ICNS

A714148

PERFORMANCE MODELING OF ADAPTIVE CONGESTION CONTROL MECHANISMS FOR INTERNET TRAFFIC

by

Dereje Tsegaye Bedane

Supervisor: Dr. Lin Guan

A Master's by Course Dissertation
Submitted in partial fulfillment of the requirements for the award
of
Master of Science in Internet Computing and Network Security
of Loughborough University

September 2008

Abstract

The need for access integrated applications such as video, voice and data with a defined quality of service parameter over the Internet by the users are currently increasing rapidly. Yet there are challenges on the Internet backbone to operate at its capacity to assure efficient service delivery to the users.

One of the major challenges is called congestion collapse which results in issues like high packet delay, high packet loss and low packet throughput in the course of data transmission for various applications on the Internet. Now a day's congestion prevention has become one of the most critical issues that must be confronted by the users. It is also a major challenge to researchers in the field of performance modelling.

So far different researches have been carried out and remarkable achievements have been made in controlling congestion collapse and achieving minimum packets loss probability in both Single and Double Threshold analysis leading to Step and Linear reduction respectively. However, as it has been suggested by the researchers, improvements are still needed to achieve better performance results in this regard.

The intentions of this project are: First formulating a new analytical model on different packet dropping function based on the previous model. Second derivations of performance metrics such as mean queue length, throughput, response time and probability of loss equations. Third validate the accuracy of the new analytical model through extensive experiment in MATLAB program. And to find out optimum packet dropping function which capture minimum packet loss probability and contribute to the research work in performance modelling.

The results of the analysis show that exponential function is an optimum function which achieved lower probability of packet loss compared with others functions when the values of the threshold are increasing. And also a reasonable increment have been achieved in throughput, average queue length, and average queuing delay as expected with a change in threshold values.

Keywords:

Analytical Model, Previous Model, Congestion Control, Optimum Packet Dropping Function, MATLAB Program.

Acknowledgement

I would like to express my gratitude to the International Livestock Research Institute IPMS project who gave me the financial support to complete this one year master programme.

I am deeply indebted to my supervisor Dr. Lin Guan for her help, stimulating suggestions and encouragement which helped me in all the time of research for and writing of this project.

Especially, I would like to give my special thanks to all my families for their encouragement to complete this project.

Contents

List of Figures

List of Tables

List of Equations

Abbreviations

AQM	ACTIVE QUEUE MANAGEMENT
DRED	DYNAMIC RANDOM EARLY DETECTION
ECN	EXPLICIT CONGESTION NOTIFICATION
FCFS	FIRST-COME FIRST-SERVED
FIFO	FIRST-IN-FIRST-OUT
GRED-I	GENTLE- RED
I.I.D	IDENTICAL INDEPENDENT DISTRIBUTION
IETF	INTERNET ENGINEERING TASK FORCE
MMPP	MARKOV MODIFIED POISSON PROCESS
MQL	MEAN QUEUE LENGTH
MQLT	MEAN QUEUE LENGTH TARGET
QOS	QUALITY OF SERVICE
RED	RANDOM EARLY DETECTION
RFC	REQUEST FOR COMMENTS
TCP	TRANSMISSION CONTROL PROTOCOL
TD	TAIL-DROP

Chapter 1: Introduction

1.1Background

Today the Internet to experience continues staggering growth as it has become a powerful platform, offering unprecedented access to the information and exchange of ideas globally.

Consequently, the need for high speed integrated application services such as voice, video and data on the Internet with specified Quality of Service (QoS) parameters request continues to increase.

However, Internet traffic is variable in nature and the demand for buffer cannot be predicted in advance. On the other hand routers/switches have limited memory space, sometimes the incoming Internet traffic exceed the outgoing buffer size. Packets are heavily loaded in the network and congestion collapse will occurs.

As a result of network congested packets queue lengths become very large, buffer overflows, packets are delayed during transmission, incomplete information accesses and the Internet quality of services deteriorate.

There are different types of techniques that can be used to manage congestion. The traditional buffer management technique called Tail-Drop (TD) approach was designed to be an efficient and implemented on routers to control congestion. [9]

The TD technique was no congestion detected until the queue become full. When the queue was full, the maximum congestion signal was generated to notify the source and all the subsequent arriving packets were dropped.

Once source detects that packets were lost, it slows down the arrival rate of packets then the capacity of the link and packets backlog in the queue decreases. When the buffer was not full, no congestion feedback signals were generated by TD technique and the source packets transfer rate increased until overflow happened again. [5]

Therefore the TD technique was a cyclical of decrease and increase of packets arrival rate until the buffer is full and not full respectively. The technique was called best effort in performance modelling and had been used for several years to control congestion in the Internet, but it has two major drawbacks. ‘Lock-Out’ and ‘Full Queues’ [3].

Lock-Out means a situation that a single connection or a few flows to monopolize the router space, preventing other connections from getting room in the router which is the result of synchronization or other timing effects. [3, 7]

Full Queues where the router was forced to have large queues to maintain high utilizations and TCP (Transmission Control Protocol) detect congestion from loss. The network force to have long standing queue in the steady- state. [3, 7]

Therefore the TD technique is not suited for interactive applications such as voice and video which requires low end-to-end delay and jitter. Since the buffer is full for long periods of time and packets are continuously dropped until room is available to accommodate them.

To overcome the TD technique drawbacks, one of the Active Queue Management (AQM) scheme known as Random Early Detection (RED) [4] technique for congestion control at routers or getaway was developed by Sally Floyd and Van Jacobson in 1993 [6, 14].

RED technique was also recommended by Internet Engineering Task Force (IETF) [3] as a better technique compared to TD technique and is indeed widely implemented in routers today.

The RED technique detects the impeding congestion before it occurs and provides feedback to the sender [9] by either marking or dropping packets even if buffer space is available. [11]

Basically RED technique performs the following two main tasks:

- Estimation of the average queue size at the gateway and
- Packet drop decision

To accomplish the above tasks RED technique is implemented in an Exponentially Weighted Moving Average (EWMA) formula and calculates average queue size, and compared with minimum and maximum thresholds [8, 12].

- When the average queue size is less than the minimum threshold, no packets are dropped.
- When the average queue size is greater than the maximum threshold, each arriving packet is dropped.
- When the average queue size is between the minimum and the maximum thresholds, each arriving packet is dropped randomly with probability and increased linearly from 0 to 1.

Consequently RED technique provides a solution to TD technique problems by maintaining a small size steady state queue which results in reduced packet loss, decreased end-to-end delay, and avoids lock-out behaviours of the routers.

It also keeps the average queue sizes small, resulting in the efficient use of bandwidth by avoiding global synchronisation and biases against bursty traffic. [3, 7].

Even though RED Mechanism is conceptually very simple, modification of the parameters used to estimate the average queue size, or to the parameters affecting the decision to drop a packet or not, can lead to significantly different queue management dynamics. [10]

Therefore, RED technique is a basis for many other AQM mechanisms and its parameters sensitive attract many researchers to seek effective design scheme to enhance the performance by tuning the parameters. As a result range of services and traffics can be accommodated and the quality of service delivery to the customers and network operators are improved.

1.2Motivation

Researchers argued that the RED gateway algorithm can be implemented efficiently, with small number of add and shift instruction for each packet arrival in the system. Since so far there was no clear description of the parameters settings [4] and exact measurement was achieved.

Based on the above idea, different researches had been carried out in the area of Adaptive Congestion Control Mechanism for Internet traffic. The study was based on RED technique of AQM mechanism and they had achieved optimum results in performance metrics by varying the parameters settings for different network traffic and contributed to the research.

Therefore, the motivation of this project is to find out the optimum mathematical dropping function to drop a percentage of the packets earlier than strictly needed and avoid congestion [9] after implementation in the derived model called the analytical model.

The new analytical model is the derivation of the Adaptive Congestion Control Mechanism Model formulated by Guan et al. [2] now called the previous model after introducing the optimum function in the state transition diagram.

The function should produce better performance metrics results at optimum RED parameter settings and to make a contribution to the research in performance modelling.

1.3Aim and Objectives

The aim of the project is to find an optimum mathematical function which give better performance metrics results particularly lower average queuing delay and high throughput [14] compared with others functions after implemented in the analytical model.

The new analytical model is derived from the previous model formulated by Guan et al [2] and the optimum function will be an input to the research in performance modelling.

The objectives will be the optimum dropping function should expect to produce the following results when the performance metrics expression is implemented in MATLAB program with a range of threshold values:

- **Achieve minimum packet dropping probability:**
  When the threshold values increases the optimum function should expect to produce low packets drop probability compared to other functions.
- **Achieve low propagation delay for maximum throughput in the network :**
  The optimum function should achieve low network delay at the maximum throughput compared with others functions.
- **Determine the optimum average queue size:**
  The calculated average queue size should expect to be the optimum approaches to the maximum threshold compared to other functions.

1.4 Structure of Dissertation

The remainder of this project is organised as follows: Chapter 2 contains detailed methodology for the implementation of the study, Chapter 3 deals with literature review, Chapter 4 performance modelling basics, Chapter 5 performance modelling and analysis of the project, Chapter 6 contain analysis of result and Finally Chapter 7 deal with conclusions and future works of the project.

Chapter 2: Methodology

This section explains detailed and systematic approaches used during the implementation of the project:

2.1 Analytical modelling

The proposed new analytical model is derive from the previous model formulated by Guan et al [2] after introducing a mathematical function to capture packets loss at the threshold value greater than or equal to L1 in the state transition diagram.

2.2 Performance matrices derivation

Subsequently from the new analytical model using virtual mathematics the balanced equations based on Markov chain state transition diagram , normalized equations based on equilibrium probability and initial state coefficient ($\pi 0$) computing formula from the normalized equations are derived.

In addition   mean queue length after applying first order derivative to the summation of both the equilibrium probability and generating function product, using Little's rules throughput and mean queue delay and finally, probability of packets loss computing formulas derivation are carry out.

2.3 Numerical results

The derived performance metrics mathematical formulas are validated by setting different values for RED parameters such as packet arrivals, departures, thresholds and for functions using MATLAB program. Numerical results are produced for comparisons to identify the optimum function.

2.4 Graphical analysis

Graphs for visualizing and comparing the mean queue length, throughput, mean queue delay and probability of packets loss against a range of thresholds values for different functions are generated using MATLAB program.

2.5 Summary of the results

Each performance metrics numerical and graphical results obtained using MATLAB program are compared for each of the functions used in the analytical modelling. Summary of the compared results, contribution to the research in performance modelling and the direction of future work are produced.

Chapter 3: Literature Review & Related works

This section deals with literature review of different research papers on the area of congestion control in both dynamic threshold and dropping function analysis in performance modelling of Adaptive Congestion Control Mechanism for the Internet traffic. The aim of the review is to appreciate the various research approaches used to prevent congestion with particular focus on achieving minimum packet loss via maximum throughput in the course of applications transmission over the Internet.

The researches are based on RED techniques which is an AQM method in heterogeneous traffic and working conditions. The following are some of the works done so far by different researchers:

3.1 Modelling with Discrete-Time Queue

Today, developments in practical world computers and communications systems are becoming more and more digital, or discrete- time in nature. Hence Michael E. Woodward wrote a book called modelling with discrete-time queue [1] which introduced the concept of developing accurate models of communications or computer networks based on discrete-time queuing theory which can be used to analyse the performance of a network.

In a continuous-time only a single state change can occur at any given time instant. This makes it difficult to apply the concept of performance modelling in the digitized computer or communication networks

However in the discrete-time because of the finite size of a time-unit, multiple state changes can occur from one time-unit to the next in digital forms. Performance modelling based on these techniques become very easy but needs conscious design concept to apply the techniques in the digitized computer or communication networks.

In addition some of the most important performance measure parameters of a communication or computer networks has been mentioned. Parameters such as throughput, message delay and probability of message loss are adopted as key performance measures for all network consideration. In particular probability of message loss can be used in assessing the transmission quality for certain types of data such as voice and video.

Further more he pointed out that sufficient state transitions descriptions using Markov chain process can be one way of modelling the performance of a digitized computer or communication networks.

Since many physical systems such as communication or computer networks operate on time-slotted basis and these can be conveniently modelled by discrete time-time Markov chain process. The process specifies a one-to-one correspondence between a time-slot in physical system and unit time in the model. Then performance measures for the system can be extracted from the equilibrium probability distribution of the Markova chain process.

The Markov chain state transition process has been illustrated by a discrete-time M/M/1/J queue system using Kendall’s notation in [13] with the assumptions that probability of packets arrival in the slot α, no packets arrival (1-α), the probability of packets departure in a slot be ß, no packets departure (1-ß) and α <ß.

Further more the queuing system in equilibrium and the state transition diagram had finite state space J (J packets or customer in a system) which satisfies the conditions to have a unique stationary probability distribution compared with infinite queuing process where the number of customers in the system will build up to infinity. This is shown by figure 1 below.

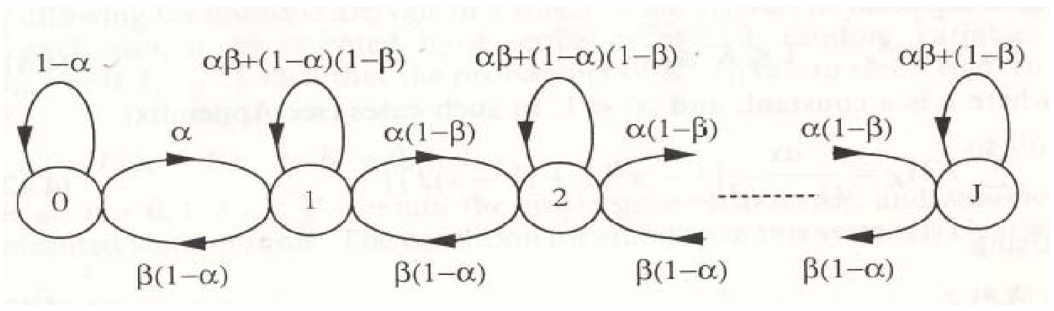

Figure 1: State transition diagram for a discrete-time M/M/1/J queue

Therefore, the above model provided vital information to both practitioners and researchers concerned with communication and computer networks performance evaluation.

And also today most researches in Adaptive Congestion Control for Internet traffic based on AQM techniques with finite buffer schema adopt the above model to measure the performance metrics of the system.

3.2 Dynamic threshold analysis

3.2.1 Congestion Avoidance Techniques Based On RED at the getaways

Sally Floyd and Van Jacobson proposed the RED gateways for congestion avoidance techniques in packet-switched Networks [6]. The gateways detect incipient congestion by computing the average queue size.

The average queue size, using a low pass filter with an exponential weighted moving average (EWMA) [4] equation has been calculated:

$$\mathbf{Avg(t + 1)= (1 - w_q).Avg(t) + w_q\, B(t)}$$

Where w_q is an averaging time constant, and B(t) is the instantaneous queue occupancy.

Equation 1: Exponential weighted moving averages

The average queue size is compared to two thresholds, a *minimum* (min_{th}) threshold and a *maximum* (Max_{th}) threshold [4] and proposed the following points:

- If the average queue size $\leq min_t$, then no incoming packets are dropped.
- If $min_t \leq$ average queue size $\leq Max_{th}$, then the arriving packets are dropped with probability P_b, where P_b, is a function of the average queue size.
- Finally, if the average queue size $> Max_t$ then all incoming packets are dropped, this is shown at figure 2 below.

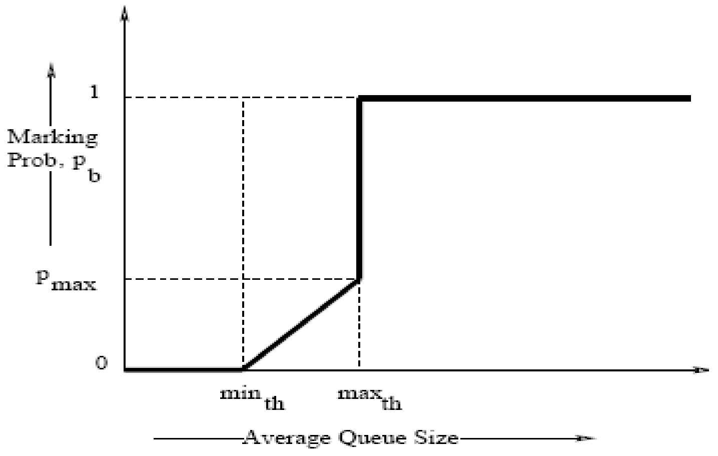

Figure 2: Dropping Probability Vs Average Queue Size of the RED algorithm

Each time a packet was marked, the probability that the marked packet was dropped for that particular connection roughly proportional to that connection's share of the bandwidth at the gateway. [6]

Two separate RED gateway algorithms were used in their proposal. The first algorithm to compute the average queue size that determined the degree of burstiness allowed in the gateway queue. The second algorithm to calculate the packet-marking/dropping probability determined how frequently the gateway marks/drops packets, given the current level of congestion.

The above two RED gateways algorithms efficient implementation as a congestion avoidance mechanism achieved the following major goals: [6]

- **Congestion avoidance: -** In RED gateway algorithm packets are dropped when average queue size exceed maximum threshold. However If the weight for the EWMA procedure has been set appropriately RED gateway guarantees that the calculated average queue size does not exceed the maximum threshold.

- **Appropriate time scales: -** In RED gateways, the time scale for congestion detection roughly matches the time scale required for connections to respond to congestion. RED gateways don't notify connections to reduce the traffic as a result of transient congestion at the getaway.

- **No global synchronization: -** RED gateways avoid global synchronization by assigning low probability of marking for each arriving packets in the event of low congestion and higher probability of marking for each arriving packet during higher congestion. The gateways avoid global synchronization by making packets at as low a rate as possible.

- **Fairness: -** RED gateway does not discriminate against particular connections or classes of connections. Packet marking for each connection is roughly proportional to that connection's share of the bandwidth. But do not attempt to ensure that each connection receives the same fraction of the total throughput.

- **Parameter sensitivity: -** RED gateways apply the following rules or assumption for parameters to give adequate performance under a wide range of traffic conditions:
  - Ensure adequate calculation of the average queue size: $w_q > 0.001$. The weight w_q should not be set too low, so that the calculated average queue length does not delay too long in reflecting increases in the actual queue length. Where w_q queue weight.
  - Set min_{th} sufficiently high to maximize network power: The thresholds min_{th} and max_{th} should be set sufficiently high to maximize network power.

- Make max_{th} - min_{th} sufficiently large to avoid global synchronization: As a rule of thumb usually max_{th} to be at least twice min_{th} would be.

However, in spite of the fact that RED is the most promising AQM [12] scheme for congestion avoidance and control, research has shown that the performance of RED is highly dependent upon the way its parameters are tuned and the network environment where it is used.

When maximum probability of marking (max*p)* is large and/or network is lightly congested, the average queue size is near min*th*; conversely when max*p* is small and/or the network is heavily congested, the average queue size is close to max*th*. Thus, the queuing delays at the routers cannot be easily estimated because the changes in the average queue size vary widely according to the parameters and congestion in the network. [12]

Finally they cited the following points as a future work of direction:

- Making conscious decisions and determination of optimal average queue size for maximizing throughput and minimizing delay for various network configurations.
- Traffic dynamic mix of TD and RED gateway implementation in the current Internet.
- Study the behaviour of RED gateway other than TCP protocols.
- Implementation of packet marking priorities based on the connection at the RED gateway.

3.2.2 Discrete-Time Performance Modelling Based On RED and Queue Threshold

Guan et al in [2] was implemented two discrete-time setting using queuing threshold congestion control analytical models for performance evaluation of Internet traffic.

The analytical models were based on RED mechanism, i.e. RED is recommended by the Internet Society in [3], which compared the performance metrics parameters against thresholds in each of the two implemented models.

The numerical analysis of the two models was conducted based on the assumption that departure always takes place before an arrival in any time unit or slot, arrivals follows an independent Bernoulli process, the system have finite waiting room or buffer space and the queuing discipline was First-come First-served (FCFS).

Model 1:

In model 1 there was a step reduction in the probability of arrival rate from α1 directly to α2 when the queuing reached at threshold value L1. However the source operates normally and a reduction in arrival rate achieved through implicit feedback from the queue to arrival process, the probability of a departure is β and arriving packets dropped with a probability 1-α2/α1 after threshold value L1.

Model 1 represented by figure-3 is shown below:

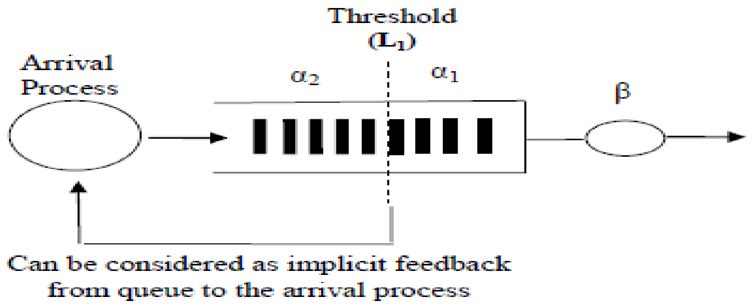
Threshold
(L_1)
Arrival
Process
α_2
α_1
β
Can be considered as implicit feedback
from queue to the arrival process

Figure 3: Single Buffer with One Threshold L_1

The corresponding state transition diagram of the above figure using Markov chain finite state space process is represented by figure 4 as shown below.

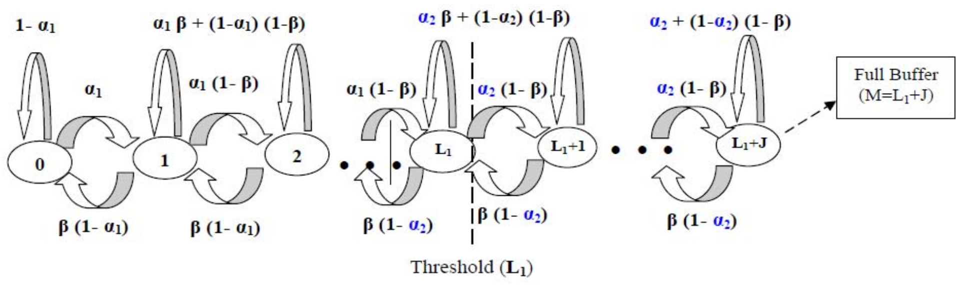

Figure 4: State Transition Diagram for Discrete-Time Finite Queue with Threshold value L_1

Subsequently from the state transition diagram assuming α1≠β, α2≠β, and α1> α2, the balanced equations, normalized equations, mean queue length, throughput, mean queue delay and probability of packet loss calculated and numerical results was generated [2].

Model 2:

Model 2 was slotted into two thresholds and probability of an arrival in a slot be α1 before the number of packets in the system reaches the first threshold L1, the probability of an arrival in a slot be reduced to α2 after the number of packets in the system reaches the second threshold, the probability of a departure be β and the dropping probability increased linearly form 0 to the maximum 1- α1/ α2 within the two thresholds.

Model 2 illustrated by figure 5 is shown below:

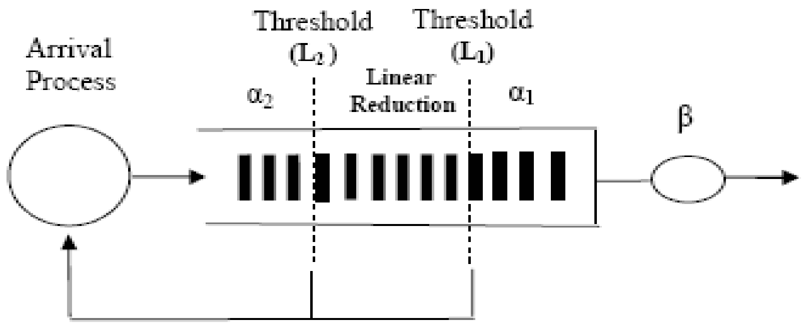

Figure 5: Single Buffer with Two Thresholds L_1 and L_2

The corresponding state transition diagram of the above figure using Markov chain with a finite state space process represented by figure 6 as shown below.

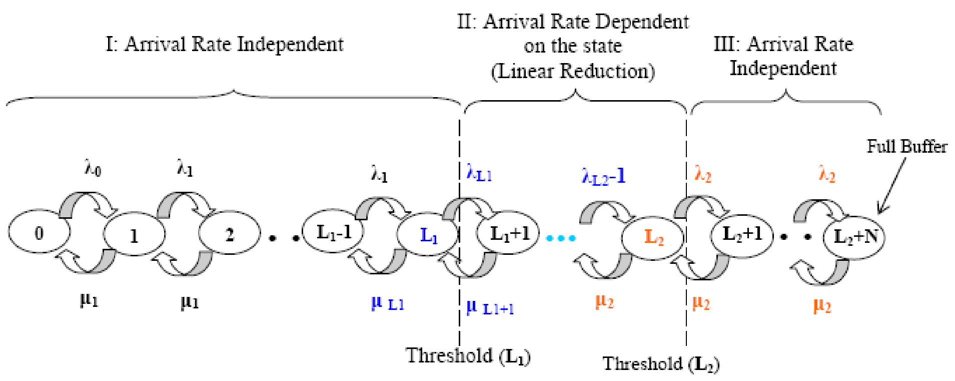

Figure 6: Double Threshold Transition Diagram for Discrete-Time Finite Queue

Therefore, from the state diagram assuming α1≠β, α2≠β, α1> α2 and full buffer (L2+N=M) situation the balanced equations, normalized equations, mean queue length, throughput, mean queue delay and probability of packet loss calculated and numerical results was generated [2].

The numerical result of model 1 produced a 'stepwise reduction' in probability of packets arrivals rate from α1 to α2. While a 'Linear Reduction' in probability of packet arrivals from α1 to α2 between the two thresholds L1 and L2 achieved in model 2. This is shown by figure 7 below.

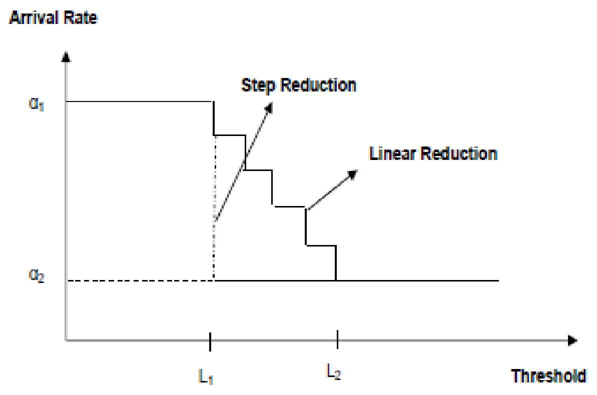

Figure 7: Step Reduction and Linear Reduction

From the overall numerical analysis Guan et al in [2] concluded that to achieve a lower delay for a specific probability of packet loss the following parameter settings should be used:

- A high maximum drop probability
- A low setting for the threshold
- A narrow separation of the threshold

Also to achieve a lower probability of packet loss the following parameters setting should be used:

- A low maximum drop probability
- A high setting for the threshold
- A wide separation of the threshold

Therefore, based on the type of services required such as real time and non-real time services, the above parameters setting should be adjusted.

Finally Guan et al in [2] suggested that future work should aim to generalize the results obtained to some extent by allowing multiple arrivals in a slot which can be applied to any arrival process. Furthermore, implementations of this model to Internet traffic e.g. a TCP/IP flow so that the technique of variable thresholds and blocking can be applied as a congestion control mechanism.

3.2.3 A New AQM Algorithm for Congestion Control

Alraddady, F. and Woodward, M. E [4] proposed a new AQM algorithm based on RED to control congestion on the Internet.

The algorithm was implemented by modifying RED and providing a flexible way of adjusting the maximum dropping probability between the two thresholds [4], improving throughput and delay compared to RED.

In RED algorithm [3], the probability of packet dropping increased linearly with the average queue size between the thresholds [2]. The slope of its packet dropping probability line depends on maximum packet dropping probability which has manually adjustable parameter. This is shown in figure 8 below.

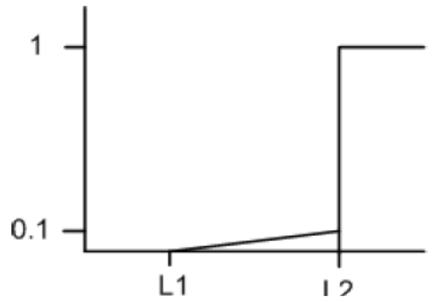

Figure 8: RED Algorithm

In contrast, the proposed model adjusted the slope of the line dynamically by adjusting maximum dropping probability depending on the average of incoming arrival rates. This is shown at figure 9 below. Where Pd is packet dropping probability, λ1 and λ2 arrival rate, L1 and L2 are thresholds.

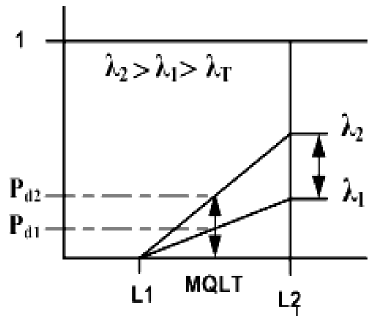

Figure 9: Proposed Algorithm

Therefore, the proposed model algorithm has variable packet dropping probability that reflects the slopes which was calculated on the bases of a linear equation [4] using Mean Queue Length Target (MQLT) and incoming packet arrival rate.

If there is a change between the current mean queue length and the target mean queue length, new maximum packet dropping probability was calculated which results in a new slope. And also moves the mean queue length back to the target mean queue length.

And also if there is an increase in arrival rate compared with the target arrival rate again, this also leads to new slope by calculating new maximum packet dropping probability.

The target mean queue length and the target average arrival rate of the system at the steady state was estimated from the know service rate and known the target delay by assuming a Poison source and using the M/M/1/K [12] model as in [4].

This was achieved by the following two steps:

- First, calculated the model the target average arrival rate for given service rate and the target delay at steady state after simulating the model M/M/1/K as in [4]
- Second, calculated the model the target average queue length at steady state for a given target average arrival rate and target delay after simulating the model M/M/1/K as in [4]

Then the average incoming arrival rate (AIAR) was calculated by using the weighted moving average for the delay of the system and the current queue length at every packet arrival and applying Little's law [1]. Next the dropping probability at the target mean queue length was calculated by using the target average arrival rate as specified by an equation (3) in [4].

The simulation result in [4] shows that for different arrival rate and fixed service rate the purposed model achieved lower delay, lower variance in average mean queue length and similar throughput compared with RED and Adoptive RED.

On the other hand the correlation effect of the throughput doesn't have significant effect on delay for Adoptive RED and the proposed model algorithm but have significant effect of increased correlation on increased delay on RED algorithm.

In general the performance of the proposed model is at least as good as Adoptive RED. But as to the original RED it has a problem of subject to further parameterization.

3.2.4 Analytical Modelling Based On Dynamic RED

Hussein Abdel-jaber, Mike Woodward, Fadi Thabtah and Mahmud Etbega [17] implemented a Discrete-Time Queue Analytical Model based on Dynamic Random Early Detection(DRED) technique to control congestion in the wireless and fixed network. They compared the results to the original DRED algorithm in terms of the performance metrics parameters.

The analytical model was enabled to accommodate single event i.e. arrival or departure of packet can take place in a slot or multiple event, where both arrival and departure could take place in the same slot with a finite capacity, including packets in the service.

The model has a single threshold (th) and packet arrival at each slot has identical independent distribution (I.I.D) Bernoulli process [1].The queuing discipline was FCFS. Figure 10 shows the queuing system of the proposed model.

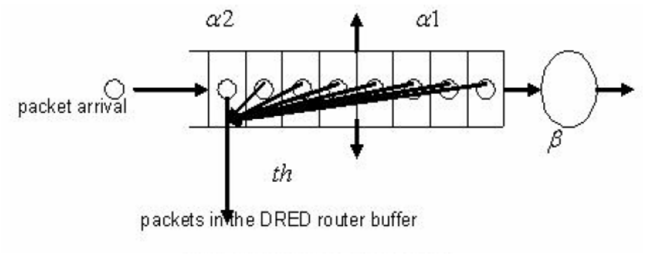

Figure 10: Queuing system model

To generate the state transition diagram they assumed packet transmitted at the rate α1 before the threshold value **th,** packet dropped probability was Dp=0. When the rate decreases from α1 into α2 after the threshold value **th**, the packet dropped probability increases from **0** to **(α1- α2)/ α1**. Average packet departure was represented by β in the model.

In addition they assumed that the analytical model queuing system in equilibrium and queue length process a Markov chain with finite state space (k). They generated the state transition diagram, as shown by figure 11 and the balanced equations from the diagram were created. [17]

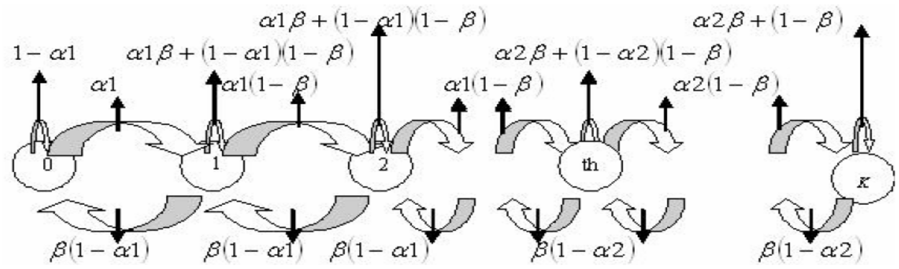

Figure 11: The state transition diagram for the DRED analytical model

Sequel to the balanced equation using virtual mathematics and Little's rule [1] they were generated performance metrics equations such as Average queue length (*aql)*, Throughput (T), Average queuing delay(*D),* Packet loss probability(P_{loss}) and Packet dropping probability (Dp) .[17]

An inputs value has been assigned for the analytical model parameters and calculated the numerical values and simulation graphs of the performance metrics. They also recorded the packet dropping probabilities results for both methods in order to evaluate which method drops fewer packets.

The simulation analysis as it was clearly shown with a diagram in[17] both method consistently produced similar results with regards to throughput(T) and packet loss

(P_{loss})When the traffic load increasing up to a certain value level. After that the original DRED algorithm dropped at a higher rate than the analytical DRED. Hence its throughput performance deteriorated.

However with regard to Average queue length (aql) and Average queuing delay (D) results, the original model DRED algorithm performed better than the analytical model DRED. [17]

With regards to packet dropping probability (Dp), the diagrams in [17] shows that the analytical model achieved smaller dropping probability than the original DRED algorithm.

Therefore the analytical model DRED technique throughput performance was sustained regardless of the traffic load rate as it was shown in [17] and produced better performance than the original DRED.

3.3 Dropping functions analysis

3.3.1 Analytical Modelling Based On Gentle-RED

LanWang, Geyong Min, Irfan Awan [8] were develop original analytical model of performance analysis using Gentle- RED (GRED-I) techniques of AQM [9] scheme under two heterogeneous classes of Internet traffic.

The traffics were non-bursty(e.g. Text data) and busrty (e.g. Web and voice) traffic classes which modelled with Poisson Process and Markov-Modulated Poisson Process (MMPP) [13] respectively.

The analytical model assumed to have MMPP average arrival rate, a single server, exponentially distributed service time for both traffic classes, and shared buffered management for the two traffic based on First-in-First-out (FIFO) queuing discipline. [9]

The threshold for each traffic class k (k = 1, 2) is denoted by th_k [8]. When the number of packets in the system exceeds threshold th_k, the forthcoming packets of Class-k will be dropped randomly depending on the dropping probability [8]. This is shown by figure 12 below.

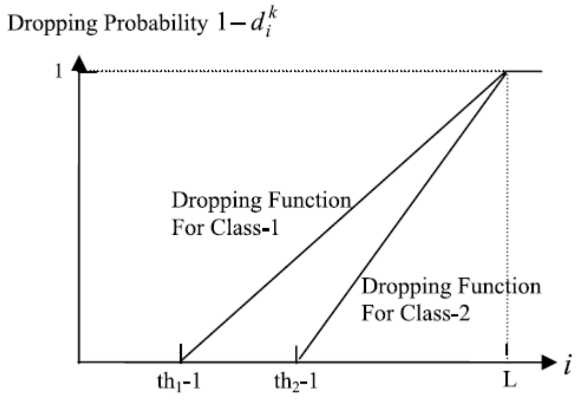

Figure 12: Dropping probability of for two traffic classes

The diagram shows that dropping probability increase linearly from 0 to the maximum dropping probability ($1 - d^k$max*)* which is set to 1.

Since data traffic was more sensitive to packet loss than bursty voice traffic, Class-1 traffic was dropped earlier than Class-2 in the presence of the sign of congestion. Thus, *th*1 was set to be less than *th*2.

Assuming the state transition diagram of the queuing system follows Markova chain to set up equilibrium equation. The probabilistic flow rate into and out of the system becoming at a certain state in equilibrium condition with the sum of probability is equal to 1. Derivation of performance metrics was performed [8].

Using virtual mathematics and Little's rules [1] different equations was derived to calculate the aggregate and marginal performance metrics such as mean queue length, response time, throughput, and loss probability. And also the accuracy of the analytical model was validated through extensive simulation experiments.

Comparisons of validating the analytical model to that of simulation was made and the following consistent results were obtained:

- The effects of the increase threshold th_1 under high bursty traffic was resulted an increase in the marginal mean queue length, mean response time and throughput but a decrease in loss probability. However at the low bursty traffic of the marginal loss probability increased as the result of growth of th1.

  The more common phenomena is that the marginal mean queue length, throughput, response time and loss probability of bursty traffic are become closer to those of low bursty traffic as threshold th1 decreases.

  On the other hand the aggregated utilization, mean queue length, throughput, and mean response time increase but the loss probability decreases as threshold $th1$ increases. The aggregated mean queue length, response time, and loss probability increase and the aggregated utilization and throughput decrease when traffic burstiness increases. Theses demonstrate the detrimental impacts of traffic burstiness on the performance of AQM mechanism. [8]

- The effects of the increased threshold th_2 with a fixed th_1 under low bursty traffic was resulted in an increase in the marginal mean queue length, mean response time and throughput but a decrease in loss probability.

  On the other hand throughput of bursty traffic tends to decrease and the mean queue length, loss probability and delay increase.

In addition the effect of th$_2$ on the marginal performance metrics for low bursty traffics more noticeable than high bursty traffics. It shows a more remarkable change when the rate of traffic is higher than when the traffic rate is lower.

- The effect of aggregate performance metrics when traffic rate increases resulted in an increase in utilization, mean queue length, throughput, mean responding time and loss probability as threshold th$_2$ increased.

  When the traffic rate decrease as threshold th$_2$ increased, there was an increase in mean queue length, throughput, mean responding time and a decrease in loss probability.

Therefore, the analytical model based on GRED-I in heterogeneous traffic environment validated using simulation and also illustrated with diagrams in [18]. The procedure used in the derivation of the model was general and can be easily extended for others AQM methods.

3.3.2 Analytical modelling and comparison of AQM based congestion control

An improvement in performance over the RED algorithm by comparing the effect of different probability of packet drop functions using instantaneous queue length model have been proposed by Lan Wange, Geyong Min, Itfan Awan.[18]

The traffic arrival process of the proposed model was two states Markov Modified Poisson Process (MMPP-2). This is shown by figure 13 below.

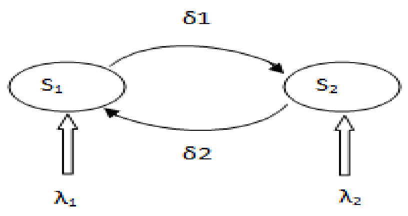

Figure 13: Two-state MMPP

The two different states S1 and S2 considered which represented two different traffic arrival rates λ_1 and λ_2 respectively. The intensities of the transition between S1 and S2 were represented by δ_1 and δ_2 respectively.

To maintain the main strategies of AQM that is to drop packet before congestion occurs, while the queue length is not smaller than the threshold value, different dropping functions have been introduced in the state transition diagram to achieve a step reduction of the arrival rate.

Theses process can be seen as a decrease of the arrival rate for each function with some probability ($1-d_{ij}$) after the threshold values. Where i,j mean number of packets in the queue and Markova state respectively.

From the above two state MMPP-2 diagram the corresponding state transition diagram in [18] were generated for calculating the performance metrics using virtual mathematics and equilibrium probabilities.

Five different functions have been used to compare their performance metrics with the change of threshold value with the same RED parameter setting.

The results of one threshold analysis clearly shows in [18] that with an increasing the threshold value mean queue length, throughput, and mean queue delay under each function are increases as expected and loss probability tends to decrease. In particularly the exponential function, $\mathbf{a^x}$**,** at **a=1.1** and with increasing threshold value, found to be an attractive function offer a better performance such as lower mean queue delay, higher throughput but less packet loss probability compared with others functions.

Two threshold analyses have been also show in [18] that the mean queue length, throughput, and mean queue delay are lower for the lower first threshold setting and increasing with enlarge the distance of two thresholds.

In general the review of the above literatures shows that RED or RED variant techniques of AQM methods are parameter sensitive and there will be a possibility of implementing those techniques through optimal setting of the parameters to achieved better results compared so far achieved.

However the optimum stetting of RED parameters required conscious design decision to achieve in minimum packet drop probability during congestion for variable Internet traffics.

In this project new analytical model from the previous model, the work done by Guan et al [2] derivation is made by introducing different packet dropping functions in the state transition diagram after the threshold value. The optimum function which produces better performance metrics results at optimum RED parameters settings should identified in order to make contribution to the research in performance modelling.

Chapter 4: Performance Modelling Basics

This section of the project deals with the fundamental properties of discrete-time queuing system that can be used in the actual implementation of the analytical model.

4.1 Queue Modelling and Fundamental Properties

4.1.1 Discreet-Time Queue

The new analytical model is based on discrete-time queue system which is one of the main mathematical techniques used to analyze packet arrival, packet departure, waiting time in the queue, and the number of servers in the system as it was used in the original model formulated by Guan et al [2]. The basic queuing system is shown by Figure 14 below.

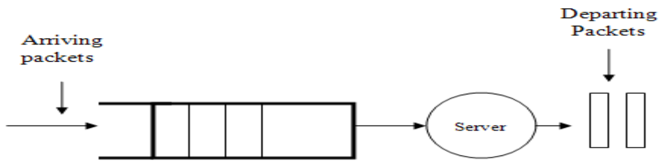

Figure 14: Basic queuing system

The new analytical model maintains the following important characteristic of a discrete-time queue system:

- The packet arrival process:

  It is assumed that the inter arrival times are follows identically independently distributed Bernoulli process,

- The service times:

  It is assumed that the service times are independent and identically distributed, and that they are independent of the inter arrival times.

- The service discipline:

  Packets are transmission based on First-come First-served discipline, i.e. in order of arrival.

  - Servers and service capacity: Single server and finite service capacity.

According to Kendall notation [12] the new analytical model classified as M/M/1/N=L1+J discreet-time queue system. Where M is stands for Markova (Memory less).

Since the model is a discrete-time queue system both inter arrival and service times are followed a geometrical distribution with zero or one arrival or service is permitted in a unit of time. This implies the arrival process is a Bernoulli process [1] , one server and having finite waiting room, represented by N=L1+J.[16]

The properties of the discrete-time queue queuing system is also enabled for the derivation and calculation of performance metrics measures such as mean queue length, mean waiting time, throughput and probability of packet loss [15] for the analytical model.

4.1.2 Performance Measures

The performance of the new analytical model assessed based on the three most commonly used parameters for measuring Internet traffics, namely, throughput, packet delay and probability of packet loss.

Throughput is the number of successful transmitted packet per mean transmission of packets, and packet delay is the time interval in unit of average transmission time of a packet from the moment a packet generated to the instant it is correctly received.
In this paper the mean value of throughput and packet delay are represented by the symbol S and W respectively.

Probability of packet loss, which is represented by PL in the model, is the fraction of packets that are lost due to no buffer space being available at the time of their arrival. [1]

4.1.3 Little's Law

In the new analytical model for calculating the performance metrics, Little's Law (1961) which is one of the basic theorems in queuing theory, is used. [1]

The mean number of packet in a stable system (over some time interval) is equal to their mean arrival rate, multiplied by the mean time in the system.

This is a general procedure adopted to measure the mean waiting time of any queuing system and the calculation depends only on mean values and not on distribution.

4.1.4 Discrete-Time Markov chain

The new analytical model is implemented by Markov chains discrete-time state transition processes either at each step the system may change its state from the current to another state, or remain in the same state, according to a certain probability distribution.

The change of states are called transition, and the probabilities associated with various state-change are called transition probabilities.

The model satisfies the Markov chain property as the evolution of the system after a given time instant depends only on the state at that instant and not on any past history.

A steady state (at equilibrium) analysis is conducted in order to derive the balance equilibrium as well as performance metrics expressions.

Chapter 5: Performance Modelling and Analysis

This section of the project deals with the actual implementation of the new analytical model derived from the previous model formulated by Guan et al [2] by carrying out the following series of steps:

5.1 Previous model

The previous model [2] was a discrete-time queuing system derived based on the Markova chain state transition process with the assumption that departure always takes place before arrival in any unit time. Each arrival follows an independent Bernoulli process with a finite buffer capacity (L1+J), the queuing discipline was First-come First-served (FCFS) and the system had a single server.

In addition the adopted model had one threshold value L1. α1 and α2 were packet arrivals before and after the threshold respectively. β was packet departure. There was a step reduction in the arrival rate from α1 to α2, when the number of packet reached at the threshold value L1. Alternatively this mean that the source continuing sending the packet at the rate equal to α1 but the arriving packet dropped with the probability 1- α2/ α1 [2]. The probabilities of no packet arrival and no packet departure are represented by (1- α1) and (1- β) respectively. The probability, no packet arrival at the threshold value greater than or equal to L1 is represented by (1- α2). This is shown in figure 15 below.

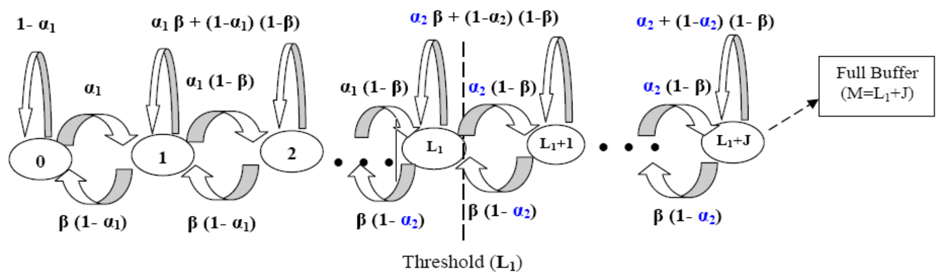

Figure 15: Discrete-Time Finite Queue with L1 Threshold state transition diagram

The new analytical model is the derivation of the above model after introducing different mathematical functions for capturing packet loss at the threshold value greater than or equal to L1 in the state transition diagram.

5.2 New Analytical model

In the new analytical model the probability of packet arrivals is represented by a1 and a2 before and after the threshold value L1 respectively and d for probability of packet departure. The maximum buffer capacity of the system represented by L1+J with a single server. The probability of no packet arrival and no packet departure is represented by ($1- a_1$) and ($1- d$) respectively. The probability no packet arrival at the threshold value greater than or equal to L1 represented by **($1-a_2$) = ($1- a1(1-D)$).** The Markov chain state transition diagram is depicted as follows in figure 16 below.

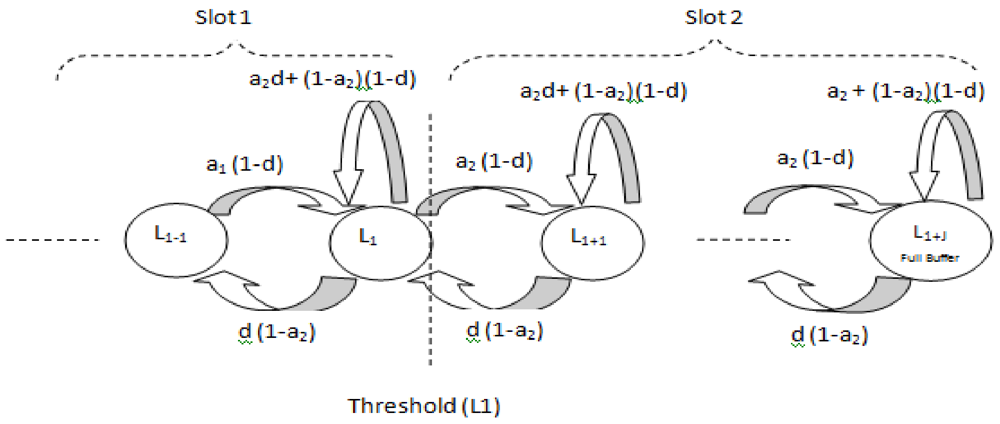

Figure 16: Single Threshold Analytical Model

When the capacity of the buffer level to accommodate packets reached at the threshold value greater than or equal to L1 some of the packets are dropped with a probability. This packet dropped is captured with a mathematical function represented by D. The packet arrival rate from **L1** up to **L1+J** which is a2 in the new analytical model is represent by an expression equal to **a1(1-D)**. This implies that there is a step reduction in the arrival rate from a1 to a2.

The diagram when the packets are dropped with a probability and how the packet loss is capture with an expression is shown below by figure 17.

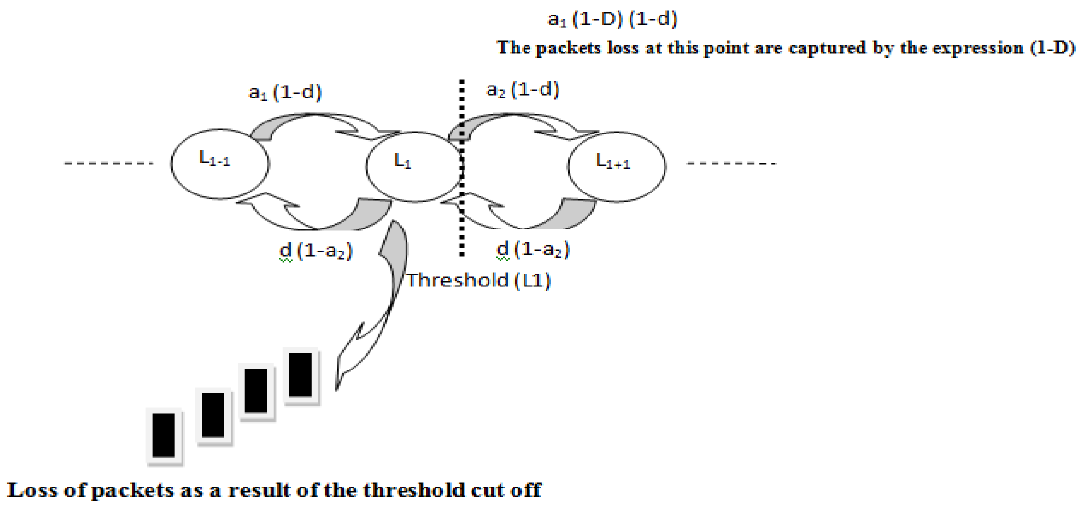
a₁ (1-D) (1-d)
The packets loss at this point are captured by the expression (1-D)
a₁ (1-d)
a₂ (1-d)
L₁₋₁
L₁
L₁₊₁
d (1-a₂)
d (1-a₂)
Threshold (L1)
Loss of packets as a result of the threshold cut off

Figure 17: Loss occurrences in the analytical Single Threshold Model

5.3 Derivation of Performance Metrics

To derive the performance metrics equations assume that the new analytical model utilised discrete-time queuing system which relies on the particular time unit called slot. The capacity of the model is finite, (L1+J) buffer space, which include packet in the service.

In addition the arrival process i.e.**(a_n)** , assume in the model is identically independently distributed Bernoulli process, **a_n €{0,1}, 0,1,2**,…., where **a_n** represent packet arrival at slot n. Then queuing discipline assume to be First-come First-serve (FCFS). [2, 17]

In order to derive the balanced equation and performance metrics measuring expression using virtual mathematics from the new analytical model state transition diagram the following sequence of steps are used:

I. From the new analytical model state transition diagram the following equilibrium probability equations are derived under the assumption **a1<d** (stable system):

$$\pi_0 = \pi_0(1-a1) + \pi_1(d(1-a1)) \quad (1)$$

$$\pi_1 = \pi_0 a1 + \pi_1[a1d + (1-a1)(1-d)] + \pi_2(d(1-a1)) \quad (2)$$

In general

$$\pi_i = \pi_{i-1}[a1(1-d)] + \pi_i[a1d + (1-a1)(1-d)] + \pi_{i+1}(d(1-a1)) \quad \text{For } i=2,3,\ldots L1-2 \qquad (3)$$

$$\pi_{L1-1} = \pi_{L1-2}[a1(1-d)] + \pi_{L1-1}[a1d + (1-a1)(1-d)] + \pi_{L1}[d(1-a1(1-D))] \qquad (4)$$

$$\pi_{L1} = \pi_{L1-1}[a1(1-d)] + \pi_{L1}[a1(1-D)d + (1-a1(1-D))(1-d)] + \pi_{L1+1}[d(1-a1(1-D))] \qquad (5)$$

$$\pi_{L1}+1 = \pi_{L1}[a1(1-D)(1-d)] + \pi_{L1+1}[a1(1-D)d + (1-a1(1-D))(1-d)] + \pi_{L1+2}[d(1-a1(1-D))] \qquad (6)$$

In general

$$\pi_i = \pi_{i-1}[a1(1-D)(1-d)] + \pi_i[a1(1-D)d + (1-a1(1-D))(1-d)] + \pi_{i+1}[d(1-a1(1-D))] \qquad (7)$$

For i=L1+2,L1+3,…,L1+J-1

$$\pi_i = \pi_{i-1}[a1(1-D)(1-d)] + \pi_i[a1(1-D) + (1-a1(1-D))(1-d)] \qquad \text{For } i=L1+J \qquad (8)$$

Equation 2: Equilibrium Equations

II. Solving the above equilibrium probability equations for each value of π_i ,For i= 1, 2, 3... L1+J with respect to π_0 using virtual mathematics values for $\pi_1, \pi_2, \pi_3, \ldots, \pi_{L1+J}$ are derived.

III. Using the normalized equation $\sum_{i=0}^{L1+J} \pi_i = 1$, and the results in step (ii) , we can solve the value for π_0 as:

$$\pi_0 = (1-d)(1-\gamma_1)(1-\gamma_2)(1-a1(1-D))/(1-a1(1-D))(1-\gamma_2)(1-\gamma_1^{L1}-d(1-\gamma_1)) + \gamma_1^{L1}(1-\gamma_1)(1-a1)(1-\gamma_2^{J+1})$$

Where $\gamma_1 = a1(1-d)/d(1-a1)$ and $\gamma_2 = a(1-D)(1-d)/d(1-a1(1-D))$

Equation 3: Initial coefficient (π_0)

IV. Using probability-generating function formula for finite queue length and substituting results obtained in step (II), we can calculate for the value of P (Z).

$$P(Z) = \sum_{i=0}^{L1+J} \pi_i Z^i \qquad \text{for } i= 1, 2, 3\ldots L1+J.$$

Equation 4: Generating Function

V. The first order derivative of the expression obtained in step (IV) evaluated at Z=1 produced an expression for calculating Mean Queue Length (MQL).

This is given by: MQL=P' (1)

$$P'(1) = \pi_0(\gamma_1 + \gamma_1^{(L1+1)}(L1-1) - L1\,\gamma_1^{L1})/(1-d)(1-\gamma_1)^2 +$$
$$\pi_0\,\gamma_1^{L1}(1-a1)(\gamma_2 + L1(1-\gamma_2) - \gamma_2^{(J+1)}(1+(L1+J)(1-\gamma_2))/(1-d)(1-a1(1-D))(1-\gamma_2)^2$$

Where $\gamma_1 = a1(1-d)/d(1-a1)$ and $\gamma_2 = a(1-D)(1-d)/d(1-a1(1-D))$

Equation 5: Mean Queue Length (P' (1))

VI. Using Little’s rules [1] we can generate formulas for mean waiting time and Throughput as follows:

- Waiting time (W) the time interval where an arbitrary customer enters into the system to the time the customer leaves, including the time spent in service.

$$W= P'(1)/S$$

Equation 6: Mean Waiting Time

- Throughput(S) is the number of customer passing through the system per unit time.

$$S= (1- \pi_0)\, d$$

Equation 7: Throughput

VII. The probabilities of packet loss, PL, the fraction of customers arrive to find no waiting room available to accommodate them. Of course this is true when the waiting room in the system is finite.

$$P_L = D\sum_{i=L1}^{L1+J-1}\pi_i + D\pi_{L1+J}d + \pi_{L1+J}(1-d)$$

Where i=L_1, L1+1,…, L1+J, D=Functions Introduced

Equation 8: Probability of Packet loss

The probability of packet loss increased from zero ‘0’ to ‘(1- D)’. Where D is the dropping function introduce in the P_L expression at the threshold L1 to capture the packet loss.

Chapter 6: Performance Evaluation

6.1 Typical Dropping Functions

In the new analytical model a step reduction in packet arrival rate from a1 to a2= a1(1-D) made by dropping some of the packets once the packet reached at threshold value greater than or equal to L1. This reduction in packet rate is achieved by introducing four different typical packet dropping functions represented by D to capture the packet loss in performance metrics expression.

The functions are:

Functions	Name and Properties
• Function One = $\log(x)^a$	Logarithmic function for a<0
• Function Two = a^x	Exponent function 0<a<1
• Function Three = a/x	Reciprocal of x function for all real values of a
• Function Four = x^a	Power of a function a<0

Equation 9: Dropping Functions

The graphical distribution of the four typical packet dropping functions against threshold value L1 are shown in figure 18.

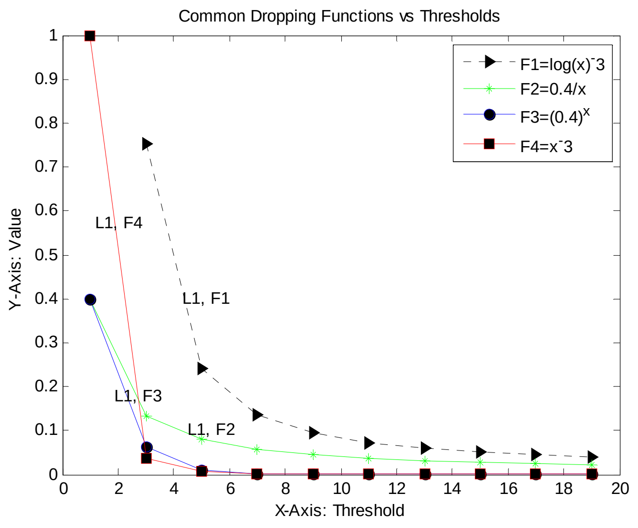

Figure 18: Typical Dropping Functions Vs Threshold (L1)

The diagram shows that the exponential function, $\mathbf{a^x}$, capture less packet dropping probability compared to the other functions.

6.2 Performance Metrics Analyzed Results

To investigate the performance metrics equation derived above using the above four typical packet dropping functions an experimental analysis is carried out using MATLAB so that numerical and graphical results are generated.

The new analytical model RED parameters input values for packet arrival, a1=0.27, packet departure, d=0.30, system capacity, Max queue length=20 and threshold, L1=[1,2,…,Max queue length] are used for measuring the performance metrics.

The numerical and graphical results of the performance metrics are shown below by the tables 1- 4 and figures 19-22 for each of the four different dropping functions.

Table 1: Probability of Packet Loss for Different Functions

Functions	Probability of Packet Loss									
F1=log(x)^-3	NaN	0.0585	0.0439	0.0346	0.0291	0.0257	0.0235	0.0219	0.0207	0.0199
F2=0.4/x	0.0514	0.0343	0.0269	0.0231	0.0208	0.0193	0.0183	0.0175	0.0169	0.0164
F3=(0.4)^x	0.0514	0.0243	0.0142	0.0125	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122
F4=x^-3	0.0609	0.0195	0.0138	0.0128	0.0125	0.0123	0.0123	0.0122	0.0122	0.0122

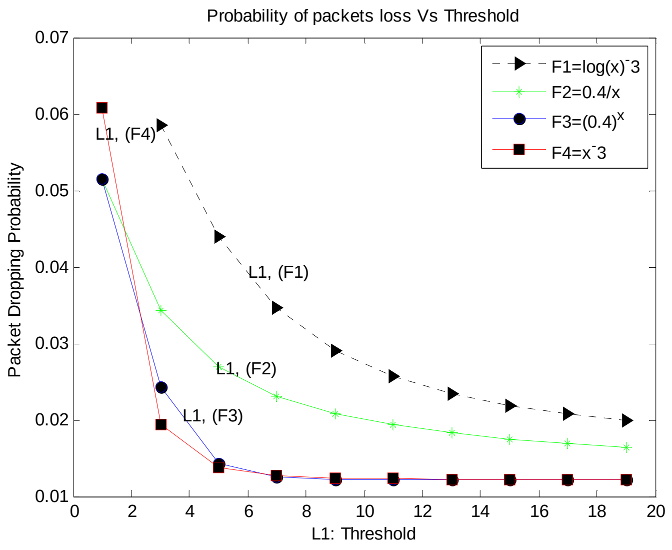

Figure 19: Probability of Packet loss Vs Threshold

The numerical values in the table-1 and figure 18 shows that the probabilities of packet loss for each of the function decrease as the threshold value L1 increases. Lower packet loss can be achieved by setting higher threshold value L1 and this is consistent to the normal system.

Table 2: Throughput for different functions

Functions	Throughput									
F1=log(x)^-3	NaN	0.9447	0.9510	0.9552	0.9577	0.9593	0.9604	0.9613	0.9619	0.9623
F2=0.4/x	0.9478	0.9553	0.9587	0.9606	0.9618	0.9626	0.9632	0.9636	0.9640	0.9642
F3=(0.4)^x	0.9478	0.9600	0.9655	0.9665	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667
F4=x^-3	0.9437	0.9625	0.9657	0.9664	0.9665	0.9666	0.9667	0.9667	0.9667	0.9667

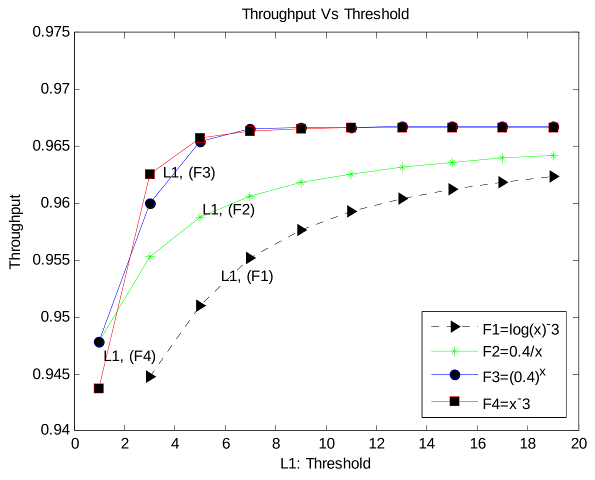

Figure 20: Throughput Vs Threshold

Table 3: Mean Queue Length for different functions

Functions	Mean Queue Length									
F1=log(x)^-3	NaN	0.5336	0.9613	1.3654	1.7755	2.1044	2.4121	2.7046	2.9821	3.2440
F2=0.4/x	0.1869	0.4861	0.8302	1.2098	1.6100	1.9782	2.3304	2.6675	2.9888	3.2933
F3=(0.4)^x	0.1869	0.4411	0.7025	1.0363	1.4110	1.8028	2.1983	2.5884	2.9673	3.3305
F4=x^-3	0.1688	0.4160	0.6974	1.0410	1.4166	1.8068	2.2005	2.5895	2.9676	3.3304

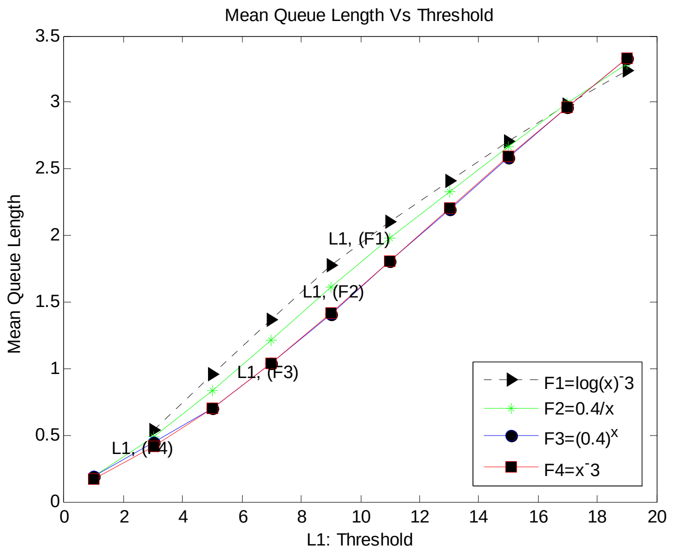

Figure 21: Mean Queue Length Vs Threshold

Table 4: Delay for different functions

Functions	Delay									
F1=log(x)^-3	NaN	0.5648	1.0108	1.4295	1.8540	2.1936	2.5114	2.8136	3.1003	3.3709
F2=0.4/x	0.1972	0.5088	0.8660	1.2593	1.6740	2.0550	2.4194	2.7682	3.1005	3.4154
F3=(0.4)^x	0.1972	0.4595	0.7277	1.0722	1.4597	1.8649	2.2740	2.6776	3.0694	3.4452
F4=x^-3	0.1789	0.4322	0.7222	1.0772	1.4657	1.8691	2.2764	2.6788	3.0699	3.4451

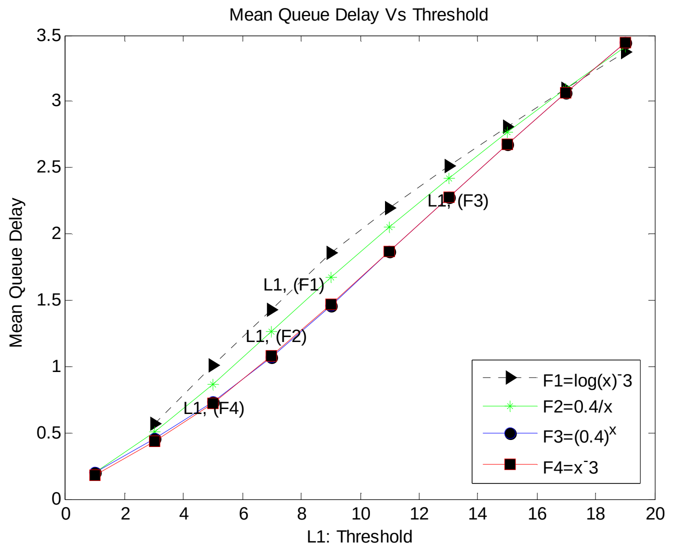

Figure 22: Mean Queue Delay Vs Threshold

Tables 2-4 and figures 20-22 above depict that with the threshold values increasing the throughput, mean queue length and mean delay for each function are increasing as expected. On the other hand as the dropping area size is decreasing the effort done by the router to avoid congestion are less and less the three performance metrics values increased reasonably.

The above numerical and graphical results clearly show that out of the four different functions the exponential function, **a^x** , the power of a function, **x^a**, exhibit lower probability of packet loss, higher throughput and lower delay compared with the other two functions.

To identify the optimum function from the exponential function, **a^x** , the power of a function, **x^a**, a similar experimental analysis as above are conducted the following numerical and graphical results shown by the tables 5-8 and figures 23-26 generated.

Table 5: Probability of packet loss for exponential and power functions

Functions	Probability of Packet Loss									
F3=(0.4)^x	0.0514	0.0243	0.0142	0.0125	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122
F4=x^-3	0.0609	0.0195	0.0138	0.0128	0.0125	0.0123	0.0123	0.0122	0.0122	0.0122

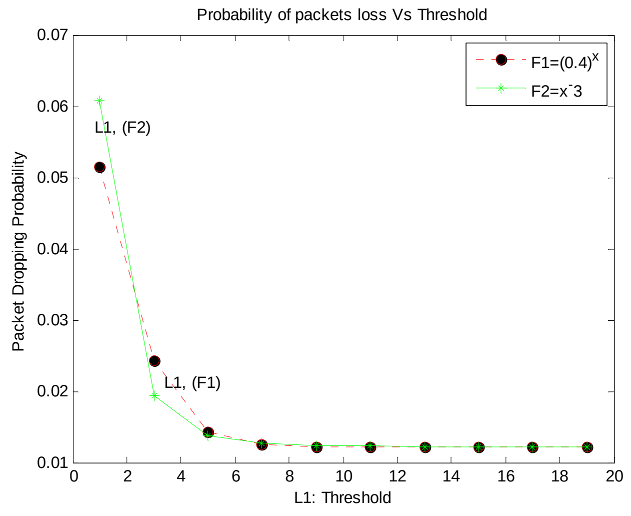

Figure 23: Probability of Packet loss Vs Threshold

Again similar to the above results table 5 and figure 23 shows that when the threshold values increases probability of packet loss decreases as expected.

Table 6: Throughput for exponential and power functions

Functions	Throughput									
F3=(0.4)^x	0.9478	0.9600	0.9655	0.9665	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667
F4=x^-3	0.9437	0.9625	0.9657	0.9664	0.9665	0.9666	0.9667	0.9667	0.9667	0.9667

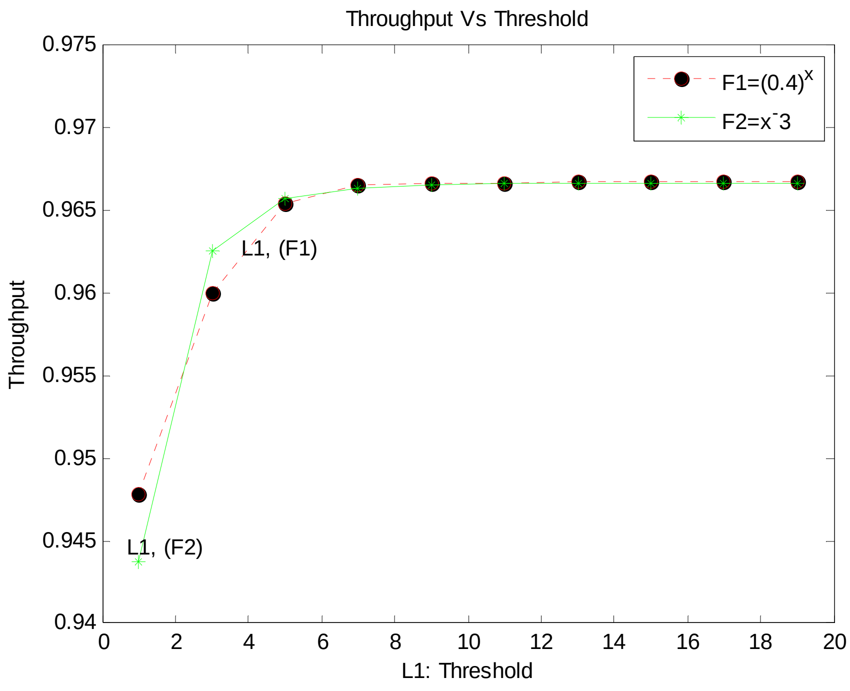

Figure 24: Throughput Vs Threshold

Table 7: Mean Queue Length for exponential and power functions

Functions	Mean Queue Length									
F3=(0.4)^x	0.1869	0.4411	0.7025	1.0363	1.4110	1.8028	2.1983	2.5884	2.9673	3.3305
F4=x^-3	0.1688	0.4160	0.6974	1.0410	1.4166	1.8068	2.2005	2.5895	2.9676	3.3304

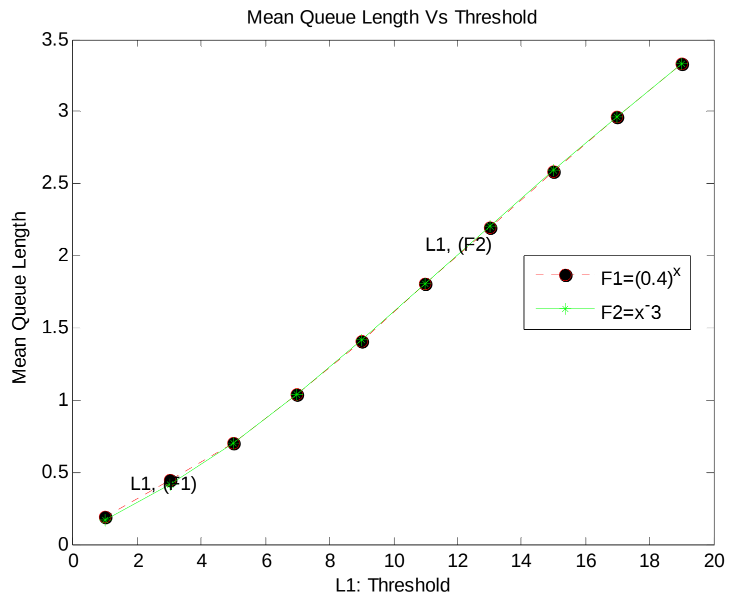

Figure 25: Mean Queue Length Vs Threshold

Table 8: Delay for exponential and power functions

Functions	Delay									
F3=(0.4)^x	0.1972	0.4595	0.7277	1.0722	1.4597	1.8649	2.2740	2.6776	3.0694	3.4452
F4=x^-3	0.1789	0.4322	0.7222	1.0772	1.4657	1.8691	2.2764	2.6788	3.0699	3.4451

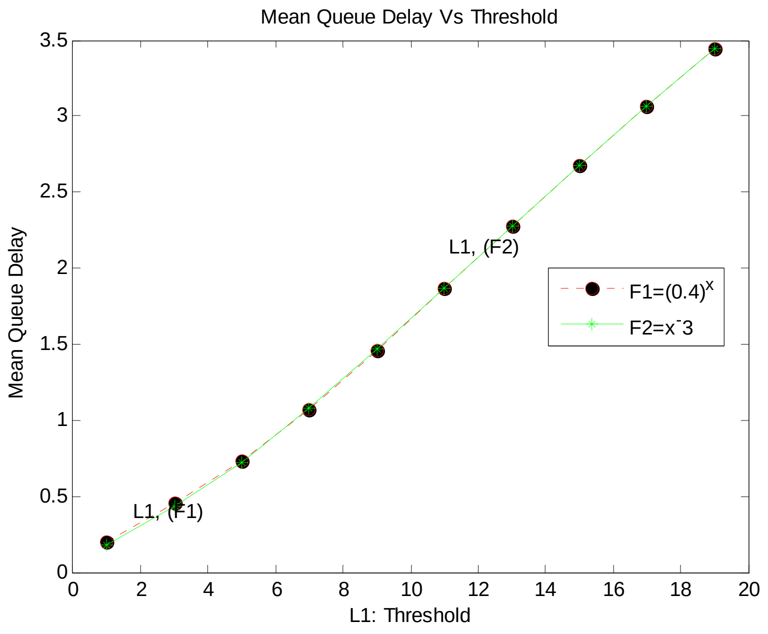

Figure 26: Mean Queue Delay Vs Threshold

Tables 6-8 and figures 24-26 above show that with the threshold values increasing the throughput, mean queue length and mean delay for the two functions are increasing. This is consistent with the normal system.

The above numerical and graphical results clearly show that out of the two functions exponential function, **a^x** , achieved lower probability of packet loss, higher throughput and lower delay compared with the power of a function, **x^a**.

Finally in order to draw a general conclusion about the exponential function, **a^x** , at different parameter values of **a,** experimental analysis is carried out and numerical and graphical results are shown by tables 9-12 and figures 27-30 are drawn.

Table 9: Probability of packet loss for different values of exponential functions

Functions	Probability of Packet Loss									
F1 =(0.01)^x	0.0142	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122
F2=(0.20)^x	0.0409	0.0138	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122
F3=(0.40)^x	0.0514	0.0243	0.0142	0.0125	0.0122	0.0122	0.0122	0.0122	0.0122	0.0122
F4= (0.60)^x	0.0562	0.0422	0.0266	0.0177	0.0142	0.0129	0.0124	0.0123	0.0122	0.0122

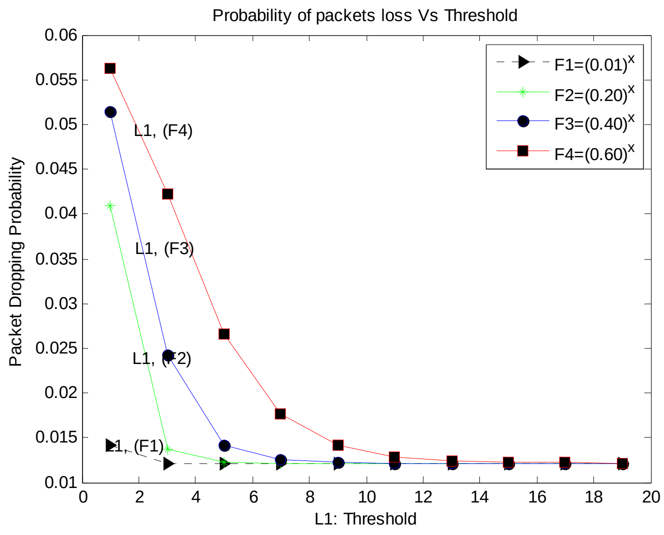

Figure 27: Probability of Packet loss Vs Threshold

Table 10: Throughput for different values of exponential functions

Functions	Throughput									
F1 =(0.01)^x	0.9655	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667
F2=(0.20)^x	0.9523	0.9657	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667
F3=(0.40)^x	0.9478	0.9600	0.9655	0.9665	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667
F4= (0.60)^x	0.9457	0.9518	0.9589	0.9635	0.9655	0.9663	0.9666	0.9667	0.9667	0.9667

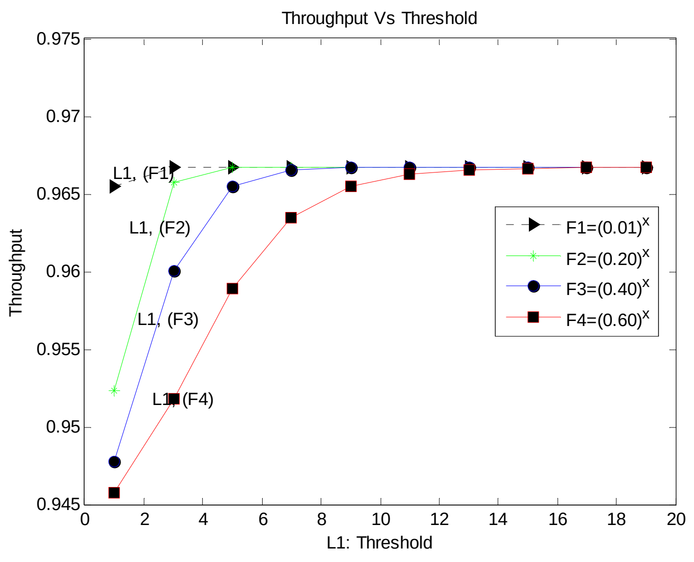

Figure 28: Throughput Vs Threshold

Table 11: Mean Queue Length for different values of exponential functions

Functions	Mean Queue Length									
F1 =(0.01)^x	0.1412	0.3729	0.6786	1.0303	1.4097	1.8026	2.1982	2.5884	2.9673	3.3305
F2=(0.20)^x	0.1824	0.3831	0.6793	1.0303	1.4097	1.8026	2.1982	2.5884	2.9673	3.3305
F3=(0.40)^x	0.1869	0.4411	0.7025	1.0363	1.4110	1.8028	2.1983	2.5884	2.9673	3.3305
F4= (0.60)^x	0.1826	0.5149	0.8270	1.1261	1.4597	1.8224	2.2048	2.5902	2.9675	3.3305

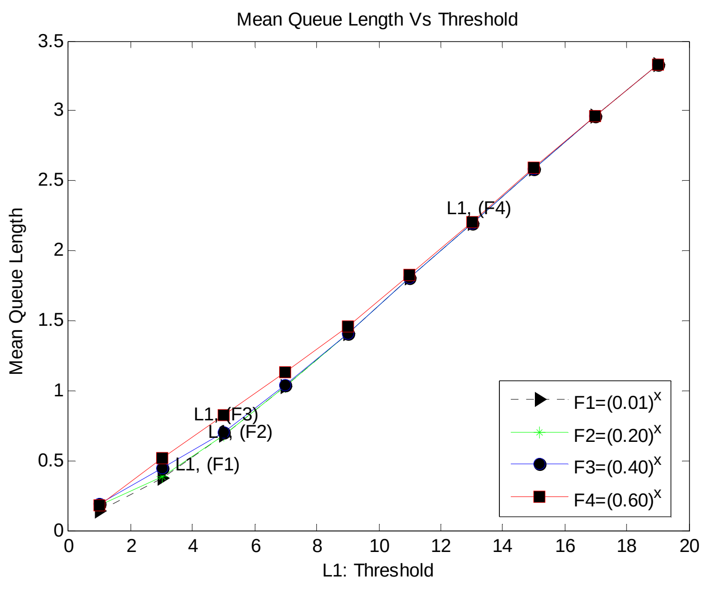

Figure 29: Mean Queue Length Vs Threshold

Table 12: Delay for different values of exponential functions

Functions	Delay									
F1 =(0.01)^x	0.1463	0.3857	0.7019	1.0657	1.4582	1.8647	2.2739	2.6776	3.0694	3.4452
F2=(0.20)^x	0.1915	0.3966	0.7028	1.0658	1.4582	1.8647	2.2739	2.6776	3.0694	3.4452
F3=(0.40)^x	0.1972	0.4595	0.7277	1.0722	1.4597	1.8649	2.2740	2.6776	3.0694	3.4452
F4= (0.60)^x	0.1931	0.5410	0.8625	1.1688	1.5119	1.8861	2.2811	2.6796	3.0694	3.4452

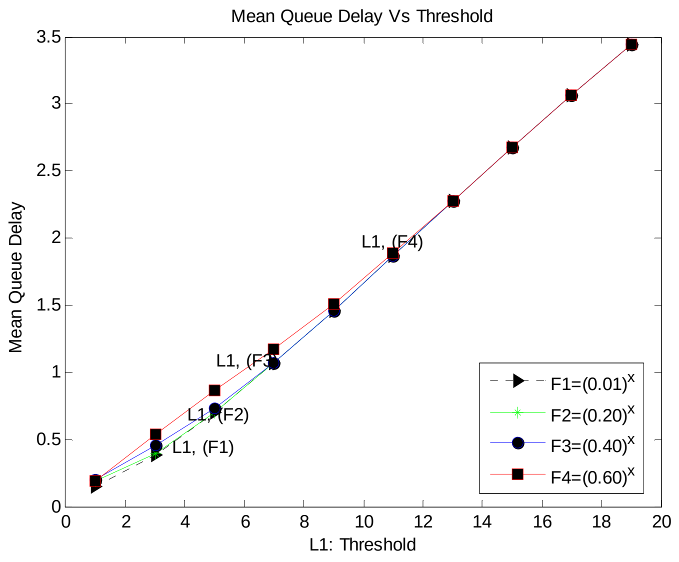

Figure 30: Mean Queue Delay Vs Threshold

Tables 9-12 and figures 27-30 clearly show that when the values of the parameter **a** approaches **0.01** the exponential function achieved lower packets loss probabilities, higher throughput and lower delay as the values of the threshold increased. These settings of the parameters can be used for data service application. While the value of the parameter **a** approaches to **1,** the settings can be used for real time applications that required lower delay.

Therefore the above results are consistent with the normal system, and the exponential function based on the above input values and parameter setting is considered to be an attractive function for AQM method and hence, this is a contribution to the research on performance modelling.

Chapter 7: Conclusion and Future Works

7.1 Conclusions

In this project, new analytical modelling based on the previous model to investigate optimum packet dropping function using virtual mathematical techniques in performance modelling has been conducted.

The new analytical modelling approaches are based on RED techniques of AQM methods with a single threshold; discrete-time queuing system, identically independently distributed packets arrivals and departures based on Bernoulli process and maintain Markov chain state transition process.

Performance metrics such as probability of packet loss, average queue length, throughput, and average queuing delay are derived for different functions and investigated the accuracy of the model through extensive experiments. To compare the parameters at the steady state, fixed input value for packet arrival and departure while changing the thresholds values are used.

The results showed that exponential function is an optimum function which achieved lower probability of packet loss, higher throughput and lower delay compared with others functions when the values of the thresholds are increasing.

7.2 Future Work

The extension of this project work will be:

- The new analytical model is implemented based on a single threshold value and achieved the above results. Two thresholds values implementation is required since RED is implemented by two thresholds values in the Previous model.

- All packet dropping functions are not examined on these paper. Much better packet dropping function could be investigated.

Reference

1. Woodward, M. E. 1993, *Communication and Computer Networks: Modelling with discrete – time Queues*, 1st Edition, Graham Lodge, England, Pentech 1- 92.

2. Lin Guan, Mike E Woodward, Irfan U Awan 2006, *A discrete-time performance model for Congestion control mechanism using queue Thresholds with QoS constraints.* Available online: http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-6/No.1-2/Lin-Guan.pdf [24 July 2008].

3. Serhat ÖZEKES* 2005, *Evaluation of Active Queue Management Algorithms.* Available on line: http://www.iticu.edu.tr/kutuphane/dergi/f7/M00105.pdf [26 July 2008].

4. F.AL-Raddady and Mike Woodward 2007, *A new active congestion control mechanism for the Internet*. Available on line: http://www.comp.brad.ac.uk/het-net/tutorials/WP06.pdf [20 March 2008].

5. Bartek Peter Wydrowski 2003, *Techniques in Internet Congestion Control*. Available on line: www.ee.unimelb.edu.au/multimedia/research/cubin_BartekWydrowski_Thesis.pdf [26 July 2008].

6. S. Floyd and V. Jacobson 1993, 'Random early detection gateways for congestion avoidance', *ACM Transactions on Networking*, vol. 1, pp. 397-413.

7. B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang 1998, 'Recommendations on Queue Management and Congestion Avoidance in the Internet', *The Internet Society*, Request for Comments RFC 2309.

8. LanWang, Geyong Min.and Irfan Awan 2007, *Performance analysis of buffer allocation schemes under MMPP and Poisson traffic with individual thresholds.* Available on line: http://www.springerlink.com/content/h064260x32430866/fulltext.pdf [11 July 2008].

9. May, M., Diot, C., Lyles, B.and Bolot, J. 2000, 'Influence of active queue management parameters on aggregate traffic performance', *INRIA*.RR3995.

10. Gonzalo R. Arce 2003, 'RED Gateway Congestion Control Using Median Queue Size Estimates', *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, VOL. 51, NO. 8.

11. ASFAND-E-YAR, I.U.AWAN and M.E.WOODWARD, *Performance Modelling Of A Multiple Threshold RED Mechanism for Bursty and Correlated Internet Traffic With MMPP Arrival Process*. Available on line: http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-7/No-1/paper-3.pdf [21 July 2008].

12. Sanjay K.Bose 2002, *Basic Queuing Theory M/M/_/_Type Queues*. Available online: http://www3.ntu.edu.sg/home/eskbose/qbook/Slide_Set_3.PDF [24 July 2008].

13. Fischer,W.,Merier-Hellstern, K.1993, 'The Markov-modulated Poisson process (MMPP) cookbook', *Perform. Evaluation* 18(2), 149–171.

14. Sally Floyd, Ramakrishna Gummadi, and Scott Shenker 2001, *Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management*. Available online: http://www.icir.org/floyd/papers/adaptiveRed.pdf [7 August 2008].

15. Wikipedia, Available online: http://en.wikipedia.org/wiki/Queueing_theory [16 August, 2008].

16. EventHelix 2000-2008, Queuing Theory. Available online: http://www.eventhelix.com/realtimemantra/CongestionControl/queueing_theory.htm. [16 August 2008].

17. Abdel-jaber, Hussein; Woodward, Mike; Thabtah, Fadi and Etbega, Mahmud 2007, 'A Discrete-time Queue Analytical Model based on Dynamic Random Early Drop' *Information Technology*, 2007. *ITNG '07. Fourth International Conference on* 2-4, Page(s):71 – 76.

18. Lan Wang,Geyong Min and Iran Awan 2005,'Analytical Modeling and Comaprison of AQM-Based Congestion control Mechanisms', *HPCC , LNCS 3726*, pp. 67 – 76.

19.

Appendix A –MATLAB Codes

```
% $$$$$$$$$$$$$$$$$$$ Declaring Variables Main Method $$$$$$$$$$$$$$$$$$$$$
clear all;

MaxQueueLength=20;
V=1;

 for  x=(1:2:20);

%***************Part I: General Function Behaviour************************
%                     L3=log(x)^-3;
%                     L4=0.4/x ;
%                     L5=(0.4)^x;
%                     L6=x^-3;
%***************Part II: Specific Function Behaviour*********************

%                   L3=(0.4)^x;
%                   L4=x^-3;
%
%*****************PartIII: Exponential Functions Behaviour***************

                L3=(0.01)^x;
                L4=(0.20)^x;
                L5=(0.40)^x;
                L6=(0.60)^x;
%
% *******************************Mean Queue Length************************
         [mql1] = MQL2(L3,V);
         [mql2] = MQL2(L4,V);
         [mql3] = MQL2(L5,V);
         [mql4] = MQL2(L6,V);


            MQLF1(V) = mql1;
            MQLF2(V) = mql2;
            MQLF3(V) = mql3;
            MQLF4(V) = mql4;

%****************************Probability of Packet Loss********************
         [pkl1] = PKL(L3,V);
         [pkl2] = PKL(L4,V);
         [pkl3] = PKL(L5,V);
         [pkl4] = PKL(L6,V);


                PKL1(V) = pkl1;
                PKL2(V) = pkl2;
                PKL3(V) = pkl3;
                PKL4(V) = pkl4;

%********************************Throughput********************************
         [thrp1] = THRP(L3,V);
         [thrp2] = THRP(L4,V);
         [thrp3] = THRP(L5,V);
         [thrp4] = THRP(L6,V);
```

```
                 THRP1(V) = thrp1;
                 THRP2(V) = thrp2;
                 THRP3(V) = thrp3;
                 THRP4(V) = thrp4;

%********************************Mean Queue Delay*************************
        [dly1] = DELY(L3,V);
        [dly2] = DELY(L4,V);
        [dly3] = DELY(L5,V);
        [dly4] = DELY(L6,V);

            DELY1(V)= dly1;
            DELY2(V)= dly2;
            DELY3(V)= dly3;
            DELY4(V) =dly4;

        V =V+1;
 end
%     fn = (1:2:20) ;
    L1=1:2:MaxQueueLength;

 %*************************Mean Queue Length*******************************
                MQLA=MQLF1
                MQLB=MQLF2
                MQLC=MQLF3
                MQLD=MQLF4

%*********************Probability of Packet Loss*************************
                PKLA=PKL1
                PKLB=PKL2
                PKLC=PKL3
                PKLD=PKL4
%    ************************Throughput***********************************
                THRPA=THRP1
                THRPB=THRP2
                THRPC=THRP3
                THRPD=THRP4

% %***********************Mean Queue Delay*********************************
                DELYA=DELY1
                DELYB=DELY2
                DELYC=DELY3
                DELYD=DELY4

%**********************Plot for Mean Queue Length************************

%  plot(L1,MQLA,':k>',L1,MQLB,'-g*',L1,MQLC,'-bo',L1,MQLD,'-rs'...
%  ,'LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
% plot(L1,MQLA,':ro',L1,MQLB,'-
g*','LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
% xlabel('L1: Threshold')
% ylabel('Mean Queue Length ')
% gtext('L1, (F1) ')
% gtext('L1, (F2)')
% gtext('L1, (F3)')
% gtext('L1, (F4)')
```

```
%******************Plot for Probability of Packet Loss******************

 plot(L1,PKLA,':k>',L1,PKLB,'-g*',L1,PKLC,'-bo',L1,PKLD,'-rs'...
     ,'LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
%  plot(L1,PKLA,':ro',L1,PKLB,'-
g*','LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
xlabel('L1: Threshold')
ylabel('Packet Dropping Probability')
gtext('L1, (F1) ')
gtext('L1, (F2)')
gtext('L1, (F3)')
gtext('L1, (F4)')

% %**********************Plot for Throughput*************************

%  plot(L1,THRPA,':k>',L1,THRPB,'-g*',L1,THRPC,'-bo',L1,THRPD,'-rs'...
% ,'LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
% plot(L1,THRPA,':ro',L1,THRPB,'-
g*','LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
% xlabel('L1: Threshold')
% ylabel('Throughput')
% gtext('L1, (F1) ')
% gtext('L1, (F2)')
% gtext('L1, (F3)')
% gtext('L1, (F4)')

%**********************Plot for Mean Queue Delay*************************

%  plot(L1,DELYA,':k>',L1,DELYB,'-g*',L1,DELYC,'-bo',L1,DELYD,'-rs'...
% ,'LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
% plot(L1,DELYA,':ro',L1,DELYB,'-
g*','LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
% xlabel('L1: Threshold')
% ylabel('Mean Queue Delay ')
% gtext('L1, (F1) ')
% gtext('L1, (F2)')
% gtext('L1, (F3)')
% gtext('L1, (F4)')




function [mql1]=MQL2(L3,L1)

% $$$$$$$$$$$$$$$$$$$ Declaring Variables For Function MQL2 Method $$$$$$$$
a1=0.27;         %a=Arrival_one Value
D=0.30;          %D=Departure Value
 Q=5 ;           %Value for the threshold
N2=15 ;
if L1<=5
  J=0;
else
       J=L1-5;
       L1=5;
  end

AwnD1=a1*(1-D); % AwnD1=AwnD_one=Arrival_one*(1-Departure)
Dwna1=D*(1-a1); % Dwna1=Dwna_one=Departure*(1-Arrival_one)
```

```
a=[[[a1^4]*[1-D]^3]/[[D^4]*[1-a1]^4]]; % coefficient of Pai4 = PL1-1
b=[[[a1^3]*[1-D]^2]/[[D^3]*[1-a1]^3]]; % coefficient of Pai3 = PL1-2
c=[[a1+D]-[2*a1*D]] ;
d=[a1-[a1*D]];
e=[D-[a1*D]];
g=[[D-[a1*D]+[a1*D]*L3]] ;
h=[[[[a1+D]-[2*a1*D]]-[a1*L3]]+[2*a1*D*L3]];
i=[[[a1-[a1*D]]-[a1*L3]]-[a1*D*L3]];

%$$$$$$$$$$$$$$$$$$ Get the coefficients of the States $$$$$$$$$$$$$$$

p0=1  ;                %Coefficient of P0
p1=[[a1]/[D*[1-a1]]];  %Coefficient of P1

for r=2:Q-1
p2to(r)=[[[a1^r]*[[1-D]^[r-1]]]/[[D^r]*[1-a1]^r]];%coefficient of p2topL1-1
end

pL1=[a*d]/g ;              % Coefficient at L1
pL1plus1=[[a*i*d]/g]/g;    % Coefficient at L1+1
pL1plus2=[pL1plus1*i]/g;   % Coefficient at L1+2
pL1plus3=[pL1plus2*i]/g;   % Coefficient at L1+3
pL1plus4=[pL1plus3*i]/g ;  % Coefficient at L1+4
pL1plus5=[pL1plus4*i]/g ;  % Coefficient at L1+5
pL1plus6=[pL1plus5*i]/g ;  % Coefficient at L1+6
pL1plus7=[pL1plus6*i]/g ;  % Coefficient at L1+7
pL1plus8=[pL1plus7*i]/g;   % Coefficient at L1+8
pL1plus9=[pL1plus8*i]/g ;  % Coefficient at L1+9
pL1plus10=[pL1plus9*i]/g;  % Coefficient at L1+10

%$$$$$$$$$$$$$$$$$$$$$$$$ Get Pai0 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

part1=[p0+p1]+[sum(p2to)];
part2=[pL1+pL1plus1+pL1plus2+pL1plus3+pL1plus4];
part3=[pL1plus5+pL1plus6+pL1plus7+pL1plus8+pL1plus9+pL1plus10];
bottom=[part1+part2+part3];

Pai0=[[1]/[bottom]];

% $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MQL $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

Part1=[Pai0*[[d/e]+[[d/e]^[L1+1]*[L1-1]]-[L1*[d/e]^L1]]]/[[1-D]*[[1-
[d/e]]^2]];
Part2=[Pai0*[[[d/e]^L1]*[1-a1]*[[i/g]+[L1*[1-[i/g]]]]]]/[[1-D]*[1-
a1+[a1*L3]]*[[1-[i/g]]^2]];
Part3=[Pai0*[[d/e]^L1]*[1-a1]*[[i/g]^[J+1]]*[1+[[L1+J]*[1-[i/g]]]]]/[[1-
D]*[1-a1+[a1*L3]]*[[1-[i/g]]^2]];
Part4=[Part2-Part3];

               mql1=[Part1+Part4]



function [dly1]=DELY(L3,L1)
```

```
% $$$$$$$$$$$$$$$$$$$ Declaring Variables $$$$$$$$$$$$$$$$$$$$$$$$$$$$
a1=0.27;         %a=Arrival_one
D=0.30;          %D=Departure
Q=5 ;            %Value for the threshold
L2=15 ;          %Value for second threshold
if L1<=5
  J=0;
else
     J=L1-5;
       L1=5;
end

AwnD1=a1*(1-D); % AwnD1=AwnD_one=Arrival_one*(1-Departure)
Dwna1=D*(1-a1); % Dwna1=Dwna_one=Departure*(1-Arrival_one)

a=[[[a1^4]*[1-D]^3]/[[D^4]*[1-a1]^4]]; % coefficient of Pai4 = PL1-1
b=[[[a1^3]*[1-D]^2]/[[D^3]*[1-a1]^3]]; % coefficient of Pai3 = PL1-2
c=[[a1+D]-[2*a1*D]] ;
d=[a1-[a1*D]];
e=[D-[a1*D]];
g=[[D-[a1*D]+[a1*D]*L3]] ;
h=[[[[a1+D]-[2*a1*D]]-[a1*L3]]+[2*a1*D*L3]];
i=[[[a1-[a1*D]]-[a1*L3]]-[a1*D*L3]];

%$$$$$$$$$$$$$$$$$ Get the coefficients of the States $$$$$$$$$$$$$$$

p0=1 ;                 %Coefficient of P0
p1=[[a1]/[D*[1-a1]]] ;  %Coefficient of P1

for r=2:Q-1
    p2to(r)=[[[a1^r]*[[1-D]^[r-1]]]/[[D^r]*[1-a1]^r]]; %Summing coefficient
of p2 to pL1-1
end

pL1=[a*d]/g   ;           % Coefficient at L1
pL1plus1=[[a*i*d]/g]/g  ; % Coefficient at L1+1
pL1plus2=[pL1plus1*i]/g ; % Coefficient at L1+2
pL1plus3=[pL1plus2*i]/g ; % Coefficient at L1+3
pL1plus4=[pL1plus3*i]/g ; % Coefficient at L1+4
pL1plus5=[pL1plus4*i]/g ; % Coefficient at L1+5
pL1plus6=[pL1plus5*i]/g ; % Coefficient at L1+6
pL1plus7=[pL1plus6*i]/g ; % Coefficient at L1+7
pL1plus8=[pL1plus7*i]/g ; % Coefficient at L1+8
pL1plus9=[pL1plus8*i]/g ; % Coefficient at L1+9
pL1plus10=[pL1plus9*i]/g ;% Coefficient at L1+10


%$$$$$$$$$$$$$$$$$$$$$$$$ Get Pai0 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

part1=[p0+p1]+[sum(p2to)];
part2=[pL1+pL1plus1+pL1plus2+pL1plus3+pL1plus4];
part3=[pL1plus5+pL1plus6+pL1plus7+pL1plus8+pL1plus9+pL1plus10];
bottom=[part1+part2+part3];

Pai0=[[1]/[bottom]];

% $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MQL $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
Part1=[Pai0*[[d/e]+[[d/e]^[L1+1]*[L1-1]]-[L1*[d/e]^L1]]]/[[1-D]*[[1-
[d/e]]^2]]
Part2=[Pai0*[[[d/e]^L1]*[1-a1]*[[i/g]+[L1*[1-[i/g]]]]]]/[[1-D]*[1-
a1+[a1*L3]]*[[1-[i/g]]^2]]
Part3=[Pai0*[[d/e]^L1]*[1-a1]*[[i/g]^[J+1]]*[1+[[L1+J]*[1-[i/g]]]]]/[[1-
D]*[1-a1+[a1*L3]]*[[1-[i/g]]^2]]
Part4=[Part2-Part3]

               mql1=[Part1+Part4];

   %$$$$$$$$$$$$$$$$$$$$$$$$ Throughput $$$$$$$$$$$$$$$$$$$$$$$$$$$

                 thrp1=[1-[Pai0*D]];

%$$$$$$$$$$$$$$$$$$$$$$$$$$$$  Delay $$$$$$$$$$$$$$$$$$$$$$$$$$$$$

               dly1=[mql1/thrp1];




function [pkl1]=PKL(L3,L1)

% $$$$$$$$$$$$$$$$$$$ Declaring Variables For Function MQL2 Method
$$$$$$$$$$$$$$$$$$$$$$$$$$$
a1=0.27;        %a=Arrival_one
D=0.30;         %D=Departure
Q=5 ;           %Value for the threshold
if L1<=5
  J=0;
else
     J=L1-5;
       L1=5;
end

AwnD1=a1*(1-D); % AwnD1=AwnD_one=Arrival_one*(1-Departure)
Dwna1=D*(1-a1); % Dwna1=Dwna_one=Departure*(1-Arrival_one)

a=[[[a1^4]*[1-D]^3]/[[D^4]*[1-a1]^4]]; % coefficient of Pai4 = PL1-1
b=[[[a1^3]*[1-D]^2]/[[D^3]*[1-a1]^3]]; % coefficient of Pai3 = PL1-2
c=[[a1+D]-[2*a1*D]] ;
d=[a1-[a1*D]];
e=[D-[a1*D]];
g=[[D-[a1*D]+[a1*D]*L3]];
h=[[[[a1+D]-[2*a1*D]]-[a1*L3]]+[2*a1*D*L3]];
i=[[[a1-[a1*D]]-[a1*L3]]-[a1*D*L3]];

%$$$$$$$$$$$$$$$$$$ Get the coefficients of the States $$$$$$$$$$$$$$$

p0=1         ;         %Coefficient of P0
p1=[[a1]/[D*[1-a1]]] ; %Coefficient of P1

for r=2:Q-1
    p2to(r)=[[[a1^r]*[[1-D]^[r-1]]]/[[D^r]*[1-a1]^r]]; %Summing coefficient
of p2 to pL1-1
end

pL1=[a*d]/g;              % Coefficient at L1
```

```
pL1plus1=[[a*i*d]/g]/g ;  % Coefficient at L1+1
pL1plus2=[pL1plus1*i]/g ; % Coefficient at L1+2
pL1plus3=[pL1plus2*i]/g ; % Coefficient at L1+3
pL1plus4=[pL1plus3*i]/g;  % Coefficient at L1+4
pL1plus5=[pL1plus4*i]/g;  % Coefficient at L1+5
pL1plus6=[pL1plus5*i]/g;  % Coefficient at L1+6
pL1plus7=[pL1plus6*i]/g;  % Coefficient at L1+7
pL1plus8=[pL1plus7*i]/g;  % Coefficient at L1+8
pL1plus9=[pL1plus8*i]/g;  % Coefficient at L1+9
pL1plus10=[pL1plus9*i]/g; % Coefficient at L1+10

%$$$$$$$$$$$$$$$$$$$$$$$$ Get Pai0 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

part1=[p0+p1]+[sum(p2to)];
part2=[pL1+pL1plus1+pL1plus2+pL1plus3+pL1plus4];
part3=[pL1plus5+pL1plus6+pL1plus7+pL1plus8+pL1plus9+pL1plus10];
bottom=[part1+part2+part3];

    Pai0=[[1]/[bottom]];



%$$$$$$$$$$$$$$$$$$$$$$$ Probability of Packet Loss $$$$$$$$$$$$$$$$$$$$$$$$

Section1=[L3*pL1]+[L3*pL1plus1]+[L3*pL1plus2]+[L3*pL1plus3];
Section2=[L3*pL1plus4]+[L3*pL1plus5]+[L3*pL1plus6];
Section3=[L3*pL1plus7]+[L3*pL1plus8]+[L3*pL1plus9];
Section4=[Section1+Section2+Section3];
Section5=[[[L3*pL1plus10]*D]+[pL1plus10*[1-D]]];

       pkl1= Pai0*[Section4+Section5]





function [thrp1]=THRP(L3,L1)

% $$$$$$$$$$$$$$$$$$$ Declaring Variables $$$$$$$$$$$$$$$$$$$$$$$$$$$
a1=0.27 ;          %a=Arrival_one
D=0.30;            %D=Departure
Q=5 ;              %Value for the threshold
L2=15;             %Value for second threshold
if L1<=5
  J=0;
else
     J=L1-5;
       L1=5;
end

AwnD1=a1*(1-D); % AwnD1=AwnD_one=Arrival_one*(1-Departure)
Dwna1=D*(1-a1); % Dwna1=Dwna_one=Departure*(1-Arrival_one)

a=[[[a1^4]*[1-D]^3]/[[D^4]*[1-a1]^4]]; % coefficient of Pai4 = PL1-1
b=[[[a1^3]*[1-D]^2]/[[D^3]*[1-a1]^3]]; % coefficient of Pai3 = PL1-2
c=[[a1+D]-[2*a1*D]] ;
d=[a1-[a1*D]];
e=[D-[a1*D]];
g=[[D-[a1*D]+[a1*D]*L3]];
h=[[[[a1+D]-[2*a1*D]]-[a1*L3]]+[2*a1*D*L3]];
i=[[[a1-[a1*D]]-[a1*L3]]-[a1*D*L3]];
```

```
%$$$$$$$$$$$$$$$$$ Get the coefficients of the States $$$$$$$$$$$$$$

p0=1  ;                 %Coefficient of P0
p1=[[a1]/[D*[1-a1]]] ;  %Coefficient of P1

for r=2:Q-1
    p2to(r)=[[[a1^r]*[[1-D]^[r-1]]]/[[D^r]*[1-a1]^r]]; %Summing coefficient
of p2 to pL1-1
end

pL1=[a*d]/g    ;          % Coefficient at L1
pL1plus1=[[a*i*d]/g]/g  ; % Coefficient at L1+1
pL1plus2=[pL1plus1*i]/g ; % Coefficient at L1+2
pL1plus3=[pL1plus2*i]/g  ;% Coefficient at L1+3
pL1plus4=[pL1plus3*i]/g ; % Coefficient at L1+4
pL1plus5=[pL1plus4*i]/g ; % Coefficient at L1+5
pL1plus6=[pL1plus5*i]/g ; % Coefficient at L1+6
pL1plus7=[pL1plus6*i]/g  ;% Coefficient at L1+7
pL1plus8=[pL1plus7*i]/g ; % Coefficient at L1+8
pL1plus9=[pL1plus8*i]/g ; % Coefficient at L1+9
pL1plus10=[pL1plus9*i]/g ;% Coefficient at L1+10


%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ Get Pai0 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

part1=[p0+p1]+[sum(p2to)];
part2=[pL1+pL1plus1+pL1plus2+pL1plus3+pL1plus4];
part3=[pL1plus5+pL1plus6+pL1plus7+pL1plus8+pL1plus9+pL1plus10];
bottom=[part1+part2+part3];

        Pai0=[[1]/[bottom]];

% $$$$$$$$$$$$$$$$$$$$$$$ Throughput $$$$$$$$$$$$$$$$$$$$$$$$$$

                 thrp1=[1-[Pai0*D]];



%*************General Dropping Functions*******
x=1:2:20
L3=log(x).^-3
L4=0.4./x
L5=(0.4).^x
L6=x.^-3
plot(x,L3,':k>',x,L4,'-g*',x,L5,'-bo',x,L6,'-rs'...
,'LineWidth',1,'MarkerSize',6,'MarkerFaceColor',[0 0 0])
xlabel('X-Axis: Threshold')
ylabel('Y-Axis: Value ')
gtext('L1, F1')
gtext('L1, F2')
gtext('L1, F3')
gtext('L1, F4')
```