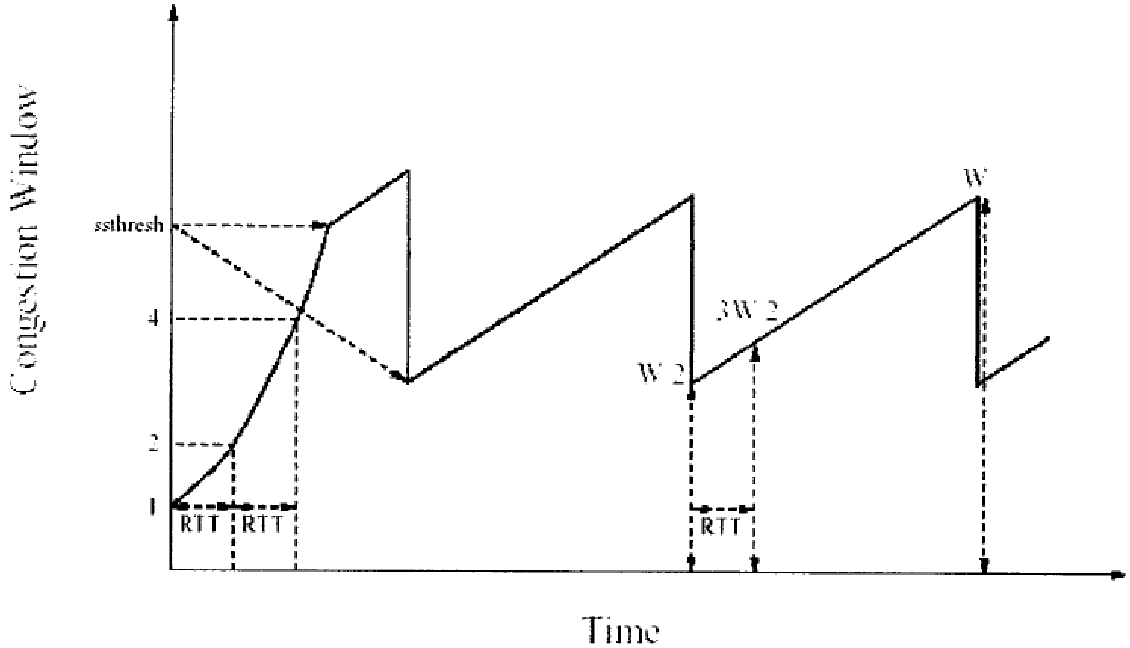

Fig 2.1 Example of TCP slow start and congestion avoidance phases (Jacobson 1988)

The TCP sources start out by assigning a value of 1 to the congestion window. The window size is exponentially increased (doubled here) for every round trip time. This continues till the window reaches the SSTHRESH value. Once the threshold value is hit, TCP slows the rate of increase of the window size. After one point in time, TCP transmission rate exceeds the network link capacity and packet loss occurs. TCP immediately detects this packet loss and reduces the congestion window size to almost half of its actual value. Once the congestion condition is cleared, TCP sources next enter into the congestion avoidance phase. The window size is increased in a linear rate at one packet per RTT. Once a steady state situation is reached, TCP oscillated between a window size between W and W/2. ‘W’ here depends on the link capacity of the network and the number of TCP connections active over the bottleneck link (Chang, 2004)

2.2.2 ACTIVE QUEUE MANAGEMENT

The TCP congestion control mechanism designed by Jacobson [1988] had no knowledge of the underlying gateway operation and the routing algorithms used in it. The primary reason behind this was the source nodes in real networks do not possess any comprehension of the packet discarding algorithm or the packet scheduling discipline. Only when a packet loss is detected, the source sending the packets reduces its transmission rates. The underlying problem with this mechanism is that a considerable amount of time passes between the packet drop at the router and the actual time when the source detects the packet loss. A large number of packets could be dropped in this manner as the source continues to transmit packets at a rate the network cannot support. To counteract this problem, the Internet Engineering Task Force (IETF) brought in "Active Queue Management" (AQM) to prevent unwanted packet loss.

AQM class of algorithms provides routers with improved queuing algorithms. This class of algorithms is termed as active since they are capable of sending dynamic signals to the sources as and when incipient congestion is detected at the router. Dynamic signaling is done through different methods; explicitly through Explicit Congestion Notification (Floyd 1994) or implicitly by dropping packets (Floyd and Jacobson, 1993) These methods of dynamic signaling to the source is in stark contrast to the Tail Drop queuing algorithms wherein packets get dropped only if queue is full and no further packets can be accommodated. Prevention of congestion collapse and improved performance were some of the primary reasons behind IEFT's recommendations for deployment of AQM in Internet routers in 1998.

This section briefly introduces some of the existing congestion control mechanisms and reviews their advantages against limitations. The final algorithm that is chosen to be designed and implemented would be reviewed and discussed in further detail towards the end of the section. Most of the algorithms presented in this section possess their own different way of making packet drop decisions. A few algorithms might drop packets only when they detect overflow in the queue while the rest few might drop arriving packets when the size of the stored packets in the buffer exceeds a certain threshold value. The following are some of the chosen algorithms under review for the current study. These are some of the commonly used congestion avoidance algorithms used in high-speed routing algorithms.

1. Drop Tail Mechanism
2. Random Drop Tail Mechanism
3. Early Random Drop
4. Partial Packet Discard and Early Packet Discard
5. DEC bit
6. RED

2.2.3 DIFFERENT SCHEMES OF ACTIVE QUEUE MANAGEMENT

2.2.3.1 DROP TAIL MECHANISM

This mechanism shown in Fig 2.2 follows the First In First Out (FIFO) queuing algorithm. Once the algorithm detects a packed queue, it starts packet dropping from the tail of the queue (Floyd and Jacobson, 1993).

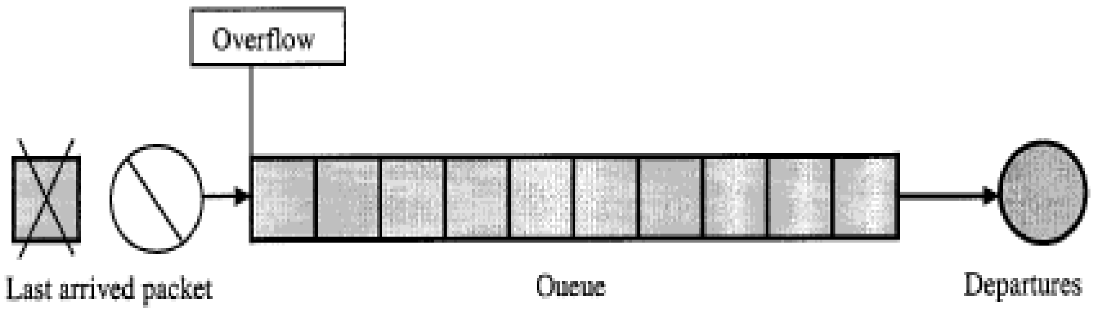

Fig 2.2 Drop Tail Mechanism (Forouzandeh, 2005)

In short, once the “queue-full” condition is detected, the last arriving packet at the router will be dropped without giving any consideration to the priority of the packet. The major drawback of this technique of FIFO scheduling is the problem of global synchronization. Dropping packets from numerous channels force them to keep resending their packets till they get transmitted and this creates congestion at the gateway. This introduces a serious case of global synchronization among all the sources connecting to the gateway.

2.2.3.2 RANDOM DROP MECHANISM

The Random Drop mechanism shown in Fig 2.3 randomly selects the packets to be dropped from all the incoming sources connected to its gateway unlike the Drop Tail mechanism that drops the last arrived packet at the tail end of the queue system.

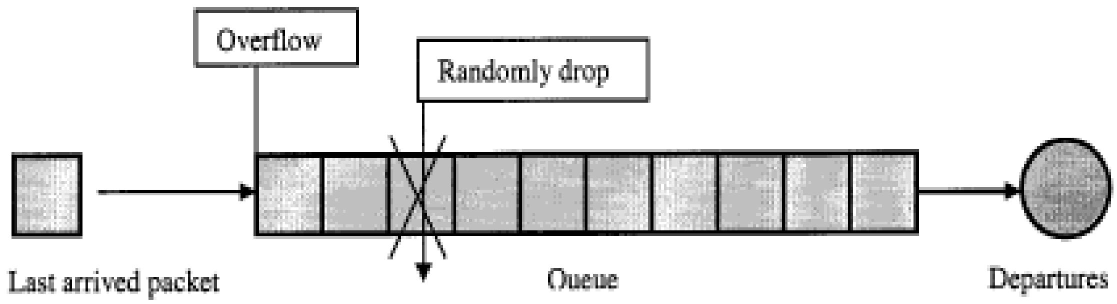

Fig 2.3 Random Drop Mechanism (Forouzandeh, 2005)

The probability of these randomly selected packets from a particular connection depends on the connection's share of bandwidth and the average rate of transmission (Mankin et al., 1991). These random packet drops are experienced by those connections that have a larger share of the gateway's bandwidth and those that generate larger amount of traffic. Connections that have smaller number of packets to generate experience smaller rate of packet loss. Mankin (1994) reviewed the Random Drop mechanism proposed by Van Jacobson and concluded that this mechanism did little to salvage the congestion recovery behavior at the gateways.

2.2.3.3 EARLY RANDOM DROP MECHANISM

This mechanism works with a fixed threshold level of the queue size as shown in Fig 2.4. Once the size of the queue is found to exceed this threshold level, all incoming packets are dropped from a gateway at a fixed probability of dropping. Hashem (1989) was one of the first researchers to investigate this mechanism of congestion avoidance scheme. Ongoing studies on this mechanism suggest that the drop probability must be dynamically tuned giving regard to the amount of network traffic in the gateway (Floyd and Jacobson, 1993).

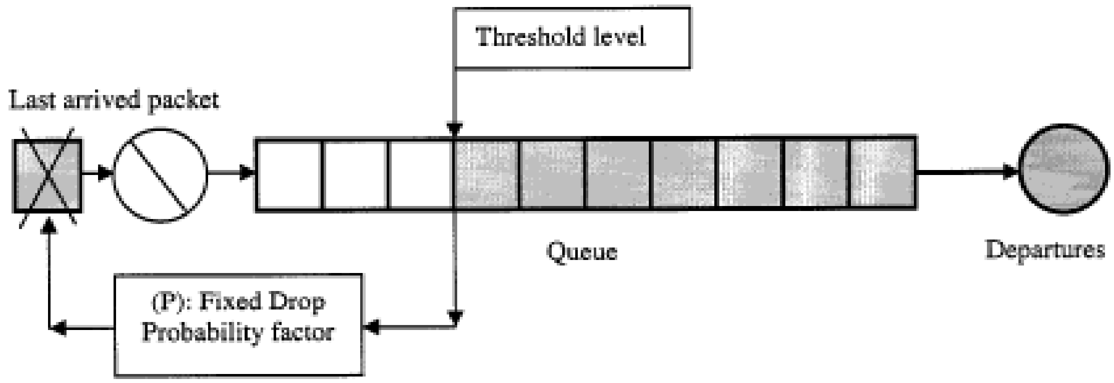

Fig 2.4 Early Random Drop Mechanism (Forouzandeh, 2005)

As mentioned earlier, Drop Tail gateways suffer from problems of loss of throughput at the gateways and global synchronization. This occurs due to all packets being dropped from the gateway during queue overflows. Early Random Drop mechanisms have a broader view of the traffic distribution as opposed to the Drop Tail mechanism and hence are recommended over the latter. However, there are some limitations tied with this mechanism of early drop. They are not effective in handling bursty traffic and misbehaving users who send this type of data.

2.2.3.4 IP SOURCE QUENCH

IP Source Quench uses the “congestion recovery” policy as described in (Mankin et al., 1991). This policy is being used as a congestion control mechanism at high speed gateways in the Internet. Whenever the gateway responds to congestion by dropping an incoming packet, this policy sends a kind of warning/feedback signals to the sources connecting to the gateway cautioning them of an impending congestion situation. This type of warning signal is termed as ICMP (Internet Control Message Protocol) in IP Source Quench mechanism. This mechanism is desirable since they yield a control on the amount of data the sources can send without overloading the gateways with unwanted congestion. The main deciding factor in IP Source Quench is the decision on when to the send ICMP message to the sources. Based on various engineering based experiments, Request for Comments (RFC) 1254 (Mankin et al., 1991) recommendations conclude that Source Quench should be initiated when nearly half of the buffer space is filled with packets. Once the ICMP message is given to the sources, it demands the TCP or any other protocol implementation connected to the sources to reduce their amount of data transmission rates.

2.2.3.5 DECbit Gateway

This is another mechanism that makes use of the congestion recovery policy similar to the IP Quench Source mechanism. The IP Quench technique sends a complete message to the source nodes to signal impending congestion while the DECbit sends only a 1-bit feedback message to the sources. This 1-bit message informs the source/sender of an impending congestion at the gateway and is enabled based on the value of average queue length. Calculations of the average queue length take into consideration the last “busy + idle” period and the current busy period. Idle time refers to periods when there were no transmission processes and busy time refers to periods when there are packet transmissions in the gateway. According to conclusions proposed in (Floyd and Jacobson, 1993), the congestion indication bit is enabled whenever the average queue length is equal or greater than 1 after each arriving packet. In this type of congestion control mechanism, an additional bit should be added on to the header in each packet since the destination nodes must echo the congestion indication bit to the source. This might be disadvantageous since it might require much overhead when compared with the other mechanisms.

2.2.3.6 RED MECHANISM

The most well known and widely used AQM mechanism is RED algorithm that was proposed by (Floyd and Jacobson, 1993) in the 1993. This is the main focus of our study. The primary objective of RED is to avoid global synchronization and to avoid bias against bursty traffic. RED provides impending congestion control instead of waiting for congestion to occur by detecting queue overflow. It also ensures that all sources and connections have an equal and fair percentage of packets dropped from them. This prevents the disparity and unfairness in drops that is observed in the Tail Drop phenomena. The following section investigates the design and implementation of RED mechanism in detail.

2.2.4 RANDOM EARLY DETECTION ALGORITHM – DETAILED VIEW

2.2.4.1 INTRODUCTION

In most traditional packet-switched networks, congestion control mechanisms are implemented in the Transport layer of the source-destination host. TCP is a dynamic reliable congestion control protocol and relies heavily on packet drops as an indicator of incipient congestion (Floyd 1994).It uses acknowledgements signals created by the destination to know if the transmitted packets are well received; lost packets are interpreted as congestion signals. To implement the end-to-end congestion control mechanisms, the IETF recommends in RFC 2309 (Branden et al., 1998) some form of active queue management in routers and buffers.

Random Early Detection (RED) algorithm was first proposed by Sally Floyd and Van Jacobson in [Floyd and Jacobson 1993] for Active Queue Management (AQM).RED algorithm is the main focus of our research work. RED probabilistically drops packets that TCP source uses as warning signals to indicate congestion existing along the transmission path. The main goal of RED is to avoid global synchronization of TCP flows, maintain high throughput with a low delay and to achieve fairness over multiple TCP connections.

The RED algorithm drops arriving packets probabilistically in contrast to other conventional queue management algorithms. The probability of drop increases as the estimated average queue size increases. RED algorithm is designed to deal with a time-averaged queue length and not the instantaneous queue. This facilitates packet drops if and only if the queue has recently been relatively full, indicating persistent congestion. Consequently, RED does not drop packets from queues that have been mostly empty in the recent past.

RED mechanism makes the decision of packet sending or dropping based on two parameters that contribute prominently to this decision; minimum threshold value (th_{min}) and maximum threshold value (th_{max}). RED calculates the average queue size with the aid of a low pass filter with an exponential weighted moving average after each packet arrival. Minimum threshold value specifies the average queue size "*below which*" no packets will be dropped while on the other hand maximum threshold value specifies the average queue size "*above which*" all incoming packets has to be dropped. As the average queue size oscillates between the th_{min} and th_{max}, packets will be dropped with a probability that varies linearly 0 to a preset probability value, P_b (discussed in later sections).

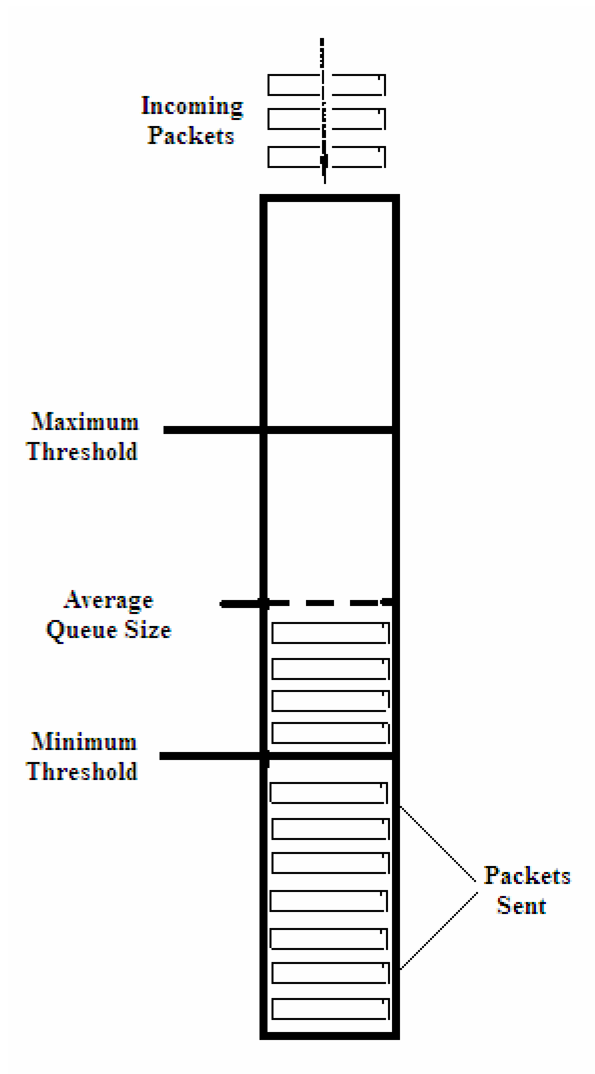

Fig 2.5 Mechanism of the RED algorithm ((Forouzandeh, 2005)

As shown in Fig 2.5 above, incoming packets aid in the calculation of average queue size. No packets are dropped as long as the average queue size is below the minimum threshold. Every arriving packet is dropped as soon as the average queue size exceeds the maximum threshold value. The most important aspect in the RED buffering mechanism as shown in Fig 2.5 is the area between the minimum and maximum threshold values. In this case, the incoming packets are dropped based on a probability termed as the "probability of discard" which is discussed in the proceeding section.

2.2.4.2 GOALS OF THE ALGORITHM

Some of the significant goals the RED buffer management tries to achieve are:

1. Congestion avoidance through controlling the average queue size and also the average queuing delay. The main function of a congestion avoidance algorithm is to detect impending congestion and to decide on the connections that have to be notified of it. In RED, if congestion is detected before the hub/gateway buffer gets full the algorithm at the gateway *marks* the packet and notifies the source to reduce the window for that connection instead of dropping the packets to notify sources of congestion.
2. Avoidance of bias against bursty traffic. Traditional drop tail and random drop gateway system possess bias against bursty traffic. In buffers of this kind, the more a connection's traffic becomes bursty, the more likely it is that queue overflow might occur in the connection with the arrival of new packets. RED accommodates short bursts that might be delay-sensitive and tries to control the average queue size from exceeding a particular threshold values.

3. Avoid global synchronization. In traditional drop tail management system, numerous connections might receive congestion signals at the same time resulting in highly undesirable oscillations in the throughput. Spikes in throughput lead to lower average throughput and introduction of jitter in the communication channel. RED avoids global synchronization through choosing congestion signals (which arriving packets to mark) randomly.

2.2.4.3 RELATED WORK

The introduction of RED had stirred considerable research interest in understanding its fundamental mechanisms, analyzing its performance and configuring its parameters to fit in various working environments. In spite of having been introduced numerous years back, it is still being used in several applications due to its efficiency in producing high throughput and avoidance of global synchronization. Numerous methods have been proposed to solve some of the deficiencies of the original RED mechanism.

Based on variation of maximum discarding probability parameter of RED with respect to the number of flows present in the system, Feng et al. (1999) proposed an adaptive discard mechanism. Floyd et al. (2001) worked on this adaptive mechanism and made numerous algorithmic modifications to it, validating their own model through various modeling and simulation techniques. Their improved version of the Adaptive RED was able to achieve a specific target average queue length under varied traffic scenarios and also reduce the parameter sensitivity that affected the performance of RED mechanism.

Inefficiency of RED algorithm in the presence of non-TCP-friendly flows (such as UDP traffic) was studied by Lin and Morris (1997). They proposed a per flow version of RED

(FRED). The FRED model resulted in a number of similar types of algorithms: RED+ (Ziegler et al., 1999), BRED (Anjum and Tassiulas, 1999), SRED (Ott et al., 1999), REM (Athuraliya et al., 2000) and Stochastic fair BLUE (Feng et al., 2001) that dealt with the problem of isolating misbehaving flows from well behaved ones.

Functionality of identifying and discriminating unresponsive, high-bandwidth best-effort flows was added to the RED algorithm through RED+ mechanism. SRED mechanism proposed a statistical algorithm to estimate the number of active flows in a bottleneck link through comparing the flow identifier of each incoming packet against a randomly chosen zombie in a zombie list. The list maintains a record of traffic flows that sent packets to the bottleneck link. Whenever a comparison is successful, a hit is assumed to have occurred and the average hit rate gives an indication of the number of active flows.

BRED works at achieving fair bandwidth allocation for varied traffic flows through penalizing non- responsive traffic that tries to steal buffer space from active traffic flows by preventively dropping the packets. REM attempts to achieve social optimality through control of rate-adaptive flows. Stochastic fair BLUE proposed a technique to implement fairness among large number of traffic flows. It uses a marking probability to detect and rate-limit non-responsive flows. The marking probability used in this proposal was derived from BLUE queue management scheme devised by Feng et al. (1999). RIO, another variant of RED mechanism was suggested by Clark and Fang (1998) that provides priority based services to different traffic flows in the Differentiated Services framework.

Performance of RED buffering mechanism was evaluated through analytical approaches as well. Peeters and Blondia (1999) performed a discrete-time analysis on RED buffers in the presence of feedback traffic. May et al., (2000) used a continuous-time Markov chain to build an analytic model of the RED mechanism and to establish the transition rates used to determine the probability of dropping in RED algorithm. Sharma et al., (2000) used ordinary differential equations to analyze performance of RED. Based on their method and using a similar approach, Kuusela and Virtamo (2000) studied the performance of a RIO controlled queue. Bu and Towsley (2001) proposed a fixed point model for a network of AQM routers. Oscillations within RED algorithm were studied by V.Misra et al., (2000). Hallot et al., (2001) proposed an alternative AQM technique wherein a proportional-integral (PI) controller was used instead of low pass filter to calculate the average queue size. This technique proved to have a better convergence speed than the original RED algorithm.

2.2.4.4 DETAILED DISCUSSION ON RED ALGORITHM

The RED algorithm has two separate parts:

(a) Estimation of the average queue size

(b) Decision of packet drop

(A) Estimation of average queue size

Calculation of average queue size plays a significant role in computing the drop probabilities. For each arriving packet, average queue size is calculated and is compared with the two threshold values (th_{max} and th_{min}).A detailed comparison strategy is shown in Fig 2.6. The figure depicts three cases which are discussed below;

Case -1: When the calculated moving average queue size exceeds the value of maximum threshold, all incoming packets are discarded.

Case -2: When the calculated moving average queue size is lower than the value of maximum threshold, no packets are dropped and all incoming packets are forwarded to their respective destination nodes.

Case -3: When the calculated moving average falls between the set values of minimum and maximum threshold, incoming packets are discarded with a drop probability.

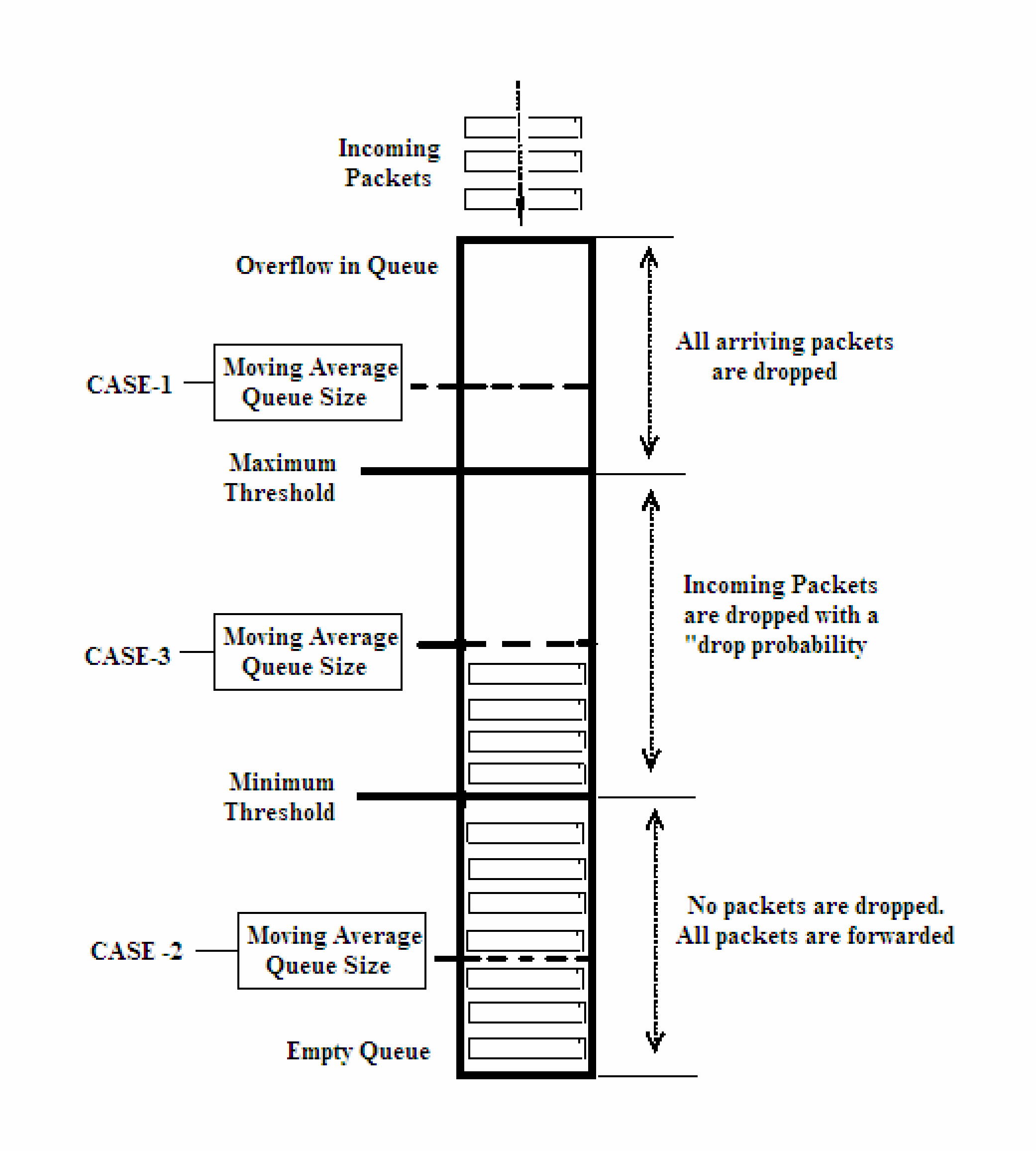

Fig 2.6 Detailed Mechanism of the RED Algorithm

As mentioned earlier, a new average queue size ($avg_{\_new}$) is calculated each time there is a new packet arrival from the existing value of average queue size ($avg_{\_old}$). This is performed with the aid of a low pass filter the formula of which was established by Floyd and Jacobson (1993) and discussed by Forouzandeh (2005).The formula for calculating the new queue size is given below;

$$\boxed{avg_{\_new} = (1-w_q) * avg_{\_old} + w_q * q} \qquad (2.1)$$

where 'q' is the current/instantaneous queue size of the stored packets in the buffer and 'w_q' is the queue weight parameter.

Floyd and Jacobson (1993) discussed various cases to find an optimal range for the choice of queue weight (w_q).This parameter decides how slowly the average queue size follows the instantaneous queue. Smaller the queue weight, slower is the movement of average queue size. Based on the recommendations of Floyd and Jacobson (1993), a suitable range for queue weight was established between a minimum of 0.001 and maximum of 0.0042 for the case where all packets had equal sizes of 1 KB at the gateway.

The authors also included calculations of average queue size taking into account the period when the queue is empty (queue is idle).Idle period occurs in cases when all the stored packets in queue are sent to their respective destinations and there is no packet arrival for a certain length of time. When a packet arrives into an empty queue (q=0) , equation (2.1) can be re-written as:

$$\boxed{avg_{\_new} = (1-w_q)^m * avg_{\_old}} \qquad (2.2)$$

$$\boxed{m = idle\_time / S} \qquad (2.3)$$

where ‘idle_time’ is the period when the queue was empty and the gateway was not receiving any new incoming packets and ‘S’ is the time needed to transmit a small packet. It is evident from equation 2.3 that the new modified formula for average queue size attempts to compensate for calculation error during idle periods.

(B) Decision of Packet Drop

RED makes the decision of packet drop based on three important parameters.

1. Value of the average queue size
2. Number of packets that were previously accommodated in the queue
3. Randomization which would ensure fair distribution of packet drops of all the sources that send packets to the gateway

They are discussed in detail in the following sections.

(a) The first parameter, probability value P_b is a function of the average queue size that lies between th_{max} and th_{min}. The average queue size is controlled between these two threshold values as mentioned in the above sections. P_b is defined as the product of max_p and P_{th}. The variable max_p assumes the highest possible value of P_b . P_{th} is a distribution function over [0,(th_{max} - th_{min})] and lies between 0 and 1.

$$\text{when} \qquad \boxed{th_{min} < avg < th_{max}} \qquad (2.4)$$

the $$\boxed{P_b = max\_p * P_{th}} \tag{2.5}$$

where $$\boxed{P_{th} = (avg - th_{max}) / (th_{max} - th_{min})} \tag{2.6}$$

and $$\boxed{P_{th} \text{ lies between 0 and 1}} \tag{2.7}$$

Hence, P_b can be re-written as;

$$\boxed{P_b = max\_p * (avg - th_{max}) / (th_{max} - th_{min})} \tag{2.8}$$

(b) The second parameter used to make the drop decision in RED is a random number, ‘R’ that lies between [0,1].

(c) The third parameter is ‘C’, a counter which holds the value of number of arriving packets that were enqueued since the last drop was performed. The drop probability increases with the increase in the counter value to maintain the average queue within threshold limits.

With the aid of all the parameters in (a), (b) and (c) outlined above, we will see how the packet drop decision is made through a simple procedure shown below in Fig 2.7. This procedure was proposed by Floyd and Jacobson (1993) and explained in detail in Forouzandeh (2005)

```
Initialize
      Avg = 0
      C = -1
For each packet arrival, calculate the new average queue size "avg"
if the queue is non-empty then
      Avg = Avg + w.(q-Avg)
Else
      m = Idle_time /  S
      Avg = Avg .(1-wq) to power m
end if
if  thmin < avg < thmax then
      increment C
      using new "Avg" and "C" calculate the probability "Pb"
      Pb = (max_p).[(Avg - thmin) / (thmax-thmin)]
      If C > 0 and C > Approx [R / Pb] then
          Drop the arrived packet
          C = 0
      end if
      if C = 0 then
            Random number [R] = Random [0,1]
      end if
else if  Avg > thmax  then
          Drop the arrived packet
          C = 0
else    C = -1
end if
when queue becomes empty then
Start counting the Idle_time
End
```

Fig 2.7 Detailed RED algorithm (Forouzandeh, 2005)

Discussion of the algorithm

Lines 2 and 3 denote the initialization steps. All the lines between 4 and 11 were previously discussed under the idle time of the queue. Line 12 discusses the situation when the average queue size lies between th_{min} and th_{max}. When the situation $th_{min} < avg < th_{max}$ holds true, the counter C is incremented each time there is a new packet arrival. Counter C was initiated to -1 in line 3 already. P_b is calculated for every arriving packet in Line 15 and the counter C starts its counting. Value stored in the counter becomes greater than zero with each subsequent new packet arrival.

The packet drop decision will depend on the probability P_b and the random number 'R' as indicated in Line 17. The second condition in 17 is the main factor in deciding the drop of the next arriving packet. The function of the random number 'R' in this condition is to randomize the final results and to distribute the drop decisions fairly among all arriving packets from the source. Line 23 gives a new value to 'R' once the drop decision is made and an arriving packet is dropped from the source.

The rest of the algorithm discusses cases when the average queue size is greater than th_{max} ($Avg > th_{max}$) where every arriving packet is dropped and the counter 'C' is incremented to zero again. Cases when $Avg < th_{min,}$ RED does not drop any packet and re-initializes the counter 'C' to -1 again.

2.2.4.5 PARAMETER SENSITIVTY OF RED ALGORITHM

Floyd and Jacobson (1993) proposed a few important rules for adequate performance of the RED mechanism under wide range of traffic conditions.

1. Queue weight (w_q) should not be set at a low value so that the calculated average queue length does not take a long time to provide actual reflection of the dynamics of instantaneous queue length.
2. (th_{min}) must be large enough to accommodate bursty traffic
3. (th_{max}- th_{min}) must be large enough to avoid global synchronization.
4. (th_{min}) = (Maximum buffer size) / 4

   (th_{max}) = 2*(th_{min})

This section discussed the various congestion control mechanisms, in particular the mechanism of AQM. Since RED algorithm is the model under present study, a detailed discussion was done on it.

2.3 FUNDAMENTALS OF SOFT COMPUTING

2.3.1 INTRODUCTION

Solutions to complex real world problems generally require intelligent methods that combine knowledge, techniques and methodologies that mimic complex human reasoning. This body of work has come to be known as Soft Computing (Zadeh, 1964). Humans possess the unique quality of reasoning and can handle any incomplete, imprecise, fuzzy information towards making intelligent decisions and choices. There are various soft computing tools used for representing such vague and fuzzy knowledge and to mimic the decision making functions of the human brain. A few of them are reviewed briefly in this section;

1. Fuzzy Logic

2. Neural Networks
3. Genetic Algorithm

2.3.1.1 FUZZY LOGIC

Fuzzy Logic represents fuzzy and vague human knowledge in specific areas of application and helps in reasoning process within the available knowledge to make intelligent decisions or undertake informed actions. It is used in direct control of different types of processes. When fuzzy logic is used in process control (such as in our study), the inferences obtained from the fuzzy decision making system is used in forming the control input to the process and the performance of the process is studied. Fuzzy logic technique is the chosen soft computing tool used in our study. Detailed discussion on it is given in the following section.

2.3.1.2 NEURAL NETWORKS

Artificial neural networks were designed to mimic the biological architecture of neurons in the human brain. These are massively connected networks consisting of “neurons” which are capable of numerical computations. ANNs find their applications in areas that require ability to approximate arbitrary non-linear functions. ANN performs non-linear function approximation through the process of “learning” by continuous interaction with the system under study.

A neural network consists of numerous neurons, organized in various layers and connected through different values of weights termed as “synapses”. Input functions given to each of these neurons are weighted and summed before being subjected to a threshold level. The aggregated inputs are then given certain activation functions to generate the output of that particular neuron. The numerous capabilities of ANN such as abilities to learn by example, massive computational power, and approximation of highly non linear functions make it an extremely “intelligent” tool of Soft Computing.

Static networks and recurrent networks are the two major classes of ANN. In static networks, the signal flow from one neuron to the other takes place in a linear fashion without any feedback signals. In the recurrent network, “learning” is achieved through example. This type of learning is termed as “Supervised Learning”. In this type of learning, a target output performance of the process is obtained through different measurements. The neural network keeps comparing its actual network output with the targeted output values and the synaptic weights of the network are adjusted to achieve the targeted output performance. Other classes of ANNs learn through “Unsupervised Learning” which is also termed as self-organization since output of each neuron is determined locally by the network itself without any data on the targeted output values. This type of learning is used in pattern recognition and data classification.

2.3.1.3 GENETIC ALGORITHMS

Genetic algorithms (GA) belong primarily to a class of intelligent systems termed as “evolutionary computing”. GAs can be termed as “derivative-free optimization” techniques. They work with population of individuals and each of this population represents a possible solution. They have the following characteristics;

1. They use evolutionary techniques such as mutation and crossover
2. They are based on probability theory

3. They conduct their search on multiple searching points

2.3.2 FUZZY LOGIC CONTROL

2.3.2.1 INTRODUCTION

In the mid 1960s L.A. Zadeh developed and proposed the theory of fuzzy logic to represent “approximate knowledge” in a systematic manner and to draw inferences from them. Fuzzy Logic deals with knowledge regarding systems that are too complex or chaotic to handle in a systematic and a practical manner. The inference engine of the fuzzy logic controller helps in making correct decisions based on the knowledge obtained from such systems. Crisp logic deals with only two states, either true (value 1) or false (value 2). Fuzzy logic works on extending the crisp two valued logic to include approximate and fuzzy knowledge. This section reviews the structure of a typical fuzzy inference system (FIS) and details on the four major units of FIS are presented.

Numerous FIS models have been proposed in Jang (1993), Kong and Kosko (1992), Lin and Lee (1991), Lin (1994), Simpson (1992), Wang (1997).The most commonly used models for inferencing are the Mamdani (Mamdani, 1977) and the Tagaki Sugeno

(Takagi and Sugeno, 1983) and (Takagi and Hayashi, 1991) model. The primary difference between the Mamdani and Sugeno inferencing model is the difference in the consequent part of the rule base. Mamdani systems deal with fuzzy antecedent and consequents (Wang, 1997) while on the other hand Sugeno systems deal with fuzzy antecedents and crisp valued consequents (Jang, 1993).

2.3.2.2 FUZZY SETS

Crisp logic allows the entities to assume only two values, either one or zero. It does not allow any uncertainty to prevail in its decision making. However, real world is characterized by uncertainties and vagueness. As an example, consider the statement, 'This distance is short'. In classical set theory or the crisp logic, this statement holds only two values, either true or false. It requires the term 'short' to be assigned an exact numeral value which would classify it as true or false. In real world, the term 'short' is subjective and varies with context and a person's perspective of the term. Let us now assign a value to distance, say a value of 'x'. Now the compatibility of the statement with 'short' is an issue of degree between 0 and 1. This is determined by "membership functions" that exist in fuzzy logic. Fuzzy logic assigns a mathematical definition to the word 'short' and evaluates the degree of belongingness of the statement to the term 'short' when the value of distance is provided.

Consider X as the universe of discourse and A as a fuzzy subset of X. Set A is assigned a membership function;

$$\mu_A : X \longrightarrow [0,1] \tag{2.9}$$

The above function associates each element 'x' present in the universe of discourse 'X' a real number $\mu_A(x)$ in the interval [0,1]. μ_A represents the degree of membership of the element 'x' in X to the fuzzy subset A. Membership functions are mathematically represented values that assign the degree of membership to its elements. The function can be defined as continuous or discrete. Triangular and Trapezoidal membership functions are the most commonly used functions for a continuous universe of discourse.

Triangular Membership Function

These are the simplest form of membership functions and are easy to represent and manipulate. They are represented by the general equation (Math Works, 2007);

$$\mu_A(x) = \begin{cases} (x-(a-b))/c & \mathit{if}\,(a-b) \le x \le a \\ 1-(x-a)/c & \mathit{if}\, a \le x \le (a+c) \\ 0 & \mathit{otherwise} \end{cases}$$

The triangular curve is a function of ‘x’ and depends on three parameters a, b, c.

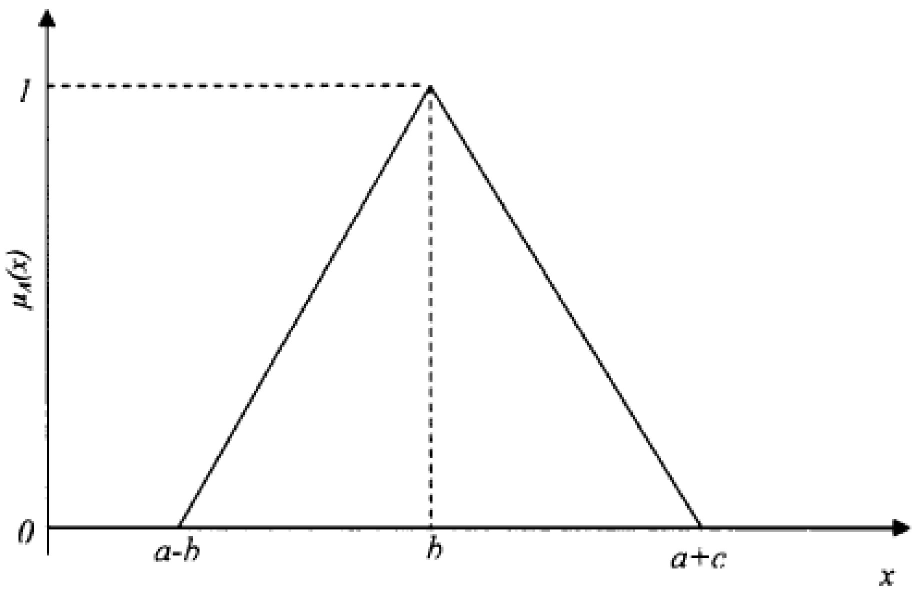

Fig 2.8 Triangular Membership Function (Math Works, 2007)

Trapezoidal Membership Function

In cases where more than one element has the highest degree of belongingness to a particular fuzzy set, this kind of membership function is the most convenient and suitable. The general equation for trapezoidal function is given by;

$$\mu_A(x) = \begin{cases} (x-(a-c))/c & if\,(a-c) \le x \le a \\ 1 & if\,a \le x \le b \\ 1-(x-b)/d & if\,b \le x \le (b+d) \\ 0 & otherwise \end{cases}$$

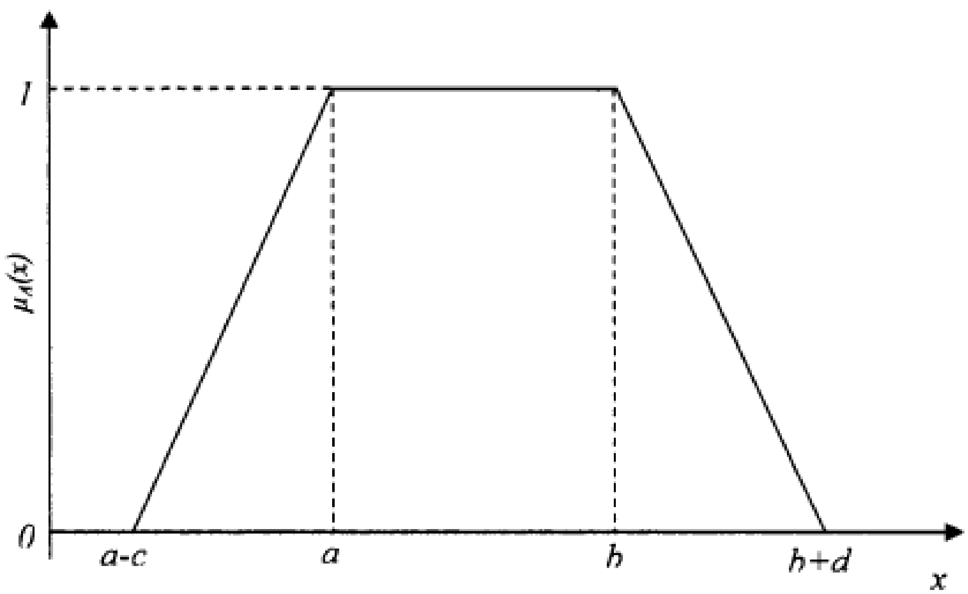

Fig 2.9 Trapezoidal Membership Function (Math Works, 2007)

There are other types of membership functions such a Gaussian MF, Pi-shaped MF, S-shaped MF, Sigmoid MF etc. However, in this study, we use only Triangular MF for its simplicity and flexibility in representation.

2.3.2.3 FUZZY INFERENCE ENGINE

The input-output mapping of the fuzzy control model is performed through the fuzzy inference system assigned with a rule base. The main components of a fuzzy inference engine are;

1. Input variables with its membership functions
2. Fuzzy rule base with linguistic rule curves

3. Output variables with its membership functions

There are three main steps involved in FIS;

1. Fuzzification
2. Inferencing
3. Defuzzification

Fuzzification involves assignment of the membership function to the input value and calculating the degree of membership of that element to the fuzzy set.

As already mentioned, the two most popular methods of inferencing are the Mamdani and Sugeno type of inference systems. Mamdani is the chosen inference system in this study.

2.3.2.5 FUZZY RULE BASE

This is the heart of fuzzy inference engine. This performs the actual mapping or inferencing between the input and output fuzzy sets. The fuzzy rule base is a collection of rule curves represented in the form of linguistic variables. A rule is represented in the form of an *If-Then* statement. The rule contains the fuzzy input set in the antecedent and fuzzy output set in the consequent.

A general rule is in the form of

If A is x *Then* B is y

In the above rule, 'x' is a fuzzy set of the input variable A and 'y' is the fuzzy set of the output variable B. Aggregation operators such as AND/OR are used when there exists more than one input or output variable. These operators are termed as t-norm and s-norm (Gupta 1991). *min* and *max* functions are the most commonly used t-norm and s-norm operators, respectively.

Implication and Composition are the two operations that happen in the inferencing process. '*min'* operators and '*max-min'* operators are used for implication and composition respectively in Mamdani inference systems.

Consider, an example to show the inferencing process (Vijaykumar 2006). Input variable X has two fuzzy sets A and B defined with triangular membership functions as shown in Fig 2.10 and output variable Y has two other sets C and D with triangular membership functions as shown in Fig 2.11.

Let us assume two fuzzy rule curves;

If X is A *Then* Y is C

If X is B *Then* Y is D

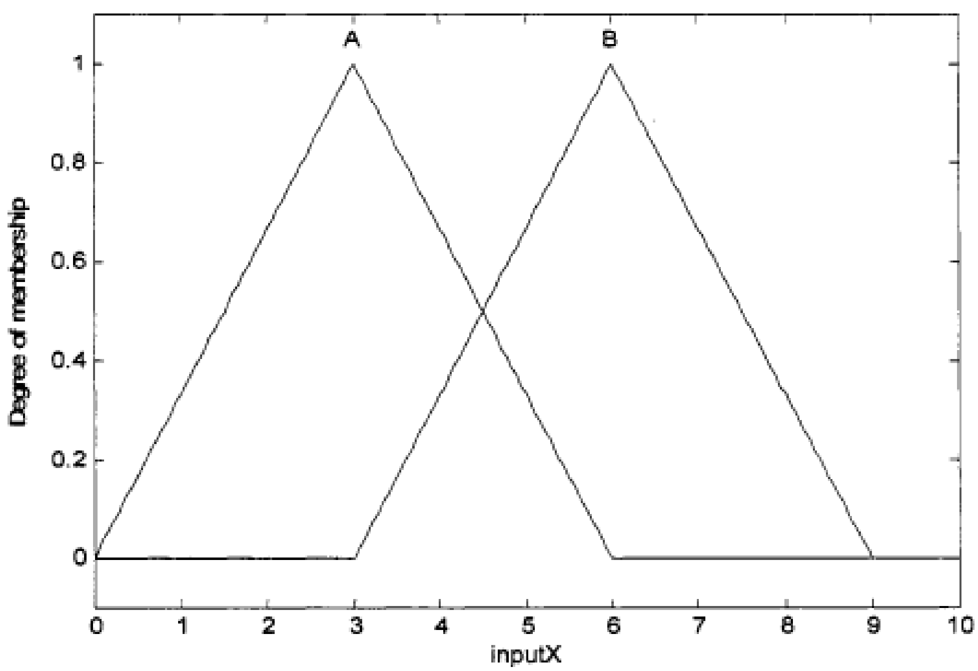

Fig 2.10 Membership Function of InputX (Vijaykumar, 2006)

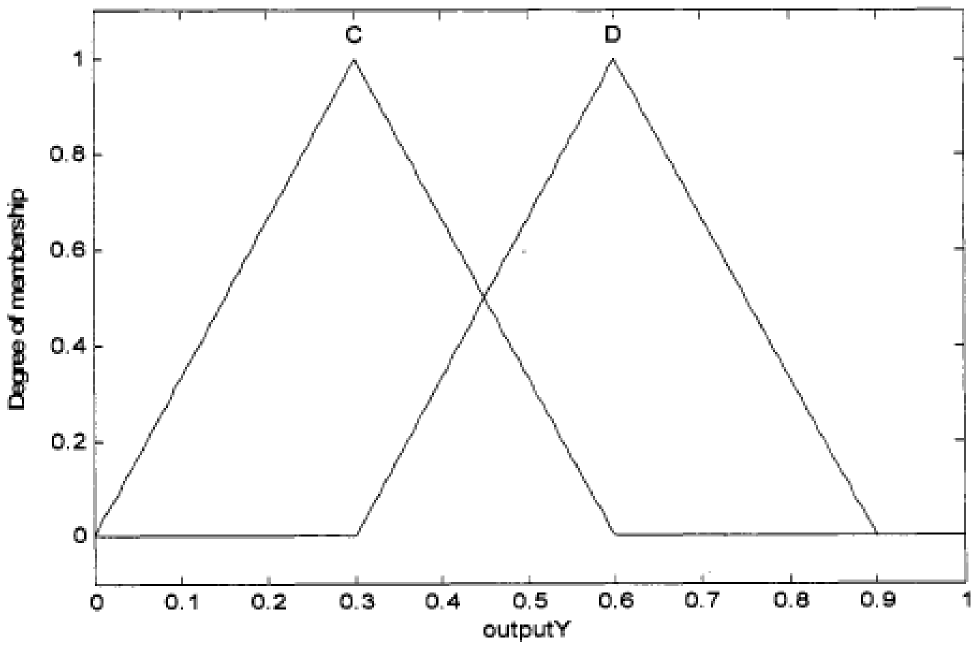

Fig 2.11 Membership Function of OutputY (Vijaykumar, 2006)

Suppose an input value of 5 is given to the fuzzy inference system, the input value is first fuzzified and its degree of belongingness to the input fuzzy set A and B is determined. Accordingly the concerned rule is fired, the result of which is projected on the output fuzzy set. This is represented in the shaded portion of the fuzzy output set given in Fig 2.12. The shaded regions in the output fuzzy set are summed by the *max-min* operator and this operation is termed as composition.

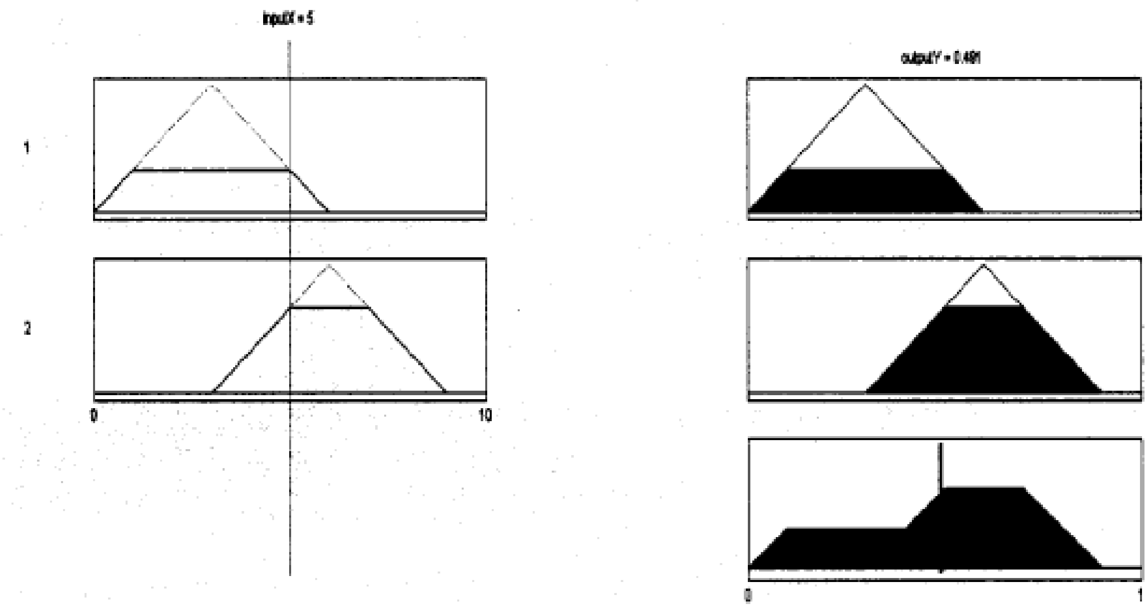

Fig 2.12 Fuzzy Inferencing (Vijaykumar, 2006)

2.3.2.5 DEFUZZIFICATION

The output set after undergoing implication and composition is in the form of a fuzzy set. It has to undergo the defuzzification process to output a crisp output value. There are many methods for defuzzification few of which are Center of Area method, Mean of Maxima method, Largest of Maximum method etc. The defuzzification method used in this study is the centre of area method or popularly known as the centroid method. The final crisp output value is calculated by determining the center of area of the output fuzzy set.

This section discussed the techniques of soft computing tools and dealt with fuzzy logic controller in detail. Based on the discussion of various components and operation on the fuzzy engine, we can conclude that fuzzy inference engines are robust in nature and can deal with real world uncertainties and vagueness. This major strength of fuzzy logic is one of the prime motivations behind using them in our present study.

2.4 SUMMARY

This chapter provided a brief background on the mechanisms and algorithms that is used in the present study. Section 2.1 introduced the area of Satellite communication and discussed the theory behind Service Level Agreements and Quality of Service measures. This was followed by survey on the various QoS techniques available in the current market. Detailed discussion on various congestion control schemes were presented in Section 2.2. TCP congestion avoidance mechanism in the internet was reviewed followed by an overview on the various available AQM techniques. The primary algorithm under study, RED was examined in detail. The goals of the algorithm, details on various RED parameter calculations and parameter sensitivity analysis were discussed. Fundamentals and three significant tools of Soft computing techniques were given a brief examination in Section 2.3. Fuzzy logic control, being the mechanism under study was predominantly discussed through analysis of the various components of fuzzy inference systems.

In conclusion, this chapter examined the three main areas relevant to the problem space and the two techniques that would be used to develop the proposed solution methodology. The next chapter analyses the problem space and formulates the solution technique with emphasis on the evaluation environment.

CHAPTER 3

PROBLEM METHODOLOGY

3.1 INTRODUCTION

This chapter introduces the problem space this study deals with and discusses the proposed problem methodology in detail. We review the design and implementation of fuzzy logic engine (FLC) under the network simulator environment. The simulator used for modeling and simulation purposes is analyzed in depth towards the end of the chapter. Given the recent advances and on-going improvements in satellite technologies, broadband satellite-based multimedia services are likely to open a promising and strong market for service providers and operators in the near future. However, the margins of safety for satellite system resources are configured as prohibitively expensive and if the physical channel is deteriorated in such a way that the communication between the gateway and the ground terminals cannot be maintained, the service level agreement (SLA) might fall in the danger of being violated. To provide predictable and reliable services, service providers of satellite communication negotiate SLAs with the customers. The service providers are expected to adhere to the agreed level of service and to deliver the promised Quality of Service to customers holding higher priorities over the services. The following elements should be taken into consideration to achieve the SLAs:

1. Bandwidth utilization
2. Quality of service (QoS)
3. Minimum targets
4. Time of day

5. Maintenance schedule
6. Weather condition
7. Other physical phenomena

Designing rules necessary to maintain the SLAs for satellite-based networks are more complex than that for non-satellite networks. Satellite-based services are more than often affected by the so-called “acts of God” such as inclement and bad weather conditions. Such uncertainties cause highly non-linear network behavior that makes it extremely difficult to engineer the SLA metrics such as throughput, latency, jitter and packet loss (discussed in Chapter 2). Problems of over-engineering the network affecting the revenue and under-engineering the network affecting the SLA commitment is a dilemma most of the service engineers face when providing communication services over the Satellite networks. Lately, to manage SLAs with the customers and to achieve packet prioritization Quality of Service is being used in both the Ground Terminals and the Hub/Gateway. Packet prioritization allows packets with higher priority levels and consequently more stringent SLAs to be serviced first at the cost of packets with lower priority when there is a contention for bandwidth.

With the advent of life critical and real-time services such remote operations over satellite, e-health etc, providing the guaranteed minimum level of services at every ground terminal through the SLAs has gained utmost priority. In communication services involving satellite networks, capacity of ground terminals might deteriorate when rain fade situation occurs due to excessive precipitation conditions. This leads to loss of service for continued periods of time and also violation of the SLA. Ground terminals are not equipped with the required intelligence to predict and react to extreme rain fade situations on its own. Additionally, the hub does not have any intelligence either to assist the ground terminals in events of rain fade occurrence.

The focus of the current work is adaptive management of the QoS at the ground terminal and the gateway level. This is done to adapt both the ground terminal and gateway to changing weather conditions and to attempt to maintain a steady throughput level and QoS requirements.

3.2 PROBLEM SPACE

The satellite communication system considered for the current research work includes three major components: the satellite, the hub gateway, and ground terminals. The satellite relays signals (both control signaling and user data) among ground terminals, as well between the hub gateway and ground terminals. The hub gateway dynamically allocates upload and download link bandwidths to ground terminals according to SLAs and conditions such as the system resources availability and weather conditions. The gateway also executes QoS control like congestion avoidance and system operation, administration, maintenance and provisioning. LANs, servers and workstations may connect to the system through the hub gateway. Besides the relay satellite and the hub gateway, the system also includes ground terminals. Ground terminals allocate resources such as bandwidth to users connected to it within the available link capacity assigned to it by the hub gateway. LANs, servers and workstations may connect to the system through ground terminals as well.

The architecture of the satellite communication system considered for the current work is shown in Figure 1. All ground terminals communicate only with the central Hub gateway, and vice versa. All communications must go through the relay satellite and the hub gateway. The satellite works as a channel relay. Bandwidth allocation algorithms may be required in both the hub gateway and ground terminals to maintain appropriate SLA requirements. The bandwidth allocation in the hub gateway is terminal-based while in ground terminal it is user-based. Packet-switched data is considered to be the main form of data communication running on the simulation system used in the current research work for various communications such as Voice over IP (VoIP), video stream (DVP), etc.

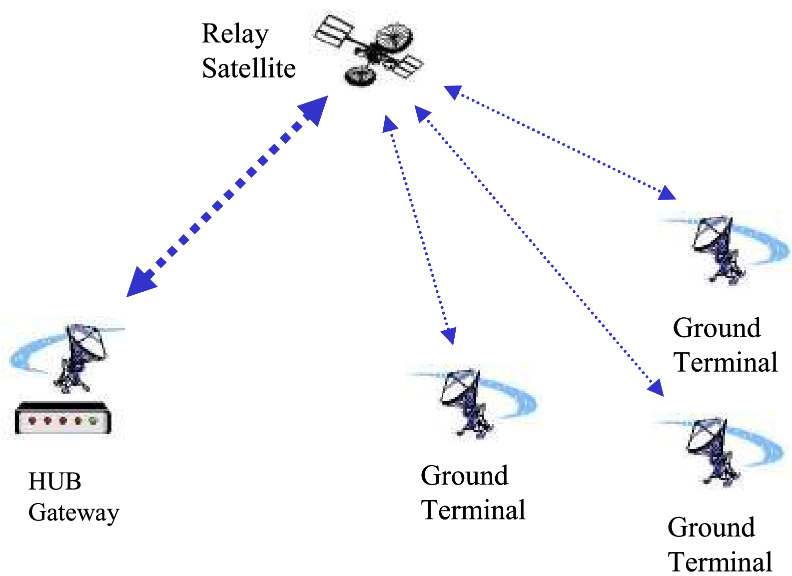

Fig 3.1 Architecture of Satellite Communication System

The most important factor taken into consideration in this research work was the impact of weather (mainly based on precipitation levels) on the QoS performance in the satellite communication system. Frequent channel degradation due to rain fade was observed in the satellite system. Currently, resources for safety backup in these systems are prohibitively expensive, e.g., spare bandwidth as the backup channel for physical links.

Upon such cases of channel degradation and the extremely limited spare resources, packet dropping in buffers in either hub gateway, or ground terminals, or both, would be inevitable and would lead to long delays and packet loss. The physical channel is deteriorated and communication between the hub gateway and ground terminals cannot be maintained. This leads to degradation of QoS and violates the SLAs leading to the so-called network congestion.

Network congestion is a fundamental problem faced in packet-switched networks. Congestion usually occurs when the transmitted number of packets exceeds the capacity of the network. The current satellite systems employ Drop Tail routing algorithms in their gateways. These discard arriving packets if the buffer at the gateway is full and follows FIFO scheduling. The Drop Tail mechanism is discussed briefly in Chapter 2. Contrary to the DT mechanism, RED algorithms start their packet drop earlier to notify traffic sources of incipient stages of congestion in the gateway. RED has been designed to substitute the DT technique of buffer management. Floyd and Jacobson (1993) claimed that RED algorithm yields better performance compared to the DT method through accommodating bursty traffic and avoiding global synchronization. Global synchronization creates high oscillations in the queue size giving rise to suboptimal throughput and high delay, jitter and packet loss rates. RED combats with global synchronization by spreading out packet losses in time and minimizing consecutive packet drops. It also attempts to avoid bias against bursty traffic connections. This is an important factor in today's internet environment since bursty traffic sources are more likely to cause congestion problems in the gateway. Consequently, RED mechanisms are widely employed in today's IP routers, though they are often disabled by the users in the real world due to their drawbacks (Bitorika, 2004). RED normally detects congestion and randomly starts dropping data packets. However, it drops packets without any priority and does not operate on a flow specific basis. This results in delay-sensitive traffic being lost altogether resulting in wasted resources and lost data from flows which may not be receiving its fair share of bandwidth. Another major drawback behind RED mechanism is there are cases wherein the primary RED parameters (probability of dropping values and the threshold functions) could be mis-configured resulting in poor bandwidth utilization and high delay-jitter performance. Tuning of RED parameters has been an inexact science for long, so much so that some researchers have advocated against using RED, due to its tuning difficulty (Jeffay et al., 2000), (May et al., 2000). Firoiu and Borden (2000) propose recommendations and guidelines on choosing the RED parameters. Misra et al., (2000) adopted a more formal, control theoretic stand-point to investigate the issues involved in selecting the RED parameters. Ziegler et al., (2000) derived quantitative models to set up RED parameters involving TCP traffic.

3.3 PROPOSED SOLUTION

3.3.1 INTRODUCTION

In our work, we aim to use the reported strength of fuzzy logic coupled with the RED algorithm to control complex and highly non-linear systems such as the satellite communication networks under study to prevent impending congestion from happening and to maintain the QoS. Linguistic knowledge of fuzzy logic is used to better understand the non-linear probability discard and threshold functions of the RED algorithm and to provide better QoS to different kinds of traffic whilst maintaining high link utilization.

This work proposes development and integration of novel algorithms for system based intelligence in the ground terminal of the satellite communication systems. Rapidly changing atmospheric conditions that in turn affect the link capacity of the channel and the varying amounts of traffic sources trying to establish connection with the gateway are used as inputs factors in engineering a real-time adaptive QoS system that attempts to provide the right level of service to the customers.

The study aims to draw upon the vast upon the vast experience in both theoretical and practical terms, of fuzzy control in design on control algorithms (Pedrycz, 1998). We aim to exploit some of the very well know advantages of fuzzy logic control such as;

1. Easy handling of inherent nonlinearities
2. Easy handling of multiple input signals
3. Less dependence on availability of precise mathematical models
4. Ability to express the control structure of a system with *apriori* knowledge

Application of fuzzy control techniques to the congestion control problems in satellite networks is suitable due to the inherent difficulty in obtaining precise mathematical models through conventional analytical methods. It is possible to obtain simple linguistic rule curves for congestion control since traffic congestion on Internet is a well known and well understood concept in the field of communications. Our design of fuzzy control system for satellite networks is based on fuzzy logic controlled RED queuing systems that works towards providing effective control in presence of dynamic network state changes, mainly due to changing weather conditions affecting link capacities and changing traffic connections accessing the gateway.

3.3.2 DESIGN OF THE FUZZY LOGIC CONTROLLER

Fuzzy logic controllers may be viewed as alternative, non-conventional way of designing feedback controllers where it is convenient and effective to build a control algorithm without relying on formal models of the system. The control algorithm is encapsulated as a set of commonsense rules. Fuzzy logic controllers have been successfully applied in control systems for which analytical models are not easily obtained or the model itself, if available, is too complex and highly non-linear (Pitsillides et all., 1997)

Design of fuzzy logic controllers (FLC) generally involves selection of various suitable mathematical operators and representations such as types and shapes of various membership functions, the number of fuzzy sets, fuzzy implication and aggregation processes, form of defuzzification functions from a wide variety of choices. Each combination of the above mentioned operators and functions yield different kinds of output performance by the FLC. However, Jager (1995) presented the case wherein optimal output performance of a FLC was achieved by appropriate tuning of the rule curves in the knowledge base. Following this, many studies propose selection of computationally lighter and well studied operators to obtain the desired behavior of FLC through tuning the rule curves only.

Our approach to tackle the presented problem space is to implement a fuzzy controlled RED queuing mechanism. We propose to eliminate the fixed threshold value, specifically the th_{max} value and the drop probability value (discussed in detail in Chapter 2) from the

RED algorithm and replace them with dynamic network state dependent values calculated by the fuzzy inference engine. The inference engine dynamically keeps calculating the threshold value (th_{max}) and the drop probability behavior based on two network state inputs namely;

1. *Link Capacity* of the channel between the gateway and the destination (Relay Satellite in our study)
2. The *number of dynamic traffic sources* accessing the gateway has a direct impact on changing queue sizes and this provides a direct measure of the future queue state.

The most important factor taken into consideration in this study was the impact of dynamic weather conditions on the physical channel between the gateway and the satellite terminal. Also resources for safety backup in these systems are prohibitively expensive, e.g., spare bandwidth as the backup channel for physical links. Under these extreme conditions, the physical channel is deteriorated and results in reduced link capacities between the gateway and the satellite terminals. The rationale behind using varying link capacities as one of the inputs to the FLC is attributed to the above outlined facts. Evidence of the effect of varying traffic sources on the changing queue size is presented in Chapter 4.

This work used a Mamdani type of Inference Engine (discussed in Chapter 2) with MIMO (Multiple Input Multiple Output) system. Generally, to define linguistic rule curves of a fuzzy variable (both input and output), triangular, trapezoidal or Gaussian membership functions are used. In our work of FLC design for dynamic weather adaptation, we use triangular membership functions since they were proven to be extremely effective in terrestrial networks (Fan et al., 2003) and also due to their computational simplicity. Test simulations were run to compare the QoS performances obtained by using different membership functions and different fuzzy set ranges. Four types of membership functions namely, gaussian, trapezoidal, triangular and generalized bell shaped functions were tested and the resulting conclusions demonstrated the suitability of triangular membership functions in achieving the target measures. Results obtained through testing different fuzzy set ranges are presented in Chapter 4. Defuzzication is performed through the Center of Area method (COA) since it is proven to yield a superior result (Braae and Rutherford, 1978).

3.3.3 DESIGN OF THE RULE BASE

A fuzzy logic controller can be thought of as a non-linear system where the relationship between the input-output systems is defined through a set of relational expressions, more popularly termed as "linguistic rule curves". To cite an example from our designed rule base: "**If** Link Capacity is *low* **and** Number of Traffic Sources is *high* **then** the Probability of Drop is high." In the above rule curves, the terms "Link Capacity", "Number of Traffic Sources", "Probability of Drop" are the *linguistic variables* and the possible values of these linguistic variables such as "low", "medium", "high" are the *linguistic values* and are modeled by the defined fuzzy sets (Zadeh, 1964)

The design of any rule base is based on the user's expertise and experience on the behavior of the system. These are easy to be related to human reasoning and gathered experiences of the designer. There are two stages involved in design of a rule base: the linguistic rules ("surface structure") are first set and then the membership values of the input-output fuzzy sets ("deep structure") are determined (Pitsillides et al., 1997). The primary key in designing a meaningful rule base system is to have minimum number of rule curves to represent the control system along with good accuracy to achieve an optimal performance. Ongoing studies suggest assistance of techniques such as neural networks, neuro-fuzzy systems or genetic algorithms to tune the fuzzy parameters online with the aid of measurements from the system. We, however use conventional trial and error method under certain rules of thumb proposed by [Floyd and Jacobson, 1993] to design our rule base. Rule base is next fine-tuned by observing progress of simulation and the output performances (such as average latency, packet loss rate etc). Fine tuning can be performed in various ways keeping in mind the different trade-offs we observe in the output performance. For example, any decrease in average latency values should be traded off against a possible increase in loss rates (evidence of this scenario is presented in Chapter 4). Rules should be tuned in a manner that would strike a balance between these two conditions. Results from such optimized tuning of the rule base are presented in Chapter 4. The rule base designed for our study is given in the Table 3.1.

	Link Capacity	Traffic Sources	th_{max}	p_drop
Rule - 1	Low	Low	Med	Med
Rule - 2	Low	Med	High	High
Rule - 3	Low	High	High	High
Rule - 4	Med	Low	Low	Low
Rule - 5	Med	Med	Low	Med
Rule - 6	Med	High	Med	Med
Rule - 7	High	Low	Low	Low
Rule - 8	High	Med	Low	Low
Rule - 9	High	High	Low	Med

Table 3.1 Fuzzy Rule Base

Rule -1 is read by the inference engine as "“**If** Link Capacity is *low* **and** Number of Traffic Sources is *low* **then** th_{max} is medium **and** p_drop is medium.” The above rule base was designed after various simulations run on the Network Simulator (discussed in the next section) for various input-output combinations and were tuned based on our knowledge of the system. Output performances for the considered various input – output combinations were studied extensively before deciding on the final rule base system. Results obtained by using the designed rule base are presented in Chapter-4.

3.4 EVALUATION ENVIRONMENT

3.4.1 Simulation Environment

Since it is a difficult task to make tests using a real network, simulation tools are used for the present study. Network simulator (ns) and Matlab ® are the two simulation tools used to run the various traffic simulations to evaluate the performance of Fuzzy-RED and to design the Fuzzy Logic Controller respectively.

The implementation of the traffic sources is based on a recent version of ns-2 simulator (Version .1b8a). ns-2 (Fall and Varadan, 2006) is an event driven object oriented simulating tool developed by the University of California, Berkeley for general purpose network simulations. It implements various network protocols such as Transport Control Protocol (TCP), User Datagram Protocol (UDP), traffic sources such as FTP, CBR, Exponential, Pareto on/off, queue management mechanisms such as RED, Drop Tail etc. NS simulator is written in two languages: an object oriented simulator, written in C++ and OTcl (an object oriented extension on Tcl) interpreter to execute the user's command scripts. The primary reason behind ns-2 using two different programming languages is because different tasks have different needs. For instance, simulation of protocols might require efficient handling of bytes in the packet header making the run-time speed very important while on the other hand, in network studies where aim is to vary certain parameters and to examine a number of scenarios, the time to change the model and run it again quickly is important.

ns tries to fulfill needs such as these with its two languages. C++ is possess fast execution speed but is slower to adapt to change and this makes it particularly suitable for a detailed protocol implementation. OTcl runs relatively slower but can be changed making the construction and running of simulations easier. In ns-2, the complied C++ objects can be made available to the OTcl interpreter. In this manner, the ready-made C++ objects can be controlled from the OTcl level itself. The authors of (Fall and Varadhan, 2006) propose basic uses of the two languages.

Purpose of OTcl:

- For configuration, setup and one-time stuff

Purpose of C++:

- Tasks that need processing of each packet of flow
- Tasks that require a change in behavior of an existing C++ class

For example, assembling and testing of latency, queuing or loss modules (packet loss rates in our case) can be done easily with OTcl objects while on the other hand, new C++ objects are required if the user wishes to accomplish more complicated tasks such as creating a special queuing discipline or model of loss.

3.4.2 PERFORMANCE METRICS

Performance metric considered in the study is directly reflective on the QoS requirements of the problem undertaken and are reviewed in Chapter 2.

Extensive studies have been undertaken on providing detailed investigations on performance and behavior of individual flows in a queue (Morris, 1997, Lin and Morris, 1997, Christiansen et al., 2000). This work studies the total traffic performance of the various queue management mechanisms (Drop Tail, RED and proposed Fuzzy-RED) under different traffic conditions (varying link capacities and traffic sources connected to the gateway). Similar to Iannaccone (2001), we outline the performance metrics studied for the different queuing mechanisms.

Packet losses

Losses occur when there is congestion in the transmission channel and these channels are forced to drop the packets since it is almost impossible to transfer them any forward to their destination nodes. TCP, the protocol studied in our work deals with the problem by asking the sender to re-transmit their lost packets again. Other protocols, such as the UDP are less capable than the TCP and are not equipped with enough intelligence to deal with situations such as lost data. TCP Loss Rate is defined as the ratio of total number of TCP packets that are dropped to the total number of TCP packets that arrive at the same output destination node.

Queuing Delay (or) Latency

Whenever the networks are free of congestion, the transmission capability of the network is greater than the amount of traffic flow in the channel. In cases such as this, the latency or queuing delay is very low. Once congestion starts to build up in the channel, the packets are forced to wait and queues are created waiting to be serviced. In these cases, latency can be very high and pose a major problem to the dynamics of the channel. Since the network operation is fast, latency is usually measured in milliseconds (ms). Non-congested channels might have a latency of 50ms, not considering the propagation delay set between the gateway and the destination nodes. Congested channels have numerous packets waiting to be transmitted and this in turn reduces the transmission capacity of the channel which makes latency values high such as 500ms, 1000ms and sometimes even as high as 5000ms. This was observed in our experiments. In cases where the link capacity was extremely limited as a result of inclement weather conditions and where the number of sources connected to the gateway was high, the latency values shot up to values as high as 1000 ms.

Jitter (or) Standard Deviation of Latency

Variance in queue delay corresponds to jitter of flows. When the latency values are more or less constant, design engineering can re-design the applications that are latency sensitive and take into consideration the high latency values attempting to achieve an improved performance of the system. On the other hand, if variation of latency (jitter) is uncertain and high, the problem gets complicated and this makes it almost impossible for re-design of the algorithm to fix the problem of latency. Problem of high values of jitter is worse than large values of latency.

Some applications are less sensitive to latency and some less sensitive to packet loss. Latency could be given less importance in cases such as downloading large files from the Internet. On the other hand, multimedia applications (VoIP) are strictly restricted in their capacity to support latency.

3.4.3 NETWORK TOPOLOGY

Simulation of the satellite network was done with the aid of ns-2 (discussed in the previous section). The hub/gateway acts as a router to the satellite communication system since all users connected to the gateway can communicate to the satellite only through the hub. Since the physical link between the satellite and the gateway is long and involves a high propagation delay, the link in our study model is modeled with a high propagation delay of 200ms to mimic the satellite-gateway link. The link between the gateway and the satellite terminal is affected continually by changing weather conditions and this has a direct impact on the link capacity of the channel between them. Multiple users (traffic sources in our study) can connect to the router to transmit their data packets to the satellite. This affects the instantaneous queue size in the buffer, results of which are shown in Chapter 4. The two important factors affecting the output performance (packet loss rate, latency and jitter) of our model are;

1. Varying link capacities (a function of dynamic weather) of the channel between the router (gateway) and destination (satellite network).
2. Number of traffic sources connecting to the router to transmit data packets to the destination.

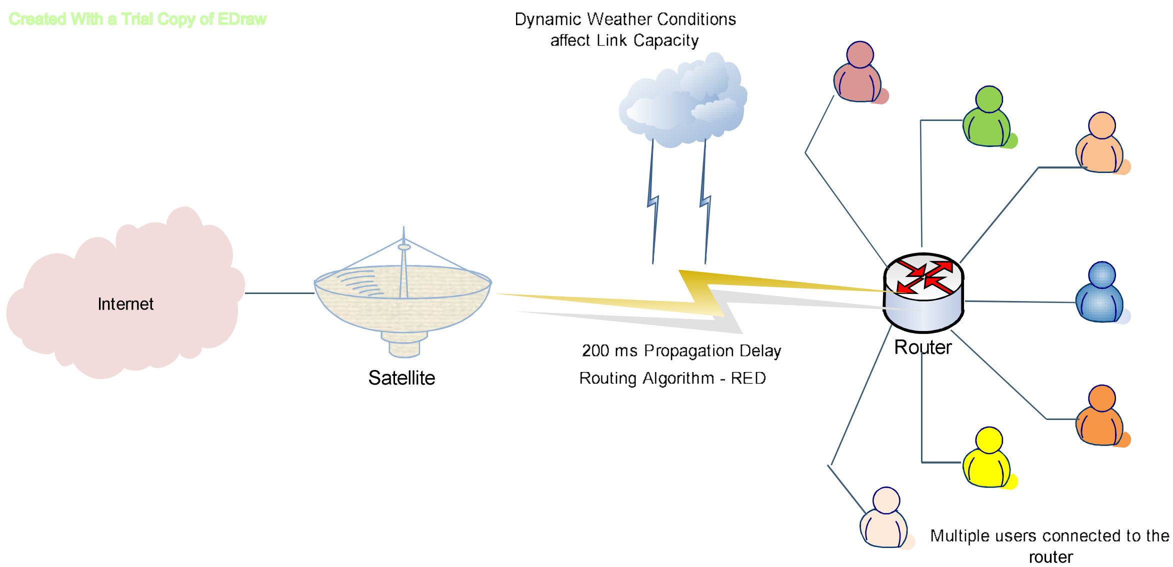

Fig 3.2 ns-2 Network Toplogy

The performance of the model is tested under 3 types of buffer management schemes:

1. Drop Tail (DT) Mechanism
2. Random Early Detection (RED) Mechanism
3. Fuzzy based RED (FRED) Mechanism

The performance of different buffer management schemes were tested on the link between the router and the destination since this is where a bottleneck situation occurred.

3.3.4 NETWORK TRAFFIC

All models considered varying number of FTP connections (traffic sources) accessing the router and the buffer size was maintained at 200 packets. We mainly considered 20, 60, 90 number of FTP connections and chose TCP packet size of 592 bytes. Each of these varying numbers of connections was modeled to start at random times: the starting time was exponentially distributed between 0 and 7 seconds. We chose a seed value of ‘2’ for the random starting times which will guarantee the same random parameters used in all test simulations. Each input link between the source and router had a propagation delay of 10ms and a link capacity of 10MB. Drop Tail buffer management scheme was used in the link between the sources and the router.

The buffer management mechanisms were tested on the bottleneck link which was modeled under varying link capacities. The ranges of link capacity considered for our study was between 120kbps to 2000kbps. The optimal output performance targeted in this study was;

Latency < 150 ms
Jitter < 30 ms
Packet Loss < 5%

Table 3.2 Quality of Service Measures

The targeted QoS measures were based on the recommendations proposed by the European Telecommunications Standards Institute (ETSI) in Ordas, 2007. In Table 3.2, the established values of latency and jitter pertain to voice communications while thee values of packet loss pertain to data communication. However, these targets are only for the purpose of demonstrating this work and can be changed for real-world applications.

Experimental results proved that the FRED mechanism based on our designed Fuzzy Logic Controller indicated improved output performance compared to other two buffer management algorithms (DT and RED) and are discussed in Chapter 4.

3.5 SUMMARY

This chapter discussed the problem associated with routing algorithms in satellite communication systems that are under continuous impact of dynamic weather patterns. The existing router system (DT algorithm) in current satellite networks was discussed followed by an in-depth analysis on modified algorithm (RED) proposed by researchers to improve the performance of the existing algorithm. A problem methodology was established to tackle the drawbacks introduced by both the existing system and the modified version it. The different components and operation of the proposed solution was outlined. This was followed by an examination on the evaluation environment under which the testing and evaluation of the proposed solution was carried out. A detailed network topology with different traffic conditions used for testing purposes was presented. The next chapter presents the performance metrics and results obtained by the proposed solution to handle the problem space.

CHAPTER 4

RESULTS AND CONCLUSION

4.1 INTRODUCTION

This chapter presents numerical results and a discussion of QoS performance measures of the three considered algorithms namely;

1. Drop Tail (DT) Algorithm
2. Random Early Detection (RED) Algorithm
3. Fuzzy – Random Early Detection (FRED) Algorithm

The above three buffer management schemes were tested on a link whose capacity is reduced at various levels (a common problem due to bad weather) to simulate the bottleneck link as discussed in Chapter 3. The performance of DT and RED algorithms under two main traffic variations were compared in terms of their latency values, packet loss rates and network power. The primary traffic variation conditions considered were;

1. Varying Link capacities
2. Varying number of FTP connections accessing the gateway

The first variation simulates bad weather conditions and the second variation simulates varying resource requirements. The implementation and simulation of the algorithms was based on a recent version of network simulator, ns-2 (Version .1b8a).

A novel fuzzy based RED algorithm was then designed bearing in mind the various limitations in the performance of both DT and RED algorithms. The two traffic variation parameters mentioned above were fed as inputs to the fuzzy inference engine that in turn outputs two of the fuzzy tuned RED parameters, namely, the buffer threshold and the probability of drop. Further simulations were done to “generate knowledge” of the optimal values of the input and output fuzzy set ranges from the rule curves for achieving the target QoS measures which are

Latency < 150 ms
Jitter < 30 ms
Packet Loss < 5%

Three priorities stages were established within the FRED model to cater to various service needs of the customers. Details of this are discussed in later sections. Simulations run for the nine different rule curves are presented at the end of the chapter along with observations and conclusion.

4.2 BEHAVIOR OF DROP TAIL ALGORITHM UNDER DIFFERENT TRAFFIC CONDITIONS

The first set of simulations was run with the DT algorithm as the buffer management mechanism on the bottleneck link between the gateway (router) and the end-host (satellite network). The simulation runtime was chosen to be 500ms after studying the trade-offs between the CPU time to run the simulation and accuracies of the results of performance measures. Table 4.1 shows the traffic specifications under which the simulations were run.

Buffer Size	200
Propagation Delay	200 ms
Simulation Run Time	500ms
Traffic Distribution of incoming packet arrival time	Exponential
Number of Sources	Variable
Application and Traffic Type	ftp/tcp

Table 4.1 Traffic Conditions for Drop Tail Algorithm

Tables 4.2- 4.4 show the performance of DT algorithm under various link capacities and various number of FTP connections (traffic sources). The performance of the algorithm is observed in terms of latency, jitter and packet loss rates (discussed in Chapter 2 & 3)

Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
1.8	20	1.66	334.48	0.35	1.17
1.4	20	1.29	470.46	0.42	1.52
1	20	0.93	690.53	0.64	1.74
0.8	20	0.74	910.17	0.78	1.62
0.4	20	0.37	1875.25	1.85	2.68
0.18	20	0.17	4316.21	4.58	3.29
0.16	20	0.15	4732.78	5.19	3.62
0.12	20	0.11	6248.35	6.89	3.91

Table 4.2 Performance of DT algorithm under 20 FTP connections

Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
1.8	60	1.67	436.06	0.65	3.55
1.4	60	1.3	546.2	0.62	3.99
1	60	0.93	766.16	0.9	4.68
0.8	60	0.74	910.15	1.14	4.84
0.4	60	0.33	2113.4	2.61	6.22
0.18	60	0.17	4717.2	5.6	6.73
0.16	60	0.15	5282.53	6.23	7.04
0.12	60	0.11	7073.97	8.46	7.48

Table 4.3 Performance of DT algorithm under 60 FTP connections

Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
1.8	90	1.7	480.91	0.8	5.84
1.4	90	1.3	576.78	0.66	6.22
1	90	0.93	800.54	0.87	7.19
0.8	90	0.74	1008.84	1.04	7.53
0.4	90	0.37	2183.51	2.42	8.96
0.18	90	0.17	4939.23	5.64	9.65
0.16	90	0.15	5693.2	6.84	10.29
0.12	90	0.11	7612.42	9.53	11.44

Table 4.4 Performance of DT algorithm under 90 FTP connections

It was observed from the above performances of DT under different traffic scenarios that jitter is the only QoS characteristic always meeting the target requirement.

Packet loss rates were initially meeting the target requirements with a lesser number of connections and higher values of network capacities. However the loss values started to increase once the number of sources trying to establish their connections with the gateway increased. Also, there was a slight fall in loss rates with decreasing link capacities.

Latency values were observed to be at an all time high irrespective of large network link capacities (1Mbps and above) and low number of FTP connections (20) connected to the gateway. Under the worst case scenario (90 FTP connections and a very low network link capacity), latency recorded an all time high value of 7600 ms as opposed to the target value of 150ms.This was concluded to be one of the major limitations of DT algorithm. Latency was out of the target limits under all traffic scenarios including the best case scenario.

4.3 BEHAVIOR OF RANDOM EARLY DETECTION ALGORITHM UNDER DIFFERENT TRAFFIC CONDITIONS

The second set of simulations was run with the RED algorithm as the buffer management mechanism on the bottleneck link between the gateway (router) and the end-host (satellite network). Table 4.5 shows the traffic specifications under which the simulations were run.

Buffer Size	200
Propagation Delay	200 ms
Simulation Run Time	500ms
Traffic Distribution of incoming packet arrival time	Exponential
Number of Sources	Variable
Application and Traffic Type	ftp/tcp

Table 4.5 Traffic Conditions for RED Algorithm

Tables 4.6- 4.8 show the performance of RED algorithm under various link capacities and various number of FTP connections (traffic sources).

Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
1.8	20	1.61	21.03	0.48	1.76
1.4	20	1.26	30.4	0.72	2.42
1	20	0.91	46.94	1.22	3.81
0.8	20	0.73	68.25	1.69	4.89
0.4	20	0.37	168.52	4.25	9.46
0.18	20	0.17	413.2	10.75	14.26
0.16	20	0.15	467.83	12.53	15.77
0.12	20	0.11	611.79	16.67	16.63

Table 4.6 Performance of RED algorithm under 20 FTP connections

Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
1.8	60	1.63	37.08	0.9	8.36
1.4	60	1.27	51.07	1.28	11.28
1	60	0.92	72.89	1.95	15.25
0.8	60	0.74	96.03	2.49	16.77
0.4	60	0.37	197.07	5.27	21.25
0.18	60	0.17	454.65	11.9	23
0.16	60	0.15	514.43	13.34	24.4
0.12	60	0.11	684.85	18.11	26.24

Table 4.7 Performance of RED algorithm under 60 FTP connections

Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
1.8	90	1.64	40.21	1.05	13.71
1.4	90	1.29	53.99	1.41	16.53
1	90	0.93	76.2	2.06	19.26
0.8	90	0.74	100.22	2.61	20.49
0.4	90	0.37	202.82	5.26	23.97
0.18	90	0.17	470.42	12.07	26.7
0.16	90	0.15	530	13.52	27.25
0.12	90	0.11	694.79	18.11	28.48

Table 4.8 Performance of RED algorithm under 90 FTP connections

As observed with the performance of jitter in DT algorithm, the jitter values in RED are always under control maintained within the target specifications.

Performance of latency in RED is almost within the target rates under high link capacities and low traffic connections. However, with increasing traffic connections and decreasing link capacities, latency values show a significant rise. The worst case scenario of very high FTP connections and very low network capacity yields a latency value of 600 ms as opposed to the target value of 150 ms.

High packet loss rates were the most important drawback observed in the RED algorithm. Loss rates were never within the specified target rates in situations with a high number of FTP connections accessing the gateway irrespective of presence of high network link capacities .Loss rates close to 30% were recorded in the worst case scenario of high FTP connection and reduced link capacity.

4.4 PERFORMANCE COMPARISON OF DT AND RED ALGORITHM

Floyd and Jacobson (1993) calculated the network power of various algorithms. Network power was calculated according to the formula given by them.

Network Power = Throughput (kbps) / Latency (ms)

Following this relation, the performance of both RED and DT were compared and evaluated. Table 4.9 shows the network power of both RED and DT mechanisms.

Link Capacity (MB)	Sources	Drop Tail	RED
1.8	20	3.105	7.284
1.4	20	1.924	5.468
1	20	1.044	3.685
0.8	20	0.666	2.72
0.4	20	0.178	1.004
0.18	20	0.0384	0.277
0.16	20	0.03	0.22
0.12	20	0.017	0.135
1.8	60	2.62	6.87
1.4	60	1.74	5.058
1	60	0.962	3.37
0.8	60	0.666	2.49
0.4	60	0.142	0.93
0.18	60	0.0345	0.25
0.16	60	0.0273	0.209
0.12	60	0.0151	0.124
1.8	90	2.496	6.82
1	90	0.929	3.36
0.8	90	0.612	2.46
0.4	90	0.155	0.91
0.18	90	0.033	0.25
0.16	90	0.025	0.2

Table 4.9 Network power of DT and RED algorithms under different traffic conditions

Values of network power under all possible scenarios of RED are observed to be higher compared to values of DT gateways. Floyd and Jacobson (1993) reached a similar conclusion using a similar topology of multiple TCP connections with large window size.

Other interesting observations include the behavior of latency and packet loss in both the models.

In both RED and Drop Tail gateways, all the three output measures show a marked change (increase/decrease) with varying combinations of link capacities and number of TCP sources connected to the gateway. With decreasing link capacities and increasing TCP connections to the gateway, there is a decrease in all three output measures, which, ofcourse, is not a surprise.

There is an increased latency introduced in models that use the Drop Tail algorithm while on the other hand, models that use the RED algorithm are forced to make a compromise on the heavy packet loss maintaining latency values at relatively lower values in comparison to Drop tail algorithms.

The *objective* now is to design a novel algorithm that would take into consideration the limitations introduced by both the algorithms and work towards minimizing the trade-offs introduced by Drop Tail gateways (high latency) and RED gateways (high packet loss).

4.5 BEHAVIOR OF QUEUE LENGTH UNDER DIFFERENT TRAFFIC CONDITIONS

Before undertaking a detailed study on parameter sensitivity of the RED parameters, we present a few test runs that were performed to study the influence of various traffic connections to the gateway on the queue length of the system under decreasing link capacities. The results are presented in Table 4.10. Simulations were run for three link capacities (1.8 MB, 0.8 MB, 0.16 MB) under 20, 60, 90 number of traffic connections.

The number of packets forwarded by the gateway to the end-host and the number of packets received at the destination node is studied. Also certain statistical parameters (maximum value, mean, standard deviation) of the instantaneous queue size under the influence of different traffic conditions were analyzed.

LC (MB)	Sources	Throughput (MB)	No of P.Sent	No of P.Recd	Q.L(max)	Q.L(mean)	Q.L(std)
1.8	20	1.657	189964	187298	100088	35754	21904
1.8	60	1.668	198209	188662	11128	54969	11691
1.8	90	1.685	204264	189118	111168	71116	11970
0.8	20	0.742	86057	83929	100088	40823	15984
0.8	60	0.754	90944	84238	100168	70422	10121
0.8	90	0.765	95148	84319	100368	78072	11954
0.16	20	0.148	17603	16858	100248	49404	12776
0.16	60	0.149	18952	16932	100568	76272	11197
0.16	90	0.15	20052	16976	100576	82097	11505

Table 4.10 Behavior of Queue Length under different Traffic Conditions

Observing the number of packets sent from the gateway to the destination, it is evident that the number of packets sent increases with increasing FTP connections. The router has more packets to forward and this could build up congestion in the network in cases of reduced network link capacity. Also queue length is found to increase with increasing amount of FTP connections. However, it was noticed that link capacities had no effect on the queue length. Varying link capacities yield a higher influence only on the number of packets sent and received in turn affecting the packet loss rate.

Based on the conclusions made from above and from earlier sections, the number of traffic sources getting connected to the gateway and the link capacities under which the entire channel works are considered to be two of the most important parameters with which further research would be performed.

4.6 EVALUATING PARAMETER SENSITIVITIES FOR KNOWLEDGE GENERATION

A parametric study was undertaken to understand the effect of tuning different RED parameters on the output QoS performances. Tuning was undertaken to develop and intuitively learn the detailed operation of RED mechanism and to generate knowledge and expertise needed to build the fuzzy rule base.

4.6.1 SIMULATION SET -1

The first set of simulations was done with 30 FTP connections accessing the RED gateway. Table 4.11 presents the output performance of default RED under decreasing link capacities with 30 sources connected to the gateway.

Algorithm	LC (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
RED	0.2	30	0.185	379.88	10.47	18.8
RED	0.18	30	0.166	425.66	11.39	18.3
RED	0.16	30	0.148	494	13.05	20.17
RED	0.14	30	0.129	7557.3	14.81	20.01
RED	0.12	30	0.111	8652.11	18.03	21.12
RED	0.1	30	0.092	9765.6	21.18	21.3

Table 4.11 Simulation Set 1 - Performance of default RED algorithm under decreasing link capacities and 30 FTP connections

a. Evaluating variation of RED parameter (th_{max}) on output performance

Table 4.12 shows the output performance for increasing values of the maximum threshold value (th_{max}) in the RED algorithm. All values of th_{max} are maintained in terms of number of packets. The minimum threshold value (th_{min}) was maintained at a default value of 50 packets for all the evaluations. Buffer size was maintained at a maximum value of 200 packets.

Default RED	0.16	30	0.148	494	13.05	20.17
th_{max}	Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
50	0.16	30	0.148	1371.9	12.06	12.9
75	0.16	30	0.148	1924.19	11.14	9.38
100	0.16	30	0.148	2357.7	10.62	7.57
125	0.16	30	0.148	2601.1	9.95	6.79
150	0.16	30	0.148	2848.06	9.62	6.25
175	0.16	30	0.148	3043	9.67	5.81
195	0.16	30	0.148	3200	9.24	5.71

Table 4.12 Evaluating the variation of RED parameter (th_{max}) on output performance with 30 FTP connections and 0.16 MB link capacity

Default RED achieved a packet loss rate of 20% with a latency value of 494 ms. It was evident from the above experiments that tuning of th_{max} value had a large impact on the packet loss rate. Loss rates as low as 5% were achieved when the th_{max} value was placed at 195 packets. However, there was an alarming increase in the latency values with increasing values of thmax and they hit an all time high of 3000 ms when the thmax is maintained at 195 packets.

We can conclude from the test cases presented that intuitive tuning of thmax does bring the loss rate under control with a trade off of high latency values.

b. Evaluating variation of RED parameter (p_drop) on output performance

Table 4.13 shows the output performance for varying values of probability of drop parameter (p_drop) in the RED algorithm.

Default RED	0.16	30	0.148	494	13.05	20.17
p_drop	Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
0.02	0.16	30	0.148	517	13.28	19.7
0.05	0.16	30	0.148	504.4	13.17	19
0.1	0.16	30	0.148	487.6	13.32	19.36
0.14	0.16	30	0.148	484.72	13.02	20.4
0.2	0.16	30	0.148	458.72	13.18	20.2
0.8	0.16	30	0.148	272.12	13.1	21.6
0.909	0.16	30	0.148	263.161	13.6	22.5
1	0.16	30	0.148	251.14	13.42	22.9

Table 4.13 Evaluating the variation of RED parameter (p_drop) on output performance with 30 FTP connections and 0.16 MB link capacity

Default RED achieved a packet loss rate of 20% with a latency value of 494 ms. Tuning p_drop values achieved latency values as low as 250 ms. Though this does not reach the target value of latency value equal to 150ms, it does bring down latency of default RED. Also observed were the increased values of packet loss rate by 2%. We can conclude from the test cases presented that tuning of p_drop does bring the latency values almost under control with a trade off of packet loss rate.

4.6.2 SIMULATION SET – 2

The second set of simulations was done with 10 TCP connections accessing the RED. Table 4.14 presents the output performance of default RED under decreasing link capacities with 10 sources connected to the gateway.

Algorithm	LC (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
RED	0.2	10	0.184	307	7.92	7.67
RED	0.18	10	0.165	371	8.81	8.41
RED	0.16	10	0.147	317	10.45	8.69
RED	0.14	10	0.129	361	12.05	9.07
RED	0.12	10	0.111	559	14.68	10.26
RED	0.1	10	0.092	665	17.7	10.58

Table 4.14 Simulation Set 2 - Performance of default RED algorithm under decreasing link capacities and 10 FTP connections

a. Evaluating variation of RED parameter (th_{max}) on output performance

th_{max}	Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
50	0.16	10	0.147	983.9	8.03	4.8
75	0.16	10	0.147	1145.98	8	4.4
100	0.16	10	0.147	1365	7.98	4
125	0.16	10	0.147	1475.98	7.99	3.9
150	0.16	10	0.147	1598.4	8.04	4
175	0.16	10	0.147	1616	7.29	3.8
195	0.16	10	0.147	1778	7.66	4

Table 4.15 Evaluating the variation of RED parameter (th_{max}) on output performance with 10 FTP connections and 0.16 MB link capacity

b. Evaluating variation of RED parameter (p_drop) on output performance

p_drop	Link Capacity (MB)	Sources	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
0.02	0.16	10	0.147	461	10.57	9.12
0.05	0.16	10	0.147	446.7	10.59	8.8
0.1	0.16	10	0.147	420.807	10.87	9
0.14	0.16	10	0.147	370.24	10.89	9
0.2	0.16	10	0.147	332.41	10.68	9.5
0.8	0.16	10	0.147	231.7	11.78	12.1
1	0.16	10	0.147	215.03	11.58	11.2

Table 4.16 Evaluating the variation of RED parameter (p_drop) on output performance 10 FTP connections and 0.16 MB link capacity

The second set of simulations result in similar performance as the ones obtained through running the first set of runs. Again, tuning of th_{max} controls packet loss rates while tuning of p_drop controls the latency values.

4.7 LEARNING IN FUZZY INFERENCE SYSTEMS

As mentioned earlier, the *objective* of this study is to design a novel algorithm that would take into consideration the limitations introduced by both the algorithms and to work towards minimizing the trade-offs introduced by Drop Tail gateways (high latency) and RED gateways (high packet loss). This is done with the aid of fuzzy logic controller or fuzzy inference engine. The fuzzy inference engine works on making decisions based on a knowledge base (presented in form of various rule curves) built into the fuzzy system in order to dynamically tune the parameters in the RED algorithm based on some system conditions such as link capacity, queue length etc.

Various evaluation simulations were run to gain a better understanding of the satellite communication channel under study and to relate the parametric influence of the RED algorithm on the three considered output performance measures.

4.7.1 FIS PROPERTIES

A detailed discussion of fuzzy logic controllers are presented in Chapter 2.The inputs to the fuzzy system are link capacity and number of traffic sources connected to the gateway, an indirect way to affect the queue lengths. This is a necessary step to overcome the automatic reduction of in the amount of source packets in ns-2 simulator when there is a reduced link capacity. The input values are fuzzified, assigned membership values and converted into input fuzzy sets shown in Fig 4.1 and 4.2. Each input variable is assigned three membership functions (High, Medium, Low).

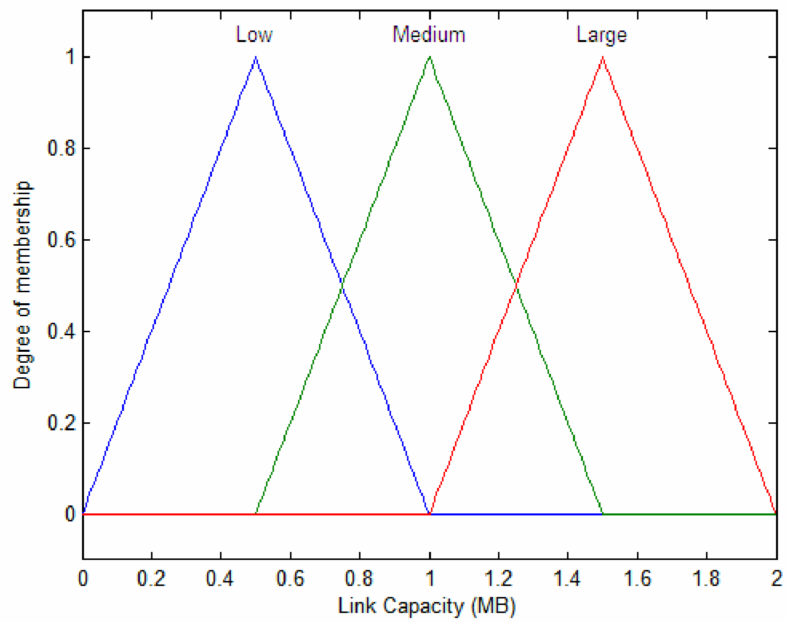

Fig 4.1 Membership functions of Input Variable 1 – Link Capacity

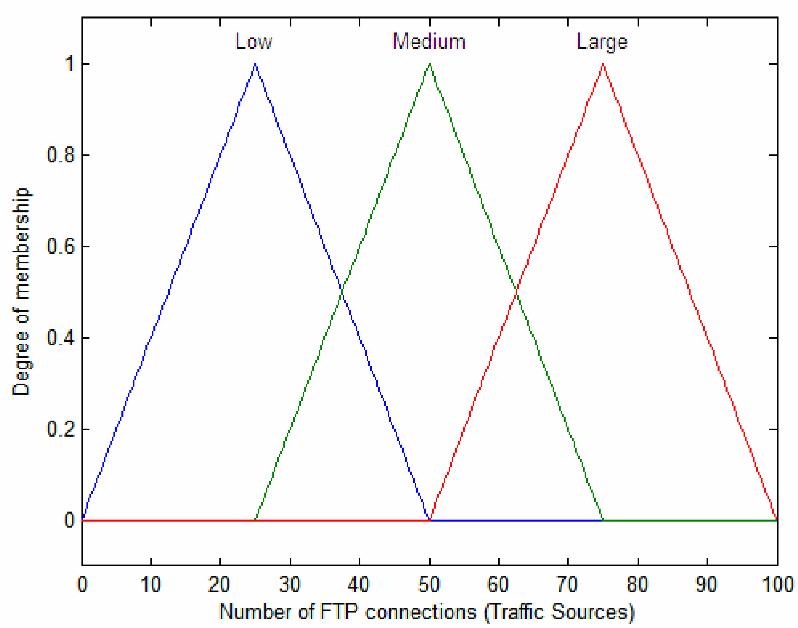

Fig 4.2 Membership functions of Input Variable 2 – Number of FTP connections

The output fuzzy sets are shown in Fig 4.3 & 4.4. Triangular membership functions were used on both the input and output variables.

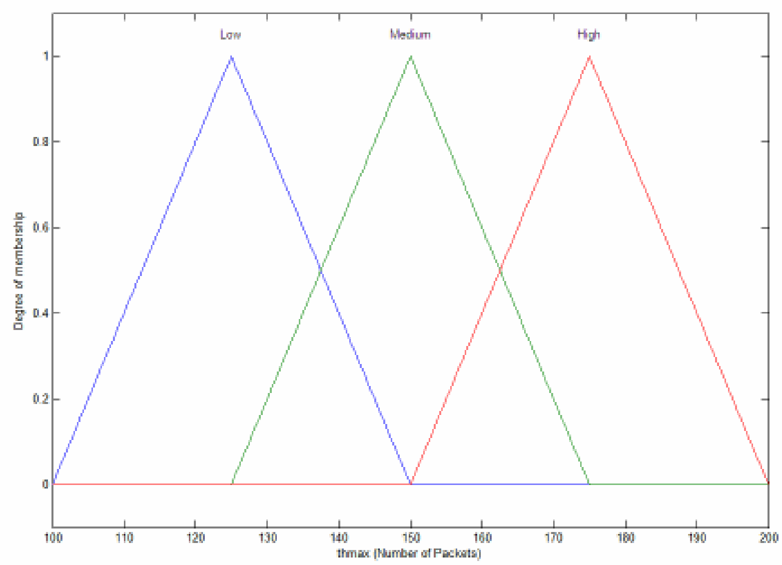

Fig 4.3 Membership functions of Output Variable 1 – th_{max}

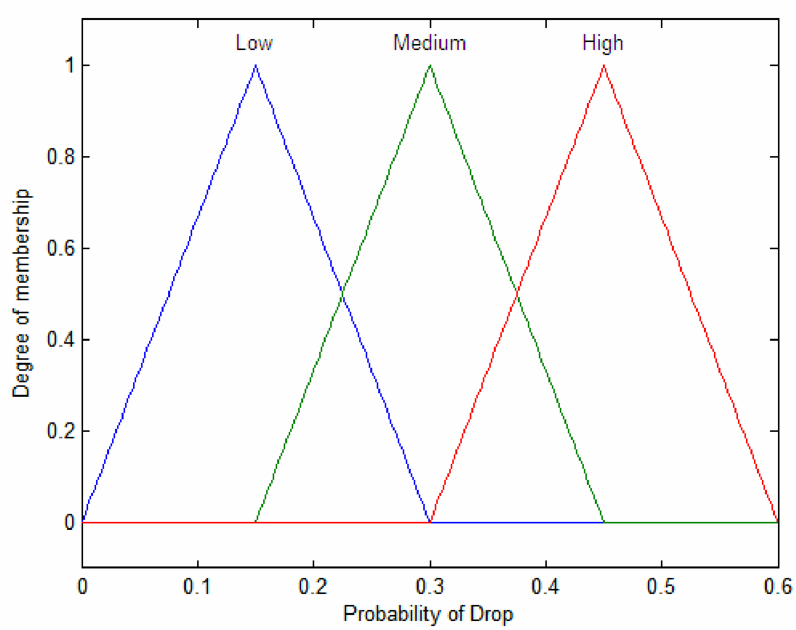

Fig 4.4 Membership functions of Output Variable 2 – p_drop

There are nine rule curves in the rule base and each rule represents a particular scenario. Simulations were run considering different values for the two input variables. The specifications are presented below;

Input Variable 1 - Link Capacity	1.8 MB assumed to belong to fuzzy set High
	0.18 MB assumed to belong to fuzzy set Med
	0.16 MB assumed to belong to fuzzy set Low
Input Variable 2 - Number of Traffic Sources	90 Sources assumed to belong to fuzzy set High
	60 Sources assumed to belong to fuzzy set Med
	20 Sources assumed to belong to fuzzy set Low

Note the term 'assumed' in the above specifications. These assumptions are based on human reasoning and the designer's expertise. Fuzzy logic converts these vague and unclear assumptions into mathematical considerations. Each vague assumption is assigned a degree of belongingness to the concerned input variable. Nine different scenarios were run in accordance to the above specifications and the designed FRED model was tested on with these scenarios. The fuzzy rule based is presented in Table 4.17.

	Link Capacity	Traffic Sources	th_{max}	p_drop
Rule - 1	Low	Low	Med	Med
Rule - 2	Low	Med	High	High
Rule - 3	Low	High	High	High
Rule - 4	Med	Low	Low	Low
Rule - 5	Med	Med	Low	Med
Rule - 6	Med	High	Med	Med
Rule - 7	High	Low	Low	Low
Rule - 8	High	Med	Low	Low
Rule - 9	High	High	Low	Med

Table 4.17 Fuzzy Rule Base

Rule -1 is read by the inference engine as ""IF Link Capacity is *low* AND Number of Traffic Sources is *low* THEN th_{max} is medium and p_drop is medium." The above rule base was designed after various simulations run for different input-output combinations and were tuned based on our knowledge of the system. Output performances for the various input – output combinations were studied extensively before deciding on the final rule base system.

Based on these rule curves and inferences made by the fuzzy engine, the output variables, th_{max} and p_drop of the RED algorithm are calculated dynamically unlike in the conventional RED approach. This is represented by the decision surfaces of the fuzzy engine used in the FRED model. Fig 4.5 and 4.6 represent the fuzzy decision surface of the output variables th_{max} and p_drop respectively.

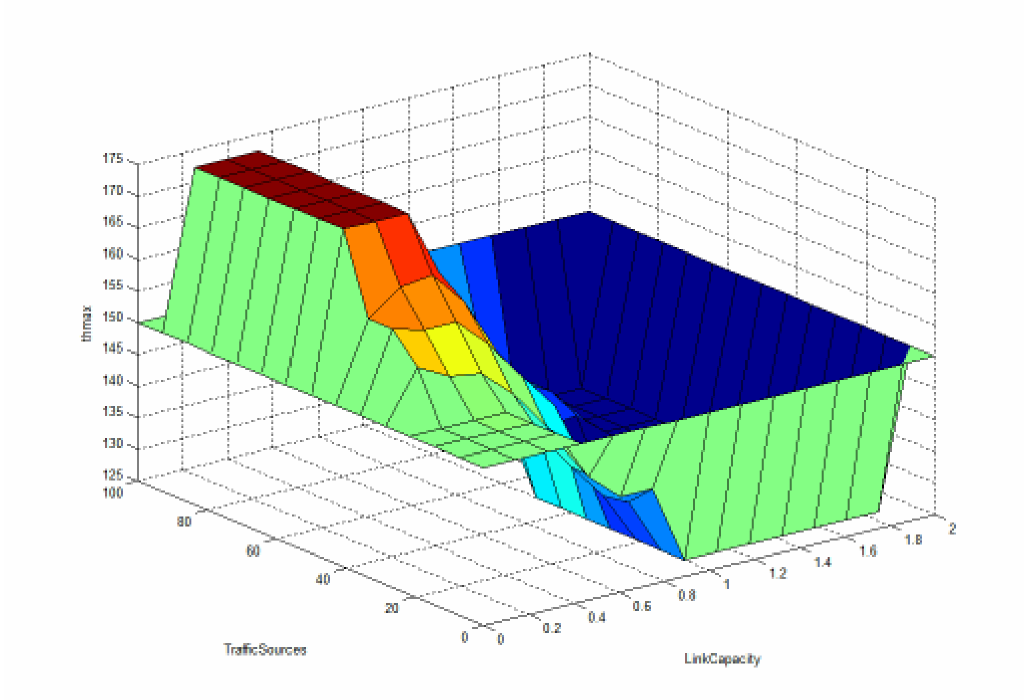

Fig 4.5 Decision Surface of the FIS for Output Variable-1

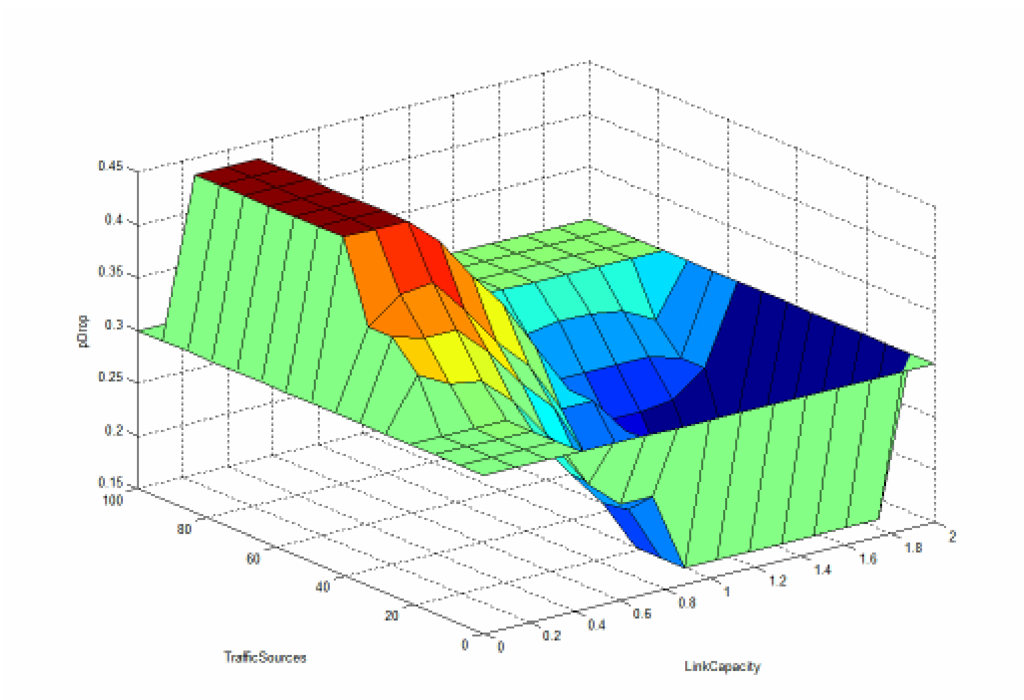

Fig 4.6 Decision Surface of the FIS for Output Variable-2

4.7.2 LEARNING IN FIS

The fuzzy sets and rule curves presented above are subjected to numerous simulation runs to test their efficiency in improving the system performance (latency, jitter and packet loss rate). The conditions under which the simulation was run were low link capacity (0.16 MB) with low number (20) of FTP connections accessing the gateway. This condition mimics rule curve 7. The default performance of the system with both Drop Tail and RED algorithms is indicated in Table 4.18. The table is presented to aid in easy comparison between default QoS performance and fuzzy based QoS performance.

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	0.148	4732.78	5.19	3.61
RED	0.1486	467.83	12.53	15.77

Table 4.18 Performance of DT and RED algorithm with 0.16 Mb link capacity and 20 FTP connections

4.7.2.1 EVALUATING THE EFFECT OF FUZZY SET RANGE OF (th_{max}) ON OUTPUT PERFORMANCE

The first few runs were performed to test different ranges of the output fuzzy set, $th_{max.}$ Table 4.19 presents the effect of different values of th_{max} on the QoS performance. The "range" column indicates the different ranges that were tried and tested on the fuzzy set "th_{max}". For example, when the range for fuzzy set "th_{max}" was set at (60-200), the FIS gave a crisp output of 130 for the parameter th_{max}. Likewise, when the range for the fuzzy set was maintained at (150-200), a crisp value of 175 was given for the thmax parameter. Also different th_{min} values were tested to investigate the optimal output performance measures.

Range	th_{min}/th_{max}	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
60-200	25,130	0.1489	2149.97	8.71	4.92
150-200	50,175	0.1489	2579.6	7.52	3.87
100-200	50,150	0.1489	2457.16	8.27	4.23
100-200	25,150	0.1489	2256.35	8.46	4.54

Table 4.19 Evaluating the variation of the fuzzy set th_{max} on the output performance under 0.16 Mb link capacity and 20 FTP connections

From the above output performance, it is quite clear that the range (100-200) for the fuzzy set thmax gives an optimal output maintaining both latency and packet loss at agreeable values. Also maintaining the value of th_{min} at 50 achieves optimal conditions. This follows the rules of thumb established by Floyd and Jacobson (1993) discussed in Chapter 2. They proposed the following conditions for optimal performance of the RED algorithm.

$$th_{min} = \text{Buffer Size} / 4$$
$$th_{max} = 2 * th_{min}$$

In our study, we worked with a buffer size of 200 packets. Following the above rule, th_{min} should be set at 50 packets while the th_{max} should be set at least at 100 packets in our model. Bearing the above conditions and rules in mind, fuzzy set range (100-200) for th_{max} and a value of 50 for th_{min} was set in our model for future test runs.

4.7.2.2 EVALUATING THE EFFECT OF FUZZY SET RANGE OF (p_drop) ON OUTPUT PERFORMANCE

The next few runs were performed to test different ranges of the output fuzzy set, p_drop. Table 4.20 presents the effect of different values of p_drop on the QoS performance. The "range" column indicates the different ranges that were tried and tested on the fuzzy set "p_drop". For example, when the range for fuzzy set "p_drop" was set at (0-1), the FIS gave a crisp output of 130 for the parameter p_drop. Likewise, when the range for the fuzzy set was maintained at (0-0.8), a crisp value of 0.35 was given.

Range	p_drop	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
0 -1	0.5	0.1485	296.04	12.66	17.79
0-0.8	0.35	0.148	340.1	12.59	17.2
0-06	0.26	0.1486	383.054	12.45	16.61

Table 4.20 Evaluating the variation of the fuzzy set, p_drop on the output performance under 0.16 Mb link capacity and 20 FTP connections

There is an optimal condition reached when the fuzzy range for the set p_drop was maintained between (0-0.6) yielding a crisp value of 0.26 for p_drop parameter. Consequently, this was the range of choice for future test runs.

4.8 FUZZY PARAMETRIC STUDY UNDER DIFFERENT TRAFFIC CONDITIONS

Following the above recognized parametric values, simulation runs were performed to test if the designed FIS model meets the established objectives of achieving trade off conditions between the performances of both Drop Tail and RED algorithms.

Three types of test simulations were run for each of the nine rule curves. The first type was named as "*Service Class A*" which aimed at reducing the packet loss and the second type named as "*Service Class B*" that aimed at reducing the latency in the model. There was also a third type of simulation namely "*Mixed Priority*" which aimed at reducing both the latency and packet loss in the model under current study.

To explain the three priority levels in further detail, we consider a sample rule curve. The traffic conditions associated with this rule curve is shown in Table 4.21.

Link Capacity	Medium(0.8 MB)
Traffic Sources	Medium(60)
th_{max}	147
p_drop	0.34

Table 4.21 Traffic Conditions

The results achieved through the developed FIS based on RED algorithm (FRED) are presented along with the output performances both Drop Tail and RED algorithms in Table 4.22. Also indicated in the table are the performances achieved through the three different priority levels.

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	0.744	990.15	1.14	4.84
RED	0.738	96.03	2.49	16.77
Service Class A	0.744	704.5	1.98	7.42
Service Class B	0.74	71.76	2.53	17.94
Mixed Priority	0.74	449.91	2.19	10.52

Table 4.22 Performance of FRED with priority levels

From Table 4.22, it is apparent that the Drop Tail algorithm achieves low packet loss (<5%) although the latency (nearly 1000ms) in the model is pretty high. The value of latency well exceeds the target (<150ms) set for the model. On the other hand RED model has a huge packet loss (almost equal to 17%) as opposed to the targets values of 5%. However, RED does maintain latency within permissible limits. The results shown for the three levels of priority are the ones obtained through the FRED model developed in earlier sections. We provide a detailed analysis of these three levels below;

Service Class A: Simulations run under this level had only its th_{max} value tuned by the fuzzy inference engine. RED parameter, p_drop was maintained at a default value set by the RED algorithm. Working through the knowledge base (stored in form of rule curves) for the given set of conditions in this simulation (Table 4.21), the fuzzy inferencing system supplied a crisp value of 147 packets for the th_{max} parameter in the RED algorithm while th_{min} was always maintained at a constant value of 50 packets with a maximum buffer size of 200 packets.

The results obtained by tuning only the th_{max} parameter in this level show that there is marked increase in latency (nearly 700 ms) as opposed to the latency of 96 ms achieved by the RED model. On the other hand the achieved latency value is lower than the latency that was achieved through the Drop Tail model which was nearly a 1000 ms.

The results for packet loss look encouraging. There is a drastic reduction of 17% loss rate obtained through RED to 7% loss rate obtained with the FRED model. However this 7% loss rate is a bit higher compared to the 4% loss rate obtained through Drop Tail algorithm.

This level is placed at the first priority since packet loss rate is the most important factor our study considers. The first priority level aims at reducing only the packet loss rates obtained through RED and does not work on achieving the target latency values.

Service Class B: Simulations under this priority level used only the p_drop parameter chosen by FIS for its simulation. Value of th_{max} was maintained at default by the RED algorithm. A packet drop probability of 0.34 was chosen for the given set of conditions (Table 4.22) under which the test simulation was run.

The latency values achieved through this priority level is lower than the ones achieved through the default algorithms. However, as expected level-2 shows a trade-off with the packet loss rate values. Packet loss rate shows an increase of nearly 2% and 14% from the values achieved through RED and Drop Tail respectively. This level is maintained at the second priority since it works on reducing the latency values of the model only and does not work with the packet loss rates as the Service Class A level does.

Mixed Priority: This level considers both the values of th_{max} and p_drop proposed by the FIS engine and tries to work on both the latency and packet loss rates of the default algorithms. It achieves a trade off between the two default cases attempting to maintain both latency and packet loss at permissible values.

The entire operation of the FRED model discussed above is presented in Fig 4.7.

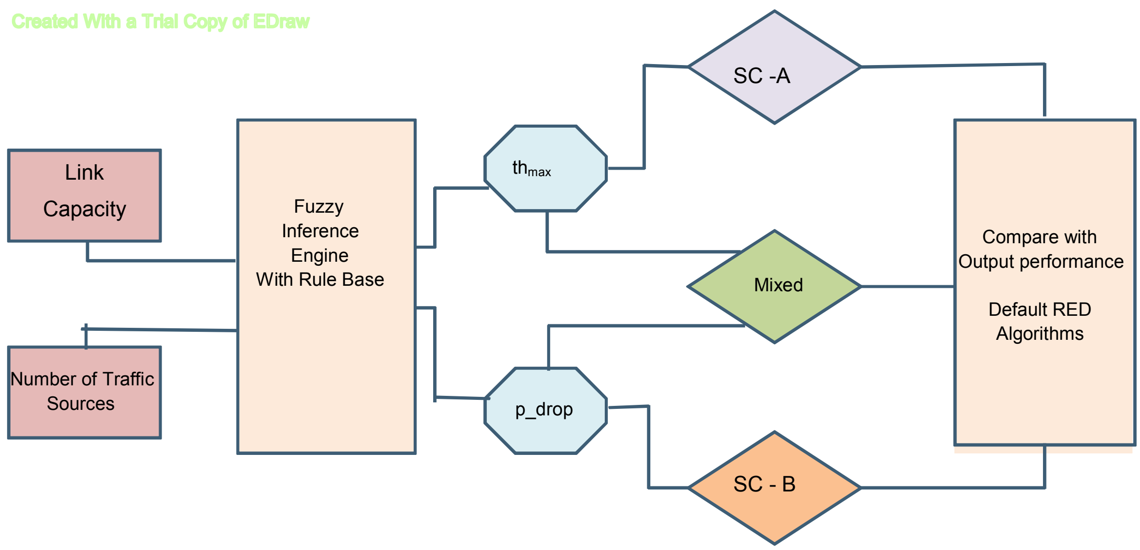

Fig 4.7 Proposed Solution – The Fuzzy-RED model

The fuzzy inference system works on fuzzification of the two input parameters, link capacity and the number of traffic sources. Inferencing is done on the rule base (with 9 rule curves) through the process of fuzzy engine. Crisp values of th_{max} and p_drop are eventually obtained after they are subjected to the process of defuzzification. Services requiring Service Class A uses only the th_{max} value from the FIS to tune their RED algorithm. On the other hand services needing Service Class B uses only p_drop value for tuning their RED algorithm. Mixed priority based services consider both the fuzzy tuned parameters to improve their QoS performances. Validation of the designed FRED model is performed through continuous comparison of its QoS performance measures with that obtained through DT and RED algorithms.

4.9 RESULTS OF FRED SYSTEM UNDER VARIOUS RULE CURVES

This section presents the results obtained for various rule curves shown in Table 4.17. Each rule curve is considered as a scenario and simulations are run according to the traffic conditions presented. QoS measures and output performance for DT, RED and FRED with various priority levels are indicated.

Rule -1 (High Link Capacity and Low Traffic Sources)

Link Capacity	High (1.8 MB)
Traffic Sources	Low (20)
th_{max}	125
p_drop	0.15

Table 4.23 Traffic Conditions for Rule -1 (Scenario 1)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	1.656	234.49	0.35	1.12
RED	1.608	21.03	0.48	1.76
Service Class A	1.653	137.97	0.42	1.43
Service Class B	1.596	19.35	0.48	1.78
Mixed Priority	1.634	134.28	0.44	1.49

Table 4.24 Output Performance of DT, RED and FRED for Rule – 1 (Scenario 1)

Rule -2 (High Link Capacity and Medium Traffic Sources)

Link Capacity	High (1.8 MB)
Traffic Sources	Medium (60)
th_{max}	125
p_drop	0.2

Table 4.25 Traffic Conditions for Rule -2 (Scenario 2)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	1.67	436.06	0.65	3.55
RED	1.627	37.08	0.9	8.36
Service Class A	1.667	217.73	0.75	5
Service Class B	1.647	27.68	0.9	8.61
Mixed Priority	1.667	172.27	0.78	5.6

Table 4.26 Output Performance of DT, RED and FRED for Rule – 2 (Scenario 2)

Rule -3 (High Link Capacity and High Traffic Sources)

Link Capacity	High (1.8 MB)
Traffic Sources	High (90)
th_{max}	125
p_drop	0.26

Table 4.27 Traffic Conditions for Rule -3 (Scenario 3)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	1.672	480.92	0.8	5.84
RED	1.644	40.21	1.05	13.7
Service Class A	1.67	276.97	0.89	8
Service Class B	1.658	31.05	1.05	14.14
Mixed Priority	1.67	192.8	0.95	9.73

Table 4.28 Output Performance of DT, RED and FRED for Rule – 3 (Scenario 3)

Rule -4 (Medium Link Capacity and Low Traffic Sources)

Link Capacity	Medium(0.8 MB)
Traffic Sources	Low(20)
th_{max}	135
p_drop	0.2

Table 4.29 Traffic Conditions for Rule -4 (Scenario 4)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss(%)
Drop Tail	0.743	910.17	0.78	1.62
RED	0.733	68.25	1.69	4.89
Service Class A	0.742	380.48	1.28	2.54
Service Class B	0.729	54.61	1.71	5.17
Mixed Priority	0.742	328.74	1.32	2.68

Table 4.30 Output Performance of DT, RED and FRED for Rule – 4 (Scenario 4)

Rule -5 (Medium Link Capacity and Medium Traffic Sources)

Link Capacity	Medium(0.8 MB)
Traffic Sources	Medium(60)
th_{max}	147
p_drop	0.34

Table 4.31 Traffic Conditions for Rule -5 (Scenario 5)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss(%)
Drop Tail	0.744	990.15	1.14	4.84
RED	0.738	96.03	2.49	16.77
Service Class A	0.744	704.5	1.98	7.42
Service Class B	0.74	71.76	2.53	17.94
Mixed Priority	744.47	449.91	2.19	10.52

Table 4.32 Output Performance of DT, RED and FRED for Rule – 5 (Scenario 5)

Rule -6 (Medium Link Capacity and High Traffic Sources)

Link Capacity	Medium(0.8 MB)
Traffic Sources	High (90)
th_{max}	163
p_drop	0.36

Table 4.33 Traffic Conditions for Rule -6 (Scenario 6)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss(%)
Drop Tail	0.744	1008.84	1.04	7.5
RED	0.741	100.22	2.61	20.49
Service Class A	0.7445	844.28	2.15	10.7
Service Class B	0.7419	75.68	2.61	20.98
Mixed Priority	0.7445	564.99	2.42	14.66

Table 4.34 Output Performance of DT, RED and FRED for Rule – 6 (Scenario 6)

Rule -7 (Low Link Capacity and Low Traffic Sources)

Link Capacity	Low(0.16 MB)
Traffic Sources	Low(20)
th_{max}	150
p_drop	0.26

Table 4.35 Traffic Conditions for Rule - 7 (Scenario 7)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	0.148	4732.78	5.19	3.61
RED	0.1486	467.83	12.53	15.77
Service Class A	0.1489	2457.16	8.27	4.23
Service Class B	0.1486	383.04	12.45	16.61
Mixed Priority	0.1489	1923.48	9.17	5.47

Table 4.36 Output Performance of DT, RED and FRED for Rule – 7 (Scenario 7)

Rule -8 (Low Link Capacity and Medium Traffic Sources)

Link Capacity	Low(0.16 MB)
Traffic Sources	Medium(60)
th_{max}	175
p_drop	0.45

Table 4.37 Traffic Conditions for Rule - 8 (Scenario 8)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss(%)
Drop Tail	0.149	5282.53	6.23	7.04
RED	0.1489	514.43	13.34	24.4
Service Class A	0.149	4390.45	10.57	9.36
Service Class B	0.148	370.88	13.36	26.15
Mixed Priority	0.149	2635.73	12.13	14.83

Table 4.38 Output Performance of DT, RED and FRED for Rule – 8 (Scenario 8)

Rule -9 (Low Link Capacity and High Traffic Sources)

Link Capacity	Low(0.16 MB)
Traffic Sources	High(90)
th_{max}	175
p_drop	0.45

Table 4.39 Traffic Conditions for Rule - 9 (Scenario 9)

	Throughput (MB)	Latency (ms)	Jitter(ms)	Packet Loss (%)
Drop Tail	0.1489	5693.2	6.84	10.29
RED	0.1489	530	13.52	27.25
Service Class A	0.1489	4651.75	11.23	13.72
Service Class B	0.1489	381.23	13.62	28
Mixed Priority	0.1489	2987.99	12.76	18.44

Table 4.40 Output Performance of DT, RED and FRED for Rule – 9 (Scenario 9)

The results obtained through these nine rule curves clearly establish the strength of the proposed FRED model in achieving perfect trade-off conditions between latency and packet loss rates. Jitter was always under control and within acceptable target conditions for all types of scenarios simulated.

Service Class A achieved very low packet loss rates compared to the ones achieved through DT and RED. On the other hand, Service Class B level of service achieved low latency values compared to the rest two algorithms. Mixed-priority level of service was able to strike a balance between latency and packet loss values compared to those achieved by DT and RED. In practice, which one of the three rules should be applied will depend on the priority of the packets. It is noted however, that for the given channel capacity, and the buffer size, the stringent QoS targets in Table 3.2 are hard to meet.

4.10 CONCLUSION

This chapter presented the results obtained through the three algorithms under study, namely Drop Tail (DT), Random Early Detection (RED) and Fuzzy –RED (FRED) algorithms. It compared the performances of both DT and RED and observed the limitations of each model. A fuzzy based RED algorithm was proposed to minimize the shortcomings of DT and RED. Properties of the proposed fuzzy algorithm were discussed and various simulation scenarios were presented to represent the learning done by the rule curves of fuzzy inferencing engine. Finally, results obtained through FRED were presented with three levels of priority assigned to each level of service availed by the customer. Final results exhibit the optimal performance of the FRED model with reasonable balance achieved between the values of latency and packet loss rates.

CHAPTER 5

SUMMARY AND FUTURE WORK

5.1 SUMMARY

This chapter presents a summary of the study undertaken on developing novel intelligence based congestion control algorithm in the satellite system ground terminals to contend with the inherent impacts of dynamic weather changes on the system.

With the advent of life critical and real-time services such as remote operations over satellite, e-health etc, providing the guaranteed minimum level of services at every ground terminal through the SLAs has gained utmost priority. In communication services involving satellite networks, capacity of ground terminals might deteriorate when rain fade situation occurs due to excessive precipitation conditions. This leads to loss of service for continued periods of time and also violation of the SLA. Ground terminals are not equipped with the required intelligence to predict and react to extreme rain fade situations on its own. Additionally, the hub does not have any intelligence either to assist the ground terminals in events of rain fade occurrence. The focus of the current work was to develop intelligent algorithms that would aid in adaptive management of the QoS at the ground terminal and the gateway level. This is done to adapt both the ground terminal and gateway to changing weather conditions and to attempt to maintain a steady throughput level and QoS requirements.

Performances in terms of latency, jitter and packet loss rate of three different algorithms were studied in depth;

1. Drop Tail mechanism, the routing algorithm used in the current satellite systems

2. Random Early Detection mechanism, proposed by researchers (Floyd and Jacobson, 1993) to overcome the limitations introduced by DT mechanism
3. Fuzzy based RED algorithm, proposed in this study, to balance the limitations introduced by DT and RED models

The modeling and simulation of the satellite communication system was executed based on a recent version of network simulator, ns-2 (Version .1b8a). Numerous simulations were run to analyze the performance of DT and RED models and the limitations of each algorithm were examined. To comprehend the robustness of the two models, the algorithms were subjected to test runs under different traffic conditions (varying network link capacity and number of FTP connections accessing the gateway).

A fuzzy logic based controller was designed to minimize and balance the trade-off conditions established by DT and RED algorithms. Linguistic knowledge of fuzzy logic was used to better understand the non-linear probability discard and threshold functions of the RED algorithm and to provide better QoS to different kinds of traffic whilst maintaining high link utilization. Rapidly changing atmospheric conditions that in turn affect the link capacity of the channel and the varying amounts of traffic sources trying to establish connection with the gateway are used as inputs factors in engineering a real-time adaptive QoS system that attempts to provide the right level of service to the customers. The final results obtained through using the proposed fuzzy tuned RED model exhibit the optimal performance of the FRED algorithm with reasonable balance achieved between the values of latency and packet loss rates. Three levels of priority were established within the FRED framework to provide different levels of QoS to the satellite service seekers.

In conclusion, this study on improvement of the congestion management scheme in current satellite systems proposed fuzzy based intelligent algorithms that attempts to satisfy the SLAs and QoS requirements under dynamic weather conditions through intelligent tuning of the RED parameters and to provide three different service classes to the customers.

5.2 RECOMMENDATIONS FOR FUTURE WORK

This research has given rise to various recommendations for future work in the fields of QoS achievement in satellite systems and also deployment of this model in adapting to different types of traffic. Effectiveness of the FRED technique established in this study was done mainly through modeling and simulation. This can be extended to the real world field work and can be tested on the real time satellite hub/gateway that is under continuous impact of changing weather conditions. Also, this study works with link between a single ground terminal and the gateway. Possible direction to future research would be to extend the developed algorithm to work with a more global system consisting of multiple ground terminals and how the weather patterns affect the overall system.

REFERENCES

Anjum F.M and Tassiulas L (1999), “Fair Bandwidth Sharing among Adaptive and Non-Adaptive Flows in the Internet”, in Proceedings of IEEE INFOCOM 1999, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, pp. 1412-1420.

Athuraliya S, Low S and Lapsley D (2000), “Random Early Marking”, in Proceedings of the First International Workshop on Quality of future Internet Services (QofIS'2000), pp. 43-54.

Bitorika A (2004), “A comparative study of active queue management schemes”, Proceedings of IEEE ICC 2004.

Braae M and Rutherford D.A (1978), “Fuzzy relations in a control setting", in IEEE Transaction Systems, Man& Cybernetics, Vol. 7, No. 3, pp. 185-188.

Braden B, Clark D, Crowcroft J, Davie B, Deering S, Estrin D, Floyd S, Jacobson V, Minshall G, Partridge C, Peterson L, Ramakrishnan L, Shenker S, Wroclawski J and Zhang L (1998), “Recommendations on queue management and congestion avoidance in the internet” , Request For Comment 2309, Internet Performance Recommendations.

Bu T and Towsley D (2001), “Fixed Point Approximations for TCP Behavior in an

AQM Network”, in UMass CMPSCI Technical Report 00-44.

Chang J (2004), “Performance analysis of a modified non-linear red algorithm for congestion avoidance in TCP/IP networks”, M.Sc. dissertation, University of Manitoba, Canada.

Christiansen M, Jeffay K, Ott D and Smith F (2000), “Tuning RED for Web Traffic”, in IEEE ACM Transactions on Networking, Vol. 9,No. 3, pp. 249-264.

Clark D and Fang W (1998), “Explicit Allocation of Best-Effort Packet Delivery Service”, in IEEE/ACM Transactions on Networking, Vol. 6, No. 4, pp. 362-373.

Fan Y, Ren F and Lin C (2003), “Design an active queue management algorithm based on fuzzy logic decision", in International Conference on Communication Technology, ICCT 2003,China, Vol. 1, pp. 286-289.

Feng W, Kandlur D.D, Saha D and Shin K.G (1999), “A Self-Configuring RED Gateway”,in Proceedings of IEEE INFOCOM 1999, pp. 1320-1328.

Feng W, Kandlur D.D, Saha D and Shin K.G (1999), “Blue: A New Class of Active Queue Management Algorithms”, Technical Report CSE-TR-387-99, University of Michigan.

Feng W, Kandlur D.D, Saha D and Shin K.G (2001), "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness", in INFOCOMM 2001, Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, pp. 1520-1529.

Fereydouni-Forouzandeh, Fariborz (2005), "FPGA implementation of congestion control routers in high speed networks", M.A.Sc. Dissertation, Concordia University, Canada.

Firoiu V and Borden M (2000), "A study of active queue management for congestion control", in Proceedings of the IEEE INFOCOMM, Vol. 3, pp. 1435-1444.

Floyd S (1994), "TCP and explicit congestion notification," Computer Communication Review, Vol. 24, No. 5,pp. 8-23.

Floyd S (1994), "TCP and Explicit Congestion Notification", ACM Computer Communication Review, Vol. 24, No. 5, pp.10-23.

Floyd S and Jacobson V (1993), "Random Early Detection Gateway in Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1, No. 3, pp. 397-413.

Floyd S, Gummadi R and Shenker S (2001), "Adaptive RED: an algorithm for increasing the robustness of RED's Active Queue Management", Technical report, ICSI. Available at http://www.icir.org/~floyd.

Gupta M.M and Qi J (1991), “Theory of t-norms and Fuzzy inference Methods”, Fuzzy Sets and Systems, Vol. 31.

Hallot C, Misra V, Gong W and Towsley D (2001),”A Control Theoretic Analysis of RED”, Proceedings of IEEE INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, pp. 1510-1519.

Hardy W.C (2001), “QoS Measurement and Evaluation of Telecommunications Quality of Service”, Print ISBN 0471499579, John Wiley & Sons, Ltd

Hashem E (1989), “Analysis of random drop for gateway congestion control”, Technical Report LCS TR-465, Laboratory for Computer Science, MIT, MA.

Iannaccone G, May M and Diot C (2001), “Aggregate traffic performance with active queue management and drop from the tail", ACM SIGCOMM Computer Communication Review, Vol. 31, No. 3.

International Organization for Standardization 8402 (1994), “Quality Management and Quality Assurance — Vocabulary”.

International Telecommunications Unit Recommendations, Y.1241 (2001), “Support of IP-based Services Using IP Transfer Capabilities”.

Jacobson V (1990), "End2end interest mailing list: Modified TCP Congestion Avoidance Algorithm".

Jacobson V (1998), “Congestion Avoidance and Control”, in Proceedings of ACM SIGCOMM 88, Vol. 18, pp. 314-320

Jager R (1995), “Fuzzy Logic in control”, Ph.D dissertation, Technische Universiteit Delft, The Netherlands.

Jamalipour A (2001), “Broad-band satellite networks - the global IT Bridge”, Proceedings of the IEEE, Vol. 89, Issue No. 1, pp 88-104.

Jang J.S (1993), “ANFIS: Adapative-network based fuzzy inference system,” IEEE Transaction Systems, Man& Cybernetics, Vol 23, pp 665-684.

Kevin Fall and Kannan Varadhan (2006), “The ns Manual”, The VINT Project, Collaboration between researchers at UC Berkeley, LBL. USC/ISI and Xerox PARC

Kong S.G and Kosko B (1992), “Adaptive Fuzzy Systems for backing up a truck and trailer”, IEEE Transactions on Neural Networks, Vol.3, pp.211-223.

Kuusela P and Virtamo J.T (2000), “Modeling RED with Two Traffic Classes”, Proceedings of the 15^{th} Nordic Teletraffic Seminar, NTS 15.

Leopoldi R (2002), “IT Services Management – A description of Service Level Agreements”.

Lin C.T (1994), “Neural Fuzzy Control Systems with Structure and Parameter Learning”, Journal of Intelligent and Robotic Systems, Vol. 17, No. 4, pp. 429-430.

Lin C.T and Lee C.S.G (1991), “Neural network based fuzzy logic control and decision system”, IEEE Transactions on Computers, Vol.40, pp.1320-1336.

Lin D and Morris R (1997), “Dynamics of Random Early Detection”, in Proceedings of ACM SIGCOMM 1997 Symposium on Communications Architectures and Protocols, pp. 127-137.

Mamdani E.H (1977), “Applications of Fuzzy logic on approximate reasoning using linguistic synthesis”, IEEE Transactions on Computers, Vol. 26, pp.1181-1191.

Mankin A (1994), “Random Drop Congestion Control”, MITRE Corp, VA 22102.

Mankin A, MITRE, Ramakrishnan K (1991), “RFC (Request for Comments): 1254”, Networking Group, Digital Equipment Corp Editors.

May M, Bonald T and Bolot J (2000), "Analytic Evaluation of RED Performance", in Proceedings of IEEE INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, pp 1415-1424.

Misra V, Gong W and Towsley D (2000), "A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", in SIGCOMM 2000: Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, pp. 151-160.

Morris R (1997), "TCP Behavior with many flows", in Proceedings of IEEE International Conference on Network Protocols, pp 205-211.

Ordas Isabel (2007), " New ETSI Model on Wideband and Noise Transmission Quality – Phase 1", European Telecommunications Standards Institute (ETSI) Workshop on Speech and Noise in Wideband Communication. Available at http://portal.etsi.org/stq/workshop2007presentations/STQ_workshop_May07_simulation_ETSI.pdf.

Ott T.J, Lakshman T.V and Wong L.H (1999), "SRED: Stabilized RED", Proceedings of IEEE INFOCOM 1999, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, pp. 1346-1355.

Pedrycz W(ed) (1998), “Computational Intelligence: An Introduction, 2nd Edition”, CRC Press.

Peeters S and Blondia C (1999), “A Discrete Time Analysis of Random Early Detection with Responsive Best-Effort Traffic”, in Proceedings of the 7th IFIP Working Conference on Performance Modeling and Evaluation of ATM and IP Networks, IFIP ATM and IP.

Pitsillides A, Ahmet Sekercioglu Y and Gopalakrishnan Ramamurthy (1997), “Effective Control of Traffic Flow in ATM Networks using Fuzzy Explicit Rate Marking (FERM)", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 2, pp. 209-225.

Sharma V, Virtamo J and Lassila P (2000), “Performance Analysis of the Random Early Detection Algorithm", in Probability in the Engineering and Informational Sciences, Vol. 16, No. 3, pp. 367-388.

Simpson P.K (1992), “Fuzzy min-max neural networks. I. Classification”, IEEE Transactions on Neural Networks, Vol. 3, No. 5 pp. 776-787.

Takagi H and Hayashi I (1991), “Neural network based fuzzy reasoning”, International Journal of Approximate Reasoning”, Vol.5, No.3, pp.191-212.

Takagi H and Sugeno M (1983), “Derivation of fuzzy control rules from human operator’s control actions”, Proceedings of IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, pp.55-60.

Taleb T, Kato N, and Nemoto Y (2005), “Recent trends in IP/NGEO satellite communication systems: transport, routing, and mobility management concerns”, IEEE Wireless Communications, Vol. 12, Issue No. 5

The Math Works Inc (2007). “Fuzzy Logic Toolbox” User’s Guide, Matlab ® , Release 7.0.4, MA, USA

Vijayakumar S (2006), “Fuzzy logic based assignable cause diagnosis using control chart patterns” M.A.Sc. dissertation, Concordia University, Canada.

Wang L.X (1997), “A course in Fuzzy Systems and Control”, Prentice Hall, Upper Saddle River, NJ, pp. 424.

Wang L, Cai L, Liu X**,** Shen X (2007), "Stability and TCP-friendliness and Delay Performance of AIMD/RED System", Computer Networks (Elsevier) *,* Vol. 51, pp. 4475-4491.

Zadeh L.A (1965), “Fuzzy sets,” Inform. Cont,Vol. 8, pp. 338–353.

Ziegler T, Fdida S and Brandauer C (2000), “Stability Criteria of RED with TCP traffic”.

Ziegler T, Fdida S and Hofmann U (1999), “RED+ Gateway for Identification and Discrimination of Unfriendly Best-Effort Flows in the Internet,” Proceedings of the seventh IFIP Working Conference on Performance Modeling and Evaluation of ATM and IP Networks, IFIP ATM and IP, pp. 27-38.