2002

AF differentiated services performance analysis

Long Vu Nguyen
University of Wollongong

Follow this and additional works at: https://ro.uow.edu.au/theses

Recommended Citation

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
AF Differentiated Services Performance Analysis

A thesis submitted in fulfilment of the requirements for the award of the degree

Master of Engineering (Honours)

from

THE UNIVERSITY OF WOLLONGONG

by

Long Vu Nguyen
Bachelor of Engineering (Honours Class I)

SCHOOL OF ELECTRICAL, COMPUTER
AND TELECOMMUNICATIONS ENGINEERING
2002
Abstract

Internet applications such as email, ftp and real-time applications require different service performance guarantees. The Internet Engineering Task Force has proposed the Integrated Services (IntServ) model, which aims to provide these guarantees by reserving network resources for individual connections. However, there are two related issues arising from the deployment of IntServ: overhead traffic and poor scalability. As a result, Differentiated Services (DiffServ) has been proposed as an alternative for IntServ. DiffServ aims to provide the same service to groups of flows with similar Quality of Service requirements. Hence, it reduces the amount of overhead traffic and improves networks scalability.

Assured Forwarding (AF) and Expedited Forwarding (EF) are two DiffServ classes apart from the default best-effort service. These service classes provide different packet forwarding treatments at a per-node level. The EF service class aims to provide a low loss, low delay and low jitter service, while the AF service class offers different levels of forwarding assurances for data packets.

This thesis focuses on the AF service class due to its suitability to a range of applications from TCP traffic to real-time interactive applications. Users can choose either a low delay and loss tolerant service or a low loss and more delay tolerant service. Furthermore, a DiffServ router can implement several AF service classes independently. Within these classes, preferred packets are given preferred treatments as indicated by the packet drop-precedences.

Previous studies have shown that network traffic exhibits properties of self-similarity [1] and long-range dependence [2], [3], [4]. However, for finite buffer queues, these properties have insignificant impact beyond a certain time-scale [5], [6]. The correlation structure of input traffic and the maximum queue size determine this time-scale.
Hence, any Markovian model can be used to model the input traffic as long as it captures the correlation structure up to the time-scale [5]. In this thesis, the performance of an AF DiffServ node is analysed based on different traffic models such as Poisson and MMPP. These Markovian traffic model (i.e. Poisson and MMPP) can be used to model voice and data traffic [7], [8]. The packet size distribution chosen is exponential to provide a mathematical tractable analysis.

The queueing mechanism used to implement AF at a DiffServ node is either Random Early Discard with In/Out Profile RIO (an enhanced version of RED, [9]) or Threshold Dropping TD, which is a special case of RIO.

The performance analyses presented in this thesis include:

- An analysis to estimate packet loss and mean delay of an AF DiffServ node (TD or RIO) with Poisson traffic and an exponential packet size distribution.

- An analysis to estimate packet loss and mean delay of an AF DiffServ node (TD or RIO) with MMPP traffic and an exponential packet size distribution.

Apart from the analyses, this thesis presents a simulation study, which compares the steady-state performance of TD DiffServ nodes to RIO ones. It also shows the insignificant impacts made by the traffic proportion of different drop-precedences on the performance of the DiffServ node (under certain circumstances).
Acknowledgments

I would like to thank my supervisors, Dr. Tony Eyers and Dr. Chun Tung Chou for their assistance, guidance throughout the entire project.
I would also like to thank other friends within the Switched Network Research Centre at the University of Wollongong for all their helps in understanding \LaTeX, discussion and proof reading.
Finally and most importantly, I would like to thank my parents, my sister and a special person, Miss Trang Kim whose love and support encouraged me to do my best.

Long Vu Nguyen
## Contents

1 Introduction ................................................................. 1  
  1.1 Background .......................................................... 1  
  1.2 Overview ............................................................. 3  
  1.3 Contributions ......................................................... 4  
  1.4 Publications .......................................................... 5  

2 Service Models for IP Networks: IntServ/RSVP and DiffServ 6  
  2.1 Introduction .......................................................... 6  
  2.2 IntServ/RSVP .......................................................... 7  
    2.2.1 IntServ Service Classes ....................................... 8  
    2.2.2 RSVP .............................................................. 9  
    2.2.3 Weighted Fair Queueing ..................................... 11  
  2.3 DiffServ ................................................................. 12  
    2.3.1 Architecture and Elements of Differentiated Services 13  
    2.3.2 DiffServ Code Point .......................................... 15  
    2.3.3 Per-Hop Behaviour ............................................ 16  
    2.3.4 Per-Domain Behaviour ....................................... 22  
    2.3.5 DiffServ Queueing Mechanisms ............................ 24  
  2.4 DiffServ and MPLS .................................................. 27
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1 MPLS Concepts</td>
<td>27</td>
</tr>
<tr>
<td>2.4.2 DiffServ / MPLS Networks</td>
<td>29</td>
</tr>
<tr>
<td>2.5 Conclusions</td>
<td>34</td>
</tr>
<tr>
<td>3 DiffServ Performance Analyses</td>
<td>36</td>
</tr>
<tr>
<td>3.1 Theoretical Analysis</td>
<td>36</td>
</tr>
<tr>
<td>3.1.1 TD and RIO Queue Performance with Poisson Traffic</td>
<td>37</td>
</tr>
<tr>
<td>3.1.2 TD Queue Performance with MMPP Traffic</td>
<td>38</td>
</tr>
<tr>
<td>3.1.3 Modelling of AF performance on Wireless Links</td>
<td>40</td>
</tr>
<tr>
<td>3.2 Simulation Studies</td>
<td>41</td>
</tr>
<tr>
<td>3.2.1 Interactions between AF and EF</td>
<td>41</td>
</tr>
<tr>
<td>3.2.2 Effect of Number of Drop-Precedences</td>
<td>42</td>
</tr>
<tr>
<td>3.2.3 Proportional Forwarding Marking Schemes</td>
<td>43</td>
</tr>
<tr>
<td>3.3 Conclusions</td>
<td>45</td>
</tr>
<tr>
<td>4 Performance Analysis of DiffServ nodes (TD and RIO) with Poisson Arrivals</td>
<td>47</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>47</td>
</tr>
<tr>
<td>4.2 Existing Work and Adjustment</td>
<td>48</td>
</tr>
<tr>
<td>4.3 Analysis and Simulation Results</td>
<td>50</td>
</tr>
<tr>
<td>4.3.1 Analysis</td>
<td>52</td>
</tr>
<tr>
<td>4.3.2 Simulation Results</td>
<td>55</td>
</tr>
<tr>
<td>4.4 The Poisson Analysis and DiffServ Performances on MMPP Traffic</td>
<td>59</td>
</tr>
<tr>
<td>4.5 Traffic Trace Investigation</td>
<td>64</td>
</tr>
<tr>
<td>4.6 Conclusions</td>
<td>68</td>
</tr>
<tr>
<td>5 Performance Analysis of DiffServ nodes (TD and RIO) with MMPP Arrivals</td>
<td>69</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>5.1</td>
<td>Existing Work</td>
</tr>
<tr>
<td>5.2</td>
<td>Analysis and Simulation Results</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Analysis</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Simulation Results</td>
</tr>
<tr>
<td>5.3</td>
<td>Conclusions</td>
</tr>
<tr>
<td>6</td>
<td>Performance Comparison of TD and RIO queues</td>
</tr>
<tr>
<td>6.1</td>
<td>TD and RIO Performance Comparison</td>
</tr>
<tr>
<td>6.2</td>
<td>DiffServ Node Capacity Planning</td>
</tr>
<tr>
<td>6.3</td>
<td>Relative Proportion Study</td>
</tr>
<tr>
<td>6.4</td>
<td>Conclusions</td>
</tr>
<tr>
<td>7</td>
<td>Conclusions</td>
</tr>
<tr>
<td>7.1</td>
<td>Overview</td>
</tr>
<tr>
<td>7.2</td>
<td>Results and Recommendations</td>
</tr>
<tr>
<td>7.3</td>
<td>Conclusions and Future Work</td>
</tr>
</tbody>
</table>

**Bibliography** | 100
## List of Figures

2.1  A Differentiated Services network and its components ........................................... 14
2.2  Classifier and Conditioner at a boundary node ............................................................. 15
2.3  Packet drop probability of a RIO queue based on the average queue length ............. 25
2.4  Packet drop probability of a TD queue based on the instantaneous queue length ....... 26
2.5  Label forwarding model of a DiffServ LSR [10] .......................................................... 31

4.1  Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - [11] approach and Simulation results. The high priority flow contributes 95% of the total load ($K=16; L=6$). ................................................................. 49

4.2  (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - Analytical and Simulation results. The high priority flow contributes 95% of the total load ($K=16; L=6$). ................................................................. 51

4.3  RIO Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the load of the low priority flow - Analytical and Simulation results. The loads of the high and middle priority flows are fixed at 0.6 and 0.2 ................................................................. 57

4.4  TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the offered load of the high priority flow - Analytical and Simulation results. The loads of the medium and low priority flows are fixed at 0.7 and 0.4 ($L_3=16; L_2=12; L_1=8$). ................................................................. 58
4.5 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total load - Analytical and Simulation results. Both flows are On-Off with the high priority flow contributes 95% of the total load ($K=16; L=6$).  

4.6 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total load - Analytical and Simulation results. Both flows are aggregates of 20 identical On-Off sources with the high priority flow contributes 95% of the total load ($K=16; L=6$).  

4.7 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total load - Analytical and Simulation results. Both flows are aggregates of 5 identical On-Off sources with the high priority flow contributes 95% of the total load ($K=16; L=6$).  

4.8 Packet size distribution of a Telstra core network trace.  

4.9 (a) Loss Probability and (b) Expected Delay of High and Low Priority Packets as a Function of the Node's Service Rate. Packet Sizes are obtained from the trace.  

4.10 (a) Loss Probability and (b) Expected Delay of High and Low Priority Packets as a Function of the Node's Service Rate. Packet Sizes are obtained from the trace.  

5.1 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the load of the low priority flow - Analytical and Simulations results. The offered load of the high priority flow is fixed at 0.8 ($K = 16; \theta = 6$).  

5.2 RIO Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the load of the low priority flow - Analytical and Simulation results. There are three 2-state MMPP flows. The loads of the high and middle priority flows are fixed at 0.6 and 0.2.  

5.3 RIO Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - Analytical and Simulation results. There are two 4-state MMPP flows. The load of the high priority flow is fixed at 70% of the combined flow.
5.4 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the load of the low priority flow - Analytical and Simulation results. There are three 2-state MMPP flows. The loads of the high and middle priority flows are fixed at 0.6 and 0.2. The thresholds and buffer size are 8, 12 and 16 packets respectively. ........................................ 82

5.5 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - Analytical and Simulation results. There are two 4-state MMPP flows. The load of the high priority flow is fixed at 70% of the combined flow. The threshold and buffer size are 6 and 16 packets respectively. .......................... 83

6.1 A RIO queue with instantaneous queue length and maximum drop probability of 1 .............................................................. ................................ 86

6.2 (a) Loss probability of high priority packets and (b) Loss probability of low priority packets of a TD queue compared with RIO queues ............... 88

6.3 (a) Mean Delay of High Priority Packets and (b) Mean Delay of Low Priority Packets of a TD queue compared with RIO queues ....................... 89

6.4 Loss Probability of High Priority Packets with smaller queues ............... 90

6.5 (a) Loss probability of high priority packets and (b) Loss probability of low priority packets as a function of the ratio link capacity / mean arrival rate of the aggregate flow (high priority packets make up 80% of the aggregate) ................................. 92

6.6 Loss probability of low priority packets as a function of the proportion of high priority packets within an aggregate (the queue utilisation is 0.65 with the threshold and buffer size are 250 and 2000 packets respectively) ...... 94
List of Tables

2.1 DSCP of the AF PHBs ......................................................... 21
2.2 EXP to MPLS Service Mapping ........................................... 33
2.3 DiffServ and MPLS Mapping ............................................. 34
List of Abbreviations

AF  Assured Forwarding
AR  Assured Rate
ATM Asynchronous Transfer Mode
BA  Behaviour Aggregate
BH  Bulk Handling
CBQ Class Based Queueing
CIR Committed Information Rate
DiffServ Differentiated Services
DSCP DiffServ Code Point
DPI Drop-Precedence Indicator
EF  Expedited Forwarding
E-LSP EXP-Inferred LSP
FCI  Forwarding Class Indicator
FEC Forwarding Equivalence Class
FTN FEC-to-NHLFE Map
GPS Generalised Processor Sharing
IETF Internet Engineering Task Force
ILM  Incoming Label Map
IntServ Integrated Services
IP  Internet Protocol
ISP Internet Service Provider
L-LSP Label-Inferred LSP
LSP Label Switched Path
LSR  Label Switched Router
MF  Multi-Field
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMPP</td>
<td>Markov Modulated Poisson Process</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multi-Protocol Label Switching</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmission Unit</td>
</tr>
<tr>
<td>NHLFE</td>
<td>Next-Hop Label Forwarding Entry</td>
</tr>
<tr>
<td>OA</td>
<td>Ordered Aggregate</td>
</tr>
<tr>
<td>PDB</td>
<td>Per-Domain Behaviour</td>
</tr>
<tr>
<td>PF</td>
<td>Proportional Forwarding</td>
</tr>
<tr>
<td>PHB</td>
<td>Per-Hop Behaviour</td>
</tr>
<tr>
<td>PHP</td>
<td>Penultimate Hop Popping</td>
</tr>
<tr>
<td>PIR</td>
<td>Peak Information Rate</td>
</tr>
<tr>
<td>PQ</td>
<td>Priority Queueing</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RED</td>
<td>Random Early Discard</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RIO</td>
<td>RED with In/Out profile</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource Reservation Protocol</td>
</tr>
<tr>
<td>SIR</td>
<td>Subscribed Information Rate</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>TCA</td>
<td>Traffic Conditioning Agreement</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TD</td>
<td>Threshold Dropping</td>
</tr>
<tr>
<td>ToS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>TSW</td>
<td>Time Sliding Window</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>VBR</td>
<td>Variable Bit Rate</td>
</tr>
<tr>
<td>VW</td>
<td>Virtual Wire</td>
</tr>
<tr>
<td>WFQ</td>
<td>Weighted Fair Queueing</td>
</tr>
<tr>
<td>WRR</td>
<td>Weighted Round Robin</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

The Internet has become an indispensable part of modern life. Different applications, such as email, web browsing, file transfer and real-time applications require different service performance guarantees. However, the current Internet provides mostly best-effort, with no Quality of Service (QoS) guarantees. Quality of Service (QoS) provision will play an important role in future integrated packet switching networks. The Internet Engineering Task Force (IETF) formed an Integrated Service (IntServ) working group and proposed new service classes such as Controlled-Load and Guaranteed Service. IntServ is based on the Resource Reservation Protocol (RSVP) signalling protocol, where network resources are reserved for individual connections. Connections are created, based on service requirements and available network resources.

There are two related issues arising from the deployment of RSVP and IntServ:

1. IntServ introduces overhead traffic, due to RSVP reservation messages.

2. IntServ offers poor scalability. In a core network, where a router may support tens of thousands of flows, it is difficult to implement RSVP to handle every individual flow.
As a result, the IETF sought a simpler alternate model. The Differentiated Service working group was formed, and developed a new service model: DiffServ, which aims to provide the same service to groups of flows with similar QoS requirements. Hence, a DiffServ router only handles a small number of Behaviour Aggregates (BA, a group of connections which require similar treatments) [12]. This reduces the overhead traffic and increases network scalability.

Originally, apart from the default best-effort service class, DiffServ comprised two service classes: Assured Forwarding (AF) and Expedited Forwarding (EF). Recently, new ideas are introduced for another service class: Proportional Forwarding (PF). These service classes provide different treatments for flows at a per-node level, hence the name Per-Hop Behaviour (PHB).

- EF aims to provide a low loss, low delay and low jitter service. EF packets are transferred at a rate equal to or greater than a pre-configured rate.

- AF, on the other hand, offers different levels of forwarding assurances for data packets. Packets are given different drop-precedences at the edge of the network and dropped accordingly during congestion periods.

- PF aims to share bandwidth fairly based on each flow's committed information rate during congestion.

EF and AF are different types of forwarding behaviour. AF is less stringent than EF, and hence is suited to a different range of applications. AF provides flexibility in terms of loss and delay. Users can choose either a low delay and loss tolerant service (e.g. for interactive real-time applications) or a low loss and more delay tolerant service (e.g. for TCP traffic). A DiffServ node can implement several AF service classes independently and within these classes, the preferred treatment of packets is indicated by the packet's drop precedences. This research focuses on the performance analysis of the Assured Forwarding service class.

Studies on network traffic show that it exhibits properties of self-similarity [1] and long-range dependence (i.e. correlations over wide range of time scales) [2], [3], [4]. However, research presented in [5] and [6] found that for a finite buffer queue, the
effect of the correlation becomes insignificant beyond a time scale known as “correlation horizon” [5] or “critical time scale” [6]. This time scale is dependent on the correlation structure of the input traffic and the maximum queue size. It is further suggested in [5] that any Markovian models can be used as long as the chosen model captures the correlation structure up to the correlation horizon.

As a result, the performance evaluation of an AF DiffServ node presented in this thesis considers two mathematically tractable models namely the Poisson and MMPP ones. These models are well known and can be used to model voice and data traffic (MMPP [7], [8]).

There are different queueing mechanisms which can be used to provide AF treatments such as RED with In/Out (RIO) [9] and Threshold Dropping (TD). In a RIO queue, packets are dropped with a probability determined by the average queue length. The TD queue, a special case of the RIO queue, drops packets based on the instantaneous queue length. A packet is dropped if the queue length exceeds its respective threshold, hence the name Threshold Dropping. This buffer threshold is indicated by the packet drop precedence. It is obvious that the TD queue is biased against bursty traffic and provides a more oscillatory performances than the RIO one.

In this thesis, we will provide performance analyses of both RIO and TD queues with Poisson and MMPP traffic models. We also compare the steady state performances of these two queues.

1.2 Overview

The thesis chapters are organised as follows:

- Chapter 2 reviews developments in the provision of quality of service for IP networks. It covers the IntServ/RSVP model, architecture and service classes of DiffServ and proposals of implementing DiffServ and MPLS in conjunction.

- Chapter 3 reviews existing performance analyses of DiffServ networks and highlight the need for an analytical evaluation of DiffServ AF nodes. This study allows network providers to dimension DiffServ networks based on traffic characteristics and required performances.
• Chapter 4 presents a Poisson based analysis where an M/M/1 based AF queue is solved with multiple thresholds for multiple drop-precedences. The analytical results are verified with simulations. We also show that the Poisson analysis holds for large aggregates of MMPP flows.

• Chapter 5 presents the MMPP based analysis of an AF DiffServ node where incoming traffic is characterised by MMPP models. The analysis presented in this chapter is an extension on literature with the multiple arrival flows modelled by multi-state MMPPs. The performance analysis (packet loss and expected delay) is also verified with simulations with 4-state and 2-state MMPPs.

• Chapter 6 presents a simulation study which compares steady-state performances of a RIO queue and a TD queue. The RIO queue is designed to provide a more stable and less oscillatory performance than the TD one. Simulation results, which emulate real network environments, show however that at steady-state, the RIO queue can be approximated by a TD one. The simulations in this chapter also show that under some conditions, the proportion of high and low priority traffic arrival to a TD DiffServ node makes no significant impact on the performance of individual flows.

• Chapter 7 summarises findings, recommendations and suggests possible future work.

In this thesis, each simulation is run for 1000 times. The length simulation runs is of 10000 seconds or more with the number of packets depends on the arrival rate of the flow.

1.3 Contributions

The contributions of this project include:

• An analysis to estimate packet loss and expected delay of an AF DiffServ node (TD or RIO) with Poisson traffic and an exponential packet size distribution.
This analysis extends the work in [11] by considering multiple Poisson flows with an adjusted delay calculation.

- An analysis to estimate packet loss and expected delay of an AF DiffServ node (TD or RIO) with MMPP traffic and an exponential packet size distribution. This analysis extends the work in [13], by considering multiple MMPP flows with different numbers of MMPP states. The derived MMPP analysis can be applied for any number of AF flows with associated drop-precedences.

- A simulation study to investigate the applicability of the Poisson based analysis with different sized MMPP aggregates. Simulation results show that the Poisson based analysis can be used for large aggregates of MMPP traffic yet is unable to provide accurate performance estimations for smaller MMPP aggregates. This study also investigates and shows the significant impact of a real-life packet size distribution on the performance of the node as compared to the Poisson based calculations.

- A simulation study which compares the steady state performance of TD DiffServ nodes to RIO ones. It shows that the TD mechanisms approximate the RIO one in terms of steady state performance (packet loss probability and expected delay). This study also shows that under certain conditions, the proportion of traffic with different drop precedences makes no significant impact on the performance of the DiffServ node.

1.4 Publications

Chapter 2

Service Models for IP Networks: IntServ/RSVP and DiffServ

This chapter outlines service models for IP networks, including Integrated Services implemented over RSVP, Differentiated Services (DiffServ) and DiffServ over MPLS networks. We consider the services which can be implemented, as well as their shortcomings. The key focus of this chapter is the concepts and architecture of Differentiated Services, such as Per-Hop Behaviour and Per-Domain Behaviour. This leads to our subsequent analysis of the AF DiffServ node and its associated scheduling algorithms.

2.1 Introduction

Currently, network providers are paying increasing attention to data as well as voice communications. This has been prompted largely by the rapid uptake of Internet services by both commercial and residential users. These new applications place different requirements on the network, which require a range of service level contracts. The current Internet (which typically provides a single, best-effort service) generally does not provide network performance guarantees. Hence, the data packets in a given flow are generally affected by other network flows.

To address this problem, the Internet Engineering Task Force (IETF) defined several service classes (e.g. Guaranteed Quality of Service and Controlled Load), which pro-
Service Models for IP Networks: IntServ/RSVP and DiffServ

provide quality of service guarantees. In these service classes, network resources (bandwidth and buffer space) must be reserved (using RSVP) on a per connection basis to meet a desired performance as calculated prior to the connection establishment. These services comprise the Integrated Services (IntServ) model where the network considers the service requirements of every individual connection (Section 2.2). The major setback of this IntServ approach is its poor scalability. It is not feasible to implement IntServ in the core network, where routers carry large numbers of connections. A solution to this is to enable a core router to treat connections with similar quality of service requirements as a single flow. This has prompted the Differentiated Services (DiffServ) model with several packet forwarding treatments such as Assured Forwarding, Expedited Forwarding and Proportional Forwarding (Section 2.3). However, there is a lack of analyses to determine the quality of service (i.e. packet loss and expected delay) received by a flow throughout a DiffServ domain. The per-domain performance is obtained from a per-node basis, and each node needs to be dimensioned to meet the desired performance. Hence, the aim of our research is to provide a performance analysis of an AF DiffServ node to aid this dimensioning.

The review of industrial trends and standardisation activities in IP QoS support shows an increasing focus on service creation using DiffServ and MPLS in conjunction. The IETF Multi-Protocol Label Switching (MPLS) working group has recently introduced the idea of a DiffServ network supported by MPLS. AF queueing algorithms such as RIO and TD can be employed by MPLS routers. As a result, the performance analyses presented in this thesis can be applied for labelled flows in DiffServ/MPLS networks once they have been switched to their associated output ports. Key developments and emerging issues in the architecture of DiffServ/MPLS networks are identified in Section 2.4.

2.2 IntServ/RSVP

IntServ uses RSVP to reserve network resources for individual flows [14]. In addition, RSVP-capable routers must also be able to schedule packets to satisfy QoS parameters such as loss and delay. There are a number of packet scheduling methods with the most popular being based on Weighted Fair Queueing (WFQ). This section
investigates RSVP and the WFQ packet scheduling algorithm and raises the issue of overhead traffic and poor scalability of IntServ networks to emphasise the need for a simpler alternate model, namely DiffServ.

2.2.1 IntServ Service Classes

This section briefly describes the two integrated service classes apart from the best-effort service: Guaranteed Service and Controlled Load.

2.2.1.1 Guaranteed Service

Guaranteed Service is the highest-level service that an IntServ network offers. It is intended for applications that require a bounded packet delay. "This service guarantees that packets will arrive within the guaranteed delivery time and will not be discarded due to queue overflows provided the flow traffic stay within its specified traffic parameters" [15]. Applications that require guaranteed service include real-time play-back services. In a play-back service, the sender packetizes some signal, and then transmits it over the network. These packets are re-assembled by the receiver as it attempts to faithfully play back the signal at some designated play-back point. The queuing delay and jitter, introduced by the network, are determined by guaranteed service based on the traffic profiles and the resources reserved for the flow by RSVP. Since this service guarantees a firm delay bound, it requires support from all nodes along the path of the guaranteed flow.

2.2.1.2 Controlled Load

Controlled Load is intended for applications which are sensitive to overloaded conditions. "This service provides the client data flow with a quality of service closely approximating the QoS that the same flow would receive from an unloaded best-effort network when the actual network is overloaded" [16]. An example of applications that require controlled-load service is adaptive real-time applications, which is a type of real-time applications where the play-back point can be adaptively moved under current network conditions. Video, which can be made to adapt by dropping or replaying a frame as necessary and voice, which can adapt imperceptibly by adjust-
ing silent periods are adaptive real-time applications [17]. These applications work well under unloaded condition yet degrade when the network’s load increases. This service assumes that a high percentage of packets are successfully transmitted and experience similarly low delay.

2.2.2 RSVP

IntServ uses the Resource Reservation Protocol (RSVP) to reserve network resources for individual flows. These resources (bandwidth and buffer space) are determined based on traffic descriptors, the desired performance requirements (end-to-end delay bound) and information about the path of transmission. RSVP gathers information about routers along the transmission route and conveys the traffic descriptor to each network element along that route.

Originally, RSVP was intended to cope with dynamic changes in network multicasting. Hence, it grants receivers the responsibilities for making requests to reserve network resources, so that senders do not have to be aware of routing changes. RSVP is a signalling protocol which establishes end-to-end IP flows. Intermediate routers maintain soft-state information for each flow, which is automatically reinstated by RSVP when lost. This section will outline RSVP operations and the use of packet scheduling methods in RSVP routers to realise QoS parameters.

2.2.2.1 RSVP Implementation

As indicated in [14], [18] an RSVP connection set up can only be done at RSVP-capable routers. For a heterogeneous network, guaranteed connections have resources reserved at RSVP-capable routers, while non-RSVP routers treat the flows as best-effort. Hence, for these networks, performance guarantees are not possible. These RSVP-capable routers handle both data and RSVP flows where the data flow is transmitted through a classifier and a scheduler as in a normal IP router while the RSVP flow (RESV and PATH messages) is passed (upstream and downstream) through an RSVP processor and two controllers [18]. The decision to accept an RSVP request is done by means of two controllers: policy and admission, to check if the user has administrative permission and enough resources to make the reservation. The RSVP request is rejected if it fails at either controller. The receiver periodically
sends the RESV messages, which are identical to the original request, once the reservation is established. At the end of the connection, the receiver sends a tear-down message to release the reserved resources. Alternatively, it can tear-down a connection by withholding the RESV messages since the reserved resources at intermediate router is maintained as soft-state. RSVP uses the following messages:

- **PATH messages**: collect information about the path from source to destination. During the establishment phase, they are sent from the sender to the receiver along the IP route. A PATH message contains a $Tspec$ that specifies the characteristics of the source’s data flow and an $Adspec$ which is used to store information about local IP routers along the route. At every node visited by the PATH message, a PATH state is created. Upon the arrival of the PATH message at the receiver, the bandwidth required for the flow is calculated.

- **RESV messages**: are responsible for establishing the connection throughout the transmission process. Upon receiving the PATH message, the receiver generates a RESV message and sends it upstream along the path traversed by the PATH message. The RESV message contains a $Tspec$ to define traffic to be sent by the sender, and a $Rspec$ (reservation specification) for use by the RSVP-capable routers. This $Rspec$ comprises $R$, the bandwidth to be reserved at each router along the path and a slack term $S$. This slack term is the difference between the desired end-to-end delay and the actual one obtained with a reserved bandwidth of $R$. At intermediate routers, RESV messages store the reservation state if the request is accepted by the router.

### 2.2.2.2 Delay Bound Calculation for RSVP Connections

Using RSVP, the end-to-end delay upper bound of a connection can be calculated at the receiver, based on the specifications of PATH messages and the $Tspec$ of the RESV messages. The traffic’s $Tspec$ is a token bucket model $(b, r)$ with token bucket depth $b$ (bytes) and token rate $r$ (bytes/second). A traffic flow is then served by a connection of bandwidth $R$ (specified by the $Rspec$). The delay it would incur based on a fluid flow model is $b/R$ as long as the bandwidth is no less than the token rate ($R \geq r$) [15]. However, as the actual scheduling only approximates the fluid
flow model, the upper delay bound is in fact greater than $b/R$. The error terms $C$ and $D$ (specified by the Adspec) of a router represents the maximum variation of the queueing delay from the fluid flow model. The new delay bound (at an individual node) is determined as

$$\frac{b}{R} + \frac{C}{R} + D$$

(2.1)

At an IP router, which uses WFQ as a packet scheduler, $C$ is set to the maximum transfer unit (MTU) of the link (usually 1500 bytes) while $D$ is the time taken to process the biggest packet size within the flow. Then for a route from sender to receiver, the total error terms ($C_{total}$ and $D_{total}$) are determined as the sum of all the respective local error terms. The end-to-end delay caused by a bandwidth reservation of $R$ is shown in Equation 2.2. The analysis leading to this delay bound is based on a fluid flow model [19], [15].

$$\text{EndtoEndDelay} = \begin{cases} \frac{(b-M)(p-R)}{R(p-r)} + \frac{M+C_{total}}{R} + D_{total} & \text{for } p > R \geq r \\ \frac{M+C_{total}}{R} + D_{total} & \text{for } R \geq p \end{cases}$$

(2.2)

where $p$ is the peak rate the flow can introduce to the network, and $M$ is the maximum packet size.

The DiffServ specification does not include such simple delay bound calculations. In particular, the per-domain performance depends on the per-node performances received along the path. Hence, this motivates performance analyses of AF DiffServ nodes, which is the aim of this thesis.

2.2.3 Weighted Fair Queueing

In the IntServ/RSVP service model, bandwidth can be shared between connections by using the Weighted Fair Queueing (WFQ) algorithm for packet transmission. This queueing algorithm belongs to the work-conserving discipline, where the server is always busy if there are packets waiting in the queue.

WFQ can be used for guaranteed service since it can bound end-to-end delays under the fluid flow model assumptions made by a Generalised Processor Sharing (GPS) server [20]. When the server is ready to transmit at time $t$, it picks the first packet that would complete service in the corresponding GPS system [21].

Every arrival and departure of a packet from the GPS system is denoted as an event
(\(t_j\) is the time at which the \(j^{th}\) event occurs). For every busy period in the GPS system, the first event of that period is reset to 0 and the virtual time at that moment is also set to 0. During a time interval between the \((j - 1)^{th}\) event and the \(j^{th}\) event \((t_{j-1}, t_j)\), there is a fixed set of busy sessions denoted as \(B_j\). \(\{\phi_i\}\) is a set of positive numbers where each \(\phi_i\) is associated with a queue \(i\) and at a \(j^{th}\) interval, the bandwidth received by a non empty queue \(i\) is \(\frac{\phi_i C}{\sum_{k \in B_j} \phi_k}\) where \(C\) is the bandwidth of the link. Then the term virtual time \(V(t)\) is determined as follows [20], [21]:

\[
\begin{align*}
V(0) &= 0 \\
V(t_{j-1} + \tau) &= V(t_{j-1}) + \sum_{i \in B_j} \phi_i 
\end{align*}
\]

When the \(k^{th}\) session's \(i\) packet arrives at time \(a_{ik}\) and has a size of \(L_{ik}\), its virtual starting time and virtual finishing time \((S_{ik}^k, F_{ik}^k)\) are determined as follows:

\[
\begin{align*}
S_{ik}^k &= \max\{F_{ik}^{k-1}, V(a_{ik})\} \\
F_{ik}^k &= S_{ik}^k + \frac{L_{ik}}{\phi_i} \quad \text{with} \quad F_{ik}^0 = 0 \quad \text{for all} \quad i
\end{align*}
\]

The virtual finishing time is then used as the time-stamp of the packet. Packets are served in an increasing order of this time-stamp.

The order of complexity for WFQ is \(O(N^2)\) where \(N\) is the number of connections supported by the node. This leads to the poor scalability of the IntServ/RSVP model where a core router has to handle a large number of connections. In comparison with the queueing algorithm used in AF, WFQ is more complicated since AF algorithms only have to determine the probability to drop a packet with an order of complexity \(O(1)\). Furthermore, WFQ does not provide different levels of forwarding assurances for data packets.

### 2.3 DiffServ

Integrated Services aims to provide service guarantees to every network connection. There are two related issues that arise from the implementation of RSVP and IntServ:

- Overhead traffic and
A study in [22] considers a core router in a large ISP that support 10,000 Voice over IP flows using RSVP in each direction. Hence, the router has to maintain state information of every flow in additional to handling RSVP messages. This limits network scalability.

DiffServ was proposed as an alternative to IntServ. It aims to provide the same service to a group of connections with similar QoS requirements. This lowers the overhead, as network nodes handle only a small number of aggregations. Hence, DiffServ should scale well in a larger network. DiffServ is widely considered as the QoS mechanism for future IP networks. The following sections provide a thorough investigation on DiffServ aspects such as the elements, the service classes and other DiffServ concepts.

2.3.1 Architecture and Elements of Differentiated Services

End-to-end connections can be established over DiffServ or non-DiffServ networks (or regions). Each DiffServ network is a set of one or more contiguous DiffServ domains, where each DiffServ domain is a contiguous set of DiffServ nodes with a common service provisioning policy and a set of DiffServ packet forwarding treatments implemented on each node [12]. There are boundary nodes and interior nodes within a domain. They must be capable of providing packet forwarding treatments indicated by the DiffServ code point (DSCP). The DSCP is stored in DiffServ (DS) field of the IP packet header. A boundary node may be required to perform packet conditioning defined by a traffic conditioning agreement with its connecting domain while interior nodes with complex classification and conditioning are analogous to boundary nodes [12].

The traffic conditioning process is defined by a traffic conditioning agreement (TCA), which include classifier rules, corresponding traffic profiles and other rules for marking, metering, shaping and dropping. The traffic profile contains characteristic information of the traffic flow such as rate and burst size, which can be used to determine in and out-of profile packets. The TCA can be explicitly or implicitly derived from a service level agreement (SLA) between the service provider and the customer, which
Figure 2.1 A Differentiated Services network and its components

could be a user organisation or another DiffServ domain. The SLA specifies the forwarding service received by the customer.

A boundary node can be an ingress or an egress node. For an ingress node, it is responsible to ensure that the entering traffic conforms to any TCA between the domain and its connecting domain. However, it is optional to have an egress node responsible for such a task. Figure 2.1 describes a typical DiffServ network with its components such as DiffServ domains, SLA, TCA, boundary and interior nodes.

Upon arrival at a boundary node of a DiffServ domain, data packets will be formed into behaviour aggregates (BA) by means of a classifier and a conditioner (as shown in Figure 2.2). Packets belonging to a BA receive a distinct treatment as they cross a DiffServ network. The role of each element can be described as follows

- **Traffic Classifier**: The classifier helps identify the set of packets which receive differentiated service by being conditioned and/or mapped to one or more BA [12]. This classifying process is based on the content of the packets DSCP field of the IP packet headers (BA classifiers). A classifier can also classify packets based on DSCP and other IP header fields such as the source and destination addresses or port addresses (Multi-Field MF classifiers). The classified packets are then passed to conditioners to be processed with a traffic profile.
Traffic Conditioner: A conditioner may consist of a meter, a marker, a shaper and a dropper. The traffic condition is measured against the traffic profile (included in the TCA) by the meter. Packets are marked as in or out-of profile and have their DSCP set accordingly. A token bucket can be used to mark these packets. The Shaper/Dropper may be used to re-shape the traffic flow or to drop "bad" packets when necessary to bring the flow into compliance with its traffic profile.

In our research, delays arising from the classifying and conditioning processes are omitted. It is assumed that packets arriving to a DiffServ core network are already marked with DSCPs. After being formed, the BAs are forwarded to interior nodes and the forwarding treatment received at these nodes by a BA is defined as its Per-Hop Behaviour (PHB). In the early stage of DiffServ, there were two types of PHB, Assured Forwarding (AF) and Expedited Forwarding (EF) in addition to the default best-effort PHB. Recently, there is a new Proportional Forwarding PHB proposed in [23]. The focus of this research however is the performance of AF DiffServ routers with a variety of traffic models.

2.3.2 DiffServ Code Point

In Differentiated Services, each IP packet is given a DSCP to indicate the forwarding treatment given to the packet by a DiffServ router. The DSCP is determined by the required level of quality of service or is based on the traffic profile of the packet.
stream. Though the DiffServ working group’s PHB RFCs recommend a DSCP for each treatment, the mapping between DSCP and PHBs at a particular DiffServ domain is left to network implementors.

A DiffServ (DS) field, within the IP packet header, is used to store these DSCPs. For IPv4 data packets, DiffServ utilises the Type of Service (ToS) octet in the IP header to store the packet DSCP while IPv6 packets use the Traffic Class Octet to store their DSCP [24]. Normally the ToS field of an IPv4 header is used to indicate the type of service that an IP packet receives, in terms of delay, throughput and reliability [25]. However, in DiffServ, the ToS field becomes redundant and is used to store the packet’s DSCP. The DS field uses six bits for DSCPs with the remaining two bits currently unused. Hence, DiffServ can provide a maximum of 64 different forwarding treatments to traffic flows. At the moment, the Assured Forwarding PHBs require 12 code points, the Expedited Forwarding PHB and best-effort each uses only 1 code point while the newly proposed Proportional Forwarding PHB can have any number of code points out of the remaining ones.

DiffServ can be deployed across various domains with different link layer technologies such as ATM and MPLS. Where there is a dynamic connection establishment, it is important to map the QoS treatments at IP level (PHBs) to a protocol message of the link layer such as ATM signalling or MPLS label distribution protocol. An algorithm which does this is outlined in [26].

In a DiffServ/MPLS environment, when the IP header is invisible to a Label Switched Router, the packet DSCP is stored in the packet’s label. Hence, DSCPs are mapped to the MPLS label field. Later sections in this chapter will further discuss this issue.

2.3.3 Per-Hop Behaviour

A PHB is a “description of the externally observable forwarding behaviour of a DiffServ node applied to particular DiffServ behaviour aggregate” [12]. PHBs specify the way a DiffServ node allocates resources for the corresponding BA. Each PHB corresponds to a DSCP stored in the DS field of the packet’s IP header. They are implemented at DiffServ routers with buffer management and packet scheduling mechanisms. There are four types of PHBs defined including the default best-effort, Assured Forwarding, Expedited Forwarding and the recently proposed Proportional
Forwarding PHB. The RFC2475 went further to define the concept of PHB group which is "a set of one or more PHBs that can only be meaningfully specified and implemented simultaneously, due to a common constraint applying to all PHBs in the set such as queue servicing or queue management policy". This section will look at these forwarding behaviours in details.

2.3.3.1 Default PHB

The Default PHB can be used to forward conventional best-effort traffic. It must be available at any DiffServ-compliant node. Packets which belong to other PHBs but fail to conform to the traffic profiles can also be provided with Default forwarding. Default PHB packets are forwarded as long as the node's output interface is free from forwarding packets associated with any other PHB. The corresponding DSCP for this forwarding treatment is 000000 [27], [24].

2.3.3.2 Proportional Forwarding PHB

The Proportional Forwarding PHB [23] was introduced in February 2001 as an Internet draft. It aims to proportionally allocate bandwidth for competing flows based on their subscribed information rate (SIR) during congestion. The SIR is a key service parameter, where users who subscribe to a higher SIR pay more for the service. If PF is used for frame relay traffic, the frame relay committed information rate (CIR) is considered as the SIR. Proportional Forwarding uses SIRs to allocate bandwidth during congestion, following the max-min fairness algorithm described in [28].

If we denote the arrival rate of flow $i$ at time $t$ as $r_i(t)$ and its SIR as $SIR_i$, then the bandwidth allocated ($BW\text{\_Allocated}_i$) for that flow can be calculated as:

$$BW\text{\_Allocated}_i = \min(r_i(t), \alpha \times SIR_i)$$

and

$$\sum_i BW\text{\_Allocated}_i < \text{Available\_BW}$$

with $\alpha$ being the maximum fractional multiplier ($\alpha \in [0, 1]$) so that the inequality still holds. A flow experiences no loss when its arrival rate is less than the allowed throughput, while excess bandwidth is shared among the remaining flows proportionally to their SIRs.
Packets that arrive to a DiffServ network are marked with PF DSCPs at the boundary nodes. The DSCPs used for PF can be chosen from the remaining ones which have not been used by other PHBs. Hence, each DiffServ interior node has a multiple-RED queue where each set of RED parameters [29] corresponds to packets with a given code point.

The DSCP given to a packet within a flow is dependent on the ratio of the sending rate of the packet to the flow's SIR. Packets sent at lower rates will receive at least equal or higher DSCPs (i.e. higher priority or lower drop-precedence) than those which were sent at higher rates. When the sending rate of a flow exceeds its SIR for a long period, which is indicated by a maximum allowable burst, all packets should be marked with the lowest priority DSCPs. If there are $n$ DSCPs that packets from a flow can be marked with, each boundary node will require $n$ token buckets for this flow (each token bucket corresponds to a DSCP). For each flow, the sum of the token rates must be equal to the flow's SIR while the bucket depths handle the flow's bursts. Since the DSCP marking is based on the ratio of sending rate over SIR, packet sent at a rate less than or equal to the SIR are given preferred treatments by the node. Hence, network resources (i.e. bandwidth) is proportionally shared between flows based on their SIRs. The multiple-RED queue used by an interior node can be replaced by a TD one since the node drops packets based on theirs DSCPs (similar to AF). Algorithms used to mark PF packets are described in Section 3.2.3.

2.3.3.3 Expedited Forwarding PHB

EF is intended for low loss, low delay and low jitter (the variation between maximum and minimum delay) services. Hence, EF packets must usually encounter short or empty queues, so that packet loss is kept to a minimum. As a result, the service rate for an EF aggregate must exceed its arrival rate. The original definition of the EF PHB aims to provide a lower bound guarantee on the service rate $R$ over an appropriate time interval, which is independent of non-EF traffic. However, there are issues arising from this definition such as the time scale at which the service rate is measured or the inability to externally determine the status of an EF packet within the queue. A formal definition of EF is provided in the extended version RFC2598B [30] to take into account these issues. This definition assumes that "EF
packets should ideally be served at rate $R$ or faster” and it “bounds the deviation of the actual departure time of each packet from the ideal departure time of that packet” where the departure time of a packet is the time the last bit of that packet leaves the node [30]. The following terms are defined,

- $l_j$ is the size of the $j^{th}$ EF IP packet and $l_j/R$ is its transmission time at the configured rate $R$ respectively.

- $d_j$ is the actual departure time of the $j^{th}$ EF packet (i.e. the time at which the last bit of the packet departs from the node including the possible layer 2 trailer).

- $a_j$ is the arrival time of the $j^{th}$ EF packet (i.e. the time at which the last bit of the packet arrives at the node including the possible layer 2 trailer).

- $f_j$ is the ideal departure time for the $j^{th}$ EF packet.

- $E_a$ is the maximum deviation between actual and ideal departure time of an EF packet.

The ideal departure time of the $j^{th}$ EF packet can be calculated as

$$
\begin{cases}
    f_0 = 0 & d_0 = 0 \\
    f_j = \max(a_j, \min(d_{j-1}, f_{j-1})) + l_j/R, & \text{for all } j > 0
\end{cases}
$$

(2.5)

with the time origin chosen such that no EF packets are in the system at time 0. This recursive calculation of $d$ and $f$ can be obtained from the following conditions:

- If the $j^{th}$ EF packet arrives to a node to find the queue empty ($a_j > d_{j-1}$), its ideal departure time is the sum of its arrival and transmitting time $f_j = a_j + l_j/R$.

- If the $j^{th}$ EF packet arrives to the node which still contains other packets awaiting service and the scheduler is running “late” (i.e. the $(j - 1)^{th}$ packet’s departure occurs after its own ideal departure time or $a_j < f_{j-1} < d_{j-1}$), where its ideal departure time is $f_j = f_{j-1} + l_j/R$. 
• If the $j^{th}$ EF packet arrives to the node which still contains other packets awaiting service and the scheduler is running “early” (i.e. the $(j - 1)^{th}$ packet’s departure occurs before its own ideal departure time or $a_j < d_{j-1} < f_{j-1}$), where its ideal departure time is $f_j = d_{j-1} + l_j/R$.

The relationship between actual and ideal departure time is

$$d_j \leq f_j + E_a \quad \text{for all} \quad j > 0$$

The configure rate $R$ and error term $E_a$ indicate the characteristics of DiffServ EF nodes.

EF services provide preferential link access for given packet classes [31], [32] (while AF does not provide explicit performance guarantees, but rather a mechanism for differential treatment for various traffic classes). As a result, the EF PHB provides a more stringent performance guarantee than the AF PHB and is suited for more critical traffic types such as routing updates [33]. Examples of EF mechanisms are Class Based Queueing (CBQ) and Priority Queueing [11], [31], [34]. EF packets are marked with a DSCP of 101110.

2.3.3.4 Assured Forwarding PHB

The AF schemes offer different levels of forwarding assurance for data packets within a DiffServ domain. The IETF DiffServ working group has defined four AF classes, within each class there are 3 priorities or drop-precedences [27]. Each AF class is independently supported by a separate queue at a DiffServ router. A DiffServ router must be able to provide a configurable bandwidth for each AF class, where packets belonging to the same AF class are dropped based on drop-precedences. Within the same AF class, packets with higher drop-precedence must be dropped with a higher probability than those with lower drop-precedence, hence determining the relative loss performance of the DiffServ router. However, a flow with lower drop-precedence may experience a longer packet delay at a router since there will be a longer queue (packets from flows with higher drop-precedences either see shorter queues, or are dropped). Hence there is a trade-off between loss and delay performances. The AF packet treatment can be applied to a wide range of applications from TCP traffic (low
loss and delay tolerant) to real-time applications (low delay and relatively tolerant to loss) and the ATM VRB service [35]. Table 2.1 contains the DSCP of AF PHBs where AF_{xy} represent the DSCP of the y drop-precedence of the x AF class.

Since AF classes are implemented independently at DiffServ routers, each AF class is a PHB Group (i.e. the different drop-precedences comprise group members) while AF (in general) can be considered as a type of PHB group. Similarly, Expedited Forwarding is a type of PHB group which contains only one PHB.

A queueing mechanism to support AF must have different packet dropping levels such as Threshold Dropping (TD) or Random Early Detection with In/Out (RIO) [11].

### 2.3.3.5 Applications of Different DiffServ Forwarding Behaviours

The default best-effort, AF, EF and the newly proposed PF PHB are different types of forwarding behaviour, suited for different services. Best-effort does not provide any differential treatment, hence is used for conventional best-effort traffic. The most stringent forwarding, EF is used for more critical services which require low loss, low delay and low jitter such as control updates. Though less stringent than EF, AF is more flexible. It allows users to choose either a low delay and loss tolerant service (e.g. for interactive real-time applications) or a low loss and more delay tolerant service (e.g. TCP traffic). In Chapter 6, we consider an aggregate of Web and voice traffic with different AF drop-precedences. Similar to AF, PF uses a multiple-RED or a TD queueing algorithm to drop packets based on packet DSCPs and the queue length (average for RED and instantaneous for TD). PF is intended for services where bandwidth is shared among competing flows in proportion to their SIRs. It can be used for frame relay traffic with its CIR mapped to the SIR.
2.3.4 Per-Domain Behaviour

Apart from the Per-Hop Behaviours, the DiffServ working group also defined Per-Domain Behaviours (PDBs). The PHBs were defined to specify the forwarding treatment received by a group of data packets with the same QoS requirements at a DiffServ node. However, the future Internet will be implemented across different DiffServ domains, which are independently maintained by different network providers. Hence, it is important to specify the forwarding path attributes on a per-domain basis. This concept can be used to assist the deployment of per-domain QoS, which is a building block for end-to-end cross-domain QoS [36].

A PDB can be considered as "the expected treatment that an identifiable or target group of packets will receive from edge-to-edge of a DiffServ domain. A particular PHB (or, if applicable, list of PHBs) and traffic conditioning requirements are associated with each PDB" [36]. PDBs which specify different lists of attributes can also use the same PHB. For example, two PDBs that require 1% and 0.1% loss respectively can share the same PHB which corresponds to an attribute of "a maximum loss probability of 0.001". The attributes of a PDB can be the expected (measurable or statistical) loss, delay or jitter performance received. Packets arriving at the edge of a DiffServ domain will be classified to form a target group by a classifier. Then, the target group is passed through a traffic conditioner to create a traffic aggregate (collection of packets with DSCPs that map to the same PHB) which then transits the domain (see Section 2.3.1).

A best-effort PDB has been also defined with the expectation that "the packets of this PDB will not be completely starved and that they consume spare resources whenever available". In other words, the attribute of the best-effort PDB is "as much as possible, as soon as possible" [36]. The following sections will discuss the attributes and implementations of PDBs other than the default best-effort PDB such as the Bulk Handling PDB [37], the Virtual Wire PDB [38] and the Assured Rate PDB [39].

2.3.4.1 Bulk Handling PDB

The Bulk Handling (BH) PDB is for "sending extremely non-critical traffic across a DS domain or DS region" [37] where the network makes no commitment in trans-
mitting these packets (similar to the bulk handing delivery in US postal service). This PDB utilises the network's unused link capacity. Packets, which subscribe to the BH PDB, will experience delay and loss whenever there is not enough network resources under the presence of packets belonging to other PDBs. As for the per-hop forwarding treatment, it is implemented such that the node's interface is either forwarding BH traffic or remaining idle. Hence, there is no impact of BH traffic to other types of traffic. BH PDB traffic has the lowest precedence among other type of traffic, even lower than the best-effort PDB traffic. There are no attributes associated with this PDB while with the best-effort PDB, the network is committed to provide its available resources. Therefore, no traffic profile is needed to monitor and police this type of traffic by edge devices.

2.3.4.2 Virtual Wire PDB

This Virtual Wire (VW) PDB is equivalent to a dedicated circuit. It has two major attributes: an assured peak rate and a bounded jitter, so that it can be considered as "the same as a wire" [38]. Hence, the VW PDB can be achieved by implementing EF at every DiffServ node along the route. At the ingress node of the domain, traffic is policed so that its rate never exceeds the peak rate defined for the "wire". Within the domain, packets which exceed this peak rate at any intermediate node will be unconditionally discarded. All other packets are guaranteed to be transmitted.

2.3.4.3 Assured Rate PDB

The Assured Rate PDB suggested in [39], which extends the AF PHB to support flows across DiffServ domains, is still in its early development. It aims to provide a committed information rate (CIR) (e.g. to support Frame Relay) to an aggregate transiting a DiffServ domain. It is suitable for services that require a rate assurance but not delay or jitter bounds. This CIR is assured with a high probability, which can be defined in the service level specification. Hence, the attributes of the AR PDB can be stated as "a CIR of $x$ will be assured for $y\%$ of the time". The per-hop treatment of this PDB is essentially AF where an entire AF class is used to provide services to the flow. An AF class can be used to support different PDBs across a DiffServ domain. Moreover, packets from two different aggregates that subscribe to the same
PDB will be marked with DSCPs from the same AF class.

Flows that receive this AR PDB are tested against the edge rules, which mark packets with different levels of drop-precedence. The marking of data packets at the domain ingress node is based on the CIR of the flow. Since there are three drop-precedences within an AF class, extra parameters such as a peak information rate (PIR) can be used to classify packets, once the sending rate exceeds the CIR. For example, packets that conform to the CIR are marked with the lowest drop-precedence (AFx1) while those which were sent at a rate ranging from CIR to PIR or greater than PIR will be marked with AFx2 or AFx3 respectively.

Many aspects of the Assured Rate PDB are open to further research such as the estimations of loss, delay and jitter, the estimation of the CIR for different traffic arrival models or the effects of a sequence of AF nodes within a DiffServ domain on its performance. This thesis analyses the performance of AF nodes, which are the building blocks of the Assured Rate PDB. These performance analyses determine loss and delay performance of individual AF nodes, and can be used in future research on AR PDB performance.

### 2.3.5 DiffServ Queueing Mechanisms

This section considers the scheduling mechanisms used for Differentiated Service Assured Forwarding PHB. These are Random Early Discard with In/Out profile and Threshold Dropping (which is a special case of the former). These mechanisms and their performance will be investigated in later chapters of the thesis.

#### 2.3.5.1 Random Early Discard with In/Out profile - RIO

RIO is an enhanced version of a RED queue [9], where the buffer management algorithm contains two RED queues corresponding to In and Out profile packets. These RIO queues drop packets based on the average queue length at the moment of packet arrivals. Similar to RED queues, RIO is a congestion and global synchronisation avoidance algorithm which is suitable for TCP traffic [40]. Furthermore, it avoids bias against bursty sources. Figure 2.3 describes the packet drop probability of a RIO queue based on the average queue length. The threshold values and maximum dropping probability for In and Out packets are \((\text{Th}_{\text{In min}}, \text{Th}_{\text{In max}}, p_{\text{in max}})\) and
Figure 2.3 Packet drop probability of a RIO queue based on the average queue length

(Th\textsubscript{Out\_min}, Th\textsubscript{Out\_max}, p\textsubscript{out\_max}) respectively. For an In profile packet, upon its arrival at the RIO queue, it is dropped with a probability determined from Figure 2.3 when the average queue length exceeds the Th\textsubscript{In\_min}. As the average queue length increases, In profile packets are dropped at a higher probability. This probability increases linearly until the average queue length approaches the maximum threshold Th\textsubscript{In\_max} and the maximum dropping probability is p\textsubscript{out\_max}. After this threshold, In profile packets are always dropped. The decision to drop Out profile packets is similar to that of In profile packets. To implement an AF class with three drop-precedences the RIO algorithm is expanded with an extra set of RED parameters so that there are three RED parameter sets corresponding to the three drop-precedences (i.e. a 3-RED queue). Subsequent RIO queues in this thesis, when used in an AF context, are of this type.

2.3.5.2 Threshold Dropping Queue - TD

The Threshold Dropping (TD) queue is a special case of the RIO queue. Instead of having two sets of parameters, each for In and Out profile packets, the Threshold Dropping queue has multiple thresholds, each corresponding to a drop-precedence. While RIO queues drop packets based on the average queue length, a TD queue makes the decision based on the instantaneous queue length. As a result, the TD
A packet drop probability of a TD queue based on the instantaneous queue length is described in Figure 2.4. Each drop-precedence is associated with a threshold. If the queue length exceeds the threshold, then all packets associated with that threshold will be dropped. This queueing mechanism is an on-off mechanism, and hence may exhibit oscillatory performance behaviour [41].

Chapters 4 and 5 present analyses of TD and RIO queues with Poisson and MMPP traffic respectively. The RIO queue considered drops packets based on the instantaneous queue length instead of the average queue length. This assumption enable a mathematically tractable solution to the RIO queueing problem. A simulation study in Chapter 6 investigates the steady-state performances of a RIO and its associated TD queue. In particular, we wish to determine if a TD queue can approximate the performance of its corresponding RIO queue. If so, then the TD queue, with a simpler implementation can be used instead of the RIO one.
2.4 DiffServ and MPLS

A DiffServ network operating over MPLS combines the advantages of these two concepts. In such networks, behaviour aggregates are grouped into DiffServ classes and their packets are given labels that allow them to be switched effectively. This section outlines the key elements of MPLS [42] then shows how they can be applied to DiffServ [10].

2.4.1 MPLS Concepts

The IETF MPLS working group has defined how network nodes can pass packets with the same “tag” efficiently in either IP or ATM networks. The working group is also considering traffic management aspects, similar to Differentiated Services ones.

Label:
In connectionless networks, when a packet travels from one node (router) to another, the node makes the required forwarding decision. Usually, this decision is based on the destination IP address. In MPLS, the router categorises packets into Forwarding Equivalence Classes (FECs) and maps each FEC to a next hop. A packet is assigned to a particular FEC once it enters an MPLS network. This FEC is encoded as a short fixed length value known as a “label”. The label is forwarded along with the packet to its next hop. At the next hop, the label is used to specify the subsequent hop and a new label instead of the packet’s network layer header. This new label replaces the old one and sent along with the packet to its next hop. However, a labelled packet can carry a number of labels referred as a “label stack” with the new label replaces the old ones or is pushed on top of the stack. This label stack is organised as a last-in, first-out with the top label determine the process of the packet. This stack starts with a level 1 label at the bottom of the stack.

Label Swapping:
Label Swapping is the process which allows “streamlined forwarding of data by using labels to identify classes of data packets which are treated indistinguishably when forwarding” [42]. Before making the forwarding decision for a packet, the Label Switched Router (LSR) examines label stack of the packet. If the stack is not empty (i.e. a labelled packet), an Incoming Label Map (ILM) and the label at the top of the
label stack are used to map to a Next Hop Label Forwarding Entry (NHLFE). If the stack is empty (i.e. an unlabelled packet), the LSR then analyses the packet’s network layer header, determines its associated FEC and maps this FEC to an NHLFE using a FEC-to-NHLFE Map (FTN). The NHLFE specifies the packet’s next hop and one out of the following operations to perform on its label stack [42]:

- Replace the label at the top of the label stack with a specified new label
- Pop the label stack (illegal for the unlabelled packets)
- Replace the label at the top of the label stack with a specified new label, and then push one or more specified new labels onto the label stack.

It may also contain some information on:

- The data link encapsulation to use when transmitting the packet
- The way to encode the label stack when transmitting the packet
- Any other information needed in order to properly dispose of the packet.

The LSR then encode the new label stack into the packet, and forwards it downstream.

**Label Switched Path (LSP):**
A Label Switched Path (LSP) of level $m$ for a particular packet is a sequence of LSRs, the first of which pushes on a level $m$ label. This level $m$ label is used to forward the packet by all LSRs along the LSP. This path ends at an LSR which makes the forwarding decision based either on a lower than $m$ level label or an network layer header.

**Penultimate Hop Popping (PHP):**
The definition of a level $m$ LSP allows the label stack to be popped at the penultimate LSR rather than the egress LSR of the LSP. This does not affect the transmission of the packet from the penultimate LSR to the egress LSR, since the decision to send the packet to the egress node is made based on the level $m$ label before it is popped.
off the stack. Hence, the egress node of the LSP does not have to be an LSR. Penulti­
mate Hop Popping (PHP) allows the penultimate hop to determine if it is indeed the
penultimate hop and thus prevents the egress LSR from double lookups of the label
(or IP address).

2.4.2 DiffServ / MPLS Networks

This section outlines the key aspects of a DiffServ/MPLS network and compares
them with MPLS equivalents. Some new terminologies are introduced to for such
networks such as Ordered Aggregates (OA) and PHB Scheduling Class. These ter­
minologies are introduced to prevent an LSR, which was also a DiffServ node from
“discriminating between packets of an AF BA based on drop-precedence and for­
warding packets of the same AF class but different drop precedence over different
LSPs” [43].

- A PHB Scheduling Class is a PHB group for which a common constraint is
  that ordering of at least those packets belonging to the same microflow must
  be preserved. A microflow is defined as “a single instance of an application-to-
  application flow of packets which is identified by source address, source port,
  destination address, destination port and protocol id” [12].

- An OA is a set of BA with a common ordering constraint. Hence, the set of
  PHBs that are applied to this set of BAs constitutes a PHB scheduling class.
The OA definition only applies to AF BA since packets of the same microflow
  must not be re-ordered if they belong to the same AF class [27].

The key processes in an MPLS network are to determine the LSP at the ingress
node and then swap labels at intermediate LSRs. In a DiffServ/MPLS router, it
becomes more complicated, since the PHB of incoming and outgoing packets must
be determined. This section examines this process.

2.4.2.1 LSPs in DiffServ MPLS Network

A DiffServ network running over MPLS uses two types of LSP: E-LSP (EXP-Inferred
LSP) and L-LSP (Label-Inferred LSP) [10].
• E-LSPs are those that can transport multiple OAs. The 3-bit EXP field of the MPLS Shim Header indicates the PHB that will be applied to the packets at the LSR. As the result, an E-LSP can support up to 8 BAs.

• L-LSP can only support a single OA. At the LSR, a packet’s forwarding treatment is determined exclusively from the packet’s label value while the drop-precedence is indicated by the EXP field or by the encapsulating link layer specific selective to drop mechanism when the Shim Header is not used (e.g. MPLS over ATM).

2.4.2.2 Label Forwarding Model for DiffServ LSRs

In addition to label swapping, determining the operations on the label stack and encoding the new label stack into the packet, a DiffServ LSR is also responsible for the determination of the incoming PHB and the outgoing PHB during the label forwarding. A model for label forwarding [10] is described in Figure 2.5 with

• “Encaps” designates the DiffServ related information encoded in the MPLS encapsulation layer (e.g. EXP field, ATM CLP, Frame Relay DE, 802.1 User-Priority).

• (*) when the LSR behaves as an MPLS ingress node and the incoming packet may be received unlabelled.

• (**) when the LSR behaves as an MPLS egress node and the outgoing packet may be transmitted unlabelled.

This label forwarding model of a DiffServ LSR is suggested by the IETF MPLS working group and it consists of four stages [10]:

1. Incoming PHB Determination: determines which BA the received packet belongs to (i.e. its PHB). There are two methods to determine the Incoming PHB of the packet, in which the either label stack or the IP packet header is used. Encaps may be used in this stage if the incoming packet is unlabelled.
2. Outgoing PHB Determination with Optional Traffic Conditioning: determines the outgoing PHB with optional traffic conditioning. When there is no traffic conditioning, the outgoing PHB determination is identical to the incoming PHB determination.

3. Label Swapping: as described in Section 2.4.1 where the LSR must choose one of the NHLFEs indicated by the outgoing PHB. In this stage, the ILM and the FTN are populated with DiffServ Context information to determine the Encaps to be encapsulated in the outgoing label. A DiffServ Context for a label is defined as comprising the LSP type, a set of supported PHBs, an Encaps→PHB mapping for incoming labels and a set of PHB→Encaps mappings for an outgoing label.

4. Encoding of DiffServ Information into Encapsulation header: determines how the DiffServ information is encoded into the packets. Similar to stage one, there are two types of encoding where it is encoded into the label stack or the IP packet header. If the outgoing packet is unlabelled, Encaps will be generated based on the outgoing PHB.

2.4.2.3 DiffServ Tunnelling Models Over MPLS

For normal DiffServ networks, tunnelling is done by encapsulating IP traffic in another IP header [44]. However, for DiffServ/MPLS networks, the LSPs can also be
considered as tunnels. The LSPs are unidirectional with the intermediate LSRs only performing operations on the DiffServ information on the top of the stack (this is similar to IP tunnels, where intermediate routers only see and operate on the outer IP header). There are two proposed models for DiffServ/MPLS tunnelling; the Pipe model (primary and must be supported) and the Uniform model (secondary and optional) [10].

In the Pipe model, the LSP egress node only considers the Tunnelled DiffServ information to forward downstream. This information is stored in the inner header (i.e. network layer header) and is not meaningful to intermediate nodes. These intermediate nodes utilise the LSP DiffServ information to apply the forwarding treatment based on it. This information is stored in the outer header (i.e. MPLS header) and has no meaning beyond the LSP egress. Changes in this information, by possible traffic conditioning at intermediate nodes, are ignored at the LSP egress. This model is appropriate in cases when the interfaces to the LSP ingress and egress of a DiffServ domain use a common set of DiffServ policies and PHBs.

In the Uniform model, a packet only contains one piece of DiffServ information. This information is encoded in the outer header. Other DiffServ information encoded elsewhere in the stack is not considered and ignored by either the intermediate nodes or the egress node. If the traffic conditioning processes alter the DiffServ information, these changes will be reflected at the LSP egress node. In this model, the MPLS network is transparent to the DiffServ operations, since they are exactly the same as if MPLS were not employed. The Uniform model is useful in situations where the traffic conditioning agreement only operates at the DiffServ domain boundaries.

2.4.2.4 Mapping of EXP and PHB

As mentioned earlier, at the boundary of DiffServ and MPLS domains, it is important to map the DSCP of the Behaviour Aggregates to MPLS service classes. The DSCP is mapped with the EXP (Experimental bits) field of the MPLS Shim Header. The mapping process to determine the incoming PHB is regarded as EXP\(\rightarrow\)PHB mapping. For labels which correspond to an E-LSP for which no EXP\(\rightarrow\)PHB mapping is explicitly signalled at LSP setup, the EXP\(\rightarrow\)PHB mapping is based on the pre-configured EXP\(\rightarrow\)PHB mapping. This pre-configured mapping must be consistent
Table 2.2 EXP to MPLS Service Mapping

<table>
<thead>
<tr>
<th>MPLS Service Class</th>
<th>MPLS FCI</th>
<th>MPLS DPI</th>
<th>Drop Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Control</td>
<td>11</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>Gold</td>
<td>11</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Silver</td>
<td>10</td>
<td>1</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0</td>
<td>High</td>
</tr>
<tr>
<td>Bronze</td>
<td>01</td>
<td>1</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>01</td>
<td>0</td>
<td>High</td>
</tr>
<tr>
<td>best-effort</td>
<td>00</td>
<td>1</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>00</td>
<td>0</td>
<td>High</td>
</tr>
</tbody>
</table>

at every E-LSP throughout the domain. Meanwhile, for those labels that have been explicitly signalled at LSP setup, the EXP→PHB mapping is based on the signalled EXP→PHB mapping. The outgoing PHB determination process is similar to the incoming one.

In [45], the authors present an example mapping of the EXP field of the MPLS Shim header and DiffServ PHBs. This EXP field comprises of a Forwarding Class Indicator (FCI) and a Drop-Precedence Indicator (DPI). The MPLS service classes are categorised as Gold, Silver, Bronze and best-effort. These service classes are mapped with EXP values and DiffServ classes, as shown in Tables 2.2 and 2.3. In this model, the Gold class consists of a low loss, low latency and low jitter service. Drop-precedence is not defined for Gold class service. Packets which are sent in excess of a peak rate (the rate at which the network guarantees to deliver with minimum delay requirement) are dropped. This service class is suitable for delay-sensitive traffic.

Meanwhile, the Silver and Bronze classes are designed for throughput sensitive traffic. Each class has two drop-precedence levels and is mapped with two AF classes. Out of profile packets are treated as best-effort and correspond to the Default service class in DiffServ.

Once packets are switched to their corresponding queues, the MPLS nodes use the same queueing mechanisms as in the case of normal IP routers. Hence, AF queueing mechanisms such as RIO and TD can be used to implement AF treatments at MPLS nodes. Though DiffServ/MPLS networks are not investigated in our research, the
Table 2.3 DiffServ and MPLS Mapping

<table>
<thead>
<tr>
<th>DiffServ Class</th>
<th>MPLS PHB</th>
<th>MPLS EXP Field</th>
<th>MPLS Service Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHB</td>
<td>DSCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EF</td>
<td>101110</td>
<td>110</td>
<td>Gold</td>
</tr>
<tr>
<td>AF11</td>
<td>001010</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AF12</td>
<td>001100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AF13</td>
<td>001110</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AF21</td>
<td>010010</td>
<td>101</td>
<td>Silver</td>
</tr>
<tr>
<td>AF22</td>
<td>010100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AF23</td>
<td>010110</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AF31</td>
<td>011010</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AF32</td>
<td>011100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AF33</td>
<td>011110</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AF41</td>
<td>100010</td>
<td>011</td>
<td>Bronze</td>
</tr>
<tr>
<td>AF42</td>
<td>100100</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>AF43</td>
<td>100110</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>DF</td>
<td>000000</td>
<td>000</td>
<td>best-effort</td>
</tr>
</tbody>
</table>

performance analyses presented in the following chapters can aid the dimensioning of a DiffServ/MPLS node.

2.5 Conclusions

This chapter has identified DiffServ as the service model to be implemented in future networks. It replaces the IntServ model where networks aim to provide QoS performance for individual connections using RSVP, hence the amount of overhead traffic and poor scalability. The architecture and elements of a DiffServ network such as Per-Hop Behaviours and Per-Domain Behaviours are considered.

Per-Hop Behaviours are the building blocks of the Per-Domain Behaviour as well as the primary elements of DiffServ networks. Per-Hop Behaviours describe the treatment received by a packet upon its arrival at a DiffServ node. Different PHBs, such as default best-effort, AF, EF and a newly introduced PF, can be applied to applications with different QoS requirements. This thesis focuses on the Assured Forwarding PHB.
The AF PHB is suited to a wide range of applications where users can choose a low loss, delay tolerant service (e.g. Web traffic) or a low delay, loss tolerant service (e.g. voice traffic). The queueing mechanism to implement AF at a DiffServ node may be either RIO or TD.

In order to determine the end-to-end performance of a flow, which may traverse several DiffServ domains, individual per-domain performances must be determined. This leads to the determination of the per-hop performance of individual DiffServ nodes along the path based on the traffic profile and the provided network resources. This thesis aims to provide such calculations for AF PHB through performance analyses with a variety of traffic models such as Poisson and MMPP.
Chapter 3

DiffServ Performance Analyses

Since the introduction of DiffServ network architecture and implementations (i.e. Per-Hop Behaviours) there have been some studies to investigate the performance of DiffServ networks. As stated in the previous chapters, our research aims to provide an AF DiffServ performance analysis, which is useful for the dimensioning of DiffServ nodes to meet a desired performance. Hence, this chapter describes existing AF analyses including theoretical analyses (Section 3.1) and simulation studies (Section 3.2). We highlight the need for AF performance analyses which can accurately estimate packet loss and delay at a DiffServ node, where packets belong to a given AF class (i.e. implemented at a separate queue) may have up to three drop-precedences.

3.1 Theoretical Analysis

The existing theoretical analyses of DiffServ performance, including an analysis of a AF node (TD and RIO) with Poisson arrivals [11], an analysis of a TD node with MMPP arrivals [13] and an analysis of AF nodes with wireless links [46] are described in this section. The packet service time distribution considered in these analyses is exponential. We see that these analyses only consider two different priority flows, and hence cannot be applied to the case of an AF node with three drop-precedences. Furthermore, the MMPP arrival model considered in [13] is a simple 2-state MMPP model, rather than a more general aggregate of MMPP flows.
3.1.1 TD and RIO Queue Performance with Poisson Traffic

In [11], the authors studied the performance of Assured Service (or Assured Forwarding AF) and Premium Service (or Expedited Forwarding EF) on traffic with Poisson arrivals with an exponential service time distribution. The queueing mechanisms considered in this study are RED with In/Out profile (RIO) and Threshold Dropping (TD) for Assured Service and Priority Queueing for Premium Service.

The states of the queue form a Birth-Death process with the death rate is equal to the service rate and the birth rate determined by the current queue size and the Poisson arrival rate. The steady-state distribution of the queue length and hence the packet loss probability and expected delay, can easily be computed for this process. The expected delay calculation presented in this analysis takes into account dropped packets. For lightly loaded conditions, the number of dropped packets is small compared to transmitted packets and this calculation approximates the expected delay of transmitted packets. However, more packets are dropped as the load increases and hence this delay calculation becomes inaccurate. This oversight needs to be adjusted. This analysis only considers the case of two priority flows (tagged and non-tagged). Given that an AF class consists of three drop-precedences, the model needs to be extended to consider more than two priority flows.

The authors also demonstrate that there is little difference in the performance of a RIO queue based on the number of tagged (high priority) packets in the queue or the total number of packets in the queue as suggested in [9]. Furthermore, [11] shows a good correlation between the Poisson analysis and simulation results obtained with long range dependent traffic.

Chapter 4 presents an extended analysis which considers a TD queue with multiple Poisson traffic flows and an exponential packet size distribution (as considered in [11]). This analysis includes the expected delays calculation without dropped packets, correcting the oversight in [11]. Simulation results were obtained to verify this analysis and to determine how closely the Poisson results track different sized MMPP aggregations.
3.1.2 TD Queue Performance with MMPP Traffic

In earlier literature, there is some research on the performance analysis of a TD queue with MMPP traffic [13]. The MMPP traffic model in [13] is a two-state MMPP with an exponential packet size distribution while the TD queue has two thresholds, corresponding to the high and low priority traffic flows. The packet loss probability and expected delay can be calculated from the queue length distribution vector \( \pi \) and the infinitesimal generator matrix \( Q \). This distribution vector and infinitesimal generator satisfy the following equations:

\[
\pi \ast Q = 0
\]

and

\[
\pi \ast e = 1
\]

where \( e = (1, \ldots, 1)' \) is a column vector of \( 4(K + 1) \) elements. The following notations are defined to construct the matrix \( Q \):

- \( Y_A(t), Y_B(t) \) and \( Z(t) \) describe the states of stream A and B and the number of packets in the system including a server and a buffer at time \( t \), respectively. Hence, \( Y_A(t) \) can take a value representing the phase of the first MMPP, \( Y_B(t) \) can take a value representing the phase of the second MMPP and \( Z(t) \) can take on values from 0 to \( K \) (i.e. the buffer is empty or full).

- \( \pi(i, j, q) \) is the limiting distribution for the Markov process \( \{ Y_A(t), Y_B(t), Z(t) \} \). \( \pi(i, j, q) \) is the probability that the two MMPP flows are at states \( i (i = 1, 2) \) and \( j (j = 3, 4) \) and a queue length of \( q \).

- \( \pi(i, q) \) is the limiting distribution of having the first MMPP in phase \( i \) and a queue length of \( q \).

- \( r_{ij} \) is the transition rate of an MMPP flow's arrival process from state \( i \) to state \( j \).

- \( i \) represents the complementary state of \( i \).

- \( \lambda_i \) is the Poisson arrival rate when the process is in state \( i \).
• $\mu$ is the service rate of the queue. We assume that the packet service times of all flows are exponentially distributed with a mean $1/\mu$.

• $K$ is the queue’s buffer size.

• $\theta$ is the buffer threshold. Low priority packets are discarded when the queue length exceeds $\theta$ upon their arrivals.

Hence, the infinitesimal generator $Q$ is given by:

$$Q_{(i,j,q)\rightarrow(i',j',q')} = \begin{cases} 
  r_{ii'} & \text{if } i' \neq i, \ j' = j, \ q' = q \\
  r_{jj'} & \text{if } i' = i, \ j' \neq j, \ q' = q \\
  \lambda_i + \lambda_j & \text{if } i' = i, \ j' = j, \ q' = q + 1, q < 0 \\
  \lambda_i & \text{if } i' = i, \ j' = j, \ q' = q + 1, q \geq 0 \\
  \mu & \text{if } i' = i, \ j' = j, \ q' = q - 1 \\
  -r_{ii} - r_{jj} - \lambda_i - \lambda_j & \text{if } i' = i, \ j' = j, \ q' = 0 \\
  -r_{ii} - r_{jj} & \text{if } i' = i, \ j' = j, \ q' = K \\
  -\mu - r_{ii} - r_{jj} - \lambda_i - \lambda_j & \text{if } i' = i, \ j' = j, \ q' = q, 0 < q \leq \theta \\
  -\mu - r_{ii} - r_{jj} - \lambda_i & \text{if } i' = i, \ j' = j, \ q' = q, \theta < q < K \\
  0 & \text{otherwise}
\end{cases}$$

In [13], the authors provide the formulae to estimate the loss probability of the high priority stream (A) and low priority stream (B) as:

$$P_{\text{loss}}(A) = \frac{(r_{12} + r_{21})(\lambda_1 \pi(1, K) + \lambda_2 \pi(2, K))}{\lambda_1 r_{21} + \lambda_2 r_{12}}$$

(3.2)

and

$$P_{\text{loss}}(B) = \frac{(r_{34} + r_{43}) \left( \lambda_3 \sum_{q=\theta}^{K} \pi(3, q) + \lambda_4 \sum_{q=\theta}^{K} \pi(4, q) \right)}{\lambda_3 r_{43} + \lambda_4 r_{34}}$$

(3.3)

However, the analysis presented in [13] does not include the calculations to estimate the expected packet delay. Also, these calculations do not apply for multiple MMPP flows where each flow is modelled by a multi-state MMPP.

Studies in [5] and [6] show that for a finite buffer queue, the effect of correlation (as shown by most network traffic [2], [3], [4]) becomes insignificant beyond a time scale. This time scale is determined by the correlation structure of the input traffic.
and the maximum queue size. Hence, any Markovian model such as MMPP can be used as long as the chosen model captures the correlation structure up to the correlation horizon [5]. Furthermore, the simulations study in Section 4 shows that the Poisson based analysis is only suited to approximate calculations of large MMPP aggregates. Therefore, an MMPP based analysis is required to provide accurate performance calculations of MMPP aggregates. The MMPP model considered must have multiple states, since an aggregate of MMPPs is another MMPP with more states. Chapter 5 presents an extended analysis of a TD queue with multiple multi-state MMPP traffic flows with exponential packet size distribution. The process to determine the number of states used and MMPP variables is not considered in the scope of this thesis. These improvements enhance the analysis presented in this section. The packet expected delay for each priority flow is also determined.

3.1.3 Modelling of AF performance on Wireless Links

In [46], the authors provide a model of AF PHB over wireless links. They consider a simulation model of IP packets (with 2 classes) transmitted from a base station to a mobile station. This is done by means of the Radio Link Protocol (RLP) located between the IP layer and the link layer. The error correction scheme implemented by the RLP is a hybrid ARQ/FEC [47] model (Automatic Repeat Request/Forward Error Correction).

The wireless link is modelled with the Gilbert-Elliot model often used for slowly fading channels [48]. This model is Markovian with two states where the time spent in each state is exponentially distributed. DiffServ is provided at the base station and modelled by a RIO queue based on instantaneous queue lengths. The service time (i.e. packet size) distribution is exponential while the packet arrival process is assumed to be Poisson or a superposition of 32 independent exponentially distributed on-off sources. The state transition diagram of this queueing problem is 2-dimensional since the wireless channel is a Markovian with two states. However for each state of the wireless channel (good or bad) the transition processes themselves are Birth-Death processes with variable rates as similarly considered in [11] and [49].

It is established that simulation results with a wireless link closely match with an-
alytical results. Since this model is simple and easy to compute, it can be used by designers to evaluate the effect of the link's parameters on its performance.

From a mathematical perspective, the model presented in [46] is similar to that in [11], [49] (our published work) with an extra dimension added to the transition diagram due to the presence of the Gilbert-Elliot model for slowly fading channels. Nevertheless, it only considers 2 traffic classes and needs to be extended to consider three drop-precedences within an AF class. Analysis of this model with a more complicated traffic model is an open issue for further research.

3.2 Simulation Studies

This section describes simulation studies on Assured Forwarding as well as the performances of some marking schemes proposed for proportional forwarding. These simulation studies consider the interaction of AF and EF, the effect of the number of drop-precedences and marking schemes of for PF packets.

3.2.1 Interactions between AF and EF

In [50], a DiffServ network with one link is designed to provide Expedited Forwarding for video-conferencing (CBR) traffic, Assured Forwarding for FTP traffic and best-effort for UDP traffic. Packets subscribed for AF are marked as In and best-effort traffic is marked as Out. These In and Out packets are processed by a RIO queue (lower priority queue) while EF packets queue at a separate queue with higher priority. The EF queue and the RIO queue share bandwidth based on either Weighted Round Robin (WRR) or Priority Queueing (where the lower priority queue starts processing only when the higher priority one is empty). In other words, the active queue management of this link is either a combination of Priority Queue (PQ) and RIO or a combination of WRR and RIO. The packet marking algorithm is Time Sliding Window (TSW) where non-conforming EF packets are dropped (to reduce the delay and jitter) while non-conforming In packets are marked as Out. Out packets are not considered by these markers.

Simulations show that in a PQ/RIO network with strictly policed EF traffic, while the
AF flow is protected from the EF flow, the best-effort UDP flow suffers significantly. Both AF and best-effort flows are starved of network bandwidth if the EF flow is not policed. This problem can be resolved by using the WRR/RIO model where there is a trade-off between the throughput of AF/best-effort flows and the delay and jitter of the EF flow.

However, the interaction between the AF traffic and best-effort traffic (i.e. In and Out traffic) for the case of WRR/RIO or the interaction of different drop-precedences in general remains an open issue. Our analyses (Chapters 4 and 5) determine the performance of an AF node on different drop-precedences and can be used to address this issue.

### 3.2.2 Effect of Number of Drop-Precedences

The study presented in [51] compares the performance of the AF packet forwarding (implementing a RED queue) with two and three drop-precedence levels for mixed TCP/UDP traffic to investigate the fairness in allocating excessive bandwidth. The simulation results, based on 1296 simulations with different parameter/traffic pattern combinations with congestion sensitive (TCP) and insensitive (UDP) traffic. Results show that the optimal number of drop-precedence levels in a traffic class depends on the traffic load and its committed information rate. If a link operates at a low load, three drop precedences are needed to provide fairness in sharing the excessive bandwidth. Therefore, in order to prevent congestion insensitive traffic (such as UDP) from being “greedy” during a reduction in the rate of congestion sensitive traffic (such as TCP), out-profile packets of these traffic must be marked differently, hence three drop-precedences. For a link which operates close to its capacity, three levels of drop-precedences are redundant since the excessive bandwidth to be shared is small.

This is the reason why the DiffServ working group defined an AF class with three drop-precedences. Our analyses reflect this choice by extending existing work to consider a general number (more than 2) of priority flows.
3.2.3 Proportional Forwarding Marking Schemes

The previous chapter considers a new DiffServ packet forwarding behaviour (Proportional Forwarding, PF) which aims to share a link's bandwidth fairly based on the subscribed information rate (SIR) of the flows sharing the link during a congestion period. This PF PHB is implemented by an n-RED queue with n sets of RED parameters. A packet will be given a DSCP by marking devices at the edge of the network. This DSCP is based on the ratio of the sending rate of the packet to the flow's SIR. There is a token bucket associated with each DSCP, and the sum of the token bucket rate equals the flow's SIR (each token rate is a fraction of the SIR). The relationship between token rates, SIR and fractions can be described as:

\[
\begin{aligned}
\text{TokenRate}_i &= f_i \times \text{SIR} \\
\sum_{i=1}^{n} f_i &= 1
\end{aligned}
\]

In [23], the authors suggest three algorithms to determine these fractions (hence the token rates). These algorithms are Equal fractions, Arithmetic Progressive fractions and Geometric Progressive fractions and all marking devices use the same algorithm throughout the domain.

3.2.3.1 Equal Fractions

In this algorithm, all the fractions are of equal value, i.e. for a PF PHB with n DSCP values, each fraction \( f_i \) is \( \frac{1}{n} \).

3.2.3.2 Arithmetic Progression (AP) Fractions

In this algorithm, the sequence \( \{f_i\} \) forms an arithmetic progression. For simplicity, the fractions which relate the token rates to the SIR can be shown as \( \{d, 2d, 3d, \ldots, nd\} \). From the condition that the sum of these fractions is equal to 1, \( d \) can be determined as,

\[
d = \frac{2}{n \times (n + 1)}
\]
3.2.3.3 Geometric Progressive (GP) Fractions

In this algorithm, the sequence \( \{f_i\} \) forms a geometric progression. For simplicity, the fractions which relate the token rates to the SIR can be shown as \( \{r, r^2, r^3, \ldots, r^n\} \). From the condition that the sum of these fractions is equal to 1, \( r \) can be found as to satisfy the equation

\[
r(2 - r^n) = 1
\]

The proportionality index which measures how the link capacity is shared between flows during congestion can be determined as:

\[
\text{PI} = \frac{\left( \sum_{i=1}^{n} \text{AR}_i \right)^2}{n \times \sum_{i=1}^{n} \text{AR}_i^2}
\]

where the allocation ratio for the \( i^{th} \) flow

\[
\text{AR}_i = \frac{\text{TR}_i}{\text{SR}_i}
\]

with the throughput ratio and SIR ratio of the \( i^{th} \) flow defined as

\[
\text{TR}_i = \frac{\text{Throughput of the } i^{th} \text{ flow}}{\sum_i \text{Throughput of all flows}}
\]

and

\[
\text{SR}_i = \frac{\text{SIR of the } i^{th} \text{ flow}}{\sum_i \text{SIR of all flows}}
\]

Simulations results in [23], with different number of flows, show that the \( n \)-RED queue's performance is much better with the implementation of the three algorithms since the marker has knowledge of the flow's SIR. The performances (in terms of fairness) of the queue with these three marking algorithms are very similar. However, as the network becomes severely congested, the equal fraction algorithm does not perform as well as the other two algorithms.

It can be seen that fairness in performance of an PF node significantly dependent on the marking of data packets with respect to its SIR. Significantly, the queueing algorithm of an PF node is similar to that of an AF node (i.e. both of them can
be implemented by a RIO queue with multiple set of RED parameters). Hence, the analyses provided in this thesis with multiple flows with different priority can be applied to PF nodes with multiple DSCPs.

### 3.3 Conclusions

This chapter has studied the existing performance analyses of the Assured Forwarding PHB and a similar one in the PF PHB. Though there are various simulation studies on the performance of Assured Forwarding with mix of TCP/UDP connections, there is a distinctive lack of theoretical analysis of AF performance based on traffic parameters. Some papers [11], [13] have considered this issue yet they either lack or provide insufficient analysis to investigate the performance of an AF node with multiple drop-precedences (which is the case when AF or PF is implemented). There is also the need for a study to investigate the Poisson based estimations of different sized MMPP aggregates as [11] have found a close match between Poisson performance estimations and simulation results for large aggregates of On-Off sources. These papers consider traffic with exponential packet size distributions. However, our studies, which consider the packet size distribution of Telstra traces, show a 4-modal packet size distribution, rather than an exponential one. This packet size distribution could significantly affect the performance of the DiffServ node.

Another issue which has not been raised in the literature is the performance comparison of the TD and RIO queueing mechanisms. The TD queue, a special case of RIO, is more biased against bursty traffic. However, it is worthwhile to compare the steady-state performance of these two algorithms, since it is easier to implement and dimension a TD queue.

These open areas provide the motivations for the following work:

1. A performance analysis of AF node with Poisson traffic with multiple drop-precedences (see Chapter 4).

2. A performance analysis of AF node with MMPP traffic with multiple drop-precedences (see Chapter 5).
3. A comparison of TD and RIO performances (see Chapter 6).

4. A study on the effect of packet size distribution on real-life traffic traces (see Chapter 4).

5. A study on the appropriateness of the Poisson analysis with different sized MMPP aggregates (see Chapter 4).
Chapter 4

Performance Analysis of DiffServ nodes (TD and RIO) with Poisson Arrivals

4.1 Introduction

There is a lack of DiffServ performance analyses on a per-node basis, which can be used to determine the end-to-end performance across DiffServ domains. The aim of this thesis is therefore to provide the per-node performance analysis of AF nodes to aid the dimensioning of DiffServ networks. The chapter is organised as follows:

- Section 4.2 shows the shortcoming of the delay calculations presented in [11]. It also provides an adjusted approach for these calculations, which is verified by simulations.
- Section 4.3 analyses the performances of a RIO queue and a TD queue which is a special case of RIO.
- Section 4.4 studies the applicability of the Poisson based analysis on aggregations of On-Off traffic flows.
- Section 4.5 investigates the performance of a TD node with traffic traces.
4.2 Existing Work and Adjustment

As mentioned in Chapter 3, in [11], the authors calculate the loss and delay performances of a DiffServ node with Poisson arrivals with an exponential service time distribution. The queueing mechanism used at the node is either a TD queue or a RIO queue. To obtain a mathematical tractable solution, this RIO queue drops packets with a probability determined by the instantaneous queue length. These calculations only consider two traffic flows and based on the queue length distribution. The calculations for packet expected delays can be expanded to include the load of each flow \( \rho_h \) and \( \rho_l \) and the packet mean service time \( \frac{1}{\mu} \). For the TD queue, these delay calculations are:

\[
\text{Delay}_{\text{high}} = \frac{\Pi(0)}{\mu} \left( 1 + \sum_{n=1}^{L} (n+1)(\rho_h + \rho_l)^n + (\rho_h + \rho_l)^L \sum_{n=1}^{K-L} (n+1+L)\rho_h^n \right) \quad (4.1)
\]

and

\[
\text{Delay}_{\text{low}} = \frac{\Pi(0)}{\mu} \left( 1 + \sum_{n=1}^{L} (n+1)(\rho_h + \rho_l)^n \right) \sum_{n=1}^{L} (\rho_h + \rho_l)^n \quad (4.2)
\]

The probability that there is no packet in the system \( \Pi(0) \) can be expressed as:

\[
\Pi(0) = \left[ \sum_{n=0}^{L} (\rho_h + \rho_l)^n + (\rho_h + \rho_l)^L \sum_{n=1}^{K-L} \rho_h^n \right]^{-1} \quad (4.3)
\]

where K and L are the buffer size and threshold respectively. These calculations imply that the packet expected delay is proportional to \( \Pi(0) \). Hence, as the offered load increases, \( \Pi(0) \) approaches zero and so do the expected delays. This contradicts the simulation observation that the expected delay approaches \( \frac{K}{\mu} \) for high priority packets and \( \frac{L}{\mu} \) for low priority packets as expected due to Little’s theory.

Figure 4.1 shows the difference between delay simulation results and those estimated by [11] for a simulation with two traffic classes. The high priority traffic flow (with a lower drop-precedence) contributes 95% of the combined flow while the remaining 5% comes from the low priority flow (with a higher drop-precedence). The total offered load of the combined flows varies within the range \([0.5,2.0]\) (from lightly loaded to heavily loaded) and the buffer size and threshold are 16 and 6 packets.
Figure 4.1 Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - [11] approach and Simulation results. The high priority flow contributes 95% of the total load ($K=16; L=6$).

respectively.

It can be seen from simulation results that the delay calculations only give close estimations for cases where the offered load is low (i.e. the number of lost packets is small compared to the total number of packets). This is due to the fact that these calculations take into account lost packets.

However, as the queue becomes heavily congested, the delay calculations of [11] approach zero as fewer packets are accepted. Meanwhile, the normalised expected delays (excluding discarded packets) obtained from simulations for high and low priority flows approach the threshold and buffer size values respectively.

As a result, it is necessary to adjust the packet expected delay calculations to only include accepted packets. The revised formulae to estimate packet expected delays for both flows of a TD queue are:

$$\text{Delay}_{\text{high}} = \frac{1}{\mu} \left[ L \sum_{n=0}^{L} (n+1)(\rho_h + \rho_l)^n + (\rho_h + \rho_l)^L \sum_{n=1}^{K-1-L} (n+1+L)\rho_h^n \right]$$

$$\text{Delay}_{\text{low}} = \frac{1}{\mu} \left[ L \sum_{n=0}^{L} (\rho_h + \rho_l)^n + (\rho_h + \rho_l)^L \sum_{n=1}^{K-1-L} \rho_h^n \right]$$

(4.4)
and

\[
\text{Delay}_{\text{low}} = \frac{1}{\mu} \sum_{n=0}^{L-1} (n + 1)(\rho_h + \rho_l)^n
\]

(4.5)

In these calculations, the term \(\Pi(0)\) is removed since it appears in both the numerator and denominator.

The formulae to calculate packet loss probability for each flow can be similarly derived as:

\[
\text{Loss}_{\text{high}} = \frac{\rho_h L \rho_h^{K-L}}{1 - \rho_h}
\]

and

\[
\text{Loss}_{\text{low}} = \frac{\rho_h L \rho_h^{K+L-1}}{1 - \rho_h}
\]

(4.6)

(4.7)

Results obtained from the previous simulations were plotted against the revised calculations [49] in Figure 4.2. The two traffic flows comprised 10,000 seconds, with simulation results obtained from 1000 runs. Figure 4.2 shows a close match between simulation and analytical results. The expected delay presented in this and other figures is normalised with respect to the mean \(\frac{1}{\mu}\) of the exponential service time distribution. A complete analysis of both RIO and TD queue will be presented in the following section.

### 4.3 Analysis and Simulation Results

The calculations presented in [11] are based on a traffic arrival model consisting of two Poisson flows with an exponential packet size distribution. Observations from the previous section show that there is a need to revise the calculation of packet expected delay to include accepted packets only. In this section, this analysis is extended for a RIO and a TD queue with Poisson arrivals associated with \(N\) drop-precedences. Also, some adjustments for the expected delay calculation are added where it is calculated for transmitted packets only.
Performance Analysis of DiffServ nodes (TD and RIO) with Poisson Arrivals

Figure 4.2 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - Analytical and Simulation results. The high priority flow contributes 95% of the total load ($K=16; L=6$).
4.3.1 Analysis

An $N$ drop-precedences RIO and TD queue are described in Figures 2.3 and 2.4 where there are $N$ flows (each flow corresponds to a level of drop-precedence) arriving at the queue. Note that the lower the drop-precedence of a flow, the higher the priority of the flow. Upon arrival at the queue, a packet will be either accepted or discarded based on the queue length. For a TD queue, a packet will be dropped if the queue length exceeds its corresponding threshold upon arrival, while in the RIO queue, the packet will be dropped with a certain probability. In our analysis, for mathematical tractability, the RIO queue drops packets based on the instantaneous queue length. This dropping probability of a packet can be determined from the queue’s corresponding RED parameters (i.e. minimum and maximum thresholds and the maximum drop probability) as follows:

- If the queue length is less than the corresponding minimal threshold, the packet will be accepted to the queue.
- If the queue length is equal to or greater than the corresponding maximal threshold, the packet will be dropped.
- If the queue length is less than the maximum threshold but not less than the minimum threshold, the packet will be dropped with a probability determined from the corresponding linear section of the graph presented in Figure 2.3.

4.3.1.1 Loss and Delay Calculations for a RIO Queue

The terms used in this analysis are defined as follows:

- $\lambda_i$ is the arrival rate of the $i^{th}$ drop-precedence flow.
- $\frac{1}{\mu}$ is the mean of the exponentially distributed service time (for all flows).
- $\rho_i$ is the load of the $i^{th}$ drop-precedence flow (determined as the ratio $\frac{\lambda_i}{\mu}$) while $\rho$ is the total offered load.
- $\Pi(n)$ is the probability that there are $n$ packets (includes the one being served) in the system at steady-state.
Performance Analysis of DiffServ nodes (TD and RIO) with Poisson Arrivals

- \( \alpha(n) \) is the acceptance probability of a packet which arrives at the queue and joins \( n \) other packets already in the system.

- \( \alpha_i(n) \) is the acceptance probability of an \( i^{th} \) drop-precedence packet which arrives to the queue seeing \( n \) other packets already in the system. This acceptance probability can be determined as 1 minus the dropping probability.

- \( p_i \) is the ratio of the \( i^{th} \) drop-precedence flow’s load to the overall load \( (p_i = \frac{\lambda_i}{\sum \lambda_j}) \). Hence, \( p_i \) is the ratio of \( \lambda_i \) over the sum of all arrival rates \( (p_i = \frac{\lambda_i}{\sum \lambda_j}) \).

- \( K \) is the buffer size of the queue. This buffer size is equal to the maximum threshold of the lowest drop-precedence (i.e. the highest priority flow).

From the above definitions the packet acceptance probability \( \alpha(n) \) can be determined as:

\[
\alpha(n) = \sum_{i=1}^{N} p_i \alpha_i(n)
\]  
(4.8)

This queue can be modelled as a birth-death process. For a state \( n \), the birth rate is \( \rho * \mu * \alpha(n) \) while the death rate is \( \mu \). The steady-state distribution of buffer content is:

\[
\Pi(n) = \Pi(0) \rho^n \prod_{i=0}^{n-1} \alpha(i)
\]  
(4.9)

with the probability of an empty buffer being \( \Pi(0) \) where

\[
\Pi(0) = \left[ \sum_{n=0}^{K} \rho^n \prod_{i=0}^{n-1} \alpha(i) \right]^{-1}
\]  
(4.10)

The steady-state queue length distribution enables the calculations of packet loss probability and expected delay for each flow. The loss probability of the \( i^{th} \) drop-precedence flow can be determined as:

\[
\text{Loss}_i = 1 - \sum_{n=0}^{K} \alpha_i(n) \Pi(n)
\]  
(4.11)

When a packet arrives at the queue which already contains \( n \) packets, it has a delay of \( n \) packets service times plus its own service time. Therefore, the expected delay
of the $i^{th}$ drop-precedence flow (excluding dropped packets) is:

$$\text{Delay}_i = \frac{1}{\mu} \sum_{n=0}^{K-1} (n+1) \Pi(n) \alpha_i(n)$$  \hspace{1cm} (4.12)

### 4.3.1.2 Loss and Delay Calculations for a TD Queue

Suppose we define the buffer threshold of the $i^{th}$ drop-precedence flow as $L_i$ packets ($L_0$ is 0). The $1^{st}$ priority flow has the lowest ($N^{th}$) drop-precedence and a buffer threshold of $L_N$, which is the buffer size of the queue.

The acceptance probability of $i^{th}$ drop-precedence packets can be determined as:

$$\alpha_i(n) = \begin{cases} 
1 & \text{if } n < L_i \\
0 & \text{if } L_i \leq n
\end{cases} \hspace{1cm} (4.13)$$

or

$$\alpha(n) = \begin{cases} 
p_1 + \ldots + p_N & \text{if } n < L_1 \\
p_2 + \ldots + p_N & \text{if } L_1 \leq n < L_2 \\
\ldots & \ldots & \ldots \\
p_k + \ldots + p_N & \text{if } L_{k-1} \leq n < L_k \\
\ldots & \ldots & \ldots \\
p_N & \text{if } L_{N-1} \leq n < L_N \\
0 & \text{if } n = L_N
\end{cases} \hspace{1cm} (4.14)$$

From (4.9) and (4.14), the queue length steady-state distribution can be determined as:

$$\Pi(n) = \Pi(0) \prod_{j=1}^{k-1} \left( \sum_{i=j}^{N} \rho_i \right)^{(L_j-L_{j-1})} \left( \sum_{i=k}^{N} \rho_i \right)^{n-L_{k-1}}$$ \hspace{1cm} if \hspace{0.5cm} $L_{k-1} < n \leq L_k$ \hspace{1cm} (4.15)

with

$$\Pi(0) = \left[ 1 + \sum_{i=1}^{N} \left( \prod_{j=1}^{i-1} (\rho_j + \ldots + \rho_N)^{(L_j-L_{j-1})} \sum_{k=1}^{L_i-L_{i-1}} (\rho_j + \ldots + \rho_N)^k \right) \right]^{-1} \hspace{1cm} (4.16)$$

From (4.13) and (4.11), the loss probability of the $i^{th}$ drop-precedence flow

$$\text{Loss}_i = 1 - \sum_{n=0}^{L_i-1} \Pi(n)$$ \hspace{1cm} (4.17)
Using (4.15), (4.16) and (4.17),

\[
\text{Loss}_i = \Pi(0) \left[ \prod_{j=1}^{i} (\rho_j + \ldots + \rho_N)^{(L_j - L_{j-1})} 
+ \sum_{j=i+1}^{N} \left( \prod_{k=1}^{j-1} (\rho_k + \ldots + \rho_N)^{(L_k - L_{k-1})} \sum_{m=1}^{L_j - L_{j-1}} (\rho_j + \ldots + \rho_N)^m \right) \right]
\]

(4.18)

Substitute (4.13) into (4.12), the expected delay of the \( i \)th drop-precedence packets can be determined as:

\[
\text{Delay}_i = \frac{1}{\mu} \frac{\sum_{n=0}^{L_i - 1} (n + 1) \Pi(n)}{\sum_{n=0}^{L_i - 1} \Pi(n)}
\]

(4.19)

Using (4.15), (4.16) and (4.19), the expected delay of the \( i \)th drop-precedence flow is:

\[
\text{Delay}_i = \frac{1}{\mu} \frac{A_i}{B_i}
\]

(4.20)

with

\[
A_i = 1 + \sum_{k=1}^{i} \left[ \prod_{j=1}^{k-1} (\rho_j + \ldots + \rho_N)^{(L_j - L_{j-1})} \sum_{n=1}^{L_k - L_{k-1}} (1 + n + L_{k-1})(\rho_k + \ldots + \rho_N)^n \right]
\]

\[
- (1 + L_i) \prod_{j=1}^{i} (\rho_j + \ldots + \rho_N)^{(L_j - L_{j-1})}
\]

(4.21)

and

\[
B_i = 1 + \sum_{k=1}^{i} \left[ \prod_{j=1}^{k-1} (\rho_j + \ldots + \rho_N)^{(L_j - L_{j-1})} \sum_{n=1}^{L_k - L_{k-1}} (\rho_k + \ldots + \rho_N)^n \right]
\]

\[- \prod_{j=1}^{i} (\rho_j + \ldots + \rho_N)^{(L_j - L_{j-1})}
\]

(4.22)

The next section will present simulation results to verify this analysis.

### 4.3.2 Simulation Results

A simulation is developed to verify the analysis described in the previous section. We simulate the performances of a RIO queue as well as a TD one with Poisson traffic with an exponential packet size distribution. For each queue, the incoming traffic
consists of three flows with different drop-precedences (i.e. priorities). These three flows correspond to the three drop-precedences of an AF class.

4.3.2.1 A RIO Queue with Multiple Priority Flows

As shown in Section 4.3.1, the loss and delay performances of a RIO queue can be determined analytically. The verifying simulation uses three RED parameter sets (minimum threshold, maximum threshold, maximum dropping probability) (13,18,1), (8,13,1) and (3,8,1) for high, medium and low priority flows respectively. The loads of the high and medium priority flows are fixed at 0.6 and 0.2 while the low priority one has its load varying within the range [0.1,0.9]. This variation of the load of the low priority flow makes the queue condition change from almost fully loaded to overloaded. From Figure 4.3, it can be seen that the Poisson based analytical calculations for a RIO queue closely match with simulation results. The RIO queue considered in both analysis and simulation is assumed to drop packets based on the instantaneous queue length (rather than the average queue length).

4.3.2.2 A TD Queue with Multiple Priority Flow

The simulation presented in this section models a TD queue with two thresholds. These thresholds, along with the queue buffer size, can be used to provide AF treatments to three flows with different drop-precedences within an AF class. The flows have Poisson arrivals and an exponential service time distribution. The medium and low priority flows (with medium and high drop-precedences) are generated with loads of 0.7 and 0.4 accordingly while the load of the high priority flow varies from 0.1 to 0.9. The buffer thresholds were set at 16, 12 and 8 packets for high, medium and low priority flows respectively. The packet loss probability and expected delay (normalised with respect to the packet mean service time) for each priority flow were measured and plotted as a function of the high priority load in Figure 4.4. It can be seen that the analytical results closely match those obtained from the simulation.

The simulation has verified our Poisson based analysis of RIO and TD queue. This analysis is an extension of [11] with some adjustments and it considers multiple traffic flows. It can be used to determine the packet loss probability and expected delay of an AF DiffServ node with drop-precedences. However, real-life traffic such as
Figure 4.3 RIO Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the load of the low priority flow - Analytical and Simulation results. The loads of the high and middle priority flows are fixed at 0.6 and 0.2.
Figure 4.4 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the offered load of the high priority flow - Analytical and Simulation results. The loads of the medium and low priority flows are fixed at 0.7 and 0.4 ($L_3=16; L_2=12; L_1=8$).
voice is not Poisson. It rather follows an On-Off model with the Off states representing silent durations. The following section will test the validity of this analysis with different sized On-Off aggregates (a special case of MMPP), which leads to the need of an MMPP based analysis (Chapter 5).

4.4 The Poisson Analysis and DiffServ Performances on MMPP Traffic

The MMPP process is a widely used traffic model. In particular, voice and data traffic can be captured by the MMPP model [8]. Several different methods exist for mapping a traffic flow into an MMPP model [52], [7], [53], [54]. This section will investigate the performance of a TD queue based on aggregates of On-Off traffic, which is a special case of MMPP. Simulation results will be compared with analytical results obtained from the Poisson analysis. In Figure 4.5, each priority flow is a single On-Off source with the high priority flow contributing 95% of the total load, which varies within the range [0.1,2.0]. The buffer size and threshold are fixed at 16 and 6 packets respectively while the duty cycle, \( \frac{\text{Time}_{\text{on}}}{\text{Time}_{\text{off}}} \), of the On-Off models is 50%.

Clearly, there is a significant difference between analytical and simulation results. The analysis underestimates the packet loss probability of both flows since the On-Off model is more bursty than Poisson (as presumed in the analysis). For the packet expected delay, it can be observed that as the total offered load increases, the analysis changes from underestimating to overestimating the simulation results. There are several explanations for this phenomenon:

- As the load of the connection is small, the packet loss probability is small. However, since the On-Off traffic is more bursty than the Poisson model, the queue is more likely to be full for On-Off traffic than Poisson traffic. Hence, the Poisson based delay calculations underestimates simulation results.

- If the offered load increases, there are more packets being dropped and the contribution of the Off periods become more significant. For the smoother Poisson model, the queue is more likely to be full than in the On-Off case where there
Figure 4.5 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total load - Analytical and Simulation results. Both flows are On-Off with the high priority flow contributes 95% of the total load ($K=16; L=6$).
are periods with no packets being transmitted. Hence, the Poisson based delay calculation over-estimates the actual expected delay of the simulations.

- If the duty cycle approaches 100% (On-Off becomes Poisson), the Poisson based estimations would give a close match to simulation results.

The experiment was extended to investigate the performance of a TD node on aggregates of On-Off sources. In this experiment, each of the priority flows consists of 20 identical On-Off sources with a duty cycle of 50%. Again, the high priority flow contributes 95% of the total offered load while the load varies within the range $[0.2,2.0]$. The buffer size and threshold are again 16 and 6 packets respectively. It can be seen from Figure 4.6 that the Poisson based estimation for packet loss probability and expected delay closely match simulation results. Similar simulations were performed where each priority flow is an aggregate of a smaller number of On-Off sources. Figure 4.7 shows the differences between simulation results and Poisson based analytical results for a TD queue with two flows, each of which is an aggregate of 5 identical On-Off sources. Here we see there are significant differences between analytical and simulation results in terms of loss probability. This demonstrates that for these On-Off aggregates, the Poisson analysis can be used for approximate calculations of DiffServ performance. However, to achieve more accurate calculations for such aggregates, an MMPP based analysis needs to be derived. Such analysis for multiple state MMPPs can also be used for aggregates of MMPPs since each aggregate is also an MMPP (albeit with more states). The infinitesimal generator and arrival rate matrices of a combined MMPP can be calculated from those of individual MMPPs [55]. Furthermore, research in [5] suggests Markovian models such as MMPP can be used to model self-similarity and long-range dependence traffic in a finite buffer queue, as long as the chosen model captures the correlation structure up to a “correlation horizon”. Hence, it is important to provide an analysis of the performance of a DiffServ node with MMPP traffic. This analysis should cover multiple drop-precedences, as well as the complexity of the MMPP model (i.e. $n$-state MMPP). The MMPP based performance analysis will be presented in Chapter 5.

The following section continues to investigate the performance of a TD DiffServ
Figure 4.6 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total load - Analytical and Simulation results. Both flows are aggregates of 20 identical On-Off sources with the high priority flow contributes 95% of the total load ($K=16; L=6$).
Figure 4.7 (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total load - Analytical and Simulation results. Both flows are aggregates of 5 identical On-Off sources with the high priority flow contributes 95% of the total load ($K=16; L=6$).
node with a traffic trace and in comparison with analytical results to show the significant of the packet size distribution.

4.5 Traffic Trace Investigation

Previous sections have investigated the performance of a DiffServ AF node (TD and RIO) with Poisson traffic. Both the analysis and simulation assume an exponential service time distribution (i.e. exponential packet size distribution). This assumption however may not hold for actual traffic at a core network.

A study on Telstra traffic traces (captured on 19/9/1999) shows that the packet size distribution is not exponential. Rather it is a 4-modal distribution with distinct regions at packet sizes of 50, 100, 550 and 1500 bytes with a mean packet size of approximately 341 bytes. Figure 4.8 shows the packet size distribution obtained from a collections of Telstra traces. A simulation has been performed with input packet sequence obtained from the trace (both time-stamps and packet sizes). Both high and low priority flows are approximately 3 minutes long. The simulation was run for different service rates of the DiffServ (TD) node. Service rates range from 1MBps to 5MBps while the combined arrival rate is approximately 3.9MBps. The buffer size and threshold are fixed at 16 and 6 packets respectively. Analytical results are
obtained from the Poisson based analysis with the mean Poisson arrival rate being the trace arrival rate. The mean packet size of the exponential distribution is set at 341 bytes. Within the two priority flows, the higher priority stream contributes 90% of the combined flow.

The trace based simulation and equivalent analytical results are shown in Figure 4.9. This figure shows that the analysis closely matches the delay performance of the node under lighter load. However, there are significant differences between analytical and simulation results for the packet loss probability, especially when the service rate increases.

The same simulation is repeated where the previously used traffic traces are replaced by modified ones. These modified traffic traces have the same time-stamp as the original traces and an exponential packet size distribution rather than a 4-modal one. The exponential distribution has the same mean as the 4-modal one, i.e. 341 bytes. Figure 4.10 shows smaller discrepancies between simulation and analytical results for the modified traces. This is expected since the packet size distribution is made to be exponential as assumed in the Poisson based analysis. It can be seen from Figure 4.10 that the packet expected delay calculation closely matches simulation results. In terms of packet loss probability, the analysis gives a closer estimation for these modified traces than for the real traces. However, the discrepancies are still significant as the node's service rate increases since the arrival process of the traces can not be accurately modelled by a Poisson process.

These simulations have shown that packet size or service time distributions of the traffic traces make a significant impact on the performance of the node. Hence, it is important to examine the impacts of packet size distribution by further analysis as well as experimental work.
Figure 4.9 (a) Loss Probability and (b) Expected Delay of High and Low Priority Packets as a Function of the Node’s Service Rate. Packet Sizes are obtained from the trace.
Figure 4.10 (a) Loss Probability and (b) Expected Delay of High and Low Priority Packets as a Function of the Node's Service Rate. Packet Sizes are obtained from the trace.
4.6 Conclusions

This chapter has presented an analysis to determine the QoS performance (loss and delay) of an AF DiffServ node on Poisson traffic with an exponential packet size distribution. This analysis considers both RIO and TD queueing mechanisms. In particular, the analysis can be applied to an AF DiffServ node with multiple priority flows. These situations correspond to an AF class with three drop-precedences.

The analysis presented in this chapter is extended from the work in [11] where the number of priority flows (i.e. drop-precedences) is generalised and with adjustments in the delay calculation to consider accepted packets only. Simulations for both RIO and TD queues were also performed to verify the analysis.

The simulation study also considers the appropriateness of the Poisson based analysis with more bursty traffic. Simulation results in this chapter illustrate the need for an MMPP based analysis, as the Poisson based analysis does not always provide accurate performance estimations for aggregates of MMPP traffic (e.g. On-Off traffic).

A study of Telstra real life traffic traces shows that the packet size distribution is not exponential. Instead, it follows a 4-modal distribution, which significantly affects the performance of the AF DiffServ node. Hence, the packet size distribution (i.e. service time distribution) needs to be considered in any performance study of real traffic traces.

Our research extends this Poisson based analysis to consider MMPP arrivals (with exponential packet size distribution) as presented in Chapter 5. Alternatively, this analysis can be extended to consider Poisson arrivals with general packet size distribution or MMPP arrivals with general packet size distributions. These issues remain open for future work.

The following chapter will provide an analysis for an AF DiffServ node (TD and RIO) with MMPP arrivals with an exponential packet size distribution.
Chapter 5

Performance Analysis of DiffServ nodes (TD and RIO) with MMPP Arrivals

As shown in Chapter 4, the Poisson based analysis can be used to approximate the performance of a TD node for large aggregates of On-Off sources. However, an MMPP analysis is needed to provide more accurate results. MMPP traffic models are often used to model aggregations of flows, such as the superposition of voice and data sources [7], [8]. Furthermore the MMPP, as a Markovian model, can be adequately used to model self-similarity and long-range dependence traffic in finite buffer queues, as long as it captures the correlation structure up to the correlation horizon [5]. This chapter presents a performance analysis of an AF DiffServ node on MMPP traffic with an exponential packet size distribution. It extends the work in [13] but considers multiple MMPP flows with different number of MMPP states. Simulation results are also obtained to verify this analysis.

5.1 Existing Work

As mentioned in Chapter 3, in [13], the authors presented a method to calculate the packet loss probability of a TD node for MMPP arrivals with exponential service time distribution. The node considered in this analysis has a threshold for the lower priority flow. The buffer size and threshold include the packet being served. How-
ever, there is no delay calculations. These calculations are provided here, based on
the queue length steady-state distribution vector $\pi$. From the definition

$$\text{ExpectedDelay} = \frac{\text{Sum of all packet delays}}{\text{Total number of packets accepted}}$$

the expected delay of high and low priority flow can be determined as:

$$\text{ExpectedDelay}_{\text{high}} = \frac{1}{\mu} \sum_{q=0}^{K-1} (q + 1) \left( \lambda_1 \pi(1, q) + \lambda_2 \pi(2, q) \right)$$  \hspace{1cm} (5.1)

and

$$\text{ExpectedDelay}_{\text{low}} = \frac{1}{\mu} \sum_{q=0}^{\theta-1} (q + 1) \left( \lambda_3 \pi(3, q) + \lambda_4 \pi(4, q) \right)$$  \hspace{1cm} (5.2)

with

- $1$ and $2$ are states of the first MMPP flow while $3$ and $4$ are states of the second
  MMPP flow.
- $\lambda_1$ and $\lambda_2$ are arrival rates which correspond to the states of the first MMPP
  while $\lambda_3$ and $\lambda_4$ are arrival rates which correspond to the states of the second
  MMPP.
- $\pi(n, q)$ is the probability that one of the MMPP flows is in state $n$ and there
  are $q$ packets in the system, including the one being served.
- $K$ and $\theta$ are the node’s buffer size and threshold respectively.
- $\frac{1}{\mu}$ is the packet mean service time of the MMPP flows. The packet service time
  distribution is exponential.

Figure 5.1 shows the simulated loss and delay performance of a TD node with two
2-state MMPP traffic flows. These results closely match the analytical ones (the formu­
lae to estimate losses are obtained from [13]). The buffer size and threshold are
Figure 5.1 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the load of the low priority flow - Analytical and Simulations results. The offered load of the high priority flow is fixed at 0.8 ($K = 16; \theta = 6$).
16 and 6 packets respectively. The buffer size values are chosen to simplify the simulation process while still show the close match between analytical and simulation results. The high priority flow’s load is fixed at 0.8 while the load of the low priority one is varied from 0.1 to 0.9.

However, the performance calculations presented in [13] only consider two 2-state MMPP flows. In DiffServ environment, it is necessary to consider more than two priority flows, e.g an AF class with three drop-precedences. It is known [55] that an aggregation of MMPP flows can also be modelled by an MMPP (with more MMPP states). Hence, our MMPP based analysis considers an AF node with multiple MMPP flows with different numbers of MMPP states.

The analysis presented in the following section was derived with these considerations. It is also verified with simulation results for two cases: three 2-state MMPP flows and two 4-state MMPP flows.

5.2 Analysis and Simulation Results

As mentioned in the previous chapter, the Poisson based analysis is inaccurate for small aggregates of On-Off sources. Instead, an aggregation of MMPP sources can be represented by a more complicated MMPP model [55]. Hence, it is important to derive a performance analysis of an AF DiffServ node, which employs TD or RIO, with MMPP traffic with exponential packet size distributions. The method presented in this section is based on [13], extended to consider \( n \) multi-state MMPP flows. Each flow is an aggregate of many microflows subscribed to the same DSCP. The queueing mechanism at the node can be either RIO or TD (a special case of RIO). Also, the expected delay calculations for each priority flow are provided in this analysis.

5.2.1 Analysis

Figures 2.3 and 2.4 describe a TD and a RIO queue with \( n \) drop-precedences. Each of these \( n \) flows has a particular drop-precedence, which defines its priority. As mentioned, a flow with lower drop-precedence has higher priority than the one with higher drop-precedence. Upon arrival at the queue, the decision to accept or discard
a packet is based on the state of the queue’s buffer (i.e. the queue length). For the TD queue, a packet will be dropped if the queue length exceeds its corresponding threshold upon arrival, while in the RIO queue, the packet will be dropped with a probability based on the average queue length. In this analysis the instantaneous queue length is used to obtain mathematical tractability.

The performance of a DiffServ node (TD or RIO) is based on the steady-state probability distribution vector \( \pi \) of the queue. In [13], the authors present a method to calculate the vector \( \pi \) by constructing an infinitesimal generator matrix of a TD queue and calculating the packet loss probabilities accordingly. This section presents the extended analysis to consider a RIO queue, which drops packets based on the instantaneous queue length, with multiple traffic flows with different drop-precedences. Similar analysis is also derived for a TD queue. The \( k^{th} \) priority flow can be modelled by an \( m_k \)-state MMPP. These analyses require the construction of the queue’s infinitesimal generator matrix and the queue length probability vector.

5.2.1.1 Loss and Delay Calculations for a RIO Queue

The following notations are defined in the solution for a RIO queue with \( n \) multi-state MMPP traffic flows:

- \( \theta_{\text{min}}^k \) and \( \theta_{\text{max}}^k \) are the threshold values of the \( k^{th} \) priority MMPP flow and \( \theta_{\text{min}}^k \leq \theta_{\text{max}}^k \). The maximum threshold of the 1\( st \) priority flow \( \theta_{\text{max}}^1 \), equals the buffer size.

- \( Y_k(t) \) and \( Z(t) \) are the state (i.e. phase) of the \( k^{th} \) flow and the number of packets in the queue (include the one being served) at time \( t \) respectively. Hence, \( Y_k(t) \) can take a value out of \( m_k \) values, which represent the states of the \( k^{th} \) priority MMPP. \( Z(t) \) can take on values ranging from 0 to \( \theta_{\text{max}}^1 \) (i.e. from empty buffer to full buffer).

- Let \( \pi \) be the steady-state probability distribution for the Markov process \( \{ Y_1(t), Y_2(t), \ldots, Y_n(t), Z(t) \} \). An element \( \pi(i_1, i_2, \ldots, i_n, q) \) of \( \pi \) is the probability that the MMPP flows are in states \( i_1, i_2, \ldots, i_n \) and the buffer occupancy is \( q \).
- \( r_{ij}^k \) is the transition rate of the \( k^{th} \) priority MMPP flow's arrival process from state \( i \) to state \( j \).

- \( \lambda_i^k \) is the Poisson arrival rate when the arrival process of the \( k^{th} \) priority MMPP is in state \( i \).

- \( \mu \) is the service rate of the queue. It is assumed that the packet service times of all flows are exponentially distributed with mean \( \frac{1}{\mu} \).

- \( \alpha_k(q) \) is the acceptance probability of an \( k^{th} \) priority packet which arrives to the queue seeing \( q \) other packets already in the system. Similar to the Poisson based analysis, this acceptance probability can be determined from the RED parameters; \( \theta_{min}^k, \theta_{max}^k \) and the maximum dropping probability of the associated flow.

This analysis employs the matrix notation of MMPP parameters to provide a compact expression for the solution. The matrices \( R_k \) and \( \Lambda_k \) are constructed based on MMPP parameters of the \( k^{th} \) priority flow. \( R_k \) is the infinitesimal generator matrix and \( \Lambda_k \) is the arrival rate matrix of the flow:

\[
R_k = \begin{pmatrix}
-r_{11}^k & r_{12}^k & \cdots & r_{1m_k}^k \\
 r_{21}^k & -r_{22}^k & \cdots & r_{2m_k}^k \\
 \vdots & \vdots & \ddots & \vdots \\
 r_{m_k1}^k & r_{m_k2}^k & \cdots & -r_{m_k m_k}^k
\end{pmatrix}
\]

\[
\Lambda_k = \begin{pmatrix}
\lambda_1^k & 0 & \cdots & 0 \\
0 & \lambda_2^k & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_{m_k}^k
\end{pmatrix}
\]

The combined MMPP of \( n \) multi-state MMPP flows has an infinitesimal generator and an arrival rate matrix as:

\[
R = R_1 \oplus R_2 \oplus \ldots \oplus R_n \quad \Lambda = \Lambda_1 \oplus \Lambda_2 \oplus \ldots \oplus \Lambda_n
\]

where \( \oplus \) represents the Kronecker-sum operation of two matrices [55], [56].

Additionally, a matrix \( M \) which represents the service rate of the queue is defined as a diagonal matrix of size \( \prod_{k=1}^{n} m_k \) where each diagonal element is \( \mu \):

\[
M = \begin{pmatrix}
\mu & 0 & \cdots & 0 \\
0 & \mu & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \mu
\end{pmatrix}
\]
The column vectors $\bar{\Lambda}_k$ is the diagonal of $\Lambda_k$.

$$\bar{\Lambda}_k = \text{diag}(\Lambda_k)$$

Since the packet acceptance probability is dependent on the instantaneous queue length for a RIO queue, the effective arrival rate matrix of the $k^{th}$ priority MMPP can be determined as the product of the MMPP's arrival rate matrix $\Lambda_k$ and the acceptance probability function $\alpha_k(q)$.

$$\Lambda_k^{\text{effective}}(q) = \alpha_k(q)\Lambda_k \quad (5.3)$$

with $0 \leq q \leq \theta_{\text{max}}^1$. Hence, the combined effective arrival rate matrix is also a function of the instantaneous queue length:

$$\Lambda^{\text{effective}}(q) = \Lambda_1^{\text{effective}}(q) \oplus \Lambda_2^{\text{effective}}(q) \oplus \cdots \oplus \Lambda_n^{\text{effective}}(q) \quad (5.4)$$

This combined effective arrival rate matrix is essential in constructing the infinitesimal generator matrix of the queue.

The infinitesimal generator matrix $Q$ is a square matrix of size $(\theta_{\text{max}}^1 + 1) \prod_{k=1}^{n} m_k$, which can be re-written in a block matrix form where $Q_{ij}$ is a matrix of size $\prod_{k=1}^{n} m_k \times \prod_{k=1}^{n} m_k$ with the indices $i$ and $j$ range within $[1 \ldots (\theta_{\text{max}}^1 + 1)]$. Each element $Q_{ij}$ can be determined as follows

$$Q_{ij} = \begin{cases} (R - \Lambda) & \text{if } i = j = 1 \\ (-M + R) & \text{if } i = j = \theta_{\text{max}}^1 + 1 \\ M & \text{if } 2 \leq i = j + 1 \leq \theta_{\text{max}}^1 + 1 \\ \Lambda^{\text{effective}}(i) & \text{if } 1 \leq i = j - 1 \leq \theta_{\text{max}}^1 \\ (-M + R - \Lambda^{\text{effective}}(i)) & \text{if } 2 \leq i = j \leq \theta_{\text{max}}^1 \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

The steady-state probability distribution vector $\pi$ can be re-written as

$$\pi = [\pi_0, \pi_1, \ldots, \pi_{\theta_{\text{max}}^1}]$$

Each element $\pi_q$ of the vector $\pi$ corresponds to the limiting probability vector when there are $q$ packets in the queue and formed by those $\pi(i_1, i_2, \ldots, i_n, q)$'s. This vector
\( \pi \) satisfies the following conditions

\[
\pi \ast Q = 0 \quad \text{and} \quad \pi \ast e = 1
\]

where \( e = (1, \ldots, 1)' \) is a column vector of \( (\theta_{\text{max}}^1 + 1) \prod_{k=1}^{n} m_k \) elements.

Let \( \pi_k \) be the probability distribution of the \( k^{th} \) MMPP phase process, we have

\[
\pi_k \ast R_k = 0 \quad \text{and} \quad \pi_k \ast e_k = 1
\]

where \( e_k = (1, \ldots, 1)' \) is a column vector of \( m_k \) elements.

The packet loss probability and expected delay of the \( k^{th} \) priority flow can be calculated as

\[
P_{\text{loss}}(k) = \frac{(\pi_k \ast \bar{A}_k)}{(\pi_k \ast \bar{A}_k)}
\]

\[
\text{ExpectedDelay}(k) = \frac{1}{\mu} \frac{\sum_{q=0}^{\theta_{\text{max}}^k - 1} (q + 1)}{\sum_{q=0}^{\theta_{\text{max}}^k - 1}} \left( \pi_q \ast \left( e_1 \otimes \cdots \otimes e_{k-1} \otimes \left( \alpha_k(q) \bar{A}_k \right) \otimes e_{k+1} \otimes \cdots \otimes e_n \right) \right)
\]

where \( \otimes \) represents the Kronecker-product operation of two matrices.

The next section shows the loss and delay calculations when the \( n \) level RIO queue becomes an \( n \) level TD queue, that is when \( \theta_{\text{max}}^k = \theta_{\text{min}}^k + 1 \) for all \( k \).

### 5.2.1.2 Loss and Delay Calculations for a TD queue

In an \( n \) level TD queue, the buffer size \( K \) can be seen as \( \theta_{\text{max}}^1 \) and \( \theta_k \) is the threshold of the \( k^{th} \) priority flow. As in the case of a RIO queue, this TD queue has an
infinitesimal generator matrix \( Q \) as \([Q_{ij}]\) with

\[
Q_{ij} = \begin{cases} 
M & \text{if } i = j + 1 \\
(R - \Lambda) & \text{if } i = j = 1 \\
(-M + R) & \text{if } i = j = K + 1 \\
\Lambda & \text{if } 1 \leq i = j - 1 \leq \theta_n \\
(-M + R - \Lambda) & \text{if } 2 \leq i = j \leq \theta_n \\
\Lambda_1 \oplus \Lambda_2 \oplus \cdots \oplus \Lambda_{n-1} \oplus 0 & \text{if } \theta_n < i = j - 1 \leq \theta_{n-1} \\
(-M + R - \Lambda_1 \oplus \Lambda_2 \oplus \cdots \oplus \Lambda_{n-1} \oplus 0) & \text{if } \theta_n < i = j \leq \theta_{n-1} \\
\Lambda_1 \oplus \Lambda_2 \oplus \cdots \oplus \Lambda_k \oplus 0 \oplus \cdots \oplus 0 \quad n-k & \text{if } 2 \leq \theta_{k+1} < i = j - 1 \leq \theta_k \\
(-M + R - \Lambda_1 \oplus \Lambda_2 \oplus \cdots \oplus \Lambda_k \oplus 0 \oplus \cdots \oplus 0) & \text{if } \theta_{k+1} < i = j \leq \theta_k \\
\Lambda_1 \oplus 0 \cdots \oplus 0 \quad n-1 & \text{if } \theta_2 < i = j - 1 \leq K \\
(-M + R - \Lambda_1 \oplus 0 \cdots \oplus 0) & \text{if } \theta_2 < i = j \leq K \\
0 & \text{otherwise}
\end{cases}
\]

(5.8)

Equation (5.8) is the extended form of (5.5) for a TD queue where the packet acceptance probability \( \alpha_k(q) \) is either 1 or 0 depending on \( q \). The queue’s steady-state probability distribution vector \( \pi \) can also be determined from the conditions

\[
\pi \ast Q = 0 \quad \text{and} \quad \pi \ast e = 1
\]

where \( e = (1, \ldots, 1)' \) is a column vector of \((K + 1) \prod_{k=1}^{n} m_k\) elements. Hence, the packet loss probability and expected delay for the \( k^{th} \) priority flow can be calculated as

\[
P_{\text{loss}}(k) = \frac{\left( \sum_{q = \theta_k}^{K} \pi_q \ast \left( e_1 \otimes \cdots \otimes e_{k-1} \otimes \Lambda_k \otimes e_{k+1} \otimes \cdots \otimes e_n \right) \right)}{(\pi_k \ast \Lambda_k)}
\]

(5.9)
and

\[
\text{ExpectedDelay}(k) = \frac{1}{\mu} \sum_{q=0}^{\theta_k-1} (q+1) \left( \pi_q \ast \left( e_1 \otimes \cdots \otimes e_{k-1} \otimes \tilde{\Lambda}_k \otimes e_{k+1} \otimes \cdots \otimes e_n \right) \right)
\]

(5.10)

The following section presents simulation results to verify the analysis derived in this section.

5.2.2 Simulation Results

The simulation in Chapter 4 is extended to provide AF treatments to MMPP flows. The simulation was run with both RIO and TD queues to verify the analysis. This simulation considers the following cases:

- Three 2-state MMPP flows with exponentially distributed service times.

- Two 4-state MMPP flows with exponentially distributed service times.

5.2.2.1 A RIO Queue with Multiple Priority Flows

The simulation was run for a RIO queue with both cases: three 2-state MMPPs and two 4-state MMPPs. In the first case, the RED parameter sets used are (13,18,1), (8,13,1) and (3,8,1) for the high, medium and low priority flows respectively. The offered loads of the high and medium priority flows are fixed at 0.6 and 0.2 respectively while the low priority flow’s load varies within the range [0.1,0.9]. This variation of the low priority flow’s load causes the queue condition to range from almost fully loaded to over loaded.

Meanwhile, the RED parameter sets used in the second case are (13,18,1) and (3,8,1) for the high and low priority flows respectively. In this simulation, the high priority flow is set to contribute 70% of the total offered load while the low priority flow makes up the remaining 30%. The arrival rate parameters of the MMPP models are adjusted to vary the offered load from 0.1 (lightly loaded) to 2 (heavily loaded).

From Figures 5.2 and 5.3, it can be seen that the analysis presented in the previous
section gives very accurate estimations for the loss and delay performance of a RIO queue with MMPP arrivals. This analysis can be used for multiple drop-precedences and complicated MMPP models.

Figure 5.2 RIO Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the load of the low priority flow - Analytical and Simulation results. There are three 2-state MMPP flows. The loads of the high and middle priority flows are fixed at 0.6 and 0.2.
Figure 5.3 RIO Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - Analytical and Simulation results. There are two 4-state MMPP flows. The load of the high priority flow is fixed at 70% of the combined flow.

5.2.2.2 A TD Queue with Multiple Priority Flows

The same set of simulations was also performed for a TD queue. The input traffic flows used in these simulations are the same as those used for the RIO queue simulations. Again, the packet service time distribution is exponential. In the simulation with three 2-state MMPP flows, the buffer size and thresholds corresponding to the
high, medium and low priority flows are set at 16, 12 and 8 packets respectively. The simulation with two 4-state MMPP flows uses buffer size and threshold of 16 and 6 packets for high and low priority flows respectively.

Figure 5.4 compares simulations and analytical results for the TD queue with three MMPP flows while Figure 5.5 presents simulation and analytical results for the 4-state MMPP case. It can be seen from these figures that again, the analysis provides performance estimations that are close to actual simulation results. This is expected since a TD queue is a special case of RIO with $\theta_{max}^k = \theta_{min}^k + 1$ for all $k^{th}$ priority flows.

5.3 Conclusions

This chapter has presented a method to determine the QoS performances (packet loss probability and expected delay) of an AF DiffServ node with MMPP traffic and an exponential packet size distribution. In this analysis, the performance (loss and delay) of the AF node (RIO and TD) can be determined, based on the network resources as well as traffic parameters (arrival rate and transition matrices). Similar to the Poisson based analysis, this MMPP analysis can be applied to multiple flows with different drop-precedences. Hence, it is useful for provisioning the AF PHB model, where each AF class has three drop-precedences. However, the analyses presented in this chapter and in Chapter 4 apply to traffic with exponential packet size distributions. Our simulations, with more realistic packet size distributions, have shown discrepancies between analytical and simulation results. A theoretical analysis with general packet size distributions remains an open issue.

Hence, the following chapter uses a realistic packet size distribution to compare the performances of RIO and TD queues, and investigate the capacity planning of a Diff-Serv node and the effects of flow proportions.
Figure 5.4 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high, medium and low priority packets as a function of the load of the low priority flow - Analytical and Simulation results. There are three 2-state MMPP flows. The loads of the high and middle priority flows are fixed at 0.6 and 0.2. The thresholds and buffer size are 8, 12 and 16 packets respectively.
Figure 5.5 TD Queue: (a) Loss probability and (b) Expected delay (normalised with respect to the packet mean service time) of high and low priority packets as a function of the total offered load - Analytical and Simulation results. There are two 4-state MMPP flows. The load of the high priority flow is fixed at 70% of the combined flow. The threshold and buffer size are 6 and 16 packets respectively.
Chapter 6

Performance Comparison of TD and RIO queues

Chapters 4 and 5 introduced methods for determining the loss and delay performances of an AF DiffServ node. The traffic model considered in Chapter 4 is Poisson with an exponential packet size distribution while Chapter 5 considered MMPP traffic, also with an exponential packet size distribution. Both methods can be used for multiple packet priorities (i.e. drop-precedences) and thus may be applied to a single AF class with three different packet drop-precedences.

Two possible AF DiffServ queueing mechanisms, namely Threshold Dropping (TD) and RED with In/Out profile (RIO), were considered in these analyses. For the RIO queue, each drop-precedence has a set of RED parameters which determine the probability a packet is dropped upon its arrival. This packet drop probability is based on the instantaneous queue length for simplicity. On the other hand, the TD queue has a set buffer thresholds where each corresponds to a drop-precedence. The buffer size is the threshold of the lowest drop-precedence. Simulation studies presented in previous chapters verified the Poisson based and MMPP based analyses for both mechanisms.

The advantage of RIO queues over TD ones is that they avoid global synchronisation of TCP traffic [40] and provides more stable performance than TD (since TD is an on-off queueing mechanism) [57]. However, in a corporate network environment with TCP and voice traffic, the main focus is to achieve negligible or zero loss and low delay. From a network provider’s perspective, a TD queue is preferable to
a RIO queue because of its simpler dimensioning. Hence, a important question to be addressed is whether a TD queue can be used to approximate the loss and delay performance of a RIO queue (i.e. dimension a TD queue to match the performances of a RIO one). This chapter examines the performance of RIO and TD queues with MMPP traffic and 4-modal packet size distributions.

Section 6.1 shows the circumstances, under which a TD queue can approximate a RIO queue. Section 6.2 focuses on the design problem of a TD DiffServ node to determine the bandwidth required to meet a a certain level of QoS (with loss probabilities as primary goals). Section 6.3 investigates the effects of relative proportions of the aggregate on an AF node’s performances.

6.1 TD and RIO Performance Comparison

In Chapters 4 and 5, AF DiffServ node analyses (TD and RIO queueing mechanisms) were developed with simple arrival models (i.e. Poisson and MMPP arrivals) and packet size distributions (i.e. exponential distribution). This section considers a scenario where a link is shared by Web traffic (TCP) and voice (UDP) traffic. Web traffic requires low loss, but tolerates longer delays than voice. Conversely, voice traffic needs a low bounded delay but is more tolerant to loss than the Web traffic. Hence, if an AF class provides service for both Web and voice traffic, the higher drop-precedences will be used for voice traffic (lower delay and loss tolerant) while the lower drop-precedences will be used for Web traffic (lower loss and delay tolerant). A simulation is designed to investigate the performances of several RIO queues and their respective TD approximations. Figure 6.1 describes a simulated RIO queue with two flows (In and Out or high and low priority flows). The threshold values for In and Out packets are \((\text{Th}_{\text{In min}}, \text{Th}_{\text{In max}})\) and \((\text{Th}_{\text{Out min}}, \text{Th}_{\text{Out max}})\) respectively.

The TD queue which approximates the performance of this RIO queue has the parameters (threshold and buffer size) calculated as

\[
\text{Threshold} = \frac{\text{Th}_{\text{Out min}} + \text{Th}_{\text{Out max}}}{2}
\]
Figure 6.1 A RIO queue with instantaneous queue length and maximum drop probability of 1

and

$$\text{BufferSize} = \frac{\text{Th}_{\text{In}_{\text{min}}} + \text{Th}_{\text{In}_{\text{max}}}}{2}$$

The parameter set \((\text{Th}_{\text{In}_{\text{min}}}, \text{Th}_{\text{In}_{\text{max}}}, \text{Th}_{\text{Out}_{\text{min}}}, \text{Th}_{\text{Out}_{\text{max}}})\) of the RIO queues considered in this simulation study is \((1950, 2050, 200, 300)\) or \((1900, 2100, 200, 300)\) or \((1850, 2150, 200, 300)\) or \((1800, 2200, 200, 300)\) (named as RIO1, RIO2, RIO3 and RIO4 respectively). These buffer configurations will be shown to satisfy the delay requirements of the traffic flows. Using the approximation method described above, these RIO queues have the same TD approximation of

$$\text{BufferSize} = 2000 \quad \text{and} \quad \text{Threshold} = 250$$

The link capacity and the traffic parameters used in this simulation were obtained from real-life situations (a link of 155Mbps, with packets whose sizes form a discrete 4-modal distribution, similar to the previously mentioned Telstra traces). Packet sizes are 50 bytes (with probability of 0.55), 100 bytes (with probability of 0.15), 550 bytes (with probability of 0.15) or 1500 bytes (with probability of 0.15). Both flows are generated with MMPP arrivals and the same 4-modal packet size distribution. High priority packets contribute 80% of the aggregate flow. The MMPP modes \((\lambda_1, \lambda_2, t_1, t_2)\) used to generate reasonably bursty traffic by setting the arrival rate ratio...
The performances (loss probability and mean delay) of the RIO and TD queues on these MMPP flows are obtained for a range of utilisations. Simulation results presented in Figures 6.2 and 6.3 show that for this particular buffer size, the difference in the loss and delay performances (for high and low priority packets) of RIO and TD queues is insignificant. In other words, the delay and loss performances of a RIO queue are roughly equivalent to that of a TD one.

It can be seen from Figure 6.2 that the loss probability of high priority packets is less than 0.1% at an utilisation of 0.85 while the loss probability of low priority packets achieves the same level at an utilisation of 0.65. There are also sharp increases in both losses as the queue's utilisation increases (at the intervals [0.85, 1.1] and [0.65, 0.75] for high and low priority packets respectively).

In terms of delay performance (Figure 6.3), the TD queue is consistently close to the RIO one with a maximum discrepancy of approximately 2%. There is a peak of low priority packets' mean delay, due to the sharp increase in the loss probability over the same utilisation range. For a smaller utilisation, the queue length rises with the utilisation. This causes the mean delay to increase. However, after reaching the peak, more low priority packets are dropped as a result, which decreases the mean delay. If the utilisation continues to increase, the delay will rise again towards the delay bound (set by the threshold). This does not happen to high priority packets since they can occupy the whole buffer.

The performances of smaller RIO queues and their respective TD approximation were also investigated. In this study, the smaller TD queue considered has a threshold and buffer size of 250 and 500 packets respectively. The buffer threshold is the average of the Out profile thresholds while the buffer size is the average of the In profile thresholds. The parameter sets of the RIO queues \((\text{Th}_{\text{In}}_{\text{min}}, \text{Th}_{\text{In}}_{\text{max}}, \text{Th}_{\text{Out}}_{\text{min}}, \text{Th}_{\text{Out}}_{\text{max}})\) are \((450, 550, 200, 300)\) or \((400, 600, 200, 300)\) or \((350, 650, 200, 300)\) or \((300, 700, 200, 300)\) (named as new_{RIO1}, new_{RIO2}, new_{RIO3} and new_{RIO4} respectively). In this simulation, high priority packets contribute slightly over half (58%) of the aggregate. When the utilisation is within the range \([1,1.2]\), significant differences in the loss performance of high priority packets were observed (Figure 6.4). This is because the range \([\text{Th}_{\text{In}}_{\text{min}}, \text{Th}_{\text{In}}_{\text{max}}]\) of the RIO queue is close
Figure 6.2 (a) Loss probability of high priority packets and (b) Loss probability of low priority packets of a TD queue compared with RIO queues
Figure 6.3 (a) Mean Delay of High Priority Packets and (b) Mean Delay of Low Priority Packets of a TD queue compared with RIO queues
to the threshold of the TD one and the instantaneous queue length falls into this region, yet it rarely exceeds the buffer size. Hence, the RIO queues drop more packets and the bigger the region \([\text{Th}_\text{In}_\text{min}, \text{Th}_\text{In}_\text{max}]\), the more high priority packets will be dropped. As the load increases, the instantaneous queue length shifts to the right and causes similar loss performance of RIO and TD queues. Throughout this range, low priority packets experience a similar loss since the queue length almost always exceeds the low priority thresholds.

Both scenarios have shown that the link can not operate near its capacity since the loss probability of low priority packets is large (e.g. approximately 30% for the large buffer case). Therefore, the utilisation of the queue needs to be kept relatively low. For those cases TD and RIO queues give similar performance. The following section will determine the utilisation level to achieve realistic loss and delay performance.

It can be concluded that for the mix of Web and voice traffic, the TD queue provides similar and occasionally, better performance compared to the RIO queues. Zero or small loss probabilities can be achieved for both Web and voice traffic. With very small Web (TCP) traffic loss, packet re-transmissions are avoided and subsequently the global synchronisation problem that may happen with TD queues. The mean delays are also bounded due to the threshold and buffer size of the TD queue.
6.2 DiffServ Node Capacity Planning

A key issue for network providers is the bandwidth provided to an aggregate to meet its desired quality of service. Having compared the performances of RIO queues with their TD approximation, the previous simulation is extended to determine the amount of bandwidth needed to meet the desired QoS of the Web/voice traffic mix at a TD DiffServ node.

This capacity planning simulation study considers an aggregate of Web (TCP) and voice (UDP) traffic. These traffic flows (Web and voice) are generated based on MMPP arrivals and a discrete 4-modal packet size distribution. Web packets are considered as high priority packets (i.e. low drop-precedence) while voice packets as low priority (i.e. high drop-precedence). Since TCP traffic requires low loss and is delay tolerant, this simulation study aims to achieve a very low packet loss probability and 20ms mean delay bound performance for the Web traffic. The voice traffic, which requires bounded delay and is relatively tolerant to loss, aims to receive small loss and a smaller bounded mean delay of 5ms. Since voice traffic has to pass through multiple nodes and voice codecs accept losses at approximately 1% the per-hop loss performance is aimed at less than or equal to 0.03% for the voice flow.

From Figure 6.3, it can be seen that a link of 155Mbps with a buffer size and threshold of 2000 and 250 packets is capable of providing these mean delay bounds. The remaining issues is whether the loss performance can be achieved. Therefore, this simulation study considers a range of utilisation over the described link with high priority packets contribute 80% of the aggregate.

Figure 6.5 shows the loss probabilities of high and low priority packets as a function of the ratio between link capacity over the aggregate’s mean arrival rate. This ratio varies from 0.46 (i.e. over-provision) to 1.85 (i.e. under-provision). This figure shows that to achieve very low loss (<0.0001) for Web traffic and less than 0.03% loss for voice traffic, the utilisation of the link cannot exceed 0.66 (i.e. the ratio of link capacity over arrival rate is greater or equal to 1.5). Hence, for this particular scenario, a 155Mbps link with a TD queue with buffer size and threshold of 2000 and 250 packets is capable of providing the desired performance as long as the utilisation is kept under 0.66.
Figure 6.5 (a) Loss probability of high priority packets and (b) Loss probability of low priority packets as a function of the ratio link capacity / mean arrival rate of the aggregate flow (high priority packets make up 80% of the aggregate)
However, to meet the desired performance for a given traffic flow, the bandwidth, buffer size and threshold need to be determined. This requires a performance analysis for AF queues with MMPP traffic with a general packet size distribution. Such an analysis remains open for future work.

### 6.3 Relative Proportion Study

One of the fixed parameters in the previous simulation study is the percentage of high priority packets within the aggregate. It remains to be investigated whether a desired performance could be achieved with a different percentage of high priority packets within the aggregate. Figure 6.5 shows that when the link capacity is reduced from 150% of the arrival rate (i.e. utilisation is more than 0.66), the loss probability of low priority packets increases rapidly over the desired level (zero losses for Web traffic and less than 0.03% for voice traffic). This section determines which percentage of high priority packets results in the best performance of a TD queue with a fixed link capacity. Again, the aggregate consists of two traffic classes with MMPP arrivals and the 4-modal packet size distribution obtained from Telstra traces.

The simulation study presented in Section 6.2 is modified to study the effects of the proportion of high priority packets within an aggregate (with a fixed utilisation) on the loss performance of both traffic classes. The delay performance, on the other hand is bounded by the buffer size and threshold of the queue. Simulation results show that for an utilisation of 0.65, there is no loss for high priority packets. The loss probability of low priority packet is drawn as a function of the proportion of high priority packets within the aggregate in Figure 6.6. It can be seen from Figure 6.6 that the desired performance (essentially small loss for low priority packets) can be achieved over a wide range of high priority packet proportions. Over the range [0%,85%] the loss probability for low priority packets is less than 0.03%, while a loss of less than 0.1% is acquired within the range [0%,95%]. It is consistent with observations from the previous section that if the utilisation is kept less than 0.66, a performance of zero high priority packet loss and very small low priority packet loss can be attained.
6.4 Conclusions

This chapter shows that for MMPP arrivals with the 4-modal packet size distribution, the steady-state performances (packet loss probability and mean delay) of a RIO queue can in most cases be approximated by a TD one, with the TD equivalent threshold values determined from the corresponding RED parameter sets. As a result, the network provider can use the analysis for a TD queue to estimate the steady-state performance of a RIO queue. However, the RIO queue is expected to provide a more stable performance and to be less biased against bursty traffic. Also examined in this chapter is the capacity planning of a TD queue (with given buffer configurations and QoS targets) and the effect of relative traffic proportions on the performance of the queue. It is found that for a particular buffer size and threshold, the bandwidth to be allocated needs to exceed a given level to achieve acceptable packet loss performances. For example, a scenario of TCP/voice traffic (80% are high priority TCP packets) requires sufficient bandwidth to keep the utilisation of the queue less than or equal to 0.66. Moreover, for this bandwidth, the relative traffic proportions do not significantly affect the queue performance.
Chapter 7

Conclusions

7.1 Overview

Differentiated Services aims to provide differential packet forwarding treatments (AF, EF and PF) to traffic flows, to provide a range of services to customers. Poor scalability, the major setback of core IntServ networks, is addressed in DiffServ by providing a forwarding treatment to an aggregate of flows with similar quality of service requirements. DiffServ can be used in conjunction with MPLS to provide differential services based on the efficient label switching mechanism.

The two main DiffServ service classes comprised in DiffServ are Assured Forwarding (AF) and Expedited Forwarding (EF). EF and AF provide different packet forwarding treatments. While EF aims to provide a low loss, low delay and low jitter service, AF offers different levels of forwarding assurances for data packets. In EF, packets are transferred at a rate equal to or greater than a pre-configured rate, while AF packets are given different drop-precedences at the edge of the network and dropped accordingly during congestion periods. AF is less stringent than EF and provides flexibility in terms of loss and delay. The EF service class is intended for more critical traffic types such as routing updates. On the other hand, the AF service class is suited to a different range of applications where users can choose either a low delay and loss tolerant service (e.g. for interactive real-time applications) or a low loss and relatively delay tolerant service (e.g. for TCP traffic). Furthermore, several AF service classes can be implemented independently and within these classes, the
preferred packet treatment is indicated by the drop-precedences. Hence, this research focuses on the performance analysis of the Assured Forwarding service class.

Since the introduction of DiffServ and its Per-Hop Behaviours, there have been some studies to investigate its performance. In particular, the performance (loss and delay) of a DiffServ node with different traffic models is a key factor in its dimensioning. While the DiffServ architecture and implementation have developed considerably over the last few years, the performance analysis of DiffServ nodes remains an open area. In particular, the following issues need to be investigated:

1. The performance of an AF DiffServ node with different traffic models, their appropriateness and the effects of a real-life packet size distributions.

2. The performance of different AF queueing mechanisms and associated AF dimensioning mechanism to meet desired QoS.

This thesis aims to address these issues. In addressing the first issue, a Poisson based and an MMPP based performance analysis of an AF node were derived. These traffic models are considered since results in [5] suggest that a Markovian model can be used to model self-similarity and long-range dependence traffic (common properties of network traffic [2], [3], [4]) in a finite buffer queue [5]. This model has to capture the correlation structure of the traffic up to a "correlation horizon" [5] or a "critical time scale" [6], which is dependent on the correlation structure of the input traffic and the maximum queue size. The Poisson and MMPP models are well known and can be used to model voice and data traffic ([7], [8]). This thesis also investigates the appropriateness of the Poisson based analysis with different sized MMPP aggregates. A 4-modal packet size distribution is considered in our simulations to investigate the effects of a real-life packet size distribution on the performance of an AF node.

The performances of two different AF queueing mechanisms (TD and RIO) are compared through a simulation study to address the second issue. This simulation is extended to consider a scenario where network utilisation is determined to meet certain QoS requirements. The traffic considered has MMPP arrivals and a 4-modal packet size distribution obtained from Telstra traces. This simulation study also investigates the effects of relative traffic proportion on the received performances.
Our results, recommendations and other areas for future work are described in the following sections.

7.2 Results and Recommendations

This thesis analyses an AF DiffServ node, with either a TD or a RIO queue, and with Poisson and MMPP arrival traffic with an exponential packet size distribution. These analyses were verified with simulations. The results and recommendations from this research are as follows:

- Chapter 4 presents a Poisson based performance analysis of an AF DiffServ node (TD and RIO) with exponential packet size distributions. This analysis is extended from [11] to consider three drop-precedences of an AF class. This chapter also investigates the performance of an AF node with real-life Telstra traffic traces, which exhibit a 4-modal packet size distribution. Simulations show that the packet size distribution significantly affects the performance of the node. This analysis is also used to estimate the loss and delay performance of an AF node on MMPP aggregates. It is found to be suited for approximate calculations for only large MMPP aggregates. Hence, an MMPP based analysis is needed.

- Chapter 5 presents an MMPP based performance analysis of an AF DiffServ node (TD and RIO) with an exponential packet size distribution. This analysis is extended from [13] to consider multiple flows with multiple-state MMPP arrivals. Since the aggregate of two MMPPs is an MMPP, this analysis can be used to provide close performance estimations for any MMPP aggregate.

- Chapter 6 presents a simulation study to compare the performances of RIO and TD queues on MMPP traffic with a 4-modal packet size distribution. The threshold values of the TD queue can be determined from the corresponding RED parameters of the RIO queues. Simulations show that for most situations, TD and RIO queues provide similar steady-state performances though the RIO behaviour is more stable. Hence, a TD queue is recommended to approximate
the steady-state performance of a RIO one. The simulation also considers a scenario where a mixture of 80% TCP and 20% UDP traffic (for high and low priority flow respectively) is queued at a TD node. In this scenario, acceptable performance (very small loss and bounded mean delay) is achieved when the queue utilisation is kept under 0.66. At this utilisation, the effects of the proportion of high priority packets on the node’s performances become insignificant.

7.3 Conclusions and Future Work

The main contribution of this thesis is the performance analysis of an AF DiffServ node, which implements either a RIO or a TD queue. In particular, this analysis models the performance of an AF DiffServ node with different classes and drop-precedences. The AF packet forwarding treatments is suited to a wide range of applications, where users can choose either to tolerate loss or delay. Our choice of traffic model is based on the fact that at a finite buffer node, the effects of correlation (an underlying property of long-range dependence traffic) becomes insignificant beyond a certain time scale. Hence, Markovian models such as Poisson and MMPP, are used to provide mathematically tractable solutions.

Our simulation study also shows that a RIO queue can be approximated by a TD one. The choice of RIO over TD is generally based on its more stable performance. Hence, instead of directly dimensioning a RIO queue, a network provider may dimension a TD one (which is simpler) and then determine the threshold parameters of the corresponding RIO queue. The simulation study also shows that for a traffic mix of 80% TCP Web traffic and 20% UDP voice traffic, the utilisation of the link must be kept no larger than 0.66 to obtain very small losses and bounded mean delays (e.g. 20ms for Web traffic and 5ms for voice traffic).

The analyses in this thesis together with these observations and recommendations, can assist the dimensioning for DiffServ networks. However, there are areas which need further investigation:
• **Analysis Extension**: In our theoretical performance analyses, the packet sizes (hence packet service time) are assumed exponentially distributed. Our study on Telstra traces shows that the packet sizes follow a 4-modal distribution (see Figure 4.8). Therefore, it is important to extend these analyses to take into account the general distribution of packet sizes, which has not been considered in this thesis. An extended analysis can be used to estimate the performances of DiffServ nodes on real-life traffic. Previous research works [58], [59], [60] can be extended since they consider a FIFO queue with MMPP arrivals and general service time distributions.

• **Per-Domain Performance**: The per-hop performance is the building block of per-domain performance, which has not been adequately studied in literature. Hence, per-node performance analysis can be extended to determine the performance of a domain (i.e. a series of DiffServ nodes) and its effects on traffic flows, especially bursty ones. This development can be a significant contribution to the dimensioning process of end-to-end DiffServ networks to meet the desired QoS.
Bibliography


Differentiated Service Performance Analysis

Long V. Nguyen, Tony Eyers, Joe F. Chicharo
The Switch Network Research Centre
University of Wollongong, Australia
long@snrc.uow.edu.au

Abstract

Differentiated service (DiffServ) has been proposed as an alternative for Integrated Service. It aims to provide the same service to a group of flows that have similar Quality of Service requirements. Assured Forwarding (AF) and Expedited Forwarding (EF) are two proposals for DiffServ provision. We present a performance analysis of an N drop-precedences Threshold Dropping (TD) queue, which is one of the proposed mechanisms for AF. In this analysis, traffic flows are assumed Poisson with exponentially distributed service time. We present simulations results that verify the analysis. This paper is an extension of the work attempted by Bolot at al [8] and Sahu [9] since it considers the general case with multiple classes of flow. We also show that the Poisson base analysis can be shown to hold for aggregation of bursty Markov sources in some cases and not to hold in others.

Keywords—Threshold Dropping, Differentiated Service, Quality of Service, drop precedence, loss probability, expected delay, poissonian hypothesis and On-Off traffic

1. Introduction

The current Internet provides Best-Effort service with no specific performance guarantees for individual application. The IETF Integrated Service (IntServ) working group, formed to address this issue, has produced RSVP and service classes such as Guaranteed-QoS, Controlled-Load. IntServ uses RSVP to provide network resources for individual flows, thereby producing significant overhead traffic. It is impossible to implement RSVP and IntServ in wide area networks due to its poor scalability. In such networks, a router will have to support thousands or even millions QoS connections and becomes more and more complicated as the number of connections increase.

Differentiated service was proposed as an alternative to IntServ. It aims to provide the same service to a group of connections that have similar QoS requirements (whilst IntServ guarantees service requirement for individual connections by using RSVP). This helps lower the overhead, as network nodes have to handle only a small number of aggregations. Hence, it improves the efficiency of the network and DiffServ should also scale well in a larger network.

Recently, there have been two proposals for DiffServ provision: Assured Forwarding (AF) and Expedited Forwarding (EF). The AF schemes offer different levels of forwarding assurance for data packets received from a customer DiffServ domain [5]. In the current definition of AF, there are 4 traffic classes and within each traffic class, there are 3 drop precedences [6]. Packets of different application are given different drop precedence. More AF classes or levels of drop precedence may be defined for local use. Moreover, a DiffServ node must allocate a configurable, minimum amount of forwarding resources (buffer space and bandwidth to each implemented AF class [6]. Examples of AF mechanisms are Threshold Dropping, Random Early Detection In-profile/Out-profile (RIO) [8].

Meanwhile, in EF schemes higher priority packets receive preferential link access over lower priority packets [7]. During congestion periods, bandwidth is reallocated from low priority flows to high priority flows to minimise the delay and delay jitter [5]. Examples of EF mechanisms are Class Based Queueing (CBQ) [7] and
Priority Queuing [8]. In comparison, AF is a simpler mechanism to implement than EF since AF's buffer management is simpler than EF's packet scheduler. Moreover, low priority flows in AF are not significantly affected by higher priority flows.

Threshold Dropping (TD) is a queuing mechanism proposed to implement AF DiffServ. In a TD node, there is a buffer threshold assigned to each level of drop precedence [8]. IP Packets with higher drop precedence are more likely to be dropped during congestion. Within a class, flows of similar QoS requirements are given the same drop precedence (i.e. the same buffer threshold). A packet is discarded when the buffer exceeds the threshold corresponding to its drop precedence at its arrival.

Traffic characteristics are important parameters to determine the performance (loss probability and mean delay of packets) of a network. There have been a number of traffic models (eg. Poisson, MMPP, Gamma, etc) proposed to capture the characteristics of IP packets in a network. Hence, it is important to analyse the effect of traffic models on a network's performance so that Internet Service Provider can dimension and design DiffServ networks accordingly. In this paper, we will present an analytical approach to estimate packet loss probability and mean delay for poisson traffic (an well known model) when applied to the Threshold Dropping associated with DiffServ. The TD queue can be considered as an AF class with a configurable amount of forwarding resources and a number of drop predecences. This paper is an extension of the work attempted by Bolot at al [8] and Sahu [9] since it considers the general case with multiple classes of flow. We also show that the Poisson base analysis can be shown to hold for aggregation of bursty Markov sources in some cases and not to hold in others.

This paper is organised as follows: in Section II we present our analytical approach to calculate the packet loss probability and expected delay for an N drop-predecences TD queue (extension from the 2 drop-predecences TD queue). Section III presents simulation results that confirm the analytical results presented in Section II. Section III also presents simulation results for aggregation of MMPP traffic sources hence highlighting the validity of the analytical results. Section IV concludes the paper.

2. Analysis

In [8] the authors suggested a general approach for loss probability and expected delay calculations for AF mechanisms. In this Section, we extend this analytical approach for a TD queue with Poisson arrivals to the N drop-predecences case. Some adjustments for the mean delay calculation are also added.

In an N drop-predecences TD queue (Figure 1), there are N flows (each flow corresponds to a level of drop precedence) arriving at the queue. A packet is discarded at its arrival when its corresponding buffer threshold has been reached or exceeded.

**Figure 1. A Threshold Dropping Queue with N drop-predecences**

This paper presents our analysis with the assumption that the incoming traffic flows are Poisson. We introduce the following terms:

- The arrival rate of the i\textsuperscript{th} priority flow is \( \lambda_i \).
- The packet service times are exponentially distributed service times with mean \( 1/\mu \).
- The loads of the i\textsuperscript{th} priority flow and the aggregation are \( \rho_i \) and \( \rho \) respectively.
- The buffer threshold of the i\textsuperscript{th} priority flow is \( L_i \) packets (\( L_0 = 0 \))
- At steady-state, the probability that there are \( n \) packets in the system is \( \Pi(n) \).
- \( \alpha(n) \) is the acceptance probability of a packet which arrives to the queue seeing \( n \) other packets already in the system.
- \( \alpha_i(n) \) is the acceptance probability of an i\textsuperscript{th} priority packet which arrives to the queue seeing \( n \) other packets already in the system. For a TD queue, this probability can be determined as

\[
\alpha_i(n) = \begin{cases} 
1 & \text{if } n < L_i \\
0 & \text{if } L_i \leq n
\end{cases}
\]

\( p_i \) is the ratio of the i\textsuperscript{th} priority flow's load to the overall load. Hence, \( p_i \) is the ratio of \( \lambda_i \) over the sum of all arrival rates.

It is important to notice that the lower the drop precedence of a flow, the higher the priority of the flow (eg. the 1\textsuperscript{st} priority flow has the lowest drop precedence and a buffer threshold of \( L_0 \), which is the buffer size of the queue). From the definition of \( \alpha(n) \) and \( \alpha_i(n) \) we have

\[
\alpha(n) = \sum_{i=1}^{N} p_i \alpha_i(n)
\]

and

\[
\alpha(n) = \left\{ \begin{array}{ll}
p_1 + \ldots + p_N & \text{if } n < L_1 \\
p_1 + \ldots + p_N & \text{if } L_1 \leq n < L_2 \\
\ldots & \ldots \\
p_1 + \ldots + p_N & \text{if } L_{k-1} \leq n < L_k \\
p_N & \text{if } L_{N-1} \leq n < L_N \\
0 & \text{if } L_N = n
\end{array} \right.
\]

329
It can be seen that this TD queue can be modelled as a birth-death process. For a state \( n \), the birth rate is \( \rho \alpha(n) \) while the death rate is \( \mu \). The steady-state distribution of buffer content is:

\[
\Pi(n) = \Pi(0) \rho^n \prod_{i=0}^{n-1} \alpha(i)
\]

with the probability that the buffer is empty \( \Pi(0) \)

\[
\Pi(0) = \left[ \sum_{n=0}^{L} \rho^n \prod_{i=0}^{n-1} \alpha(i) \right]^{-1}
\]

or

\[
\Pi(0) = \left[ 1 + \sum_{n=0}^{L} \prod_{j=0}^{n} (\rho_j + \ldots + \rho_N) \frac{L_n - L_{n-1}}{L_{n-1}} \sum_{k=0}^{L_n - L_{n-1}} (\rho_k + \ldots + \rho_N) \right]^{-1}
\]

From (3) and (4), we obtain:

\[
\Pi(n) = \Pi(0) \prod_{j=0}^{n} (\rho_j + \ldots + \rho_N) \frac{L_n - L_{n-1}}{L_{n-1}} \sum_{k=0}^{L_n - L_{n-1}} (\rho_k + \ldots + \rho_N) \text{ if } L_{n-1} < n \leq L_n
\]

The loss probability of the ith priority flow is determined as:

\[
\text{Loss}_i = 1 - \sum_{n=0}^{L} \alpha_i(n) \Pi(n)
\]

or

\[
\text{Loss}_i = 1 - \sum_{n=0}^{L} \Pi(n)
\]

Using (6), (7) and (8), loss probability of the ith priority flow is:

\[
\text{Loss}_i = \Pi(0) \prod_{j=0}^{n} (\rho_j + \ldots + \rho_N) \frac{L_n - L_{n-1}}{L_{n-1}} \sum_{k=0}^{L_n - L_{n-1}} (\rho_k + \ldots + \rho_N)
\]

Clearly, when a packet arrives at the queue which already has \( n \) packets, it has a delay of \( n \) packets service times plus its own service time. Therefore, the mean delay of the ith priority flow (excluding rejected packets) is:

\[
\text{Delay}_i = \frac{1}{\mu} \frac{\sum_{n=0}^{L} (n+1) \Pi(n) \alpha_i(n)}{\sum_{n=0}^{L} \Pi(n) \alpha_i(n)}
\]

Substitute the values of \( \alpha(n) \) into the above equation, we have

\[
\text{Delay}_i = \frac{1}{\mu} \frac{\sum_{n=0}^{L_n} (n+1) \Pi(n)}{\sum_{n=0}^{L_n} \Pi(n)}
\]
In these mean delay calculations, discarded packets are not included since the retransmission mechanism is not defined while in [8], the authors accounted for discarded packets. However, in [9], the authors provided delay calculations that include the probability for the system to be empty. The calculations for mean delays are:

\[
\text{Delay}_{\text{high}} = \frac{\Pi(0)}{\mu} \left( 1 + \sum_{n=1}^{K-L} (\rho_n + \rho^*)^n \right)
\]

and

\[
\text{Delay}_{\text{low}} = \frac{\Pi(0)}{\mu} \left( 1 + \sum_{n=1}^{L} (\rho_n + \rho^*)^n \right)
\]

(19)

(20)

These calculations imply the proportionality of mean delay to \( \Pi(0) \). Hence as load increases, \( \Pi(0) \) approaches zero and so does the expected delay. This contradicts the observation that the expected delay approaches \( K/\mu \) for high priority packets and \( L/\mu \) for low priority packets (Little's theory). In our analysis, the term \( \Pi(0) \) is cancelled since it appears in both the numerator and denominator of the calculation.

In the next Section, we will present simulation results that verify our theoretical analysis. We will also compare our mean delay calculations with [8]'s delay calculations. Also, simulation results will be shown to raise the question if the analysis develop for Poisson traffic can be applied to aggregates of bursty Markov sources.

3. Simulation Results

In this section, we present the results obtained from simulation, compare them with analytical calculations. Also, we show our observation that the Poisson based analysis in some cases hold for bursty input traffic. In our simulation, data packets (packet size and time stamp) are generated based on the traffic model (Poisson and 2-state On-Off) to form traffic flows. These flows are fed into a TD queue model and simulation results (loss probability and mean packet delay) are measured and compared with estimated ones.

3.1. Verification of the analysis

A simulation was developed to obtain experimental results and verify our analytical approach. The results are used to compare with the calculation provided by Bolot et al [8]. This simulation was developed for the 2-class case and expanded to a multiple priority case. We repeat the experiment in [8] by introducing a high priority and a low priority flow to the TD queue. High priority packets account for 95% while low priority packets account for 5% of the load. The buffer size and threshold was set to 16 and 6 packets respectively since the traffic flows consist of a single source.

Figure 2 shows simulation results in comparison with our analysis while Figure 3 compares simulation results with [8]. The expected delay presented in our figures is normalised with respect to the mean service time of the queue \( 1/\mu \). Figure 2 shows the loss probability and expected delay (for both priorities) obtained from the simulation and our analysis. It can be seen that our analyses closely match with simulation results (the points are on top of the analytical graphs).

\[ \text{Figure 2. (a) Loss Probability and (b) Expected Delay} \]

(normalised with respect to Packet Mean Service Time) of High and Low Priority Packets as a Function of the Total Load - Analytical and Simulation Results. High Priority Packets Contribute 95% of the Load \( K=16; L=6 \)

\[ \text{Figure 3. Expected Delay (normalised with respect to Packet Mean Service Time) of High and Low Priority Packets as a Function of the Total Load - [8]'s Approach} \]
and Simulation Results. High Priority Packets Contribute 95% of the Load (K=16; L=8)

We have established that the delay calculation presented by Bolot et al [8] matches with the results obtained when discarded packets are included. Figure 3 shows that during the period while the queue is not heavily loaded, the calculated results are close to simulated ones since the number of discarded packets are insignificant. However, as the queue becomes heavily congested the delay calculation of Bolot et al [8] approaches zero since there are less and less packets that get accepted. Meanwhile, the normalised expected delay (excluding discarded packets) approaches the buffer threshold value.

The simulation was also developed to verify our analytical approach to determine loss and delay in a multiple priority TD queue. In this simulation, a 3-priority TD queue was implemented with 3 single-source flows corresponding to the 3-drop priorities. Medium and low priority flows are set with load of 0.7 and 0.4 accordingly while the load of the high priority flow is varied from 0.1 to 0.9. The buffer thresholds were set at 16, 12 and 8 packets for high, medium and low priority flow respectively. The loss probability and expected delay were measured and plotted as a function of the high priority load in Figure 4.

It can be seen that the analytical results matches those obtained from simulation. Hence, this validates our analysis for a TD queue with a generalised number of drop precedences. In a DiffServ environment, this can be useful to estimate loss and delay where for each service class data packets are associated to more than two levels of discarding priority.

3.2. Poisson based analysis with bursty input sources

The analysis presented in Section II was developed with the assumption that input traffic are Poisson. It is clear that this Poisson based analysis does not cover the case when the input is a single bursty flow such as MMPP. However, if we alter the experiment by replacing the Poisson input sources by aggregations of bursty sources such as MMPP or On-Off, calculation and simulation results are shown to match in some cases and not in others. The Poisson parameter of the traffic model is calculated based on the On-Off parameters. Figure 5 presents loss and delay obtained from simulation with 2 single-source flows (high and low priority). The bursty sources are On-Off with duty cycle of 50%. These two sources are selected such that the high priority source contributes 95% of the total load while the low priority source contributes 5% of the total load.
are On-Off with High Priority Packets contribute 95% of the Load (K=16; L=6).

In this simulation the overall load of the TD queue ranges from 0.1 (lightly loaded) to 2 (heavily loaded). It can be seen that there is a difference between analytical and simulation results for both loss and delay. However, if the single bursty sources are replaced by aggregation of bursty sources (in the simulation, 20 identical On-Off sources are treated as a flow), the obtained simulation results are close to our predicted results. Figure 6 shows the match between simulation results and analysis. Each On-Off source is the same as in the previous example. The high priority flow still contributes 95% of the packets while the low priority flow contributes only 5%.

The high priority flow still contributes 95% of the Load (K=16; L=6)

Hence, it can be seen that for some cases (with large aggregation of bursty sources) the Poisson analysis holds while in other cases (small aggregations), it does not. As a result it is necessary to investigate the performance of a TD queue with other traffic models.

4. Conclusions

This paper has presented an analytical approach to determine QoS metrics of a TD queue (loss probability and expected delay). This method can be applied to provide the solution for a TD queue with multiple discarding priority TD queues. We have also corrected the discrepancies of the delay calculations provided in [8] and [9]. Our analytical calculations were verified with simulation results in Section III.

It is noted that our analytical approach allows a service provider to determine what performance is expected based on the traffic parameters as well as network resources in a multiple priority situation. These calculations can be used to help dimension the network to satisfy QoS requirements: loss and delay.

Moreover, we showed that the poisson hypothesis cannot hold for single bursty traffic flow yet it can hold for large aggregation of bursty sources. The question arises is to determine the sufficient size of the aggregation so that the poisson analysis can be applied. Also, it emphasises the need to investigate DiffServ performances with different traffic models.

5. References
