BANDWIDTH ASSURANCE IN A DIFFERENTIATED SERVICES NETWORK

A Dissertation
by
IK-JUN YEOM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2001

Major Subject: Computer Engineering
BANDWIDTH ASSURANCE IN A DIFFERENTIATED SERVICES NETWORK

A Dissertation

by

IK-JUN YEOM

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

A. L. Narasimha Reddy
(Chair of Committee)

Pierce E. Cantrell
(Member)

Riccardo Bettati
(Member)

Nancy Amato
(Member)

Chanan Singh
(Head of Department)

May 2001

Major Subject: Computer Engineering
ABSTRACT

Bandwidth Assurance in a Differentiated Services Network. (May 2001)

Ik-Jun Yeom, B.S., Yon-Sei University;
M.S., Texas A&M University
Chair of Advisory Committee: Dr. A. L. Narasimha Reddy

The differentiated services (DS) architecture has been proposed for providing different levels of services and has recently received wide attention. A packet in a DS domain is classified into a class of service according to its contract profile and treated differently by its class. There are currently two types of service standardized, Expedited Forwarding (EF) and Assured Forwarding (AF). In this dissertation, we focus on AF service.

In AF service, customers make contract in terms of bandwidth with their service providers. At ingress nodes, the service provider marks incoming packets IN or OUT based on the current sending rate and the contracted rate. At core routers, each packet observes different drop precedences according to its marking. With these simple marking and forwarding schemes, the service provider can provide different amounts of bandwidth to meet the customers’ requests.

However, it has been observed that the drop precedences by themselves cannot achieve the desired rates because of the strong interaction of the transport protocol with packet drops in the network. In this dissertation, we propose and evaluate a number of techniques to better achieve the throughput guarantees in such networks. The proposed techniques consider modifying the transport protocol at the sender, the marking strategies at the marker and the dropping policies at the router. It is shown that these techniques improve the likelihood of achieving the desired throughput
guarantees and also improve the service differentiation.

We also develop analytical models of TCP behavior in a DS network. The models quantitatively characterize TCP throughput as functions of the contract rate, the packet-drop rate and the round-trip time. The models analytically explain the previous observations such as unfair sharing of excess bandwidth among TCP flows with different contract rates.

We look at the problem of achieving specific QoS goals of individual flows by flexibly managing resources available to an aggregated source. It will be shown that an aggregated source can maintain state of individual flows at the edge of the network and utilize this state effectively in adaptively marking packets of individual flows to meet their QoS goals.
To My Parents: Donhee Yeom and Hyunsik Yoon
ACKNOWLEDGMENTS

First of all, I must express my gratitude towards my advisor, Dr. A. L. Narasimha Reddy for his advice, direction, attention and encouragement over the past five years. I would not have been able to complete this work without his support. I would like to also thank Dr. Pierce Cantrell, Dr. Riccardo Bettati, Dr. Nancy Amato and Dr. Marina Vannucci for the valuable comments, time and help to complete my work. I also thank my master’s committee members, Dr. Nitin Vaidya and Dr. Gwan Choi.

I extend a thanks to the student members of the Computer Engineering Group here at Texas A&M University. Especially, I would like to thank Inkoo Kim, Ravi Wijayaratne, Xiaonan Ma and Aman Garg for their help, comments and discussion. I also thank the many Korean Students, Jonghyune Kim, Hoki Kim, Sangig Rho, Youngho Sohn, Soonil Ho and Dr. Soonchul Baeg.

I worked at Nortel Networks, Billerica, MA as a summer internship from April to July 2000. I really enjoyed working there, and the work gave me inspiration and valuable experience. Especially, I would like to thank Dr. Victor Firoiu, Manager - The QoS group, Nortel Networks.

Finally, I thank my parents for instilling in me confidence and a drive for pursuing my Ph. D and dedicate this dissertation to them.

This work has been supported in part by Texas ATP, National Science Foundation (NSF) and EMC corp.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>6</td>
</tr>
<tr>
<td>REALIZING THROUGHPUT GUARANTEES IN A DIFFERENTIATED SERVICES NETWORK</td>
<td></td>
</tr>
<tr>
<td>A. Background</td>
<td>7</td>
</tr>
<tr>
<td>B. Policies to achieve target rates</td>
<td>11</td>
</tr>
<tr>
<td>1. Limiting OUT packets</td>
<td>11</td>
</tr>
<tr>
<td>2. Inverse-rate drop policy</td>
<td>12</td>
</tr>
<tr>
<td>3. Three drop precedences</td>
<td>16</td>
</tr>
<tr>
<td>4. Two-windows TCP</td>
<td>20</td>
</tr>
<tr>
<td>C. Comparison of all the schemes</td>
<td>23</td>
</tr>
<tr>
<td>1. Impact of a non-responsive source</td>
<td>24</td>
</tr>
<tr>
<td>2. Impact of different RTTs</td>
<td>25</td>
</tr>
<tr>
<td>D. Effect of aggregated sources</td>
<td>29</td>
</tr>
<tr>
<td>E. Summary</td>
<td>32</td>
</tr>
<tr>
<td>III</td>
<td>34</td>
</tr>
<tr>
<td>MODELING TCP BEHAVIOR IN A DIFFERENTIATED SERVICES NETWORK</td>
<td></td>
</tr>
<tr>
<td>A. TCP modeling in a DS network</td>
<td>36</td>
</tr>
<tr>
<td>1. Packet marking with TSW rate estimator</td>
<td>36</td>
</tr>
<tr>
<td>2. Modeling for two-drop precedence</td>
<td>40</td>
</tr>
<tr>
<td>a. A flow through an undersubscribed path</td>
<td>40</td>
</tr>
<tr>
<td>b. A flow through an oversubscribed path</td>
<td>46</td>
</tr>
<tr>
<td>c. Combined model</td>
<td>47</td>
</tr>
<tr>
<td>d. Simulations</td>
<td>49</td>
</tr>
<tr>
<td>e. Throughput analysis</td>
<td>59</td>
</tr>
<tr>
<td>3. Modeling for three-drop precedence</td>
<td>62</td>
</tr>
<tr>
<td>a. Simulations</td>
<td>65</td>
</tr>
<tr>
<td>B. A throughput model for aggregated flows</td>
<td>66</td>
</tr>
<tr>
<td>1. Packet marking of aggregated flows</td>
<td>66</td>
</tr>
<tr>
<td>2. Modeling for throughput of aggregated flows</td>
<td>69</td>
</tr>
<tr>
<td>3. Simulations</td>
<td>71</td>
</tr>
<tr>
<td>C. Discussion and related work</td>
<td>78</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>D. Summary</td>
<td>82</td>
</tr>
<tr>
<td><strong>IV</strong> DEALING WITH AGGREGATED FLOWS</td>
<td>83</td>
</tr>
<tr>
<td>A. Aggregate marking for aggressive bandwidth management</td>
<td>84</td>
</tr>
<tr>
<td>B. Simulations</td>
<td>86</td>
</tr>
<tr>
<td>1. Dealing with different RTTs</td>
<td>86</td>
</tr>
<tr>
<td>2. Aggressive bandwidth management in congested networks</td>
<td>88</td>
</tr>
<tr>
<td>3. Scalability</td>
<td>91</td>
</tr>
<tr>
<td>4. Impact on network</td>
<td>93</td>
</tr>
<tr>
<td>5. Impact on other flows at transient time</td>
<td>96</td>
</tr>
<tr>
<td>C. Discussion and related work</td>
<td>96</td>
</tr>
<tr>
<td>D. Summary</td>
<td>99</td>
</tr>
<tr>
<td><strong>V</strong> ADAPTIVE MARKING FOR AGGREGATED FLOWS</td>
<td>100</td>
</tr>
<tr>
<td>A. Achieving target rates</td>
<td>105</td>
</tr>
<tr>
<td>B. Performance evaluation through simulations</td>
<td>111</td>
</tr>
<tr>
<td>1. Impact of different Δ</td>
<td>111</td>
</tr>
<tr>
<td>2. Impact of different observation periods</td>
<td>113</td>
</tr>
<tr>
<td>3. Impact of window size for rate estimator</td>
<td>115</td>
</tr>
<tr>
<td>4. Dealing with network dynamics</td>
<td>121</td>
</tr>
<tr>
<td>5. Dealing with different RTTs</td>
<td>122</td>
</tr>
<tr>
<td>6. Dealing with unresponsive flows</td>
<td>122</td>
</tr>
<tr>
<td>C. Receiver-side marking strategies</td>
<td>124</td>
</tr>
<tr>
<td>1. Performance evaluation through simulation</td>
<td>129</td>
</tr>
<tr>
<td>D. Summary</td>
<td>133</td>
</tr>
<tr>
<td><strong>VI</strong> CONCLUSIONS</td>
<td>136</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>139</td>
</tr>
<tr>
<td>VITA</td>
<td>145</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Comparison of all the schemes</td>
<td>23</td>
</tr>
<tr>
<td>II</td>
<td>Throughputs of different schemes at different RTTs</td>
<td>28</td>
</tr>
<tr>
<td>III</td>
<td>Performance summary of different schemes at different RTTs</td>
<td>29</td>
</tr>
<tr>
<td>IV</td>
<td>Loss rates of IN/OUT packet</td>
<td>51</td>
</tr>
<tr>
<td>V</td>
<td>Loss rates of \textit{green/yellow/red} packet</td>
<td>66</td>
</tr>
<tr>
<td>VI</td>
<td>Quantitative comparisons</td>
<td>88</td>
</tr>
<tr>
<td>VII</td>
<td>Simulation results with adaptive marking</td>
<td>109</td>
</tr>
<tr>
<td>VIII</td>
<td>Expected bandwidth achieved by individual flows</td>
<td>111</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Network elements in a DS domain</td>
</tr>
<tr>
<td>2</td>
<td>RIO drop policy</td>
</tr>
<tr>
<td>3</td>
<td>Network topology for simulations</td>
</tr>
<tr>
<td>4</td>
<td>Realized rates at different levels of subscription</td>
</tr>
<tr>
<td>5</td>
<td>No-OUT scheme at different subscription levels</td>
</tr>
<tr>
<td>6</td>
<td>Inverse-rate drop scheme at different subscription levels</td>
</tr>
<tr>
<td>7</td>
<td>Three drop precedences at different subscription levels</td>
</tr>
<tr>
<td>8</td>
<td>Modified congestion avoidance algorithm</td>
</tr>
<tr>
<td>9</td>
<td>Two-windows TCP at different subscription levels</td>
</tr>
<tr>
<td>10</td>
<td>Impact of a non-responsive UDP flow</td>
</tr>
<tr>
<td>11</td>
<td>Impact of a non-responsive flow with reservation</td>
</tr>
<tr>
<td>12</td>
<td>Comparisons between aggregated sources and single source</td>
</tr>
<tr>
<td>13</td>
<td>Measured rates vs. time of single and aggregated sources with 2Mbps subscription</td>
</tr>
<tr>
<td>14</td>
<td>TSW algorithm</td>
</tr>
<tr>
<td>15</td>
<td>Microscopic view of packet marking</td>
</tr>
<tr>
<td>16</td>
<td>Packet sent in each period</td>
</tr>
<tr>
<td>17</td>
<td>Reservation and congestion windows</td>
</tr>
<tr>
<td>18</td>
<td>A simple network topology</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>19</td>
<td>Throughputs of TCP-Reno flows in 40/30 Mbps links</td>
</tr>
<tr>
<td>20</td>
<td>Throughputs of TCP-Reno flows 20/15 Mbps links</td>
</tr>
<tr>
<td>21</td>
<td>Relative error between simulation and estimation</td>
</tr>
<tr>
<td>22</td>
<td>Simulation with cross traffic</td>
</tr>
<tr>
<td>23</td>
<td>Simulation with merged network topology</td>
</tr>
<tr>
<td>24</td>
<td>Simulation with split network topology</td>
</tr>
<tr>
<td>25</td>
<td>Achieved excess bandwidth</td>
</tr>
<tr>
<td>26</td>
<td>Observations from the model</td>
</tr>
<tr>
<td>27</td>
<td>Drop probabilities of three-drop precedence</td>
</tr>
<tr>
<td>28</td>
<td>Throughputs of TCP-Reno flows with different reservation rate in three-drop precedence network</td>
</tr>
<tr>
<td>29</td>
<td>Packet marking of aggregated flows</td>
</tr>
<tr>
<td>30</td>
<td>Aggregated network topology with different RTTs</td>
</tr>
<tr>
<td>31</td>
<td>The probability of a packet marked IN</td>
</tr>
<tr>
<td>32</td>
<td>Simulations with aggregated flows with different RTTs</td>
</tr>
<tr>
<td>33</td>
<td>Relative error</td>
</tr>
<tr>
<td>34</td>
<td>Aggregated network topology with different numbers of individual flows</td>
</tr>
<tr>
<td>35</td>
<td>Simulations with aggregated flows within different numbers of individual flows</td>
</tr>
<tr>
<td>36</td>
<td>Aggregated network topology with multiple aggregations</td>
</tr>
<tr>
<td>37</td>
<td>Dealing with multiple aggregations</td>
</tr>
<tr>
<td>38</td>
<td>Simulation with correlated packet losses</td>
</tr>
<tr>
<td>39</td>
<td>Dealing with different RTTs within aggregated sources</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>40</td>
<td>Dealing with different target rates within an aggregation</td>
</tr>
<tr>
<td>41</td>
<td>Network topology with different congestion levels</td>
</tr>
<tr>
<td>42</td>
<td>Bandwidth comparison with different marking schemes</td>
</tr>
<tr>
<td>43</td>
<td>Impact of marking algorithms on throughput</td>
</tr>
<tr>
<td>44</td>
<td>Queue lengths of different marking algorithms</td>
</tr>
<tr>
<td>45</td>
<td>Congestion windows of different marking algorithms</td>
</tr>
<tr>
<td>46</td>
<td>An algorithm for adaptive marking</td>
</tr>
<tr>
<td>47</td>
<td>Marking rate vs. achieved rate</td>
</tr>
<tr>
<td>48</td>
<td>Multi-hop topology</td>
</tr>
<tr>
<td>49</td>
<td>Achieved rate with static contract rates</td>
</tr>
<tr>
<td>50</td>
<td>Achieved rates with the adaptive marking</td>
</tr>
<tr>
<td>51</td>
<td>Network topology for adaptive aggregated marking</td>
</tr>
<tr>
<td>52</td>
<td>Realized throughput with the adaptive marking</td>
</tr>
<tr>
<td>53</td>
<td>Simulation topology</td>
</tr>
<tr>
<td>54</td>
<td>Marking rate with different $\Delta$</td>
</tr>
<tr>
<td>55</td>
<td>Throughput with different $\Delta$</td>
</tr>
<tr>
<td>56</td>
<td>Marking rate with different observation period and constant $\Delta$</td>
</tr>
<tr>
<td>57</td>
<td>Throughput with different observation period and constant $\Delta$</td>
</tr>
<tr>
<td>58</td>
<td>Marking rate and throughput with different observation period and different $\Delta$</td>
</tr>
<tr>
<td>59</td>
<td>Marking rate and throughput of Flow 0 with different window size and O.P. (1)</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>60</td>
<td>Marking rate and throughput of Flow 0 with different window size and O.P. (2) ........................................... 120</td>
</tr>
<tr>
<td>61</td>
<td>Average throughput with different window size and O.P. ................................. 121</td>
</tr>
<tr>
<td>62</td>
<td>Marking rate and throughput in network changed over time ............................ 123</td>
</tr>
<tr>
<td>63</td>
<td>Throughput of flows with different RTT ......................................................... 124</td>
</tr>
<tr>
<td>64</td>
<td>Throughput comparison with adaptive and proportional marking with unresponsive flows (Flow 0-6) ......................................................... 125</td>
</tr>
<tr>
<td>65</td>
<td>Throughput comparison with adaptive and proportional marking with unresponsive flows (Flow 7-9) ......................................................... 126</td>
</tr>
<tr>
<td>66</td>
<td>Reserved bandwidth transfer by receivers in a DS network .......................... 128</td>
</tr>
<tr>
<td>67</td>
<td>A simple algorithm for receiver-based bandwidth contract ............................ 129</td>
</tr>
<tr>
<td>68</td>
<td>Marking rate and throughput when the bandwidth is sufficient ........................ 131</td>
</tr>
<tr>
<td>69</td>
<td>Marking rate and throughput when the bandwidth is not sufficient ..................... 132</td>
</tr>
<tr>
<td>70</td>
<td>Marking rate and throughput when the bandwidth is plentiful .......................... 134</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Quality of Service (QoS) guarantees over networks have received wide attention recently with the increased use and need for mechanisms for delivering audio and video across networks. Different multimedia applications require different guarantees for delivery. Video-on-demand applications can tolerate large delays and primarily require throughput guarantees. Video-conferencing and IP telephony require delay guarantees as well as throughput guarantees. Much work has been done in designing multimedia applications that can tolerate delay and throughput variations by adjusting the quality of the transmitted stream. Simultaneously research is being carried out in providing service guarantees in networks.

Many scheduling approaches have been studied for providing QoS guarantees. These include variants of fair scheduling [1, 2, 3, 4] and priority scheduling [5, 6]. Recently, there has been a push to minimize the amount of work that needs to be done in a router to provide guarantees. The Differentiated Services (DS or diff-serv) framework is a proposal to provide service differentiation over networks with scalable manners [7, 8, 9, 10, 11]. Fig. 1 shows the network elements of the DS network.

In the DS domain, customers may contract for a certain level of service with the service provider. This contract can be made for individual or aggregated flows. The routers at the edge of the network, called boundary routers or ingress routers, may monitor, measure, shape, classify and mark packets of flows (individual or aggregated) according to the subscribed service. The core routers forward packets differently to provide the subscribed service. The core routers only need to provide several

The journal model is IEEE/ACM Transactions on Networking.
forwarding schemes to provide service differentiation, and thus we can deploy the DS to large networks.

Currently, two types of packet forwarding schemes (called Per-Hop Behavior or PHB) have been standardized by IETF (Internet Engineering Task Force) [9, 10]. Expedited Forwarding (EF) PHB was proposed in [9] as a premium service for the DS network. The EF PHB can be used to guarantee low loss rate, low latency, low jitter and assured throughput through end-to-end path like a “virtual leased line.” The departure rate of the EF traffic should equal or exceed a configurable rate independent of the intensity of any other traffic. The departure rate is measured over any time interval equal to or longer than a packet time. Several types of queue scheduling schemes (e.g., a priority queue, a single queue with a weighted round robin scheduler, and class-based queue [12]) may be used to implement the EF PHB.

Assured Forwarding (AF) PHB was proposed in [11, 10]. In AF PHB, the edge devices of the network monitor and mark incoming packets of either individual or
aggregated flows. A packet of a flow is marked IN (in profile) if the temporal sending rate at the arrival time of the packet is within the contract profile of the flow. Otherwise, the packet is marked OUT (out-of-profile). The temporal sending rate of a flow is measured using TSM (Time Sliding Window) or a token bucket controller.

The core routers in the network provide RIO (RED with IN/OUT) drop policy [11]. RIO drop policy is illustrated in Fig. 2. Each router maintains a virtual queue for IN packets and a physical queue for both IN and OUT packets. When the network is congested and the queue length exceeds minTh_OUT, the routers begin dropping OUT packets first. If the congestion persists even after dropping all incoming OUT packets and the queue length exceeds minTh_IN, IN packets are discarded. With this dropping policy, the RIO network gives preference to IN packets and provides different levels of service to users based on their service contracts. This research work focus on the AF PHB in the DS network.

To provide better control in achieving QoS goals in the AF PHB, three-drop precedence policy [13, 14] was proposed as an extension of two-drop precedence of AF PHB. In a three-drop precedence network, the edge devices mark a packet as one of green, yellow or red depending on the sending rate and the reservation rate for each color. Generally, it is recommended that the reservation rate for yellow is set to be equal to or greater than the reservation rate for green. If the current sending rate is
less than the reservation rate for green, the packet is marked as green. If the sending rate is greater than the reservation for green but less than the reservation for yellow, the packet is marked as yellow. Otherwise, the packet is marked as red. The core routers provide differentiation by dropping red packets first, yellow packets second and green packets.

With appropriate network provisioning, it is expected that this could result in bandwidth guarantees as long as the IN traffic doesn’t exceed the link capacities in the network. Without other reservation mechanisms or route pinning mechanisms, the edge routers can’t anticipate how all the flows get routed through the network. Hence, it is possible that occasionally the IN traffic may exceed the capacity on a particular link resulting in loss of guarantees.

This dissertation consists of two parts: In the first part, we study individual flows in the DS network. Recently, a number of studies on the individual flows in DS network have been done in [11, 15, 16, 17, 18, 19]. Some of the conclusions of the earlier work include: (a) an application may not realize its reserved rate even when bandwidth is not oversubscribed, (b) the achieved rate is a strong function of the round-trip-time (RTT) over the network and (c) the realized rate is not proportional to the reserved rate when resources are plentiful or when resources are oversubscribed.

In this dissertation, we propose new techniques to realize the reserved rate. The proposed techniques deal with either or both of sender, marker, and dropper. We evaluate our techniques through a number of simulations. We also develop analytical models of TCP throughput in the DS network for quantitative understanding of TCP behavior. The models are validated through a number of simulations.

While we worked on the first part, we have observed that behaviors of aggregated flows in the DS network are quite different from the behaviors of the individual flows and leave us a number of unsolved problems. For example, (a) unfair distribution of
contract rate within aggregations, and (b) resource, here contract rate, management and control for achieving QoS (Quality of Service) goals of individual flows within aggregation. This motivates the second part of this dissertation. We mainly focus on aggregated flows in the second part: We propose static and dynamic marking schemes for aggregated flows and present a number of simulations to study the behaviors of aggregated flows in the DS network. We also consider receiver-intensive applications such as FTP (File Transfer Protocol) and HTTP. We propose a simple mechanism for receiver-side marker and study the interaction between the sender-side and the receiver-side markers.

This dissertation is organized as follows: In Chapter II, we study the interaction between the transport layer protocol and the differentiated drop policies of the network in realizing the contract rates. We propose a number of mechanisms to better realize the contract rates and evaluate the proposed mechanisms through extensive simulations. In Chapter III, we propose analytical models of TCP behavior in a DS network. Our models quantitatively characterize TCP throughput as functions of the contract rate, the packet-drop rate and the round-trip time (RTT). We also extend our models to aggregated flows. The models are validated through a number of simulations. In Chapter IV, we address issues of aggregated flows in the DS network and propose simple techniques for aggregated marking. In Chapter V, we propose a more sophisticated marking algorithm for aggregated flows. We show the algorithm achieves the possible QoS goals of individual flows while maximizes throughput for the unreachable goals to avoid resource wastage. We also propose a simple scheme for receiver-side markers and study the interaction between the sender-side and the receiver-side markers through simulations. In Chapter VI, we summarize this dissertation and present directions for future work.
CHAPTER II

REALIZING THROUGHPUT GUARANTEES IN A DIFFERENTIATED SERVICES NETWORK

The realized throughput is a result of the combination of dropping policy of the network and the policy of the transport protocol in how it reacts to these drops. TCP reacts to congestion by halving the congestion window and increases the window additively when packets are delivered successfully. Exponential decrease (halving the congestion window) is required to avoid congestion collapse [20] and TCP treats a packet drop as an indication of congestion. The additive increase is due to two reasons [20, 21]: (a) to allow congestion to be cleared and (b) to allow fair share of bandwidth among multiple flows.

In the DS network, however, these additive-increase and multiplicative-decrease make it hard to protect the reservation rate. When TCP reacts to an OUT packet drop by halving its congestion window and increases additively, it may not reach its reservation rate. In this chapter, we study the interaction between the transport protocol and the differentiated drop policies of the network in realizing the reserved throughputs. We propose a number of mechanisms to better realize the target rates. Specifically, the chapter makes the following contributions: (1) proposes new schemes for improving the realization of reserved rates in a differentiated services network, (2) evaluates the proposed schemes through simulations to show that the new schemes provide better realization of throughput guarantees. (3) evaluates the sensitivity of the throughput guarantees to different parameters such as round-trip-times (RTTs), (4) studies the impact of aggregation on the results and (5) proposes new quantitative measures for evaluating the various schemes.
A. Background

Let \( r_i \) denote the reserved rate of a flow \( i \) and \( C \) represent the capacity of a bottleneck link in the network. If a number of flows pass through this link, then \( \sum_{i=1}^{n} r_i \) of the link capacity is allocated for IN packets, where \( n \) is the number of flows going through the link. The excess bandwidth at this link is then given by

\[
e = C - \sum_{i=1}^{n} r_i
\]  

(2.1)

when excess bandwidth is equally shared. This excess bandwidth at the links beyond the allocated/reserved capacity can then be shared by all the flows. This excess bandwidth can be shared in many different ways. Sharing proportional to the reserved rates and equal sharing are two of the logical choices. We will deal with equal sharing of the excess bandwidth. Equal sharing allows flows without any reservation to continue receiving some service while proportional sharing may deny them of such service. Hence, a flow’s target rate is given by

\[
t_i = r_i + \frac{e}{n}
\]  

(2.2)

The marking, dropping schemes along with the transport protocol’s reaction to congestion determine how closely the flow can realize the target rate. To understand the dynamics of this interaction, we conducted several simulation experiments using the ns-2 [22] simulator.

Fig. 3 shows a simple network topology that enables studying this interaction. Sources 1,2,...10 are TCP-Reno sources. The marker uses a sliding window marking strategy proposed in [11]. The router uses RED parameters 20/40/0.5 for the OUT packets and 50/100/0.02 for the IN packets. The reserved rates for each flow are shown in the figure. The total allocated bandwidth is 7.2 Mbps. The link bandwidth
is set at 12 Mbs, 8Mbs and 6 Mbs in three different experiments to simulate allocations of 60%, 90%, and 120% of capacity. All the flows are assumed to have the same RTTs of 40 msec. and run for 30 seconds. The results of the simulation are shown in Fig. 4.

The realized throughput is considerably different from the target rates for most of the flows. TCP reacts to a packet loss by halving its congestion window. TCP then slowly increases the congestion window on successful transmission of packets. This additive increase of congestion window after a packet loss results in different recovery times to regain the congestion window for different flows. A flow with a larger rate takes longer time to reach its original rate compared to a flow with a smaller rate. Hence, the smaller flows realize higher rates. Again, we observe that the realized rates are biased toward the smaller rates. Why did packet loss occur even when the allocated capacity is below the link capacity? First, the best effort flows may transmit data above the excess bandwidth available. Second, TCP flows continue increasing their transmission rates even after reaching their reserved rates until a packet drop. Third, bursty arrival may cause some packets to be dropped even when the combined
Fig. 4. Realized rates at different levels of subscription
sending rate is lower than the link capacity. For each flow, the loss in bandwidth from the target rate is a function of the number of packets of that flow dropped and the recovery time (to the original sending rate) as a result of a packet drop. In the following sections, we will look at methods that affect (a) the dropping rate and (b) the recovery time as a result of a packet drop. The dropping rate can be affected by the marker, the network, the sender (burstiness of the sender, for example) and the interaction among the different flows at the network queues. The recovery time is, however, only affected by the transport protocol.

These results indicate the need for studying mechanisms that can better realize the target rates. We study a number of mechanisms to better realize the target rates. We will look at three different levels of bandwidth allocation: (a) 60% of the available bandwidth is allocated, (b) 90% of the available bandwidth is allocated, and (c) 120% of the available bandwidth is allocated. With careful provisioning of the resources, the network should never operate in mode (c). However, without any mechanisms like RSVP [23] or path pinning, the flows with reservations could all go through the same link to result in oversubscription. We study this mode (c) as well to understand how well the different schemes work in such a situation. Recently, simulation work on marking and dropping strategies has been done [11, 15, 17, 16]. These results indicated the need for better mechanisms for marking and dropping. Some of the conclusions of the earlier work include: (a) an application may not realize its reserved rate even when bandwidth is not oversubscribed [11, 15], (b) flows with smaller RTTs may achieve higher bandwidth than flows with larger RTTs, (c) that realized rate is not proportional to the reserved rate when resources are plentiful [15]. These results provided strong motivation for our work reported here. Some of the earlier work [17, 16] has focused on similar issues in networks where marking strategies are different than the ones studied here. Our work considerably extends the
earlier work and proposes new approaches to improving the realization of bandwidth guarantees.

We will consider two quantitative measures in evaluating various schemes. First, do the flows receive bandwidth corresponding to the reserved rates? Second, is the excess bandwidth fairly shared among all the flows? For an individual flow, without the knowledge of other flows, achieving the reserved rate is important. However, if the service provider cannot provide mechanisms to share the excess bandwidth fairly, the users may not perceive service differentiation with higher reserved rates and may not be willing to pay for higher-cost services.

The rest of this chapter is organized as follows: Section B proposes a number of schemes to better achieve target rates in a differentiated services network. Section C presents a comparative evaluation of the proposed schemes. Section D presents an evaluation of the proposed schemes when flows may be aggregated. In Section E, we summarize this chapter and provide directions for further research.

B. Policies to achieve target rates

1. Limiting OUT packets

We first focus on the flows with the higher target rates. We noticed that in the earlier simulation, there were no IN packets dropped. A packet drop and a resulting contraction in sending rate by TCP resulted in realizing rates below the target rates. If a flow sent out few OUT packets, it is likely that this flow will not experience as many packet drops and hence may be able to realize the target rate. This policy aims to impact the number of packets dropped for each flow to better realize the target rates.

We modified the marker to send back information to the sender whenever one of
its packets is marked OUT. The sender reduces the window by a packet as a result of this indication to avoid sending out any more OUT packets into the network. The result of this modification is shown in Fig. 5. It is observed that the flows with higher target rates realize better throughput compared to the original RIO scheme. All the flows nearly achieved their reserved rates. However, the flows with higher targets do not come close to achieving their target rates at 60% level. At this subscription level, there is considerable excess bandwidth and the no-OUT mechanism doesn’t try to capture any of the excess bandwidth since it prevents a flow from sending above its reserved rate. Thus, the excess bandwidth is fully captured by non-reserved best effort flows. This is clearly unacceptable since non-reserved flows get more bandwidth than other reserved flows. This scheme doesn’t result in fair sharing of the excess bandwidth, and hence cannot provide proper service differentiation. At the 120% level, there is no excess bandwidth, and hence this scheme achieves rates very close to the targets. Limiting the OUT packets works well when the subscription level is high and doesn’t work well at lower subscription levels.

2. Inverse-rate drop policy

This mechanism requires that every packet that is injected into the network is stamped with the service level besides the IN/OUT marking. The service level could be the reserved rate. The dropping policy is modified to take the service level into account. The higher the service level, the lower the probability for dropping a packet of that flow. Since each packet is marked with the service level, there is no need to maintain a state for each flow at the router. The rationale for this inverse drop policy (with respect to the service level) is that the flows at higher service level should get less packets dropped to counter the longer recovery times of the flows with higher target rates. This mechanism requires modifications to the marking and dropping policies.
Fig. 5. No-OUT scheme at different subscription levels
The realized throughput of a TCP flow [21, 24, 25] is given by

\[ t \leq 1.2 \times \frac{B}{(RTT \times \sqrt{p})} \]  

(2.3)

where B is the packet size of the flow, RTT is the round-trip-time and p is the drop probability. Consider two flows with different target rates \( t_1 \) and \( t_2 \) with the same packet size and RTT. Then, based on the above equation

\[ \frac{t_1}{t_2} = \sqrt{\frac{p_2}{p_1}} \]  

(2.4)

When we compare the number of packets dropped in a unit of time

\[ \frac{d_1}{d_2} = \frac{(t_1 \times p_1)}{(t_2 \times p_2)} = \sqrt{\frac{p_1}{p_2}} = \frac{t_2}{t_1} \]  

(2.5)

This indicates that the number of packets dropped in a unit time should be inversely proportional to the target rate of that flow.

In a differentiated services network, a flow consists of IN packets and OUT packets. The rate of IN packets corresponds to the reserved rates and the rate of OUT packets correspond to the share of excess bandwidth. Ideally, the rate of OUT packets are the same for any two flows when excess bandwidth is equally shared. Also, if the RIO parameters are chosen carefully and if the network is not oversubscribed, the packet drops will all be OUT packets. So, the number of packets dropped in a unit time based on only the OUT packets of each flow is given by

\[ \frac{d_1}{d_2} = \frac{(OUT_1 \times p_1)}{(OUT_2 \times p_2)} = \frac{p_1}{p_2} \]  

(2.6)

where \( OUT_1 \) and \( OUT_2 \) correspond to the rate of OUT packets for flows 1 and 2 respectively and \( OUT_1 = OUT_2 \) in ideal situations. From the above two equations, it is clear that the drop probabilities should be inversely proportional to the target rates in a differentiated services network where most of the drops are OUT packets,
i.e.,

\[ p_1/p_2 = t_2/t_1 \]  \hspace{1cm} (2.7)

The marker is unaware of the excess bandwidth and hence cannot easily determine the target rates. Hence, we use the reserved rates for marking the packets instead of the target rates. This analysis gives an idea of how dropping policy can be modified to counter the TCP congestion avoidance to better realize the target rates. The advantage of this scheme is that it doesn't require modifications to the TCP layers and hence can work with the existing TCP software.

Consider a non-responsive UDP flow sending data at rate \( s_i \), with a reserved rate of \( r_i \) and a target rate of \( t_i \). Then, the excess packets above \( t_i = s_i - t_i \) should ideally be dropped from this flow [26]. The drop probability for this flow then should be

\[ p_i = (s_i - t_i)/s_i = (1 - t_i/s_i) \]  \hspace{1cm} (2.8)

This also shows that the drop probability should be inversely related to the target rate of the flow (for a given sending rate), i.e., the higher the target rate, the lower the drop probability should be.

We used the following equation to calculate the drop probability of a flow with a reserved rate of \( r_i \)

\[ p_i = k/(mk + r_i/r_{\text{min}}) \]  \hspace{1cm} (2.9)

where \( k, m \) and \( r_{\text{min}} \) are suitably chosen constants. We chose \( r_{\text{min}} \) to be 0.1 Mbs, the smallest reservable rate in our simulations. When \( r_i = 0 \), \( p_i = 1/m \). Hence, \( m \) can be chosen based on the target drop probability for a flow with zero reservation i.e., best-effort flow. Similarly, the parameter \( k \) can be chosen based on the target drop probability required for the flow with \( r_{\text{min}} \) reservation. We chose \( k = 4 \) and \( m = 2 \).

Fig. 6 shows the results of this mechanism at different subscription levels. Most
of the flows achieved their reserved rates better than in the original RIO scheme. The results also indicate that this inverse-drop policy achieved rates fairly close to the target rates. However, this scheme has a bias towards the higher rates. Similar trends in performance are observed at different subscription levels. Target rates are nearly achieved with a bias toward the higher rate flows.

To verify the sensitivity of this approach to the parameters chosen, we ran the simulations with different sets of target rates with the same parameters of k, m and \( r_{\min} \). The results had similar trends. The higher rate flows achieved targets better than unmodified RIO scheme.

3. Three drop precedences

This approach uses three drop precedences, IN, OUT-IN and OUT-OUT. The marker continues marking packets IN when they conform to the leaky bucket profile. Instead of marking the remaining packets simply as OUT, it separates them into two categories. The marker keeps track of the long-term sending rate of the sender. If the long-term sending rate is higher than the reserved rate, more number of packets are marked OUT-OUT and if the long-term sending rate is lower than the reserved rate, more number of packets are marked OUT-IN. The router drops OUT-OUT packets earlier than the OUT-IN packets to give preference to responsive flows. This enables the marker to give a better treatment to flows that are falling behind in realizing their reserved rates.

Ideally, realized throughput of the sender should be used as a long-term measure. The acknowledgments may reach the sender through a different path without going through the marker. Also, if the sender is sending different sized packets, the marker would have to be aware of the size to estimate the goodput correctly even if the acknowledgments are going through the marker. To minimize these problems, we
Fig. 6. Inverse-rate drop scheme at different subscription levels
simply use the sending rate of the sender. As a result, for flows that don't respond to congestion, most of the packets above the reserved rate will be marked OUT-OUT. Flows that respond to congestion may have sending rates below the reserved rates and hence receive better treatment at the bottleneck link since more of their packets (above the reserved level) are marked OUT-IN.

Long-term sending rates are calculated by the same scheme as the scheme used to mark packets IN or OUT, but with a longer time window of 10 seconds. For the simulations, we use RED parameters 20/40/0.5 for OUT-OUT, 30/50/0.5 for OUT-IN and 50/100/0.02 for IN packets.

This approach requires modification to the marker and the router. The router deals with three drop precedences rather than two precedences in the other schemes. The marker also has to keep state information for each flow. However, the state information need only be kept at the ingress router for each flow and not at every router the flow passes through in the network. Earlier schemes we discussed did not have a need to maintain any state information for the flows. This scheme requires no changes to the sender.

The results of the simulations are shown in Fig. 7. Most of the flows achieved their reserved rates better than in the original RIO scheme. It is also observed that this scheme realizes rates fairly close to target rates. However, we note that the flows with smaller targets exceed their targets and the flows with higher targets don't reach their targets. Similar trends are observed at different subscription levels. The target rates are nearly achieved with a bias toward the smaller flows. At lower subscription levels, all the flows send data above the reserved rates and hence the OUT packets are marked mostly OUT-OUT reducing the scheme to a two-drop precedence scheme.
Fig. 7. Three drop precedences at different subscription levels
4. Two-windows TCP

Since the unmodified TCP protocol is unaware of reservation rates, congestion avoidance mechanisms could not protect the reserved rate of the flow. To avoid this problem, the congestion window of TCP is broken up into two pieces, the reserved window, \textit{rwnd}, and the excess bandwidth window \textit{ewnd} such that \textit{cwnd} = \textit{rwnd} + \textit{ewnd}. The reservation window \textit{rwnd} is obtained by multiplying the RTT of the flow with its reserved rate \textit{r}. TCP is then modified to only reduce the \textit{ewnd} by half when an OUT packet is dropped and \textit{cwnd} as a whole is reduced only as a response to an IN packet loss. This requires that the sender keep track of how his packets are marked IN/OUT and then determine the congestion avoidance mechanism based on the dropped packet’s marking. An OUT packet drop is considered as an indication of oversubscription of the excess bandwidth and an IN packet drop is considered an indication of oversubscription of link bandwidth. The modified congestion avoidance algorithm is shown in Fig. 8. This scheme requires modifications to the transport protocol. It also requires that the sender be informed of IN/OUT markings of packets. This can be achieved by integrating the marker with the sender or by the marker informing the sender of the markings. A similar scheme has been recently studied in [16] with a slightly different marking strategy. We include this scheme here for completeness and to compare the other schemes with this approach.

The results of the simulations are shown in Fig. 9. It is observed that this scheme also achieves rates better than the original RIO algorithm. At 60\% and 90\% subscription levels, the flows with smaller target rates exceed their targets and the flows with the higher targets don’t reach their target rates. At 120\% subscription, the flows with higher target rates are favored.

The throughput of each flow can be divided into IN packets corresponding to
Upon a packet loss:
if OUT packet loss
    rwnd = rtt * reserved_rate
if rwnd < cwnd
    ewnd = cwnd - rwnd
    cwnd = rwnd + ewnd/2
else
    cwnd = cwnd/2

Fig. 8. Modified congestion avoidance algorithm

the reserved rate and OUT packets corresponding to the shared excess bandwidth. After an OUT packet drop, the time to recover to the original congestion window is inversely proportional to the window size because of TCP’s fast recovery mechanism (where congestion window is increased by one only after all the packets in the window are acknowledged). Hence, the flows with smaller target rates send more OUT packets than the flows with higher target rates and get more excess bandwidth. However, the IN packet throughputs are hardly affected by the target rates because the congestion window size is maintained to be larger than the reserved window. Thus, at lower subscription levels, even though the flows achieve their reserved rates, the flows with higher target rates cannot reach their target rates. TCP’s burstiness was observed to cause packets to be marked OUT even when there is sufficient bandwidth leading to discrepancies in the realized throughputs. Timer based transmission of packets could reduce the burstiness to increase the realization of target rates [16]. Suggested modifications (two-windows, use of timers) may make TCP more aggressive than current versions of TCP. Also, modifications to widely deployed software may be harder than modifications to the new network hardware to be deployed.
Fig. 9. Two-windows TCP at different subscription levels
Table I. Comparison of all the schemes

<table>
<thead>
<tr>
<th>Subscript. Lvl.</th>
<th>0</th>
<th>0.1</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>Util.</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>60%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target Rate</td>
<td>0.48</td>
<td>0.58</td>
<td>0.98</td>
<td>1.48</td>
<td>2.48</td>
<td>100%</td>
<td>0</td>
</tr>
<tr>
<td>Original RIO</td>
<td>0.63</td>
<td>0.66</td>
<td>0.80</td>
<td>1.10</td>
<td>2.01</td>
<td>87%</td>
<td>0.29</td>
</tr>
<tr>
<td>Inv. Rate Drop</td>
<td>0.33</td>
<td>0.35</td>
<td>0.83</td>
<td>1.49</td>
<td>2.77</td>
<td>96%</td>
<td>0.19</td>
</tr>
<tr>
<td>No-OUT</td>
<td>1.66</td>
<td>0.24</td>
<td>0.62</td>
<td>1.12</td>
<td>2.12</td>
<td>96%</td>
<td>0.61</td>
</tr>
<tr>
<td>Two Windows</td>
<td>0.38</td>
<td>0.52</td>
<td>0.71</td>
<td>1.06</td>
<td>2.46</td>
<td>86%</td>
<td>0.23</td>
</tr>
<tr>
<td>Three drop prec.</td>
<td>0.56</td>
<td>0.65</td>
<td>0.81</td>
<td>1.20</td>
<td>2.12</td>
<td>90%</td>
<td>0.22</td>
</tr>
<tr>
<td><strong>90%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target Rate</td>
<td>0.08</td>
<td>0.18</td>
<td>0.58</td>
<td>1.08</td>
<td>2.08</td>
<td>100%</td>
<td>0</td>
</tr>
<tr>
<td>Original RIO</td>
<td>0.09</td>
<td>0.35</td>
<td>0.64</td>
<td>1.04</td>
<td>1.81</td>
<td>98%</td>
<td>0.15</td>
</tr>
<tr>
<td>Inv. Rate Drop</td>
<td>0.02</td>
<td>0.17</td>
<td>0.59</td>
<td>1.05</td>
<td>2.11</td>
<td>99%</td>
<td>0.03</td>
</tr>
<tr>
<td>No-OUT</td>
<td>0.30</td>
<td>0.16</td>
<td>0.53</td>
<td>1.04</td>
<td>1.91</td>
<td>99%</td>
<td>0.13</td>
</tr>
<tr>
<td>Two Windows</td>
<td>0.04</td>
<td>0.07</td>
<td>0.58</td>
<td>0.89</td>
<td>2.17</td>
<td>94%</td>
<td>0.10</td>
</tr>
<tr>
<td>Three drop prec.</td>
<td>0.20</td>
<td>0.21</td>
<td>0.57</td>
<td>0.99</td>
<td>1.97</td>
<td>99%</td>
<td>0.09</td>
</tr>
<tr>
<td><strong>120%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target Rate</td>
<td>0.00</td>
<td>0.08</td>
<td>0.42</td>
<td>0.83</td>
<td>1.67</td>
<td>100%</td>
<td>0</td>
</tr>
<tr>
<td>Original RIO</td>
<td>0.01</td>
<td>0.09</td>
<td>0.43</td>
<td>0.78</td>
<td>1.67</td>
<td>99%</td>
<td>0.03</td>
</tr>
<tr>
<td>Inv. Rate Drop</td>
<td>0.01</td>
<td>0.08</td>
<td>0.42</td>
<td>0.82</td>
<td>1.65</td>
<td>99%</td>
<td>0.01</td>
</tr>
<tr>
<td>No-OUT</td>
<td>0.01</td>
<td>0.09</td>
<td>0.42</td>
<td>0.82</td>
<td>1.65</td>
<td>99%</td>
<td>0.01</td>
</tr>
<tr>
<td>Two Windows</td>
<td>0.01</td>
<td>0.04</td>
<td>0.26</td>
<td>0.66</td>
<td>1.81</td>
<td>93%</td>
<td>0.12</td>
</tr>
<tr>
<td>Three drop prec.</td>
<td>0.01</td>
<td>0.09</td>
<td>0.47</td>
<td>0.88</td>
<td>1.54</td>
<td>99%</td>
<td>0.06</td>
</tr>
</tbody>
</table>

C. Comparison of all the schemes

Table I compares all the studied schemes against each other. For each scheme, we compute the realized utilization of the link. We also computed the mean-square-error (MSE) of the target rates and the realized rates. Since No-OUT scheme cannot let the flows with reservations share the excess bandwidth, we will not consider this scheme further, even though it shows good results at higher subscription level.

The results show that all the considered schemes do better at realizing reserved rates than the original RIO scheme. The new schemes (except for No-OUT) also achieve better realization of target rates (observed by MSE). The mean-square error of
each scheme increases as the subscription level is decreased. At 60% subscription level, there remains 40% excess bandwidth. The difficulty in sharing this excess bandwidth among all the sources causes the achieved rates to diverge from the target rates. The results of unmodified RIO and three-drop schemes have the same trend that the flows with lower reserved rates get more than their target rates, and the flows with higher reserved rates cannot reach their target rate. It is because of two fundamental reasons. First is the TCP congestion avoidance mechanism. TCP was designed to share bandwidth equally (the additive increase results in fairer share [20, 27]), and thus the flows having more bandwidth lose more when congestion occurs. The second reason is that the profile marker does not know the target rates and uses reserved rates. In case of higher subscription level, the target rates and the reserved rates are not much different, and each source gets close to the target rates. The results of inverse-rate drop and two windows schemes have opposite trend to the results of unmodified RIO and three-drop schemes. The divergence of the results of inverse-rate drop scheme from the target rates is due to dropping packets based on reserved rates instead of target rates. At lower subscriptions, this results in larger errors.

1. Impact of a non-responsive source

A source not responding to congestion, ideally should not take bandwidth away from sources that respond to congestion. Otherwise, a non-responsive flow can disturb the throughput guarantees of responsive multimedia flows. Fig. 10 shows the impact of a non-responsive source with no reservation on the remaining sources. The simulation consisted of 10 TCP sources at 1 Mbps subscription and 10 best-effort sources along with a non-responsive UDP source at varying rates on a 20Mbps link. As the non-responsive source rate is increased, the reserved TCP flows start losing their reserved bandwidth in the unmodified RIO. However, the other schemes better protect the
TCP flows with reservations. As the non-responsive source rate is increased, it realizes more and more bandwidth until it reaches a bandwidth of 10 Mbps. In inverse-drop and three-drop precedences schemes, the TCP reservations are protected, while the best-effort connections realize less and less bandwidth with increasing non-responsive source rates. In the unmodified RIO and the two-windows schemes, the reserved flows also get affected. The inverse-rate scheme and the three-drop precedence schemes provide better service differentiation between reserved flows and best-effort flows. In all the schemes, the non-responsive source gets contained around 10Mbps.

Fig. 11 shows the impact of a non-responsive source with a reservation of 1 Mbps on 10 other TCP sources with reservations of 1Mbps. The link bandwidth is kept at 12 Mbps. The non-responsive source sending rate is kept at 6 Mbps. The non-responsive flow achieves better bandwidth in all the cases. The non-responsive flow achieved a rate of 2.76 Mbps in the unmodified RIO case, 2.4 Mbps in the inverse-rate drop scheme, 1.5 Mbps with three drop precedences and 2.5 Mbps in the two-windows scheme with a corresponding loss in the realized bandwidth of the TCP sources. All three proposed schemes contained the non-responsive flow better than the unmodified RIO scheme. The performance with three drop precedences is much better than the other schemes. It is also observed that all the schemes performed worse at reducing the impact of a non-responsive flow when the non-responsive flow has a reservation. This indicates that other schemes to identify (and possibly punish) non-responsive flows are required in addition to the marking and dropping policies.

2. Impact of different RTTs

The bandwidth realized by TCP flows is sensitive to RTTs between the senders and the receivers. So far, our simulation experiments considered flows with the same RTTs. How do the results get affected when different RTTs are considered. To understand
Fig. 10. Impact of a non-responsive UDP flow
this, we considered flows at RTTs of 20ms, 40ms, 60ms, 80ms, and 100ms. At each RTT, we considered three flows, one with a high reservation, the second flow with a lower reservation and a third flow that is best-effort for a total of 15 flows. In the first experiment, we considered reservation levels of 1Mbps and 0.5Mbps and a link bandwidth of 10 Mbps (with 7.5 Mbps allocated out of 10Mbps). In a second experiment, we considered the same flows on a link bandwidth of 30Mbps such that only 25% of the bandwidth is allocated.

The simulation results are shown in Table II. Clearly, in all the schemes, the flows with smaller RTTs experienced better service than flows with longer RTTs. However, no scheme seems to have a clear advantage over the others. To quantify the differences, we used the following measures: (a) Mean Square Error (MSE) used earlier based on the achieved and target rate differences, (b) Utilization, (c) Fairness within a service class (best-effort, 0.5M reservation, 1M reservation) measured by the maximum rate divided by the minimum rate within the same reservation level,
Table II. Throughputs of different schemes at different RTTs

<table>
<thead>
<tr>
<th>Scheme</th>
<th>20ms BE/0.5M/1M</th>
<th>40ms BE/0.5M/1M</th>
<th>60ms BE/0.5M/1M</th>
<th>80ms BE/0.5M/1M</th>
<th>100ms BE/0.5M/1M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target rates</td>
<td>0.17/0.67/1.17</td>
<td>0.17/0.67/1.17</td>
<td>0.17/0.67/1.17</td>
<td>0.17/0.67/1.17</td>
<td>0.17/0.67/1.17</td>
</tr>
<tr>
<td>Original RIO</td>
<td>0.46/0.71/1.23</td>
<td>0.29/0.68/1.09</td>
<td>0.23/0.61/1.09</td>
<td>0.27/0.61/0.90</td>
<td>0.18/0.50/0.92</td>
</tr>
<tr>
<td>Inv. rate drop</td>
<td>0.09/0.65/1.05</td>
<td>0.08/0.64/1.41</td>
<td>0.06/0.56/1.17</td>
<td>0.02/0.55/1.19</td>
<td>0.01/0.55/1.20</td>
</tr>
<tr>
<td>Three drop prec.</td>
<td>0.43/0.69/1.19</td>
<td>0.28/0.70/1.12</td>
<td>0.26/0.62/1.02</td>
<td>0.19/0.52/1.01</td>
<td>0.20/0.53/1.02</td>
</tr>
<tr>
<td>Two windows</td>
<td>0.12/0.49/1.27</td>
<td>0.01/0.67/1.03</td>
<td>0.02/0.53/0.99</td>
<td>0.03/0.51/0.93</td>
<td>0.01/0.49/0.91</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target rates</td>
<td>1.50/2.00/2.50</td>
<td>1.50/2.00/2.50</td>
<td>1.50/2.00/2.50</td>
<td>1.50/2.00/2.50</td>
<td>1.50/2.00/2.50</td>
</tr>
<tr>
<td>Original RIO</td>
<td>3.09/3.17/3.37</td>
<td>1.66/1.93/2.20</td>
<td>1.18/1.50/1.71</td>
<td>1.02/1.03/1.40</td>
<td>0.96/1.14/1.40</td>
</tr>
<tr>
<td>Inv. rate drop</td>
<td>1.03/2.87/6.84</td>
<td>0.46/1.81/4.58</td>
<td>0.29/1.08/3.38</td>
<td>0.23/0.94/2.12</td>
<td>0.19/0.87/2.32</td>
</tr>
<tr>
<td>Three drop prec.</td>
<td>3.09/3.17/3.37</td>
<td>1.66/1.93/2.20</td>
<td>1.18/1.50/1.71</td>
<td>1.02/1.03/1.40</td>
<td>0.96/1.14/1.40</td>
</tr>
<tr>
<td>Two windows</td>
<td>1.98/3.22/4.01</td>
<td>1.42/2.02/2.66</td>
<td>0.93/1.63/1.82</td>
<td>0.73/1.12/1.56</td>
<td>0.63/1.10/1.26</td>
</tr>
</tbody>
</table>

(d) Service differentiation measured by the minimum rate achieved at a particular reservation level divided by the maximum rate achieved at the next lower reservation level, and (e) Reservation success measured by counting the number of flows reaching the reserved rate. Since we considered three different service levels, we get three fairness values (one for each class) and two service differentiation values (min of 0.5M/max of BE, min of 1M/max. of 0.5M). Ideally, fairness values should equal 1 and the service differentiation values should be greater than 1. These quantitative measures are shown in Table III for both the experiments. For example, two-windows scheme achieve a fairness measure of 3.18 for 1 Mbps flows at 25% subscription i.e., a flow (with RTT of 20ms) achieved 3.18 times the bandwidth achieved by another flow with the same reserved rate of 1Mbps. Similarly, a service differentiation of 0.4 between 0.5 Mbps flows and 1 Mbps flows means that a flow with a reservation of 1 Mbps achieved only 40% of the bandwidth achieved by a flow with a reservation of only 0.5 Mbps due to differences in RTTs.

From the table, it is observed that proposed schemes do better at realizing reserved rates and provide better service differentiations. It is also observed that the MSE is higher at lower subscriptions for all the schemes, pointing again to the dif-
Table III. Performance summary of different schemes at different RTTs

<table>
<thead>
<tr>
<th>Scheme</th>
<th>MSE(Mbps)</th>
<th>Util.</th>
<th>Fairness (BE/0.5M/1M)</th>
<th>Serv. Differentiation (0.5M/BE)/(1M/0.5M)</th>
<th>Reservation success</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original RIO</td>
<td>0.14</td>
<td>98%</td>
<td>2.33/1.41/1.39</td>
<td>1.09/1.30</td>
<td>80%</td>
</tr>
<tr>
<td>Inv. rate drop</td>
<td>0.16</td>
<td>99%</td>
<td>8.43/1.17/1.37</td>
<td>8.85/1.84</td>
<td>100%</td>
</tr>
<tr>
<td>Three drop proc.</td>
<td>0.12</td>
<td>99%</td>
<td>2.11/1.31/1.11</td>
<td>1.21/1.54</td>
<td>100%</td>
</tr>
<tr>
<td>Two windows</td>
<td>0.15</td>
<td>87%</td>
<td>15.3/2.25/1.33</td>
<td>3.17/0.57</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>MSE(Mbps)</th>
<th>Util.</th>
<th>Fairness (BE/0.5M/1M)</th>
<th>Serv. Differentiation (0.5M/BE)/(1M/0.5M)</th>
<th>Reservation success</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original RIO</td>
<td>0.83</td>
<td>90%</td>
<td>3.18/2.77/2.49</td>
<td>0.37/0.44</td>
<td>100%</td>
</tr>
<tr>
<td>Inv. rate drop</td>
<td>1.50</td>
<td>90%</td>
<td>5.37/3.28/2.95</td>
<td>0.84/0.80</td>
<td>100%</td>
</tr>
<tr>
<td>Three drop proc.</td>
<td>0.83</td>
<td>90%</td>
<td>3.18/2.77/2.40</td>
<td>0.37/0.44</td>
<td>100%</td>
</tr>
<tr>
<td>Two windows</td>
<td>0.82</td>
<td>87%</td>
<td>3.13/2.92/3.18</td>
<td>0.56/0.40</td>
<td>100%</td>
</tr>
</tbody>
</table>

difficulty in assigning the excess bandwidth fairly. The three-drops scheme and the two-windows scheme have better mean square error from the target rates at 75% and 25% subscription level, respectively. They also have better fairness measures compared to the other two schemes. In three-drops and inverse-rate drop schemes, all flows reach their reserved rates. Inverse-rate drop scheme achieves the best service differentiation among the schemes considered.

D. Effect of aggregated sources

In this section, we present simulation results with aggregated sources. The motivation of these simulations is the fact that it may not be practical to reserve bandwidth and put a profile marker at each host in the real Internet. A profile marker may be assigned to a group of hosts (a university or a company, for example). In this situation, we have two issues. First issue is how does aggregation impact the guarantees of all the flows, aggregated or otherwise. The second issue is how to assign the reserved group bandwidth to each host in the group so that an individual application (such as IP telephony) can realize the necessary guarantees. In this section, we focus on the first issue.
The simulation topology is the same as the topology used in the previous simulations shown in Fig. 3 except that the odd-numbered sources (1, 3, 5, 7, and 9) now consist of an aggregation of three separate flows. Hence, at each reservation level, we have an individual flow and an aggregated flow. We used the same RED parameters and reservation rates, and the bottleneck bandwidth is set to 9 Mbps. Since a total of 7.2 Mbps is allocated, the subscription level is 80%. We ran two simulation experiments using this configuration. In the first experiment, we used the same RTT, 60 ms for all the 20 flows. In the second experiment, we used RTTs of 40 ms, 60 ms, and 80 ms for the three sources of an aggregated pool, and an RTT of 60 ms for single sources. Flows within an aggregated source may have different RTTs as they may be talking to different hosts after passing through the same bottleneck link.

Fig. 12 shows the results of the simulations. In all the schemes, the results are not much different across the two experiments. This may indicate that aggregation may blunt the effect of RTT differences. In all the schemes, the aggregated sources realize higher throughputs than a single source. The three flows within an aggregated source claim three times as much of the shared excess bandwidth than a single source and hence the difference.

Fig. 13 shows the bandwidth realized by two sources, one aggregated and one single source, both at 2 Mbps reservation. In the figure, the peak rates of the aggregated sources are much higher than the peak rates of the single source. It means that the aggregated sources get more excess bandwidth than the single source. The figure also shows that the measured rate of the single source fluctuates more frequently than the rate of aggregated sources. Since each of the aggregated sources reacts to congestion individually, the rate of the aggregated sources is smoother.
Fig. 12. Comparisons between aggregated sources and single source
Fig. 13. Measured rates vs. time of single and aggregated sources with 2Mbps subscription

E. Summary

Multimedia applications require throughput guarantees for delivery over networks. In this chapter, we have proposed and evaluated several schemes to improve the realization of throughput guarantees in the future internet that employs a differentiated services framework. The proposed schemes are shown to improve the realization of reserved rates and to provide better service differentiation than the original RIO scheme. The results also show that non-responsive sources can be controlled better by the proposed schemes. Among the proposed schemes, inverse-rate drop realized rates closer to target rates and provided better differentiation of services. Three drop precedences scheme has advantage in controlling a non-responsive sources and reducing the impact of differences in RTTs. Both these schemes do not require modifications to transport (TCP) layers. The two-windows scheme modifies TCP to provide similar
benefits. All of the proposed schemes require enhancements to the basic RIO scheme, either at the marker, sender or in the network.

We also reported on extensive simulations on the sensitivity of the results to differences in RTTs of flows and aggregation of flows. Our results indicate that the provided guarantees are soft i.e., not realized in all the situations. It was observed that a non-responsive flow can disturb the throughputs of other flows. It was observed that differences in RTTs resulted in the realization of different rates even when flows had the same reservations. However, it was shown that the impact of differences in RTTs could be reduced by the aggregation of sources and fair sharing of bandwidth at the edge routers. Even though fair-sharing of excess bandwidth proved to be a challenge, the reservations could be mostly met.
CHAPTER III

MODELING TCP BEHAVIOR IN A DIFFERENTIATED SERVICES NETWORK
As we have observed in the previous chapter, it is difficult to guarantee requested throughput to individual TCP flows in AF networks. The similar observations have been addressed in several recent studies [17, 28, 29]. When a TCP flow detects packet loss, the flow assumes that the packet loss is due to congestion in the network. A TCP flow tries to avoid this congestion by halving its transmission rate. This reduced rate may be less than its reservation rate, and this results in loss of throughput. Earlier simulation results [17, 28] have shown that flows with relatively higher reservations may realize less throughput than their target rates while flows with smaller reservation may realize throughput higher than their target rates. It has been also shown that it is difficult to guarantee absolute bandwidth with a simple marking and dropping scheme [29]. There is a clear need to understand the end-to-end performance of a TCP flow in a diff-serv network that is characterized by per-hop-behaviors (PHBs).

Many studies proposed throughput models for TCP flows by characterizing the behavior of the congestion avoidance schemes [21, 24, 25, 30]. With these models, we can analyze and predict TCP throughput in networks when packets are not marked or treated differently within the network.

The objective of this chapter is to propose steady-state throughput models for TCP flows in a differentiated services network as a function of reservation rates, packet drop rates and round-trip times. It is expected that such a model provides an insight into the end-to-end performance of TCP flows in a diff-serv network. It is also expected that such a model may lead to improvements/modifications of the basic PHBs in order to realize end-to-end performance goals. Some of the questions we expect to answer are: (a) how is the steady-state TCP throughput related to a
flow’s contract rate? (b) how does the realized bandwidth vary as a function of packet drop rate? (c) what is the realizable bandwidth for a best-effort flow? (d) what is the cut-off contract rate below which contracts can be met? (e) given a steady-state bandwidth goal, what contract rate should a flow request from the network (when these can be different)?

To develop the models, we first characterize the behavior of a TCP flow in a two-drop precedence network, called RIO (RED In/Out) [10, 11]. Then, we extend this model to a network with three-drop precedences [13, 14, 28]. To complete our study, we extend the models to aggregated marking schemes.

This chapter makes the following significant contributions: (1) we present a simple analytical model for the steady-state throughput of a TCP flow in a differentiated services network, (2) we present extensive simulations to validate the model in different scenarios and different network conditions, (3) we derive the throughput of an individual flow within an aggregation when the aggregated source employs proportional marking, and (4) we show that the analytical model provides intuitive understanding of TCP behavior in a differentiated services network. The developed model shows that a TCP flow cannot always achieve its contract rate because of the sawtooth behavior in congestion avoidance.

The rest of this chapter is organized as follows: Section A proposes and derives our models and presents simulations to validate the models. In Section B, we extend our models to aggregated flows. Section C discusses our models and presents the summary of related studies on TCP modeling and differentiated services. In Section D, we summarize this chapter.
A. TCP modeling in a DS network

In this section, we present models for TCP throughput in a DS network and illustrate the difference between the achieved throughput of a TCP flow and its contract rate quantitatively. The models are for two-drop precedence and three-drop precedence as described in the previous section. We assume that packets are dropped randomly rather than in bursts since the packets are randomly picked to be discarded in RED routers when the average queue length of the router is less than a certain threshold. TCP models for regular networks have considered bursty losses [25] and random losses [24]. Our assumption of random losses may not hold in all the situations in a diff-serv network. The implications of this assumption are discussed in Section C. We define IN/OUT packet loss probability, \( p_{\text{in}} \) and \( p_{\text{out}} \) as the ratio of IN/OUT packets dropped by the number of IN/OUT packets sent. Our models are based on the IN/OUT packet loss rates observed by individual flows.

1. Packet marking with TSW rate estimator

While deriving our models in the following sections, we consider that a flow’s sending rate is estimated by a TSW rate estimator for packet marking. In this section, we discuss packet marking with TSW rate estimator. Fig. 14 shows the TSW algorithm proposed in [11]. It is shown that the TSW marker remembers the history of the past \( \text{win.len} \) interval and smoothes out the TCP’s burstiness. \( \text{win.len} \) is recommended to be in the order of an RTT in [11]. A marker marks a packet IN when this \( \text{avg.rate} \) is less than the contract rate. When the \( \text{avg.rate} \) is greater than the contract rate, a packet is marked IN with the probability, \( \text{contract.rate}/\text{avg.rate} \). The rest are marked OUT.

It is typically known that a TCP source sends packets in a burst within an
Upon each packet arrival:

\[
\text{avg\_rate} = \frac{\text{avg\_rate} \times \text{win\_len} + \text{pkt\_size}}{\text{win\_len} + \text{now} - \text{last\_arrival}}
\]

\[
\text{last\_arrival} = \text{now}
\]

\text{win\_len} : a constant

\text{avg\_rate} : a flow's estimated sending rate

\text{pkt\_size} : the packet size of the arriving packet

Fig. 14. TSW algorithm

RTT interval. More precisely, a TCP source sends packets until the total number of unacknowledged packets is \(cwnd^1\) (congestion window) and waits for ACK arrival. The TCP receiver sends an ACK for every one or two packets received, and the TCP sender uses this ACK-clocking to send new data packets. Consequently, a TCP flow's sending rate averaged in each RTT interval is \(cwnd \times \text{pkt\_size}/\text{RTT}\). Here we define \textit{reservation window (rwnd or R)} as

\[
R = \frac{\text{contract\_rate}}{\text{pkt\_size}} \times \text{RTT}
\]  

(3.1)

With \(rwnd\), we can see the ideal packet marking behavior as one of the following two cases:

- When \(cwnd \leq rwnd\): every packet is marked IN.

- When \(cwnd > rwnd\): a packet is marked IN with the probability \(rwnd/cwnd\), and the rest are marked OUT.

It is not feasible, however, to measure RTT accurately, and it is also not feasible to synchronize measuring intervals to packet sending intervals in a TCP source. To

\footnote{In real TCP, window size is in unit of bytes, but we use unit of packets for simplicity.}
observe practical packet marking behavior, we present Fig. 15 observed from a simulation using ns-2 [22]. This figure shows packet marking of a TCP flow: The contract rate is 0.5 Mbps; Packet size is 1KB; Average RTT (including queueing delay) is 0.14 second; No IN packet drops, and OUT packet drop rate is 0.027; Average achieved throughput is 0.69 Mbps. The employed marker is general TSW packet marker with fixed win.len (= 1 second). Here note that rwnd = 0.5Mbps * 0.14sec./1KB = 8.75 packets.

Fig. 15(a) shows packets sent over time with markings. It is observed that cwnd increases by one at each RTT, and that every packet is marked IN until cwnd = 12, which is greater than rwnd. From cwnd = 13, the number of OUT packets in each cwnd increases as cwnd increases. Fig. 15(b) shows packet distribution over time. Simulation time is divided with the average RTT interval (0.14 sec.), and we count the number of IN/OUT packets sent in each interval. Each bar represents the total number of packets sent. The black portion indicates the number of IN packets, and the white portion indicates the number of OUT packets. Here note that each interval may be not exactly synchronized to the individual RTT due to queueing delay variations. This figure also shows that the most packets sent in an interval where the total number of packets sent is less than rwnd are marked IN, and that the number of OUT packets increases as the total number of packets increases.

From the above observations, we confirm that the practical packet marking behavior is approximately the same as the ideal behavior for this scenario. In the rest of this chapter, we assume the ideal marking behavior: (1) Every packet in a cwnd is marked IN when cwnd is less than rwnd. (2) A packet is marked IN with the

---

2In Fig. 15(b), the black portion is shown in bottom, and the white portion is shown on top. This is only for visual comparison of number of IN/OUT packets and does not mean that packets are marked IN first.
(a) Individual packet marking

(b) IN/OUT packet distribution

Fig. 15. Microscopic view of packet marking
probability \( rwnd/cwnd \), and the rest are marked OUT when \( cwnd \) is greater than \( rwnd \).

2.  Modeling for two-drop precedence

In a steady state, a flow going through a differentiated services network can experience different levels of congestion based on its contract rate and the network dynamics. A flow that experiences no IN packet drops \( (p_{in} = 0) \) is said to observe an *undersubscribed path*. A flow that does not transmit any OUT packets either because every OUT packet is dropped \( (p_{out} = 1) \) or because the sending rate is less than the contract profile is said to observe an *oversubscribed path*. We develop separate models for these two situations and combine these two to model a general situation where a flow may experience a non-zero drop rate for IN and OUT packets. It is to be noted that a single network may be classified differently by the nature of packet drops experienced by individual flows. Different routers may observe different levels of congestion and may drop IN and OUT packet at the same time. The general model is applied in such situation.

While we develop the models, we consider the average round-trip time (RTT) and the average packet size of a flow. We assume that there are no ACK drops in the network and that window size is not limited by the advertised window of the receiver. Our model relies heavily on earlier work in [25].

a.  A flow through an undersubscribed path

We assume that a flow through the undersubscribed path does not observe any IN packet drop. The steady state TCP throughput is defined as

\[
B = \frac{\text{Total number of packets sent} \times k}{\text{Total transmission time}}
\]  

(3.2)
where $k$ is a packet size. It is noted that, strictly, $B$ means the sending rate rather than throughput. We assume that impact of packet loss in throughput is negligible. It has been similarly assumed in [25].

We define a “period”, $A_i$, as the time between immediately after one packet drop and just before next packet drop. $N_i$ is defined as the number of packets sent in period $A_i$. In the steady state, we can assume that the number of packets sent, $N$, in each period is the same. Similar assumption has been made for modeling TCP throughput in [21, 24, 25, 30]. Then, we can express the throughput as

$$B = \frac{E[N] \times k}{E[A]} \tag{3.3}$$

To derive $N_i$ and $A_i$, we define $W_i$ as the window size at the end of period $A_i$ and $X_i$ as the number of rounds in period $A_i$ as shown in Fig. 16. To model delayed-ACK, we assume that $d$ packets are acknowledged by one ACK as in [25]. Then, we have

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{d} \tag{3.4}$$
since the window is increased by one at every $d$ rounds. Now, $N_i$ is expressed by

$$N_i = \sum_{j=0}^{x_i/d-1} \left( \frac{W_i-1}{2} + j \right)d + \beta_i$$

(3.5)

$$= \left\{ \frac{X_iW_i-1}{2d} + \frac{X_i(X_i - 1)}{d} \right\}d + \beta_i$$

(3.6)

$$= \frac{X_i}{2} \left( \frac{W_i-1}{2} + W_i - 1 \right) + \beta_i$$

(3.7)

where $\beta_i$ is the number of packets sent in the last round. We define $N_{out(i)}$ and $N_{in(i)}$ as the number of packets sent marked OUT and IN in each period, respectively. Then, the mean of $N_i$ is given by

$$E[N] = E[N_{out}] + E[N_{in}]$$

(3.8)

If $p_{out}$ is the probability that an OUT packet is dropped, then the probability that $i^{th}$ OUT packet is dropped is given as

$$P[N_{out} = i] = (1 - p_{out})^{i-1}p_{out}$$

(3.9)

Then, the mean of $N_{out}$ is

$$E[N_{out}] = \sum_{i=1}^{\infty} i(1 - p_{out})^{i-1}p_{out} = \frac{1}{p_{out}}$$

(3.10)

To derive $E[N_{in}]$, we consider two scenarios shown in Fig. 17. In Fig. 17(a), $R \geq W/2$ and as a result of an OUT packet drop, sending rate has fallen below the reservation rate. In Fig. 17(b), $R < W/2$ and even after an OUT packet drop, the sending rate remains above the contract rate. Then, $N_{in(i)}$ is given by

$$N_{in(i)} = \sum_{j=0}^{x_i/d-1} \min\left\{ \frac{W_i-1}{2} + j, R \right\}d$$

(3.11)

$$= \begin{cases} 
X_iR & \text{if } R \leq W_i-1/2 \\
\sum_{j=0}^{(x_i/d-W_i+R-1)} \left( \frac{W_i-1}{2} + j \right)d + dR(W_i - R) & \text{otherwise} 
\end{cases}$$

(3.12)
After algebraic manipulations, we have

\[
N_{\text{in}(i)} = \begin{cases} 
 dR(W_i - \frac{W_{i-1}}{2}) & \text{if } R \leq \frac{W_{i-1}}{2} \\
 d\{RW_i - \frac{R}{2}(R + 1) - \frac{W_{i-1}}{8}(W_{i-1} - 2)\} & \text{otherwise}
\end{cases}
\]  

(3.13)

From (3.7) and (3.8), we have

\[
E[N] = E[N_{\text{out}}] + E[N_{\text{in}}] + E[\beta] = \frac{E[X]}{2}(\frac{E[W]}{2} + E[W] - 1) + E[\beta]
\]  

(3.14)

When \( R \leq E[W]/2 \), from (3.10) and (3.13), we have

\[
E[N] = \frac{1}{p_{\text{out}}} + \frac{E[W]d}{2}R + E[\beta] = \frac{E[W]d}{4}(\frac{E[W]}{2} + E[W] - 1) + E[\beta]
\]  

(3.15)

From (3.15), it follows that

\[
E[W] = \frac{2R + 1}{3} + \frac{4}{3}\sqrt{\frac{R^2}{4} + \frac{3}{2d_{\text{out}}}} + \frac{1}{16} + \frac{1}{4}R
\]  

(3.16)
For small values of \( p_{\text{out}} \) and large values of \( R \), it is observed that

\[
E[W] \approx \frac{2R + \sqrt{4R^2 + \frac{2A}{d_{\text{out}}}}}{3}
\]  
(3.17)

When \( R > E[W]/2 \), we also have

\[
E[N] = \frac{1}{p_{\text{out}}} + d\{E[W]R - \frac{E^3[W]}{8} - \frac{R^2}{2} - \frac{1}{2}(R - \frac{E[W]}{2})\} + E[\beta]
\]  
(3.18)

\[
= \frac{E[W]d}{4}(\frac{E[W]}{2} + E[W] - 1) + E[\beta]
\]  
(3.19)

From (3.19), it follows that

\[
E[W] = R + \frac{1}{2} + \sqrt{\frac{2}{d_{\text{out}}} + \frac{1}{4}}
\]  
(3.20)

For small values of \( p_{\text{out}} \) and large values of \( R \), it is observed that

\[
E[W] \approx R + \sqrt{\frac{2}{d_{\text{out}}}}
\]  
(3.21)

Now, we consider \( E[A] \) in (3.3). From Fig. 16, \( E[A] \) is given by

\[
E[A] = (E[X] + 1)_{\text{RTT}}
\]  
(3.22)

\[
= \left(\frac{E[W]d}{2} + 1\right)_{\text{RTT}}
\]  
(3.23)

We assume that \( \beta_i \) is uniformly distributed from 1 to \( W_i \) and that \( X_i \) and \( W_i \) are i.i.d. random variables as in [25]. Then we have \( E[\beta] = E[X] = E[W]/2 \). Thus, from (3.3), (3.7) and (3.23) the throughput of a TCP flow is expressed as follows,

\[
B = \frac{(\frac{3d}{8}E[W] + \frac{2-d}{4})E[W]k}{(\frac{dE[W]}{2} + 1)_{\text{RTT}}}
\]  
(3.24)

where \( E[W] \) is given by (3.16) and (3.20).

Now, we extend this to include time-outs. When a packet loss is detected by a time-out, the sender does not send packets until the time-out. Thus, to consider
time-outs, (3.3) is extended to

$$B = \frac{\left\{ \left( \frac{2d}{8} E[W] + \frac{2-d}{4} E[W] + P_{TO} E[N_{TO}] \right) k \right\} - E[W]}{(\frac{dE[W]}{2} + 1)RTT + P_{TO} E[T_{TO}]}$$ (3.25)

Here, $P_{TO}$ and $T_{TO}$ represent the probability that packet loss is detected by a time-out and the time duration taken for detecting a time-out, respectively. $N_{TO}$ is the number of packets sent at a time-out period and typically one or two$^3$.

From [31], time-out in TCP-Reno occurs (1) when two packets are dropped and the congestion window is less than 10 packets, or (2) when three or more packets are dropped within a window and the number of packets between the first and the second packet drops is less than $2 + 3W/4$. In this chapter, we will make the simplifying assumption that three or more packet drops result in a time-out. The probability that there are three packet drops and the number of packets between the first and the second packet drops is greater than $2 + 3W/4$ is very low. Then, $P_{TO}$ is the probability that three or more packets are dropped when there is at least one packet drop. Thus, $P_{TO}$ is expressed as,

$$P_{TO} = \frac{1-(1-p)E[W]-R-(E[W]-R)p(1-p)E[W]-R-1}{1-(1-p)E[W]-R} - \frac{E[W]-R}{1-(1-p)E[W]-R}$$ (3.26)

To model exponential backoff in time-out, we manipulate $E[T_{TO}]$ in (3.25). Exponential backoff is a scheme to avoid serious congestion by doubling waiting time for retransmission when consecutive packet losses occur. The waiting time for retransmission is doubled until six consecutive packet losses. Thus, $T_{TO}$ is given by

$$T_{TO} = \begin{cases} 2^l T_0 & \text{for } 0 \leq l \leq 6 \\ 64 T_0 & \text{for } l > 6 \end{cases}$$ (3.27)

$^3$In [25], $E[N_{TO}]$ is given by $1/(1 - p)$ where $p$ is the drop probability. If $p$ is less than 0.5, $E[N_{TO}]$ is in between 1 and 2. Refer [25] for detail.
where $T_0$ is the time taken to detect the first time-out, and $l$ is the number of consecutive packet losses. The probability of more than six consecutive packet losses is very low, and we ignore it for simplicity. Then, the mean of $T_{TO}$ is

$$E[T_{TO}] = \frac{\sum_{i=0}^{6} (2p_{out})^l - T_0}{\sum_{i=0}^{6} I_{p_{out}}}$$

$$= \frac{(1 - (2p_{out})^7)(1 - p_{out})}{(1 - 2p_{out})(1 - p_{out})}T_0$$

(3.28)

(3.29)

b. A flow through an oversubscribed path

A flow through the oversubscribed path observes IN packets loss as well. There are two possible situations: (1) Every packet is marked as IN when the sending rate is less than the contract profile ($W_i < R$). (2) Whenever a packet is marked as OUT, the packet is dropped ($W_i = R$). In both the situations, OUT packets are not transmitted. Thus, the total number of packets sent in each period, $N_i$, is given by

$$N_i = N_{in(i)}$$

(3.30)

Here note that we ignore one OUT packet sent when $W_i = R$. Then, $E[N_{in}]$ is calculated with similar ways in (3.10) as following

$$E[N_{in}] = \min(\sum_{i=1}^{\infty} i(1 - p_{in})^{i-1} p_{in}, \frac{3d}{8}R^2 + \frac{d}{4}R) + E[\beta]$$

$$= \min(\frac{1}{p_{in}}, \frac{3d}{8}R^2 + \frac{d}{4}R) + E[\beta]$$

(3.31)

(3.32)

Then, from (3.14), (3.32) and (3.30) we have

$$E[N] = E[N_{in}] = \frac{3d}{8}E[W] + \frac{d}{4}E[W] + E[\beta] = \min(\frac{1}{p_{in}}, \frac{3d}{8}R^2 + \frac{d}{4}R) + E[\beta]$$

(3.33)

$$E[W] = \min(\sqrt{\frac{1}{9} + \frac{8}{3dp_{in}}} - \frac{1}{3}, R)$$

(3.34)
Applying (3.34) to (3.25), we can model TCP throughput in the oversubscribed path where $P_{TO}$ and $E[T_{TO}]$ are given by

$$P_{TO} = 1 - (1 - p_{in})^{E[W]} - E[W] p_{in} (1 - p_{in})^{E[W]-1} - \frac{E[W](E[W]-1)}{2} p_{in}^2 (1 - p_{in})^{E[W]-2} \over 1 - (1 - p_{in})^{E[W]}$$  \hspace{1cm} (3.35)

$$E[T_{TO}] = \frac{(1 - (2p_{in})^7)(1 - p_{in})}{(1 - 2p_{in})(1 - p_{in})^7} T_0$$  \hspace{1cm} (3.36)

\(c\). Combined model

During simulations, we observed that in some situations both IN and OUT packets are dropped. In such cases, we cannot apply either of the earlier models. In this section, we combine the models for undersubscribed and oversubscribed paths so that our model can fit this general situation.

To combine the two models, we extend the definition of period as follows: (1) Undersubscribed period $A_u$ is defined as a period ended by an OUT packet drop, and let $B_u$ be throughput achieved in $A_u$ and $p_{out}'$ be the probability of an OUT packet loss in such a period. (2) Oversubscribed period $A_o$ is defined as a period ended by an IN packet drop, and let $B_o$ be throughput achieved in $A_o$ and $p_{in}'$ be the probability of an IN packet loss in such a period. We assume that IN and OUT packet losses are random and not correlated each other. Then, average throughput, $B$ in steady state is

$$B = Q_u B_u (p_{out}') + Q_o B_o (p_{in}')$$  \hspace{1cm} (3.37)

where $Q_u$ and $Q_o$ is the probability that $A_u$ and $A_o$ occur, respectively, and $Q_u + Q_o = 1$. From the definition of $A_u$ and $A_o$, $Q_u$ is the ratio of the number of OUT packet losses and the total number of losses, and $Q_o$ is the ratio of the number of IN packet
losses and the total number of losses.

\[
Q_u = \frac{(1 - p_m)p_{\text{out}}}{(1 - p_m)p_{\text{out}} + p_m p_{\text{in}}} \quad (3.38)
\]

\[
Q_o = \frac{p_m p_{\text{in}}}{(1 - p_m)p_{\text{out}} + p_m p_{\text{in}}} \quad (3.39)
\]

where \( p_m \) is the probability that a packet is marked IN. Since we assume that there is no OUT packet transmitted in \( A_o \), \( p'_{\text{out}} \) is equal to \( p_{\text{out}} \). Then, we can directly use (3.25) for \( B_u \) in (3.37).

\( p'_{\text{in}} \) is not equal to \( p_{\text{in}} \) since \( p'_{\text{in}} \) is the IN packet loss rate from IN packets sent in \( A_o \) while \( p_{\text{in}} \) is the IN packet loss rate from IN packets sent in both \( A_o \) and \( A_u \). For calculating \( p'_{\text{in}} \), we divide packets into three groups, OUT packets sent in \( A_u \), IN packets sent in \( A_u \), and IN packets sent in \( A_o \). Note that there is no OUT packet sent in \( A_o \). Then, we have

\[
E[N] = Q_u (E[N_{\text{out}}] + E[N_{\text{in}}']) + Q_o E[N_{\text{in}}'] \quad (3.40)
\]

where \( N_{\text{out}}, N_{\text{in}}' \) and \( N_{\text{in}}'' \) are the number of OUT packets sent in an \( A_u \), the number of IN packets sent in an \( A_u \), and the number of IN packets sent in an \( A_o \), respectively.

From the definition of \( p_m \),

\[
p_m = \frac{Q_u E[N_{\text{in}}'] + Q_o E[N_{\text{in}}'']}{E[N]} \quad (3.41)
\]

From (3.40) and (3.41), and using \( E[N_{\text{out}}] = 1/p_{\text{out}}, E[N] \) is expressed by

\[
E[N] = \frac{Q_u}{1 - p_m} E[N_{\text{out}}] \quad (3.42)
\]

\[
= \frac{1}{(1 - p_m)p_{\text{out}} + p_m p_{\text{in}}} \quad (3.43)
\]

---

\(^4\)This assumption may not hold when the network changes quickly. However, when a network is stabilized, and thus queue length in RIO routers does not change quickly, OUT packet should be discarded first, and the assumption may be protected.
Now, $p'_m$ is

$$p'_m = \frac{Q_u E[N^u_m] + Q_o E[N^o_m]}{E[N] - Q_u E[N_{out}]} p_{in}$$

(3.44)

$$= \frac{E[N] - Q_u E[N_{out}]}{E[N] - Q_u (E[N_{out}] + E[N^u_m])} p_{in}$$

(3.45)

$$= \frac{p_m p_{in}}{1 - p_{out} (1 - p_m) (E[N_{out}] + E[N^u_m])}$$

(3.46)

Here, note that $(E[N_{out}] + E[N^u_m])$ is the total number of packets sent in an $A_u$. We can reasonably assume that $R > E[W]/2$ since both IN and OUT packets are dropped (as in Fig. 17(a)). Therefore, from (3.19) and (3.21), $(E[N_{out}] + E[N^u_m])$ is calculated as follows,

$$E[N_{out}] + E[N^u_m] = \frac{3d}{8} (R + \sqrt{\frac{2}{dp_{out}}})^2 + \frac{2 - d}{4} (R + \sqrt{\frac{2}{dp_{out}}})$$

(3.47)

$$= \frac{3d}{8} R^2 + (3d \sqrt{\frac{2}{dp_{out}} - d}) \frac{R}{4} + \frac{3d}{4p_{out}} + \frac{2 - d}{4} \sqrt{\frac{2}{p_{out}}}$$

(3.48)

Note that $E[\beta]$ in (3.19) is given by $1/2E[W]$. Applying (3.46) to the model for oversubscribed path, we can finally compute (3.37).

d. Simulations

In this section, we validate our models through simulations. First, we use a simple network topology which has one bottleneck link so that we can understand the results of the models intuitively. Then, we use complex topologies to show that the models can model TCP flows in various network conditions. We also compare the results with the result of the simple model based on (3.17) and (3.21) instead of (3.16) and (3.20). The simple models are useful to observe and explain throughput of a TCP flow intuitively.

In the simulations, we use Network Simulator version 2 (ns-2) [22]. Our diff-serv implementation is validated in [28, 32]. TCP-Reno source is used for senders.
Fig. 18. A simple network topology

Receivers do not employ delayed-ACK \((d = 1)\) for simplicity. For droppers, we use RIO presented in [11] with parameters \((\text{minTh}/\text{maxTh}/\text{pmax})\) 20/40/0.5 for OUT packets and 40/80/0.02 for IN packets. We ran each simulation for five minutes and collected statistics after one minute in order to avoid data from transient state. One minute would be enough for a flow to reach steady state since RTT of a flow is set to less than one second in every simulation.

At first, we ran four simulations with a simple network topology shown in Fig. 18. In each simulation, we set the bottleneck link bandwidth to different level so that the network experienced different subscription levels. The total contract rate of flows is 25 Mbps in each simulation. We set the bottleneck bandwidth to 40, 30, 20 and 15 Mbps. The contract rate of an individual flow is randomly picked from 0 to 1 Mbps.

The loss rates of IN/OUT packets in each simulation are presented in Table IV. ‘Max’ and ‘Min’ rows show the maximum and minimum loss rate among 50 flows, respectively, and ‘Mean’ shows the average loss rate of the 50 flows. It is clearly
Table IV. Loss rates of IN/OUT packet

<table>
<thead>
<tr>
<th>Bottleneck</th>
<th>40 Mbps</th>
<th>30 Mbps</th>
<th>20 Mbps</th>
<th>15 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss rate</td>
<td>$P_{in}$</td>
<td>$P_{out}$</td>
<td>$P_{in}$</td>
<td>$P_{out}$</td>
</tr>
<tr>
<td>Max</td>
<td>0</td>
<td>0.0233</td>
<td>0</td>
<td>0.0645</td>
</tr>
<tr>
<td>Min</td>
<td>0</td>
<td>0.0149</td>
<td>0</td>
<td>0.0366</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
<td>0.0186</td>
<td>0</td>
<td>0.0485</td>
</tr>
</tbody>
</table>

observed that the simulations with 40, 30 and 20 Mbps link bandwidth are operating as the undersubscribed path. It is to be noted that and the total contract rate of 25 Mbps exceeds the link bandwidth of 20 Mbps, the packet losses indicate that we should apply the undersubscribed model in this situation. The OUT packet loss rates vary over a wide range from 1.49% to 30.56%. In the simulation with 15 Mbps bottleneck link, a few IN packets are dropped even though some OUT packets are transmitted.

Fig. 19 and 20 compares throughput observed in simulations with the estimated throughput from our models. In the figures, the square and the star indicate the simulated throughputs and the estimated throughput, respectively. ‘x’ is the estimation using the simple models. The dashed line shows achieved rate = contract rate.

Fig. 19(a), 19(b) and 20(a) show simulations in undersubscribed path. In Fig. 20(a), we applied the undersubscribed model even though the bottleneck bandwidth is less than the total contract rate since no IN packet-drops were observed. It is clear that the original model for undersubscribed path estimates the individual throughput achieved by both IN and OUT packets quite accurately. It is observed that the estimation of the simple model is little higher than the simulations and the estimations of the original model. With 15 Mbps link capacity, we applied the combined model since there are both IN and OUT packet drops. Note that the throughput achieved only by IN packets is not presented in Fig. 20(b) since most throughput
is achieved by IN packets. It is shown that the combined model can estimate TCP throughput in a simple network topology very accurately.

Fig. 21 shows the relative error between the estimations and the simulations. There are 200 samples from four simulations. It is observed that the relative errors of 93% of samples are less than 10%, and that the maximum error is less than 25%. From Fig. 19, 20 and 21, we confirm that the models are very accurate in a single bottleneck link network.

To observe TCP throughput with cross traffic, we conducted a simulation with a network topology as shown in Fig. 22(a). In this experiment, link delay and capacity of each link are set to 10 ms and 30 Mb, respectively. The contract rates of 30 TCP flows are randomly picked from 0 to 1 Mbps, and the total contract rates add up to 15 Mbps. At each switch, cross traffic consists of five exponential on/off sources with different on/off periods. The average sending rate of each source is 2 Mbps, and thus the sending rate of cross traffic at each switch is 10 Mbps. Each cross traffic stream is injected into $R_i$ and leaves at $R_{i+1}$. Fig. 22(b) shows the results. The drop rate of each flow is calculated by the total number of drops divided by the total number of packets sent from the source. This simulation shows that our model can be used to estimate throughput of a flow going through multiple links.

We conducted two other simulations with a merged topology and a split topology shown in Fig. 23(a) and 24(a). In the merged topology, flows are merged three times, and it is expected that flows will experience different levels of congestion. In the split topology, flow 1 to 10 and 11 to 20 reach $R_4$ and $R_5$, respectively, through $R_1$ and $R_3$, and flow 21 to 30 and 31 to 40 reach $R_4$ and $R_5$ through $R_2$ and $R_3$. Different capacity and delay characteristics are assigned to each link. The link capacity and delay of each link are shown in the figures. The contract rate of each flow is set to 1 Mbps in both the simulations. Fig. 23(b) and 24(b) show the results. In these
(a) 40 Mbps bottleneck link

(b) 30 Mbps bottleneck link

Fig. 19. Throughputs of TCP-Reno flows in 40/30 Mbps links
Fig. 20. Throughputs of TCP-Reno flows 20/15 Mbps links
Fig. 21. Relative error between simulation and estimation

simulations, it is observed that both IN and OUT packets are dropped. Hence we apply the combined model. It is shown again that the model keeps track of the different achieved rates under different RTTs and different link capacities.

The simulations presented in this section show that (1) The original models based on (3.16) and (3.20) are quite accurate in estimating individual TCP throughput in diff-serv networks. (2) The simple models based on (3.17) and (3.21) are also accurately estimating TCP throughput. (3) The models work in multiple-link networks, merged and split network topologies with different levels of congestion. In the rest of this chapter, we use the simple models instead of the original models since the simple models are easy to understand and give intuitive insight of TCP throughput in diff-serv networks.
(a) Simulation topology

(b) Throughputs

Fig. 22. Simulation with cross traffic
(a) Simulation topology

(b) Throughputs

Fig. 23. Simulation with merged network topology
(a) Simulation topology

(b) Throughputs

Fig. 24. Simulation with split network topology
e. Throughput analysis

In this section, we discuss the excess bandwidth $B_e$ which is defined as the difference between realized throughput and its contract rate. In this discussion, we focus on the flows through the undersubscribed path and use the simple models for undersubscribed path without time-out consideration for simplicity. The contract rate is expressed by ($k \times R$/RTT) from (3.1). Then, the excess bandwidth is

\[
B_e = \begin{cases} 
\frac{k}{4\text{RTT}}(3\sqrt{\frac{2}{p_{out}}} - R) & \text{if } R \geq E[W]/2 \\
\frac{k}{4\text{RTT}}(\sqrt{R^2 + \frac{6}{p_{out}}} - R) & \text{otherwise}
\end{cases}
\]  

(3.49)

If $B_e$ of a flow is positive, that means the flow obtains more than its contract rate. Otherwise, it does not reach its contract rate. Fig. 25 shows excess bandwidth achieved by an individual TCP flow in different conditions. We also present Fig. 26 to illustrate the following observations.

From (3.49), Fig. 25 and 26, we can observe that:

- When a flow reserves relatively higher bandwidth ($R \geq \sqrt{2/p_{out}}$), $B_e$ is decreased as the reservation rate is increased. Moreover, if $R$ is greater than $3\sqrt{2/p_{out}}$ (see line C in Fig. 26), the flow cannot reach its reservation rate.

- When a flow reserves relatively lower bandwidth ($R < \sqrt{2/p_{out}}$, see line B in Fig. 26), it always protects at least its reservation rate. As it reserves less bandwidth, it obtains more excess bandwidth. These observations correspond to the simulated results in [28]. TCP’s multiplicative decrease of sending rate after observing a packet drop results in a higher loss of bandwidth for flows with higher reservations. This explains the observed behavior.

- The above equation also shows that as the probability of OUT packet drop decreases, the flows with smaller reservation benefit more than the flows with
Fig. 25. Achieved excess bandwidth
- A: $B = \text{Contract rate}$

- B: $R = \sqrt{\frac{2}{P_{\text{out}}}}$

- C: $R = 3\sqrt{\frac{2}{P_{\text{out}}}}$

- D: $B = \frac{k}{2RT} \sqrt{\frac{6}{P_{\text{out}}}}$

Fig. 26. Observations from the model
larger reservations. This again validates the difficulty in providing service differ-
entiation between flows of different reservations observed in [28] when there
is plenty of excess bandwidth in the network.

- The realized bandwidth is observed to be inversely related to the RTT of the
  flow.

- For best-effort flows, $R = 0$. Hence, $B_c (= k\sqrt{6/\text{pout}}/2\text{RTT}, \text{see line D in
  Fig. 26})$ gives the bandwidth likely to be realized by flows with no reservation.

- Comparing the above best-effort bandwidth and when $R \geq \sqrt{2/\text{pout}}$, we realize
  that the reservation rates larger than 3.5 times the best-effort bandwidth cannot
  be met.

- The above equation (3.49) clearly shows that excess bandwidth cannot be
  equally shared by flows with different reservations (a goal of recent simulation
  studies [15, 28]) without any enhancements to basic RIO scheme or to TCP’s
  congestion avoidance mechanism.

3. Modeling for three-drop precedence

In this section, we extend the models for two-drop precedence to the models for three-
drop precedence. Three-drop precedence policy was proposed as an extension of two-
drop precedence in [13, 14]. In a three-drop precedence network, the edge devices
mark a packet as one of green, yellow or red depending on the sending rate and the
reservation rate for each color. Generally, it is recommended that the reservation rate
for yellow is set to be greater than or equal to the reservation rate for green. If the
current sending rate is less than the reservation rate for green, the packet is marked
as green. If the sending rate is greater than the reservation for green but less than
the reservation for yellow, the packet is marked as yellow. Otherwise, the packet is marked as red. The core routers provide differentiation by dropping red packets first, yellow packets second and then the green packets. It is expected that three color marking and dropping policies give better control in realizing performance goals.

In the three-drop precedence network, there may be three possible subscription levels: (1) Only marked red packets are dropped (Fig. 27 (a)), (2) Every marked red packet is dropped, and some of marked yellow (Fig. 27 (b)) packets are dropped, and (3) Every marked red and yellow packets are dropped, and some of marked green packets are dropped (Fig. 27 (c)). To develop the models, we define $p_{\text{color}}$ and $R_{\text{color}}$ as the drop probability and the reservation window of packets marked as that color, respectively.

- When $p_{\text{red}} > 0$ and $p_{\text{yellow}}, p_{\text{green}} = 0$.
  This case is the same as the undersubscribed model of two-drop precedence. A red packet is treated as an OUT packet. and both yellow and green packet are treated as an IN packet. Therefore, we can directly use (3.16), (3.20), (3.25) and (3.26) just after replacing $p_{\text{out}}$ and $R$ to $p_{\text{red}}$ and $R_{\text{yellow}}$, respectively.

- When $p_{\text{red}} = 1$, $1 > p_{\text{yellow}} \geq 0$ and $p_{\text{green}} = 0$. 

Fig. 27. Drop probabilities of three-drop precedence
This case is oversubscribed for yellow packets and undersubscribed for green packets. A red packet is treated as an OUT packet, and a green packet is treated as an IN packet. A yellow packet is considered as an IN packet compared to a red packet and considered as an OUT packet compared to a green packet. Therefore, we combine the undersubscribed model and the oversubscribed model. As the undersubscribed model, we have

\[
E[W] = \begin{cases} 
R_{green} + \sqrt{\frac{2}{dp_{yellow}}} & \text{if } R_{green} > E[W]/2 \\
\frac{2}{3}(R_{green} + \sqrt{R_{green}^2 + \frac{6}{dp_{yellow}}}) & \text{otherwise}
\end{cases} 
\] (3.50)

As the oversubscribed model, \(W_i\) is limited by \(R_{yellow}\). Thus, we also have

\[
E[W] \leq R_{yellow}
\] (3.51)

Therefore, finally, \(E[W]\) is expressed by

\[
E[W] = \begin{cases} 
\min\{R_{green} + \sqrt{\frac{2}{dp_{yellow}}}, R_{yellow}\} & \text{if } R_{green} > E[W]/2 \\
\min\{\frac{2}{3}(R_{green} + \sqrt{R_{green}^2 + \frac{6}{dp_{yellow}}}), R_{yellow}\} & \text{otherwise}
\end{cases} 
\] (3.52)

Now, with (3.52), we can use (3.25) and (3.26) after replacing \(p_{out}\) and \(R\) to \(p_{yellow}\) and \(R_{green}\).

- When \(p_{red}, p_{yellow} = 1\), and \(p_{green} > 0\).

This case is the same as the case in which every OUT packet is dropped in the two-drop precedence. So, (3.34) and (3.35) are applied by changing \(p_{in}\) and \(R\) into \(p_{green}\) and \(R_{green}\), respectively.
a. Simulations

The models derived in the previous section have been extended directly from our two-drop precedence model. It has been already shown that the model can estimate individual TCP throughput in diff-serv network environment through a number of simulations. In this section, we show that the models can be extended to estimate TCP throughput when droppers employ three-drop precedence through a simple network topology.

In the simulations, we used the same topology presented in Fig. 18 except three-drop precedence is employed with parameters 20/40/0.5 for red packets, 40/60/0.1 for yellow packets and 60/80/0.02 for green packets. Three simulations with different bottleneck bandwidths (40, 20 and 10 Mbps each) were conducted to reflect three situations described in the previous section. In each simulation, the contract rate for yellow of each flow is randomly picked from 0 to 1 Mbps, and the contract rate for green is set to a half of the contract rate for yellow of that flow to follow [13] which recommends that the contract rate for green should be set less than the contract rate for yellow. The total reservation rate for yellow is set to 25 Mbps, and the total reservation rate for green is set to 12.5 Mbps. Each simulation ran for five minutes, and we collected statistics in the last four minutes.

Table V shows the loss rate of green/yellow/red packets in each simulation. N/A for \( p_{\text{color}} \) in the table means that no packet is marked as that color at all. The table shows that the simulations reflect the three different cases effectively. Individual throughputs from the simulations and the models are presented in Fig. 28. From Fig. 28(a), it is shown that the simple model for undersubscribed path can estimate throughput accurately in a three-drop precedence network. Fig. 28(b) and 28(c) also show that our models for oversubscribed path can be applied for three-drop
Table V. Loss rates of green/yellow/red packet

<table>
<thead>
<tr>
<th>Bottleneck</th>
<th>40 Mbps</th>
<th>20 Mbps</th>
<th>10 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>$p_{green}$</td>
<td>$p_{yellow}$</td>
<td>$p_{red}$</td>
</tr>
<tr>
<td>Min</td>
<td>0</td>
<td>0</td>
<td>0.096</td>
</tr>
<tr>
<td>Mean</td>
<td>0</td>
<td>0</td>
<td>0.074</td>
</tr>
</tbody>
</table>

precedence.

B. A throughput model for aggregated flows

The diff-serv architecture is being proposed for aggregate reservation. So far, our models have assumed that individual flows can reserve bandwidth. In this section, we develop a throughput model of a individual flow within an aggregated reservation.

1. Packet marking of aggregated flows

In Section 1, we have described packet marking behavior with TSW rate estimator when a marker marks packets belonging to an individual TCP flow. We now discuss the marking behavior when the same marker marks packets from aggregated TCP flows. Fig. 29 shows the conceptual model which we look at in this section.

When a marker is assigned to an individual TCP flow, it is observed that IN packets are distributed more at the beginning of each period due to TCP’s well-known sawtooth behavior. When several (or more) TCP flows are aggregated, the impact of an individual sawtooth behavior is reduced, and the aggregated sending rate is stabilized (even though there still exist some variations). If the marker does neither maintain per-flow state nor employ other specific method for distinguishing individual flows, an arriving packet, therefore, is marked IN with the probability
Fig. 28. Throughputs of TCP-Reno flows with different reservation rate in three-drop precedence network
Fig. 29. Packet marking of aggregated flows

\[(\text{contract rate}/\text{aggregated sending rate})^5\]. Here we define \(p_m\), the probability of a packet marked IN as

\[p_m = \frac{\text{contract rate}}{\text{aggregated sending rate}}\]  \hspace{1cm} (3.53)

In a steady state where the aggregated sending rate is maintained, \(p_m\) is approximately equal for every individual flow. A flow sending more packets then gets more IN packets, and consequently, the contract rate consumed by individual flows is proportional to their sending rates\(^6\). We call this marking behavior proportional marking. In the following section, we propose a throughput model for individual TCP flows sharing a contract rate with a general TSW marker (showing proportional marking behavior), which does not maintain per-flow state nor employ any sophisticated mechanism for

\(^5\)When the aggregated sending rate is less than the contract rate, every packet is marked IN.

\(^6\)This behavior is different from the marking of individual flows in Section A. In the marking of individual flows, the contract rate for an individual flow is fixed. It has been observed that the realized throughput is not proportional to the contract rate. In the marking of aggregated flows, however, the contract rate consumed by an individual flow is not fixed even though the aggregated contract rate is fixed.
aggregated flows.

2. Modeling for throughput of aggregated flows

In this section, we develop a simple model for an individual flow with the aggregate contract rate in a two-drop precedence network. In the model, we assume that all the packets of aggregated flows are of the same size, \( k \), a receiver does not employ delayed-ACK \( (d = 1) \) and the network is not oversubscribed. We define \( r_A \) as the reservation rate which is contracted for the aggregation and \( r_i \) as the reservation rate which is achieved by \( i^{th} \) individual flow.

\[
    r_A = \sum_{i=1}^{n} r_i \quad (3.54)
\]

where \( n \) is the number of flows.

For simplicity, we consider that there is no time-out and \( r_i \) is greater than \( \sqrt{2/p_{out}} \). Then, from (3.1) and the simple model presented in (3.21) and (3.24), the throughput of \( i^{th} \) flow is given by

\[
    B_i = \frac{3k}{4\text{RTT}_i} \left( \frac{\text{RTT}_i}{k} r_i + \sqrt{\frac{2}{p_i}} \right) \quad (3.55)
\]

\[
    = \frac{3}{4} r_i + \frac{3k}{4\text{RTT}_i} \sqrt{\frac{2}{p_i}} \quad (3.56)
\]

where \( p_i \) is the OUT packet drop probability of \( i^{th} \) flow. Then, aggregated throughput \( B_A \) is

\[
    B_A = \sum_{i=1}^{n} B_i \quad (3.57)
\]

\[
    = \sum_{i=1}^{n} \left\{ \frac{3}{4} r_i + \frac{3k}{4\text{RTT}_i} \sqrt{\frac{2}{p_i}} \right\} \quad (3.58)
\]

\[
    = \frac{3}{4} r_A + \frac{3k}{4} \sum_{i=1}^{n} \frac{1}{\text{RTT}_i} \sqrt{\frac{2}{p_i}} \quad (3.59)
\]
When a marker employs proportional marking, $r_i$ is linearly proportional to $B_i$.

$$r_i = p_m \cdot B_i \quad (3.60)$$

where $p_m$ gives the probability that a flow's packet is marked IN. Then, (3.56) is rewritten by

$$B_i = \frac{3p_m}{4} B_i + \frac{3k}{4 \text{RTT}_i} \sqrt{\frac{2}{p_i}} \quad (3.61)$$

$$= \frac{3k}{4 \text{RTT}_i - 3p_m \text{RTT}_i} \sqrt{\frac{2}{p_i}} \quad (3.62)$$

From (3.60), $r_A = p_m \cdot B_A$ and therefore,

$$p_m = \frac{r_A}{B_A} \quad (3.63)$$

$$= \frac{3r_A}{4} + \frac{3k}{4} \sum_{i=1}^{n} \frac{1}{\text{RTT}_i} \sqrt{\frac{2}{p_i}} \quad (3.64)$$

Substituting $p_m$ with (3.64) and after some manipulations, we have

$$B_i = \frac{m_i}{\sum_{j=1}^{n} m_j} \cdot \frac{3r_A}{4} + \frac{3k}{4} m_i \quad (3.65)$$

where $m_i$ is given by

$$m_i = \frac{1}{\text{RTT}_i} \sqrt{\frac{2}{p_i}} \quad (3.66)$$

(3.65) relates the realized bandwidth of an individual flow to the aggregate reservation $r_A$ and the network conditions (RTT and $p_i$) observed by various flows within the aggregation.

From (3.59), $B_e$ (the excess bandwidth) of aggregated flows is calculated as follows,

$$B_e = \frac{3}{4} r_A + B_s - r_A \quad (3.67)$$

$$= B_s - \frac{1}{4} r_A \quad (3.68)$$
where \( B_s = \frac{3k}{4} \sum_{i=1}^{n} \frac{1}{\text{RTT}_i \sqrt{p_i}} \), and it is approximately the throughput which the aggregated flows can achieve without the contract rate \( r_A = 0 \). Based on the above analysis, the following observations can be made:

- The total throughput realized by an aggregation is impacted by the contract rate. Larger the contract rate, the smaller the excess bandwidth claimed by the aggregation (referring to (3.68)).

- When the contract rate is larger than 4 times \( B_s \), the realized throughput is smaller than the contract rate (referring to (3.68)).

- A larger (number of flows) aggregation realizes more throughput than a smaller aggregation with identical contract rates (referring to (3.59)).

- The realized throughput of a flow is impacted by the other flows in the aggregation (as a result of the impact on \( p_m \)) when the proportional marking is employed (referring to (3.64) and (3.65)).

3. Simulations

In this section, we present simulations to validate the model for aggregated flows. To observe behaviors of individual flows within a large aggregation, we conducted two simulations with an aggregation of 50 individual flows. The simulation topology is shown in Fig. 30. The aggregate reservation rate is 10 Mbps, and the bottleneck capacity is set to 20 or 30 Mbps so that the bottleneck is not oversubscribed. The RTT without queueing delay of each flow is randomly picked from 50 to 160 ms.

To confirm the assumption that \( p_m \) is the same for all the flows within an aggregation, we present Fig. 31. The solid lines represent \( p_m \) of aggregations in both simulations, which is the contract rate divided by avg. aggregated sending rate. The
square and the circle show $p_m$ of individual flows, which is measured by the number of IN packets divided by the number of IN and OUT packets sent by that flow. It is clearly shown that individual $p_m$ is approximately equal to aggregated $p_m$, and we confirm that individual flows within an aggregation have the same $p_m$.

Fig. 32 shows the simulated and estimated throughput. We also present relative errors in Fig. 33. It is clearly shown that the model can estimate the individual throughput of aggregated flows very accurately (maximum error is less than 25%). It is observed that the estimated throughputs are slightly higher than the simulation results.

Now, we discuss the impact of the number of flows within an aggregation. From (3.59) it is shown that the aggregated throughput achieved by larger aggregated flows is higher than the aggregated throughput by smaller aggregation with a given contract rate and network. To observe this impact through simulation, we used a network topology shown in Fig. 34. There are seven aggregations, and $i^{th}$ aggregation
Fig. 31. The probability of a packet marked IN

consists of $2^{i-1}$ individual flows. Contract rate for each aggregation is set to 1 Mbps. Bottleneck link is set to 40 Mbps so that the network is not oversubscribed.

Simulation results are shown in Fig. 35. Fig. 35(a) shows throughput achieved by individual flows. Each aggregation is separated by a vertical dashed-line. It is observed that individual throughput is decreased as more individual flows share one aggregated contract rate. Our model is observed to be fairly accurate. Fig. 35 shows the total throughput achieved by each aggregation and the estimated throughput using (3.59). It is clearly shown that the aggregated throughput is increased as more individual flows are aggregated.

We present another simulation to show how the model works in a complicated network topology. Fig. 36 shows the network topology used in the simulations. There are five markers and aggregated sources, each aggregated source consists of ten individual sources. Bandwidth of every link except the link between the two routers is 10 Mbps, and bandwidth of the link between two routers is limited to 8 Mbps. With this topology, we conducted two simulations. In the first simulation, each ag-
(a) Bottleneck link = 20 Mbps

(b) Bottleneck link = 30 Mbps

Fig. 32. Simulations with aggregated flows with different RTTs
Fig. 33. Relative error

Fig. 34. Aggregated network topology with different numbers of individual flows
Fig. 35. Simulations with aggregated flows within different numbers of individual flows
Fig. 36. Aggregated network topology with multiple aggregations

ggregated source reserves 1 Mbps. We assign $RTT^i_j$, RTT of $i^{th}$ individual source in $j^{th}$ aggregated source excluding queueing delay as

$$RTT^i_j = 130 + 4 \times (i - 1) \times (j - 5.5) \text{ (ms)}$$

(3.69)

This results in five aggregated sources with varying differences in RTTs. For example, aggregated source 1 has a (min RTT, max RTT) = (130 ms, 130 ms) compared to that of aggregated source 5 with (58 ms, 202 ms). In the second simulation, RTT distribution of each aggregation is the same as each other and given by

$$RTT^i_j = 130 + 16 \times (i - 5.5) \text{ (ms)}$$

(3.70)

The reservation rate of $j^{th}$ aggregation is given by $0.3 \times j$ Mbps.

Fig. 37 shows the results. Fig. 37(a) shows the results of the first simulation with different RTTs and 37(b) shows the results of the second simulation with different aggregate reservations. It is observed that individual flows achieve different bandwidths
within an aggregation. These differences increase with increased differences in the RTTs of the individual flows. It is shown that the model can estimate the individual throughputs in the network with several aggregated sources.

C. Discussion and related work

While developing the models, we assumed that the packets are dropped randomly. This assumption is valid in undersubscribed networks since a packet is randomly picked to be discarded in a router when queue length is less than $\text{max}_\text{th}$. However, when the queue length builds up, OUT packets may be dropped in a burst. This could be a potential source of error in our models. This will result in an inaccurate estimation of time-outs. However, operational dynamics help from this becoming a significant source of error since (1) a flow through a congested path may not send too many OUT packets, and (2) OUT packets are not sent out in a burst. When all the OUT packets are being dropped, a flow can rarely increase its window to such an extent that it sends out significant number of OUT packets within a window. It is possible to enhance the presented model to consider this correlation to further improve the accuracy.

To understand the possible inaccuracy due to correlated losses, we ran a simulation (with simulation topology in Fig. 18 and 17 Mbps bottleneck link capacity) where the queue length oscillated around $\text{max}_\text{th_out}$. The queue length and the throughput comparison of the model with simulation are shown in Fig. 38. Even in this situation where packet losses are forced to be correlated, the model is within 20-30% of the actual throughput. Deriving a more accurate model that considers this correlation is part of our future work.

After the congestion avoidance scheme for TCP was proposed in [20], several
(a) Different RTT distributions

(b) Different aggregate reservations

Fig. 37. Dealing with multiple aggregations
(a) Queue length

(b) Throughputs

Fig. 38. Simulation with correlated packet losses
models of TCP behavior have been proposed in [21, 25, 30, 31, 24]. Each model estimates the steady state throughput of an individual TCP flow in terms of RTT and the probability of a packet-drop. In [24], a performance model for a TCP flow was proposed with the assumption that the TCP source avoids retransmission time-outs and has a sufficient receiver window. In [21], Floyd et al derived a similar model to the model in [24]. Padhye et al proposed a new TCP throughput model in [25, 30]. The model characterizes not only the behavior of fast retransmission mechanism but also the effect of TCP’s timeout mechanism. Recently, it has been shown that some simplified assumptions and mathematical simplifications of the model in [25] result in errors in estimating individual flow throughputs [33].

A number of studies on quantitative analysis of differentiated services have been recently presented [19, 34, 35, 36]. May et al [36] presented models of packet behavior at a switch as a function of load within a differentiated services network. In [35], Dovrolis et al addressed the issues of packet schedulers for differentiated services. They described the impact of scheduling schemes including WFQ (Weighted Fair Queueing) and EDF (Early Deadline First) and proposed new scheduling schemes for differentiated services. Sahu et al [34] characterized packet behaviors (delay and loss) of TCP flows through Markov analysis. In this study, a stochastic Markov model is used for deriving a model for TCP throughput in a differentiated services network. Our modeling provides a simpler and a more intuitive characterization of the TCP behavior in differentiated services networks. Our model has been recently extended to token-bucket marking in [18]. We have extended our model to consider two-window TCP in [37]. Azeem et al [19] examined the performance problem of TCP flows caused by time-outs in differentiated services network and proposed new marking and shaping schemes to reduce time-outs. Our models have been extended for performance evaluation (average throughput, packet delay and loss rate) of IP
networks [38].

D. Summary

In this chapter, we have derived throughput models of individual TCP flows in a differentiated services network. Our models estimate throughput in terms of RTT, packet drop rates and the reservation rate of a TCP flow. Our models are developed for various conditions in a differentiated services network including two-drop precedence, three-drop precedence and aggregated marking. We presented a number of simulations to validate our models. The simulation results show that the models can predict the throughput of individual and aggregated TCP flows quite accurately in various conditions.

Our model makes the following observations: (a) Flows with larger contract rates are at a relative disadvantage compared to flows with smaller contract rates; (b) Throughputs achieved by flows with larger contract rates may not reach their contract rates due to TCP’s sawtooth behavior; (c) The contract rate shared by aggregated flows is consumed unfairly within the aggregation; (d) TCP’s throughput is impacted by RTT even in a diff-serv network.
CHAPTER IV

DEALING WITH AGGREGATED FLOWS

In the previous chapters, we have studied behaviors of individual TCP flows in the DS network. We also observed that behaviors of individual flows with an aggregated contract rate are different from the behaviors with an individual contract rate. This chapter focuses on aggregated sources and specifically on techniques for achieving specific performance goals of individual flows within an aggregation while adhering to the service contracts.

We address issues of flows with an aggregated service contract and propose simple marking schemes for aggregated flows. In this study, we assume that a customer with an aggregated source employs his/her own marker to manage individual flows within the aggregation. The network provider may monitor and remark packets to ensure compliance of the contract.

The immediate motivation for this work came out of an observation that there may exist serious unfairness within aggregated flows, while the total throughput of the aggregation reaches its target rate. The unfairness can be caused by different round-trip times (RTTs), different link bandwidths, or different levels of congestion experienced by individual flows within the network.

The impact of different RTTs within an aggregated source is illustrated in we present simulation result with network topology in Fig. 32. It is clear that there exists unfair bandwidth sharing within aggregated sources. This unfairness increases as the differences in RTTs increase.

Fair sharing of bandwidth is used here as an example. In general, individual flow requirements will be considered as targets. In the current simulation, the marker shows proportional marking behavior. As a result, the realized
gated source may employ its own marker such that packets of individual flows may be marked differently (based on their QoS goals) while ensuring that the aggregated traffic marking does not violate the contract with the network provider. In the following sections, we present such strategies for marking packets of individual flows to improve QOS for both sending and receiving data.

The rest of the chapter is organized as follows: In Section A, we propose aggregate marking schemes to manage contract rate effectively within an aggregation. Section B presents simulations with the proposed marking schemes. In Section C, we discuss the simulation results and present related work. Section D summarizes this chapter.

A. Aggregate marking for aggressive bandwidth management

We consider maintaining state for each flow within an aggregation at the boundary router. Average sending rate of a flow is maintained as state information for each flow at the marker of the aggregated source. This information is used in balancing resources across the different flows within the aggregation.

If we apply the proportional marking strategy with TSW (Time Sliding Window) proposed by [11] for aggregated sources, the cumulative total sending rate of \( n \) aggregated sources \( B \) is

\[
B = \sum_{i=1}^{n} b_i \quad (4.1)
\]

where \( b_i \) is the individual sending rate of \( i^{th} \) flow. When \( B \) is less than the contract rate \( M^1 \), every packet is marked IN. If \( B \) is greater than \( M \), then a packet is marked

---

\(^1\)In this chapter, contract rate means contracted profile rate for AF traffic between users and network provider and thus, it is interchangable with marking rate.
IN with a probability of $M/B$. Therefore, we have,

$$M = \sum_{i=1}^{n} m_i = \frac{M}{B} \sum_{i=1}^{n} b_i$$

(4.2)

$$m_i = \frac{M}{B} b_i$$

(4.3)

where $m_i$ is the marking rate of $i$th flow. Here note that $M/B$ is the same to every individual flow within the aggregation. Thus, $m_i$ is proportional to $b_i$. The proportional marking has merit in its simple implementation since it does not need to maintain per-flow state. However, it also has two undesirable properties: (1) Contract rate is unfairly distributed within an aggregation, and (2) Marking rate of an individual flow is affected by other flows within the aggregation. Increasing throughput of an individual flow increases $B$ and decreases $M/B$. Although a flow maintains its throughput ($b_i$), its marking rate($m_i$) is reduced causing $b_i$ to decrease, and vice versa.

Now we propose two aggregate marking algorithms, called IN-fair and BW-fair marking. IN-fair marking scheme distributes contract rate to individual flows within an aggregation equally. The IN-fair marker maintains per-flow state and marks a packet by its current sending rate and individual marking rate. The individual marking rate is given by

$$m_i = \frac{M}{n}$$

(4.4)

where $n$ is the number of active flows within an aggregation. It is clear that the individual marking rate is not affected by other flows throughput.

The BW-fair marking realizes equal throughput of individual flows within an aggregation. The marking rate of an individual flow within an aggregation is given inversely proportional to its current sending rate. The individual marking rate in
BW-fair marking is given by

\[ m_i = \frac{M}{B(\bar{b} - \frac{b_i}{b_{max}})} \left(1 - \frac{b_i}{b_{max}}\right) \]  

(4.5)

where \( \bar{b} \) is the mean of \( b_i \) for all the flows, and \( b_{max} \) is the maximum throughput among the flows within the aggregation.

B. Simulations

We will show that the proposed bandwidth management results in improved realization of individual target rates. We modified ns-2 [22] to implement the new marking algorithm. In all simulations, we used a TCP-Reno agent in ns-2 as a source and FTP application as a traffic generator.

1. Dealing with different RTTs

To show how IN-fair marking deals with different RTTs, we conducted the same simulation as in Fig. 36 with IN-fair marking. Fig. 39 compares the individual throughputs achieved with the proportional marking and IN-fair marking. It is clear that the goal of fair sharing within the aggregation is better achieved.

To compare the results quantitatively, we present Table VI. The average throughput of each aggregation is not much different from each other in both schemes even though RTTs of individual sources are different. The row STD shows the standard deviation among the individual rates within an aggregation. It is observed that STD increases significantly with increased RTT differences within an aggregation. The IN-fair marking algorithm achieves significantly smaller variation compared to proportional marking. The row Max/Min compares the maximum and minimum rates realized within an aggregation. Again, it is observed that fairness is considerably
Fig. 39. Dealing with different RTTs within aggregated sources
Table VI. Quantitative comparisons

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT Max/Min</td>
<td>1.32</td>
<td>2.42</td>
<td>3.48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Proportional marking**

<table>
<thead>
<tr>
<th></th>
<th>Average (Kbps)</th>
<th>STD (Kbps)</th>
<th>Max/Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>112.7</td>
<td>14.3</td>
<td>1.41</td>
</tr>
<tr>
<td></td>
<td>110.0</td>
<td>19.3</td>
<td>1.72</td>
</tr>
<tr>
<td></td>
<td>112.2</td>
<td>21.2</td>
<td>2.06</td>
</tr>
<tr>
<td></td>
<td>113.8</td>
<td>44.4</td>
<td>3.83</td>
</tr>
<tr>
<td></td>
<td>114.5</td>
<td>62.1</td>
<td>4.80</td>
</tr>
</tbody>
</table>

**IN-fair marking**

<table>
<thead>
<tr>
<th></th>
<th>Average (Kbps)</th>
<th>STD (Kbps)</th>
<th>Max/Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>116.9</td>
<td>7.7</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>111.4</td>
<td>8.8</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>115.5</td>
<td>12.3</td>
<td>1.39</td>
</tr>
<tr>
<td></td>
<td>113.3</td>
<td>14.8</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td>117.0</td>
<td>25.3</td>
<td>1.82</td>
</tr>
</tbody>
</table>

**BW-fair marking**

<table>
<thead>
<tr>
<th></th>
<th>Average (Kbps)</th>
<th>STD (Kbps)</th>
<th>Max/Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>113.4</td>
<td>4.3</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>115.3</td>
<td>5.2</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>117.4</td>
<td>5.6</td>
<td>1.12</td>
</tr>
<tr>
<td></td>
<td>115.3</td>
<td>6.2</td>
<td>1.17</td>
</tr>
<tr>
<td></td>
<td>112.8</td>
<td>8.8</td>
<td>1.22</td>
</tr>
</tbody>
</table>

Improved with the IN-fair marking algorithm.

We present another simulation using the same topology in Fig. 36. RTTs of all the flows are set to 60 ms. Target rate of $j^{th}$ individual flow in $i^{th}$ aggregation is set to

$$t_j^i = 100 + 2 \times (i - 1) \times (j - 5.5) \text{ (Kbps)}$$

Each flow is assigned IN bandwidth proportional to its target. Fig. 40 shows the results. It is observed that every flow reaches its target rate and that weighted IN-fair marking easily extends to achieve specific performance goals.

2. Aggressive bandwidth management in congested networks

Results from the earlier section showed that IN-fair marking is an effective bandwidth management strategy. Can we extend this idea further to allocate different amounts of IN-profile bandwidth based on the congestion currently being experienced by the flows? This, in effect, is the same problem as before, but in a different context: Can
we effectively manage IN-profile bandwidth to improve performance of specific flows (with longer RTTs or those experiencing congestion) within an aggregation. However, as we will point out later, reallocating IN-profile bandwidth based on congestion may have important consequences on the network service providers.

Fig. 41 shows the network topology used for simulations. There are two aggregated sources, and each aggregated source consists of ten individual sources. Each aggregated source reserves 1 Mbps. The network consists of a 1 Mbps link between the router and node 'A' and 2 Mbps link between the router and node 'B'. Individual sources 1~8 of each aggregated source send packets to node 'A' and individual sources 9 and 10 send packets to node 'B' through the router. In this topology, therefore, the link between the router and node 'A' is 160% subscribed, and the link between the router and node 'B' is 20% subscribed if we assume that each individual source expects to get 0.1 Mbps \((M/\text{number of individual sources})\). The network as a total has enough capacity \((3 \text{ Mbps})\) to support the two aggregated sources \((\text{total reservation of } 2 \text{ Mbps})\). Due to the dynamic nature of flows, one of the links may be oversubscribed as in this example. Each individual source has same RTT as 40 msec. With this simulation topology, we conducted two simulation experiments. In the first simulation, we applied proportional marking to the marker for aggregation 1 and the
IN-fair marking for aggregation 2. In the second simulation, we applied proportional marking for aggregation 1 and the BW-fair marking for aggregation 2.

Fig. 41. Network topology with different congestion levels

Fig. 42(a) shows the results of Simulation 1. Each bar shows the average throughput of individual sources, and the dark portion in each bar indicates the throughput achieved by IN packets. In proportional marking, it is observed that each packet is marked OUT with the same probability even if its source cannot reach its target rate, and IN packets are unfairly distributed to flows achieving higher rates. In IN-fair marking, however, each individual source shares IN packet throughput equally even with different levels of congestion. Clearly, fair sharing of IN-profile bandwidth improved the performance of the flows through the congested link. Both the aggregated sources stay within the contract-profile, but the second source achieved a higher bandwidth through the congested link than the first source. This is a result of managing IN-profile bandwidth effectively by distributing it fairly among the individual sources.
When source 1 and source 2 compete for bandwidth on the congested link, source 2 achieves higher share due to marking higher number of packets IN.

Fig. 42(b) shows the results of Simulation 2. The BW-fair marking is more aggressive by the fact that it sends more IN packets on congested links than on uncongested links so as to get more bandwidth in congested links. Again, aggregated source 1 used proportional marking and aggregated source 2 used the BW-fair marking. In proportional marking, the flows through congested link in Fig. 42(b) loose more bandwidth than the flows in Fig. 42(a). The flows using the BW-fair marking get more bandwidth than the flows using the IN-fair marking.

The BW-fair marking is more aggressive than the IN-fair marking in trying to meet the performance goals. The goal here is to achieve 0.1 Mbps for each individual source while staying within the contract-profile. As can be seen from Fig. 42(b), the BW-fair marking algorithm allocates more IN-profile bandwidth to sources observing congestion than the ones that are not experiencing congestion. As a result, these sources claim a larger share of the congested link bandwidth, exceeding the individual targets of 0.1 Mbps. The flows within aggregated source 1 achieve significantly less bandwidth due to proportional marking. These two experiments show that individual marking strategies employed by customers can impact each other even when every source stays within the contract-profile.

3. Scalability

From the previous experimental results, it is clear that the proposed marking algorithms reduce the impact of differences in RTTs among the flows within an aggregation and achieve better throughput with the same reserved rate in congested links. However, this improvement in throughput is caused not by improvement of network performance nor increase of resources but by aggressive behavior of sending more IN
Fig. 42. Bandwidth comparison with different marking schemes
packets through the congested links. How well do these algorithms and allocated benefits scale if every source employs the proposed algorithm?

This section discusses the impact of the proposed marking algorithms on entire networks. To observe the scalability of these strategies, we conducted two sets of simulation experiments. In the first set, we used the same simulation topology as in Fig. 41. In the second set, we added one best-effort aggregated source of ten individual sources. Each set consists of three experiments. For both the aggregated sources, we applied proportional marking in the first experiment, the IN-fair marking in the second experiment, and the BW-fair marking in the third experiment in each set.

The results are presented in Fig. 43. In Fig. 43(a), the total throughput of each experiment is similar to the throughputs of the others since network resources are limited even though the proposed algorithms send more IN packets. Only difference between proportional marking and the proposed algorithms is that throughput of proportional marking strategy in the congested link is achieved by both IN and OUT packets while throughputs of the proposed algorithms in the congested link are achieved only by IN packets. This results in different performance in the experiments with best-effort flows. In Fig. 43(b), IN packet throughput of each flow is protected even in the presence of best-effort flows, but the OUT packet throughput is reduced by best-effort flows. As a result, the new marking algorithms achieve higher throughputs through the congested link.

4. Impact on network

Fig. 44 shows the instantaneous queue length of the router on the congested link. The graph marked ‘Exp.1’ shows the queue length when both the aggregated sources use proportional marking scheme, and the graph marked ‘Exp.2’ shows the queue
Fig. 43. Impact of marking algorithms on throughput
Fig. 44. Queue lengths of different marking algorithms

length when both the aggregated sources use Algorithm 1. The graph marked ‘Combined’ shows queue length when one aggregated source uses proportional marking and the other uses Algorithm 1. The graph, ‘Exp.1’ shows that queue length is maintained between 20 and 40 packets after slow-start since maximum and minimum thresholds for OUT packets are 20 and 40 packets. The queue length in Exp.2 is much higher than in Exp.1 because most packets sent to congested link are marked IN. Since IN packets are configured with the RED (min, max) thresholds of (40, 100), the packets observe longer queue lengths (and delays) at the router. In the combined experiment, queue lengths again approach Exp.1 after a burst of losses around 12 seconds. Since the drop thresholds are configured differently for IN and OUT packets, the new marking algorithm resulted in larger queuing delays through the congested path in the network. It is noted that instantaneous queue lengths may exceed the RED thresholds since RED uses average queue lengths.
5. Impact on other flows at transient time

In this section, we consider the situation when users switch their packet marking schemes for achieving desired throughput and discuss the impact of marking schemes on the individual flows at transient time. We did a simple simulation with topology described in Fig. 41. Aggregated source 1 uses proportional marking scheme during the whole simulated period, while aggregated source 2 switches from proportional marking to Algorithm 1 at the simulated time of 30 seconds.

Traces of congestion windows (cwnds) of two individual sources sending packets on the congested link within each aggregated source are presented in Fig. 17. During the interval from 0 to 30 seconds, cwnds of two sources are not much different from each other. During 30-42 seconds, source 2 continues increasing its window even though the congestion level on this link hasn’t changed. The individual sources are still TCP sources. Due to the higher service level of these packets (all packets marked IN), the packets are not dropped. During this transition (until the queue lengths build up to a level where IN packets have to be dropped), this source continues sending packets at a higher rate even though the link is congested. From 30 second, cwnd of source 2 is drastically increased without packet loss since it sends only IN packets after switching marking scheme. As a result, cwnd of source 1 is reduced to one.

C. Discussion and related work

Our simulation experiments show that: (a) Even though individual TCP flows may respond to congestion, this congestion avoidance backoff can be muted by moving to next higher service level (as seen in Fig. 45). (b) During such transitions, the congestion may actually increase (as observed by increased queue lengths) even when individual flows respond to congestion (Fig. 44). (c) When employed universally,
Fig. 45. Congestion windows of different marking algorithms
these strategies can result in a bidding war for resources as each aggregated source shifts its resources to congested links (by moving up the service levels) and (d) the congested links eventually settle down to serving packets of highest service level while shutting down service for other packets (best-effort flows in our simulations, as seen in Fig. 43(b)).

Pricing [39, 40] will have an important effect on a number of the above observations, specifically on the nature of moving up the service levels. It has been suggested [29] that resources should be priced based on the level of congestion to balance load evenly across the network links. Network providers may employ fair-sharing techniques [1, 2, 3, 4, 26, 41] to balance resource utilization among competing aggregated sources at the time of congestion. In such a case, shifting resources to congested links will likely have less impact than observed in this study.

Aggregated sources also pose interesting new questions. As observed in this study and by several others, aggregated sources do not behave like TCP sources even when all the constituent flows within the aggregation are TCP flows. With suggested modifications to TCP [16], individual TCP flows will appear less responsive to congestion. If the aggregated source readjusts its resources among the individual flows, even when an individual flow backs off, it is likely that another flow within the aggregation may send more packets through the congested link. All of these issues point to the need for studying network bandwidth management and network dynamics further.

Recent work on diff-serv networks mostly dealt with individual sources [11, 15, 17]. Adaptive marking to achieve throughput targets for single sources is studied in [17]. Aggregation of individual traffic sources and the resulting traffic distributions have been studied [42, 43]. Aggregation in the context of RSVP has been studied in [44, 45, 46]. Our work looked at the problem of sharing available resources (IN-
profile bandwidth) among the individual flows of an aggregation to achieve specific performance goals [47].

D. Summary

In this chapter, we have studied the impact of aggregated sources on the guarantees/performance provided by a differentiated-services network. We have shown that differences in RTTs of individual flows have less of an impact on the realized throughputs of aggregated sources than on the throughputs of individual sources. We have shown that proportional marking of packets within an aggregation can lead to significant unfairness among the flows within the aggregation. We have proposed new marking algorithms for aggregated sources that improve fairness among the individual flows within the aggregation. The proposed algorithms maintain state for individual flows at the boundary marker to achieve specific performance goals. We have presented simulation results to show the impact of the proposed algorithms on realized throughputs, network congestion and the scalability of the proposed approaches.
CHAPTER V

ADAPTIVE MARKING FOR AGGREGATED FLOWS

In the previous chapter, we have presented new marking schemes for aggressive bandwidth management. The simulation results show that the new schemes improve throughput when the network resources are enough to meet the individual QoS requirements. However, it is also observed that, if the resources are not enough, the new schemes cause severe congestion resulting in resource wastage due to their aggressive manner in managing the contract rate.

In this chapter, we discuss how to control individual throughputs with marking rates within an aggregation and propose an adaptive marking strategy. The most desirable situation is clearly to guarantee individual target rates for all the individual flows. However, it is also clear that there exist situations in which some targets cannot be reached: (i) When there is a severe congestion along the path and the current available bandwidth is less than the target. (ii) When the contract rate is not enough to achieve the target. If we try to achieve the target by increasing marking rate of an individual flow observing the case (i), it makes the congestion more severe and results in resource wastage. This is the very undesirable situation for both customers and service providers.

We here propose an adaptive marking scheme for aggregated flows which guarantees at least one of the followings to all the aggregated flows:

1. Individual target rate when it is reachable.

2. Maximized throughput without IN packet loss when the current available bandwidth is less than the individual target.

3. Throughput achieved with $M/n$ marking rate where $M$ and $n$ are the total
marking rate and the number of aggregated flows, respectively.

Initially, we set the marking rate of each flow proportional to its target. If every flow gets throughput more than their marking rate without IN packet loss, then the adaptive marker works as a weighted IN-fair marker. On the other hand, if the network path of a flow is oversubscribed\(^1\) and observes IN packet losses (resource wastage), the adaptive marker adjusts marking rates of individual flows in order to avoid IN packet loss (achieving the second objective).

However, it is not easy for a marker to find whether a flow observes an oversubscribed network or not unless the marker is combined into the sender. For marking of aggregated flows, the marker cannot be combined into an individual sender. Thus, it can be just estimated from the current throughput. To estimate the current condition of a flow, we use throughput model proposed in [37]. From the model, throughput \(B\) of a TCP flow experiencing oversubscribed network is given by

\[
B = \min\left\{ \frac{3}{4} \frac{m}{RTT}, \frac{k}{RTT} \left( \sqrt{\frac{1}{9} + \frac{8}{3 p_m}} - \frac{1}{3} \right) \right\}
\]  

(5.1)

where \(m\) is the contract rate of the flow (or the IN-marking rate), \(k\) is the packet size, and \(p_m\) is the probability of IN packet loss. From (5.1), when throughput achieved by a flow is less than \(0.75 m\), the flow should observe an oversubscribed network.

Therefore we classify a flow into one of the following three states and treat these states differently. Here, \(t_i\) is the target rate of \(i^{th}\) individual flow, \(m_i\) is the marking rate, and \(h_i\) is the realized throughput. The target rate can be specified by the individual users, and \(\sum t_i\) is the contract rate for the aggregation.

---

\(^1\)In [37], oversubscribed network has been defined as a situation in which a flow does not transmit any OUT packets since every OUT packets are dropped or no OUT packet is sent when the sending rate is less than the contract rate. In a oversubscribed network, a flow usually experiences some number of IN packet losses.
At every observation period:
1. for $i \leftarrow 1 \text{ to } n$
2. if $0.75m[i] < b[i] < t[i]$
3. \hspace{1em} $m[i] = m[i] + \Delta (b[i] - t[i])$
4. else if $b[i] \leq 0.75m[i]$
5. \hspace{1em} $m[i] = m[i] - \Delta (0.75m[i] - b[i])$
6. else if $b[i] > t[i]$
7. \hspace{1em} $m[i] = m[i] - \Delta (b[i] - t[i])$
8. Do Max-Min fair with $m[i]$ and $M$

$m[i]$: Marking rate of $i^{th}$ flow
$b[i]$: current rate of $i^{th}$ flow
$t[i]$: Target rate of $i^{th}$ flow
$M$: Total marking rate = Aggregate contract rate
$n$: Number of flows

Fig. 46. An algorithm for adaptive marking

- $b_i \leq 0.75m_i$: In this state, the flow observes an oversubscribed network, and some IN packets are lost. Thus, the marker reduces $m_i$ so that $b_i$ is maintained to be higher than $0.75m_i$ to avoid wasting resources.

- $0.75m_i < b_i < t_i$: In this state, the flow does not reach its target. Since the network is not oversubscribed, $b_i$ can be increased by increasing $m_i$. Thus, the marker increases $m_i$ of that flow if resources are available.

- $t_i \leq b_i$: In this state, the flow already achieved its target. Thus, the marker reduces $m_i$ to avoid wasting resources.

Fig. 46 shows an example algorithm for the adaptive marker.

**Theorem:** The adaptive marking algorithm finds $m_i$ for which $b_i \geq \min\{t_i, b_{a,i}, b_{b,i}\}$ for $1 \leq i \leq n$ where $t_i$ is the target rate, $b_{a,i}$ is maximum achievable rate such that $b_i \geq 0.75m_i$, and $b_{b,i}$ is rate achieved with $M/n$. 
Assumption:

1. \( \frac{\partial b_i}{\partial m_i} \geq 0 \): Throughput of a flow does not decrease as the marking rate of the flow increases when other network condition is not changed.

2. \( \frac{\partial^2 b_i}{\partial m_i^2} \leq 0 \): There exists only one marking rate \( m_i \) at which \( b_i = 0.75m_i \).

3. If \( b_i < 0.75m_i \), there is IN packet drop.

Proof:

For each flow,

- When \( \min\{t_i, b_{a,i}, b_{b,i}\} = t_i \):
  
  If current \( b_i \) is less than \( t_i \), \( b_i \) should be greater than \( 0.75m_i \) since \( t_i \) is less than \( b_{a,i} \). Then, \( m_i \) increases from line 3. From the assumption that \( \frac{\partial b_i}{\partial m_i} \geq 0 \), \( b_i \) eventually reaches \( t_i \). Since \( t_i \) is less than \( b_{b,i} \), \( m_{t,i} \), at which \( b_i \) is equal to \( t_i \), is less than \( M/n \), and line 8 does not change \( m_i \). Refer to Fig. 47(a).

- When \( \min\{t_i, b_{a,i}, b_{b,i}\} = b_{a,i} \):
  
  If current \( b_i \) is less than \( b_{a,i} \), \( b_i \) is greater than \( 0.75m_i \) from the definition of \( b_{a,i} \) and, at the same time, less than \( t_i \) since \( t_i \) is greater than \( b_{a,i} \). Then, from line 3, \( m_i \) increases until \( b_i \) reaches \( b_{a,i} \). Here also, since \( b_{a,i} \) is less than \( b_{b,i} \), \( m_{a,i} \) is less than \( M/n \), and line 8 does not change \( m_i \). Refer to Fig. 47(b).

- When \( \min\{t_i, b_{a,i}, b_{b,i}\} = b_{b,i} \):
  
  If current \( b_i \) is less than \( b_{b,i} \), \( b_i \) is greater than \( 0.75m_i \) and less than \( t_i \) since \( b_{b,i} \) is less than \( b_{a,i} \) and \( t_i \). Then, from line 3, \( m_i \) increases until \( M/n \). Then, \( b_i \) reaches \( b_{b,i} \). Refer to Fig. 47(c).
Marking rate (m)  Achieved rate (b)  $b = 0.75m$

(a) $\min\{t_i, b_{a,i}, b_{b,i}\} = t_i$

(b) $\min\{t_i, b_{a,i}, b_{b,i}\} = b_{a,i}$

(c) $\min\{t_i, b_{a,i}, b_{b,i}\} = b_{b,i}$

Fig. 47. Marking rate vs. achieved rate
Time complexity of this algorithm is $O(n \log n)$ where $n$ is the number of flows, and this is allowable for an edge device marker. In this algorithm, we use TSW [11] to smooth out the individual throughput. It is important to choose $\Delta$, observation period and window size for rate estimator properly. We study impact of $\Delta$, observation period and window size in the following simulations.

A. Achieving target rates

In this section, we show how the proposed marking scheme realizes achievable individual target rates and finds maximized throughputs when the target rates are not achievable.

We consider a multi-hop path as shown in Fig. 48. There are $n$ routers, and cross traffic is injected to this network at the $i^{th}$ router and exits at the $(i+1)^{th}$ router.

To observe how the marking rate is adjusted and an individual flow achieve its target rate, we conducted a set of simulations. In the simulation, we set the link capacity 3 Mbps and use 10 TCP flows for cross traffic. The contract rate for each TCP flow is randomly selected from 0 to 1 Mbps, and the total contract of cross traffic is 2.7 Mbps so that the subscription level is 90%. The number of routers ($n$) is five. For the tagged flow, we use single TCP flow.
First, to observe path characteristics, we use static contract rate for the tagged flow. We vary the contract rate from 0 to 0.8 Mbps. Fig. 49 shows the results. The dashed line indicates the achieved rate is equal to the 75% of marking rate. It is clear that the achieved rate increases as the marking rate increases until 0.5 Mbps. It is also observed that after 0.55 Mbps the achieved rate does not increase even if we increase the marking rate upto 0.8 Mbps. This observation supports our assumption that if we increase the marking rate more than the point in which the achieved rate is the 75% of the marking rate, the flow observes oversubscribed path and wastes the marking rate. In this example, the maximum achievable rate is about 0.42 Mbps.

Now the tagged flow is an individual flow within an aggregation with aggregated contract rate. The marker for the aggregation employs the adaptive marking. We vary the target rate for the tagged flow from 0.1 to 0.5 Mbps. Fig. 50 shows the results. In each figure, dots indicate instantaneous marking and achieved rate, and a
square shows the average.

In this path, a flow gets 0.15 Mbps with zero contract rate. When the target rate is 0.1 Mbps (Fig. 50(a)), therefore, the marking rate stays around zero. When the target rate is achievable (less than 4.2 Mbps), it is observed that the adaptive marking scheme finds the minimum marking rate to realize the target (Fig. 50(a) to 50(c)). In Fig. 50(d), it is also observed that the marking rate stays less than 0.55 Mbps to avoid resource wastage when the target is unachievable.

So far, we have looked at throughput of an individual flow within an aggregation. Now we observe aggregated flows. Fig. 51 shows the simulation topology. There are nine flows aggregated. The contract rate for the aggregation is 5 Mbps. We set the individual target rates differently and offer different cross traffic at each link between $R$ and a receiver to produce differently network condition.

Fig. 52 shows realized throughputs and marking rates of some individual flows. When the targets are achievable (Fig. 52(a)), the realized throughputs stay around their targets while the marking rates keep changing to adapt the network condition. When the target is unachievable (Fig. 52(b)), the marking rate is managed to maintain the achieved rate to be the 75% of the marking rate.

Table VII shows the summary of the results. It is clear shown that the adaptive marking avoids resource wastage ($p_{in} = 0$ for all the flows) and, at the same time, maximizes throughput for the flows having unachievable targets ($p_{out} \approx 1$ and utilization by IN packet $\approx 1$). The flows with infinite target rates are for simulating FTP kind of applications and consume the residual contract rates.
Fig. 50. Achieved rates with the adaptive marking
Fig. 51. Network topology for adaptive aggregated marking

<table>
<thead>
<tr>
<th>Target rate</th>
<th>Marking rate</th>
<th>Achieved rate</th>
<th>$p_{in}$</th>
<th>$p_{out}$</th>
<th>Util. by IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.034</td>
<td>0.108</td>
<td>0</td>
<td>0.051</td>
<td>0.321</td>
</tr>
<tr>
<td>0.3</td>
<td>0.301</td>
<td>0.302</td>
<td>0</td>
<td>0.102</td>
<td>0.852</td>
</tr>
<tr>
<td>0.5</td>
<td>0.529</td>
<td>0.507</td>
<td>0</td>
<td>0.154</td>
<td>0.947</td>
</tr>
<tr>
<td>0.7</td>
<td>0.468</td>
<td>0.339</td>
<td>0</td>
<td>0.998</td>
<td>1</td>
</tr>
<tr>
<td>0.9</td>
<td>0.591</td>
<td>0.408</td>
<td>0</td>
<td>0.951</td>
<td>0.999</td>
</tr>
<tr>
<td>1.1</td>
<td>0.768</td>
<td>0.621</td>
<td>0</td>
<td>0.973</td>
<td>0.999</td>
</tr>
<tr>
<td>Inf</td>
<td>0.769</td>
<td>0.808</td>
<td>0</td>
<td>0.006</td>
<td>0.794</td>
</tr>
<tr>
<td>Inf</td>
<td>0.769</td>
<td>0.801</td>
<td>0</td>
<td>0.007</td>
<td>0.799</td>
</tr>
<tr>
<td>Inf</td>
<td>0.769</td>
<td>0.921</td>
<td>0</td>
<td>0.003</td>
<td>0.611</td>
</tr>
</tbody>
</table>

Table VII. Simulation results with adaptive marking
Fig. 52. Realized throughput with the adaptive marking
Table VIII. Expected bandwidth achieved by individual flows

<table>
<thead>
<tr>
<th>Destination</th>
<th>Sink 0</th>
<th>Sink 1</th>
<th>Sink 2</th>
<th>Sink 3</th>
<th>Sink 4</th>
<th>Sink 5~9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Achieved BW (Mbps)</td>
<td>0.125</td>
<td>0.25</td>
<td>0.375</td>
<td>0.5</td>
<td>0.625</td>
<td>0.75</td>
</tr>
</tbody>
</table>

B. Performance evaluation through simulations

In this section, we present a number of simulations and discuss the results. We set up the network topology shown in Fig. 53 using ns-2 [22]. There are four aggregations and ten sinks. Each aggregation consists of 10 individual TCP flows. $i^{th}$ individual flow of each aggregation sends packets to $i^{th}$ sink through $R0$ and $R1$. Contract rate of an aggregation is 5 Mbps. Individual target rate is set to 0.5 Mbps for simplicity. Link bandwidth between $R0$ and $R1$ is 22.5 Mbps which is higher than the total contract rate (20 Mbps). Link bandwidth between $R1$ and each sink is set differently so that each flow within an aggregation experiences different network conditions. Link bandwidth between $R1$ and $i^{th}$ sink is set to $0.5 \times (i + 1)$ Mbps. Propagation delay of each link is 5 msec. If the bandwidth is equally shared, individual flows of each aggregation are expected to get the bandwidth as in Table VIII. For droppers, we use RIO presented in [11] with parameters 20/40/0.5 for OUT packets and 40/80/0.02 for IN packets.

1. Impact of different $\Delta$

We study impact of different $\Delta$ through simulation. Again, $\Delta$ controls the rate at which the edge marker adapts the marking rate based on observed bandwidths of flows. In this simulation, we set $\Delta$ of each marker to 0.5%, 1%, 5% and 10%. Observation period of each marker is 100 msec. which is greater than RTT ($= 40$ msec.).
Fig. 53. Simulation topology

Fig. 54 shows the result. Note that we do not specify the marking rate of Flows 3-9 in Fig. 54(a)-54(d) since they already reach their target rates (see Fig. 55), and thus their marking rates are not much different from each other. Initially, marking rate for each individual flow is 0.5 Mbps. Then, Flow 0 of each aggregation shares 0.5 Mbps link between R1 and Sink 0 with other three flows and gets about 0.125 Mbps average throughput. Thus, it is observed that the marking rate of Flow 0 in each aggregation is reduced so that the current throughput is maintained to be at least 75% of its marking rate.

Different $\Delta$ impacts the time taken to reach the steady state. With smaller $\Delta$, it takes a longer time to reach a steady state. With larger $\Delta$, it is more likely to have oscillations. This is very similar to a heavily damped or under damped control in traditional control systems. Under stable network conditions, a flow’s marking rate reaches steady state at a rate of $(1 - \Delta)$ per observation period. So over $k$ observation periods, the error is $(1 - \Delta)^k$. If steady state marking error goal is $\delta$, then we can
find the suitable transient time \( (k \text{ observation periods}) \) through

\[
(1 - \Delta)^k < \delta
\]  

(5.2)

In other words, given a goal of how fast we need to reach a steady state (i.e., given \( k \)), we can find the rate of adaptation \( \Delta \), through (5.2). However, it is also observed that average throughput of different \( \Delta \) is not much different from each other over long period of time in Fig. 55.

2. Impact of different observation periods

In this section, we set observation period of each marker differently. First, we change observation period from 0.05 sec. to 1 sec. with a fixed \( \Delta \) (1%) to study impact of observation period on time taken for individual marking rate to reach steady state.

In the second simulation, we select \( \Delta \) so that the amount of marking rate change in a given time \( \tau \) is equal to each marker. Given \( o_i \) (observation period) and \( \Delta_i \), if we want two flows to converge to their target in the same amount of time, from (5.2) we have

\[
(1 - \Delta_i)^{\frac{\tau}{o_i}} = (1 - \Delta_j)^{\frac{\tau}{o_j}}
\]  

(5.3)

\[
\frac{o_i}{o_j} = \frac{\log(1 - \Delta_i)}{\log(1 - \Delta_j)}
\]  

(5.4)

From (5.4), following four (\( \Delta \), Obs. period) pairs are selected for each marker: (0.5%, 0.05 sec.), (1%, 0.1 sec.), (5%, 0.5 sec.) and (10%, 1 sec.).

Fig. 56 shows the result of the first simulation. In Fig. 56(a)~56(d), it is clearly shown that time to steady state is linearly proportional to the observation period when \( \Delta \) is the same. Note the marking rate of Flow 0: when obs. period = 1 sec, it takes about 200 sec. to reach 0.2 Mbps. With 0.5 sec. of obs. period, it is reduced to about 100 sec., and so on. In Fig. 57, it is also shown that throughput of flows
(a) $\Delta = 0.5\%$

(b) $\Delta = 1\%$

(c) $\Delta = 5\%$

(d) $\Delta = 10\%$

Fig. 54. Marking rate with different $\Delta$
Fig. 55. Throughput with different $\Delta$

using small observation period is slightly higher than throughput of flows using large period in Flows 1~6. It is because a marker with small observation period can adjust marking rate quickly according to the change in network conditions.

Fig. 58 shows the result of the second simulation. In Fig. 58(a), it is clearly shown that the converging time in the adaptive marker can be effectively controlled using (5.4). It is also shown that marking rates with small $\Delta$ and small observation period change smoothly due to small $\Delta$. In Fig. 58(b), it is observed that average throughput of different aggregations using different observation periods is not much different from each other. However, it is also shown that throughput of flows using small observation periods is slightly higher than throughput of flows using large periods in Flows 1~5.

3. Impact of window size for rate estimator

In this section, we study impact of window size of TSW rate estimator. TSW rate estimator was proposed to estimate sending rate for packet tagging and shown that it is effective to smooth out TCP burstiness in [11]. We use this rate estimator for adapting the marking rate to the current individual throughput. Generally, there is a trade-off in choosing window size: With a small window, the estimated throughput reflects the changes of throughput quickly but fails to smooth the burstiness. With a
Fig. 56. Marking rate with different observation period and constant $\Delta$.
Fig. 57. Throughput with different observation period and constant $\Delta$

large window, throughput is effectively smoothed out but not changed quickly.

To observe impact of window size, we conducted simulation with different sizes of window for each marker. Observation period is set to the same as its window size, and $\Delta$ is 100% for every marker.

Fig. 59 and 60 show the marking rate and throughput of Flow 0 within each aggregation. It is observed that the marking rates oscillate due to the large $\Delta$ (100%). In Fig. 59(a) and 59(b), the marking rate oscillates over large range (0 ~ 0.7 Mbps) since TCP throughput cannot be smoothed with the small window (0.05/0.1 sec.). Here note that throughput stays at the bottom of the marking rate. When a set of packets arrives at the marker, estimated throughput instantaneously goes up due to the small window. This increases the marking rate. However, this increasing marking rate is not effective since the source stops sending until receiving ACKs (an RTT = 40 msec. delay). At this time, throughput decreases, and the marking rate also decreases. Then, the next set of packets observes the decreasing marking rate. This is due to the fact that the marking rate is changed quickly based on the changes of throughput. With a large window (0.5/1 sec.), the marking rate stays around 4 Mbps even though it oscillates over 0.3 ~ 0.5 Mbps, and throughput is managed to achieve around 0.22 Mbps in Fig. 60(a) and 60(b).
Fig. 58. Marking rate and throughput with different observation period and different $\Delta$
Fig. 59. Marking rate and throughput of Flow 0 with different window size and O.P.
(a) Window = 0.5 sec.

(b) Window = 1 sec.

Fig. 60. Marking rate and throughput of Flow 0 with different window size and O.P.

(2)
Fig. 61. Average throughput with different window size and O.P.

Fig. 61 shows the average throughput of individual flows. With a small window, the flows observing congested links (Flows 0 ~ 2) cannot reach their targets while the other flows get much higher than their targets (since resources are shifted to those flows).

4. Dealing with network dynamics

In this section, we study how the adaptive marker adjusts the marking rate to changes in network conditions over time. To simulate changes in network conditions, we start Aggregation 0, 1, 2 and 3 at time 0 sec., 60 sec., 120 sec. and 180 sec. and stop at time 240 sec., 300 sec., 360 sec. and 420 sec., respectively. (∆, Obs. period) pair is set to (1%, 0.1 sec.) for all the markers.

Fig. 62 shows the marking rate and throughput of individual flows within Aggregation 0 over time. In Fig. 62(b), every flow reaches its target rate until Aggregation 1 starts sending packets at time 60 sec. Thus, it is shown that the marking rate of every flow is equal to each other as 0.5 Mbps. In time duration from 60 sec. to 120 sec., throughput of Flow 0 falls down to 0.25 Mbps since 0.5 Mbps of link bandwidth is shared with Aggregation 1, and the marking rate is also reduced to avoid IN packet loss. Marking rates of other flows increases to utilize the total contract rate. At 120
sec. and 180 sec., Aggregation 2 and 3 start sending packets, respectively. Then, similarly, throughputs of Flows 1 and 2 do not reach their current marking rate, and the marking rate is reduced. At time 240 sec., 300 sec. and 360 sec., throughput increases as other aggregations stop sending. Here, note that the marking rate of Flow 0 does not increase to 0.5 Mbps after time 360 sec. since its throughput already reaches its target rate.

5. Dealing with different RTTs

In this section, we observe how the adaptive marker deals flows with different RTTs. In this simulation, we use the same topology as used in the previous simulations except that propagation delay of link between R1 and each sink is randomly selected from 10 msec. to 60 msec. and that bandwidth is set to 3 Mbps for all the links between R1 and sinks.

Fig. 63 shows the result. It is clear that the adaptive marking effectively removes RTT-bias of TCP flows and realizes QoS goals of individual flows within aggregations.

6. Dealing with unresponsive flows

In this section, we compare the adaptive marking with proportional marking in presence of unresponsive flows. In network topology in Fig. 53, we attach three UDP sources at R1 with negligible contract rate. The UDP flows are connected to Sink 7, 8 and 9 and start sending packets at 180 sec, 120 sec. and 60 sec., respectively. The sending rate of each UDP flow is 3 Mbps. With this topology, we conducted simulation two times. In the first simulation, the adaptive marker is employed, and the proportional marker is employed in the second.

Fig. 64 and 65 show the results. In Fig. 65(a), 65(b) and 65(c), it is observed
Fig. 62. Marking rate and throughput in network changed over time
that throughput with the proportional marking is affected by UDP flows. Throughput with the adaptive marking is maintained at a stable rate. In Fig. 64(a), it is shown that Flows 0~6 within the aggregation also maintain their throughput. In Fig. 64(b), however, throughputs of Flows 0~6 are constantly fluctuating over time even though they do not observe UDP flow along their path. It is because marking rate of an individual flow is directly affected by other flow’s current throughput in the proportional marking case.

C. Receiver-side marking strategies

In the above sections, we have discussed marking strategies for achieving bandwidth targets for sending data. However, when the customer wants to receive data from another host (e.g., using get command in FTP, browsing web-sites or Video-on-demand), it does not provide service differentiation, since the marker in receiver’s side can mark
(a) Throughput of Adaptive marking

(b) Throughput of Proportional marking

Fig. 64. Throughput comparison with adaptive and proportional marking with unresponsive flows (Flow 0-6)
(a) Throughput comparison of Flow 7

(b) Throughput comparison of Flow 8

(c) Throughput comparison of Flow 9

Fig. 65. Throughput comparison with adaptive and proportional marking with unresponsive flows (Flow 7-9)
only ACK packets. Even when the application is sending-intensive, with asymmetric link bandwidths, it may be necessary to ensure that the ACKs get sufficient bandwidth through the reverse path [4].

A receiver-controlled scheme using the explicit congestion notification (ECN) bit was proposed in [11]. ECN bit scheme was originally designed to provide congestion avoidance without packet drops. When congestion starts to occur, a router sets the ECN bit of packets instead of dropping them. The receiver copies the ECN bit in ACK replies to the sender, and then the sender reduces window size (or rate) to avoid congestion. In receiver-controlled scheme, a profile meter, installed at the receiver, measures the incoming rate. If the rate is within the profile, the meter resets the ECN bit, so that the sender maintains its transmission rate.

In this section, we propose a strategy for providing better service for receiving-intensive applications in a sender-controlled network. The basic idea is to inform the sender’s side about the receiver’s target rate and to transfer receiver’s contracted bandwidth to the sender’s edge router. The network marker on the sender side adds the transferred bandwidth to the sender’s profile and upgrades OUT packets to IN within the increased profile. A signaling protocol similar to RSVP [23] needs to be developed to enable such transfers of IN-profile bandwidth. In the proposed scheme, however, the signaling protocol messages need to be processed only by the edge routers unlike RSVP which need to setup every router along the flow’s path. Thus, it can be implemented without compromising the scalability of the current diff-serv architecture.

Fig. 66 shows a simple example of receiver’s side marking strategy. It is possible to achieve this transfer through “bandwidth brokers” in different networks. We expect this signaling to result in a transfer of IN-profile bandwidth from the receiver to the edge router connected to the sender. The edge router connected to the sender will
Fig. 66. Reserved bandwidth transfer by receivers in a DS network

maintain state for this flow and use the transferred bandwidth to upgrade OUT packets to IN transparently so as to improve the service delivered to the receiver. The bandwidth is not transferred to the sender in order to ensure that the transferred bandwidth is applied to upgrade service to the requesting receiver (and not to other flows being served by the sender). This is somewhat akin to the “receiver-pay mode” available in telephone networks in the form of collect calls.

How much bandwidth should a receiver transfer to the sender side to achieve a target rate? If the receiver transfers bandwidth aggressively, the sender could exploit this by reducing the number of packets marked IN to this flow. Ideally, the receiver should transfer minimal bandwidth to the sender to reach its target rate. The proposed algorithm for the receiver profile meter is presented in Fig. 67.

In the above algorithm, the receiver profile meter keeps and updates average
At every observation period:
1. if $b[i] < t[i]$
2. if $b^{out}[i] > 0$
3. $m^r[i] = \min(b^{out}[i], \Delta(t[i] - b[i]))$
4. else
5. if $m^r[i] > 0$
6. $m^r[i] = \min(m^r[i], \Delta(t[i] - b[i]))$

$b[i]$: Average rate of $i^{th}$ flow
$b^{out}[i]$: Average OUT packet rate of $i^{th}$ flow
$t[i]$: Target rate of $i^{th}$ flow
$m^r[i]$: Marking rate of $i^{th}$ flow paid by the receiver

Fig. 67. A simple algorithm for receiver-based bandwidth contract

rates and average OUT packet rate of subscribed flows. When the average rate is less than the target rate requested by the receiver (line 1), and if there is excess bandwidth (achieved by OUT packet) (line 2), the meter transfers some amount of contract rate to the sender's marker (line 3). The amount is limited by the current excess bandwidth in order to avoid resource wastage. If the average rate is higher than the target rate, then the meter takes back the contract rate to reduce payment.

1. Performance evaluation through simulation

In this section, we present simulations of receiver-based strategy when the adaptive marking is employed by the sender's marker. We use the same topology shown in Fig. 53 and observe throughput realized by the interaction between the sender's strategy and the receiver's strategy. Each aggregated sender has a contract rate of 5 Mbps. Sender's target rate of an individual flow is set to 0.5 Mbps. To observe the interaction with the receiver strategy, the receiver of Flow 0 within Aggregation 0 sets a target rate of 1 Mbps. We set link bandwidth differently to produce the following three scenarios:
1. Sufficient bandwidth: Network links have enough bandwidth to satisfy the receiver’s target rate of 1 Mbps. Link bandwidth between \( R0 \) and \( R1 \) is 22.5 Mbps, and bandwidth between \( R1 \) and each sink is 3 Mbps.

2. Insufficient bandwidth: Network links do not have enough bandwidth to reach the receiver’s target. We set link bandwidth between \( R1 \) and Sink 0 to 2.2 Mbps.

3. Plenty of bandwidth: Network links have plenty of bandwidth to satisfy every flow’s target rate. We set link bandwidth between \( R0 \) and \( R1 \) to 30 Mbps.

In all the above simulation scenarios, \( \Delta \) and the observation period are 5% and 0.1 sec. for both sender’s and receiver’s side. We present marking rate of individual flows within Aggregation 0 and throughput of Flow 0 in Aggregation 0.

Fig. 68 shows the result of the first scenario. It is observed that Flow 0 can realize the requested throughput 1Mbps. As the receiver increases its contribution, the sender’s marker reduces the marking rate of Flow 0 since its throughput exceeds the sender’s target rate (0.5 Mbps) and some of the other flows have not reached their targets.

Fig. 69 shows the result of the second scenario. Compared to Fig. 68(b), it is observed that throughput is achieved by only IN packets in Fig. 69(b) because the network bandwidth is not enough. Hence, in Fig. 69(a), the receiver stops transferring resources at around 0.7 Mbps to avoid resource wastage even though the throughput does not reach the target rate of 1 Mbps. Again, we observed that the sender moves its resources to other flows since other flows have not reached their targets.

Fig. 70 shows the result of the third scenario. In Fig. 70(a), note that the individual sender’s marking rate stays around 0.5 Mbps. The sender’s marker continues allocating 0.5 Mbps to this flow since all the flows have exceeded their target rates.
Fig. 68. Marking rate and throughput when the bandwidth is sufficient
Fig. 69. Marking rate and throughput when the bandwidth is not sufficient
The resources contributed by the receiver stay around 0.4 Mbps since this amount is enough to reach the target rate.

D. Summary

Our simulation experiments on sender’s marking strategies show that: (a) Aggregate source can effectively manage resources by marking packets of individual sources differently. (b) The adaptive marker guarantees the target rate for the achievable target or the maximized throughput without IN packet loss for the unachieved target to all the flows within an aggregation. (c) The adaptive marker performance is impacted by choosing the adaptation rate and the observation period. (d) The adaptive marking strategy is effective to deal flows with different RTTs and unresponsive flows.

Our results on receiver-based strategies show that: (a) A simple technique of transferring bandwidth from receiver to the sender’s side can be effective in improving the performance seen by a receiver, (b) the sender can exploit receiver’s willingness to pay by moving its resources to other flows within its aggregation, and (c) if sender employs proportional marking, a receiver willing to pay for improved service can extract a higher amount of the sender’s bandwidth for its flow.

In this chapter, we have studied how QoS could be improved by marking strategies at the sender and by a receiver. We have proposed an adaptive marking algorithm that enables reaching specific QoS goals of individual flows within the aggregation. We have presented simulation results to show the impact of the proposed algorithms on realized throughputs, network congestion and the scalability of the proposed approaches. The proposed adaptive marking scheme is shown to provide predictable and robust performance. With the proposed receiver-based scheme, the receivers can achieve improved service even in a sender-controlled network. We have shown that the
Fig. 70. Marking rate and throughput when the bandwidth is plentiful
amount of bandwidth paid by the receivers can be impacted by the sender’s marking scheme and observed network.
CHAPTER VI

CONCLUSIONS

This dissertation studies Assured Forwarding (AF) Per-Hop Behavior (PHB) in the differentiated services (DS) architecture. The DS architecture is a scalable solution for providing service differentiation and QoS for multimedia/real-time applications. The routers at the edge of the DS domain monitor and mark incoming packets of flows (individual or aggregated) based on the service profile contracted between the service provider and the customers. The networks provides different PHBs to different packets.

In AF PHB, the service profile is a certain amount of bandwidth. The edge routers mark a packet IN when the temporal sending rate is within profile and mark OUT otherwise. The core routers gives preference to IN packets while dropping OUT packets disproportionately at the time of congestion. This preferential drop mechanism is expected to provide better throughput for IN packets than OUT packets. With appropriate network provisioning, it is expected that this could result in bandwidth guarantees.

This dissertation starts by addressing the problem of bandwidth assurance when network traffic mainly consists of TCP flows. A TCP flow reduces its sending rate by half to respond to packet loss, and this congestion control scheme makes it hard to protect its contract rate. It is observed that a flow with higher contract rate may not reach its contract rate while a flow with lower contract rate gets more than its contract rate.

We propose a number of schemes to realize better service assurance without loss of scalability of the DS network. The proposed schemes include:

- Limiting packets marked OUT by maintaining TCP sending rate within profile.
• Providing drop probability inversely proportional to the contract rate for protecting flows with higher contract rate and realizing equal share of excess bandwidth.

• Providing three drop precedences based on long-term and short-term sending rate for better handling of TCP’s burstiness.

• Dividing the congestion window in TCP sender into two windows to respond differently to IN and OUT packet loss.

The proposed schemes are evaluated through extensive simulation works, and the results show that the schemes provide better realization of bandwidth assurance.

TCP throughput is usually determined by the round trip time (RTT) and the drop probability. In AF networks, the contract rate is also an important factor to determine the TCP throughput. This dissertation presents analytical models of TCP behavior in AF networks. The models quantitatively characterize TCP throughput as functions of the contract rate, the drop probability and RTT. The models are validated through a number of simulations in different scenarios and different network conditions. This models analytically explain the previous observations such as:

• When a flow reserves relatively higher bandwidth, the excess bandwidth is decreased as the contract rate is increased. If the contract rate is larger than a certain rate in a given network condition, the flow cannot reach its contract rate.

• When a flow reserves relatively lower bandwidth, it always realizes at least its contract rate. As it reserves less bandwidth, it obtains more excess bandwidth.

• As the probability of OUT packet drop decreases, the flows with less contract rate benefit more than the flows with higher contract rate.
• The realized throughput is inversely related to the RTT of the flow.

The DS architecture allows aggregated flows as well as individual flows. It also allows a customer to employ his/her own marker to manage individual flows within the profile while the service provider monitors the incoming traffic at the edge and remarks when the profile is not confirmed. This dissertation looks at behaviors of aggregated TCP flows in the DS network. It is addressed that the contract rate for an aggregation may be consumed unfairly by individual flows due to the different network conditions which the individual flows observe.

We propose simple marking schemes for aggregated flows to avoid the unfairness and provide individual QoS goals. The proposed schemes maintain per-flow state at the edge and distribute the contract rate to individual flows for (1) proportional share of the contract rate based on the individual target rates (*IN-fair marking*) and (2) inversely proportional share based on the current throughput to realize equal throughput within the aggregation (*BW-fair marking*). It is shown that the proposed schemes effectively remove the unfairness and realize the individual QoS goals.

However, it is also observed that the proposed schemes cause severe congestion resulting in wastage of the contract rate due to their aggressive manner in managing the contract rate. To alleviate this problem, we propose a new marking scheme which adaptively changes marking rate for individual flows to achieve their target rates and, at the same time, avoid severe congestion. The proposed adaptive marking scheme estimates the current network conditions based on TCP throughput model and determines whether an individual target rate is achievable or not. If it is achievable, the marker tries to achieve it aggressively within *Max-min fair* shared amount of the contract rate. Otherwise, the marker maximizes throughput without IN packet loss. The adaptive marking scheme is shown to provide predictable and robust performance.
REFERENCES


35-48.

[20] V. Jacobson and M. Karels, “Congestion avoidance and control,” in Proc. SIG-

[21] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the
Internet,” IEEE/ACM Transactions on Networking, vol. 7, no. 4, pp. 458-472,
August 1999.

http://www-nrg.ee.lbl.gov/ns/


simple model and its empirical validation,” in Proc. SIGCOMM’98, Vancouver,

approximately fair bandwidth allocations in high speed networks,” in Proc. SIG-

[27] D. Chiu and R. Jain, “Analysis of increase and decrease algorithms for congestion
avoidance in computer networks,” Journal of Computer Networks and ISDN


VITA

Ikjun Yeom received the B.S. degree in Electronic Engineering from Yonsei University, Seoul, South Korea, in 1995, and the M.S. degree in Computer Engineering at Texas A&M University, College Station, TX in August 1998. He received the Ph.D degree in Computer Engineering at Texas A&M University in May 2001. He worked at DACOM co. located in Seoul between 1995 and 1996. His research interests are in Internet QoS provisioning, the differentiated services network, and Internet congestion control.

His permanent address is 82 Brick Kiln Road Building No. 9 Apt. No. 105, Chelmsford, MA 01824.

The typist for this thesis was Ik-Jun Yeom.