**Using TCP Models To Understand Bandwidth
Assurance in a DiffServ Network**

by

Mandeep Singh Baines

B.Eng., University of Victoria, 1999

A thesis submitted to
the Faculty of the Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of System and Computing Engineering
Carleton University

2002

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

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

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

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

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

0-612-79690-6

Canada

This undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

Using TCP Models To Understand Bandwidth Assurance in a DiffServ Network

submitted by Mandeep Singh Baines , B Eng.
in partial fulfillment of the requirements for
the degree of Master of Applied Science

_____
Thesis Supervisor
_____

Chair, Department of Systems and Computing Engineering

Carleton University

Date _____

Abstract

The standardization of Differentiated Services Assured Forwarding (AF) Per Hop Behavior (PHB) provides a scalable means of deployment of QoS in IP networks. Recent studies have shown unexpected results in terms of the achieved bandwidth of TCP flows in such networks. This thesis presents a comprehensive analytical model to predict the bandwidth achieved by aggregates of TCP flows. The model predicts achieved bandwidth in four different cases: an over-provisioned network, an under-provisioned network, a near-provisioned network and a network with three levels of drop-precedence. In developing the model, all parameters used are measurable using standard tools and information available from routers and network management tools in today's networks.

Simulation is used to establish the validity of the model and understand its scope of applicability and limitations. The model explains why excess bandwidth is biased based on factors such as RTT, packet size, and CIR. Finally, there is a novel extension of the model to predict the bandwidth of TCP flows in a network with multiple congested nodes.

Acknowledgements

I would like to thank my advisor Mike Devetsikiotis for guidance and patience. I would also like to thank Nabil Seddigh, Peter Piedad, and Biswajit Nandy of Tropic Networks, without whose help this research would neither have been started or finished.

Table of Contents

Chapter	
1	Introduction 1
1.1	Background 1
1.2	Previous Research 3
1.3	Motivation and Objectives of this Thesis 6
1.4	Thesis Organization 7
2	Assured Forwarding 8
2.1	RFC 2597 9
2.2	Multi-Level Random Early Detection 11
2.3	Packet Marking 13
3	Basic Performance Model for TCP 16
3.1	TCP Congestion Avoidance 16
3.2	A Basic Model of the TCP Congestion Avoidance Algorithm 17
3.3	Model Assumptions 19
3.4	Extending the Basic TCP Model for Aggregate of Flows 19

4	Simple TCP Model for DiffServ	22
4.1	Over-provisioned Case	22
4.2	Near-provisioned case	24
4.3	Under-provisioned Case	25
4.4	Comparison with Other Models	25
5	Simulation and Observations	28
5.1	Methodology	29
5.2	Observations	30
6	Multiple Congested Nodes	36
6.1	Simulation	37
7	Three Drop-precedence Levels	41
7.1	Case 1: PIR Over-provisioned	41
7.2	Case 2: CIR Over-provisioned	42
7.3	Case 3: Under-provisioned	43
7.4	Simulation	43
8	Conclusions and Recommendations for Further Study	46
8.1	Conclusions	46
8.2	Recommendations for Further Study	49
	Bibliography	51

List of Tables

Table

6.1	Percentage Error for TCP SACK Predictions	40

List of Figures

Figure

3.1	Impact of NoF	21
5.1	Simulation Topology	29
5.2	CIR versus Achieved Bandwidth	32
5.3	RTT versus Achieved Bandwidth	33
5.4	Number of Flows versus Achieved Bandwidth	34
5.5	Packet Size versus Achieved Bandwidth	35
6.1	Multiple Congested Nodes	38
6.2	Bandwidth versus Number of nodes for TCP SACK	38
7.1	Three Drop Simulation	44
7.2	Drop Distribution	45

Chapter 1

Introduction

1.1 Background

The rise in popularity of the Internet coupled with new applications such as voice and video has fueled the need for research to improve the Quality of Service (QoS) delivered by today's IP networks. In the current single class best-effort service model, all user packets compete equally for network resources. By providing multiple levels of service over the same physical network, Internet Service Providers (ISPs) can increase revenues and satisfy customer demand for better QoS. One proposed way of providing multiple levels of service is the Differentiated Services (DiffServ) architecture [1].

DiffServ relies on packet marking and aggregated treatment of packets on a per-class basis. It operates on the premise that complicated functionality should be moved toward the edge of the network with very simple functionality at the core. Packets are marked at the edge of the network with a DiffServ code point (DSCP). Each DSCP corresponds to a different forwarding treatment, or Per-Hop Behavior (PHB), in the core. The IETF's DiffServ Working Group has defined two PHB groups: the Expedited Forwarding (EF) PHB [10] and the Assured Forwarding (AF) PHB [7]

group.

This thesis focuses on the latter PHB group. Typically, an ISP would use the AF PHB to provide a service where customer packets from a site are forwarded with high probability as long as the aggregate traffic from the site does not exceed a previously agreed upon target rate profile, or Committed Information Rate (CIR). The site may exceed the profile with the understanding that the excess traffic is not delivered with as high a probability as the traffic that is within the profile.

The AF PHB group is divided into four independently forwarded classes. Within each class, a packet is assigned to one of three different levels of drop precedence. However, during times of congestion, packets with higher drop precedence are forwarded with higher probability, or dropped with lower drop probability, than packets with lower drop precedence.

MRED-based (Multi-Level Random Early Detection) [11] Active Queue Management (AQM) schemes are one way to provide Assured Forwarding. The basis of the MRED mechanism is RED-based [6] differentiated (i.e. according to precedence level) dropping of packets during congestion at the router.

Though the AF PHB has become standardized, there are still some open issues that need to be understood. The most important question relates to the kind of end-to-end service that can be created using such schemes. There is concern that AF should show some measure of predictability and fairness of service for paying customers.

If subscribers were charged based on their CIR then, in a fair system, one would expect subscribers who have paid the same amount to obtain equal bandwidth. Unfortunately, this is not the case. Earlier work [17, 15] has shown that bandwidth

is shared in a manner that is not totally dependent on the CIR.

In [17], Yeom and Reddy use simulation to study the interaction between the transport protocol and the drop policy of the network in realizing throughput guarantees. They investigate the interaction between TCP and RIO-based differentiated dropping. In addition they propose and evaluate a new transport protocol and drop schemes to provide better realization of throughput guarantees. In their analysis of the interaction of TCP and RIO, they show that unfairness exists in the distribution of excess bandwidth in a Differentiated Services network.

In [15], Seddigh et al. carry out a detailed experimental study of five factors that impact throughput assurances for TCP and UDP flows in a RIO based Differentiated Services network. The five factors are RTT, number of flows in an aggregate, size of the target rate, packet size, and presence of non-responsive UDP traffic. Their main conclusion is that in an over-provisioned network, all target rates are achieved regardless of the five factors. However, the five factors clearly affect the distribution of excess bandwidth in an over-provisioned network.

Both studies suggests that it is possible for subscribers with the same CIR to obtain dramatically different shares of the excess, or uncommitted, bandwidth. Customers will be unhappy to know that another customer who has paid for the same CIR is getting significantly more bandwidth than they are.

1.2 Previous Research

Since CIR is not the single determining factor in how bandwidth is distributed in a DiffServ network, it is important to understand exactly how the CIR and the five factors mentioned earlier affects bandwidth distribution. In the literature there

are many studies which have looked at this problem in a best effort context.

In [5], Sally Floyd presents a simulation and heuristic analysis of the problem. She explores the bias in TCP/IP networks against connections with multiple congested nodes, connections with longer RTTs, and the bias of Drop Tail and Random Drop gateways against bursty traffic.

In her simulation, she investigates a topology with n congested gateways and $n + 1$ connections, labeled connections 0 through n . Connection 0 goes through all congested gateways while connections 1 through n go through only one congested gateway each. She concludes that the TCP's Increase-by-One window-increase algorithm can result in unacceptably low throughput for a long connection with multiple congested gateways. However, using simulation with a Constant-Rate window-increase algorithm she shows that this low throughput is not unavoidable.

In order to explain this result, she develops a heuristic algorithm, presented below, which computes the bandwidth share, p , of a connection in terms of RTT ratio, $\frac{r_0}{r_1}$ (where r_0 is the RTT of connection 0 and r_1 is the RTT of connection 1), and number of congested links traveled, n . From equation 1.1 it is easy to see that the network bias against long RTT connections is the major factor in connections with multiple congested nodes achieving low throughput.

$$p = \frac{1}{1 + \sqrt{n} * \frac{r_0}{r_1}} \quad (1.1)$$

This "heuristic analysis" was one of the first attempts at developing a performance model for the steady-state behaviour of TCP. Many later studies have proposed steady-state analytical throughput models of TCP in best-effort networks. Some of the earliest work was done in [12], where Mathis et al. developed a steady-state

throughput model of the saw-tooth behavior of the TCP Congestion Avoidance algorithm.

Under the assumption of constant RTT, low to moderate packet loss, and random constant drop probability, the congestion window (cwnd) follows a perfect saw-tooth. Periodically the congestion window rises to a maximum value at which time packet loss causes the window to be halved due to the fast retransmit mechanism. By counting the number of packets sent in one saw-tooth and measuring its period, Mathis et al were able to come up with the following equation for bandwidth of a single TCP flow:

$$BW = \frac{MSS}{RTT} \frac{C}{\sqrt{p}} \quad (1.2)$$

where MSS is the maximum segment size, C is a constant which is dependent on the type of TCP, and p is the packet drop probability.

In [12], a number of situations are pointed out where the model is not expected to apply. A few of these include the case where the data receiver is announcing too small an offered window or the sender does not always have data to send. Also, the model does not account for timeouts or short connections that don't reach steady state.

In [13], Padhye et al. develop a model which captures not only the behavior of TCP's fast retransmit mechanism but also the effect of TCP's timeout mechanism on throughput. Their model is more accurate than the one developed by Mathis, especially for the case of networks where there are many timeouts. However, in this paper the authors ignore the effect of timeouts.

In [14] and [16], the authors propose simple models of TCP behavior in a Diff-Serv networks. In [14], Sahu et. al propose a single flow model for TCP with token-

bucket marking. In [16], Yeom and Reddy propose a model for TCP with short term rate-based marking. Here, we propose a model for TCP aggregates with long term rate-based marking. Long term rate-based marking is known [2] to work well with flow aggregates while short-term rate based marking can perform better for a single flow.

1.3 Motivation and Objectives of this Thesis

This paper proposes to extend [12] and factor the CIR into a new TCP performance model, which looks at TCP aggregates in a RIO-based DiffServ network.

There are four main contributions of this work:

- (1) We extend previous work in best-effort TCP modeling and develop a formula for the steady-state throughput of an aggregate of long-lived TCP flows that subscribe to a service based on the AF PHB. In this formula, we include the CIR as one of the key factors affecting throughput.
- (2) We use the formula to make key observations about how the individual factors affect the throughput. Some of these observations have already been presented in other papers [15, 17] (as observations of simulation results or test network results). Using the analytical model, we confirm and explain these previous observations. Such work can have a significant contribution to better understanding how bandwidth is distributed in an AF network and how TCP congestion avoidance causes unfair bandwidth distribution.
- (3) We present a novel extension of our work in which we examine the effect of multiple congested nodes on bandwidth assurance.

- (4) We extend our own work to look at the case where there are three levels of drop precedence.

1.4 Thesis Organization

The remaining chapters of the thesis are organized as follows:

- Chapter 2:** Reviews the fundamentals of the Assured Forwarding PHB: packet metering, marking, and MRED-based differentiated dropping.
- Chapter 3:** Presents the original TCP performance model developed by Mathis et al. Also, the early model is extended to cover the case of the aggregation of many TCP flows. This earlier work is the basis for the development of the model presented in this thesis.
- Chapter 4:** Develops a new performance model which predicts achieved bandwidth of TCP flow aggregates in a RIO-based DiffServ network.
- Chapter 5:** Simulation and analysis are used to develop key theories on achievable bandwidth in a DiffServ network. The five factors are all looked in detail in regards to how they bias achieved bandwidth.
- Chapter 6:** Presents a novel extension of the work and looks at the case of multiple congested nodes and how this factor biases achieved bandwidth.
- Chapter 7:** Extends the analysis to the particular case of three levels of drop precedence. The AF PHB is made up of 4 classes with three levels of drop precedence per class.
- Chapter 8:** Conclusions and Recommendations for further study.

Chapter 2

Assured Forwarding

The AF PHB can be used to develop an assured IP service where packets are forwarded with high probability as long as they are within the profile or committed information rate (CIR). Packets that exceed the CIR, out of profile packets, are still forwarded but with the understanding that excess traffic is delivered with lower probability. It also guarantees that the network does not reorder packets that belong to the same microflow, no matter if they are in or out of profile.

The AF PHB group is divided into four independently forwarded classes. Within each class, a packet is assigned to one of three different levels of drop precedence. However, during times of congestion, packets with higher drop precedence are dropped with higher probability, or forwarded with lower probability, than packets with lower drop precedence.

Generally, the router at the edge of the network determines which packets are assigned to which class of service and which level of drop precedence, based on a service level agreement (SLA), which has been worked out a priori between the ISP and customer. Packets are assigned by the edge router, to the appropriate class and drop precedence level by packet marking. Basically, the Type of Service (ToS) field of the IP header is used to store the mark or DiffServ codepoint (DSCP) which is

used by routers in the core to give the packet the appropriate treatment.

Packets in different classes are forwarded independently of one other. A DS node does not have to implement all four general case AF classes. However, packets in one class should be forwarded independently of packets in another class (independent scheduling, reserved bandwidth). This is accomplished by reserving resources (bandwidth, queuing) for each class a priori. The drop precedence of a packet is used during times of sustained congestion to determine its drop priority. A congested node protects packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value.

In a DS node, the level of forwarding assurance of an IP packet thus depends on (1) how much forwarding resources has been allocated to the AF class that the packet belongs to, (2) what is the current load of the AF class, and, in case of congestion within the class, (3) what is the drop precedence of the packet.

2.1 RFC 2597

The Assured Forwarding standard is defined in RFC 2597. Below is a brief summary of some of the requirements taken from the RFC.

Requirement 1 Within each AF class, a DS node MUST accept all three drop precedence codepoints and they MUST yield at least two different levels of loss probability. If a DS node only implements two different levels of loss probability for an AF class x, the codepoint AFx1 MUST yield the lower loss probability and the codepoints AFx2 and AFx3 MUST yield the higher loss probability.

In networks where long-term congestion is rare, it may be reasonable for a DS node to implement only two different levels of loss probability. And since the user is

primarily concerned with protecting his agreed upon CIR, it makes sense to protect these packets from the lower drop precedence packets.

Requirement 2 A DS node MUST NOT reorder AF packets of the same microflow when they belong to the same AF class regardless of their drop precedence.

In order to preserve packet ordering, one could simply store all packets in an AF class in a single queue. This would not only preserve packet ordering for packets in a microflow but for all packets in general.

Requirement 3 An AF implementation MUST detect and respond to long-term congestion within each class by dropping packets, while handling short-term congestion (packet bursts) by queuing packets.

This requires a mechanism for detecting long-term congestion. How do we measure congestion? Often the presence a queue forming is taken to suggest the presence of congestion. Therefore, we can measure long-term congestion by using filtering or smoothing techniques to remove short-term spikes and calculate the long-term average queue size.

Requirement 4 The dropping algorithm MUST be insensitive to the short-term traffic characteristics of the microflows using an AF class.

Flows with different short-term burst shapes but identical longer-term packet rates should have packets discarded with essentially equal probability. This requirement is somewhat related to the third requirement except that now we are dealing with flows. This is a somewhat more difficult problem to solve. Flows that are overly bursty might experience more drops than a less bursty connection.

Requirement 5 The dropping algorithm MUST treat all packets within a single class and precedence level identically.

This means that at any given point in time, all packets within a precedence level are dropped with equal probability. Therefore, the discard rate of a particular microflow's packets within a single precedence level will be proportional to that flow's percentage of the total amount of traffic passing through that precedence level.

Requirement 6 The congestion indication feedback to the end nodes, and thus the level of packet discard at each drop precedence in relation to congestion, MUST be gradual rather than abrupt, to allow the overall system to reach a stable operating point.

This suggests a relation between dropping and level of congestion with more packets being dropped at higher levels of congestion.

2.2 Multi-Level Random Early Detection

All of the requirements from the previous section seem to suggest a MRED (Multi-Level Random Early Detection) [11] based implementation. This is no coincidence since the Assured Forwarding PHB was designed with a MRED-based implementation in mind. Actually, there are no non-MRED based implementations of the AF PHB that the author is aware of.

The basis of the MRED mechanism is RED-based [6] differentiated (i.e. according to precedence level) dropping of packets during congestion at the router. RED is based on detecting incipient congestion and providing feedback to end hosts by dropping the packets.

It operates based on average queue length, avg , that is calculated using an exponential weighted moving average of the instantaneous queue length. Equation 2.1, below, is a smoothing function (low-pass filter) that can be used to calculate the average queue size. The factor w_q determines the time constant of the low-pass filter. q is the instantaneous queue size at a packet arrival.

$$avg = (1 - w_q)avg + w_qq \quad (2.1)$$

RED drops packets with probability, p , given by equation 2.2 which is dependent on the the average length of the queue, avg , from equation 2.1. The drop probability increases from 0 to maximum drop probability (max_p) as the average queue size increases from minimum threshold (min_{th}) to maximum threshold (max_{th}). If average queue size exceeds max_{th} , all packets are dropped.

$$p = \frac{avg - min_{th}}{max_{th} - min_{th}} max_p \quad (2.2)$$

MRED is a generic term used to describe any scheme where drop probability for packets with different drop precedence needs to be calculated independently for each drop precedence. This is achieved by maintaining multiple sets of RED thresholds - one for each drop precedence. In this way, it is possible to provide many different levels (in this case two or three) levels of drop precedence while utilizing only one queue.

The average queue used in the drop decision can be calculated using a number of different schemes. The parameters for packets of different drop precedence levels can generally be set in three ways: (1) partially overlapped (2) overlapped and (3) staggered. A staggered approach seems to provide the best performance [11].

There are many different flavours of MRED, they differ mainly in how they calculate the average queue size, *avg*, for the different levels of drop precedence. The two most popular in the literature are WRED (Weighted RED) and RIO-C (RED with In/Out and Coupled Average Queues). WRED calculates a single average queue length that includes arriving packets of all drop precedence levels. In RIO-C, the average queue length for packets of different drop precedence can be calculated by adding its average queue to the average queue of packets of lower drop precedence.

The analysis in this paper is done independent of flavour of MRED. However, the simulation study presented in chapter 5, uses a RIO-Ci implementation.

2.3 Packet Marking

The Assured Forwarding PHB is composed of two main ingredients: MRED-based Active Queue Management in the core and packet marking based on SLAs at the edge. In this section, different packet marking approaches will be looked at in greater detail.

In the literature, there are two different types of markers: token-based and window-based [2].

For the case of two levels of drop precedence, the token-based approach works such that token are accumulated (into a bucket) at a constant bit-rate up to a maximum (the bucket size). If at a packet arrival, the number of tokens in the bucket is greater than or equal to the packet size, the packet is marked with the highest drop precedence and that many tokens are removed from the bucket. If there are not enough tokens the packet is marked with the lowest drop precedence and no tokens are removed from the bucket.

There are two algorithms which extend the token-based marker to three levels of drop precedence: the Single Rate Three Color Marker (srTCM) described in [8] and the Two Rate Three Color Marker (trTCM) which is described in [9].

The approach we will be looking at for our analysis is the sliding window approach [2]. It is divided into two components: a meter and a marker. For the meter, an exponential weighted moving average (EWMA) rate estimator is used to smooth out the burstiness of TCP traffic as well as to be sensitive to instantaneous sending rates. Presented below, is the formula used:

$$rate_1 = \frac{pktsize + (rate_0 * window)}{\Delta t + window} \quad (2.3)$$

where $rate_1$ is the newly calculated rate, $rate_0$ is the previously calculated rate, $window$ is the window size in units of time, Δt is the time since the last packet arrival and $pktsize$ is the size of the currently arrived packet.

In terms of marking algorithms there are two different approaches proposed. The first approach is to use a relatively long window - on the order of a TCP sawtooth - and mark packets as lower precedence with probability $P = (rate_1 - CIR)/rate_1$ when the average rate exceeds CIR. The second approach is to use a small window - on order of an RTT - and look for when the rate exceeds 1.33 CIR, and then mark all packets with lower drop precedence. A scheme for extending this approach to three levels of drop precedence is described in [3]. The second approach tries to track the TCP sawtooth of a single TCP flow while the first approach is more general and can be applied to flow aggregates and other types of traffic. In our simulations we use the first approach.

Though the AF PHB has become standardized, there are still some open issues that need to be understood. The most important question relates to the kind of

end-to-end service that can be created using such schemes. There is concern that AF should show some measure of predictability and fairness of service for paying customers. In the next section we begin the development of our performance model by looking at Mathis et al.'s work [12] in TCP modeling.

Chapter 3

Basic Performance Model for TCP

3.1 TCP Congestion Avoidance

In [12], the authors develop an analytical model for the TCP flow-control algorithm. TCP uses a window-based flow-control approach. The receiver advertises a window to the sender, which is the largest amount of unacknowledged data it is willing to receive. This value is usually set to the amount of buffer the receiver has available for receiving packets.

The sender maintains a window of its own known as the congestion window, or *cwnd*. The sender can send $\lfloor cwnd \rfloor$ packets unacknowledged. *cwnd* starts with a size of size 1 and increases exponentially until the window size reaches a threshold, *ssthresh*, and increases linearly after that.

In the exponential (or slow-start) phase, a source increases its window by 1 every time it receives an acknowledgment. If every packet is acknowledged, *cwnd* doubles every RTT. In the linear (or congestion-avoidance) phase, the sender increases its window by $1/\lfloor cwnd \rfloor$ every time it receives an acknowledgment. This increases the window by 1 every RTT. A threshold variable called the slow-start threshold or *ssthresh* controls the transition from the exponential to the linear phase. The sender

initializes this variable to half the initial window size, and, on detecting a loss, resets it to half the current window size.

The two most widely deployed variants of TCP are TCP Tahoe and TCP Renoe. The Tahoe version detects losses using timeouts. On a timeout, it decreases its flow control window to 1, sets *ssthresh* to half the current window size, and enters slow-start. The window rises exponentially to half its previous value, then continues with a linear increase.

The Reno version detects losses using both timeouts and the receipt of three acks with the same sequence number. On a timeout, a TCP Reno source behaves in the same way as a TCP Tahoe source. However, on a fast retransmit, TCP Renoe decreases both *ssthresh* and its *cwnd* to half its previous value. At this point, since *cwnd* is already as large as *ssthresh*, the source goes directly into congestion avoidance, skipping the slow-start phase.

3.2 A Basic Model of the TCP Congestion Avoidance Algorithm

The model developed in [12] applies whenever TCP's performance is determined primarily by the congestion avoidance phase. The model does a good job predicting the bandwidth of a sustained TCP connection subject to light to moderate packet loss. It assumes that TCP avoids retransmission timeouts and always has sufficient receiver window and sender data.

The model works by measuring the size of the sawtooth which is created by the additive increase and multiplicative decrease behavior of the TCP congestion avoidance algorithm. Assume that TCP is running over a lossy path which has a constant round trip time (RTT) because it has sufficient bandwidth and low enough

total load that it never sustains any queues. Also, assume that there is random packet loss at constant probability, p . Therefore, in the average approximately $\frac{1}{p}$ consecutive packets are sent between drops.

Under these assumptions the congestion window traverses a perfectly periodic sawtooth. If we let the maximum value of the window be W packets, then the minimum window must be $\frac{W}{2}$ packets. If the receiver is acknowledging every packet, each cycle must be $\frac{W}{2}$ round trips, or $RTT \frac{W}{2}$ seconds long. The total data delivered is the area under the sawtooth, which is:

$$\left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2 = \frac{3}{8}W^2 \quad (3.1)$$

By assumption, each cycle delivers $\frac{1}{p}$ packets (neglecting the data transmitted during recovery). Solving for W :

$$W = \sqrt{\frac{8}{3p}} \quad (3.2)$$

Substitute W into the bandwidth equation below:

$$BW = \frac{\text{data per cycle}}{\text{time per cycle}} = \frac{MSS \times \frac{3}{8}W^2}{RTT \times \frac{W}{2}} = \frac{MSS/p}{RTT \sqrt{\frac{2}{3p}}} \quad (3.3)$$

Collect the constants in one term, $C = \sqrt{3/2}$, then we arrive at:

$$BW = \frac{MSS}{RTT} \frac{C}{\sqrt{p}} \quad (3.4)$$

3.3 Model Assumptions

The model is based on a number of assumptions. It is not expected to apply in situations where the congestion avoidance does not fully control TCP performance. Some of these situations are:

- (1) If the data receiver is announcing too small an offered window, then TCP's performance is likely to be dictated by the receiver's window and not at all by the Congestion Avoidance algorithm.
- (2) If the sender does not always have data to send, the model is not likely to apply.
- (3) The elapsed time consumed by TCP timeouts is not modeled. TCP implementations suffer from timeouts when they experience multiple packet losses within one round trip time. The model does not fit environments where there are many timeouts.
- (4) TCP implementations which use other window opening strategies (eg. TCP Vegas) will not fit the model.

3.4 Extending the Basic TCP Model for Aggregate of Flows

Because the AF PHB is concerned with aggregates of flows, we begin developing our model by extending equation 3.4 to handle an aggregate of flows with the same source and destination by summing the throughput of the individual flows. The equation for an aggregate of flows is then:

$$BW = \sum_{i=1}^{NoF} \frac{MSS_i}{RTT_i} \frac{C}{\sqrt{p_i}} \quad (3.5)$$

where NoF is the number of flows in the aggregate, MSS_i is the average packet size for TCP flow i , RTT_i is the average round trip time for TCP flow i , and p_i is the packet drop probability for TCP flow i .

We can construct a simple model for an aggregate of flows by assuming that all flows have the same average round trip time, $RTT_i = RTT$, the same average packet size, $MSS_i = MSS$, and the same packet drop probability, $p_i = p$. This is similar to the approach followed by Firoiu and Borden in [4].

We can now drop the index i and reduce the formula to the following:

$$BW = NoF \times \frac{MSS}{RTT} \frac{C}{\sqrt{p}} \quad (3.6)$$

We validated equation 3.6 by running simulations where we increased the number of flows per aggregate for every successive simulation run. As the number of flows per aggregate increased, the percentage error remained low. For each predicted point in the graph below, we used measured values for RTT , and p , while the other two factors, NoF and MSS , were kept constant and fixed beforehand.

The network topology used for simulation appears in Figure 3.1. The RED parameters were 10, 300, 0.05 with a physical queue size of 500. There were two aggregates competing for the 50 Mbps of bandwidth available at the bottleneck link. Both aggregates had the same round trip time, packet size, and number of flows. The "Measured" value is an average of the measured bandwidth of the two aggregates.

Figure 3.1: Impact of NoF

Chapter 4

Simple TCP Model for DiffServ

In this chapter, a performance model for MRED-based Assured Forwarding networks with two levels of drop precedence is presented. In developing the model, the work done by Mathis et al in [12] is extended by adding the CIR as a factor in determining achieved bandwidth. In addition to the assumptions made in [12] and presented in Section 3.3 of this paper, the following additional assumptions apply:

- (1) Non-overlapping RED thresholds. The maximum RED threshold for dropping out-of-profile packets is less than the minimum RED threshold for dropping in-profile packets.
- (2) Packets are marked at the edge of the network by a long-term TSW marker [2] based on a CIR.

An Assured Forwarding DiffServ network can be in one of three modes of operation: over-provisioned, under-provisioned and near-provisioned. In the following sections, these three cases will be looked at in turn.

4.1 Over-provisioned Case

A network is said to be over-provisioned when:

- (1) The cumulative sum of the CIRs of all of the aggregates using a link is less than the bandwidth of the link.
- (2) All packet drops are out-of-profile packets. There are no in-profile packet drops.

The probability of dropping an out-of-profile packet will be represented by the symbol p_{out} . The probability p_{out} can be related to p by the probability P_{out} , of marking a packet as out-of-profile. If we assume a long term rate-based [2] policer that marks CIR worth of packets in-profile over the long term, then packets should be marked out-of-profile with probability:

$$P_{out} = \frac{avgrate - CIR}{avgrate} = \frac{BW - CIR}{BW} \quad (4.1)$$

when the *avgrate* exceeds *CIR*. In the equation above, *avgrate* is a sliding window estimate of the bandwidth, *BW*.

An expression for p in terms of p_{out} can now be developed.

$$p = P_{out} \times p_{out} = \frac{BW - CIR}{BW} p_{out} = p_{out} - p_{out} \frac{CIR}{BW} \quad (4.2)$$

Equation 3.6 can now be redeveloped in terms of p_{out} . Replacing the p in equation 3.6 with the expression above results in the following equation:

$$BW = \frac{NoF \times MSS}{RTT} \frac{C}{\sqrt{p_{out} - p_{out} \frac{CIR}{BW}}} \quad (4.3)$$

Solving the quadratic, and rejecting the negative root ($BW > 0$):

$$BW = \frac{CIR}{2} + \sqrt{\left(\frac{CIR}{2}\right)^2 + \left(\frac{NoF \times C \times MSS}{\sqrt{p_{out}} RTT}\right)^2} \quad (4.4)$$

Note that equation 4.4 is equivalent to equation 3.6 when $CIR = 0$.

It is easy to derive an equation for the excess bandwidth (i.e. the bandwidth obtained beyond the CIR).

$$BW_{ex} = \sqrt{\left(\frac{CIR}{2}\right)^2 + \left(\frac{NoF \times C \times MSS}{\sqrt{p_{out}}RTT}\right)^2} - \frac{CIR}{2} \quad (4.5)$$

4.2 Near-provisioned case

A network is said to be near-provisioned when:

- (1) The cumulative sum of the CIRs of all of the aggregates using a link is close to (within 10%) than the bandwidth of the link.
- (2) Both in and out-of-profile packets are dropped.

One cannot always assume that there are no in-profile packets dropped. It is common to have both in and out-of-profile packets dropped when the network is near fully provisioned. At such a time, the queue size can fluctuate between the in-profile and out-of-profile random dropping zones. We begin by looking at the probability of having a packet marked in or out-of-profile.

$$P_{in} = \frac{CIR}{BW}, P_{out} = \frac{BW - CIR}{BW}$$

From this, it is possible to develop a relation between the p in equation 3.6 and p_{in} and p_{out} , which correspond to the probabilities of dropping in and out-of-profile packets respectively.

$$p = P_{in}p_{in} + P_{out}p_{out} = \frac{CIR}{BW}p_{in} + \frac{BW - CIR}{BW}p_{out} \quad (4.6)$$

Plugging the p above into 3.6 and solving for BW , it is possible to come with a new expression for bandwidth which is presented below.

$$BW = \frac{p_{out} - p_{in}}{2p_{out}} CIR + \sqrt{\left(\frac{p_{out} - p_{in}}{2p_{out}} CIR\right)^2 + \left(\frac{NoF \times C \times MSS}{\sqrt{p_{out}} RTT}\right)^2} \quad (4.7)$$

Note that if we let $p_{in} = 0$, then equation 4.7 reduces to equation 4.4.

4.3 Under-provisioned Case

A network is said to be under-provisioned when:

- (1) The cumulative sum of the CIRs of all of the aggregates using a link is greater than the bandwidth of the link.
- (2) All packet drops are in-profile packets.
- (3) There are no packets marked out-of-profile.

If no packets are marked out-of-profile (i.e. $BW < CIR$), the probability of marking a packet in-profile is now 1 and p is simply equal to the probability of dropping in-profile packets.

$$p = p_{in}$$

Plugging the expression for p above into equation 3.6 results in the following expression for bandwidth:

$$BW = \frac{NoF \times C \times MSS}{\sqrt{p_{in}} RTT} \quad (4.8)$$

4.4 Comparison with Other Models

There are two other studies which have looked at problem of developing an accurate performance model for TCP flows in an Assured Forwarding network. In

[16], the authors develop a performance model for networks which use a short-term TSW marker. For the over-provisioned case they come up with the following result:

$$B = \frac{k}{2RTT} \left(R + \sqrt{R^2 + \frac{6}{p_{out}}} \right) \quad \text{if } R \geq \sqrt{\frac{2}{p_{out}}} \quad (4.9)$$

$$B = \frac{3k}{4RTT} \left(R + \sqrt{\frac{2}{p_{out}}} \right) \quad \text{if } R < \sqrt{\frac{2}{p_{out}}} \quad (4.10)$$

where k is the packet size, and $R = \frac{CIR}{k} \times RTT$. We will disregard equation 4.10 because it applies to the case where the marker is marking less than CIR packets in-profile. This is really a defect of the short-term TSW marker. It is possible show that equation 4.9 is equivalent to equation 4.4 which is developed in Section 4.1 of this study.

In [14], the authors develop a performance model for networks which use a Token Bucket marker. For the over-provisioned case they come up with the following equation:

$$r = \frac{1}{2} \left(A + \sqrt{A^2 + \frac{6}{p_2 T^2}} \right) \quad \text{if } A \leq \frac{1}{T} \left(\sqrt{2(B + \frac{1}{p_2})} + 2\sqrt{2B} \right) \quad (4.11)$$

$$r = \frac{3A}{4} + \frac{3\sqrt{B + \frac{1}{p_2}}}{2\sqrt{2}T} \quad \text{if } A > \frac{1}{T} \left(\sqrt{2(B + \frac{1}{p_2})} + 2\sqrt{2B} \right) \quad (4.12)$$

where r is the rate in packets per second, A is the assured rate (or CIR) in packets per second, B is the token bucket size, and T is the average round trip time. We will disregard equation 4.12 because it applies to the case where the marker is marking less than CIR packets in-profile. This is really a defect of the marker. It is possible show that equation 4.9 is equivalent to equation 4.4 which is developed in Section 4.1 of this study. Therefore, the long-term TSW, the short-term TSW markers, and the Token Bucket marker can be described by the same equation as long as CIR packets are being marked in profile.

This chapter has presented three formulas to predict achieved bandwidth in the under-provisioned, near-provisioned and over-provisioned cases. In Chapter 5, simulation is used to validate the model presented in this chapter. The simulation work focuses on the over-provisioned case because any CIR guarantee will have to rely on an over-provisioned network.

Chapter 5

Simulation and Observations

In this section we use the model to make certain key observations on how the four factors (packet size, number of flows, RTT, and CIR) affect the steady-state throughput. Also, we further validate these observations, by showing them to be true in simulation.

The simulation study was performed using the ns-2 network simulator version 2.1b7, along with the Nortel Networks DiffServ code. This DiffServ code has since been integrated into the ns network simulator and is available in ns version 2.1b8.

The topology, presented in Figure 5.1, consists of two aggregates of TCP flows. One aggregate sends traffic from source-0 to sink-0 and the other aggregate of flows sends traffic from source-1 to sink-1. We used SACK TCP because it is able to remain in Congestion Avoidance at higher loss rates than other TCPs. The constant C for our particular setup was 1.31 as suggested in [12].

Unless noted otherwise, an aggregate of 15 TCP flows was connected between each source and sink. The individual TCP flows were long-lived FTP applications which transmitted MSS (in this simulation 1000 bytes) size packets. RIO-C [13] was used as the AQM scheme and the RED parameters were 10,300,0.05 and 300,400,0.02 for out-of-profile and in-profile packets respectively. A Time Sliding Window (TSW)

[13] policer was used to mark packets as either in or out-of-profile using a CIR of 5.0 Mbps per aggregate.

The simulation length for each simulation was set to 5000 seconds. This was deemed to be sufficient since it allowed Congestion Avoidance to experience 50 or more cycles in each of the simulations. The first 1000 seconds of the simulations were ignored in order to eliminate any transient effects. Results were then averaged over the last 4000 seconds of the simulation.

Figure 5.1: Simulation Topology

5.1 Methodology

In each of the simulations that follow, we used formula 4.4 to calculate the predicted bandwidth. The formula requires that the RTT , CIR , MSS , drop probability of out-of-profile packets, p_{out} , and number of flows, NoF , be measured. In the simulations the first three factors were fixed beforehand. In a real network the CIR

is controlled by network management, the number of flows and packet size can be approximated using average values.

The other two factors, p_{out} and RTT , were both calculated using simulation measurements. The drop probability was calculated by using measurements of out-of-profile packet drops and arrivals at the core node.

The average RTT was calculated using equation 5.1 below.

$$RTT = R_0 + \sum_{i=1}^n \frac{Q_{avg_i}}{L_i} \quad (5.1)$$

where Q_{avg_i} is the average queue size at link i and L_i is the link speed at link i . The average RTT is the sum of the average waiting times in queues along the path and R_0 , the propagation and transmission time on the rest of the round trip. In this particular topology the queue size is zero everywhere except at the first core node.

5.2 Observations

Observation 1 All else being equal, an aggregate with a smaller CIR will obtain more excess bandwidth than an aggregate with a larger profile.

This can be demonstrated by showing that $\frac{\partial BW_{ex}}{\partial CIR}$ is always less than zero. If this partial derivative is less than zero, then a smaller CIR should result in more excess bandwidth, all else being equal.

$$\frac{\partial BW_{ex}}{\partial CIR} = \frac{CIR}{4 * \sqrt{\left(\frac{CIR}{2}\right)^2 + \left(\frac{NoF * C * MSS}{\sqrt{p_{out} RTT}}\right)^2}} - \frac{1}{2} \leq 0 \quad (5.2)$$

The second term (which is positive) of the square root can be removed and the inequality will still hold.

$$\frac{CIR}{4 * \sqrt{\left(\frac{CIR}{2}\right)^2}} - \frac{1}{2} \leq 0 \quad (5.3)$$

This further reduces to:

$$0 \leq 0$$

Proving the inequality to indeed be true. This is not always a desirable result. From a business perspective, a customer may expect that buying a larger profile would result in a larger share of the excess bandwidth and not a smaller share. Also, the same conclusion can be derived from simulation graph presented in Figure 5.2. In the simulation, the CIR of aggregate 1 was increased with each simulation run and the bandwidth measured. The CIR of aggregate 0 was kept constant at 5.0 Mbps. It should be easy to see that aggregate 0 gets more of the excess bandwidth (difference between the bandwidth and the CIR) even though it has a smaller CIR. In the very last point on the graph, aggregate 0 obtains around 17 Mbps of bandwidth resulting in 12 Mbps of excess bandwidth while aggregate 1 obtains 32 Mbps of bandwidth, resulting in only 7 Mbps of excess bandwidth.

Observation 2 All else being equal, an aggregate with a larger RTT will obtain less excess bandwidth than an aggregate with a smaller RTT.

By careful examination of equation 4.5, this observation should become obvious. Also, the same conclusion can be derived from the simulation graph presented in Figure 5.3. In the simulation, the core-core link delay was reduced to 25ms. The source-edge and edge-sink delays were increased to 37.5 ms for aggregate 0. The RTT of aggregate 1 was increased with each simulation run through its source-edge

Figure 5.2: CIR versus Achieved Bandwidth

and sink-edge delays. As demonstrated below, our analytic model confirms earlier results shown in [15, 17].

Figure 5.3: RTT versus Achieved Bandwidth

Observation 3 All else being equal, an aggregate with a larger number of flows will get more excess bandwidth than an aggregate with fewer flows.

By careful examination of equation 4.5, this observation should become obvious. Also, the same conclusion can be derived from the simulation graph presented in Figure 5.4. In the simulation the number of flows in aggregate 1 was increased with simulation run. This same result has been shown in [15].

Observation 4 All else being equal, an aggregate with a larger MSS (maximum segment size) will get more excess bandwidth than an aggregate with a smaller MSS.

By careful examination of equation 4.5, this observation should become obvious. Also, the same conclusion can be derived from the simulation graph presented in

Figure 5.4: Number of Flows versus Achieved Bandwidth

Figure 5.5. In the simulation the packet size of the aggregate 1 flows was increased with each simulation run. This result has been shown via experimentation in [15].

Figure 5.5: Packet Size versus Achieved Bandwidth

Chapter 6

Multiple Congested Nodes

In this section we investigate the effect of multiple congested nodes on achieved bandwidth.

Letting $p_{out,i}$ represent the probability of dropping out-of-profile packets at a particular node i and p_{out} represent the total out-of-profile packet drop probability over the whole path, it is possible to develop a formula for p_{out} in terms of the individual $p_{out,i}$.

For the case of a flow which travels through n congested nodes, the probability of not dropping an out-of-profile p_{out} , is related to the $p_{out,i}$ via the following formula:

$$p_{out} = 1 - \prod_{i=1}^n (1 - p_{out,i}) \quad (6.1)$$

We can now use equation 6.1 and equation 4.4 to present a fifth key observation.

Observation 5 All else being equal, an aggregate going through more congested nodes will obtain less excess bandwidth than an aggregate going through fewer congested nodes.

From equation 4.4, it is possible to see that a larger value for p_{out} , results in less excess bandwidth. Then from equation 6.1, one can see that each congested node

adds one term to the product thereby increasing p_{out} .

Equation 6.1 is validated in the simulation section that follows.

6.1 Simulation

The topology used in this section is similar to that used in the Section 5. A similar topology was used in [12] and [5] to observe the effects of multiple congested nodes. Because of the increase in the size of the simulation topology, many parameters from Section 5 were reduced in value to allow the simulation to complete in a reasonable amount of time.

An aggregate of ten TCP flows was connected between each source i and sink i . The individual TCP flows were long-lived FTP applications which transmitted MSS (in this simulation 1000 bytes) size packets. The RED parameters were $\{10, 80, 0.1\}$ and $\{80, 150, 0.02\}$ for out-of-profile and in-profile packets respectively.

The CIR for the long (the aggregate of flows going from sink-0 to source-0) and short aggregates (the aggregates of flows going from source-1 to sink-1, ..., source- n to sink- n) was kept the same, and four different values were tried: 2.5 Mbps, 5.0 Mbps, 7.5 Mbps, and 10.0 Mbps. Each CIR value corresponds to a different level of provisioning ranging from 20% to 80% of the bottleneck bandwidths.

In each successive simulation, we varied the number of congested links. The drop probability and queue size were measured for each congested node. These two values were then used to calculate the RTT and the end-to-end out-of-profile drop rate for short and long aggregates. Also, the steady-state throughput obtained by the long aggregate of TCP flows was recorded.

Figure 6.2 shows graphs of the measured and predicted throughput for the long

Figure 6.1: Multiple Congested Nodes

Figure 6.2: Bandwidth versus Number of nodes for TCP SACK

source (source-0 to sink-0) TCP aggregates. The solid-line curves are the measured bandwidth and the X-line curves are the predicted bandwidth. Each curve graphs the bandwidth versus the number of congested nodes along the path. Each pair of predicted and measured curves approaches its CIR as the number of congested links increases.

From the figure, it is easy to see that the bandwidth decreases with the number of congested nodes and eventually stabilizes at the CIR. This can be explained through equation (5) presented in Section 4. As the number of congested nodes increases so do pout and the RTT, decreasing the value of the second term of the square root. Eventually, this term is negligible compared to the left-hand term $(\frac{CIR}{2})^2$. The bandwidth formula then reduces to $BW = CIR$.

At higher provisioning levels, the measured bandwidth seems to diverge from target rate when the number of congested nodes is high. We believe this is due to the fact that the average queue size is very large when the provisioning level is high. A high average queue size results in a higher probability of dropping packets and hence, a higher probability and consecutive packet drops resulting in timeouts. Also, at higher provisioning levels, in-profile packet drops can no longer be ignored.

Another observation, which can be made from the graphs, is that as the rate of provisioning goes up, the aggregate obtains less excess bandwidth. This is because there is less excess bandwidth to be distributed among aggregates.

Table 6.1 shows the percentage error for selected points from the graphs. The average error seems to increase as the level of provisioning increases. This may be due to the fact that as the provisioning level increases so do the queue sizes of the nodes along the path and hence the drop probability. The higher the drop probability, the

# of nodes	CIR in Mbps			
	2.5	5.0	7.5	10.0
1	4.10%	5.57%	1.75%	0.43%
2	0.86%	2.63%	2.59%	1.07%
5	0.47%	0.27%	3.95%	6.78%
10	2.77%	5.78%	5.17%	7.96%
AVG	1.85%	3.57%	3.99%	5.26%

Table 6.1: Percentage Error for TCP SACK Predictions

more timeouts become a factor and the less accurate the model is. For most cases in Table 6.1, the percentage error was less than 5%. The low percentage errors reported verify that the model is a good one. Simulations were also run using TCP Reno. For these simulations, the percentage error was larger with most cases less than 10%.

Chapter 7

Three Drop-precedence Levels

In this section, the model is extended from two drop precedence levels to three drop precedence levels. There have been various policers proposed to mark packets for treatment based on 3 levels of drop precedence. In the policer proposed in [3] packets are marked green, yellow, or red with red having the highest drop precedence. In this policer there is a notion of a peak information rate or PIR. When an aggregate of flows exceeds its CIR, its non-conforming (beyond CIR) packets are marked yellow. If in addition, an aggregate is exceeding its PIR, packets in excess of the PIR are marked red.

7.1 Case 1: PIR Over-provisioned

- (1) The network is PIR over-provisioned when the cumulative sum of the PIRs of all of the aggregates using a link is less than the bandwidth of the link.
- (2) All packet drops are red packets. There are no green or yellow packet drops.

We begin by looking at the probabilities of marking a packet red, green, or yellow when the bandwidth is greater than the Peak Information Rate (PIR).

$$P_{green} = \frac{CIR}{BW}, P_{yellow} = \frac{PIR-CIR}{BW}, P_{red} = \frac{BW-PIR}{BW}$$

Next, we develop a relation for the p from Equation 3.6 in terms of the drop probabilities: p_{green} , p_{yellow} and p_{red} .

$$p = P_{red} \times p_{red} \text{ since } p_{green} = p_{yellow} = 0$$

$$p = \frac{BW - PIR}{BW} p_{red}$$

Following the same procedure as in Section 4, we get the following relation for bandwidth:

$$BW = \frac{PIR}{2} + \sqrt{\left(\frac{PIR}{2}\right)^2 + \left(\frac{NoF \times C \times MSS}{\sqrt{p_{red}} RTT}\right)^2} \quad (7.1)$$

7.2 Case 2: CIR Over-provisioned

- (1) The network is CIR over-provisioned when the cumulative sum of the CIRs of all of the aggregates using a link is less than the bandwidth of the link.
- (2) All packet drops are yellow packets. There are no green packet drops.
- (3) There are no packets marked red.

Now we look at the case when the bandwidth is less than the peak-rate but greater than the CIR. For this case, the probability of a packet being marked a particular color is:

$$P_{green} = \frac{CIR}{BW}, P_{yellow} = \frac{BW - CIR}{BW}, P_{red} = 0$$

Developing a relation for p in terms of p_{green} and p_{yellow} , we get:

$$p = BW - CIRBW p_{yellow} \text{ since } p_{green} = 0$$

Plugging the expression above into the p in Equation 3.6 and solving as we have been doing, results in the following formula for bandwidth:

$$BW = \frac{CIR}{2} + \sqrt{\left(\frac{CIR}{2}\right)^2 + \left(\frac{NoF * C * MSS}{\sqrt{p_{yellow}}RTT}\right)^2} \quad (7.2)$$

7.3 Case 3: Under-provisioned

- (1) The network is under-provisioned when the cumulative sum of the CIRs of all of the aggregates using a link is greater than the bandwidth of the link.
- (2) All packet drops are green packets.
- (3) There are no packets marked yellow or red.

Lastly, we look at the case where the bandwidth is less than the CIR. For this case, all packets are marked green.

$$P_{green} = 1, P_{yellow} = 0, P_{red} = 0$$

$$p = p_{green}$$

We can simply replace the p in equation 3.6 with p_{green} and get the following equation for bandwidth:

$$BW = \frac{NoF * C * MSS}{\sqrt{p_{green}}RTT} \quad (7.3)$$

7.4 Simulation

The topology in Figure 5.1 was used for experiments in this section. Both aggregates had 15 micro-flows and a packet size of 1000 bytes. The CIR was the same for both aggregates in each simulation and the PIR was set to 1.5 times the CIR. In each successive simulation the CIR was increased and the average bandwidth of both flows measured. Also, the average and packet drop probability were measured

in order to calculate the estimate for the bandwidth. Depending on the provisioning level for the simulation, equation 10, 11, or 12 was used for the calculation. Below is a graph of the results. The calculated values are denoted by the dashed lines.

Figure 7.1: Three Drop Simulation

The first estimate for region 2 (CIR Over-provisioned) and the first estimate for region 3 (Under-provisioned) are considerably inaccurate. This may be due to not all packet drops being of the same colour as per the assumptions. As shown in the graph below, not all packet drops are of one color near the edge of a region. This is especially true for the two points (18 Mbps and 27 Mbps) which are the most inaccurate. By the not including the drops of the other color in the calculation we get an over-estimate of the bandwidth.

Figure 7.2: Drop Distribution

Chapter 8

Conclusions and Recommendations for Further Study

8.1 Conclusions

Recent simulation and experimental studies [15] [16] have shown interestingly unexpected results in terms of the achieved bandwidth of TCP flows in an AF-based Differentiated Services network. One of the studies [15] shows that the bandwidth of TCP flows in such network is heavily influenced by five main factors: RTT, packet size, CIR, number of flows, and drop probability. In this thesis, an analytical model is developed which predicts the bandwidth of TCP aggregates in such networks was developed. The model calculates predicted bandwidth as a function of the five factors listed above.

There has been earlier research [12] [13] in developing analytical models to predict bandwidth of TCP flows but all of that work was in Best Effort networks. The model presented here extends that work and looks at the case of an AF-based Differentiated Services network. The model is beneficial not only in showing quantitatively how the factors affect bandwidth but also in expanding current understanding of how the five factors affect AF achieved bandwidth.

In a DiffServ network, the drop probability that is used by a node to drop

packets is determined by the long term congestive queue size and the colour of packets arriving at the node. To account for this, the model characterizes the network to operate in one of three operating modes: under-provisioned, near-provisioned and over-provisioned. Over-provisioned is deemed to be the most common since it is the only of the three modes in which a provider can make any sort of bandwidth guarantee. By definition, there is not enough bandwidth available in the network to satisfy the CIR of all flows in the under-provisioned and near-provisioned cases.

The model is beneficial in expanding current understanding of how the five factors (RTT, packet size, CIR, number of flows in an aggregate, and number of congested nodes along the path) affect AF achieved bandwidth. Based on the model, four key observations are made:

- (1) All else being equal, an aggregate with a smaller CIR will obtain more excess bandwidth than an aggregate with a larger profile.
- (2) All else being equal, an aggregate with a larger RTT will obtain less excess bandwidth than an aggregate with a smaller RTT.
- (3) All else being equal, an aggregate with a larger number of flows will get more excess bandwidth than an aggregate with fewer flows.
- (4) All else being equal, an aggregate with a larger MSS (maximum segment size) will get more excess bandwidth than an aggregate with a smaller MSS.

Simulation is presented to confirm these observations. Also, the simulation helps to illustrate the extent to which bandwidth is effected.

There is also new work to understand the effect of multiple congested nodes on bandwidth assurance. There has been earlier work [12] [5] which has looked at

the effects of multiple congested nodes on bandwidth in a Best Effort network. The analysis presented here looks at the case of an AF-based DiffServ network. However, there is no reason that the analysis could not also be applied to a Best Effort network. This is a possible activity for future work. Simulation is presented in order to validate the model for multiple congested nodes. The majority of the results obtained through simulation are within 5% of the value predicted by the model.

Initially, the the analysis is focused on networks with two levels of drop precedence. However, the DiffServ Working Group has proposed four classes of three levels of drop precedence each. So an extension to the model is presented which looks at the three drop precedence level case.

The TCP models developed in this thesis do have a number of limitations as far as applicability is concerned. First, the model is not applicable for aggregates consisting of short TCP flows (mice). Second, it assumes that the long-lived flows are operating consistently in the congestion avoidance region - i.e. no timeouts.

In conclusion, the main contribution of this thesis is the development of an analytical model to predict the achieved bandwidth of TCP aggregates in a Differentiated Services Network. The model is developed for the case of networks that support both two levels of drop precedence and for the case of networks that support three levels of drop precedence. Further, the model covers the case of (i) over-provisioned, (ii) under-provisioned and (iii) near-provisioned networks. Another contribution of this thesis is to extend the model for a network with multiple congested nodes. Simulation results validate the bandwidth predictions made using the model.

8.2 Recommendations for Further Study

As pointed out in the conclusion, the model has some limitations because certain factors such as timeout are not accounted for. If the drop provability is low, timeouts should be rare and the simulation seems to suggest that this is the case. In the various scenarios that were looked at in simulation the results were close to the theoretical expected result. However, there is previous research that has looked at the effects of timeouts in a Best Effort network and it may not always be safe to assume a low drop probability (e.g. wireless networks). In a future study, the model presented here could be extended to account for timeouts.

The majority of the simulation presented is for the case of an over-provisioned network. This case is assumed to be common place. Further simulation is required to validate the developed models for under-provisioned networks, and near-provisioned networks.

An analytical model is first developed for a two drop precedence network and then a three drop precedence network because this is the number dictated in the Assured Forwarding RFC. However, the DiffServ Working Group chose the number three arbitrarily. An interesting project for future work might be to extend model to n levels of drop precedence. It would be interesting to see how the number of drop precedence levels affects bandwidth assurance.

New work is presented to look at the case of multiple congested nodes. This work could also be applicable to Best Effort networks. A future study could apply the model presented here to a Best Effort scenario. In addition, a comparison could be done of this model to the other models for multiple congested nodes.

There is also potential for this work to be applicable in industry. The model

could be integrated as part of a network planning/engineering tool. The ability to predict the achieved bandwidth of TCP aggregates should be of definite used to network engineers when they layout and provision their networks.

Bibliography

- [1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, June 1998.
- [2] David Clark and Wenjia Fang. Explicit allocation of best effort packet delivery service. IEEE/ACM Transactions on Networking, 6(4):362–373, August 1998.
- [3] Wenjia Fang, Nabil Seddigh, and Biswajit Nandy. A time sliding window three color marker. RFC 2859, June 2000.
- [4] Victor Firoiu and Marty Borden. A study of active queue management for congestion control. In Proceedings of INFOCOM, pages 1435–1444. IEEE, March 2000.
- [5] Sally Floyd. Connections with multiple congested gateways in packet-switched networks, part 1: One-way traffic. Computer Communications Review, 21(5):30–47, October 1991.
- [6] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, 1(4):397–413, August 1993.
- [7] J. Heinanen, F. Baker, W. Weiss, and W. Wroclawski. An assured forwarding PHB. RFC 2597, June 1999.
- [8] J. Heinanen and R. Guerin. A single rate three color marker. RFC 2697, September 1999.
- [9] J. Heinanen and R. Guerin. A two rate three color marker. RFC 2698, September 1999.
- [10] V. Jacobson, K. Nichols, and K. Poduri. An expedited forwarding PHB. RFC 2598, June 1999.

- [11] R. Makkar, I. Lambadaris, J. H. Salim, N. Seddigh, B. Nandy, and J. Babi-arz. Empirical study of buffer management schemes for diffserv assured forwarding PHB. In Proceedings of 9th International Conference on Computer Communications and Networks, Las Vegas, October 2000.
- [12] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. Computer Communications Review, 27(3):67–82, July 1997.
- [13] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In Proceedings of SIGCOMM, Vancouver, Canada, September 1998. ACM.
- [14] Sambit Sahu, Philippe Nain, Christophe Diot, Victor Firoiu, and Donald Towsley. On achievable service differentiation with token bucket marking for TCP. In Proceedings of SIGMETRICS, pages 23–33, Santa Clara, CA, June 2000. ACM.
- [15] Nabil Seddigh, Biswajit Nandy, and Peter Pieda. Bandwidth assurance issues for TCP flows in a differentiated services network. In Proceedings of GlobeCom, Rio De Janeiro, December 1999.
- [16] I. Yeom and A. Reddy. Modeling tcp behavior in a differentiated services network. IEEE/ACM Transactions on Networking, February 2001.
- [17] Ikjun Yeom and A. L. Narasimha Reddy. Realizing throughput guarantees in a differentiated services network. In Proceedings of International Conference on Multimedia Computing and Systems, volume 2, pages 372–376. IEEE, 1999.