Analysis and Simulation of Weighted Random Early Detection (WRED) Queues

EECS 891 Project
by

Mark Wurtzler

BSEE, Northwestern University, Evanston, Illinois (2002)

Submitted to the Department of Electrical Engineering and Computer Science and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science

Dr. David Petr, Chairperson

Dr. Victor Frost

Dr. Gary Minden

Date of Acceptance

Abstract

This project examines the effects of weighted random early detection (WRED) packet discard on dropping probabilities for multi-class traffic. The flexibility of the WRED parameters will be illustrated with respect to performance parameters and traffic characteristics. Recent advances in analytic RED modeling will be described and extended to WRED and analytic results compared to those found using the simulation model developed for this project. Guidelines for setting WRED parameters will also be examined for various traffic scenarios.

Acknowledgements

The author would like to express his gratitude to Dr. David Petr for his dedication as a teacher of network and queue theory, as well as his support and guidance as an advisor.

Table of Contents

List of Figures:

List of Tables:

Section 1: Introduction

1.1 Problem Description

Modern networks require the integration of a variety of data flows into an infrastructure that may not be precisely suited to handle the requirements and characteristics of the traffic. Under heavy loads, decisions must be made about which packets will be discarded in order to maintain stability in the network. The tail drop mechanism, in which all packets are discarded once a queue becomes full, ensures that all of the positions in the queue are used to their full potential. If all arriving packets conformed to either a uniform or a basic Poisson distribution with exponential interarrival times, tail drop discarding would likely provide an excellent means of utilizing a queue when the load approaches or exceeds capacity, though differentiation with respect to dropping probability may still be desirable for different classes of traffic.

However, much of the traffic on the Internet is inherently composed of bursts of data. Services requiring human interaction, including web browsing and TCP signaling, often involve quick transfers of data followed by periods of inactivity. The queues located at routers and switches must have the ability to handle these bursts. As access speeds to the Internet improve, burst characteristics may become more pronounced, increasing the severity of the problem. When dial-up service was the major means of access to the Internet, the low transfer rate constraints imposed by the delivery mechanism (modem) produced a less pronounced burst than with a modern broadband system. It has been found that a 33Kbps modem user produces a peak rate that is about 3.3 times the average transfer rate, whereas a similar customer using a 1Mbps broadband connection produces a peak rate approximately 100 times the average rate [1]. Some of the issues with quality of service can be alleviated by the aggregation of large numbers of flows onto large links [2]. However, as broadband Internet service reaches less populated communities, traffic may remain bursty due to a low number of subscribers connected to a single link. Because a tail drop queue operates near its capacity under heavy load, traffic bursts have no place to be stored upon arrival and are discarded at a disproportionate rate. The inequity in dropping probability between the two types of traffic may grow to unacceptable levels. The simulation result in Figure 1 illustrates the problems that arise under heavy load (ρ=1.2) with the use of tail drop blocking when combining an even amount of a bursty traffic flow, in this case batch Poisson arrivals, with an equal amount of smooth constant interarrival traffic:

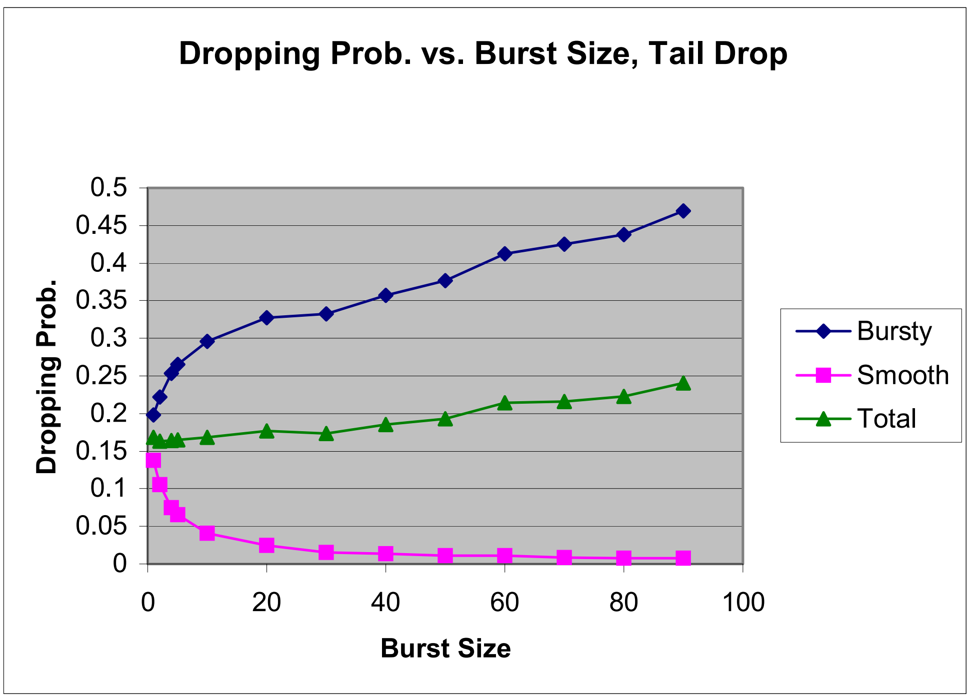

Figure 1: Blocking Probability using Tail Drop, ρ=1.2, Queue Size=100, Mean Exponential Packet Length=1000, Link Rate = 10000

As the figure shows, the bias against bursty traffic increases rapidly as the burst size becomes larger, though the overall load remains constant. Note that the dropping probabilities are not precisely equal at a burst size of one, because the bursty model still uses Poisson arrivals instead of the constant interarrivals of the smooth source. The bursty traffic is much more prone to blocking than the smooth traffic with which it is sharing the queue for all but the smallest burst sizes. As burst size decreases, the overall blocking rate approaches the minimum of 1/6 as required for stability in a M/M/1/K system at ρ=1.2. At larger burst sizes, smooth packets have a lower probability of being dropped due to long windows of time between bursts in which the queue may not be operating at overload conditions. However, the dropping probability for the batch Poisson arrivals increases rapidly to nearly 50% as the burst size approaches the queue size.

A well-designed network must be capable of properly handling these bursts of data and the heavy loads that may be encountered. At peak usage, when the load ρ may increase to unity and beyond, the network designer may want packets discarded in a manner that provides equitable service and does not overly discriminate against bursty traffic. This project addresses, through analytic and simulation modeling, how Weighted Random Early Detection can be used to effectively establish a flexible service that can be configured to provide the desired relationship between the blocking probabilities of two disparate traffic flows.

1.2 Project Purpose and Motivation

The goal of this project is to extend the analytic model used for Random Early Detection (RED) to WRED and to use simulation to examine the differences from the analytic solutions. The simulation portion of the report will allow examination of the performance characteristics of WRED in a more realistic environment. This work will show how performance varies with respect to the multitude of configuration parameters available to WRED technology.

Though much work has been put into the evaluation of RED in the last few years, the research of RED variants like WRED is still a nascent field. Technologies that can combine the desired benefits of early detection with a package that allows for greater control of tradeoffs inherent in RED may hold great promise.

Section 2: Analysis of Random Early Detection and WRED

2.1 Background on RED

In its most general form, Random Early Detection is the probabilistic discard or marking of packets as a function of queue fill before overflow conditions are reached. The probability of a packet being marked or dropped is determined by a monotonically increasing drop function d(k).

The merits of RED have been greatly debated over the last ten years. The first paper detailing random early detection's merits was the Floyd and Jacobson paper of 1993 [3]. Some of the key reasons stated for RED adoption were as follows:

- Congestion Avoidance – RED allows for queue congestion to be managed before a critical overflow point is reached. Also, keeping the queue size lower decreases delay for those packets that are not dropped.
- Global TCP Synchronization Avoidance – By marking packets for early discard, the number of consecutive drops can be reduced. Many Internet designers were concerned that consecutive drops when queues became full could cause global instability in the network as many queues signal their source to reduce their window at the same time
- Fairness – Reduces the bias against bursty traffic, as mentioned earlier. RED will avoid a situation in which bursty traffic faces extreme packet loss compared to smooth traffic.

While many of these phenomena have been seen in controlled experiments, much active research still involves the refinement and verification of these claims in more realistic networks. Some of Floyd and Jacobson's claims have since been refuted under certain conditions, including the consecutive packet drop claim. [4]

The RED dropping probability is a linear function that determines whether a packet will be admitted to the queue when *k* packets are currently waiting. The dropping probability d(k) is determined by a set of parameters that create the drop function. The important parameters include:

min_{th}	Minimum Queue Fill for RED Dropping
max_{th}	Maximum Queue Fill for RED Dropping (Normally set to K, the maximum queue size)
max_p	Maximum Probability of RED Dropping

For instantaneous queue size *k,* d(k) is as follows:

$$d(k) = 0 \text{ if } k < min_{th}$$
$$d(k) = 1 \text{ if } k > max_{th}$$
$$d(k) = (\max{}_p)\left(\frac{k - \min{}_{th}}{\max{}_{th} - \min{}_{th}}\right) \text{ otherwise}$$

[4]

producing the function plotted in Figure 2 if max_{th} is set equal to queue size K:

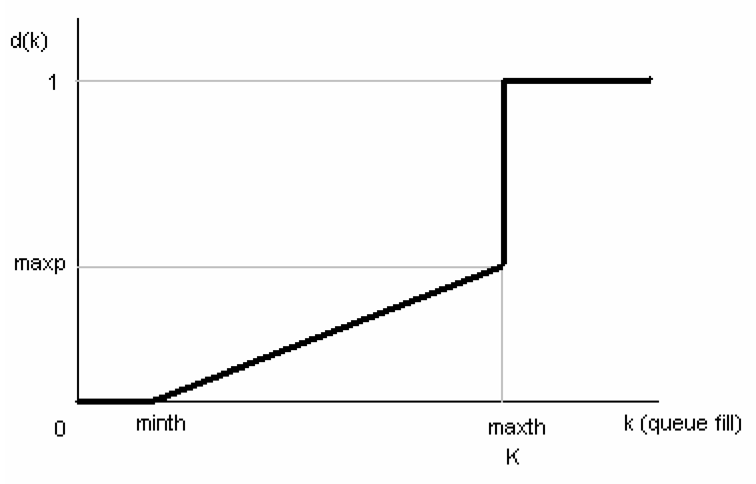

Figure 2: RED Dropping Function

2.2 Queue Averaging Pros and Cons

One major area of study in RED analyzes the use of an exponentially weighted moving average for estimation of the queue fill [5]

$$\hat{k} = (1 - w)\hat{k}_{n-1} + wk_n$$

Decreasing the weight *w* applied to the new sample of the queue fill makes the system less aggressive in dropping packets during sudden bursts. It has been shown that the careful adjustment of the queue sample weight can reduce dropping probability as compared to the instantaneous queue size method used in this report [6]. However, it has been shown that a low weighting can lead to a greater number of consecutive packet drops than a tail drop queue [4], and therefore it may contradict one of the major reasons for RED implementation with respect to TCP synchronization. This project will use instantaneous queue size due to the fact that its more aggressive discard and sensitivity to bursts will exacerbate differences in fairness between classes of traffic. Instantaneous queue sampling also makes possible the type of analysis to be covered later in this section. The proper configuration of the queue fill average is a major area of note in RED research, but it is beyond the scope of this report on WRED.

2.3 Introduction to RED Analysis

The groundwork for RED analytic modeling was created by a breakthrough paper by Bonald, May, and Bolot published in IEEE INFOCOM in 2000 [4]. The work addresses the problems arising from the mixture of smooth UDP traffic and bursty TCP traffic in a traditional tail drop queue, and it was the first successful attempt to create a satisfactory analytic model of packet discard in RED. The following sections detail the model described in this paper and its adaptation to WRED analysis.

2.4 The Bonald/May/Bolot RED Model [4]

The paper uses continuous-time Markov chain analysis to establish the transition rates, and consequently the stationary state probability vector π, used to find the blocking probability for bursty traffic. However, many simplifications and approximations must be used to create a model that can be solved analytically. Many RED implementations use queue size averaging; however, for simplification much of the paper uses instantaneous queue size to determine the RED drop probability. This simplification will also be used in this project for the analytic and simulation model so that the two can be properly compared and also to operate more easily within the limits of the Extend Discrete Event modeling library. The bursty traffic source used for this model is a batch Poisson arrival, in which packets arrive in groups of size B with exponential interarrival times. Through referenced in the paper as "TCP", the model does not exactly match the characteristics of

TCP flows in practice [7]. The service time distribution of the individual packets are exponential.

The analytic solution requires an approximation that is problematic for small queue sizes. Approximation #1 in [4] states that the RED router uses the same dropping probability d(k) on all packets in the same burst, where *k* is the instantaneous queue size at the time the first packet in the burst arrives at the router. In a real implementation, packets will be discarded at the dropping probability corresponding to the queue fill when they individually arrive, which will be higher for later packets in a burst than for the first packet. Thus, any analytic solution for a RED queue that is calculated in this manner will prove to be only a lower bound for the dropping probability of a RED system. The Poisson Arrivals See Time Averages (PASTA) property states that the continuous-time stationary state distribution π will equal the distribution of the number of packets in the queue upon burst arrival [4]. This property is necessary to produce the following dropping probabilities for tail drop (TD) and RED [4]:

$$P_{TD} = \pi(K) + \pi(K-1)\frac{(B-1)}{B} + ... + \pi(K-B+1)\frac{1}{B}$$

$$P_{RED} = \pi(K) + \pi(K-1)d(K-1) + ... + \pi(1)d(1)$$

Once the state probabilities are found, the dropping probability can be found using the above formulas. The offered load can be found by the formula $\rho = \frac{B\lambda}{\mu}$. The rate of burst arrival is considered λ, while the packet service rate remains μ.

The conclusion of the paper remarks that RED balances the blocking probability between bursty and smooth traffic by increasing the smooth traffic blocking probability rather than by lowering the bursty blocking probability. The simulations in this project show that this is not the case across all ranges of the RED parameters. (See Figure 14/15)

2.5 Extension of Markov Chain Analysis to WRED

Using the same approximations, WRED can be analyzed in a very similar way to RED. WRED uses the same parameters as RED, but it has the ability to perform RED on traffic classes individually. For example, this figure shows a WRED system in which both classes have the same max_p but different min_{th}. The Class 2 traffic begins RED discard when the queue contains one packet, and Class 1 traffic discard begins at k=2. These WRED dropping functions will be used in the analytic example in Section 2.6.

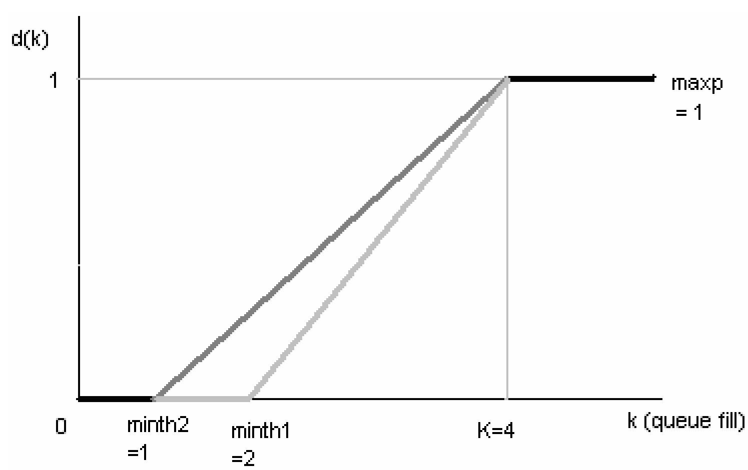

Figure 3: WRED Dropping Functions

WRED analysis is similar to RED analysis, but the transition rates for the CTMC will differ and the general complexity of the Markov chain will increase. This section will take the reader through a complete example of the calculation of WRED blocking probability for a system with two bursty input sources that are assigned different RED blocking probabilities.

2.6 WRED Analysis Example

For this example, the burst size is B=2, and the offered load is 0.6. The resulting λ is 0.3 bursts/sec for a normalized service rate (μ) of 1. Note that K is still equal to 4, and that this Markov chain corresponds to the number in the total system (queue fill plus one packet in the server) in order to ease the calculations. Thus, the dropping probability when the number in the system (x) is 5 corresponds to the dropping probability of a queue fill *k* of 4, and so on. The arrival loads of each class are equal at 0.15 bursts/sec for each class.

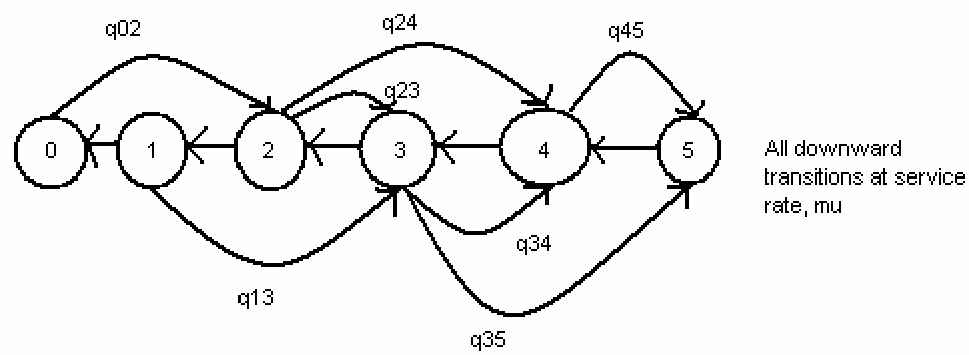

Figure 4: WRED CT Markov Chain

The dropping probabilities from Figure 3 are as follows:

Table 1: Dropping Probabilities of WRED Example

Number in system (x)	Class 1 Dropping $d_1(x)$	Class 2 Dropping $d_2(x)$
0	0	0
1	0	0
2	0	1/4
3	1/3	1/2
4	2/3	3/4
5	1	1

Calculating the transition probabilities:

For a packet arriving when there are zero or one packets in the system (zero in queue), there will be no dropping:

$$q_{02} = \lambda$$
$$q_{13} = \lambda$$

When x=2 (queue fill = 1), only class 2 packets will have a chance of dropping:

$$q_{24} = \frac{1}{2}\lambda + \frac{1}{2}\lambda\binom{2}{2}\left(1-\frac{1}{4}\right)^2 = 0.2344 \text{ (Transition rate if no packets drop)}$$

$$q_{23} = \frac{1}{2}\lambda\binom{2}{1}\left(\frac{1}{4}\right)\left(\frac{3}{4}\right) = 0.05625 \text{ (Transition if one packet in the burst is dropped)}$$

When x=3, both classes may experience dropping:

$$q_{35} = \frac{1}{2}\lambda\binom{2}{2}\left(\frac{2}{3}\right)^2 + \frac{1}{2}\lambda\binom{2}{2}\left(\frac{1}{2}\right)^2 = 0.1042$$

$$q_{34} = \frac{1}{2}\lambda\binom{2}{1}\left(\frac{1}{3}\right)\left(\frac{2}{3}\right) + \frac{1}{2}\lambda\binom{2}{1}\left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = 0.1417$$

When x=4, the only way for the state to transition upward is if no packets are dropped:

$$q_{45} = \frac{1}{2}\lambda\left[1-\left(\frac{2}{3}\right)^2\right] + \frac{1}{2}\lambda\left[1-\left(\frac{3}{4}\right)^2\right] = 0.1490$$

Packets will be serviced with exponential service time at a rate of 1, so:

$$q_{54}, q_{43}, q_{32}, q_{21}, q_{10} = 1$$

The following transition rate matrix is then produced:

$$\mathbf{Q} = \begin{pmatrix} -0.3 & 0 & 0.3 & 0 & 0 & 0 \\ 1 & -1.3 & 0 & 0.3 & 0 & 0 \\ 0 & 1 & -1.29 & 0.05625 & 0.2344 & 0 \\ 0 & 0 & 1 & -1.2459 & 0.1417 & 0.1042 \\ 0 & 0 & 0 & 1 & -1.1490 & 0.1490 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

Solving for $0 = \boldsymbol{\pi}\mathbf{Q}$ subject to $\sum \pi_i = 1$

[redacted]

The blocking probabilities can be found by using equations similar to those for RED:

$$P_{RED,1} = \sum_x \pi_x * d_1(x)$$

$$P_{RED,2} = \sum_x \pi_x * d_2(x)$$

Resulting in blocking probabilities:

$P_{RED,1} = 0.0983$

$P_{RED,2} = 0.1673$

Section 3: Simulation Model Design

3.1 Introduction to the Simulation Model

The RED simulation model for this project was designed to resemble the simulation used by the Bonald paper [4]. The CAD package Extend [8] and its Discrete Event library were used for all simulation data presented in this report.

Modeling and verifying RED performance presents some unique challenges. Due to the approximations used in the analytic model presented in [4], simulation data cannot be expected to directly compare to analytic solutions for small queue sizes or large bursts. The RED mechanism in simulation, as well as in a real-world environment, will discard packets based on the queue size when each packet enters the system, not when each burst arrives. For RED, and especially for WRED, the Markov Chain model also becomes cumbersome as the queue and burst sizes are expanded, producing the need for simulation.

3.2 The Bursty "TCP" Source Model

To simulate the TCP-type packets on the network link, a batch Poisson source model was created in Extend. A standard Poisson generator block is attached to the Discrete Event library's "Unbatch" block, which multiplies the number of packets generated when triggered by the standard Poisson source. The attributes are added later, including packet length, so that the burst will consist of non-identical packets arriving at the same time. The Unbatch block is controlled by the "Burst Length" control so that the number of packets released upon a triggering Poisson arrival can easily be changed. The RED Class is assigned as an attribute so the packet can be routed later in the model. All of the simulations used in this project have exponential packet length distributions with a mean of 1000 bits.

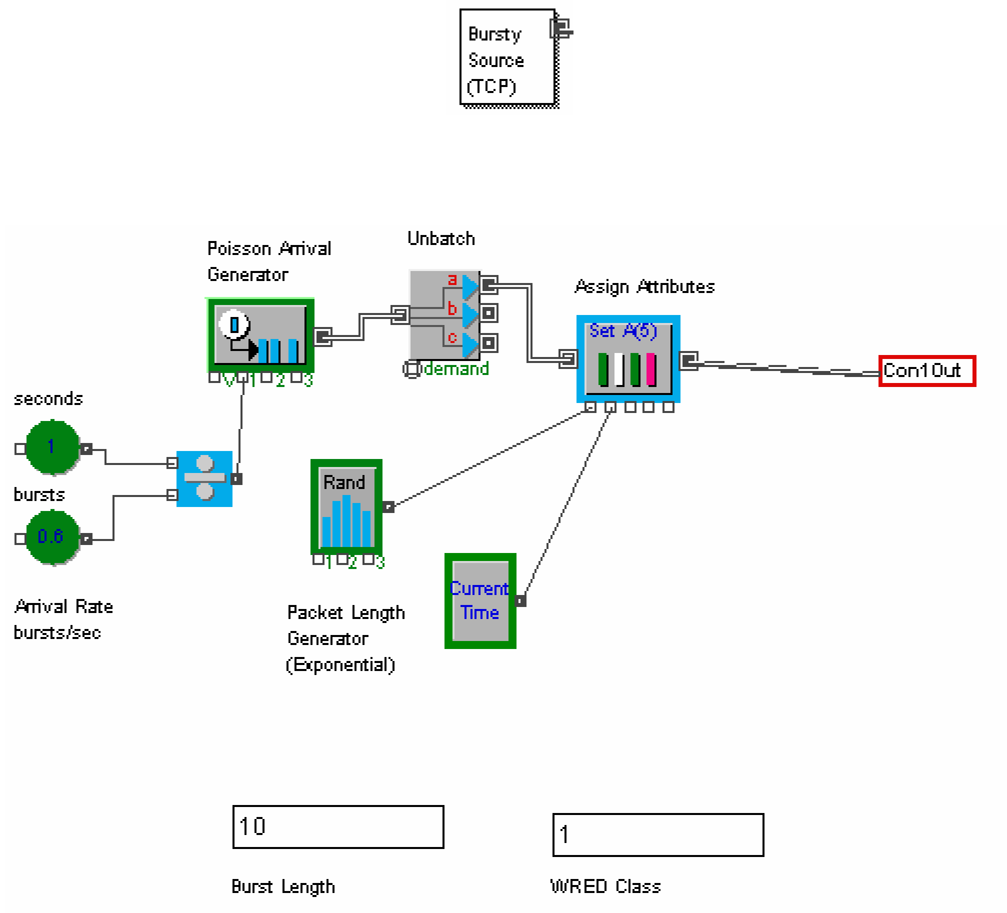

Figure 5: Bursty Source Model Block

3.3 Smooth "UDP" Source Model

The smooth traffic generator structure is similar to the bursty packet generator, except for the use of constant interarrivals. Some previous RED models have used constant bit rate sources. Though this model uses constant interarrivals, it retains the exponential packet lengths of 1000 bits from the bursty source model. The result is a source that provides a steady stream of variable length packets.

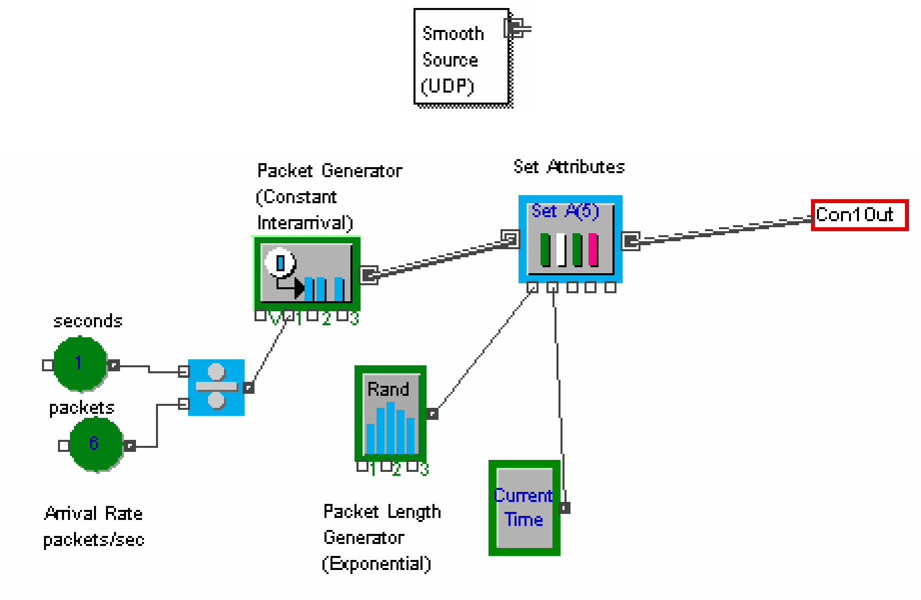

Figure 6: Smooth Source Model Block

3.4 The Random Early Detection Block

A key challenge in this project was the creation of a Random Early Detection block for Extend. Without using any custom libraries, it was possible to make a device that takes as inputs both the packet stream and the queue size and then reroutes the desired proportion of packets for marking or dropping. The block distinguishes between packets that were dropped for exceeding max_{th} and those dropped by RED before queue overflow between min_{th} and max_{th}. The routing architecture can be seen in Figure 7:

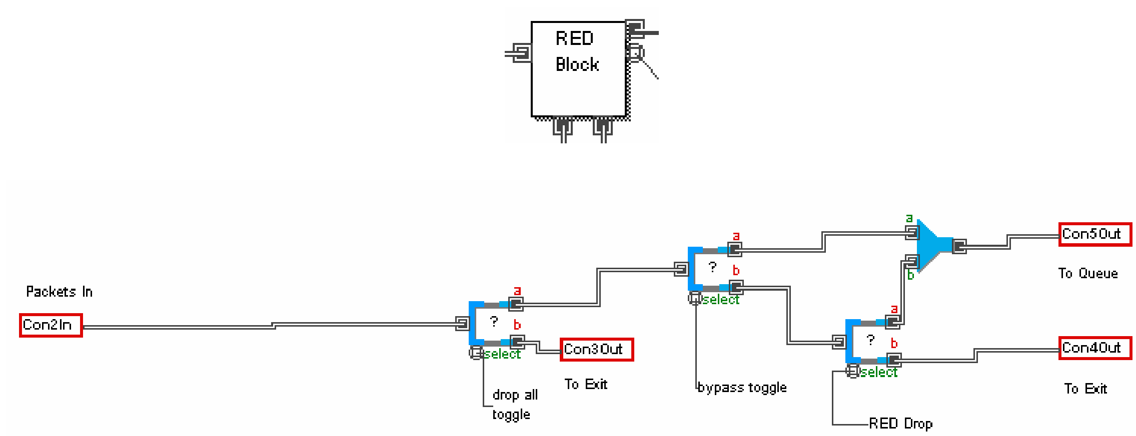

Figure 7: RED Block Routing Architecture

A “drop all” toggle diverts all input packets when max_{th} has been exceeded. For cases where max_{th} was set to the maximum queue size, this would represent the type of dropping that occurs in a tail drop queue. As will be used later, this RED block can function as a tail drop block if min_{th} is set equal to max_{th}.

In sub-overflow conditions, packets are then routed to the RED Drop switch so long as the queue fill is above min_{th} by the “bypass toggle”, in which case the RED dropping mechanism will be activated. The “Red Drop” input determines if the packet should be discarded for this particular case, and is calculated from the logic in Figures 8 and 9.

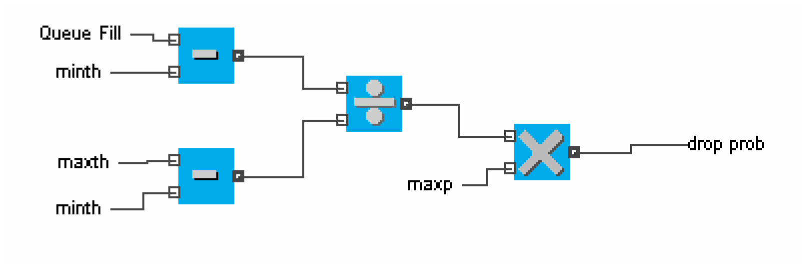

Figure 8: RED Dropping Probability

The RED dropping probability is produced using the formula given in Section 2.1. The “drop prob” output sends the dropping probability to the decision logic, which then compares the dropping probability to a random number between 0 and 1 to determine if the RED dropping switch should be activated.

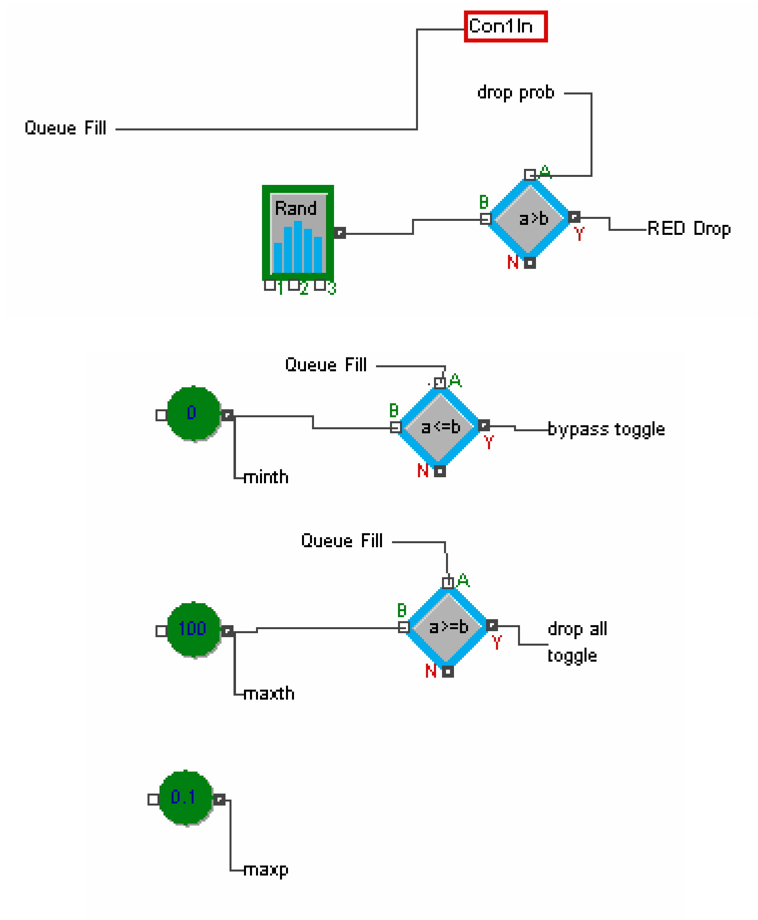

Figure 9: RED Logic and Decision Blocks

3.5 WRED Model Design

With the major components designed, the entire WRED model can be constructed by the addition of a basic FIFO queue and a server delay block that delays each packet based on the packet length assigned at the source. Other blocks have capabilities like counting the number of packets discarded by the RED blocks and measuring average queue fill and packet delay.

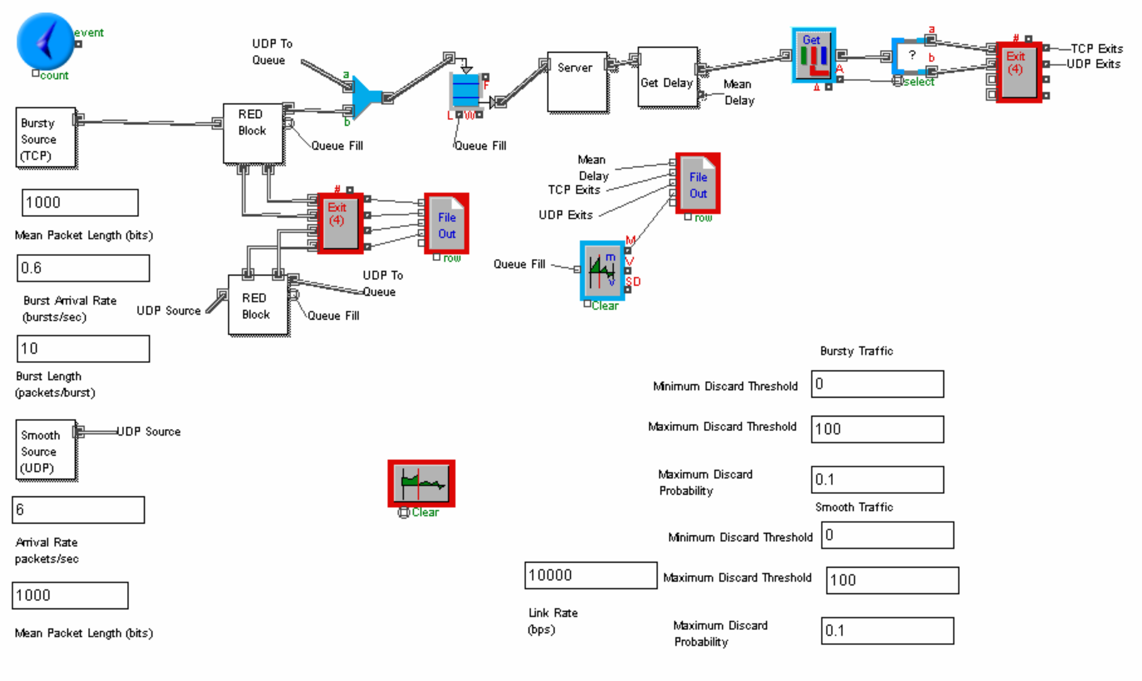

Figure 10: WRED System Model

WRED is implemented by placing a RED block immediately after each source to decide which packets will be admitted to the FIFO queue. The RED parameters can then be adjusted for each input class individually. The packets that are not discarded by the RED block are combined and enter the queue.

Section 4: Simulation Model Verification

4.1 RED Verification Issues

Verification is particularly difficult for RED queues due to the fact that the analysis techniques available at this time are only an approximation to the blocking probability encountered in a realistic implementation. Recal that Approximation #1 in [4], which states that in the analytic solutions all packets are discarded with the dropping probability assigned to the queue fill at the beginning of the burst, is not realistic for small queues. Unfortunately, analytic solutions for large queues can be cumbersome due to the large number of states and transition probabilities that must be found in the Markov chain, particularly for WRED and for large burst sizes. The Extend WRED model used in this project can be checked, however, by verification of the individual components and comparison to known results. After the blocks are shown to perform correctly, the overall model can then be found to operate within the bounds of the analytic solution for a manageable queue size.

4.2 Bursty Source Model Verification

Using analytic techniques similar to those shown in Section 2, a tail drop Markov chain can be constructed and used to find the blocking probability of a tail drop system with a bursty traffic source. Tail drop verification will allow the source to be verified separately from the rest of the model, as the RED block does not perform any complex operations and Approximation #1 [4] is not used. It should be expected that the tail drop blocking probability found in simulation match the probability from tail drop analysis. The simulation and analysis are constructed using the following parameters:

Table 2: Simulation Setup: Bursty Source Verification using Tail Dropping

Burst Size	3
System Size	5
Queue Size	4
Burst Arrival Rate	2
Mean Packet Length	1000 bits
Link Capacity	10000 bits
Offered Load	0.6
Total Simulation Time	7000s
Run-in Time	2000s
Number of Runs	5

The simulation time and number of runs are chosen to provide approximately 150,000 packets for sampling and measurement. The run-in time is used to ensure that the queue has reached its steady-state occupancy distribution before blocking calculation begins. Table 3 shows the excellent agreement between the simulation model and the analytic model, verifying the simulation model in tail-drop mode.

Table 3: Simulation Results: Bursty Source Verification using Tail Dropping

# Of Packets Simulated	149793
# Of Packets Blocked	24395
Blocking Probability/Sim	0.1629
Blocking Probability/Analytic [9]	0.1632
% Error	0.21%

4.3 RED Block Verification

RED block verification was performed by comparing the performance of the Extend RED model to the analytic and simulation results from the Bonald paper [4]. Due to Approximation #1, the blocking probability will be much greater in simulation than in analytic calculation for small queues. However, increasing the queue size leads to solid correlation between the simulations performed by Bonald and those in this project. For verification, the Extend model was given the following parameters used in the previous paper.

Table 4: Simulation Setup: RED Block Verification

Burst Size	variable
Queue Size	40
Burst Arrival Rate	2
Mean Packet Length	1000 bits
Link Capacity	10000 bits
Offered Load	variable
minth	20
maxth	40
maxp	1
Total Simulation Time	7000s
Run-in Time	2000s
Number of Runs	5

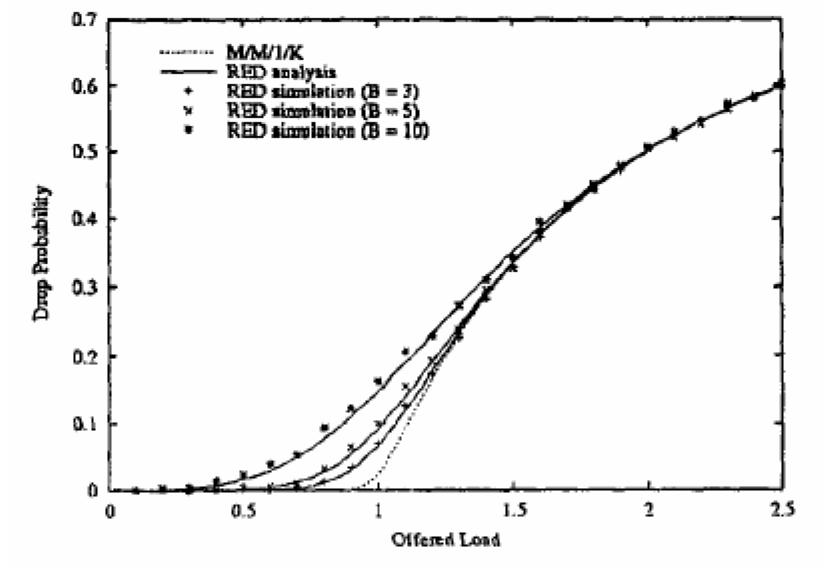

Figure 11: Dropping Probability vs. Offered Load: Published Results [4]

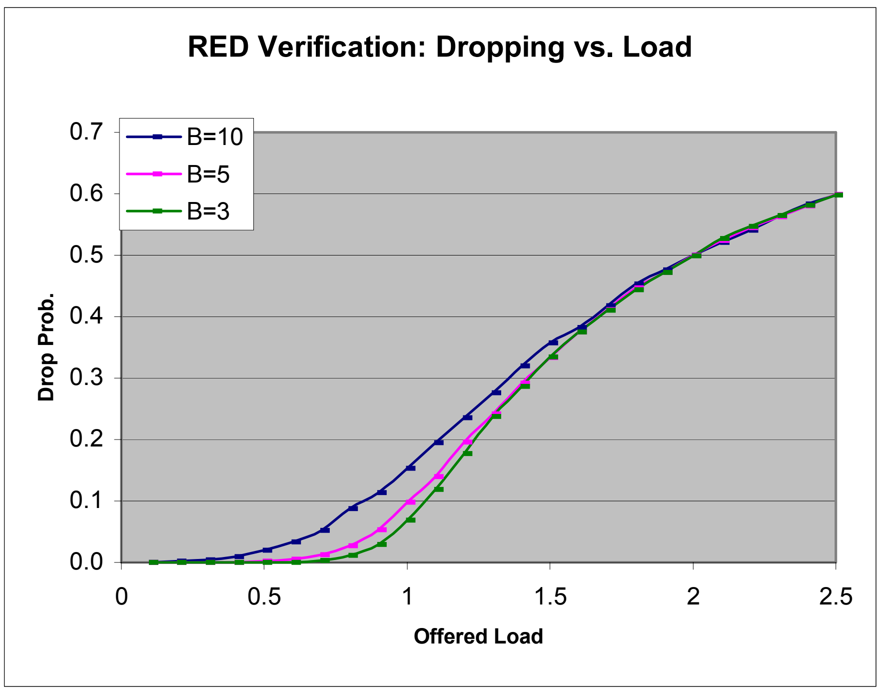

Figure 12: Dropping Probability vs. Offered Load: Extend RED Model

The dropping probability is found to match closely both the simulation results and analytic results from [4] across a range of offered loads and burst sizes. The previously published results shown in Figure 11 also illustrate that, at larger queue sizes, the dropping probability analysis approximation does not result in a large amount of error.

4.4 WRED: Comparison of Simulation vs. Analysis

The RED block and the burst source model have been shown to behave correctly for the less complex tail drop and RED packet dropping schemes. Now, a full WRED system will be simulated and compared to results found using analysis similar to that in Section 2.6. The offered load ρ is 0.6 with a burst size of 3 and a system that can hold nine packets in the queue as well as one in the server. Two bursty sources with equal arrival rates are assigned different dropping priorities (parameters listed correspond to number in queue, not system):

$$\min_{th1} = 4 \quad \max_{th1} = 9$$

$$\min_{th2} = 1 \quad \max_{th2} = 9$$

$$\max_{p1} = \max_{p2} = 1$$

Solving the transition rate matrix results in a state probability vector of:

$$\boldsymbol{\pi} = \begin{pmatrix} 0.4549 & 0.0909 & 0.1092 & 0.1310 & 0.0657 & 0.0560 & 0.0414 & 0.0250 & 0.0161 & 0.0075 & 0.0023 \end{pmatrix}$$

Using Approximation #1 [4], the drop probabilities are:

$$P_{RED,1} = 0.0362$$

$$P_{RED,2} = 0.1491$$

However, these provide only a lower bound for the blocking probability. A worst-case blocking scenario is needed to provide an upper bound. If the assumption is made that all drops occur at the maximum possible probability, the worst case dropping probability will come from the queue state B-1 packets greater than the current state.

Essentially, the worst-case approximation will use the dropping probability that the last packet in the burst will encounter, provided that none of the previous packets are dropped. This is a highly unlikely case, but it will provide some manner of upper estimate. This approximation gives:

$$P_{RED,1} = 0.0707$$

$$P_{RED,2} = 0.2487$$

The simulation is conducted using the following parameters:

Table 5: Simulation Setup: WRED Model Verification

Burst Size Class 1	3
Burst Size Class 2	3
System Size	10
Queue Size	9
Burst Arrival Rate	1/class
Mean Packet Length	1000 bits
Link Capacity	10000 bits
Offered Load	0.6
Total Simulation Time	7000s
Run-in Time	2000s
Number of Runs	5

The WRED simulation dropping probabilities are found to be within the estimates established with the analytic work:

$$P_{RED,1} = 0.0551$$

$$P_{RED,2} = 0.1731$$

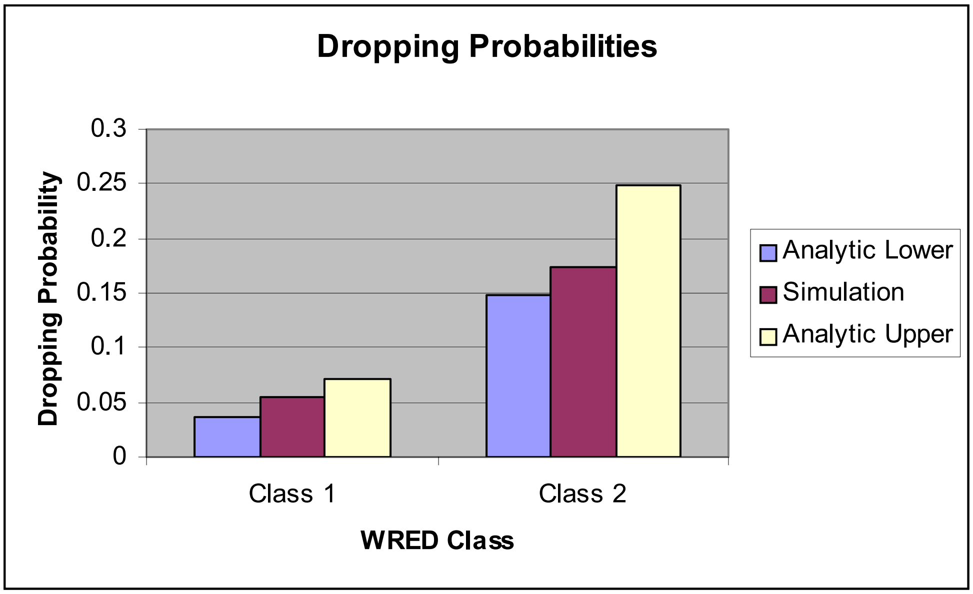

Figure 13: Dropping Probability by WRED Class

Section 5: Simulation using the WRED Model

5.1 Introduction to WRED Simulation

This simulations in this project focus on the fairness considerations for a multi-class traffic scenario in which one class "TCP" consists of bursty batch Poisson arrivals and the other class "UDP" consists of constant interarrival data. The packet lengths for both classes are exponentially distributed with a mean of 1000 bits. Unless otherwise noted, the traffic load is equally divided between the two classes.

5.2 Parameter Adjustment without WRED

To provide a baseline measurement of how the system behaves without class-specific RED parameters, simulations were performed with the model using the same parameters for each class, representing simple RED. The min_{th} and max_p parameters were adjusted across their full ranges to test how the system reacts. The burst size is 10 and the queue size is 100. In Figure 14, the RED aggressiveness is progressively increased by means of lowering the minimum threshold for dropping from K (queue size) to 0. In Figure 15, the RED aggressiveness is increased by increasing the maximum drop probability from 0 to 1 with a minimum drop threshold of zero.

Table 6: Simulation Setup: RED Parameter Variation

Burst Size Class 1	10 pkts/burst
Burst Arrival Rate	0.6 bursts/s
Packet Arrival Rate Class 2	6 pkt/s
Queue Size	100
maxth1, maxth2	100
Mean Packet Length Class 1 and 2	1000 bits
Link Capacity	10000 bits
Offered Load	1.2
Total Simulation Time	7000s
Run-in Time	2000s
Number of Runs	5

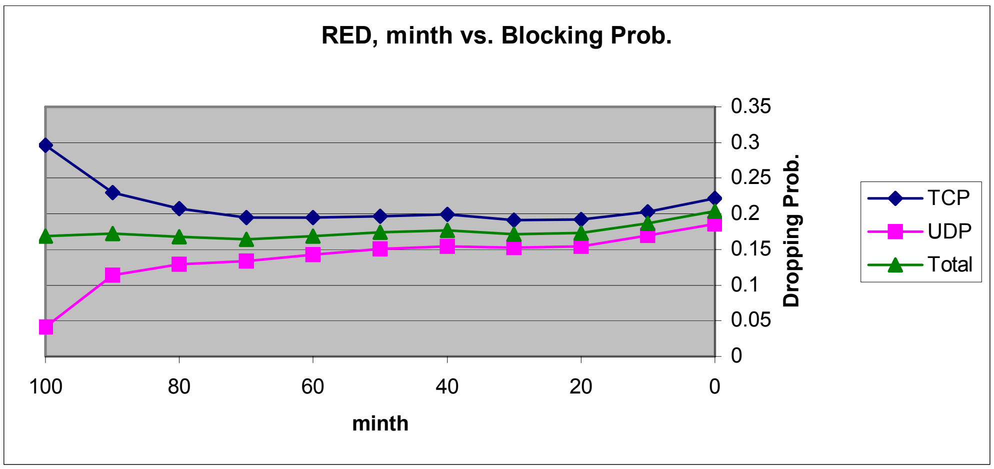

Figure 14: Dropping Probability vs. Minimum Drop Threshold, ρ=1.2, max_p=1

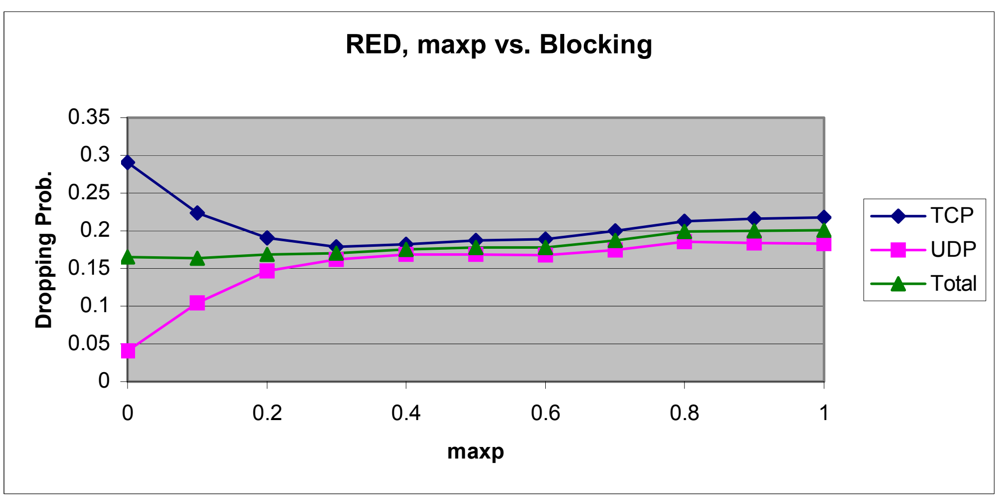

Figure 15: Dropping Probability vs. Maximum Dropping Probability, ρ=1.2, min_{th}=0

The simulation shows that, as RED becomes more aggressive, fairness initially increases rapidly. The best fairness occurs in Figure 15 when the minimum threshold is low and the max_p value is moderate. Overall dropping probability increases gradually until a point, in this case max_p=0.8 or min_{th}=20, where many packets are likely being dropped prematurely causing a large increase in total blocking. The increased aggressiveness of RED can be seen in the mean queue fill corresponding with the same max_p values as in Figure 15:

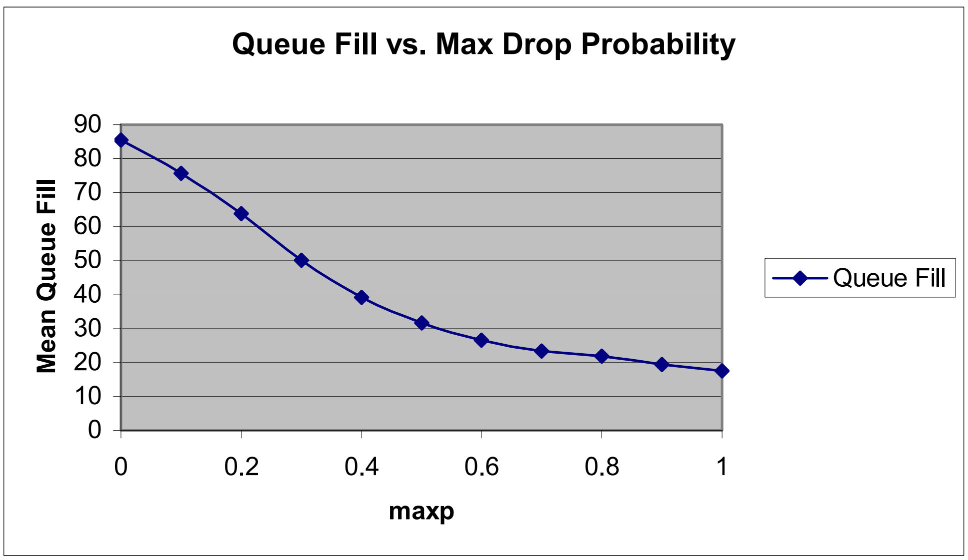

Figure 16: Queue Fill vs. Maximum Dropping Probability, ρ=1.2

Queue occupancy decreases with increased aggressiveness in the RED dropping parameters, and lower queue fill also results in lower delay for those packets entering the queue. A slight cost is incurred, relative to tail drop, in overall drop probability. Now that a range of acceptable parameters has been found, RED settings are chosen to be minth=50 and maxp=0.5 to maximize fairness and the simulation used to create Figure 1 is recreated using RED to provide a comparison of tail drop to RED:

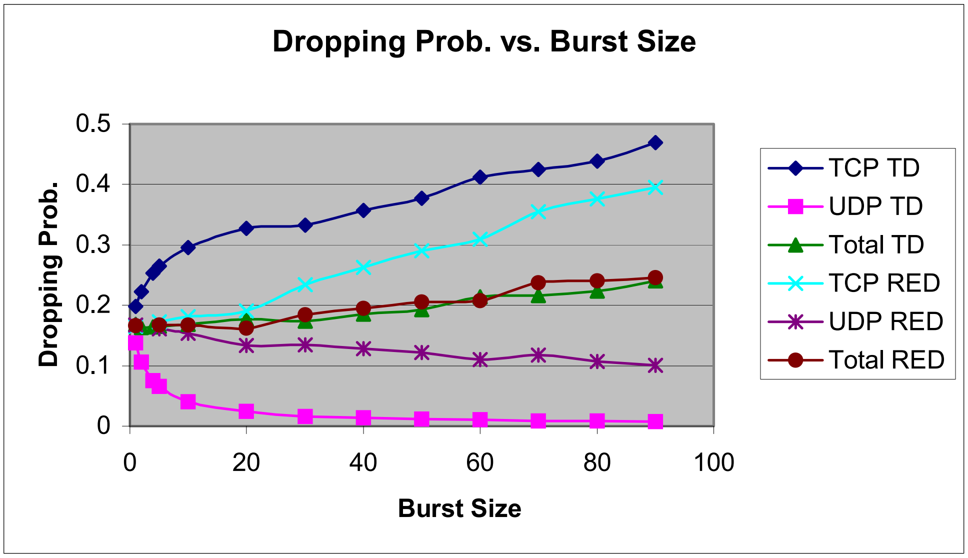

Figure 17: Dropping Probability using Tail Drop and RED, ρ=1.2, Queue Size=100, Mean Exponential Packet Length=1000, Link Rate = 10000

For smaller burst sizes, RED greatly increases fairness for the TCP traffic stream. As the burst size becomes closer to the queue size, the dropping probability for bursty traffic approaches that of a tail drop case.

5.3 WRED Simulation with Class Distinction

RED has been shown to increase fairness for mixed traffic with relatively small burst sizes. One possible use of WRED is to increase fairness beyond what is possible through RED. With a class-specific RED configuration, it should be possible to overcome traffic bursts and provide equitable service quality to multiple classes of traffic. The following simulation shows the effect of decreasing the minimum dropping threshold for smooth UDP Traffic by the burst size, ten packets, relative to TCP traffic. The goal is to provide a near-equal dropping probability for a burst size of ten packets. Simulation parameters are the same as those listed in Table 5 and produce the following result:

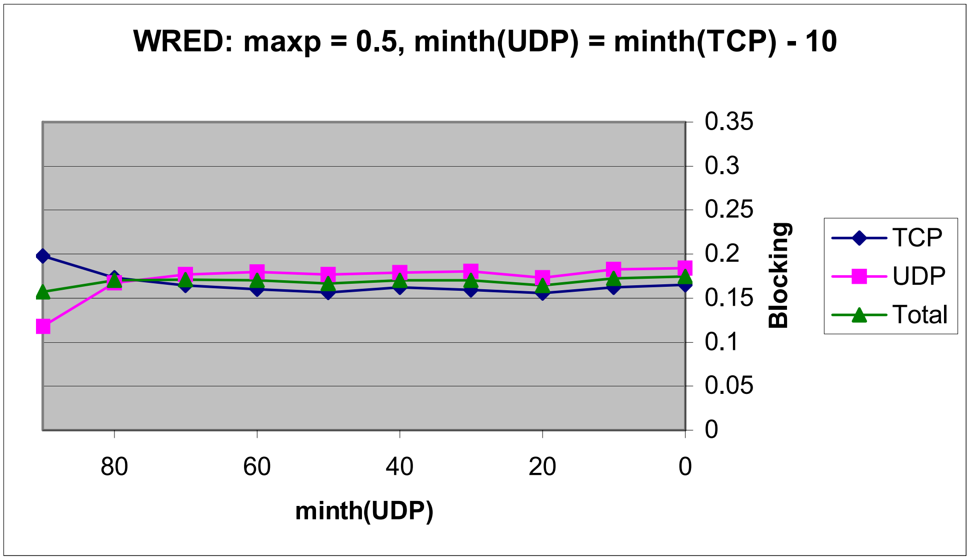

Figure 18: WRED Dropping Probability with Ten Packet Minimum Threshold Preference for TCP Traffic, maxp=0.5, ρ=1.2

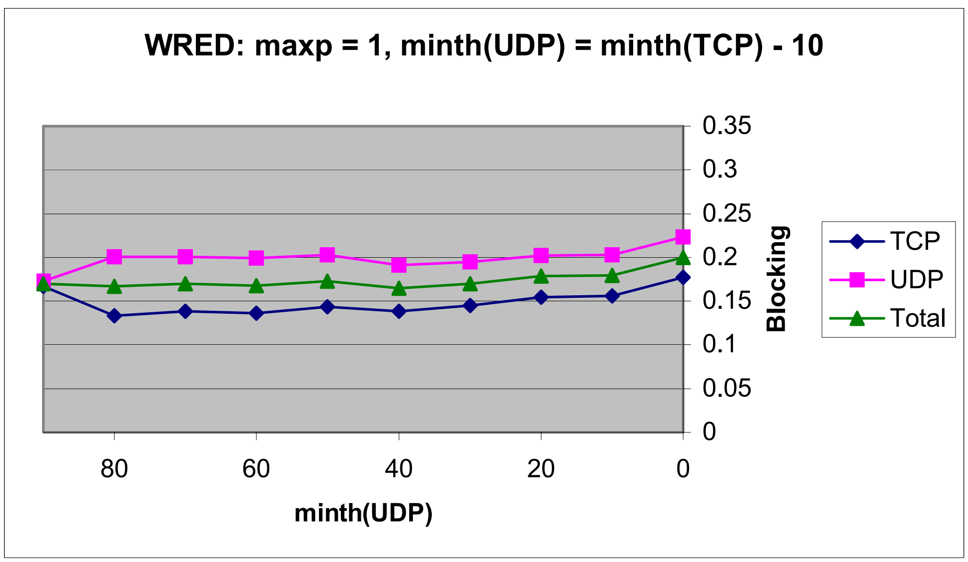

Figure 19: WRED Dropping Probability, Ten Packet Minimum Threshold Preference for TCP Traffic, maxp=1, ρ=1.2

Across the range of most min_{th} values, a ten packet difference in minimum threshold overcompensates and produces a more favorable dropping probability for TCP than for the UDP packets, which are dropped at a lower queue fill. Overall fairness is slightly improved for maxp = 0.5 as compared to the RED dropping with no class preference in Figure 14. To make the dropping probabilities even more similar, the difference between minimum thresholds is lowered to five packets, or one half of the burst size:

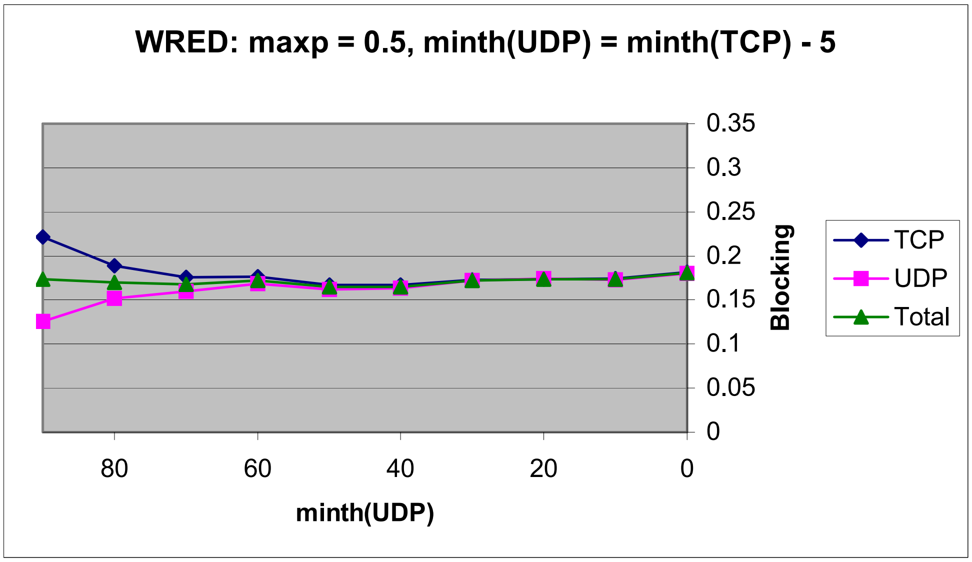

Figure 20: WRED Dropping Probability, Five Packet Minimum Threshold Preference for TCP Traffic, maxp=0.5, ρ=1.2

Providing a minimum threshold preference of five packets for a TCP source with a burst size of ten produces near equality with respect to dropping probability when WRED is sufficiently aggressive. Overall dropping probability is not greatly affected by this WRED implementation. The preference of five packets is maintained with minth(UDP) = 50 and minth(TCP) = 55 and then investigated for sensitivity to overall offered load.

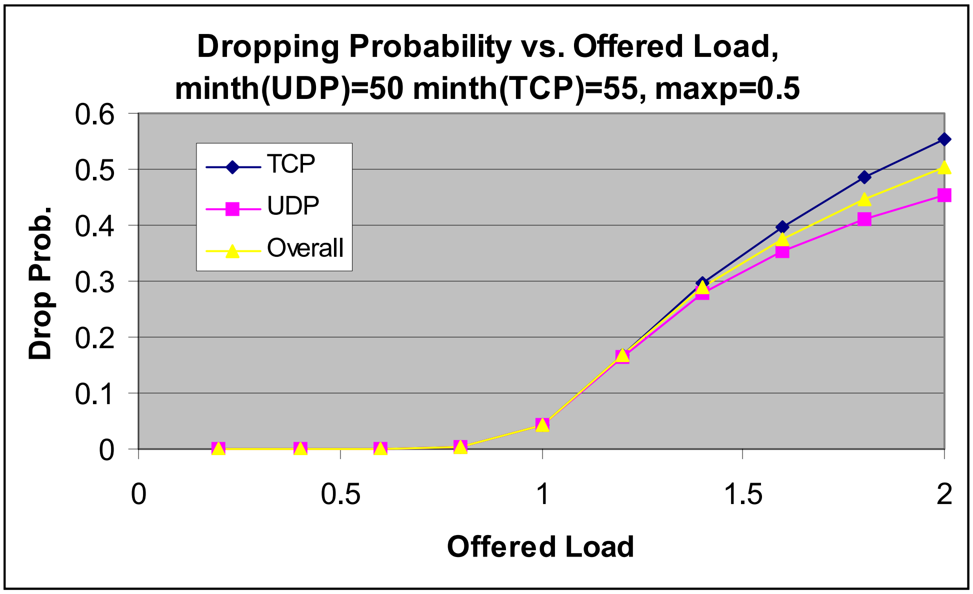

Figure 21: WRED Fairness vs. Offered Load

Figure 21 shows that a system can exhibit excellent fairness characteristics across a normal range of operating loads, in this case from ρ=0 to ρ=1.2, but it may struggle to maintain fairness for very high loads. This holds true when the traffic load is split 50/50, but it is worthwhile to investigate what happens as the traffic composition becomes biased towards the bursty or smooth side. Figure 22 shows how fairness is affected by the predominance of one type of traffic over the other at ρ=1.2:

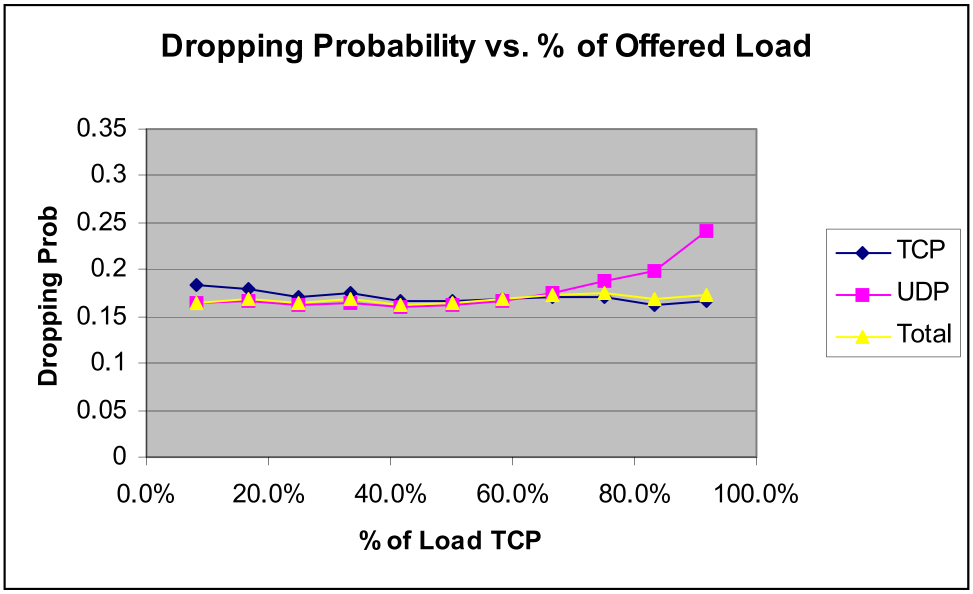

Figure 22: Class Dropping Probabilities vs. % Of Traffic That is TCP, minth=55/50, maxp=0.5, ρ=1.2

The traffic-weighted overall dropping probability is not greatly affected by the makeup of the traffic. However, the type of traffic that is less dominant experiences a greater chance of dropping, particularly for UDP packets when TCP makes up more than about 70% of the traffic load. When establishing WRED parameters, it may be desirable to compensate for this disparity if the system will be operating with a substantial majority of the traffic belonging to a single class for a long period of time.

5.4 Congestion and Delay Considerations for WRED

In addition to the increased fairness across different classes of traffic, RED claims to offer better performance with respect to congestion and delay. Increased aggressiveness in early discard will decrease the mean queue fill, and thus decrease the amount of waiting time a packet will experience in the queue. A simulation is performed using the best parameters found in Section 5.3, including a five packet TCP preference in

min_{th} and max_p=0.5, and the delay is found to decrease with lower minimum discard thresholds:

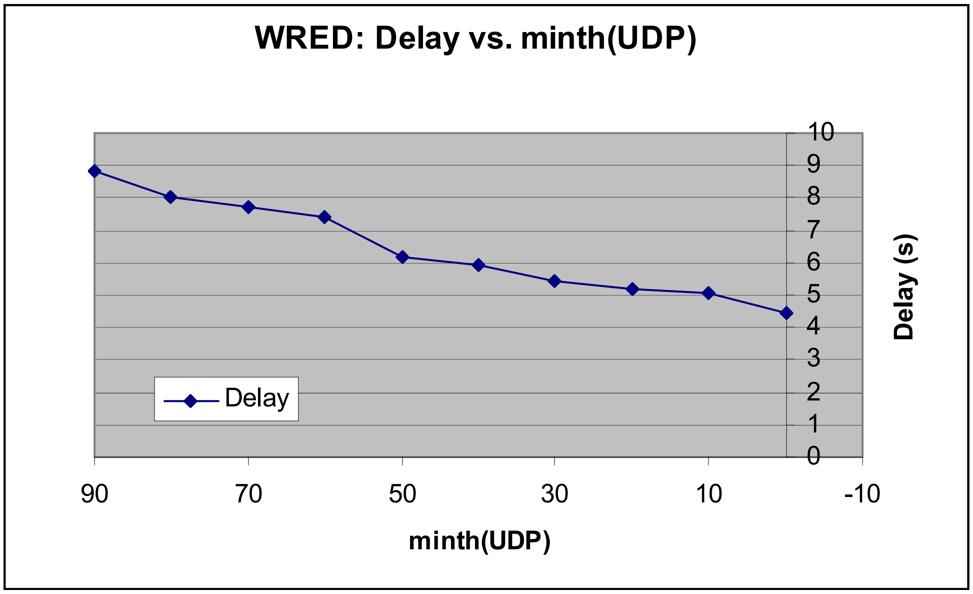

Figure 23: WRED Packet Delay vs. minth(UDP) , ρ=1.2

As mentioned in [3], RED will result in a lower percentage of drops occurring from queue overflow. The parameters used for the delay measurement are repeated to find the percentages of total drops resulting from buffer overflow for each class:

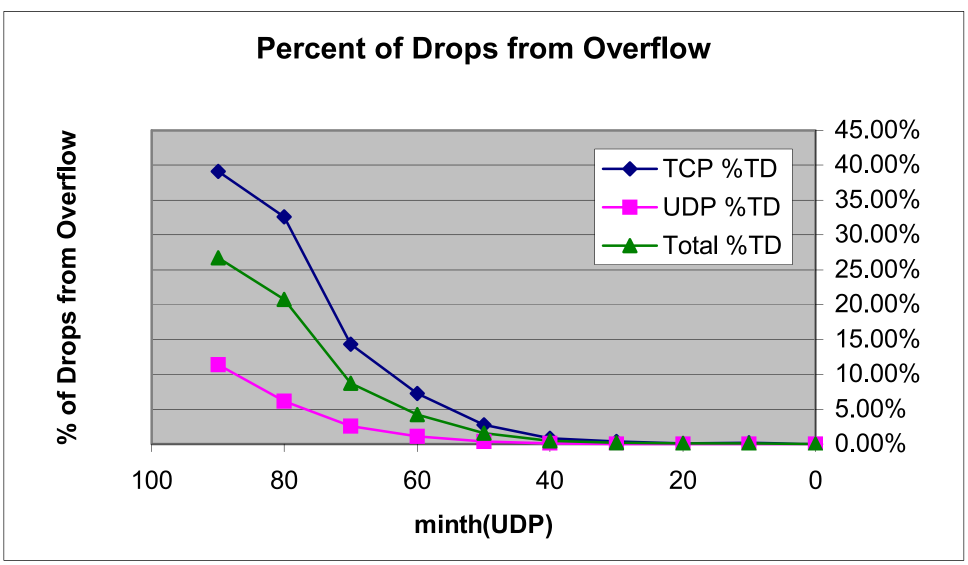

Figure 24: WRED: Queue Overflow vs. minth(UDP) , ρ=1.2

As expected, the percentage of packets dropped from overflow decreases as WRED is implemented in a more aggressive manner. With less aggressive RED, TCP packets are much more likely to overflow the buffer due to burst arrivals at high queue fill states. WRED dropping lowers the average queue fill and results in a system that drops packets early in a probabilistic random fashion instead of the discard of the tail-drop case, in which drops suddenly occur when the queue becomes full. As a result, the scenario of complete network-wide congestion is less likely to occur with a sufficiently aggressive RED queue than with a tail drop queue.

Section 6: Final Conclusions and Future Work

6.1 Conclusions

WRED retains many of the beneficial attributes of RED while adding additional configuration options. Through an iterative process, WRED parameters were found that provide close to an exact match in dropping probability for two classes with different arrival statistics. Many of the benefits of RED that were promoted in [3] were shown to be true for this WRED model, and the cost in overall dropping probability, as compared to a tail drop queue, was relatively low.

It was discovered that the WRED parameters were not required to be extremely aggressive in order to receive equal dropping probability during slight overload conditions, provided that the minimum thresholds are correctly distinguished. The use of WRED min_{th} parameters of approximately one half of the queue size provided dropping probabilities for the two classes that are almost equal. More aggressive WRED parameters led to increased total packet dropping, and were not shown to provide much additional benefit with the burst sizes used in the project.

Proper parameter settings were shown to provide strong fairness characteristics over most realistic traffic loads when the balance of smooth and bursty traffic was similar to their ratio when the parameters were initially found. If one class of traffic dominates, the other class may face increased blocking probability, particularly if bursty TCP traffic is predominant. The overall traffic-weighted mean dropping probability was not greatly affected.

The WRED extensions of the Bonald RED analytic work were also used to provide bounds for the dropping probability of the simulation model. The Extend model is found to obey these bounds.

6.2 Future Investigations

This project focused on the performance of an individual queue within a larger system. Some of the RED claims, particularly regarding TCP synchronization, would be best investigated by implementing WRED in a multi-node network simulation. Source models could be constructed that follow TCP specifications more precisely. If these models were created, it would be worthwhile to investigate the use of WRED to mark packets and signal the TCP source to reduce its load, instead of simply dropping the packets. The WRED system would then becomes a feedback control system that would require further analysis. Some initial work into this type of system has already begun for the RED case [10].

Section 7: Works Cited

[1] D. Clark, W. Lehr, et al., "Provisioning for Bursty Internet Traffic: Implications for Industry and Internet Structure." MIT ITC Workshop on Internet Quality of Service. Nov, 1999. http://www.ana.lcs.mit.edu/papers/PDF/ISQE_112399_web.pdf

[2] J. Cao, W. Cleveland, et al., "Internet Traffic Tends Toward Poisson and Independent as the Load Increases." *Nonlinear Estimation and Classification*. http://cm.bell-labs.com/cm/ms/departments/sia/doc/lrd2poisson.pdf

[3] Floyd, S. & Jacobson, V. (1993). Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*. Vol. 3. 397-413.

[4] T. Bonald, M. May, et. al., "Analytic Evaluation of RED Performance." *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. IEEE , Volume: 3 , 26-30 March 2000
Page(s): 1415 -1424 vol.3

[5] P. Kuusela and J.T. Virtamo, "Modeling RED with Two Traffic Classes." http://keskus.tct.hut.fi/tutkimus/com2/publ/ntsRED2cl.pdf

[6] Feng, W.-C.; Kandlur, D.D.; Saha, D.; Shin, K.G. A self-configuring RED gateway *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.* IEEE , Volume: 3 , 21-25 March 1999
Page(s): 1320 -1328 vol.3

[7] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self-Similar Nature of Ethernet Traffic", *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp 1-15, February 1994.

[8] Extend Model Software & Documentation, http://www.imaginethatinc.com, 2002.

[9] D. Petr, University of Kansas EECS 963 Class Notes. (Summer 2003)

[10] Firoiu, V.; Borden, M. "A study of active queue management for congestion control." *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.* IEEE , Volume: 3 , 26-30 March 2000
Page(s): 1435 -1444 vol.3