EDGE-BASED POINT-TO-MULTIPOINT QUALITY OF SERVICE GUARANTEES IN PRIVATE NETWORKS

By

Satish Raghunath

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer and Systems Engineering

Approved by the
Examining Committee:

Prof. Shivkumar Kalyanaraman, Thesis Adviser

Prof. Koushik Kar, Member

Dr. K.K. Ramakrishnan, Member

Prof. Kenneth Vastola, Member

Prof. Bulent Yener, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2004
(For Graduation May 2004)

EDGE-BASED POINT-TO-MULTIPOINT QUALITY OF SERVICE GUARANTEES IN PRIVATE NETWORKS

By

Satish Raghunath

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer and Systems Engineering

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Prof. Shivkumar Kalyanaraman, Thesis Adviser
Prof. Koushik Kar, Member
Dr. K.K. Ramakrishnan, Member
Prof. Kenneth Vastola, Member
Prof. Bulent Yener, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2004
(For Graduation May 2004)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I express my heartfelt gratitude to all those who were a source of support during my doctorate. I would especially like to thank Kartikeya Chandrayana for his help during various phases of my work and the numerous stimulating discussions. This thesis would not have been possible without the expert guidance and support from my advisor Prof. Shiv Kalyanaraman. I have had the good fortune of working with top networking researchers of the field at AT&T Labs and utilize data and computing resources at AT&T. In addition to material support, I have immensely benefited from the guidance and inputs I received while collaborating with Dr. K.K. Ramakrishnan from AT&T Labs. Finally, I would not be here without the infinite good wishes, sacrifices and support from my family. Thank you.

ABSTRACT

Virtual Private Networks (VPNs) have become the solution of choice for site-to-site enterprise networking needs. VPNs allow geographically spread-out locations to communicate in a secure and reliable fashion. Service providers have recognized that with increase in the number of VPNs and their size, simple meshed architectures that connect customer sites with peak bandwidth reservations are not scalable solutions. The current growth in size and complexity of customer networks can be sustained only if efficient solutions are evolved to design and provision VPNs.

In this thesis we design, implement and evaluate new architectures to provision IP VPNs. Any solution that is deployable features minimal and inexpensive upgrades to existing hardware and software of the core networks. Thus the main theme of our solutions is the approach of leveraging the existing core network infrastructure and building new intelligence at the network edge, hence the term "Edge-based".

Quality of Service (QoS) metrics, e.g., probability of loss and delay, are employed to measure the reliability of a service. There are important limitations in existing methods to assure QoS with reference to next generation networks: a) Traditional QoS models have either concentrated on point-to-point models or have relied on signaling-based mechanisms in the core network. However, customer networks employing VPNs frequently need assurances toward multiple destination sites and providers need architectures that do not require complex mechanisms like signaling in the core network; b) Existing literature relies on precise specification of customer traffic to provide bounds on QoS metrics. Often service providers do not have such information.

In this thesis we propose new models to provision point-to-multipoint QoS that overcome these limitations. By expanding the spatial granularity of QoS assurances, we eliminate the need to provision point-to-point links for each source-destination pair while still providing statistical assurances on QoS. Since our architecture is completely edge-based, it is easily deployed without any changes to the network

core.

By decoupling the computation of QoS assurances from instantaneous traffic characteristics, we are able to provide bounds on QoS metrics independent of the nature of customer traffic information. In order to provide such a-priori assurances with an edge-based architecture, we build an analytical framework that allows us to compute QoS metrics without needing to know the exact nature or number of the flows in the system. Our framework allows computing QoS assurances given just a topology and routing scheme with appropriate edge-based admission control components.

We then design and build techniques to realize the proposed framework with existing SNMP-based measurement infrastructure. Employing real SNMP measurements from a large IP VPN provider, we evaluate these techniques and articulate the impact of our proposals on provisioning, traffic engineering and enhanced operational efficiency.

In summary, this thesis provides a new edge-based architecture for efficient provisioning of a-priori point-to-multipoint QoS assurances and demonstrates the tools required for deployment with existing Internet Service Provider (ISP) infrastructure.

ABSTRACT

Virtual Private Networks (VPNs) have become the solution of choice for site-to-site enterprise networking needs. VPNs allow geographically spread-out locations to communicate in a secure and reliable fashion. Service providers have recognized that with increase in the number of VPNs and their size, simple meshed architectures that connect customer sites with peak bandwidth reservations are not scalable solutions. The current growth in size and complexity of customer networks can be sustained only if efficient solutions are evolved to design and provision VPNs.

In this thesis we design, implement and evaluate new architectures to provision IP VPNs. Any solution that is deployable features minimal and inexpensive upgrades to existing hardware and software of the core networks. Thus the main theme of our solutions is the approach of leveraging the existing core network infrastructure and building new intelligence at the network edge, hence the term "Edge-based".

There are important limitations in existing architectures with reference to point-to-multipoint QoS for VPN-like applications: a) they have either concentrated on point-to-point models or have relied on complex signaling-based mechanisms in the core network; b) they rely on precise specification of customer traffic to provide bounds on QoS metrics whereas such information is hard to come by in reality.

In this thesis we propose new models to provision point-to-multipoint QoS that overcome these limitations. We propose edge-based solutions that are easily deployed and managed. We outline mechanisms to learn customer traffic characteristics exploiting existing SNMP measurement infrastructure and examine the proposals with real data from a large IP VPN service provider. In addition we outline an analytical framework resulting in simple admission control conditions that facilitate decoupling computation of QoS assurances from instantaneous traffic characteristics. We are able to provide bounds on QoS metrics independent of the nature of customer traffic information given just a topology and routing scheme with appropriate edge-based admission control components.

In summary, this thesis provides a new edge-based architecture for efficient provisioning of a-priori point-to-multipoint QoS and the tools required for deployment with existing Internet Service Provider (ISP) infrastructure.

CHAPTER 1
Introduction

Since its inception, the Internet has grown into a large world-wide network serving a wide variety of applications. The remarkable size and heterogeneity in users of the Internet requires architectural proposals to be simple and platform-independent in order to be deployable. In their pioneering paper, Saltzer et al [103] formulated a set of design guidelines, called the *End-to-End principles*, for a general distributed system. Their observations regarding keeping the "lower layers" as simple and functionally complete as possible, have been reflected in the design philosophies that succeeded in the Internet.

While the cost of implementing a simple network is low, the lack of fine-grain functionality (e.g., per-flow resource management) implies that the network cannot provide high-confidence assurances on service quality. Additional mechanisms have to be deployed in the higher layers to realize such assurances on service quality. Such mechanisms involve regulating the traffic entering the network such that certain performance metrics are ensured to be within satisfactory levels. These regulatory methods are termed as *admission control* techniques. The Quality of Service (QoS) assured in such a case is a property of the admission control algorithm.

In this thesis, we examine the problem of providing a desired level of QoS by retaining the core of the network unchanged. We term such proposals as being "Edge-based" due to the fact that all required capabilities are deployed at the entry of the network (the *Edge*). We build the admission control conditions required to achieve a given level of QoS and demonstrate its working by evaluating them in a wide variety of practical scenarios. We show that the proposals work independent of traffic or topology specifics.

We begin by providing a brief background on existing work in QoS (§1.1). We then present the significance of the problems (§1.2) and novelty in the solutions (§1.3) presented in this thesis.

1.1 QoS Preliminaries

A QoS assurance specifies an allowable range of values for certain performance metrics. For a performance metric M representing delay, jitter or loss rate, the QoS assurance is either specified deterministically as $M \leq m$ or probabilistically as $Pr\{M > m\} \leq \epsilon$.

In order to evaluate the QoS metric on an end-to-end basis for path in the network, the effect of each hop has to be well understood. There have been a variety of approaches toward a solution to this problem [42]. Cruz [24, 25] pioneered a modular approach by proposing a "Network Calculus", which has over the recent years developed into a mature set of tools for delay analysis of networks [75]. These techniques allow every scheduler in a path to be abstracted by a *service curve*. The elegant mathematical properties of these service curves allow them to be easily convolved together to form an effective service curve for the whole path. The arrivals are characterized by generic sub-additive curves allowing for a wide range of input processes. Given an arrival curve and an effective service curve for the path it is possible to compute deterministic bounds on backlog, delay and rate assurances. One of the first proposals to demonstrate these techniques was the seminal work on Generalized Processor Sharing (GPS) networks due to Parekh and Gallager [91, 92].

In networks such as those described by the Integrated Services Architecture (Intserv [9]), network resources are provisioned per-flow. The Network Calculus framework elegantly solves the problem of analyzing the resultant cascade of single-flow single-server systems. However, the size of the Internet requires that the network architecture scale with very large number of flows. Consequently, a deployable architecture such as Differentiated Services (Diffserv [6]) resorts to class-wise resource allocation, where each class is an aggregation of flows and the number of classes remains small with growing number flows. With such a framework flows are no longer isolated, instead they are multiplexed at each hop with numerous other flows of the same class. Although the traditional network calculus techniques can be applied to arrive at assurances for each of the aggregates, the service characteristics experienced by the constituent flows can be vastly different [17].

When flows are multiplexed, there is an opportunity to exploit their statistical

characteristics to achieve utilization gains. For instance, if the probability of a flow transmitting at peak rate is low, the reserved capacity can be less than the peak rate, allowing the resource to be shared by higher number of flows. However, with deterministic analysis, the statistical nature of input flows cannot be exploited. In addition, each hop of multiplexing increases the burstiness of participant flows due to the fact that the packets of a given flow could potentially be delayed by a burst due to all other flows in the system. Thus multiplexing introduces complex distortions in the statistical nature of input flows. A deterministic analysis accounts for the worst-case possibilities. Hence, with increase in number of hops of multiplexing, upper bounds on delay and backlog obtained using deterministic techniques become very conservative [75, 135]. The achievable utilization levels are unacceptably low. Hence there are two principal concerns - handling per-hop increase in burstiness of flows and exploiting statistical multiplexing gains.

1.1.1 Handling Traffic Distortion due to Multiplexing

A remedy for the increase in burstiness of flows at each hop could involve - a) changing the behavior of each hop so that no increase in burstiness occurs, i.e., flow characteristics remain unchanged at the exit of a multiplexer; b) leaving the node architecture unchanged and improving analytical techniques to handle these distortions better.

1.1.1.1 Changing Node Structure

If each multiplexer ensures that the inter-packet time remains approximately the same as it was when the flow entered the network, the amount of distortion in flow characteristics is minimized. A family of scheduling disciplines, called Rate Controlled Scheduling Disciplines (RCSD), proposed by Zhang and Ferrari [130], attempts to achieve this goal. RCSD requires jitter controllers at each hop per-flow which ensure that the inter-packet times are retained intact. Stoica and Zhang [116] improve on this idea by demonstrating that introducing per-flow information in packets can help eliminate per-flow state in the core routers, thus making the proposal much more scalable. Their proposal involves upgrading the core routers to to conform to the Core Jitter Virtual Clock (CJVC) scheduling discipline which

look up packet headers to gather scheduling information. Kaur and Vin [60] adapt this proposal for work conserving schedulers called Core-stateless Guaranteed Rate (CSGR) schedulers.

In general, the distortions introduced by multiplexing are due to the manner in which flows are *aggregated*. Cobb and Gouda build a generic framework called the *Flow Theory* [22, 21] to analyze the effects of multiplexing on flow characteristics. They characterize the effects of using different strategies for aggregation and the impact on preserving flow character when these aggregators are coupled with schedulers. In the context of Diffserv, one could couple any of these aggregation techniques with the *Expedited Forwarding* per-hop behavior [26] (e.g., implemented using the Packet-Scale Rate Guarantee (PSRG) [1, 74] servers) to obtain end-to-end rate guarantees and delay bounds.

1.1.1.2 Retaining the Core Node Unchanged

If the core network is to be retained without any changes, one would have to consider a different set of options. Reisslein et al suggest using bufferless multiplexing [101] to avoid traffic distortion while maintaining a simple core network. In the presence of traffic distortions due to buffered multiplexing, any end-to-end QoS analysis has to quantify the effect of these distortions. Observing that the maximum burstiness increase at a hop is limited by the upper bound on the busy period at the multiplexer, Kurose [72] built an analytical framework to specify the worst-case traffic envelope after successive hops of multiplexing. A similar approach is adopted by Chang [15] to analyze backlog and delay in an end-to-end setting and by Vojnović and Le Boudec [121] to analyze a Diffserv EF network for end-to-end delay assurances.

1.1.2 Statistical Multiplexing Gains

In order to exploit the variations in input traffic to better share the network resource probabilistic techniques are employed. Specifically, an input process $X(t)$ is characterized by a probabilistic envelope like $Pr\{X(t) > f(t)\} \leq \epsilon$ instead of a deterministic envelope like $X(t) \leq f(t)$. The difference between the above specifications is enormous when it comes to resource provisioning. Put in simple terms,

provisioning according to a deterministic envelope implies taking care of all the worst-case possibilities, however small their chance of occurrence. On the other hand, a probabilistic envelope states that if the chance of an event occurring is less than some small ϵ it can be ignored. Using such a description of input traffic, the probability of violating a particular delay or backlog bound can be derived for use in admission control policies.

While the use of probabilistic envelopes yields efficient provisioning strategies, deriving these envelopes for Internet traffic (whose characteristics are assumed unknown) is not a trivial task. Early attempts to find probabilistic envelopes [72, 127, 110, 34] involved using bounding random variables, moment generating functions etc. Some of the recent proposals have focused on envelopes that can be easily enforced using deterministic components [67, 66, 7, 107].

From the standpoint of simple network implementations which exploit statistical multiplexing, deterministically enforceable probabilistic envelopes are highly desirable.

1.1.3 Putting it Together

The need for scalable network architectures forces a compromise in terms of the coarse granularity of network support for QoS. The problems facing current networks with flow multiplexing are complex, but well-defined. While some of the proposed solutions require an upgrade of existing networks to employ intelligent scheduling disciplines, many others attempt to provide a QoS assurance by quantifying the trade-off forced by aggregate scheduling.

1.2 Challenges Facing QoS Provisioning

The preceding discussion points to the conclusion that there are analytical techniques available to understand, to a limited extent, the complex nature of multiplexed networks. Thus by approximately computing flow characteristics at a given multiplexer inside the network, the flow's statistical envelope and hence the delay and backlog bounds can be obtained. Moreover, with flows sharing a link, statistical multiplexing gains can be obtained by employing these probabilistic envelopes

for admission control. Thus, the network can be analyzed by breaking down the problem into sub-problems dealing with individual links.

But such an approach falls short of providing a framework that can be managed at the edge of the network on multiple counts. First, QoS assurances and admission control decisions for the network have to be made on an end-to-end basis. The question then is what should be the target for QoS metrics at each hop; e.g., if we are building an end-to-end delay service, the admission control regime at a hop would have to test if a flow violates a *pre-determined* target delay at that hop. Existing techniques do not provide this translation of an end-to-end assurance to per-hop thresholds.

Second, in order to compute the metrics of interest, existing methods need the characteristics of each flow in the system. Further, computations need to be performed per-hop with the addition of a new flow. This implies that a network cannot be analyzed *independent* of what traffic it may carry.

Third, the preceding reference to statistical multiplexing gains only referred to those obtained due to *flows* sharing certain *links*. They do not exploit gains due to properties of traffic across paths originating from the same customer network. Observe that a customer network is limited by its access link. So increase in traffic due to a source network along one path, usually implies a reduction on another path. Thus there are gains to be tapped not only on a per-path basis, but also across paths.

Fourth, since traffic characteristics across multiple paths are not exploited, the spatial granularity of QoS is limited. That is, assurances are provided only on a point-to-point basis instead of the more natural (but hard to realize) *point-to-anywhere* basis.

The next step toward "better-than best-effort" service would be to address these concerns through scalable strategies. Thus we arrive at a set of desirable tools and algorithms:

1. *Expanded Spatial Granularity:* A typical source network features traffic toward multiple destination sites. A user would ideally wish to free himself of the need to setup point-to-point service agreements for each destination site. A service

with expanded spatial granularity specifies an assurance (e.g., a minimum bandwidth guarantee) toward multiple destinations.

2. *A Priori Assurances:* The provider network should ideally be able to quantify the QoS his network can provide for the purposes of advertising a service without knowledge of future traffic.

3. *QoS Assurances Decoupled from Traffic Profile:* The QoS offered by a network should be invariant with changes in incident traffic. A user subscribes to a service assuming that its parameters remain at promised levels for a reasonable duration. This implies that assurances that are computed for a network should not depend on the nature of traffic at that time.

4. *Edge-based Services Architecture:* The provider must be able to deploy strategies which achieve the above objectives without having to effect changes to the core network.

5. *Exploit Statistical Multiplexing Gains:* While conservative provisioning techniques can achieve many of the above objectives, the challenge is to maximize resource utilization while realizing these goals.

1.3 Contributions

In order to realize the goals that were articulated in §1.2, we proceed in a step-by-step manner, by tackling the simpler problems initially and then employing those results as building blocks for the rest. We thus start with an architecture that is edge-based but does not provide *a priori* assurances. We then examine deterministic assurances in a simplistic network without cross-traffic and just two flows. Exploiting the insight gained therein, we propose a framework which decouples traffic profile and QoS assurances. We summarize the contributions of this thesis in the following:

- *The Point-to-Set Architecture:* We begin by building a service with expanded spatial granularity. The QoS assurances provided by this architecture are not

a priori. The assurances however are toward a *finite set* of destinations in stark contrast to the existing point-to-point services.

1. The service is completely deployed at the edge of the network.
2. A simple probabilistic admission control regime helps exploit *intra*-path and *inter*-path statistical multiplexing gains.
3. A novel metric to compare the service with the point-to-point service is forged and called the *flexibility* of the service.
4. All enforcement components are simple deterministic shapers and policers.

- *A Deterministic Approach to Delay Assurances:* We apply network calculus to analyze a simplistic network consisting of a cascade of FIFO multiplexers with no cross-traffic. We demonstrate that with appropriate limits enforced on burstiness of admitted flows and number of hops in the path, delay bounds can be provided. We also find that a pure deterministic approach is very conservative to allow good resource utilization.

- *Statistical Delay Assurances Decoupled from Traffic Profile:* We develop a framework to obtain statistical delay assurances for an arbitrary FIFO network and provide a simple set of constraints to be enforced by the admission control module so that the assurances do not depend on traffic characteristics. The framework leads us toward realizing many of the goals discussed in §1.2:

  1. The admission control decisions are taken at the edge of the network and do not need per-hop computations.
  2. The resultant assurances do not depend on number of admitted flows or their specific flow characteristics.
  3. For a given topology, we are able to arrive a QoS assurances independent of future traffic character.

- *Trade-offs in Edge-based Architectures:* Having specified a model and verified its working in simulation, we examine the cost of opting an edge-based architecture. Although a simple core network leads to lower utilization as compared

to a signaling-based approach, we demonstrate techniques that can bridge the performance gap and still retain the benefits of deployability and scalability. We show that learning traffic matrix information and spatial structure of interactions can lead to remarkable improvements in performance.

- *Realizing an Edge-based framework: Traffic Matrix Estimation* We build algorithms and techniques that exploit existing measurement information to learn the parameters that are required to realize an edge-based architecture as described above. We demonstrate that using simple SNMP-based measurements we can realize and adaptively provisioned Point-to-Set architecture.

CHAPTER 2
Review of Quality of Service Proposals for the Internet

2.1 Summary

We present a survey of recent literature in the area of QoS. We classify the various approaches in the following manner:

- Scheduling disciplines
  - Minimum bandwidth and delay assured scheduling, Fair Queuing
  - Service-curve based scheduling
  - Hierarchical schedulers
- Source characterization
  - Deterministic envelopes to limit source burstiness
  - Statistical envelopes
- Network architectures
  - Per-hop re-shaping
  - Dynamic Packet State
  - Bufferless models and other proposals
- Admission Control
  - Measurement-based
  - Distributed and End-point based

2.2 Introduction

Application requirements for delay and rate guarantees have resulted in a lot of research interest in the area of Quality of Service [42]. At the highest level, the problem of QoS can be defined as that of providing assurances on loss, delay and

bandwidth. A wide variety of approaches have been proposed to realize a solution to the QoS problem. Most of the solutions attempt to either characterize the properties of the source or that of the network so that a statement on QoS can be made with certainty. By specifying properties of the sources, such as limits on burstiness, one can obtain the worst-case delay and loss rates at a queue. On the other hand, by building networks with schedulers which specialize in providing delay and rate assurances on a per-flow basis, one can obtain end-to-end QoS properties.

The present day networks feature a mix of both these streams of thought. In this chapter we will review the current literature relevant to these ideas. We begin by examining schedulers that attempt to provide minimum rate and delay guarantees (§2.3). We then examine ways to characterize sources so that the effects of multiplexing in a network can be quantified (§2.4). Finally, we examine network architectures which combine these techniques to provide end-to-end QoS guarantees (§2.5).

2.3 Scheduling

The simplest and most widely deployed scheduling discipline is the First-In-First-Out scheme. An important handicap of the FIFO scheduler is that it does not provide flow isolation. The packets of a particular flow might be arbitrarily delayed by those of other flows. The scheduling disciplines which we will review, attempt to provide flow isolation and/or service guarantees.

The Delay Earliest Due-Date (Delay-EDD [37]) is one such discipline which provides a worst-case delay bound. Incoming packets are tagged with a timestamp depending on the delay assured to the flow. The packets are then served in the order of the timestamps. A modified version of Delay-EDD, called the Jitter-EDD [120] provides not only a bound on the maximum delay, but also a bound on the deviation of delay. However, this service does not provide fair treatment to every flow. Fairness is quantified as the difference in service that two flows obtain from the server.

The Weighted Fair Queuing [27] (similar to Virtual Clock [132, 131]) algorithm was thus intended to provide a "rate proportional" service to flows in a system. It emulated a bit-by-bit round-robin scheme and hence attempted to provide fairness

and a minimum rate guarantee. There are also scheduling disciplines which combine some aspects of Virtual Clock and Jitter-EDD (e.g., the Leave-in-Time service [38]). Since the introduction of WFQ, there has been a lot of work to formalize and improve the properties of this algorithm.

2.3.1 Fair Queuing

Parekh and Gallager [91, 92] led the way in discovering the capabilities of ideal fair queuing disciplines. They proposed the Generalized Processor Sharing (GPS) discipline, which is an idealized version of WFQ. The most important observation was that the GPS server could provide flow isolation, a minimum rate guarantee and with leaky-bucket constrained sources it could provide bounded delay service. They demonstrated a network of GPS servers which could provide all these assurances on an end-to-end basis. The problem with GPS was that it was an ideal discipline which cannot be realized in practice. But they did provide a packetized version called the Packet-by-Packet GPS which involved computing *virtual finish times* for packets according to the ideal GPS server and then serving the packets in the order of these finish times. The overhead involved in computing the virtual finish time meant that it was not practical.

Golestani proposed one of the early simplifications to PGPS by approximating the virtual finish time by the finish time of a packet in service for a flow that was previously idle. Thus there was no need to simulate the ideal GPS discipline to compute the finish time making this a Self-Clocked Fair Queuing scheme [48, 49]. This approximation made it easier to implement PGPS but the worst-case delay and fairness characteristics degraded. Additionally, Bennett and Zhang [2] demonstrated that contrary to popular belief, WFQ could have larger than expected discrepancy in delay and fairness as compared to the ideal GPS. They proposed what is known as the Worst-case Fair Weighted Fair Queuing or WF^{2}Q. Start-time Fair Queuing [52], which used start-times to order the packets, was also proposed to fix SCFQ. Even though some of the fairness issues with WFQ and SCFQ were solved, the complexity of maintaining a sorted list of packets meant an $O(log\ N)$ computation.

In order to reduce the complexity of fair queuing, Deficit Round Robin [105]

scheme got rid of finish time computations to provide an $O(1)$ implementation. However, the trade-off was once again a lack of bounded delay and degraded short-term fairness coupled with increased burstiness. The Credit-based fair queuing [4] scheme attempts to remedy these problems in DRR and is able to provide the service equivalent to SCFQ at $O(1)$ when packets are of fixed size.

Thus all the proposals above had a performance at best equivalent to SCFQ in terms of delay bounds and fairness. Some of them had better complexity. Stiliadis and Varma [113] proposed a new class of rate-proportional servers which had both the attributes - better delay bounds and fairness than SCFQ while being of lower complexity. Frame-based Fair Queuing (FFQ) and Starting-potential based Fair Queuing (SPFQ) [111] are instances of such schedulers where the timestamp computation is $O(1)$. But the overall scheduling complexity still depends on maintaining an ordered list of packets which costs $O(log\ N)$ where N is the number of packets backlogged.

The preceding discussion points to the fact that timestamp-based algorithms incur an $O(log\ N)$ time complexity and remedies like DRR result in increased output burstiness. Recently a new scheme called Smoothed Round-Robin (SRR [18]) solved these problems - it provides an $O(1)$ scheduler with good short-term fairness and low output burstiness.

2.3.2 Service Curve based Scheduling

One common feature of these scheduling schemes is the nature of the *service curve*. The service curve defines the lower bound on the cumulative number of bits of a backlogged flow served in a given amount of time. The service curve offered by these schedulers is a straight line with slope given by the capacity. In general, a service curve can be any arbitrary non-decreasing function. Georgiadis et al attempted to relax the minimum rate guarantee by proposing the Guaranteed Rate service [45] where the rate guarantee was only on a sufficiently large interval of time. Goyal and Vin proposed generalizations to this model to handle time-varying rate allocations [51]. But these attempts did not explicitly deal with service curves.

Parekh and Gallager [91] introduced what they termed a Universal Service

Curve. This was a piece-wise linear curve which was the service offered by a backlogged GPS server. There were further generalizations by Stiliadis and Varma who proposed the class of Latency-Rate servers [112] which had PGPS, VirtualClock, DRR etc., as special cases. Cruz provided a service curve definition [104] which is now considered the most general definition. End-to-end bounds on delay and rate that were derived for Virtual Clock and other disciplines [39, 40, 125, 57] can be shown to be special cases of the results obtained using the general service curve.

Service curves allow the maximum flexibility in tailoring a network's offering to best fit the needs of the user and help in reducing over-allocation. Service curve based Earliest Deadline scheduling (SCED) has been demonstrated to be optimal in terms of schedulable region [44]. There has also been a lot of work to characterize the service curve of a network of nodes [75].

2.3.3 Hierarchical link sharing

The previous sections discussed scheduling disciplines which provided assurances per-flow. In practical situations there is a need for subdivision among constituents of a flow. E.g., a link might be shared by two customers and each customer might have multiple classes of traffic. To provide each customer with an overall rate guarantee which is then shared among its sub-classes, we need a hierarchical discipline. The Hierarchical Round-Robin (HRR [59]) and multi-frame Stop-and-go [47] are examples of non-work conserving hierarchical disciplines.

However, these schemes do not provide for fair and dynamic link-sharing. If there is unused bandwidth available, one would want to utilize it among backlogged flows according to some policy. Floyd and Paxson [43] outlined a link sharing scheme based on Class-Based Queuing (CBQ). A more formal framework based on fair queuing algorithms was formulated by Bennett et al [3]. Neither of these proposals decoupled bandwidth and delay allocations in a clear manner. The Hierarchical Fair Service Curve (H-FSC [117] was the first link sharing scheme to decouple delay and bandwidth allocation using a formal service-curve based approach. A similar functionality based on rate-controlled scheduling was proposed by Goyal and Vin [50]. Open issues in hierarchical link sharing involve fairness issues between

non-leaf nodes in the hierarchy.

In summary, recent work has lent us a good understanding of the complexity in providing bounded delay schedulers [126] and provided newer more efficient implementations. In the context of this thesis, the solution provided by specialized scheduling involves a network-wide upgrade and extensive changes to the core network. The focus of our work is to use the existing core network and examine edge-based solutions for the same problems.

2.4 Source Characterization

While providing QoS assurances there is often a need to precisely define the nature of traffic offered by sources. If properties like burstiness and average rate are known, one can derive bounds on delays that are incurred (e.g. GPS [91, 92]). Further, to characterize the effects of scheduling mechanisms on the properties of the original source traffic, we would need a precise way to describe sources. There have been two main approaches to this problem, one being deterministic and the other probabilistic. The former seeks to express bounds on the traffic with possibly time-varying but deterministic functions. The probabilistic techniques try to describe the source traffic through stochastic models or attempt to estimate various moments of the process (e.g., mean and variance).

2.4.1 Deterministic Envelopes

One of the most popular deterministic envelopes is the one induced by a leaky-bucket regulator. Such an envelope enforces an upper-limit on the burstiness of the source and the average rate. The notion of leaky-bucket regulation has been formalized by Cruz [24] and extended for the time-varying case by Chang et al [14]. Le Boudec [75, 8] summarizes a rich set of tools which form part of what is now known as "Network Calculus" based on deterministic envelopes including the leaky-bucket regulator. Konstantopoulos and Anantharam [69] prove that with the objective of minimizing delay or buffer size the leaky-bucket regulator is the optimal scheme for an output of desired burstiness. There have also been online procedures to estimate the burst parameter of a leaky-bucket source [119].

Despite these positive aspects, the leaky-bucket regulator captures only one timescale of the traffic properties and that is a major limitation when it comes to highly bursty sources like VBR video. Ferrari and Verma use a deterministic envelope which involves specifying the minimum and average inter-packet spacing, a maximum packet length and an interval length for rate computation. However even this model fails to capture multiple timescales of burstiness. D-BIND [65] provides piece-wise linear characterization which remedies this problem. A similar model is a bank of leaky-bucket regulators in parallel as used in [101]. Empirical models also have been examined with demonstrable gains for VBR video traffic [124].

But the most popular and easily deployed solution remains the leaky-bucket shaper. Though deterministic envelopes are easy to deploy and analyze, admission control and provisioning based on deterministic envelopes leads to low utilization [68]. This leads us to the need for statistical descriptions of traffic.

2.4.2 Statistical Envelopes

One of the early efforts at statistical description of sources was due to Kurose [72], involving a set of bounding random variables. Yates et al [128] followed up on this work and examined end-to-end delay distributions. It was quickly realized that finding random variables that upper-bound traffic is, in general, a hard problem. Yaron and Sidi [127] provided a generalization of the leaky-bucket envelope, called Exponentially Bounded Burstiness (EBB), by specifying the probability $Pr\{\int_s^t R(\tau)d\tau \geq \rho(t-s)+\sigma\} \leq e^{f(\sigma)}$. Starobinski and Sidi extended this specification to a more general set of bounding functions called Stochastically Bounded Burstiness [110]. Although a calculus [127, 77] for multi-hop delay computation was demonstrated with this framework, it was hard to enforce or find appropriate envelopes for traffic in practice.

Knightly [66] proposed bounding the second moment of the traffic by assuming that the traffic was shaped by a leaky-bucket shaper. These envelopes known as Rate Variance envelopes had the advantage that they were easily enforceable using the underlying leaky-bucket parameters. These bounds were later derived in a more general setting by Boorstyn et al [7] as a special case of Effective Envelopes.

Boorstyn et al [7] generalized the deterministic network calculus to obtain statistical bounds on delay and backlog using these envelopes.

Another alternate approach has been to employ bounds on the moment generating function of the traffic proposed by Kelly and popularly known as effective bandwidth [61]. There has been rich set of tools developed to estimate the effective bandwidth for both Markovian and non-Markovian sources [64, 13, 70]. These envelopes have been successfully employed in analyzing buffer overflow probabilities in ATM multiplexers [34].

Modeling sources as either on/off processes [76, 106, 33] or fractional Brownian motion [89] queuing analysis has been carried out using results from large deviations theory [123]. Traditional queuing analysis has also been applied to quantify traffic distortions due to multiplexing [83, 73].

In summary, statistical envelopes are most useful when they can be easily enforced on sources or readily derived from traffic measurements. A wide range of theoretical tools have been developed and applied toward computing probabilistic delay and backlog bounds. In this thesis we use rate variance statistical envelopes to derive bounds on delay.

2.5 Network Architectures

Architectural proposals aimed at achieving end-to-end QoS distinguish themselves in the manner in which they handle scalability and heterogeneity. One of the early proposals was the Integrated Services architecture (Intserv [9]) which aimed to provide fine-grain QoS at the flow-level. Intserv featured per-flow resource reservation using a signaling protocol. Although it has been shown that such a reservation-based approach performs better than a "best-effort" service [12] in terms of QoS seen by the flow, the implementation complexity in the presence of a large number of flows is too high. Clark et al [20] proposed the first alternative to such a guaranteed service and called it "predictive service". This was meant for adaptive applications which could tolerate some variation in QoS.

The Differentiated Services architecture (Diffserv [6]) gives up the fine granularity of Intserv and achieves scalability. Each hop in a Diffserv network provides

class-based services and these classes are intended to be aggregates of flows. The number of classes are expected to be much smaller than the number of flows in the system. While the system is scalable, the service provided to each flow is not clearly quantifiable. Multiplexing at each hop distorts the properties of flows and the problem of quantifying such distortions is usually very complicated when source characteristics are unknown. With the acceptance of such aggregate scheduling mechanisms, there was a need to investigate mechanisms to preserve flow characteristics or at least quantify the effective service provided to individual flows.

Zhang and Ferrari [130] propose to use a class of schedulers called the Rate-controlled Service Disciplines which ensure that a flow's characteristics remain unchanged as it passes through the network. Hence end-to-end delay analysis becomes simple and possible. However, the need to install shaping mechanisms per-flow at each router in the core is again not a scalable solution. One of the first attempts to make the core stateless was the Core-Stateless Fair Queuing (CSFQ) due to Stoica and Zhang [115] which aimed to provide approximate end-to-end fairness properties without maintaining per-flow state in the core. Their later work called the Scalable Core (SCORE [116]) demonstrated per-flow guaranteed services without any state in the core. They introduced the idea of inserting rate information in the packets called Dynamic Packet State. A variant of this idea which provides scalable delay guarantees was VTRS [136, 137]. Another means to retain flow characteristics unchanged was proposed by Cobb [22, 21] which involved deploying the so-called fair aggregators at each node to take care that packets don't lose their original spacing.

Instead of attempting to retain a flow's character by inserting information into packets, Reisslein et al [101] propose a bufferless network architecture and demonstrate end-to-end delay guarantees. Other proposals for specialized networks also exist (e.g. GPS [92], EDF [107, 108], SCED [104], Co-ordinated Multihop scheduling [80], proportional Diffserv [29]). Some of the new techniques in providing delay guarantees include Competitive analysis to achieve jitter control [82] and measurement analytic approach [35].

A related thread is that of providing QoS guarantees on not just a point-to-point basis, but a point-to-anywhere basis as envisioned by Clark et al [19].

LIRA [114], Point-to-Set [96] and the Hose Model [31] are examples of such architectures. While the Hose Model provides a solution for the point-to-multipoint QoS problem, it relies on signaling-based reservation and per-hop admission control.

In almost all of the above proposals, the need for admission control exists and its treatment can often be decoupled. In the following section we examine some approaches to admission control.

2.6 Admission Control

The admission control problem involves deciding on whether to admit a new flow into the network. There have been a variety of approaches which can mostly be classified as being deterministic (e.g., Measurement-based admission control for Intserv by Jamin et al [58], Schedulability conditions due to Liebeherr et al [81]) or statistical (e.g., Effective Bandwidth approach [78], Kalman filters [32], Bayesian approach [46]). Knightly and Shroff [68] present a detailed survey and performance study of of admission control algorithms.

Typically admission control is envisioned to be a central entity with control over what enters the network. But recently, there have been proposals for distributed admission control [62]. Simulation studies also indicate that distributed endpoint based admission control might be a possibility [11].

But the efficiency of an admission control regime is important from the point of view of network utilization. There have been attempts at making admission control accurate and robust. Firoiu et al [41] provide a new deterministic approach to admission control known as discretized admission control to reduce the complexity of schedulability conditions. On the statistical front, Grossglauser and Tse [54] examine the effects of estimation errors in MBAC and propose means to handle them. Qiu and Knightly [95] tackle the error due to Central Limit Theorem approximations and demonstrate efficient maximal rate envelopes obtained from measurements.

To sum up, a lot of recent work has focused on making admission control distributed and efficient. In this thesis we leverage some of the past insight and develop a new edge-based statistical admission control framework and demonstrate its ability to deliver multiplexing gains and enhanced utilization.

2.7 Positioning our work

Having reviewed recent literature, we now relate our work to existing techniques. The problem of enhanced spatial granularity in QoS has received attention only recently. We provide the first edge-based architecture toward realizing point-to-multipoint QoS. In order to provide a priori assurances with such models, we build a new framework to decouple computation of delay assurances from exact traffic profile. In conjunction, these contributions allow us to build a QoS architecture with enhanced spatial granularity featuring a priori service assurances.

In addition to being the first edge-based proposal [96, 98] for the point-to-multipoint QoS problem, we demonstrate the importance of traffic matrix information [99] and evolve new directions for estimation that are tailored for VPNs.

CHAPTER 3
Increasing Spatial Granularity of QoS: A Point-to-Set architecture

3.1 Summary

As a first step toward building an edge-based service with a priori assurances and enhanced spatial granularity, we begin by considering a subset of the problem. We extend the traditional point-to-point allocation model to a point-to-set model where the user has considerable flexibility in apportioning the allocated bandwidth among a finite set of destinations. Following is the organization of this chapter.

- Review of related work and motivation
- An ideal point-to-set service and its implementation
- Evaluating the ideal service

3.2 Introduction

The best-effort traffic in Internet is inherently of the *point-to-anywhere* nature, i.e., sources direct packets to any possible destination. In contrast, traditional quality-of-service (QoS) models set up premium services on a *point-to-point* basis (eg: virtual leased lines, frame-relay, ATM services, int-serv [9] etc). Recently, with the advent of IP differentiated services [6, 19, 114] there has been interest in expanding the *spatial granularity* of QoS models, i.e., providing assurances toward a set of destinations. Clark and Fang [19] proposed that a pool of "assured" service tokens could be allocated to a user or site with the flexibility to mark packets sent to any arbitrary destination with such tokens. While such a *"point-to-anywhere"* assured service model is very appealing to users, the large spatial granularity of the service makes efficient admission control and provisioning virtually impossible [114]. We consider the subset of this problem by examining assurances to a fixed set of destinations.

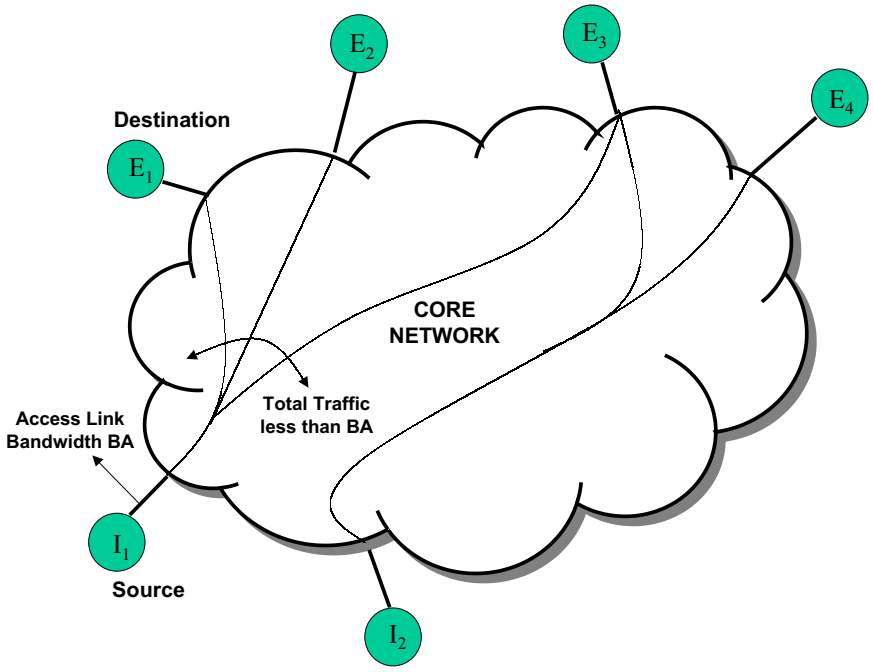


Figure 3.1: The Point-to-Set Concept: The total offered traffic due to I_1 is limited by its access link bandwidth. Provider gains most if the reserved bandwidth is less than or equal to this quantity.

3.2.1 The Point-to-Set concept

Consider a private network of sites I_1, I_2, E_1, E_2, E_3 and E_4 as shown in Fig. 3.1. The aggregate traffic from I_1 (called the *point*) toward E_1, E_2 or E_3 (called the *set*) is bounded by the capacity of the access link (say, "peak"). Given the point-to-point allocation model, site I_1 would require a link with capacity equal to "peak", to each destination in the *set* for an assured service toward the sites in the set. As such, the total purchased capacity from the provider (which is three times the peak here) exceeds the access link capacity leading to wastage of resources. We propose a *point-to-set service* wherein a customer buys a bandwidth *less than or equal* to his peak requirement (or a given total bandwidth), but is assured that his traffic needs to any destination in the set are met with high *probability*. In other words, the user buys bandwidth to a set of destinations, instead of purchasing point-to-point links to the destinations and retains the freedom of deciding the fraction of bandwidth allocated to a specific destination. Thus there is a cost saving in that the

point-to-point links need not be leased from I_1 to each member of the set. For the provider, the paths connecting edge I_1 to the set $\{E_1, E_2, E_3\}$ can be multiplexed with other contracts by exploiting statistical properties of the traffic.

3.2.2 An Ideal Point-to-Set Service

Before trying to build an architecture to realize the point-to-set concept, it is useful to consider the ideal implementation. A user would want to be assured a bandwidth equal to the peak requirement toward *any* destination. A more restricted version of the ideal case is where the set of destinations is finite and the user still has the assured bandwidth toward any node in this finite set.

Consider how a provider would implement this ideal service. An efficient provisioning strategy would reserve a network-wide total bandwidth less than or equal to the peak requirement of the customer. However, the user can offer traffic at this peak rate toward any destination. Since the user does not specify the exact load toward a given destination, the provider needs to accurately predict demand to avoid over-provisioning. Notice how this is different from a point-to-point model, where peak bandwidth would be reserved toward each destination.

We investigate such an ideal strategy as part of the Deterministic Point-to-Set architecture (D-P2Set §3.5). The name indicates the deterministic admission control strategy. In practice, the dynamic tracking and provisioning schemes featured in D-P2Set are hard to implement due to complex and time-varying statistical characteristics of Internet traffic. The intuitive appeal of a point-to-set service is in the fact that it has the potential to provide inexpensive and *flexible* services to the customer while allowing statistical multiplexing gains to the provider. The important questions to be answered then are:

- Are there quantifiable benefits that make deployment of point-to-set services an attractive option?
- What are the simplifications to the ideal service that will render the architecture practical and realizable?
- Is it possible to build such a service with a minimal edge-based approach?

We build such a model in the Statistical Point-to-Set service (S-P2Set §4) with some simplifications to the ideal model so that it can be implemented at the edge of the network with just simple shaping components.

3.3 Overview and Related Work

We give a brief description of the D-P2Set and S-P2Set architectures and compare them with existing models.

3.3.1 Two approaches for a Point-to-Set Service

Both the D-P2Set and S-P2Set models are completely edge-based solutions. The network is viewed as a collection of paths and each path is associated with a "path capacity". Admission control is per-path and based on the available path capacity. The D-P2Set model features:

- Per-path policers for every contract at the edge of the network to enforce the provisioned path capacity.
- A central admission control entity based on the per-path assured rate requirement specified by the contract.
- Dynamic bandwidth estimation to adopt the reserved per-path bandwidth.
- No signaling

In contrast, the S-P2Set model eliminates bandwidth estimation and the need for the customer to specify a per-path assured rate. The S-P2Set model features:

- Contracts which specify per-destination mean and variance of traffic fraction. This is much more flexible than a fixed per-path assured rate.
- Per-path dual-leaky-bucket shapers to enforce the contracted per-path traffic fraction.
- A probabilistic admission control computed at the edge.
- No signaling or bandwidth estimation.

Thus the S-P2Set model is simpler, more flexible and exploits statistical multiplexing gains by employing probabilistic admission control.

3.3.2 Comparing with previous work

Clark et al [19] introduced the idea of going beyond point-to-point services and providing flexibility to users, while at the same time allowing multiplexing gains for the provider.

LIRA [114] considers the problem of large spatial granularity, where QoS assurances are for a large set of destinations (possibly unlimited). By employing enhancements to routing protocols the authors provide a way to achieve per-packet admission control so that a user can employ the allocated bandwidth toward any destination. Consequently, LIRA faces scalability issues when there are large number of destinations. In the present paper, we do not require any changes to the core network or the routing protocols. Further, we consider admission control on aggregates and assurances toward a finite set of destinations. Hence our proposal is not affected by most of the scalability issues mentioned above.

The closest to our proposal is the *Hose Model*. Duffield et al [31, 30] propose a framework for Virtual Private Network (VPN) resource management and introduce the idea of a "hose" as a resizeable access link for a VPN node. They attempt to solve a part of the problem tackled here, namely, that of going beyond the point-to-point allocation model and do not treat the problem of admission control. The hose is intended to provide bandwidth toward the set of destinations and is implemented by the provider using a reservation tree structure [71, 56]. They suggest resizing the hose using online prediction of traffic characteristics to obtain further multiplexing gains. The advantages of the hose model are manifold:

- The customer maintains a single logical interface with the provider and does not need to provision point-to-point links for each destination.
- Multiplexing gains are obtained due to aggregation at the hose-level.
- The customer does not have to specify a traffic matrix indicating per-destination bandwidth requirement.

But the the Hose Model needs reservation to be set up using signaling and per-link admission control. Further it requires solving a complex optimization problem to find a reservation tree structure for the purposes of provisioning. Since the hose is resizeable, every change in allocated bandwidth is accompanied by per-link admission control. The resizing itself depends on a Gaussian bandwidth demand predictor. Thus the performance of the model is dependent on traffic statistics being amenable to the predictor assumptions.

The Point-to-Set model moves away from the signaling paradigm and features an edge-based solution. The D-P2Set model features a bandwidth demand estimation module like the hose model (specifically, the "Resized Provider Pipes" implementation), but does not require a mesh of provider-pipes between every ingress and egress pair. D-P2Set simulation results indicate that the timescale of provisioning changes can become a crucial performance parameter.

To solve this problem, we eliminate the need for demand estimation and bandwidth reservation. Instead, we adopt a probabilistic admission control regime to exploit statistical multiplexing gain. We call this new model the Statistical Point-to-Set architecture (S-P2Set). Other than the absence of reservation and demand estimation, a key difference from the Hose Model is in the traffic matrix assumption. The Hose Model gets rid of the traffic matrix specification completely. The S-P2Set model strikes a compromise by requiring only the mean and variance of the fraction of total traffic directed toward a destination. This is much more flexible and easily characterized than the per-destination rate specification used in D-P2Set. We demonstrate in Chapter 8 that such traffic matrices can be learnt using existing SNMP-based measurement information.

The highlights of this comparison are captured in Table 3.1.

In order to build a probabilistic admission control mechanism for the point-to-set architecture, we adopt a dual-leaky-bucket regulated source characterization (similar to [107]) and relate the parameters to statistical characteristics. Unlike existing work, we obtain bounds on per-path traffic statistics at the edge of the network exploiting the point-to-set model. We then employ the per-path information to evaluate the admission control criterion.

Attribute	Customer Pipe	Hose	D-P2Set	S-P2Set
Impl.	Point-to-Point links for each src-dest pair	A single *hose* from customer; network-wide reservation	Fully Edge-based	Fully Edge-based
B/w Resvn	Static - Whole link is reserved	Dynamic dep. on demand	Dynamic dep. on demand	No reservation
Signlng	None	Required to update reservations	Not required	Not Required
Traff Matrix	Need info about every src-dest pair	Not reqd.	Need peak rate for each source dest pair	Need mean variance of traff. fracn
Traff Stats	Provision for peak traffic	Gaussian predictor to track demand	Adaptive EWMA predictor	Does not need online traff. stats.
Algo. Cplxty	None	Complex provisioning algorithm	Simple edge-based algorithm	No b/w reservation
Adm Ctrl	Deterministic, one-time	Need per-link per-hose computation for every change in reservation	Edge-based, deterministic, one-time computation	Edge-based, statistical, one-time computation
Muxing gains	None	Statistical gains due to hose-level aggregation	Limited gains due to aggregation	High gains due to statistical admission ctl

Table 3.1: Comparison of models for handling QoS with increased spatial granularity.

We begin (§3.4) by examining a key concept in the development of both flavors of Point-to-Set - that of a path. We then describe the D-P2Set architecture in §3.5. We treat the statistical approach in detail in Chapter 4.

3.4 Notion of a Path

In the following sections we refer to a path as a sequence of multiplexers defining the route from an ingress node to an egress node. Additionally, we associate

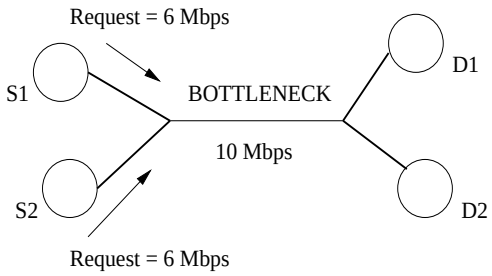


Figure 3.2: With Distributed admission control, two network entry points have to take care not to over-book a shared link. In the example above the requests from $S1$ and $S2$ are admissible individually, but not together

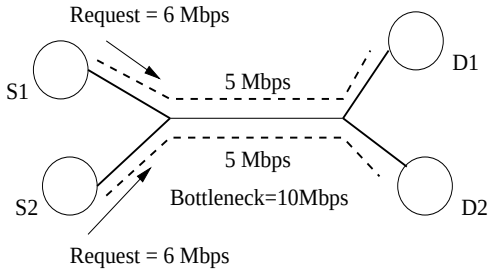


Figure 3.3: The dashed line denotes a source-destination path. By statically apportioning capacity among paths, we can avoid the problem of over-booking. Each network edge sees $5Mbps$ as the capacity available

a capacity with each path. The objective of this path abstraction (as explained below) is to break down the network of shared links into a set of disjoint virtual links.

Consider the network illustrated in Figure 3.4. Each ingress I_k is connected to an egress E_k by a path. Some of these paths are indicated by dashed lines. The capacity that is available to the traffic that originates at an ingress (say, I_1) and is destined to an egress (say, E_1) is not readily apparent. Thus if we are to admit an aggregate toward egress E_1 we would like to have some mechanism to measure the available capacity.

One option would be to devise an online measurement technique which would inform the ingress about the current available capacity. However such information would be dynamic and not suitable for admission control. Thus we aim to somehow *divide* the total network capacity among the different ingress-egress pairs, so that

the total capacity available for aggregates between (I_k, E_j) is known beforehand.

We could have a signaling protocol reserve bandwidth for a given path (ingress-egress pair). Instead, in the present analysis we present a simpler mechanism which does not reserve any bandwidth but still achieves the equivalent of provisioned paths.

First, we examine each link traversed by a path and associate with it a share of the bandwidth as specified in Algorithm 1 (described later). Second, we introduce a centralized admission control component which uses path capacity as computed by this algorithm to admit contracts. Since all traffic admitted into the network is assumed to have been approved by the admission control module, the network is provisioned as if the paths have capacities computed by Algorithm 1.

The admission control algorithm admits contracts on a per-path basis by considering the path capacity. Thus the bandwidth on each link in the network is automatically shared among the paths that traverse the link without having to do any kind of reservation. We now formally state the definition of a path as discussed

Algorithm 1 Computing Path Capacity

```
Input: Paths P_i, Path Capacity Γ_i
Input: Capacity of multiplexer i, C_i
for Each path P_i do
  Γ_i ← ∞
  for Each multiplexer M_j ∈ P_i do
    P ← Set of Paths incident at M_j
    N ← |P|
    if Γ_i > C_j/N then
      Γ_i ← C_j/N
    end if
  end for
end for
```

in the preceding paragraphs.

Definition 3.4.1. A path is defined as a sequence of multiplexers $\{M_j\}$ such that M_j is connected to M_{j+1} and for any $i \neq j$, $M_j \neq M_i$. That is the paths considered for analysis are loop-free.

The computation of path capacity as described in Algorithm 1 and the meaning of path utilization target is captured in the following definitions.

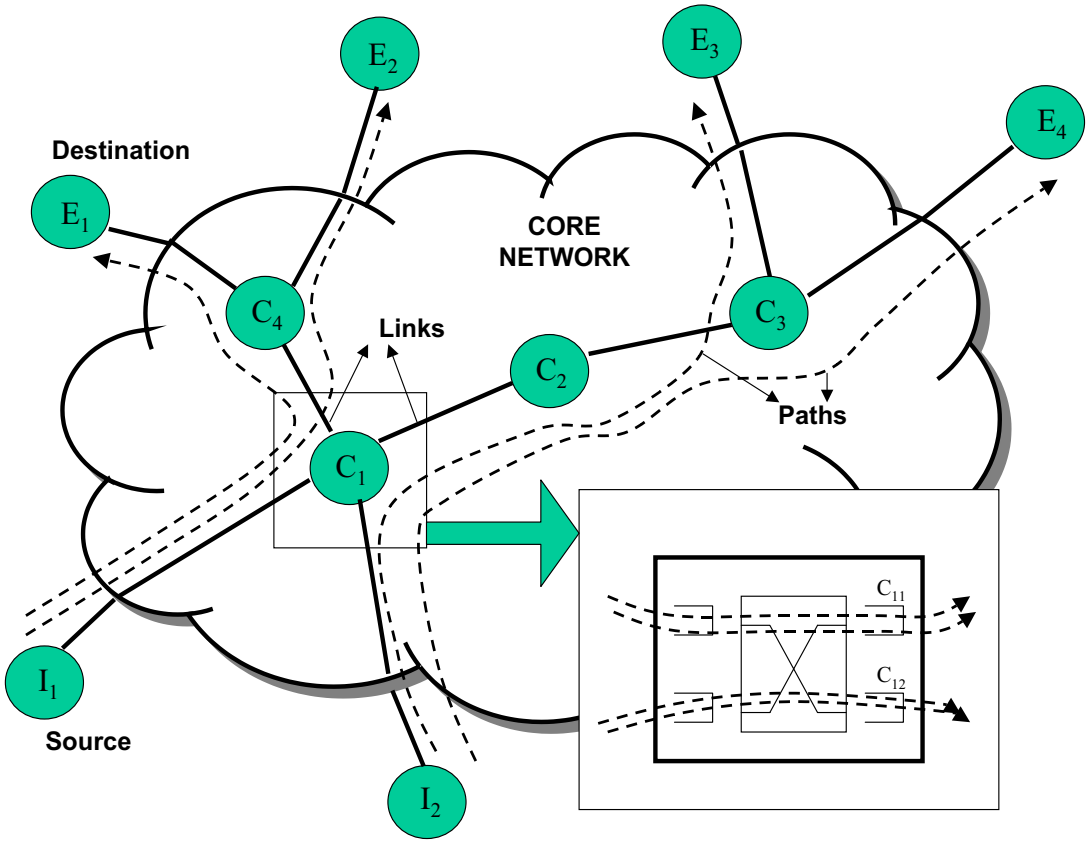


Figure 3.4: A Path is defined as a sequence of multiplexers. The figure shows paths connecting ingresses (I_k) to egresses (E_k). In the inset there is an illustration of the multiplexers in the core node C_1. There are two output multiplexers labeled as C_{11} and C_{12} each shared by two paths

Definition 3.4.2. *Consider a path P_i defined by the sequence of multiplexers $\{M_j\}$. Let C_j be the capacity of M_j. Let n_j be the number of paths passing through M_j. Then, the capacity of the path P_i, is defined as:* $\Gamma_i = \min_{j \in P_i} \frac{C_j}{n_j}$.

Definition 3.4.3. *The utilization target for the path, u_i is such that for the flows (σ_i, ρ_i) admitted on the path P_i,* $\frac{\sum_i \rho_i}{\Gamma_i} \leq u_i$.

This definition of path capacity has some notable drawbacks:

- The capacity of a path is decided by the bandwidth of the most traversed link and the bottleneck link.

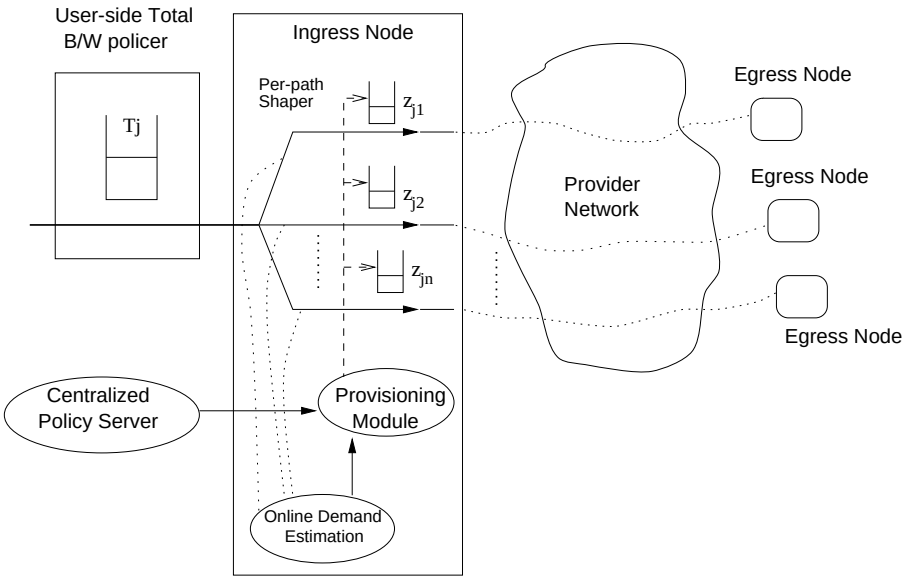


Figure 3.5: The D-P2Set Architectural Model with a Single Customer j

- An equal division of resources among all paths is not necessarily the best, since some paths might have to carry more traffic.

We shall present a definition that remedies these problems in a future section. For the present, we use this definition.

3.5 D-P2Set: A Deterministic Approach

In the following sections we describe an implementation of the near-ideal edge-based point-to-set architecture. It features a deterministic admission control regime and hence the name.

The D-P2Set model (Figure 3.5) allows a site j to claim a pool of premium tokens T_j per unit time. The model lets the site j to flexibly use these tokens for the traffic toward it's set, subject to the limit that on each path ji the *peak rate* is less than or equal to p_{ji}. Observe that the user has the flexibility to use up to the peak rate p_{ji} on any single path, but pays the provider only based upon the aggregate pool of premium tokens T_j per unit time. This is more economical for the user if $T_j < \sum_i p_{ji}$ because unlike point-to-point leased lines or frame-relay CIR[1] it need not pay for unused committed rates. The essential data plane component

[1]CIR = Committed Information Rate parameter of frame-relay service

is a traffic conditioner which now consists of a set of token buckets: one for the overall pool of tokens with rate T_j and one for each path (Figure 3.5). From the provider perspective, however, this edge-based scheme does not yield multiplexing gains compared to peak-rate provisioning *unless a dynamic measurement and re-provisioning strategy is used*. That is, in the absence of dynamic re-provisioning, the provider will have to have per-path shapers (at the ingress edge block in Figure 3.5) operating at the peak rates p_{ji}.

To achieve multiplexing gains, we develop the following components:

1. **An online demand estimation** module which monitors the varying per-path traffic demands. It accounts for potential long-range dependence in traffic by exploiting the concept of *correlation horizons* [53]. In particular, the module detects the current correlation horizon dynamically and uses the mean and deviation of demand estimated on that horizon (Section 3.5.2).

2. **A dynamic provisioning** module which uses these per-path demand estimates, d_{ji} (d_{ji} is the estimate for site's traffic from j to destination i in the set). This module operates in two phases: a) per-path provisioning phase and b) per-contract provisioning phase. The two phases can be explained as follows. On a given path, there could be many contending contracts. The sum of the demand estimates of all contracts on that path may exceed the path capacity, i.e., $\sum_j d_{ji} > C_{ji}$, where C_{ji} is the capacity of the path ji. The "Per-Path" provisioning ensures that this constraint is taken care of. Assume that, given the above constraint, each contract was allocated y_{ji} on the path ji. The sum of the allocations made at each path, for a particular contract, may exceed the contract's total bandwidth T_j, i.e., $\sum_i y_{ji} > T_j$. "Per-Contract" provisioning is performed to ensure that the final allocations conform to the contracted bandwidth. The resultant per-path rate limits z_{ji} are enforced at the provider-side shapers for each path ji (note that $\sum_i z_{ji} \leq T_j$).

3. **An admission control** module that ensures that the sum of assured rates along a path is less than the path capacity.

The choice of provisioning time-scales is important since it decides the respon-

siveness of the provisioning algorithm to traffic changes. In §3.6 we examine the performance of the model for various values of the provisioning timescale.

3.5.1 Source Traffic Specification

The D-P2Set service is offered by means of contracts. The contract for user j consists of:

- The finite, known set of destinations S_j with cardinality N_j.
- T_j, the aggregate traffic to the set, S_j, i.e. if t_{ji} is the traffic generated toward destination $i \in S_j$, then $\sum_i t_{ji} \leq T_j$.
- The per-path peak rate, p_{ji} on path ji.
- The per-path minimum assured rate, a_{ji} on path ji

The per-path peak rates are assumed to have been determined in a characterization phase before the contract is arrived up on. The minimum per-path assured rate ensures that the user can get a minimum provisioning of a_{ji} on path ji at any time. As a simplification, we shall assume the same value a_{ji} for all paths ji, calculated as T_j/N_j (henceforth, referred to as a_j). This constraint is enforced by admission control. Beyond that, the provisioning may depend upon the demand estimation and multiplexing characteristics of the path. Errors in provisioning or admission control in this regime manifest as delays (on which there is no assurance) and packet losses. One can imagine a regime without such a minimum service expectation, which could lead to higher multiplexing gains, but also lead to higher multiplexing costs due to provisioning/admission control errors.

3.5.2 Online Demand Estimation

Experimental evidence [79] suggests that network traffic exhibits properties of self-similarity and long range dependence (LRD). In order to track demand patterns it is essential to address issues due to long-range dependence. The immediate question to be answered is that of the amount of correlation information and the appropriate timescale to be considered for demand estimation. One would expect that

Algorithm 2 Adaptive calculation of *EWMA* weight parameter

$mindeviation = MAXNUM$;
$index = 0$;
for $weight = 0.05$ to 0.99 **do**
{Update the mean deviation using latest sample in arrivals and meanarrivals as previous prediction}
$meandev[index] = meandev[index] + abs(meanarrivals[index] - arrivals)$;
{To find the weight with minimum deviation}
if $meandev[index] < mindeviation$ **then**
$mindeviation = meandev[index]$;
$optimalwt = index$;
end if
{Update the moving average for arrivals}
$meanarrivals[index] * = (1 - weight)$;
$meanarrivals[index] + = weight * arrivals$;
{For calculation of deviation}
$sqrmeanarrivals[index] * = (1 - weight)$;
$sqrmeanarrivals[index] + = \ weight * arrivals * arrivals$
$index + +$;
$weight + = 0.05$;
end for
{Using *optimalwt* calculate prediction}
$std\_deviation = sqrt(sqrmeanarrivals[optimalwt] - (meanarrivals[optimalwt])^2)$;
$prediction = meanarrivals[optimalwt] + 2 * std\_deviation$

traditional finite memory models would not be sufficient in modeling such traffic. However, recent work [53] analyzing finite buffer queues with LRD input indicates that only a finite amount of correlation need be considered for certain queuing performance measures (e.g., loss rate). The amount of correlation that needs to be taken into account depends not only on the correlation structure of the source traffic, but also on time scales specific to the system under study. However, the paper establishes the existence of a correlation horizon at any point in time. In particular, it is shown that the impact of correlation in the arrival process on loss rates becomes negligible beyond a time-scale, referred to as the *correlation horizon*. This implies that we may choose either self-similar models or models with finite memory as long as it captures correlation up to the correlation horizon for the system.

We choose a simple Exponentially weighted moving average (EWMA) pre-

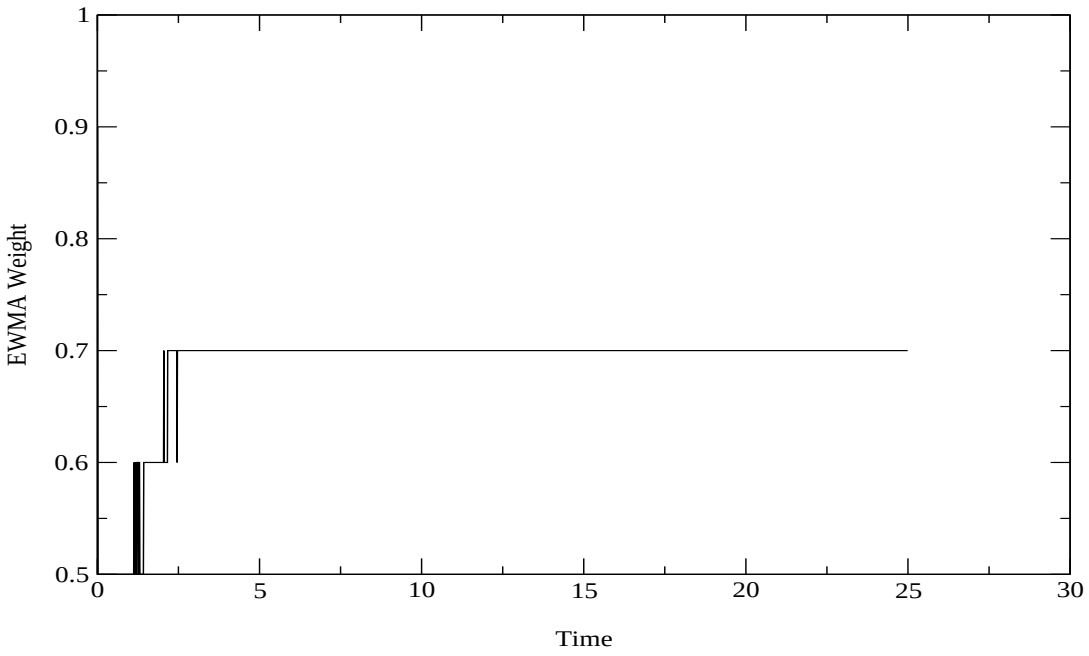


Figure 3.6: EWMA weight α vs Time

dictor. The weight parameter of the EWMA predictor is changed dynamically to suit the traffic characteristics depending upon the current correlation horizon. The demand estimation scheme hence maintains a *vector of EWMA predictors* for different weight parameters ($\alpha \in [0.05, 0.95]$). At the end of a measurement window, the weight parameter *corresponding to the minimum deviation from the observed sample for the previous interval* is chosen. The minimum deviation here implies that the corresponding weight parameter reflects underlying correlation horizon of the traffic. The details of the algorithm can be found in the pseudo code. The predictor then provides the average (μ) and variance (σ^2) of the tracked process. The estimate, e, is calculated as

$$e = \mu + 2\sigma$$

Figure 3.6 shows a typical graph for the variation of the weight parameter with time, for a specific simulation run featuring traffic generated by superposition of truncated Pareto sources [97]. The weight parameter initially varies, but later on stabilizes around a single value. The estimates produced by the predictor are plotted against the actual samples, in Figure 3.7. The estimates can be seen to be effectively tracking the samples. The final estimate of demand is referred to as d_{ji} in subsequent

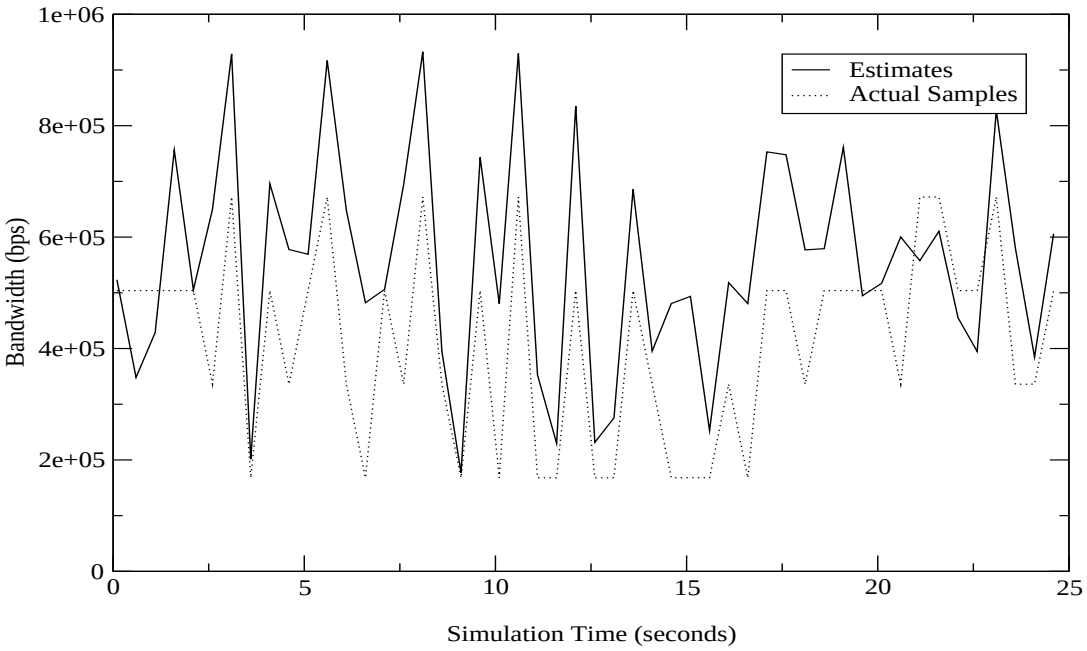


Figure 3.7: Online Estimates of Traffic vs Time

sections. Further details regarding the demand estimation module may be obtained from [97].

3.5.3 Per-Path Dynamic Provisioning

As mentioned earlier the dynamic provisioning decision is made in two phases, and uses the demand estimate d_{ji} to finally yield the rate limits for the shapers z_{ji}. The per-path decision is made with the knowledge of demand estimates for competing contracts on that path. The decision is then used in the second phase (per-contract provisioning) to calculate shaper settings. Let there be N contracts on the path. Then the per-path provisioning proceeds as follows:

- If the sum of per-path provisioning estimates (over all contracts) on a path ($\sum_j d_{ji}$) is less than or equal to the path capacity C, then allocate $y_{ji} = d_{ji}$ to each shaper. Otherwise:

- For all the contracts where $d_{ji} \leq a_j$ allocate d_{ji}, i.e., $y_{ji} = d_{ji}$. Set

$$C' = C - \sum_{\{j:d_{ji} \leq a_j\}} d_{ji} \tag{3.1}$$

Algorithm 3 Per-Path Provisioning Decision

For all $d_{ji} \leq a_j$ allocate d_{ji}. Delete these contract from the List. Calculate available resource A, as the difference of allocations made and the path Capacity C.
For the rest of the contract allocate a_j, where a_j is the minimum assured rate for the Contract j. Define $x_{ji} = a_j$ and re-calculate A.
while $A > 0$ **do**
  Define *diff* = Minimum$(d_{ji} - x_{ji})$ over all j, M as the number of contracts for which provisioning decision has to be made.
  if *diff* $\geq A/M$ **then**
    Define *increment* $= A/M$
    Allocate resources equal to *increment* to each contract.
    $A = 0$
  else
    increment = *diff*
    Allocate resources equal to *increment* to each contract.
    Update x_{ji} as x_{ji} += *increment*
    A -= *diff* * M
    Delete the contracts for which $(d_{ji} - x_{ji})$=0 from the List (of contracts for which provisioning has to be made).
    Update M.
  end if
end while

for every allocation.

- For the remaining contracts use a max-min strategy to arrive at the allocations y_{ji} subject to the following constraints:

$$\sum_{\{j:d_{ji}>a_j\}} y_{ji} = C', \tag{3.2}$$

$$y_{ji} \leq d_{ji}, \forall\, j \tag{3.3}$$

$$y_{ji} \geq a_j, \forall\, j : d_{ji} > a_j \tag{3.4}$$

The solution to the above problem is suggested in Algorithm 2.

In the case where, $d_{ji} \leq a_j$, another allocation strategy could have been always provisioning a_j. Though this assures a contract of a minimum guaranteed bandwidth, it will be at the cost of greater losses for some other contract, which makes excursions to peak. Assuming that a contract will make excursions above and below the

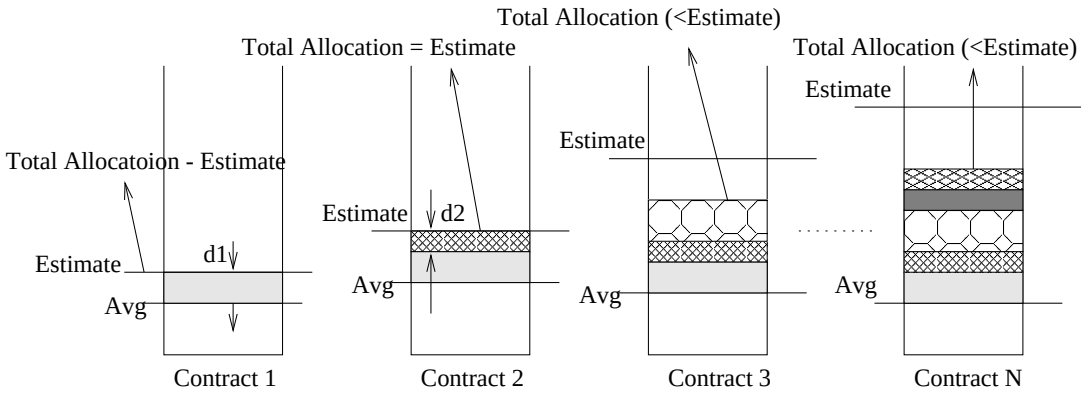


Figure 3.8: Per-Path Re-provisioning Policy

contracted minimum assured rate (a_j), by allocating bandwidth only equal to the estimate (even though it is below a_j) we distribute losses uniformly over time.

Figure 3.8 describes the per-path provisioning policy. We first allocate resources to all the contracts whose estimate is less than the minimum assured rate and eliminate them from the provisioning list. Figure 3.8 shows all the remaining contracts. Here we first allocate the corresponding assured rate *avg* to each contract. Then, the minimum difference between the estimate and the difference, $d1$, is calculated and is allocated to each contract, and the available bandwidth on the path is updated. With this allocation, we find that the first contract's demand has been met, hence it is removed from the list. Again, the new minimum difference, $d2$, between the estimate and the allocated (till now) is calculated and dispersed amongst all the remaining contracts, eliminating yet another contract from the list. This process is repeated recursively until, we run out of the available bandwidth on the path.

3.5.3.1 Per-Contract Re-provisioning

At the end of the Per-Path Re-provisioning, the allocations y_{ji} are obtained. This means that the per-path allocation is y_{ji} to contract j on path ji. However, the total suggested allocations for the contract may exceed his total contracted bandwidth T_j. So another level of re-provisioning amongst different paths in a contract is needed. This re-provisioning can be again expressed as an optimization problem. Let z_{ji} be the final allocation made (by agent) to Contract j on path ji

and N_j be the number of destinations in the Contract j's set. Then, for the case where $\sum_{i=1}^{N_j} y_{ji} > T_j$, a max-min strategy can be used to arrive at z_{ji} subject to the following constraints:

$$\sum_{i=1}^{N_j} z_{ji} \;=\; T_j, \tag{3.5}$$

$$z_{ji} \;\leq\; y_{ji},\, \forall\, i \tag{3.6}$$

$$z_{ji} \;\geq\; min(y_{ji}, a_j), \forall\, i \tag{3.7}$$

The solution to the above problem can again be computed using the Algorithm 3 albeit taking care of the new variables and the constraints.

3.5.4 Admission Control

A centralized module makes all the admission control decisions. This module is assumed to have a map of the network including link bandwidths (available through OSPF extensions [55]) and information regarding the current set of contracts. On the arrival of a new customer, the following steps are undertaken by the policy server. First, the contract is mapped to a set of paths leading from the source edge to the destination edges. This is done using the information from the underlying routing infrastructure. Second, the decision whether to admit this contract is taken. This is done by computing the available bandwidth on every path ji and ensuring that it is greater than the contract's minimum assured rate a_{ji}, for each path ji. The available bandwidth is the path capacity reduced by the minimum assured rates a_{ji} of all existing contracts (i.e., $C_i - \sum_j a_{ji}$).

3.6 Results and Discussion

In this section we present the simulation results. We evaluate the point-to-set model with a single ingress topology and a multiple ingress topology. With the single ingress topology, the performance of the provisioning algorithm is first examined and then, the customer and provider gains are presented. For the multiple ingress topology we evaluate the customer gains. Finally, we examine the effect of the provisioning timescale on the performance of the model.

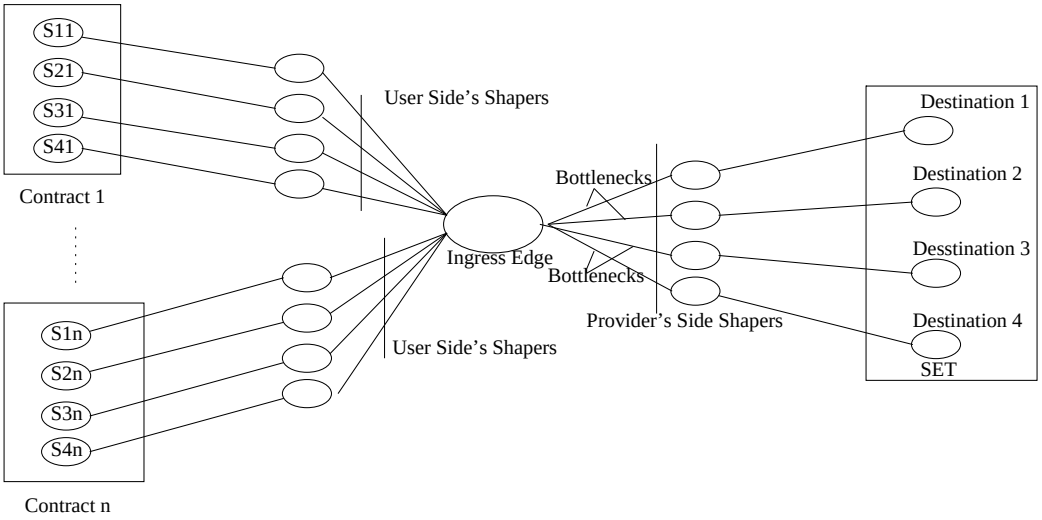


Figure 3.9: Single Ingress Topology

3.6.1 Single Ingress Topology

The topology shown in Figure 3.9 (single ingress) was simulated on NS [118]. All the contracts have an equal total contracted bandwidth, T, of 2 Mbps, a peak rate of 0.75 Mbps and have a common set, of 4 destinations. The constrained paths are of 25 Mbps and the one way end-to-end propagation delay is 30ms. All the paths other than the constrained paths are of 10 Mbps. The source traffic is shaped by static (customer side) shapers to ensure conformity with the contract, in terms of the peak rates. The provider network features the dynamically provisioned shapers which adapt to the user's traffic distribution with respect to his destinations.

The point-to-set model was tested in the following settings:

- **Traffic simulated by agents provided by NS:** Constant Bit Rate traffic sources (CBR) with rates varying randomly with time were employed. A *uniform* random variable was sampled to obtain the rates for the CBR agents every 0.15 seconds. The traffic was generated such that it is distributed uniformly across destinations. Thus it is equally likely that a particular rate is observed toward a destination. Note that the dynamically provisioned shapers adapt to the traffic profile of the customer and are not in anyway aware of this arrangement. So even if a traffic profile features sources biased to certain elements of the destination set, the shapers will adapt appropriately.

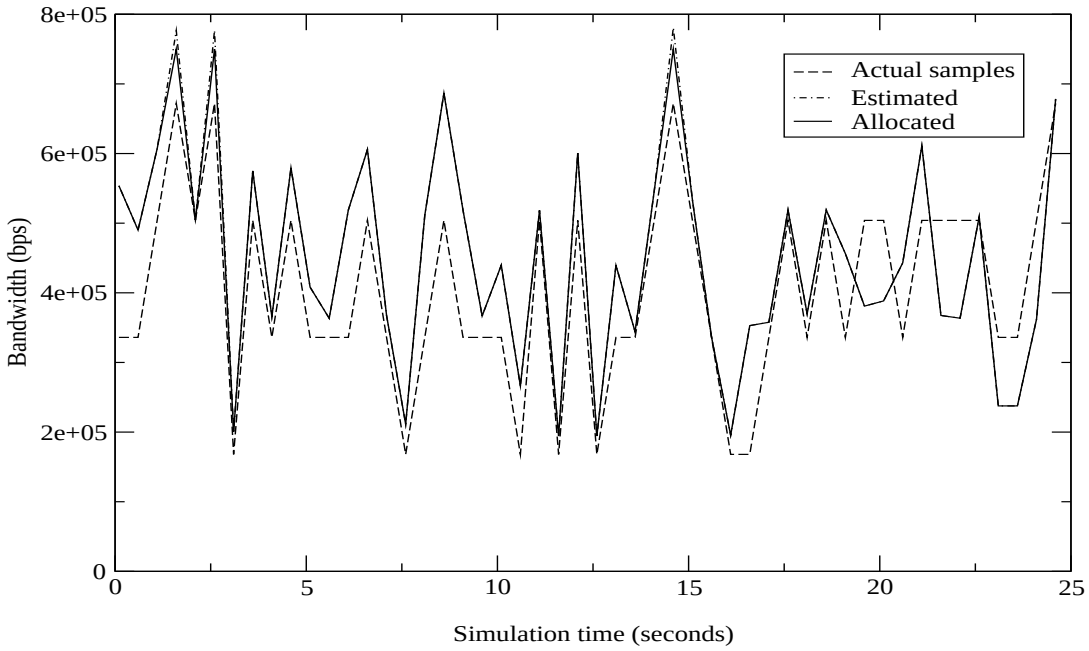


Figure 3.10: Allocated, Estimated bandwidth and Actual Traffic Samples vs Time

- **Trace driven simulation:** MPEG encoding of the Star Wars movie, converted into the NS trace format, is available at [10]. The "Application/Traffic/Trace" agent in NS was employed to generate video traffic from this trace. The source was attached to the NS UDP agent. Every source picks a random start point in the trace file [118].

3.6.1.1 Performance of the provisioning scheme

In Section 3.5.3 we presented a strategy to handle per path bandwidth provisioning. The performance of this strategy decides the costs to the customer. Figure 3.10 shows the plots for the actual allocation made for a contract along with the traffic samples and estimates versus time. The Allocation curve is almost coincident with the Estimate curve. This is indicative of the fact that most of the time the contract was allocated what it demanded. Since the customer's traffic is loosely policed, his peak rate at some time instants exceed the negotiated peak, i.e. 0.75MBps and in such cases the allocation is clipped to the contract's peak. In effect the plot shows that the contract requirements are met.

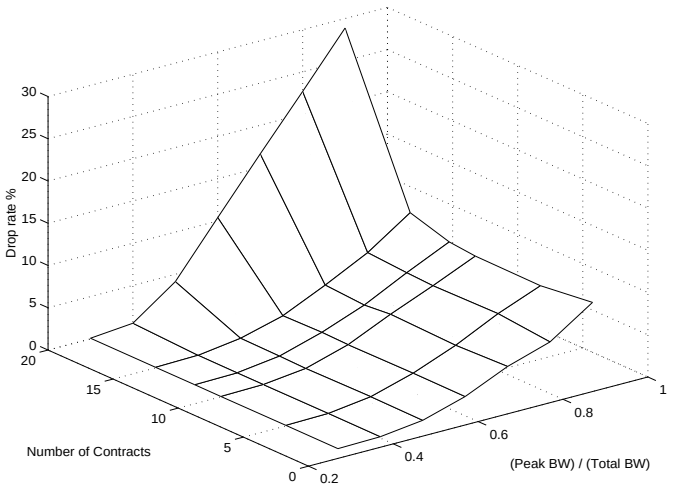


Figure 3.11: Drop Rates (%) against number of contracts and Peak/Total ratio

3.6.1.2 Customer gain with simulated traffic

To examine the customer gain, we employ the ratio of the per-path peak, p_{ji}, to the total contracted bandwidth for the set of destinations, T_j. Observe that the customer is paying for T_j, that is T_j/N toward each destination in the set, while being allowed to send p_{ji} toward destination j. Thus, higher the ratio p_{ji}/T_j, higher the gain for the customer. However, it is important to note that a higher value of this ratio implies *greater strain* on the provisioning algorithm and a higher trade-off in terms of drops. This is because, the provider has to maintain a "zero sum game" in terms of the bandwidth allocated to this contract toward the destinations in its set. Also, the tracking algorithm has to deal with higher variability of offered load toward a destination.

In the simulations, p_{ji} values were assumed to be equal (say, P_j) Figure 3.11 and Table 3.2 depict the drop rate with increasing number of contracts for various values of P_j/T_j, for simulated traffic. For a given number of contracts the drop rate increases with the ratio P_j/T_j. However, we observe from the graph (and from Table 3.2) that the drop rates are within 0.8% even for a peak to total ratio of 0.5. This implies that with a low drop rate, the customer can offer a peak rate of $N\Sigma_i p_{ji}$, (N=4 in the simulations) while paying for $2P_j$ (assuming the same peak toward all destinations, $P_j = p_{ji}$ for all i).

Num Contracts	$p_j/T_j = 0.5$	$p_j/T_j = 0.6$	$p_j/T_j = 0.7$	$p_j/T_j = 0.8$
10	0.7425	2.2410	4.5084	6.5784
12	0.8202	2.3092	4.2187	6.7317
15	0.8812	2.2225	4.5198	7.0512

Table 3.2: Simulated Traffic: Drop rates (%), with 25M constrained link, total bandwidth per contract = 5M and per path peak varying from 2.5M to 4M

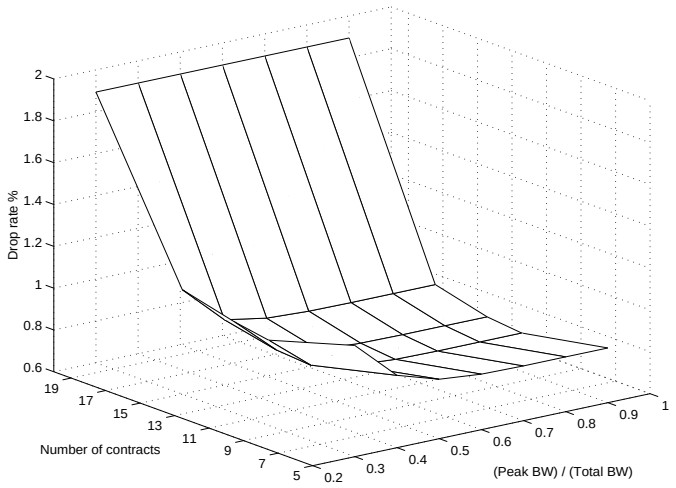


Figure 3.12: Star Wars trace: Drop Rates (%) against number of contracts and Peak/Total ratio

3.6.1.3 Customer gain with trace-driven simulations

In this section we shall examine the same parameters as those in the previous subsection, the difference being the source of traffic. The results here are pertaining to packets generated from traces of actual traffic. In Figure 3.12 the drop rates (%) are plotted against the number of contracts and the ratio p_j/T_j. The plot and the data in Table 3.3 indicate that even with a peak to total ratio of 0.6 the drop rates are within 0.8%. This reinforces the fact that there is cost saving for the customer in opting for the point to set model, if the specified error rates are tolerable.

3.6.1.4 Provider gain with trace-driven simulations

In the case of static provisioning, the provider sets aside the peak contracted bandwidth causing resources to be under-utilized. However, with dynamic provi-

Num Contracts	$p_j/T_j = 0.5$	$p_j/T_j = 0.6$	$p_j/T_j = 0.7$	$p_j/T_j = 0.8$
10	0.8007	0.7863	0.7834	0.7833
12	0.8167	0.8042	0.8013	0.8012
15	0.8769	0.8676	0.8654	0.8654

Table 3.3: Star Wars trace: Drop rates (%), with 25M constrained link, total bandwidth per contract = 5M and per path peak varying from 2.5M to 4M

sioning of resources, the provider sets aside as much bandwidth as is demanded by the customer traffic. If the average demand seen for a contract j is $\bar{d}_{ji}$ toward destination i, then the ratio of peak P_j to $\bar{d}_{ji}$ quantifies the average gain for the provider. Taking note of this fact, the Table 3.4 presents relevant data for a particular path. Each row of the table gives data for a particular range of gain. Thus in the simulation with 12 contracts, there were 3 contracts for which the gain was in the range $(1.0, 1.5)$ and the average gain was 1.14. From Table 3.4 we see, that higher gains (for the provider) are accompanied by a higher drop rate (for the customer). But the average gain with drop rates within 0.65% was 1.14. This means the provider can increase his resource utilization by about 39% $(1 - \bar{d}_{ji}/P_j)$ while keeping drop rates within 0.65%.

It is important to observe here that with 15 contracts, although the sum of the peak rates (37.5M) exceeds the path capacity the drop rates are low. This means that as compared to static provisioning, point-to-set service delivers considerable gains to the provider. However if the number of contracts being multiplexed increased to 20, the drop rates go beyond 1% since the sources are more likely to overwhelm the path capacity.

3.6.2 Multiple Ingress Topology

In the previous sections, the topology used had a single ingress. Here the case of multiple ingresses is examined. The topology (a graph derived from the well know MCI backbone topology, Figure 3.14) shows the ingress and egress nodes in bold. The sources generate traffic as described in Section 3.6.1. The core of the network has links which are depicted by thick lines. These links were chosen to be 75Mbps.

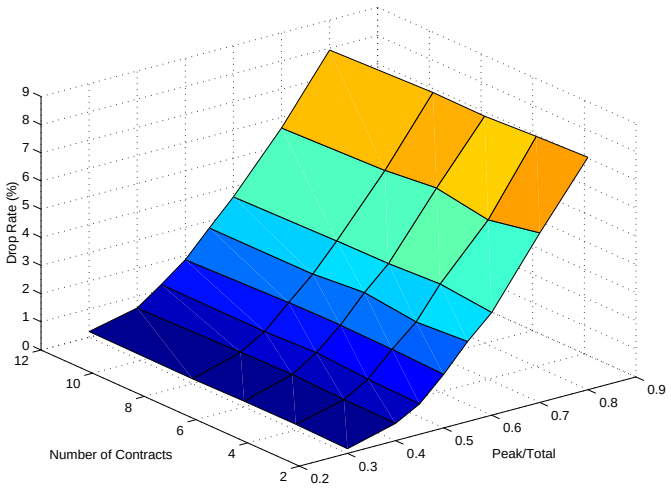


Figure 3.13: Simulated Traffic: Drop Rates (%) against number of contracts and Peak/Total ratio for multiple ingress topology

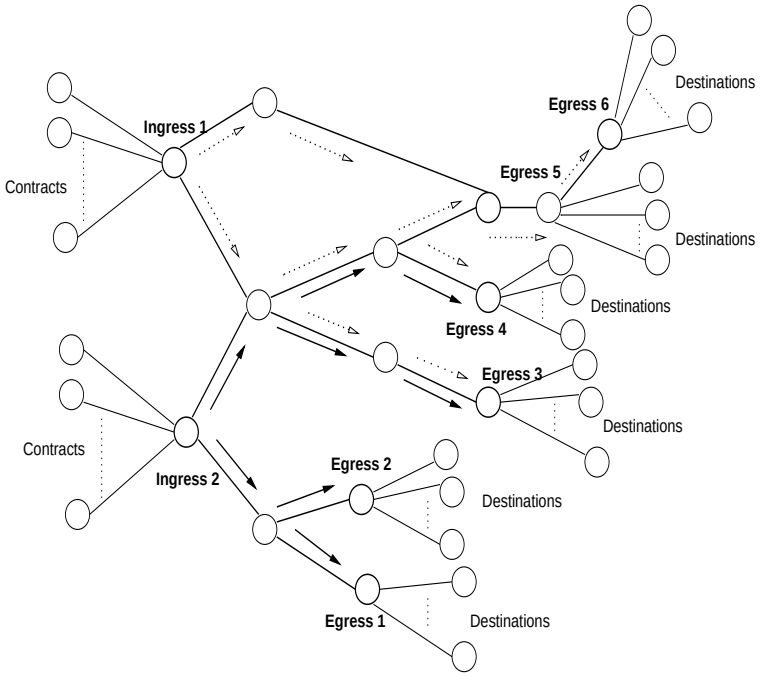


Figure 3.14: Multiple Ingress Topology

Num Contracts	Gain Range	Avg Gain $P_j/\bar{d}_{ji}$	Drop Rate (%)	Num Contracts seeing gain
12	(1.0,1.5)	1.14	0.04	3
	(1.5,2.0)	1.65	0.66	3
	(2.0,2.5)	2.03	0.40	1
	(2.5,3.0)	2.86	1.40	2
	(3.5,4.0)	3.15	1.00	2
	(4.0,4.5)	4.45	2.20	1
15	(1.0,1.5)	1.09	0.13	3
	(1.5,2.0)	1.70	0.65	4
	(2.0,2.5)	2.39	0.40	1
	(2.5,3.0)	2.77	0.50	1
	(3.5,4.0)	3.53	1.47	3
	(4.0,5.5)	4.80	1.60	3
20	(1.0,1.5)	1.15	0.88	5
	(1.5,2.0)	1.69	1.26	6
	(2.0,2.5)	2.19	1.00	1
	(3.5,4.0)	3.30	2.84	5
	(4.0,4.5)	4.42	3.47	3

Table 3.4: Star Wars trace: Provider gain with 25M constrained path, total bandwidth per contract = 5M and per path peak = 2.5M

The access links which input traffic for each contract, were chosen to be 10Mbps. The total (T_j) for each contract was 10Mbps. The minimum assured rate was thus 2.5Mbps. For simplicity, the per-path peak values were assumed equal. The traffic from *Ingress 1* is marked by dotted arrows, while that from *Ingress 2* is marked with bold arrows. Traffic from each of the ingresses reach four egresses. Due to the way the traffic is routed, the path connecting *Ingress 2* to *Egress 4* and those linking *Ingress 1* to *Egress 5 and 6* share a link. Thus this link has three paths through it. In the simulation a simplistic strategy of dividing the link capacity by three was used to decide the capacity of each of these paths. Similarly the other path capacities are decided.

The drop rates observed with simulated traffic and multiple edge topology is seen in Figure 3.13. The drop rates for a peak to total ratio of 0.45 is about 1.2%. With the single ingress topology, the observed drop rate for p_j/T_j set to 0.5 is around

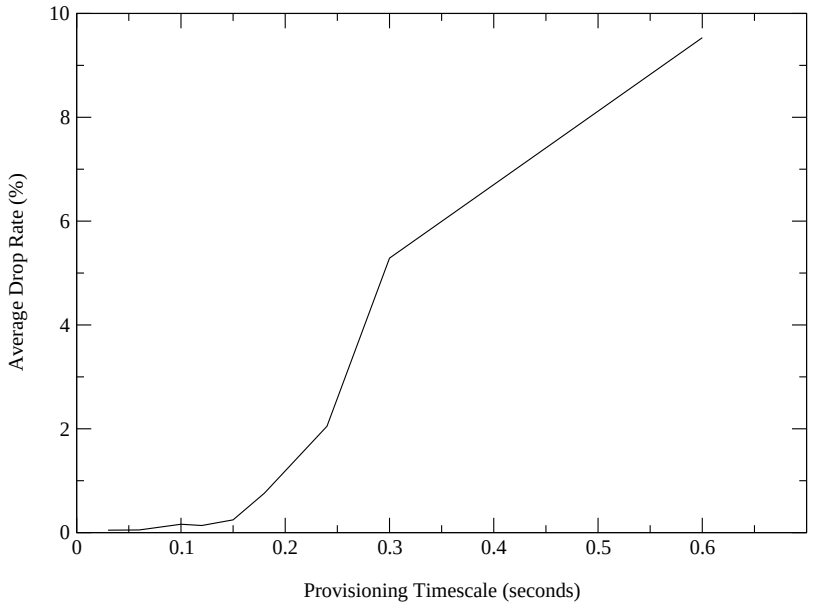


Figure 3.15: Drop Rates (%) against different provisioning timescales

0.8%. Thus the multiplexing of paths in the topology caused a degradation in the service. But still the customer sees a gain of about 44.4% with a modest drop rate of 1.2%.

3.6.3 Effect of provisioning timescale on performance

The provider-side shapers are dynamically re-provisioned periodically every T seconds. The choice of T affects the performance of the provisioning algorithm. This is because, T decides the responsiveness of the decision process to changes in the input traffic profile. If the the response is quick, the drop rates are reduced. In the above simulations the timescale was set to 1.5 times the edge to edge propagation delay (0.1 seconds). If the timescale is varied to different multiples of RTT, we observe the drop rates as in Figure 3.15. As the provisioning timescale increases with respect to the edge to edge propagation delay, the drop rates increase.

3.7 Conclusions

This chapter developed the building blocks for an ideal implementation of point-to-set assured service capabilities. The contributions include the development of a simple architecture, contract model, online demand estimation, dynamic provi-

sioning and suggestions for a simple admission control technique. Simulation results were presented to demonstrate the gains to the customer and provider due to the scheme. The scheme was shown to deliver gains to the provider in terms of resource utilization while keeping the drop rates low. The customers on the other hand stand to gain in terms of cost savings in that only the required total bandwidth needs to be purchased.

However, the model suffers from some notable drawbacks:

- Provisioning timescale is a critical parameter deciding the performance of the model. If the "right" setting is not employed, the customers can suffer from high loss rates.
- The online demand estimation suffers if the traffic gets burstier. This module is the means to multiplexing gain in D-P2Set and hence essential.
- The contract definition needs the specification of a per-path assured rate. It is usually hard to arrive at a setting which will not be too conservative yet achieving low loss rates.
- Choosing a value for the per-path permissible peak rate is not straightforward. We did not examine any objective criteria to choose this parameter.

Notice that the estimation of bandwidth demand and a fuzzy description of the allowable per-path traffic variability (i.e., per-path peak as compared to the assured rate) are at the heart of the problems. In the next chapter we shall tackle these issues by employing a statistical approach to point-to-set.

CHAPTER 4
Statistical Point-to-Set Service Architecture

4.1 Summary

While the point-to-set architecture was demonstrated to have advantages, the D-P2Set model suffered from problems due to bandwidth estimation and source traffic specification. In this chapter we shall tackle these issues.

- An overview of a simplified statistical model
- A probabilistic admission control test and its evaluation
- Quantifying allowable traffic variability via *flexibility*
- Performance evaluation

4.2 Introduction

Among the prime motivations for choosing a service architecture with enhanced spatial granularity, is the ability to exploit multiplexing gains due to traffic aggregation at the customer-level. In other words, the variability in per-destination offered load offers potential for multiplexing gains. The customer network offers a *bounded* total load that is split among its destinations implying that a substantial increase in load toward one destination is accompanied by a decrease in traffic toward some other destination.

The dynamic re-provisioning featured in D-P2Set and the Hose Model seeks to capture this aspect. Another key aspect of these models is the extent of information assumed about the traffic matrix. Clearly, more assumed information leads to simpler implementations. The Hose Model does not make any assumptions regarding the traffic matrix and hence lends itself to a wide variety of scenarios. However, the trade-off is the complexity of the provisioning mechanisms. The D-P2Set model assumes a per-path peak and minimum assured rate as being given. Even if such

parameters for offered load are specified, it still does not free us from implementation complexities involving dynamic tracking. The *deterministic* nature of such a minimum rate and peak rate specification fails to capture the dynamics of traffic and hence is very rigid.

In this chapter we strike a *middle ground* in terms of traffic information assumed. Instead of assuming nothing as in the Hose model or opting for rigid deterministic parameters as in D-P2Set, we shall exploit the statistical characteristics of the per-path traffic. We look at the traffic offered by a network as being divided among the destinations according to some proportion. We characterize the fraction of traffic offered toward each destination as a random variable. We demonstrate that given just a mean and variance for these fractions, we can exploit multiplexing gains without having to resort to demand estimation or re-provisioning. Further, the components of the architecture are *simple deterministic policer and shaper* elements implemented at the edge.

In addition to the simplicity in implementation, there is an important by-product of this statistical characterization - that of a new intuitive measure of the *flexibility* of a point-to-set service. As the term suggests, a point-to-set service has maximal flexibility if the customer traffic has minimal restrictions in terms of how the traffic is apportioned among its destinations. Such a service would perfectly mimic the quality and predictability of a point-to-point service toward each destination while exploiting multiplexing gains. In quantitative terms, the flexibility is defined as a limit on the variance (normalized by the mean) of the per-path traffic fraction.

For a detailed comparison with the traditional point-to-point model and the Hose Model, the reader is referred to §3.3. Our starting point (§4.2.1) will be the changes to the D-P2Set model that would yield us a simpler and more effective architecture.

4.2.1 Building A Deployable Model

In the ideal point-to-set model, the onus of gathering information regarding user traffic is completely on the provider. In order to simplify the model from the perspective of making the network simple, the user could be required to conform to

a certain traffic profile.

The resource wastage in a point-to-point allocation model is due to over-provisioning caused by lack of knowledge regarding the fraction of total load offered toward a destination. The solution to this could be in assuming something about the per-destination load. At one extreme would be the choice of assuming that a fixed fraction of the total traffic is offered toward each destination; this is no better than the point-to-point model in terms of either flexibility or multiplexing gain. In order to allow for the dynamic nature of traffic we could strike a *middle ground* between the two extremes of assuming all or nothing about the per-destination traffic. We could assume that the fraction of traffic toward a given destination is random, but has a given mean and variance (m, v).

This approach would allow the traffic fraction toward a destination to vary within the limits specified by (m, v). Further, we show that knowing the leaky-bucket parameters shaping the total traffic, one can compute bounds on the probability of observing a particular load toward a given destination. We employ this approach to demonstrate that a simple probabilistic admission control scheme can be derived in terms of the mean and variance parameters of per-destination traffic fraction and the leaky bucket parameters of the total traffic. We then show that these (m, v) parameters can be enforced using simple deterministic shaper elements.

Note that a higher variance for the per-destination fraction implies greater freedom to the user as regards to bandwidth usage. This is in essence *higher flexibility*. Exploiting this intuition, we define flexibility as an upper bound on the variance to mean ratio of per-destination traffic fraction. We demonstrate via simulations that this definition satisfies all the intuitive requirements of such a measure. We then explore the role of flexibility in the trade-off between loss rate and multiplexing gains.

Thus the highlights of this chapter are: a) A novel architecture for statistical edge-based bandwidth provisioning toward a set of destinations; b) A simple means to capture and enforce the per-destination traffic statistics; c) A probabilistic admission control test that allows a flexible service for the user with simultaneous multiplexing gains to the provider; and d) A solution that can be deployed at the

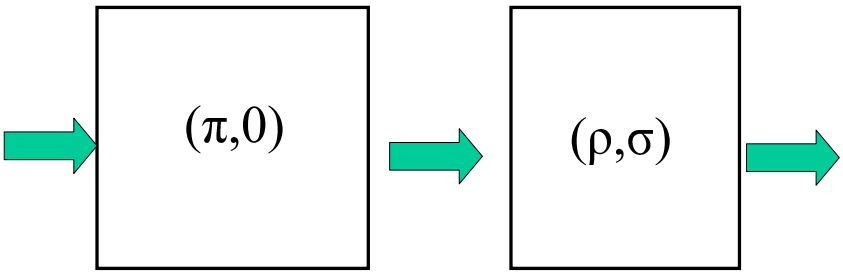


Figure 4.1: A dual-leaky-bucket regulator has two shapers in series.

edge nodes of the provider network without altering the core.

4.3 The S-P2Set Architecture

We first outline the assumptions and notations for the rest of the paper (§4.3.1). After an overview of the architecture (§4.3.2) the components are defined (§4.3.3).

4.3.1 Notations and Assumptions

Table 4.1 provides a brief description of the symbols that are used in the succeeding sections. In the following sections, a "user" refers to a customer network offering traffic. A "flow" is a traffic aggregate emanating from a network. The user traffic is assumed to be shaped by a dual-leaky-bucket regulator of the form (π, ρ, σ) (Fig. 4.1). Thus, the cumulative offered traffic $A(t)$ in time t always satisfies $\{A(t) \leq \pi t, A(t) \leq \rho t + \sigma\}$. This is equivalent to having (ρ, σ) and $(\pi, 0)$ leaky-bucket shapers in series. A QoS commitment to the user is termed as a contract (defined in §4.3.3).

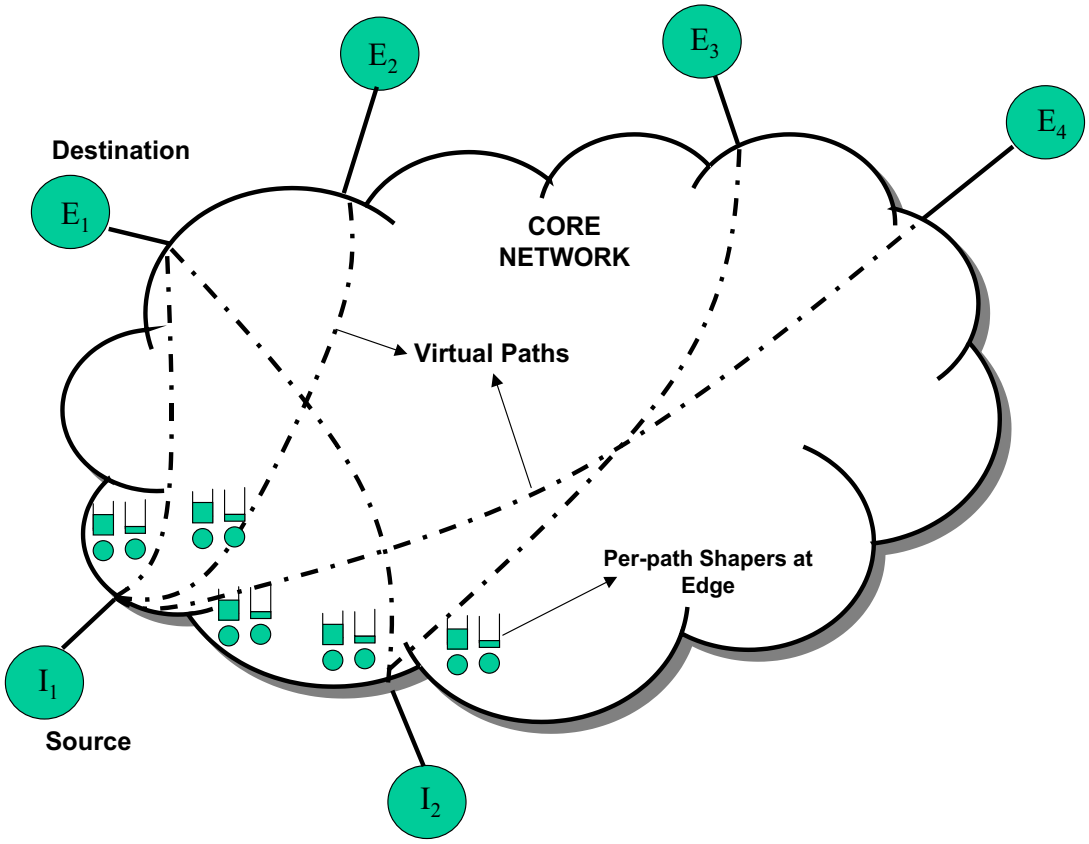


Figure 4.2: The S-P2Set Architectural Model consists of dual-leaky-bucket regulators per-path for a source network offering traffic. In the figure, traffic from network I_1 is directed toward E_1,E_2 and E_4. Each of these virtual paths is regulated at the ingress.

The admission control module is assumed to know the paths connecting ingresses and egresses. Routes are assumed to remain stable.

4.3.2 Overview

The S-P2Set architecture is depicted in Fig. 4.2. Each user network that enters into a contract with the provider is assumed to specify the set of destinations and the mean and variance of per-destination traffic fraction. This fraction is enforced via dual-leaky buckets as shown in Fig. 4.2.

The admission control module is a central entity to the provider network that knows the paths connecting the provider edge nodes. A contract requires bandwidth

Symbol	Meaning
π_j,ρ_j, σ_j	Peak rate, Avg rate, Bucket for user j
π_{ij},ρ_{ij}, σ_{ij}	Peak, Avg, Bucket for user j toward dest i
p_{ij}	Random var. for traffic fraction toward i for user j
m_{ij}	Mean of fraction of load toward dest i for user j
v_{ij}	Variance of traffic fraction toward dest i for user j
C_i	Capacity of path i
D_{max}	Max permissible delay at ingress
ϵ	Max allowed capacity violation probability
Y_j	Random var. for total traffic due to user j
X_{ij}	Random var. for traffic toward node i for user j
F, f	Flexibility
Γ_j	Capacity of link j

Table 4.1: Table of Notations

provisioning toward every destination in its set. Consequently, a new contract can be admitted only if the bandwidth requirement can be accommodated along each *path* connecting the ingress node to a destination. Thus while deciding to admit a contract, the module checks whether adding this contract would cause input rate to exceed the "path capacity". For a detailed discussion of path capacity the reader is referred to §3.4. Here we provide an intuitive description of the concept.

A given path from an ingress to an egress may share one or more physical links with other paths in the network. Hence its capacity is not the same as that of the physical links constituting the path. For admission control purposes, it is convenient to introduce a notion of a *virtual path* of fixed capacity connecting the ingress to the egress. In other words, a virtual path is a means to apportion link capacities among paths. As shown in Fig. 4.2 a virtual path between an ingress-egress pair appears as if it is dedicated to this pair.

Then the task of the admission control test is to verify that the probability that the input traffic exceeds the path capacity is less than a given threshold for every path affected by the new contract. The idea of a virtual path thus allows us to achieve admission control at the edge of the network.

4.3.3 Definitions

We first recall the definition of path capacity and present the contract specification.

Definition 4.3.1. Consider a path defined by the sequence of links $\{M_j\}$. Let Γ_j be the capacity of M_j. Let n_j be the number of paths passing through M_j. Then, the capacity of the path, is defined as: $C = \min_j \frac{\Gamma_j}{n_j}$.

Definition 4.3.2. A Contract for user network j consists of the dual-leaky-bucket characterization of the total traffic given by $(\pi_j, \rho_j, \sigma_j)$, the finite set of destination nodes, S_j, the set of pairs $\{(m_{ij}, v_{ij}) \mid i \in S_j\}$ where (m_{ij}, v_{ij}) are the mean and variance of the random variable p_{ij} indicating the fraction of total traffic toward i. So if total traffic is given by Y_j and X_{ij} indicates the traffic toward destination i, $X_{ij} = p_{ij}Y_j$, $p_{ij} \in (0, 1]$ and $\sum_i X_{ij} = Y_j$.

4.3.4 Admission Control Test

The key idea that we exploit here is that of getting an *a priori* estimate of the fraction of total traffic that is offered along a given path. Denote the total traffic (bits per second) offered by customer j as Y_j. Let X_{ij} denote the traffic due to customer j on the path leading to the destination i. If p_{ij} are fixed constants, the provider can provision the right amount of bandwidth toward each destination. This would be identical to a point-to-point service toward each of the destinations. A more interesting and realistic situation is when p_{ij} are not fixed.

Thus we define p_{ij} as a random variable with mean and variance (m_{ij}, v_{ij}). For simplicity, we assume p_{ij} are independent of Y_j, i.e., the fraction of traffic toward a destination is independent of the total volume of traffic offered by the network. We now impose the constraint that Y_j is policed to a peak rate π_j and shaped by a leaky-bucket shaper (ρ_j, σ_j).

Our goal is to reserve only as much bandwidth as a customer offers toward a destination. Thus our admission control strategy should consider the traffic that a customer might provide toward a particular destination and also attempt to exploit multiplexing gains. Using the random variables $X_{ij}(t)$ we could formulate an

admission control condition as follows - admit a new contract if:

$$\forall i,\ Pr\{\sum_j X_{ij}(t) > C_i\} \ < \ \epsilon \tag{4.1}$$

$$\forall i,\ \sum_j m_{ij}\rho_j \ < \ C_i \tag{4.2}$$

Here $\epsilon < 1$ is a given constant.

Observe that Equation (4.1) serves our objectives well. First, it reserves per-path bandwidth depending on the amount of traffic that the contract might offer. Second, it allows us to exploit *statistical* multiplexing gains by not choosing peak provisioning. The parameter ϵ provides a control on how conservative the admission control gets. Lower the value of ϵ higher the reserved bandwidth and lower the multiplexing gain. Also note that the equation inherently captures the fact that a customer network might vary the fraction of traffic it sends along a given path. Comparing this strategy with a deterministic strategy readily points us to the gains in exploiting the varying nature of customer's traffic. A deterministic point-to-point service toward each destination would need a fixed p_{ij} or would have to choose peak provisioning. In §4.5 we quantify these gains using simulations.

The relation of ϵ to the loss rate experienced along a path is not so simple due to the distortions introduced in traffic characteristics by multiplexing at successive hops. We shall return to this aspect of loss rates in §4.4.1.

4.3.5 Quantifying Flexibility

An ideal Point-to-Set service provides an abstraction of a point-to-point link toward each destination for a contract. The source network has the *flexibility* to offer an arbitrary fraction of its total traffic toward any destination. In a realistic implementation, there will be a limit to the variability in the source network's traffic. To measure how close to ideal an implementation is, we could examine the flexibility it offers. We expect a measure of flexibility to satisfy these intuitive requirements:

- Higher the flexibility, greater is the freedom to the user in terms of load distribution with respect to the destinations.

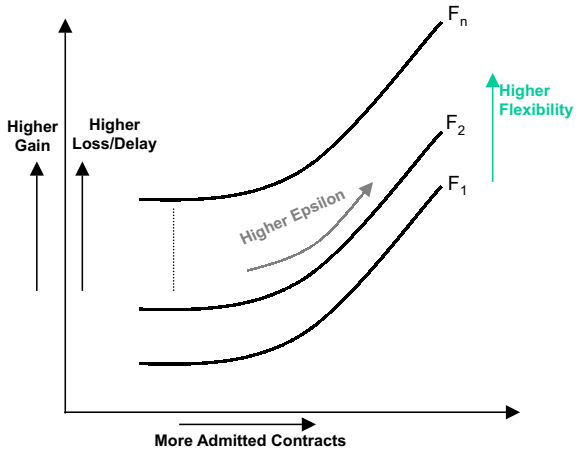


Figure 4.3: Schematic showing the significance of Epsilon (ϵ) and Flexibility. Higher flexibility requires lesser number of admitted contracts if loss rates and delays have to be maintained at the same level.

- If loss rates are kept low, higher the flexibility, closer is the service to a point-to-point regime.

  Define flexibility, f so that:

$$\frac{\sqrt{v_{ij}}}{m_{ij}} \leq f \ \ \forall i, j$$

The definition implies that a higher value of f allows for higher variance in per-path offered load. In order to attain lower loss rates and still allow for higher f one would have to admit lesser number of contracts, i.e., employ a lower value of ϵ. On the other hand, allowing higher variances for a given set of admitted contracts can lead to higher loss rates. Thus, the following can be stated as the properties of f:

- For the same multiplexing gain, higher the flexibility, higher the loss rates to the users.

- For the same loss rate, higher the flexibility lower the multiplexing gain.

These properties are captured in the schematic diagram in Fig. 4.3. As we go along one of the curves, we are holding flexibility constant while increasing the violation probability ϵ and hence increasing multiplexing gain and loss rate. If we move up

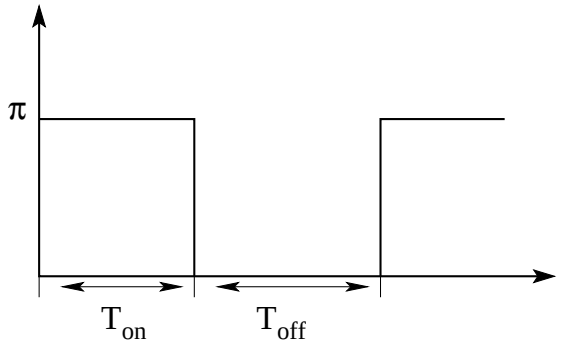


Figure 4.4: The Extremal On-Off Source

vertically (increase flexibility) for the same number of admitted contracts, we again increase loss rates. These observations are verified through simulations in §4.5. The preceding discussion thus points to a trade-off between flexibility and loss rate.

4.4 Evaluating the Admission Control Decision

In the following paragraphs we derive approximations that will help us evaluate the admission control test in Equation (4.1).

4.4.1 Per-Path Traffic Statistics

In order to evaluate Equation (4.1), we would need the distribution of X_{ij}. An alternative approach would be to *bound* the distribution somehow, exploiting the fact that Y_j was constrained by $(\pi_j, \rho_j, \sigma_j)$. We thus obtain an upper bound on the mean and variance of the process. To do this we employ a technique similar to [67] and observe that the extremal "on-off" source (Figure 4.4) has the maximum variance among all rate patterns that can be obtained given the $(\pi_j, \rho_j, \sigma_j)$ characterization, if mean is set at ρ_j (Proposition 4.4.1). Our approach differs from that of [67] in having a bound independent of a specific interval or duration, and in considering the dual leaky-bucket shaped inputs specified by $(\pi_j, \rho_j, \sigma_j)$.

We then employ this upper bound on the variance of rate to obtain a bound on per-path traffic statistics (Proposition 4.4.2). Equipped with this result we consider an approximate evaluation of Equation (4.1) (Proposition 4.4.3).

We note that although the extremal on-off source has maximum variance it does not necessarily maximize buffer overflow probability [28, 63, 100]. The extremal

source has been used in the past [34, 94, 107] with reference to bandwidth and buffer allocation. Here we employ the source owing to the fact that it leads to more conservative provisioning while easing analysis.

Proposition 4.4.1. *Consider a source shaped as $(\pi_j, \rho_j, \sigma_j)$. A transmission pattern with mean ρ_j, that maximizes the variance of rate is given by the periodic extremal on-off source, wherein the source transmits at the peak rate π_j for a duration $T_{on} = \frac{\sigma_j}{\pi_j - \rho_j}$ and switches off for $T_{off} = \frac{\sigma_j}{\rho_j}$.*

Proof. Consider the density function $f_X(x)$ corresponding to the extremal source and its variance v_X:

$$\begin{aligned} f_X(x) &= \frac{T_{off}}{T_{on}+T_{off}}\delta(x) + \frac{T_{on}}{T_{on}+T_{off}}\delta(x-\pi_j) \\ v_X &= \pi_j^2 \frac{T_{on}}{T_{on}+T_{off}} - \rho_j^2 \\ &= \pi_j\rho_j - \rho_j^2 \end{aligned} \tag{4.3}$$

Let $f_Y(y)$ denote any other density function such that Y is shaped according to $(\pi_j, \rho_j, \sigma_j)$ and has mean ρ_j. Compare its variance, v_Y with that of X:

$$\begin{aligned} v_X - v_Y &= E\{X^2\} - E\{Y^2\} - (E\{X\})^2 + (E\{Y\})^2 \\ &= E\{X^2\} - E\{Y^2\} \\ &= \pi_j\rho_j - \int_0^{\pi_j} y^2 f_Y(y)dy \\ &= \int_0^{\pi_j} \pi_j y f_Y(y)dy - \int_0^{\pi_j} y^2 f_Y(y)dy \\ &= \int_0^{\pi_j} y(\pi_j - y) f_Y(y)dy \\ &\geq 0 \end{aligned}$$

□

With this proposition, we can now consider the first and second moments of the per-path traffic due to a contract, namely, X_{ij}. The statistical characteristics of traffic is altered by each hop of multiplexing. Multiplexing introduces correlation among flows and increases burstiness [75]. Although the mean remains the

same, the variance of rate is higher at a node further along a path in the network. This has implications on provisioning buffers inside the network. We can evaluate Equation (4.1) to ensure that the mean of admitted traffic remains below the path capacity and the buffer requirement at the edge of the network is low. To achieve low loss rates, buffers inside the network have to be appropriately set. We first examine Equation (4.1) and treat buffer dimensioning in §4.4.3.

Proposition 4.4.2. *If Y_j, the total traffic due to customer j, shaped by a dual leaky-bucket shaper (π_j,ρ_j,σ_j) has a mean ρ_j and X_{ij} is the fraction of Y_j along path i, the mean and variance of X_{ij} are given as follows.*

$$E\{X_{ij}\} = m_{ij}\rho_j \tag{4.4}$$

$$Var\{X_{ij}\} \leq m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j) \tag{4.5}$$

Proof.

$$\begin{aligned} E\{X_{ij}\} &= E\{p_{ij}Y_j\} \\ &= E\{p_{ij}\}E\{Y_j\} \\ &= m_{ij}\rho_j \\ Var\{X_{ij}\} &= E\{X_{ij}^2\} - (E\{X_{ij}\})^2 \\ &= E\{p_{ij}^2\}E\{Y_j^2\} - m_{ij}^2\rho_j^2 \\ &\leq (v_{ij} + m_{ij}^2)\pi_j\rho_j - m_{ij}^2\rho_j^2 \\ &= m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j) \end{aligned} \tag{4.6}$$

□

Observing that for each path, the statistical characteristics of the traffic offered by a given customer is independent of those of others at the *edge of the network* we now propose an approximation.

Proposition 4.4.3. *Define the Gaussian random variable Z_i with mean $m_{Z_i} = \sum_j m_{ij}\rho_j$ and variance $v_{Z_i} = \sum_j m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j)$. Then for sufficiently large number of admitted customers, we have the following approximation.*

$$Pr\{\sum_j X_{ij} > C_i\} \leq Pr\{Z_i > C_i\}$$

$$\approx \frac{1}{\sqrt{2\pi}} exp\left(-\frac{(C_i - m_{Z_i})^2}{2v_{Z_i}}\right) \tag{4.7}$$

Proof. Since $X_{ij}, \forall j$ are independent, $Var\{\sum_j X_{ij}\}$ is given by $\sum_j Var\{X_{ij}\}$ which is less than v_{Z_i} as defined in Proposition 4.4.3. Note that $Var\{X_{ij}\}, \forall j$ can be assumed to be small compared to v_{Z_i} for sufficiently large number of customers. Then we can invoke the Central Limit Theorem to approximate Equation (4.1) by the Gaussian complementary cumulative probability as specified above in Equation (4.7). □

4.4.2 Enforcing the per-path limits

Once a contract is admitted, the provider needs to ensure that the offered traffic adheres to the per-path mean and variance restrictions. Thus we need shaping elements that will enforce the terms of the contract, viz., the per-path mean and variance specified by (m_{ij}, v_{ij}). As demonstrated by the following proposition, it is straightforward to derive the dual leaky bucket shaper $(\pi_{ij}, \sigma_{ij}, \rho_{ij})$ for the path i in terms of (m_{ij}, v_{ij}) and (π_j, ρ_j).

Proposition 4.4.4. *Define the dual leaky bucket shaper $(\pi_{ij}, \sigma_{ij}, \rho_{ij})$ such that:*

$$\pi_{ij} = \pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) \tag{4.8}$$

$$\sigma_{ij} = \sigma_j \tag{4.9}$$

$$\rho_{ij} = m_{ij}\rho_j \tag{4.10}$$

This dual leaky bucket shaper ensures that the per-path traffic fraction with mean $m_{ij}\rho_j$ has variance less than $m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - m_{ij}\rho_j)$

Proof. Denote the variance of the output process of this shaper by v. From Equa-

tion (4.3) we see that

$$\begin{aligned} v &\leq \rho_{ij}(\pi_{ij} - \rho_{ij}) \\ &= m_{ij}\rho_j(\pi_j(\frac{v_{ij}}{m_{ij}} + m_{ij}) - \rho_{ij}) \end{aligned}$$

□

With the above proposition, we now have the ability to implement the model with simple shaping elements.

4.4.3 Buffer Dimensioning

In order to decide the size of buffers at each hop, we can either set a limit on the maximum tolerable per-hop delay or constrain the maximum burstiness of the input traffic at each node.

Let the maximum tolerable per-hop delay be D_{max}. The corresponding buffer size at a multiplexer serving at rate C would be given by $C \times D_{max}$. While this strategy is simple and limits the maximum delay incurred, it can result in higher loss rates owing to increased burstiness inside the network.

The alternative of limiting the input burstiness at a given node inside the network is slightly more involved. We first note that the worst-case burstiness of a flow at the exit of a node increases in proportion to the sum of the burst characterizations of other flows being served by the same node. To limit the burstiness of a flow incident at a given node, we must limit the increase in burstiness due to every previous hop through which this flow passed. We do this by limiting the maximum increase in burstiness at the ingress.

Consider a multiplexer M. Let P denote the set of multiplexers feeding traffic to M and L denote the set of incident flows at M. Let D_i^{max} denote the maximum tolerable delay at multiplexer i. We can set the buffer size at a multiplexer i to $\sum_{l \in L} \sigma_l$ or a quantity that upper bounds it, as given below.

$$\sum_{l \in L} \sigma_l = \sum_{p \in P} \sum_{l \in p} \sigma_l$$

$$\leq \sum_{p \in P} D_p^{max} C_p = D_M^{max} \tag{4.11}$$

If we set D_{max} to be the maximum tolerable delay at every ingress, we can recursively compute the bound given in Equation (4.11) for a specific topology.

By using the second strategy, we observe that higher buffers are allocated at a multiplexer further along a path. Thus loss rates are reduced as compared to the first strategy. However, the trade-off is the assumption that the paths between every ingress and egress be known. Since this assumption is required to compute path capacities for admission control, it does not increase the complexity of the service.

4.5 Performance Evaluation

In this section we verify working of the model using extensive simulations. In the succeeding sections the performance evaluation is performed with the following objectives:

- To verify the superiority of the probabilistic admission control condition in terms number of admitted contracts in comparison to point-to-point allocation model (§4.5.2).
- To validate the intuition behind the definition of flexibility (§4.5.3).
- To examine the role of ϵ as a "control knob" on how conservative the provisioning gets, i.e., lower ϵ should give us lower losses and delays with lower multiplexing gain and vice-versa (§4.5.4).
- To understand the aspects of utilization (viz., average and maximum) affected by varying ϵ and flexibility (§4.5.5).
- To study the effect of bias in offered load toward a few destinations in the destination set on multiplexing gains (§4.5.6).

We begin by detailing the method used to setup the simulations.

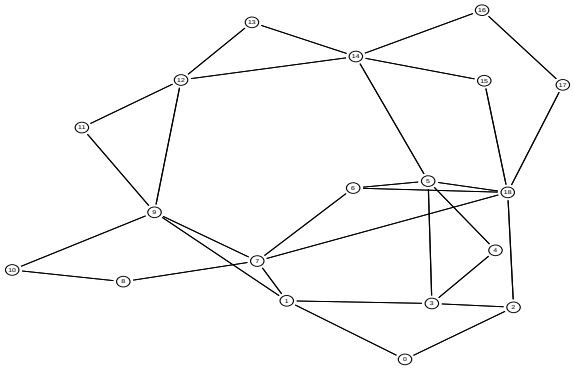


Figure 4.5: The MCI topology used in simulations. Link capacities were set to $10\ Mbps$ **and propagation delay was set to** $10\ ms$

4.5.1 Methodology

In order to evaluate the scheme, we employ Auckland IV traffic traces [86] [88] with the MCI backbone topology (shown in Fig. 4.5) in NS-2 simulation environment [118]. Each simulation consists of two phases - an admission control phase and a traffic generation phase. The simulation is started with a set of values for flexibility, ϵ and D_{max} and is provided with randomly generated contracts. As examined earlier the contract consists of the set of destinations, the dual-leaky-bucket parameters for the total traffic and per-destination mean and variance for the traffic fraction. The contracts are admitted one after another until the admission control test fails. Then the traffic generation phase starts where the network performance metrics are measured for the admitted contracts.

To generate a contract randomly the following procedure was followed. For a destination set with 4 nodes, three uniform random numbers, $r_i, i = 2 \ldots 4$ are generated in the range $[min, max]$. Then setting $r_1 = 1$ and $\sum_i r_i m_{1j} = 1$ we obtain $m_{ij} = r_i m_{1j}$. For a given flexibility v_{ij} can then be computed. The total traffic is then apportioned according to a Normal random variable with mean and variance (m_{ij}, v_{ij}) with negatives mapped to a small positive fraction.

The range $[min, max]$ decides the *bias* toward a subset of destinations in the set. If the range is small and around 1, traffic is equably directed to all nodes in the set. Higher the value of max greater the spread of the load distribution among destinations. This aspect helps us gauge the effect of bias on utilization - a higher

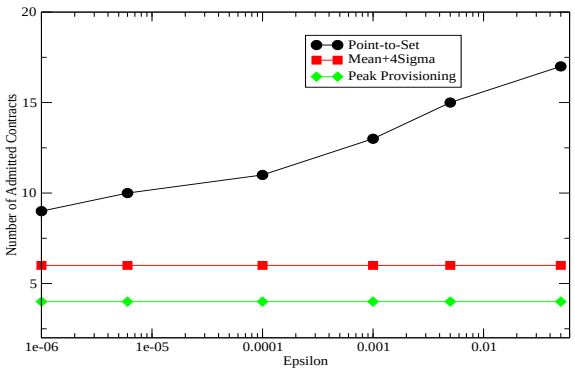


Figure 4.6: Number of Admitted Contracts increases with increasing epsilon. The probabilistic admission control beats both $mean + 4 * sigma$ and peak provisioning

bias can be expected to cause lower utilization.

In the simulations, the dual-leaky-bucket regulator parameters for all contracts was set at ($0.75\ Mbps$, $0.5\ Mbps$, $100\ kb$). The link capacities were set to $10\ Mbps$ and their delay was chosen to be $10\ ms$. In the succeeding sections, each point in a graph indicating a simulation result, is the average of 10 simulation runs.

4.5.2 Comparing with the Point-to-Point Model

The motivation for deploying point-to-set services is in the fact that there are multiplexing gains for the provider. In order to examine this aspect for the MCI topology, we compare the number of admitted contracts in a point-to-set service to that in a point-to-point service.

A point-to-point service provisions links at peak rate toward each of the destinations in the set. In addition to this, we introduce a model where the provider reserves $m_{ij} + 4\sqrt{v_{ij}}$ instead of doing a probabilistic admission control. Although this scheme is deterministic, it exploits the additional information regarding per-destination traffic fraction. Fig. 4.6 shows the number of admitted contracts under these three schemes and clearly a probabilistic scheme performs much better.

4.5.3 Admitted Contracts and Flexibility

Before we study the effect of ϵ and flexibility on loss rates and delays, we must first understand their significance from the standpoint of number of admitted

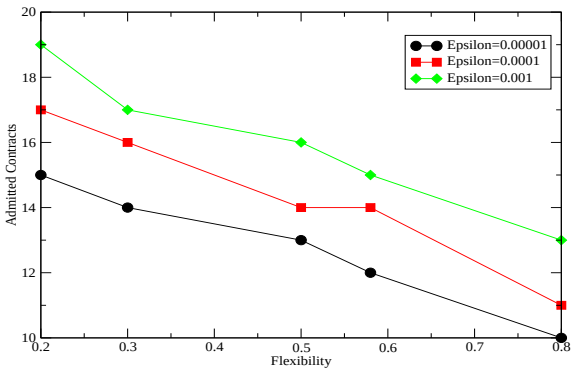


Figure 4.7: For a fixed violation probability (ϵ), higher flexibility implies lesser number of admitted contracts.

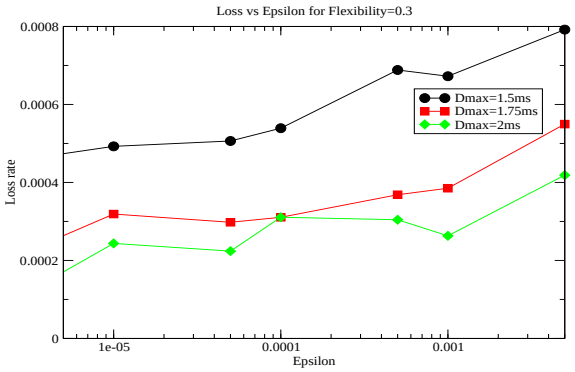


Figure 4.8: Losses increase with more admitted contracts (increasing ϵ) and lower buffer sizes (decreasing D_{max}).

contracts. In order to do this, we assign a fixed value to epsilon and study the number of contracts that can be admitted for various values of flexibility.

Intuitively, if a higher value of flexibility is allowed, the variance of per-destination traffic fraction can be higher. This points to the fact that for the same ϵ lower number of contracts will be admitted. This is observed in Fig. 4.7. As before, with higher values of epsilon, the number of admitted contracts is higher.

This agrees with our initial description of flexibility as a measure in Fig. 4.3.

4.5.4 Effect of Parameters on Loss and Delay

In the preceding sections we observed that the ϵ and flexibility can be used to increase or decrease the number of admitted contracts. This implies that these two parameters serve as a handle on how conservative the provisioning gets. For

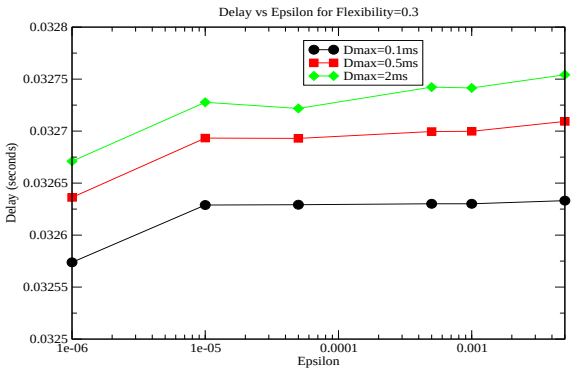


Figure 4.9: Higher buffer sizes (D_{max}) and more number of admitted contracts imply higher average end-to-end delays.

the provider, these parameters present a trade-off between multiplexing gains and loss rates. For the user, flexibility offers a trade-off between freedom with respect to per-destination load variation and cost of the service.

Fig. 4.8 demonstrates the variation of loss rates with ϵ for different buffer sizes. Recall that D_{max} decides the maximum permissible delay at the ingress and hence the buffer sizes (§4.4.3). With higher buffers, as expected, loss rates are lower. In the present simulations, a D_{max} of $2ms$ is shown to reduce the loss rates considerably. Loss rates consistently increase with higher ϵ since there is higher multiplexing.

The reduction in losses in Fig. 4.8 with higher buffer sizes comes at the cost of increased delays. As seen in Fig. 4.9, the average end-to-end delay experienced increases with higher ϵ and D_{max}. Thus the setting of D_{max} and ϵ allows the provider to trade-off loss and delay with multiplexing gain.

Fixing ϵ and flexibility sets an upper-bound on the number of admissible contracts. If the provider chooses to allow for a higher flexibility he must admit lesser contracts to maintain the probability of capacity violation at ϵ. Hence higher flexibility for the same number of admitted contracts comes at the cost of lower multiplexing gain. In addition, the higher variance in per-destination load leads to higher losses and delay. In Fig. 4.10 and Fig. 4.11 these aspects are demonstrated. To provide the same loss and delay characteristics at higher flexibilities, the number of contracts must reduce.

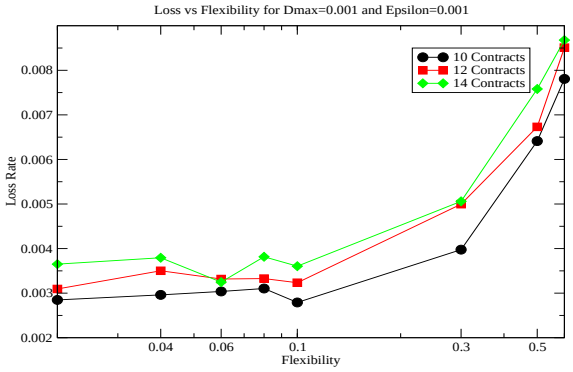


Figure 4.10: Maintaining losses at roughly the same level with increase in flexibility requires admitting lesser number of contracts.

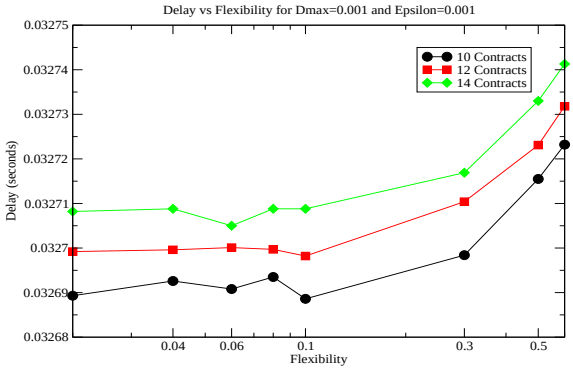


Figure 4.11: Keeping delays low with increasing flexibility requires admitting lesser number of contracts.

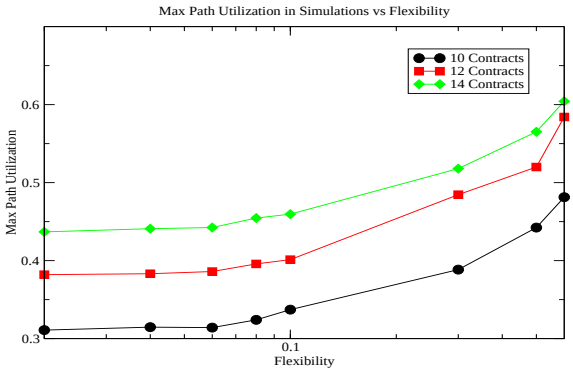


Figure 4.12: Although the average utilization remains the same, increasing flexibility allows the maximum utilization levels to be higher. Increasing ϵ provides an additional dimension in which to raise maximum utilization levels.

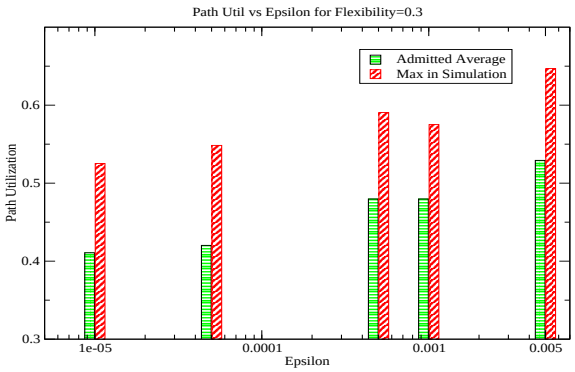


Figure 4.13: Average Path Utilization increases with increasing ϵ.

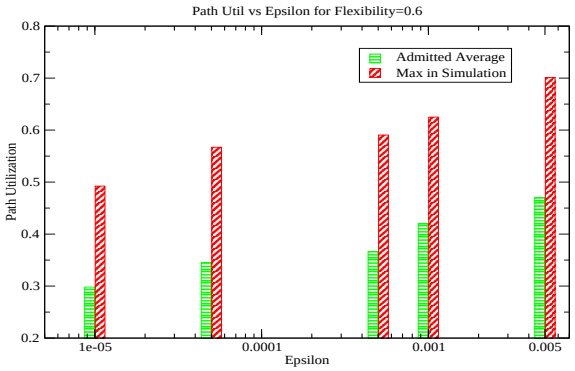


Figure 4.14: If probability of capacity violation (ϵ) is to be maintained at the same level for higher flexibility, number of admitted contracts decreases and hence the average utilization decreases (compare with Fig. 4.13).

4.5.5 Utilization

A higher ϵ allows admission of more number of contracts and hence allows for increasing the average utilization. Flexibility introduces an additional dimension to this aspect by allowing increase in the *maximum* achievable utilization for a given average utilization.

To illustrate this ability of flexibility, we turn to Fig. 4.12. For a fixed number of admitted contracts, the maximum path utilization increases with flexibility. It is important to note that in this case the number of admitted contracts has been held constant and not the violation probability. Consequently, higher flexibility allows higher utilization at the cost of a worse violation probability.

The role of ϵ in increasing the achievable average utilization is illustrated in

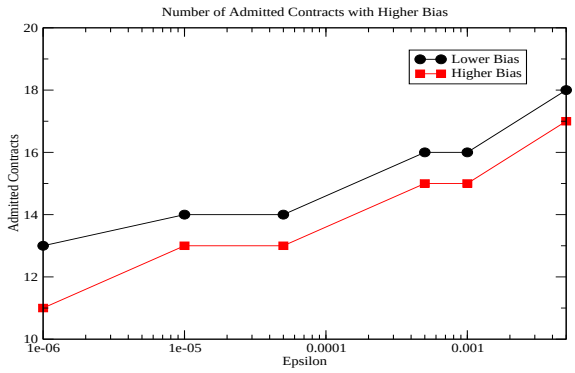


Figure 4.15: Number of admitted contracts decreases with increase in *bias*. A higher bias indicates that a higher fraction of traffic is directed at a smaller subset of destinations

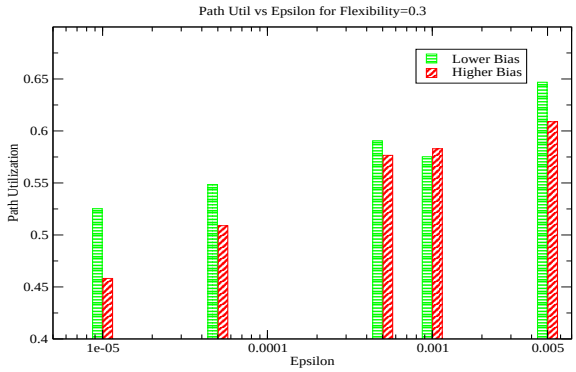


Figure 4.16: Maximum measured utilization decreases with increase in *bias*. A higher bias indicates that a higher fraction of traffic is directed at a smaller subset of destinations

Fig. 4.13 and Fig. 4.14. In this case the average utilization for a given ϵ as seen in Fig. 4.14 is lower as compared to Fig. 4.13 for a higher flexibility. This is because, the number of contracts admitted has to reduce to accommodate the same violation probability.

4.5.6 Effect of Bias in Traffic

The admission control criterion rejected a contract if it violated the capacity constraint of even one path. If there is more demand toward certain destinations, i.e. the load is biased, there would be some resource wastage. Here we just present this effect and do not provide a solution.

We recall that in the simulations the quantities m_{ij} were computed as $m_{ij} =$

$r_i m_{1j}, i > 1$ where $r_1 = 1$, and $r_i, i > 1$ is a uniform random variable over $[max, min]$. Further $m_{1j}(\sum_i r_i) = 1$. If we increase max we increase the bias of traffic toward certain destinations. Thus we obtain the number of admitted contracts and utilization for lower and higher bias cases in Fig. 4.15 and Fig. 4.16.

We see that the number of admitted contracts is lower for the same ϵ if the bias is higher. Similarly, the maximum measured utilization is lower in the case of higher bias in most cases.

4.6 Conclusions

This chapter studied a novel QoS architecture called the S-P2Set architecture. The traditional point-to-point model was extended to be able to provide considerable freedom to the user network in dynamically apportioning the allocated bandwidth among a *finite set* of destinations. The model captured the statistical characteristics of per-destination load by first defining the per-destination traffic *fraction* as a random variable and second, letting the user specify a mean and variance for this random variable. The maximum permissible value for the ratio of this variance to the mean was defined to be the *flexibility* of the model. Exploiting the independence of user aggregates at the network edge nodes, a simple probabilistic admission control test was derived. The admission control procedure exploited the notion of a virtual path connecting the ingress to each destination egress with means to compute this path's capacity. The admission control test then involved computing the probability of violating any of the virtual path capacities.

The architecture was implemented in the NS-2 simulation environment and tested with real traffic traces. The simulation results demonstrated the superiority of the model over point-to-point models. The significance of flexibility and the permissible capacity violation probability (ϵ) was characterized and verified by simulations. The parameters were shown to provide a control over the trade-off between multiplexing gains and loss rates.

Future work will involve studying means to further improve multiplexing gains by possible improvements to the admission control test. Reducing resource wastage when there is higher bias in offered load toward certain destinations also needs to

be investigated.

CHAPTER 5
A Deterministic Approach to Delay Analysis

5.1 Summary

Having examined increased spatial granularity, we now turn our attention to a priori assurances. We begin by studying a simple deterministic approach. We consider a simple network of cascaded FIFO nodes without cross-traffic and examine delay bounds. The highlights of this chapter are as below:

- Network Calculus preliminaries
- A simple network model
- A deterministic delay bound
- Drawbacks of the approach and conclusions

5.2 Introduction

Aggregate packet scheduling has attracted a lot of research attention lately. The purpose is to provide a scalable service with guaranteed rate and bounded delay. Diffserv [6] has been proposed as an architecture to achieve service differentiation. Diffserv envisages guarantees to aggregates with certain pre-defined *Per-hop Behavior* (PHB) at the individual routers. Accordingly, with expedited forwarding PHB (EF-PHB) [26], the EF aggregates must be guaranteed a particular minimum rate at each node of the network. FIFO packet scheduling has been proposed to support EF-PHB.

Recent work [17, 135] has shown that delay bounds computed (using deterministic techniques) with a FIFO network depends on the utilization level and the number of hops. Consequently, it is seen that for number of hops being as low as 3, the utilization must be kept below 50% [135]. Thus providing delay bounds with a FIFO network and aggregate scheduling needs more attention. Simulation studies conducted in [102] also illustrate scenarios where a FIFO network can lead

to very high end-to-end delays. A solution to the problem could be to extend the diffserv framework with some deadline information inserted in the packets so that aggregates can be treated accordingly. One such solution is suggested in [135].

If we are interested in a solution that does not involve changes to the network core, we would have to look for other options. A simpler incremental strategy can be evolved wherein, the number of hops of FIFO multiplexing is reduced. A specialized scheduler and shaper can be inserted after a fixed number of FIFO nodes. If we can quantify the degradation (compared to a network of specialized schedulers) in the assurance on latency after a given number of hops, we could evolve design guidelines for such incremental deployment. The analysis in [17, 135] is aimed at deriving achievable utilization bounds. Here we are interested in quantifying delay characteristics in terms of the input leaky-bucket parameters of flows. We examine this approach by obtaining, for a simple network scenario, the number of hops of FIFO multiplexing before which a worst-case latency target is violated.

We utilize results from network calculus [24, 25, 75], while dealing with leaky-bucket constrained flows. We utilize these results to obtain a characterization of the flow after it is multiplexed through n hops. An upper-bound on the latency, in terms of the flow parameters and number of hops can then be calculated. We then examine whether we can use deterministic techniques to compute useful delay bounds.

Thus the goals of this chapter are to:

- Examine a simple FIFO network using deterministic network calculus for computation of delay bounds.
- Evaluate the feasibility of the technique and the effect of parameters like path-length, input burstiness etc.

The rest of the chapter is organized as follows. In Section 5.3 we detail some definitions of terms and notations used in the paper. In section 5.4 we obtain the burstiness bound and effective service curve for a simple network. In Section 5.5 we discuss an incremental upgrade strategy using the results. Section 5.6 provides conclusions and points to future work.

5.3 Notation and background

We utilize results from deterministic network calculus [75]. Following are some basic definitions that will be useful in the succeeding sections. For a detailed introduction, the reader is referred to [75].

- **Wide-sense increasing functions**. A function f such that $f(s) \leq f(t)$ for all $s \leq t$ is wide-sense increasing. Define the set F to be the set of wide-sense increasing functions f such that $f(t) = 0$ for $t < 0$.

- **Data Flows**. A data flow, represented by a cumulative function $R(t) \in F$, is defined as the number of bits seen on the flow in the time interval $[0, t]$, and $R(0) = 0$.

- **Arrival Curve**. Given a function $\alpha \in F$, a flow R is constrained by α if and only if for all $s \leq t$, $R(t) - R(s) \leq \alpha(t - s)$. R is said to have α as an arrival curve and is said to be α-smooth.

- **Min-plus Convolution and De-Convolution**. For functions f and g from set F, min-plus convolution is defined as:

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t - s) + g(s)\}$$

  and de-convolution is defined as:

$$(f \oslash g)(t) = \sup_{u \geq 0} \{f(t + u) - g(u)\}$$

- **Service Curve**. Consider a flow going through a system S, with input and output functions R and R^*. S offers to the flow a service curve β if and only if $\beta \in F$ and $R^* \geq R \otimes \beta$.

- **Strict Service Curve**. A system S offers a strict service curve β to a flow if, during any backlogged period of duration u, the output of the flow is at least equal to $\beta(u)$.

$[x]^+$	x if $x > 0$, zero else
$[x(t)]\ 1_{\{t>y\}}$	$x(t)$ if $t > y$, zero else
$\beta_{R,T}$	Rate (R) latency (T) curve
$\gamma_{r,b}$	Leaky bucket with rate r, bucket b
β_i^θ	Service curve family with param θ
β_i^n	Service curve at node n, flow i
$\beta_i^{m,n}$	Combined Service for nodes m to n, flow i
$b_i^{(n)}$	Burstiness of flow i after n nodes

Table 5.1: Notations used in the paper

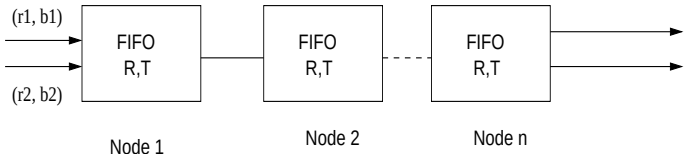


Figure 5.1: Token-bucket constrained flows fed to a cascade of FIFO nodes

- **Rate-latency Service**. A service of the form $R[t-T]^+$, where R denotes the rate and T the latency, is known as a rate-latency service, denoted as $\beta_{R,T}$.

The notations employed in the following sections has been summarized in Table 5.1.

5.4 Cascading FIFO nodes

In this section, we examine the effect of successive FIFO schedulers on the burstiness of two multiplexed flows. We first present an existing result regarding burstiness increase for one node. We extend this result for n nodes. We then obtain the effective service curve for two multiplexed flows that traverse a string of n nodes.

5.4.1 Burstiness increase due to FIFO nodes

The following theorem appears in [75].

Theorem 5.4.1. *Consider a node serving two flows, 1 and 2, in FIFO order. Assume that flow 1 is constrained by one leaky bucket with rate r_1 and burstiness b_1, and flow 2 is constrained by a sub-additive arrival curve α_2. Assume that the node guarantees to the aggregate of the two flows a rate latency service curve $\beta_{R,T}$. Call*

$r_2 := \inf_{t>0} \frac{1}{t}\alpha_2(t)$ *the maximum sustainable rate for flow 2. If* $r_1 + r_2 < R$*, then at the output, flow 1 is constrained by one leaky bucket with rate* r_1 *and burstiness* b_1^* *with*

$$b_1^* = b_1 + r_1\left(T + \frac{\hat{B}}{R}\right)$$

and

$$\hat{B} = \sup_{t\geq 0}[\alpha_2(t) + r_1 t - Rt]$$

Theorem 5.4.1 can be specialized to obtain the burstiness increase when flow 2 is also leaky bucket constrained. We then have,

$$b_1^* = b_1 + r_1\left(T + \frac{b_2}{R}\right) \tag{5.1}$$

Armed with this result we consider a scenario depicted in figure 5.1. We first note that equation (5.1) can be used to obtain a leaky-bucket characterization of flow 1 as it enters node 2. If we apply theorem 5.4.1 again, we obtain the bound for burstiness after passing through two nodes. Thus we obtain the following result for n FIFO nodes:

Theorem 5.4.2. ***Burstiness Increase due to*** n ***FIFO nodes.*** *Consider* n *nodes serving two flows, 1 and 2, in FIFO order. Assume that flow* i *is constrained by the leaky bucket* (r_i, b_i) *when it enters node 1. If* $r_1 + r_2 < R$*, then, at the output of the* n^{th} *node, flow 1 is constrained by one leaky bucket with rate* r_1 *and burstiness* $b_1^{(n)}$ *with*

$$b_1^{(n)} = b_1 + (T + \frac{b_1}{R})\left(\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \frac{(r_1 r_2)^i}{R^{2i-1}} \begin{pmatrix} n \\ 2i \end{pmatrix}\right) + \\ +(T + \frac{b_2}{R})\left(\sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} \begin{pmatrix} n \\ 2i-1 \end{pmatrix} \frac{r_1^i r_2^{(i-1)}}{R^{2i-2}}\right) \tag{5.2}$$

Proof. For n=1, the result clearly holds.

For n=2, consider a FIFO node whose inputs are $\gamma_{r_1, b_1^{(1)}}$ and $\gamma_{r_2, b_2^{(1)}}$. Using

theorem 5.4.1 we have,

$$b_1^{(2)} = b_1^{(1)} + r_1\left(T + \frac{b_2^{(1)}}{R}\right) \tag{5.3}$$

$$= b_1 + \left(T + \frac{b_2}{R}\right)2r_1T + \left(T + \frac{b_1}{R}\right)\frac{r_1r_2}{R} \tag{5.4}$$

which coincides with the expression obtained by substituting $n = 2$ in theorem 5.4.2. Assume the result holds for an arbitrary natural number m. The following steps prove that the result must also hold for $m + 1$.

$$b_1^{(m+1)} = b_1^{(m)} + r_1\left(T + \frac{b_2^{(m)}}{R}\right) \tag{5.5}$$

$$\begin{aligned} = {} & b_1 + \left(T + \frac{b_2}{R}\right)\left(\sum_{i=1}^{\lfloor\frac{m+1}{2}\rfloor}\begin{pmatrix} m \\ 2i-1 \end{pmatrix}\frac{r_1^i r_2^{(i-1)}}{R^{2i-2}}\right) \\ & + \left(T + \frac{b_2}{R}\right)\left(r_1 + \sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\frac{r_1^{(i+1)}r_2^i}{R^{2i}}\begin{pmatrix} m \\ 2i \end{pmatrix}\right) \\ & + \left(T + \frac{b_1}{R}\right)\left(\sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}\frac{(r_1r_2)^i}{R^{2i-1}}\begin{pmatrix} m \\ 2i \end{pmatrix}\right) + \\ & \left(T + \frac{b_1}{R}\right)\left(\sum_{i=1}^{\lfloor\frac{m+1}{2}\rfloor}\begin{pmatrix} m \\ 2i-1 \end{pmatrix}\frac{(r_1r_2)^i}{R^{2i-1}}\right) \end{aligned} \tag{5.6}$$

Noting that,

$$\begin{pmatrix} n \\ r \end{pmatrix} + \begin{pmatrix} n \\ r-1 \end{pmatrix} = \begin{pmatrix} n+1 \\ r \end{pmatrix}$$

the appropriate terms can be combined in the above equation to yield,

$$\begin{aligned} b_1^{(m+1)} = {} & b_1 + (T + \frac{b_1}{R})\left(\sum_{i=1}^{\lfloor\frac{m+1}{2}\rfloor}\frac{(r_1r_2)^i}{R^{2i-1}}\begin{pmatrix} m+1 \\ 2i \end{pmatrix}\right) + \\ & +(T + \frac{b_2}{R})\left(\sum_{i=1}^{\lfloor\frac{m+2}{2}\rfloor}\begin{pmatrix} m+1 \\ 2i-1 \end{pmatrix}\frac{r_1^i r_2^{(i-1)}}{R^{2i-2}}\right) \end{aligned} \tag{5.7}$$

which is the desired form for $m+1$.

□

It is easy to see that equation (5.2) reduces to (5.1) for $n = 1$.

5.4.2 Effective service curve for n FIFO nodes

Consider now, n FIFO nodes that are each characterized by a minimum service curve $\beta_{R,T}$. In this section we are interested in the effective service curve a flow gets out of the n nodes when multiplexed with another flow. For this purpose we employ a theorem from [75] and exploit the fact that convolution of two service curves is equivalent to the service offered by two nodes with those service curves, in succession.

Theorem 5.4.3 appears in [104, 75].

Theorem 5.4.3. *Consider a lossless node serving two flows, 1 and 2, in FIFO order. Assume that packet arrivals are instantaneous. Assume that the node guarantees a minimum service curve β to the aggregate of the two flows. Assume that flow 2 is α_2-smooth. Define the family of functions β_1^θ by*

$$\beta_1^\theta = [\beta(t) - \alpha_2(t-\theta)]^+ 1_{\{t>\theta\}} \tag{5.8}$$

Call $R_1(t), R_1'(t)$ the input and output for flow 1. Then for any $\theta \geq 0$

$$R_1'(t) \geq R_1 \otimes \beta_1^\theta$$

If β_1^θ is wide-sense increasing, flow 1 is guaranteed the service curve β_1^θ.

If flow i is constrained by a leaky bucket γ_{r_i,b_i}, equation 5.8 can be further specialized [75] and stated as

$$\begin{aligned} \beta_1^\theta &= [R(t-T) - \gamma_{r_2,b_2}(t-\theta)] \ \textit{with}\ \theta = (T + \frac{b_2}{R}) && (5.9)\\ &= [(R-r_2)(t-\theta)] \ \textit{with}\ \theta = (T + \frac{b_2}{R}) && (5.10) \end{aligned}$$

Then β_1^θ is a service curve guaranteed to γ_{r_1,b_1}. Let the service seen by flow i at node j be denoted by β_i^j. Let the effective service seen by flow i if it passes through

n nodes be denoted as $\beta_i^{1,n}$. Then we have,

$$\beta_i^{1,n} = \beta_1^1 \otimes \beta_1^2 \otimes \ldots \otimes \beta_1^n$$

The following proposition then gives the effective service curve offered by n FIFO nodes in succession.

Proposition 5.4.1. ***An Effective Service Curve for N FIFO nodes***. *Consider n lossless nodes serving two flows, 1 and 2, in FIFO order. Assume that packet arrivals are instantaneous. Assume that each node guarantees a minimum service curve $\beta_{R,T}$ to the aggregate of the two flows. Assume that flow i is constrained by the leaky-bucket γ_{r_i,b_i}. Then the effective service curve for the n nodes, for flow 1, is given by:*

$$\beta_1^{1,n} = [(R - r_2)(t - \sum_{i=0}^{n-1} \theta_i)]^+ 1_{t > \sum_{i=0}^{n-1} \theta_i} \tag{5.11}$$

$$\theta_i = \begin{cases} T + \frac{b_2}{R} & i = 0 \\ T + \frac{b_2^{(i)}}{R} & else \end{cases} \tag{5.12}$$

and $b_2^{(n)}$ is given in equation (5.2).

Proof. Intuitively, since each FIFO node offers to flow 1, a service equal to rate $R - r_2$ with a latency θ_i (defined above), the effective service is again a rate-latency function with the latency being the sum of the latencies at each node.

For $n = 1$, we have equation (5.9).

In the following equations, θ_i is defined by equation (5.12). For $n = 2$, consider,

$$\begin{aligned} \beta_1^{1,2} &= \beta_1^1 \otimes \beta_1^2 \\ &= \inf_{0 \le s \le t} \{\beta_1^1(s) + \beta_1^2(t-s)\} \\ &= \inf_{0 \le s \le \theta_0} \{\beta_1^2(t-s)\} \wedge \inf_{s > \theta_0} \{\beta_1^1(s) + \beta_1^2(t-s)\} \end{aligned}$$

Evaluating the above equation for different values of t we find the following. For

$t \leq (\theta_0 + \theta_1)$, $\beta_1^{1,2} = 0$. For $t > (\theta_0 + \theta_1)$,

$$\begin{aligned}
\beta_1^{1,2} &= \beta_1^2(t - \theta_0) \\
&\quad \wedge \inf_{\theta_0 < s < t - \theta_1} \{\beta_1^1(s) + \beta_1^2(t - s)\} \\
&\quad \wedge \inf_{s \geq t - \theta_1} \{\beta_1^1(s)\} \qquad (5.13) \\
&= (R - r_2)(t - \theta_0 - \theta_1) \wedge (R - r_2)(t - \theta_0 - \theta_1) \\
&\quad \wedge (R - r_2)(t - \theta_0 - \theta_1) \qquad (5.14) \\
&= (R - r_2)(t - \theta_0 - \theta_1) \qquad (5.15)
\end{aligned}$$

which is the required form for n=2.

Now assume that the result holds for m. Consider, $n = m + 1$. Tracing the steps in equations (5.13,5.14,5.15) with $\sum_{i=0}^{m-1} \theta_i$ and θ_m, we easily see the result for $n = m + 1$. □

We have thus obtained the effective service curve for n FIFO nodes in equation (5.11) if the inputs are leaky bucket constrained.

5.5 Incremental deployment of specialized schedulers

The motivation for the analysis of the previous section lies in quantifying the effect of multiple hops of FIFO multiplexing. Given that a network-wide upgrade for specialized scheduling (e.g., rate controlled scheduling) is prohibitive, the question we would like to answer is as follows. What is the degradation in service offered to a flow if it has to be multiplexed across a fixed number of FIFO nodes, as compared to a network with schedulers that do not increase burstiness? The result in equation (5.11) is a step toward finding a useful answer to this question. In this section we conduct some simple numerical studies using the results of the previous sections. We first examine the effect of number of hops on the burstiness of the flow, given the initial leaky-bucket constraints of the flow. We then consider the reverse case, we fix a target worst-case latency and the number of hops, and present directions for choosing bucket depths for the flows at the entry of the network.

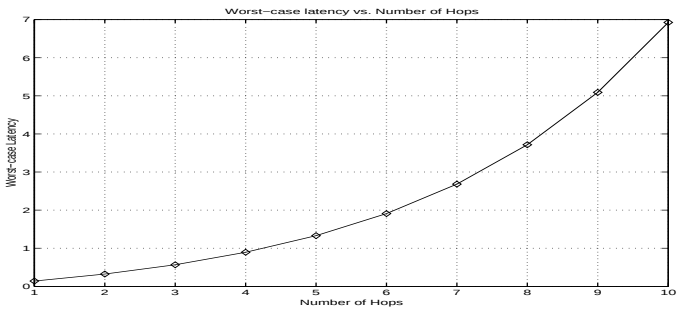


Figure 5.2: Effect of number of hops on the upper-bound on latency

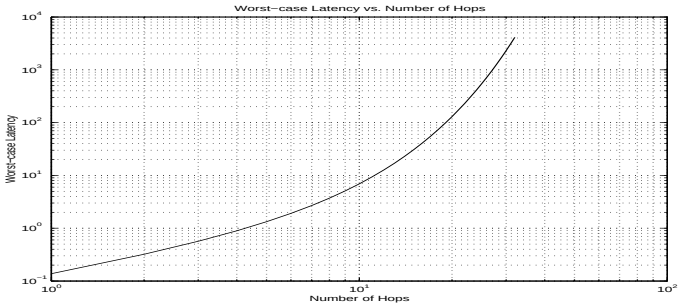


Figure 5.3: Effect of number of hops on latency upper-bound (plotted till 32 hops)

5.5.1 Effect of number of hops

Consider a network with priority queues throughout. The high priority flow obviously faces a latency of only $\tau = N(T + l^L_{max}/C)$ if there are N nodes to be traversed and l^L_{max} is the maximum packet size of the low-priority flow. The upper-bound on the latency faced in a FIFO network on the other hand is obtained by $\hat{\tau} = \sum_{i=0}^{n-1} \theta_i$. Comparing $\hat{\tau}$ and τ directly provides us with a measure of the degradation in the upper-bound on latency suffered due to the series of FIFO nodes.

We numerically evaluate the value of $\hat{\tau}$ for 1 to 10 hops and present it in figure (5.2). While viewing this plot, it is important to note that these values are for the worst-case scenario, that is it only means that the latency will never exceed those values. To obtain figure (5.2) we use values of $(r_1 = 1000000, b_1 = 10000, r_2 = 1000000, b_2 = 10000, R = 3000000, T = 0.1)$. Thus given the leaky-bucket constraints for the flows, we could decide on the number of hops after which we place a specialized scheduler if we know the tolerable degradation in delay. From the plot, we see that for a worst-case target latency of 3, every 6^{th} hop must involve

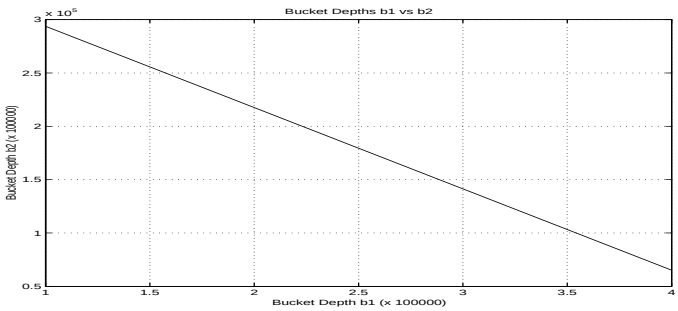


Figure 5.4: Worst-case line relating bucket depths b_1 and b_2

a shaping component.

The upper bound rapidly increases with the number of hops and is not really useful for larger n. This is illustrated in figure (5.3) where the number of hops is varied up to 32.

5.5.2 Choosing bucket depths

Using equations (5.11,5.12), we can find an expression for $\hat{\tau}$ in terms of b_1, b_2. It will be of the form

$$\hat{\tau} = c_0 + c_1 b_1 + c_2 b_2 \;\; c_i > 0$$

For each fixed $\hat{\tau}$ we then obtain an "operating line" along (or below) which b_1 and b_2 can be chosen. We can set the target $\hat{\tau}$ to a value greater than c_0 to obtain a line which can help us choose b_1 and b_2. We use the values of r_1, r_2, R, and T as $(r_1 = 1000000, r_2 = 1000000, R = 3000000, T = 0.1)$ and obtain the relation between the latency $\hat{\tau}$, b_1 and b_2, for 5 hops, as:

$$\hat{\tau} = 1.29 + 2.43 \times 10^{-6} b_1 + 1.85 \times 10^{-6} b_2$$

Setting the target latency as $\hat{\tau} = 2$, we obtain,

$$b_1 = 2.94 \times 10^5 - 0.76 b_2$$

This is plotted in figure (5.4). Thus given a target latency, we can find the operating line on or below which the bucket depths may be chosen.

5.6 Conclusions

For a simple network of FIFO nodes, we obtained a worst-case bound on the burstiness increase assuming leaky-bucket inputs. Utilizing this result, an effective service curve was obtained. An example was used to illustrate the use of the results to choose the number of hops between upgraded nodes.

There are a lot of negatives to the method outlined:

1. *Complexity:* Even for a simple two flow network with just cascaded nodes and no cross traffic, the analysis is complex. Clearly for a bigger network this method is not viable.

2. *Conservative Bounds:* The worst-case nature of deterministic analysis leads to very conservative bounds. It does not exploit the statistical nature of input traffic.

3. *Need flow parameters:* In order to obtain delay bounds we require the leaky-bucket description of each flow. If we want to provide a priori assurances, this information cannot be assumed. The bounds should be independent of such detail.

But there are some beneficial aspects of the above exercise.

1. *Computing burstiness bounds:* The analysis indicates a method to recursively evaluate the burstiness of a flow.

2. *Hints at statistical analysis:* There are certain aspects which clearly call for a statistical treatment. The latency experienced at a node is clearly not equal to the worst-case behavior of all other flows. If we can quantify the probability that a flow offers a certain burst-size, we can improve the latency bound. The second such aspect is the recursive computation of burstiness increase - the average increase in burstiness can be a lot lower than the worst-case.

We use these insights to build a statistical framework in the next chapter where the bounds are not dependent on flow parameters explicitly and are much tighter due to probabilistic analysis.

CHAPTER 6
Decoupling Delay Assurances and Traffic Profile

6.1 Summary

By combining some of the useful aspects of deterministic analysis from Chapter 5 with statistical techniques we re-examine a priori delay assurances. The goal of this chapter is to arrive at a framework to provide edge-to-edge statistical delay assurances without requiring the exact knowledge of number or nature of flows. The rest of the chapter progresses as below:

- Review of related work
- Motivation
- Overview of the proposed framework
- Deriving bounds independent of flow character
- Performance evaluation

6.2 Introduction

Bounded delay services are an essential aspect of networks intended for multimedia traffic. The delay incurred by a packet depends on the the queuing delay incurred at each hop in the path. In order to provide an upper-bound on the end-to-end delay, it is necessary to characterize the worst-case queue size that a packet might encounter at each hop. Computing the expected queue size requires information about all the flows that are incident at that hop. This implies that an end-to-end delay assurance calculation requires information about traffic at each hop in the path.

However, if we assume the exact nature of traffic at each hop (e.g., the number of flows, their leaky-bucket characteristics etc), the bounds so derived would not be useful for scenarios featuring traffic with characteristics other than what was

assumed. To rephrase the above, with existing techniques, if we do not know the exact nature of the flows in the system, we cannot obtain bounds on end-to-end delay.

For *a priori* assurances which can hold true for a large number of traffic scenarios, the assurances must be computed with minimally restrictive assumptions. The *a priori* nature of such bounds implies that we cannot assume exact knowledge of number or nature of flows in the system. In this chapter we illustrate a set of techniques which involve easily enforced assumptions on traffic character, while allowing for a large variety of scenarios in which such assurances are valid.

The following key observations about computing per-hop queuing delay, allow us to decouple computation of delay bounds from traffic profile:

1. Given the leaky-bucket description of flows as (σ_i, ρ_i), backlog and delay bounds can be computed in terms of $\sum_i \sigma_i$, $\sum_i \sigma_i^2$ and $\sum_i \rho_i$, implying that we need only bound the properties of the aggregate.

2. Each hop serves traffic directed toward multiple egresses. The set of flows originating at a particular ingress and directed toward a particular egress traverse the same *path*. Thus each hop consists of multiple aggregates, one corresponding to each path passing through this hop. Thus per-hop characteristics are bounded once aggregates are characterized.

3. The nature of flows admitted on a given path can be controlled at the ingress.

We demonstrate that enforcing the following conditions at the ingress allows us to bound the relevant quantities mentioned above:

- For each flow traversing path p, limit the ratio of σ/ρ to k_p at the ingress.
- Limit the utilization of path p to u_p

From observation (2), it follows that the properties all flows at a hop can be expressed in terms of that of each aggregate. We prove that the properties of each of these aggregates can be bounded by using the enforced limits on utilization and σ/ρ.

We begin by reviewing statistical QoS literature in §6.3.

6.3 Statistical Quality of Service

A statistical quality assurance on a performance metric M typically specifies an upper-bound on the probability $Pr\{M > m\}$. Such assurances are derived by using a probabilistic description of the input traffic.

The task of finding probabilistic envelopes for input traffic is not trivial. In his analytical framework to obtain end-to-end assurances, Kurose [72] proposes the use of *bounding random variables* so that for an arrival process X we have $Pr\{X > x\} \leq Pr\{Y > y\}$. Since it is hard to choose such random variables for Internet traffic, various other methods have been developed. Yaron and Sidi [127] proposed a probabilistic bound of the form $Pr\{X(t) > \rho t + \sigma\} \leq e^{-f(\sigma)}$ ($X(t)$ denotes the number of arrivals in bits in $[0, t]$) as an extension of the traditional leaky-bucket envelope. Starobinski and Sidi [110] generalize these bounds to obtain *stochastically bounded burstiness*, expressed as $Pr\{X(t) > \sigma\} \leq f(\sigma)$. Although a calculus was built using these stochastic envelopes to derive delay and backlog bounds at a multiplexer, the problem of enforcing these envelopes using simple deterministic network components remained open. Similar issues arise in proposals (Elwalid [34]) using a bound on the moment generating function, called the the *effective bandwidth* characterization (Kelly [61]).

Knightly [67, 66] showed that *Rate Variance* envelopes obtained as an upper bound on the variance of $X(t)/t$ had a direct relation with enforceable leaky-bucket parameters. This meant that although the input process was shaped using a deterministic component, a stochastic envelope corresponding to these parameters could be derived. Boorstyn et al [7] generalized this idea to obtain local and global effective envelopes. An alternative means to obtaining probabilistic assurances with deterministic (leaky-bucket) envelopes is provided by the Benes approach to estimating unfinished work in a queuing system. Following one such method outlined by Norros et al [90], Sivaraman and Chiussi demonstrate end-to-end delay assurances in EDF networks [107].

6.4 Motivation

Statistical envelopes that can be derived from leaky-bucket parameters are preferable from the perspective of simplicity in implementation. For example, consider the rate-variance envelope adapted for leaky-bucket shaped traffic. Indicate the cumulative arrivals in $[0, t)$ by $X(t)$. The the rate-variance envelope is defined as:

$$RV(\tau) = Var\left(\frac{X(t, t+\tau)}{\tau}\right)$$

If the flow is stationary, ergodic and is upper bounded such that $X(t, t+\tau) \leq b(\tau)$ for all $\tau, t > 0$ then,

$$RV(\tau) \leq \phi\frac{b(\tau)}{\tau} - \phi^2$$

$$\phi = \lim_{t\to\infty} \frac{b(t)}{t}$$

Now to specialize this for a leaky-bucket shaped flow (σ, ρ), we note that $\phi = \rho$ and $b(t) = \rho t + \sigma$. Hence, $RV(t) = \rho\sigma/t$. Further, for a multiplexer with N flows and a maximum busy period of T, it can be shown that [67]:

$$Pr\{D > d\} \approx \max_{0\leq\tau\leq T} \frac{1}{\sqrt{2\pi}} exp\left(-\frac{(C(\tau+d) - m_\tau)^2}{2v_\tau^2}\right)$$

$$\begin{aligned} m_\tau &= \sum_{j=1}^{N} \tau\phi_j = \tau\sum_{j=1}^{N} \rho_j \\ v_\tau^2 &= \sum_{j=1}^{N} \tau^2 RV_j(\tau) = \tau\sum_{j=1}^{N} \sigma_j\rho_j \\ T &= \frac{\sum_{j=1}^{N} \sigma_j}{C - \sum_{j=1}^{N} \rho_j} \end{aligned}$$

An important observation to be made here is that the required flow parameters are mostly the properties of the aggregate (e.g., $\sum_j \rho_j$). An alternate approach using the Hoeffding bound for evaluating a sum of independent processes demonstrated by Chang et al [16] and later adapted by Vojnović et al for EF networks, also provides delay bounds in terms of properties of the aggregate. This bound takes the following

form for a multiplexer with capacity C:

$$Pr\{D > d\} \leq$$

$$\sum_{k=0}^{K} exp\left(-\frac{2[C(\tau_k + d) - \tau_{k+1} \sum_{j=1}^{N} \rho_j)^+]^2}{(\tau_{k+1} \sum_{j=1}^{N} \rho_j + \sqrt{\sum_{j=1}^{N} \sigma_j^2})^2 \bigwedge (4 \sum_{j=1}^{N} \sigma_j^2)} \right) \tag{6.1}$$

for any $K \in \mathbb{N}$, and $0 = \tau_0 \leq \tau_1 \leq \ldots \leq \tau_K = \tau$ where τ is the upper-bound on the busy period. In the above equation, $x \wedge y$ denotes $min(x, y)$ and N denotes the number of flows feeding into the queue, each of which has a leaky-bucket description given as (σ, ρ).

Once again we make the observation that if the properties of the aggregate are known, most of the quantities in the above equations are known. However in order to compute the properties of the aggregate we need the characteristics of every flow.

If we find a way to upper-bound the properties of the aggregate using *pre-determined* network parameters and independent of individual flow descriptions, we can obtain an upper-bound on the probabilities of interest. Such a method would significantly alter the requirements for statistical assurances for the following reasons:

1. *Independence from Specific Traffic Scenarios:* The computation of a network's delay properties would not depend on the specific traffic profile at some instant.

2. *A Priori Assurances:* Since we do not need the flow descriptions, the QoS assurances can be found *a priori* given the network topology and routing protocol. The assurances are re-computed when the topology or routing changes happen and not when traffic changes happen.

3. *Edge-based Assurances:* With a link-state routing protocol, the topology and routing information is known network-wide. This implies that the QoS assurances could be computed anywhere including the *Edge* of the network.

4. *No per-hop Signaling:* For Admission control decisions based on thresholding QoS metrics, per-hop signaling would not be required anymore since these metrics can be computed at the edge.

In the following sections we shall examine one such framework which decouples QoS computation from instantaneous traffic profile. We begin by providing an overview of the techniques employed (§6.5) and then examine the framework in detail (§3.4).

6.5 Overview of the Proposed Framework

The development leading to delay computations independent of traffic profile proceeds in a sequence of logical phases.

First, instead of examining the character of each flow incident at a multiplexer we group the flows into groups traversing the same path in the network and bound the properties of these aggregates. Assuming that all flows originating at an ingress node and destined to the same egress node traverse the same set of multiplexers, we associate an aggregate with each "path" in the network. The advantage of this approach is that, unlike the number of flows, the number of paths feeding into a multiplexer is a constant (under identical routing conditions).

Second, we quantify the effect of a path in altering the flow's burstiness by restraining the nature of flows admitted along the path. We then arrive at a set of bounds for an aggregate traversing a path. Thus the information regarding flow parameters is indirectly obtained via the constraints on the flows that are admitted.

Having freed ourselves of the need to know about the number flows and their description, we then demonstrate that the delay computation at a multiplexer can be expressed in terms of the bounds so obtained.

Before examining the analysis, we refer the reader to §3.4 for a description of the notion of a path and path capacity.

6.6 Network Model and Assumptions

We consider a provider network which has a central admission control entity. Every customer is assumed to enter into a service contract with the provider and pass the admission control test.

Each node in the network has a set of input and output multiplexers. For the purposes of analysis we shall be considering the output multiplexers of each node.

Thus each node has a multiplexer associated with each link incident at the node. The output queues are assumed to be FIFO.

The points of entry for traffic from the customer are termed ingresses and the points of exit from the provider network are termed the egresses. The route from ingress to egress is assumed to be fixed and known for the purposes of analysis. If the route changes, the admission control conditions are recomputed and contracts re-negotiated. Route changes are assumed to be infrequent.

We assume that there is only one route that is used to direct packets from an ingress to an egress.

6.7 A Framework to Decouple Delay from Traffic Profile

To obtain bounds on the flow characteristics at a multiplexer, we consider a multiplexer M_n, fed by $M_1 \ldots M_{n-1}$. Our strategy will be to express the properties of a flow at M_n in terms of the flows entering $M_1 \ldots M_{n-1}$. The intuition behind the method is that if we bound the flow's properties at the entry of the network, we can obtain its description at any node inside the network by a recursive computation.

Thus we limit the ratio of σ/ρ at the network edge for each path p at k_p. Clearly, this ratio increases due to increase in burstiness as we progress along a path. We show that the properties of the aggregate at M_n can be derived in terms of $k_p^{(n)}$ (k_p at multiplexer n) and a limit on the maximum admissible rate parameter ρ^{max}. We then demonstrate the means to obtain the burst ratio for path p at multiplexer n, $k_p^{(n)}$ using a simple recursive relationship. We now state the main theorem, prove the supporting lemmas and finally provide a proof of the theorem.

Theorem 6.7.1. Consider a multiplexer M_n. Let $P_1 \ldots P_K$ denote the paths passing through M_n. Let F_i denote the set of flows traversing P_i. Let the capacity of P_i be Γ_i and let its utilization be limited to u_i (i.e., flows are admitted on P_i so long as $\sum_i \rho_i/\Gamma_i \leq u_i$). Let the following conditions be satisfied:

1. $\rho_i \leq \rho^{max}$, $\forall i$
2. $\frac{\sigma_i}{\rho_i} \leq k_p$, $\forall\, i \in$ *Path* p *at the first multiplexer on the path* p *(ingress).*

We then have the following result.

$$\begin{aligned}
\sum_j \rho_j &\leq \sum_{p=1}^{K} u_p \Gamma_p \triangleq \mu_1 && (6.2)\\
\sum_j \rho_j \sigma_j^{(n)} &\leq \rho^{max} \sum_{p=1}^{K} k_p^{(n)} u_p \Gamma_p \triangleq \mu_2 && (6.3)\\
\sum_j (\sigma_j^{(n)})^2 &\leq \rho^{max} \sum_{p=1}^{K} (k_p^{(n)})^2 u_p \Gamma_p \triangleq \mu_3 && (6.4)
\end{aligned}$$

where $k_p^{(n)}$ indicates the bound on the ratio $\frac{\sigma_i^{(n)}}{\rho_i}$ for flows i on path p at multiplexer n.

Clearly, the bounds are not dependent on either the number of flows or the parameters of the flows (ρ^{max} is a predetermined constant). In order to be able to obtain the bounds above we need to first show that $k_p^{(i)}$ exists (Lemma 6.7.1) and can be obtained (Lemma 6.7.2) for any multiplexer i.

Lemma 6.7.1. *Denote the ratio of σ_i to ρ_i for a flow i along path p at multiplexer m as $k_{p,i}^{(m)}$. Then we have:*

$$\forall i \in p \, at \, m, \;\; k_{p,i}^{(m)} \leq k_p + \sum_{i=1}^{m-1} \beta^{(i)} = k_p^{(m)} \tag{6.5}$$

where $\beta^{(i)} = \sum_j \sigma_j^{(i)} / C_i$, the cumulative burst period at multiplexer i.

Proof. Let the path p contain the sequence of multiplexers $\{1, \ldots, n\}$.

$$\begin{aligned}
\sigma_i^{(m)} &= \sigma_i^{(m-1)} + \beta^{(m-1)} \rho_i \\
k_{p,i}^{(m)} &= k_{p,i}^{(m-1)} + \beta^{(m-1)} \\
&= k_{p,i}^{(1)} + \sum_{i=1}^{m-1} \beta^{(i)}
\end{aligned} \tag{6.6}$$

However, at the ingress node we have $k_{p,i}^{(1)} \leq k_p$. Hence,

$$k_{p,i}^{(m)} \leq k_p + \sum_{i=1}^{m-1} \beta^{(i)} = k_p^{(m)} \tag{6.7}$$

□

Lemma 6.7.2. *The values of $\beta^{(m)}$ and $k_p^{(m)}$ can be bounded by the following recursive formulation with paths $\mathcal{P}$ through m.*

$$\beta^{(m)} \leq \left(\frac{\sum_{p \in \mathcal{P}} k_p^{(m)} \Gamma_p u_p}{C_m} \right) \tag{6.8}$$

$$k_p^{(m)} = k_p^{(m-1)} + \beta^{(m-1)} \tag{6.9}$$

Proof. The proof involves two parts; first, to prove that the recursive relations are true; second, to prove that the recursion can be solved for both feedforward and non-feedforward flows.

$$\begin{aligned} \beta^{(m)} &= \frac{\sum_i \sigma_i}{C_m} = \frac{\sum_{p \in \mathcal{P}} \sum_{f \in p} \sigma_f}{C_m} \\ \sum_{f \in p} \sigma_f &\leq \sum_{f \in p} k_p^{(m)} \rho_f \\ &\leq k_p^{(m)} \Gamma_p u_p \\ \beta^{(m)} &\leq \left(\frac{\sum_{p \in \mathcal{P}} k_p^{(m)} \Gamma_p u_p}{C_m} \right) \end{aligned}$$

Equation (6.9) follows from Equation (6.6). The next part of the proof involves showing that the recursion terminates. In the case where the paths are *feedforward*, it is clear that the recursion terminates at the ingresses where $k_p^{(m)}$ is bounded by k for each path. It remains to be shown that the recursion can be solved even in the presence of non-feedforward paths (Figure 6.1).

Definition 6.7.1. *Let path P_i traverse the multiplexers $M_i^{(0)} \ldots M_i^{(k)}$ in that order. A feedback association is said to exist among paths $P_1 \ldots P_n$ if:*

$$\exists\, l_j,\ j = 1 \ldots n,\ s.t.\ \{M_1^{(l_1)} M_1^{(l_1+1)} \ldots\ M_2^{(l_2)} M_2^{(l_2+1)} \ldots\ M_2^{(l_n)} \ldots\ M_1^{(l_1)}\} \tag{6.10}$$

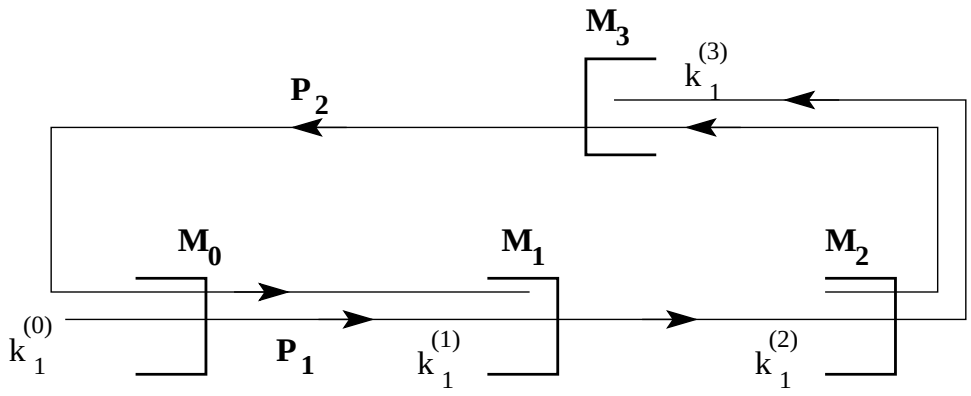


Figure 6.1: A simple network of multiplexers illustrating flows creating feedback. To compute the character of a flow on path P_1 after it exits M_0 we need information about flows on path P_2. But to specify flows on P_2 at the exit of M_3 we need to able to specify those on P_1 after they exit M_0. Thus there is a recursion.

forms a loop in the topology.

Let path i of length $l+1$ consist of the multiplexers $M_i^{(0)} \ldots M_i^{(l)}$ in that order and let the corresponding burst ratios be represented by $k_i^{(M_i^{(0)})}, \ldots k_i^{(M_i^{(l)})}$. At the start of each path, the burst ratio of every flow admitted on the path is fixed as part of the admission control regime. Thus $k_i^{(M_i^{(0)})}$ is known. However, to compute $k_i^{(M_i^{(1)})}$ we need information about $k_j^{(M_i^{(0)})}, \forall j$. Due to the presence of feedback some of these $k_j^{(M_i^{(0)})}$ values might themselves depend on $k_i^{(M_i^{(1)})}$. Thus we need to examine all the equations relating these quantities and solve them as a system. We thus have the following result which constructs the required linear system of equations.

Claim 6.7.1. *A feedback association with n paths involves n linear equations in n burst ratios $k_i^{(m)}$ so that the burst ratios for all multiplexers can be computed by solving the linear system.*

Proof of Claim

Our strategy will be to treat $k_i^{(M_i^{(1)})}, \forall i$ as the variables to solve for. We shall show that by obtaining these quantities, $k_i^{(M_i^{(r)})}, \forall i, r$ can be computed.

Clearly, when there are n paths, we have n unknown $k_i^{(M_i^{(1)})}$ values. It remains to be shown that there are n equations relating these quantities and that these equations can be solved.

From Equation (6.9) and Equation (6.6) we have,

$$k_p^{(M_p^{(1)})} \leq k_p^{(M_p^{(0)})} + \left(\frac{\sum_{q \in \mathcal{P}} k_q^{(M_p^{(0)})} \Gamma_q u_q}{C_{M_p^{(0)}}} \right) \tag{6.11}$$

Consider recursively expanding the terms $k_q^{(M_p^{(0)})}$ for each q in Equation (6.11). Note that due to the presence of feedback each $k_q^{(M_q^{(i)})}$ might depend on $k_r^{(M_q^{(i)})}$ for all paths r. Now, let the recursion stop whenever we obtain $k_q^{(M_q^{(0)})}$ or $k_q^{(M_q^{(1)})}$ for any q. That is, we do not expand the burst ratio if it is associated with the first or second multiplexer on that path. We can then have three types of terms on the right-hand-side of Equation (6.11):

1. $c_{q0} k_q^{(M_q^{(0)})}$ for a known constant c_{q0} and some path q
2. $c_{q1} k_q^{(M_q^{(1)})}$ for a known constant c_{q1} and some path q
3. $c_{p1} k_p^{(M_p^{(1)})}$ for a known constant c_{p1} and the path p under consideration

Thus we can write:

$$c_{p1} k_p^{(M_p^{(1)})} - \sum_q c_{q1} k_q^{(M_q^{(1)})} \leq \sum_q c_{q0} k_q^{(M_q^{(0)})} \tag{6.12}$$

But we obtained this equation by considering the burst ratios at the beginning of a path p. If we repeat the procedure for each path, we get n such equations. Consider these n equations similar to Equation (6.12) with equality. We see a system of n equations of the form:

$$\mathbf{Ax} = \mathbf{b}$$

With this, we now have the proof of Lemma 6.7.2. □

Note that, although we proved that there are n linear equations to solve for, we have not examined the existence of a unique solution to the system (i.e., we have not proved the matrix $\mathbf{A}$ to be non-singular). We are now ready to examine Theorem 6.7.1.

Proof of Theorem 6.7.1. In the following proof , we will drop the superscript (n) and use just σ_j to denote the burstiness at multiplexer n for flow j.

$$\begin{aligned}
\sum_j \rho_j &= \sum_{i=1}^{K} \sum_{j \in F_i} \rho_j \leq \sum_{i=1}^{K} u_i \Gamma_i \triangleq \mu_1 \\
\sum_j \sigma_j \rho_j &\leq \rho^{max} \sum_j \sigma_j = \rho^{max} \sum_{p=1}^{K} \sum_{i \in p} \sigma_i \\
&\leq \rho^{max} \sum_{p=1}^{K} k_p^{(n)} \Gamma_p u_p \triangleq \mu_2 \\
\sum_j \sigma_j^2 &= \sum_{p=1}^{K} \sum_{j \in p} \sigma_j^2 \\
&\leq \sum_{p=1}^{K} k_p^{(n)} \sum_{j \in p} \sigma_j \rho_j \\
&\leq \rho^{max} \sum_{p=1}^{K} (k_p^{(n)})^2 \Gamma_p u_p \triangleq \mu_3
\end{aligned}$$

□

The objective of obtaining these bounds is to evaluate the delay target at each multiplexer along a path. We can apply these results to any expression for $Pr\{D > d\}$ which depends on aggregate properties discussed in the preceding theorem. E.g., we apply these bounds to results obtained in [66] to obtain Equation (6.13) for a multiplexer M_i with paths $P_1 \ldots P_K$ and busy period bound T_i.

$Pr\{D_i > d_i\} \leq$

$$\max_{0 \leq \tau \leq T_i} \frac{1}{\sqrt{2\pi}} exp\left(-\frac{(C(\tau + d_i) - \tau \mu_1)^2}{2\tau \mu_2} \right) \tag{6.13}$$

We can use this expression to compute d_i given a probability of violation P_{vio}. Having demonstrated the tools to obtain d_i for a given topology, we now examine the admission control algorithm which can enforce the required constraints on input flows.

6.7.1 Admission Control Algorithm

Algorithm 4 Admission Control Algorithm

Input: (ρ, σ), leaky-bucket parameters of the flow being admitted
Input: Path P_i, Util. target u_i, Path Capacity Γ_i, Current path util v_i
if $\rho > \rho^{max}$ OR $\rho/\Gamma_i + v_i > u_i$ **then**
  Reject flow
end if
if $\sigma/\rho > k$ **then**
  Reject flow
end if
$v_i \leftarrow v_i + \rho/\Gamma_i$
Accept Flow

In order to ensure that the flows admitted into the network actually obtain the assurances described in §3.4 we need an admission control algorithm that enforces all the requirements. Briefly, the tasks of the admission control module would be to enforce the limits on burst ratios k_p, and to prevent violation of the utilization target. Algorithm 4 achieves these goals. The admission control module is assumed to be a central entity with information about the whole network. The time complexity of the algorithm is $O(1)$ in the number of admitted flows. The state requirement of the algorithm is *linear* in the number of admitted flows since we would have to keep a record of the admitted flows for use when they are removed from the network.

6.8 Delay Allocation

In the following paragraphs we consider the question of dividing a delay budget $D_b^{(i)}$ among the multiplexers along a path P_i. That is, given a limit on the edge-to-edge delay for path i as $D_b^{(i)}$, how should this delay be allocated among the hops in the path? The answer to this question will be useful if one needs to do per-hop admission control. In such a situation, one would evaluate the probability of violating a per-hop delay target.

We first examine why this should be a problem worth discussion. The burstiness of incident flows increases with increase in the number of hops of multiplexing. Consequently, the probability of violating a given delay target is higher for a multiplexer downstream along a path. Thus if equal delay targets are allocated to each

hop, the end-to-end probability of violation for the delay increases. Instead, if we obtain the delay targets by assigning a low constant probability of violation, P_{vio}, at each hop, we obtain an end-to-end assurance that is met with low probability of violation. That is, at multiplexer i, we set $Pr\{D_i > d_i\} \leq P_{vio}$ and obtain a delay target d_i for that hop. In this case, the delay targets are higher in multiplexers downstream. Also, since each hop has to satisfy a violation probability bound of P_{vio}, we require that $D_b^{(i)} \geq \sum_i d_i^*$.

Algorithm 5 Delay Allocation Algorithm

```
Input: Probability of violation P_vio
Input: Delay Budget for Path i, D_b^(i)
Input: Paths P_1 ... P_n

Compute d_i^* for each multiplexer
d_i ← d_i^*
for all Paths P_i do
  if D_b^(i) < Σ_{i:M_i∈P_i} d_i^* then
    Return Error
  end if
  S ← { M_i: P_i is the only path through M_i }
  |S| ← Number of elements in S
  for all M_i ∈ S do
    d_i ← d_i^* + (D_b^(i) − Σ_j d_j^*) / |S|
  end for
end for
Return vector of d_i values
```

We then propose a delay allocation scheme as follows. First, set a desired limit on probability of violation for a multiplexer, say, P_{vio}. Second, compute d_i^* for each multiplexer M_i by using Equation (6.13) or a similar expression using the results of §6.7. Allocate d_i^* to M_i. Distribute $(D_b^{(i)} - \sum_i d_i^*)$ equally among those multiplexers that have only path P_i passing through them. To appreciate the reason behind this condition, note that there might be multiple paths passing through a multiplexer. Increasing the delay target for this multiplexer might affect several paths. Algorithm 5 summarizes this procedure.

6.9 Results

The objective of this section is to evaluate a given topology for end-to-end delay assurances. Specifically, we apply the analytical results of §6.7 to compute the delay that can be assured *a priori* for a given probability of violation, P_{vio}. We compare the prediction with simulation results and examine the effect of topological character on the results. The simulation setup is described in §6.9.1. In §6.9.2 we examine numerical results to evaluate the delay allocation algorithm and some strategies for parameter setting. In §6.9.3 the algorithm is evaluated with an NS-2 simulation setup.

6.9.1 Simulation Setup

For all the experiments, two topologies were employed, viz., a synthetic *Transit-Stub* topology generated by GT-ITM [129] and a real (Telstra) ISP topology as discovered by *Rocketfuel* [109]. Table 6.1 provides the details for these topologies. A pruned version of the Telstra topology was used; due to space constraints only a schematic with nodes representing major cities is shown in Figure 6.2. A set of nodes were chosen as ingress-egress pairs. Traffic is destined to the egress from the ingress. The number of such pairs define the number of paths that were evaluated in one run of the simulation and that is indicated under the column *Paths* in Table 6.1. In the simulations, we have not considered paths with feedback; we intend to examine simulations for such scenarios as part of future work. In the experiments that follow, we set $(\rho^{max}, \sigma^{max}) = (0.2Mbps, 40kbits)$. We discuss some strategies to set the parameter values in §6.9.2. As indicated in Table 6.1, all links were assigned an identical bandwidth ($10Mbps$ in the case of GT-ITM, $20Mbps$ in the case of Telstra) and a propagation delay of $10ms$. The packet simulations were carried out in the Network Simulator NS-2 using trace-driven traffic sources[2].

We first present results for numerical experiments in §6.9.2 and then present the validation of the model through simulation in §6.9.3.

[2]http://www-tkn.ee.tu-berlin.de/research/trace/ltvt.html

Topology	Nodes	Links	Paths	Path Len	B/W
GT-ITM	100	193	20	7-13	10M
Rocketfuel (Telstra)	64	116	15	6-12	20M

Table 6.1: Topologies used for simulations

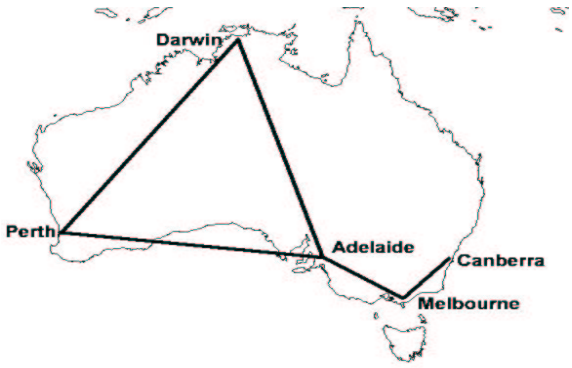


Figure 6.2: Telstra Network Topology (only a few nodes are shown here)

6.9.2 Numerical Experiments

In the following sections, we employ Theorem 6.7.1 and Equation 6.13 to arrive at end-to-end delay values.

6.9.2.1 Delay Allocation

While presenting the analysis in §6.7, we observed that an equal allocation of delay among the hops of a path is not a suitable choice. In order to verify the assertion, we choose a path in the GT-ITM topology. We set a delay budget of $D_b = \sum_i d_i^*$. Figure 6.3 shows the result of distributing D_b equally among the hops of a path as against setting d_i^* as the delay target for hop i. Clearly, the probability of violation is higher if we choose equal allocation for the same delay budget. The figure shows the result for a long path (10 hops) and a short path (6 hops).

6.9.2.2 Setting Parameter Values

For an analysis independent of traffic character, we assumed that the maximum admissible burst ratio for aggregates at the ingress and that utilization targets are known. The value of ρ^{max} has to be decided depending on the typical size of

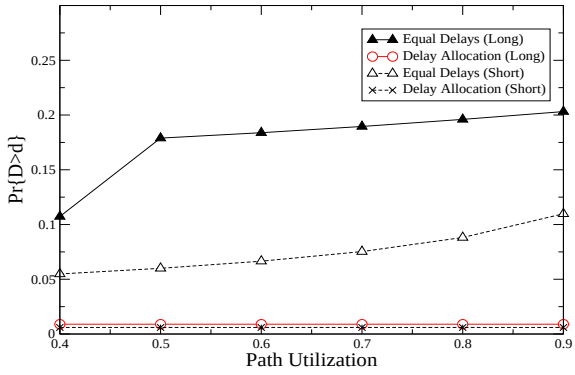


Figure 6.3: Equal delay budget allocation leads to high violation probability. Probability of violation increases with path length (hops).

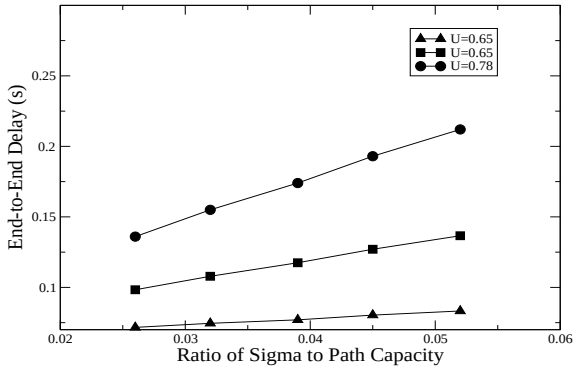


Figure 6.4: Predicted Delays for path from Canberra to Adelaide for increasing σ^{max} as a function of path utilization

aggregates that are admitted into the network. Given this value and for $P_{vio} = 0.001$ we consider choosing the utilization target and k_p (equivalently, $\sigma^{max} = \rho^{max} k_p$).

Consider a Path i with capacity Γ_i (as given in Definition 3.4.1). Figure 6.4 depicts the tradeoff that is inherent in a choice of utilization target with respect to the delay that can be assured. The path under consideration is from the Telstra topology, connecting Canberra to Adelaide. For a specific utilization target, we compute the end-to-end delay assurance as the ratio of σ^{max}/Γ_i increases. Thus if we have a requirement for a delay d', we could choose whether to set a higher utilization target or allow higher burstiness for flows.

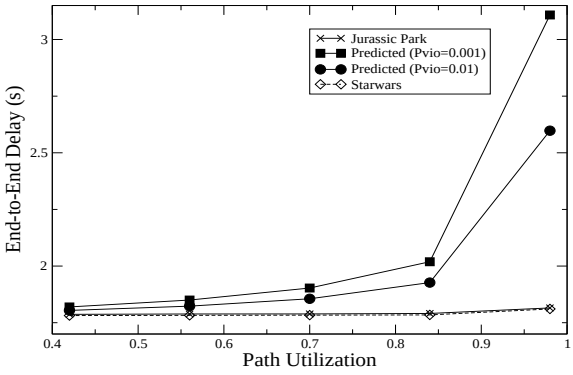


Figure 6.5: Predicted delay and simulation results for a GT-ITM generated topology with traffic from movie traces

6.9.3 Simulation Results

In order to verify the delay bounds we conducted simulation studies on the GT-ITM generated topology and the Telstra topology. In order to understand the topological significance to the results of the analysis, we consider the effects of path length and node degree (number of links incident at a node). In the succeeding sections, the 95% confidence intervals were too small for the scale of the graph and hence are not shown.

6.9.3.1 Accuracy of the bounds

Figures 6.5 and 6.6 show the comparison between predicted delays and the measured delays in simulation. We consider results for two trace driven simulations (Starwars and Jurassic Park) for a path in GT-ITM topology and one in Telstra (Canberra to Darwin). The X-axis denotes the path utilization limit enforced by the admission control algorithm and does not reflect measured utilization.

For lower utilization values, the simulation results are close to the predicted delays. As we increase the utilization factor, the prediction diverges and becomes quite conservative for utilization greater than 0.75. This is due to the fact that the analysis considers a worst-case scenario for each flow. Since in reality, a flow might not always find a queue due to the burst parameters of other flows, the predicted delays are conservative. However, note that P_{vio} allows us to choose how conservative the provisioning can be. Higher the value of P_{vio}, lower the delay targets.

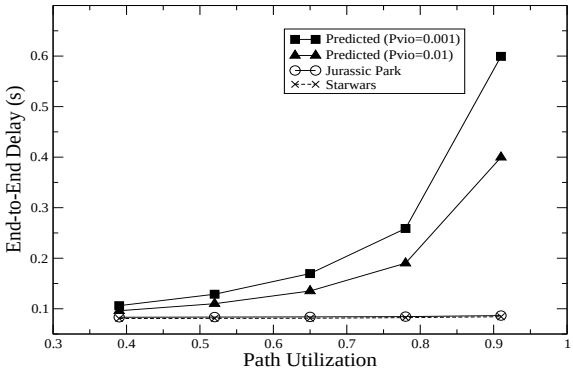


Figure 6.6: Predicted and Simulated Delay between Canberra and Darwin

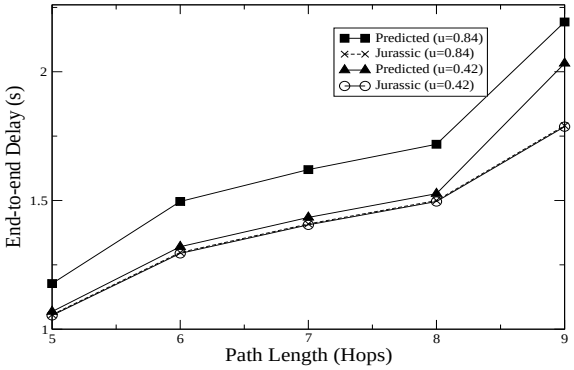


Figure 6.7: Increase in End-to-end delay bound with increasing path length (GT-ITM)

6.9.3.2 Effect of path length

One of the most important aspects of multiplexing across multiple hops is the potential increase in burstiness of the traffic. For the same path capacity one would expect that the delays assured are much higher for a longer path. We examine this scenario in Figure 6.7. For shorter path lengths, the delay bound is fairly accurate. As the path length increases, the delay bound becomes more conservative. For very long path lengths, the bound may be too high to be useful as an assurance. However, as we see in this experiment path lengths of up to 8 hops were fairly accurately modeled.

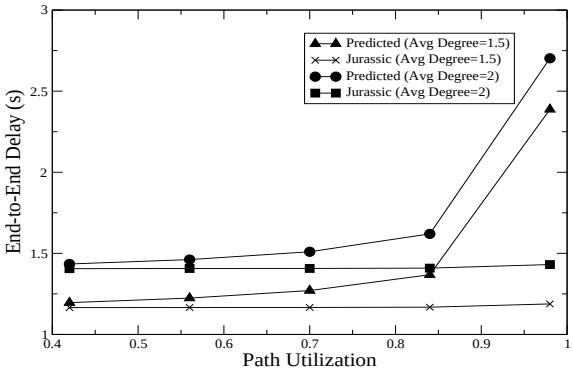


Figure 6.8: Increase in End-to-end delay bound with increasing average node degree (GT-ITM)

6.9.3.3 Effect of node degree

Another interesting aspect of the topology that affects the delay bounds is the degree of a node. The number of links feeding into a multiplexer is referred to as the degree. To understand why this is important, note that a multiplexer which is fed with multiple paths will probably have a higher number of flows. Intuitively, the probability of delay violation is higher when there are more number of flows with comparable burst parameters.

To illustrate this, we choose two paths in the GT-ITM topology with the *same length* (number of hops) and *capacity*. The paths are distinguished by their *average degree*, i.e., the sum of the degrees of each multiplexer on the path divided by the path length. Figure 6.8 shows the results of the experiment with paths from the GT-ITM topology. For the path with higher average node degree we observe that the measured delay in simulation and the predicted delays are higher. Thus while evaluating a path for delay assurances, the average node degree is an important measure.

6.10 Conclusions

The problem of delay assurances in FIFO networks is an important one. In order to measure the delays that can be assured by a given network topology, we must be able to set a delay target for each hop, that can be met at a given probability level. We provided analytical tools to arrive at delay targets for a FIFO network

independent of exact traffic description, given some restrictions on the nature of flows that can be admitted. We demonstrated an admission control algorithm that is scalable and does not require per-hop signaling, to enforce these assumptions.

In order to evaluate the analytical results we considered a generated topology and a real Internet topology. We found that, for utilization targets in the range of $0.4 - 0.75$, the model provides good delay estimates. For higher utilization targets, the worst-case nature of the analysis leads to conservative predictions. We evaluated the capabilities of the chosen topologies for a selected set of paths. The dependence of end-to-end delay bounds on topological features such as degree and path length was explored. Given a FIFO network topology, we can provide end-to-end delay assurances for selected paths, under limitations on path length and degree. Future work will involve improving the path capacity computation algorithm and accuracy of burstiness bounds.

CHAPTER 7
Tradeoffs in Edge-based Resource Allocation

7.1 Summary

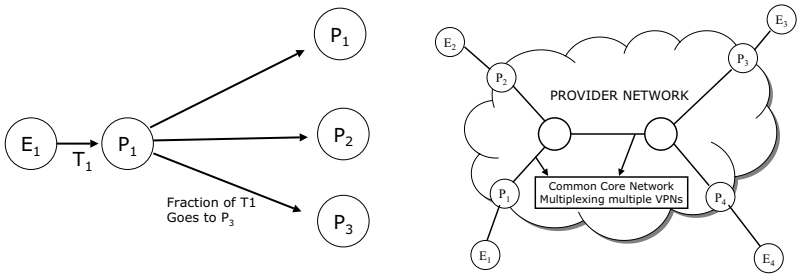
In the preceding chapters, we specified an edge-based QoS framework which is based on the choice of simpler core network mechanisms and sophisticated edge mechanisms. While the advantage of deployment ease has already been articulated, we have not yet examined the cost of forgoing core network based mechanisms. In this chapter we conduct a simulation study to examine the cost of opting an edge-based architecture. We examine means to reduce this cost while retaining the advantages of a deployable architecture. The contents of the chapter are organized as below:

- Examining the various design choices available to a provider
- Evaluation of the relative cost of these choices
- Simulations to examine viable choices for typical VPN service scenarios
- Conclusions on trade-offs involved in design process

7.2 Introduction

A Virtual Private Network (VPN) securely connects multiple customer sites that are possibly geographically spread out and wish to communicate among each other. Frequently, such a network provides a pre-specified Quality of Service assurance (a Service Level Agreement - SLA) in the form of expected loss rates and delays. A service provider provisions the network to ensure that the SLAs for an admitted VPN are met based on information provided by the VPN customer.

With appropriate admission control mechanisms at the entry of the network, the provider can meet the QoS requirements. A typical admission control test involves deciding whether to admit a new flow into the network. The decision depends on whether existing contracts are violated, in which case the new flow



(a) (b)

Figure 7.1: Admission Decisions involve point-to-multipoint traffic aggregates. Here the aggregate T_1 from E_1 is split among egresses P_1 to P_3

cannot be admitted. When admitting a new VPN, the admission criterion has to account for *traffic aggregates* that will be introduced from all sites of the new VPN customer into the network. In this sense it involves multiple steps, each of which resembles a traditional admission control problem [68]. But unlike the problem of admitting a new flow onto a link, one has to deal with *point-to-multipoint* nature of the traffic from each customer site.

Consider the example where we need to decide whether to admit the VPN with endpoints E_1, E_2, E_3, E_4, as shown in Figure (7.1a). The provider edge routers corresponding to these endpoints are denoted as P_1, P_2, P_3, P_4. The traffic aggregate emanating from the network at E_1 possibly contains traffic toward E_2, E_3 and E_4. Consider the admission decision for the aggregate T_1 as depicted in Figure (7.1b). There are two pieces of information that an admission control entity (like the one discussed in Chapter 4) needs here:

1. A traffic matrix that provides statistics about traffic exchanged between E_1 and any of the other endpoints.

2. The capacity available between P_1 and any of the other network edges through which the customer endpoints are reached.

In an ideal situation, the customer traffic is perfectly characterized so that we have a traffic matrix specifying the amount of traffic that is directed toward each of the other endpoints. Further, the network would support per-hop signaling-based admission control so that one has a precise idea of the capacity available to a given

endpoint. However, neither of these pieces of information are easily available in a real situation. It is usually hard to obtain the customer's traffic matrix because it is often unknown even to the customer. Further, today's core networks do not support per-hop admission control functions. The question then becomes, what is the relative importance of these components and what mechanisms can help a provider go beyond a naive peak provisioning approach while still being relevant from a deployment perspective. The service provider would naturally want to exploit the multiplexing gains offered by the temporal and spatial variability in the traffic generated by the endpoints of VPNs in the network. There are two levels of multiplexing that can be taken advantage of:

1. multiplexing of traffic from the endpoints of a given VPN sharing a part of the network

2. multiplexing of traffic from different VPNs sharing the network

Further, one would want to know if there are characteristics specific to VPN structure which can be exploited to supplement the lack of information or network mechanisms.

Typically, VPNs feature distinct structural characteristics. E.g., many VPNs fall in the category of a hub-and-spoke VPN. In such a VPN, there is a customer site which acts as the hub for all communication in the network and traffic from other sites in the network is primarily to and from this hub site. Clearly, while provisioning such a VPN the admission control mechanism needs to primarily consider the path to the hub site. Another example where VPN structure plays a simplifying role is when there are clusters of customer sites in and around a geographic area with most of the traffic staying within the cluster. On the other hand, meshed structures where all sites interact with each other to varying degrees, imply the need for sophisticated provisioning mechanisms.

Thus, while choosing admission control strategies it is useful to consider the nature of VPNs being considered. In this chapter we attempt to quantify the tradeoffs in choosing among different admission control strategies. Depending on the customer traffic information and network mechanisms at one's disposal, our results

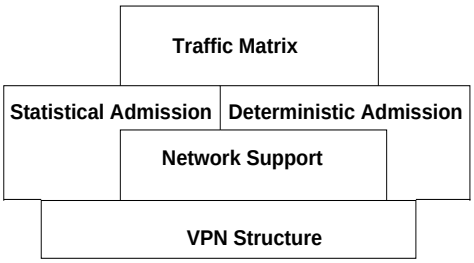


Figure 7.2: The important parameters influencing design choices and the interesting combinations are depicted. E.g., one could build a statistical admission mechanism with network support in terms of signaling but without traffic matrix information (as Hose Model does)

help in making a decision regarding what admission control mechanisms to opt for. Specifically, we consider the following issues:

- How important is network support (signaling and per-hop admission control) in terms of gains in resource utilization?
- What is the role of traffic matrix information - what are the scenarios where having pair-wise information yields noticeable gains?
- How does VPN structure affect the choice of the admission control scheme?

In the rest of the chapter, we study these issues by observing performance changes in the presence and absence of the relevant components. We first describe the various design choices and detail how we model them in a simulation framework (§7.3). The framework is then employed to conduct a comparative analysis (§7.4).

7.3 Parameters of Interest

We begin our study with a discussion of the parameters that affect scalability and achievable resource utilization gains. Figure (7.2) summarizes the various parameters at play. The parameter stack depicts the combinations of variables that a given solution can take into account. While provisioning a VPN, one could choose to perform a simple deterministic admission control while relying on signaling ("Network Support") and exploiting the structure of the VPN to enhance gains. Any

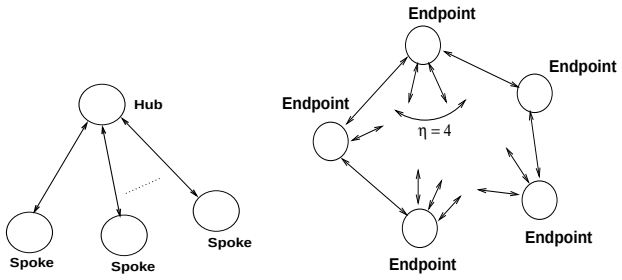


Figure 7.3: (a) A Hub/Spoke VPN; (b) A VPN with each endpoint communicating with multiple endpoints, η is the maximum number of endpoints with which a given endpoint communicates

evaluation of the importance of these parameters requires their clear definition. In the succeeding paragraphs, we describe how we quantify these variables.

7.3.1 VPN Structure

A commonly occurring structure in VPNs is the "Hub/Spoke" model. As seen in Figure (7.3a), there is a central "Hub" node with which every other endpoint communicates. As far as the provider network is concerned the capacity requirements are between a given spoke and the hub.

We cover other structures using a generic model (seen in Figure (7.3b)). This model features a parameter η indicating the maximum number of endpoints with which an endpoint communicates in the VPN. In Figure (7.3b) one of the endpoints has four arrows leading from it indicating its communicating peers. While studying the impact of structure of VPNs on admission control strategies, we vary: a) the percentage of VPNs that belong to each category and b) the value of η for generic VPNs.

7.3.2 Admission Control

Given parameters characterizing traffic, the provider has to make a decision of whether to admit this new customer. In the succeeding sections we use the dual leaky-bucket specification to describe the traffic from a customer endpoint. The dual leaky-bucket consists of three parameters (π, ρ, σ) indicating respectively the peak rate, sustainable average rate (leaky-bucket rate) and the burst parameter (bucket

size). If the arrival process is indicated by $A(t)$, conformance to (π, ρ, σ) means $A(t, t+\tau) \leq min(\pi\tau, \rho\tau + \sigma)$. Thus for an endpoint i, the traffic specification is given by $(\pi_i, \rho_i, \sigma_i)$. With this data, one could opt for either a probabilistic admission test or a deterministic one.

7.3.2.1 Statistical Admission Control

Statistical admission schemes typically evaluate the probability of violating a given QoS metric (please see §6.3 for a review of related work). In the current context we are interested in evaluating the utility of deploying a statistical scheme with reference to other parameters like signaling and traffic matrix information. In order to conduct such a study, we need to examine both per-hop admission control and edge-based admission control. To recall our discussion in chapters 5 and 6, multiplexing flows onto a single queue distorts the statistical characteristics and complicates mathematical analysis. Since the goal of this exercise is to evaluate statistical schemes in presence of other network mechanisms and not to examine deployment concerns, we opt for a framework that avoids the traffic distortions due to per-hop multiplexing. Such a framework is provided by the bufferless multiplexing proposal due to Rajagopal et al [101].

With bufferless multiplexing, the statistical properties of flows are not altered at intermediate hops. If the flows entering a network are statistically independent, they remain so throughout their transit path in the network. This framework lends itself to elegant mathematical formulation and there are closed form expressions for loss probabilities based on large-deviations theory. We adapt these results for our model where flows are characterized by dual leaky bucket shapers. The worst case loss probability for a flow with peak rate π_i at a multiplexer with capacity C is given by:

$$\frac{1}{Cs^{*^2}\sqrt{2\pi\mu''_U(s^*)}} e^{-s^*(C-\pi_i)+\mu_U(s^*)} \tag{7.1}$$

where s^* is the unique solution to

$$\mu'_U(s^*) = C - \pi_i \tag{7.2}$$

and for the set of flows I incident at the multiplexer, μ_U is defined as:

$$\begin{aligned}
\mu_U(s) &= \sum_{j \in I - \{i\}} \mu_{U_j(s)} \\
\mu_{U_j(s)} &:= \log E[e^{sU_j}] \\
U_j &= \begin{cases} \pi_j & \text{with probability } \frac{\rho_j}{\pi_j} \\ 0 & \text{with probability } 1 - \frac{\rho_j}{\pi_j} \end{cases}
\end{aligned}$$

7.3.2.2 Deterministic Admission Control

A simple strategy in admitting a new flow is to quantify its peak bandwidth requirements and reserve that amount inside the network. The first option is to reserve the peak rate specified by the customer. In the presence of more elaborate traffic matrix information, e.g., mean and variance of expected load on a source-destination pair basis, we could enhance this scheme. The reservation could then be equal to the mean in addition to a multiple of the standard deviation.

7.3.3 Signaling

In the presence of network support, admission control and bandwidth reservation decisions can incorporate information from each hop of a path along which a flow is admitted. Given the traffic characteristics at each hop, we could either do statistical or deterministic admission control and exploit traffic matrix information if it is available. Such a framework allows high resource utilization but relies on a lot information and support from both the customers and the network.

While the resource gains are desirable, we would certainly wish to relax the amount of network support required. In chapters 3 and 4 we described edge-based mechanisms to substitute the functions of signaling. We briefly recall those ideas here and refer the reader to §3.4 for more details. Signaling-based reservation protocols help prevent over-booking capacity in the network by accounting for existing commitments at each hop. By building the notion of an edge-to-edge virtual path with an associated capacity, we can achieve the same effect. The knowledge edge-to-edge path capacity allows us to conduct admission control for a virtual link connecting the ingress to the egress. We employ this notion to quantify the value

added with signaling in §7.4.4.

In §3.4 we arrived at path capacity using a static apportioning scheme (Algorithm 6). The disadvantage with such a scheme is that it treats all paths equally while in reality, some paths are more trafficked than others. We examine a measurement-based path capacity algorithm in §7.5 that is adaptive and remarkably improves the static scheme.

Algorithm 6 Path Capacity of path p

Denote capacity of link l as C_l and that of path p as C_p
Input: $\mathcal{L}_p \leftarrow$ { Set of links in p }
Input: $\mathcal{P}_l \leftarrow$ { Set of paths traversing link l }
for each link $l \in \mathcal{L}_p$ **do**
  $|P_l| \leftarrow$ Number of paths traversing link l
  $S_p(l) \leftarrow \frac{C_l}{|P_l|}$
end for
$C_p \leftarrow \min_{l \in \mathcal{L}_p} S_p(l)$

7.3.4 Traffic Matrix Information

The final parameter we shall introduce is the customer traffic matrix. The traffic matrix specifies information about expected traffic between a given pair of nodes.

Typically, the traffic originating from a source node is split among a set of egresses (see Figure (7.1)). If there is information about per-destination traffic trends, the allocation can be tailored accordingly. If the source rarely directs traffic at peak rate toward a single destination, there can be multiplexing gains compared to peak provisioning. Information about pair-wise traffic trends could either be specified by the customer or some measurement-based mechanism may be used to learn these trends. In the absence of such pair-wise information about the traffic matrix, there would be some over-provisioning. But the upside would be a simpler framework.

As introduced in §7.3.2, we employ (π, ρ, σ) specification to describe aggregate traffic. The traffic matrix could then be specified either as:

1. A set of triples $(\pi_j, \rho_j, \sigma_j)$ governing the traffic toward endpoint j so that

$\sum_j \rho_j = \rho$, $\sum_j \sigma_j = \sigma$ and $\pi_j \leq \pi$ or,

2. A set of mean and variance values (m_j, v_j) for random variables $p_j \in [0,1]$ which represent the fraction of the aggregate directed toward destination j. Thus if A is the aggregate traffic and A_j is toward destination j, we have $A_j = p_j A$ and $A = \sum_j A_j$

We choose the second option since it is a more intuitive description (e.g., 70% of the aggregate is directed toward destination 1) and leads to convenient implementation. As seen below, it can be shown that a dual leaky-bucket description can be deduced from (m_j, v_j) and (π, ρ, σ).

We are given that the aggregate A conforms to (π, ρ, σ) and that (m_j, v_j) are the mean and variance values of the fraction of traffic directed toward destination j. If the long-term average of the aggregate is given by ρ, its variance is bounded by $\pi\rho - \rho^2$ (Chapter 4). Thus we can obtain the mean and variance of A_j as follows:

$$\begin{aligned} E\{A_j\} &= E\{p_j A\} \\ &= m_j \rho \\ Var\{A_j\} &= E\{A_j^2\} - (E\{A_j\})^2 \\ &\leq m_j \rho(\pi(\frac{v_j}{m_j} + m_j) - m_j \rho) \end{aligned} \tag{7.3}$$

Thus a dual leaky-bucket specified as:

$$\begin{aligned} \pi_j &= \pi(\frac{v_j}{m_j} + m_j) \\ \sigma_j &= \sigma \\ \rho_j &= m_j \rho \end{aligned}$$

represents the per-destination aggregate whose characteristics are described by p_j (the random variable denoting fraction of traffic toward j).

In summary we presented the important variables that decide resource utilization gains while provisioning VPNs.

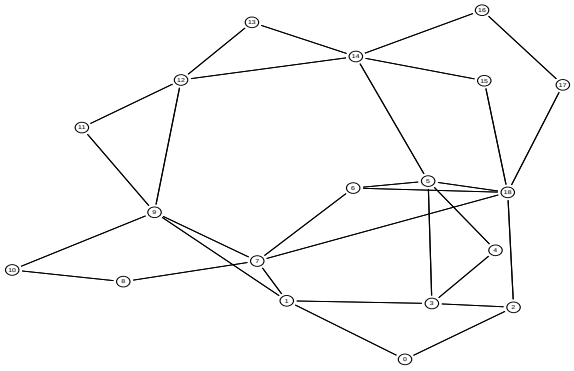


Figure 7.4: The MCI topology was used in the experiments. Link capacities were set to 100 $Mbps$ **and propagation delay was set to** 10 ms

7.4 Comparative Analysis

In the following paragraphs we proceed by considering different interesting combinations of the the parameters described in the previous sections. We begin by specifying details about the topology used and the methodology for generating VPNs.

7.4.1 Topology and experimental setup

We use the old MCI backbone topology (shown in Fig. 7.4) for our experiments. Each experiment consists of two phases - a VPN generation phase and an admission control phase. The experiment is started with a set of values for the structure of VPNs to be generated and dual-leaky-bucket parameters for the aggregate traffic from an endpoint. The VPN generation routine then produces VPNs of varying sizes. These VPNs are the fed to the admission control routine one after another. To generate a VPN endpoint's characteristics randomly the following procedure was followed. Every endpoint in the VPN has a set of destinations with which it communicates. The procedure involves picking a random subset of endpoints and deciding the fraction of traffic that is destined to each destination. E.g., for a destination set with 4 nodes, three uniform random numbers, $r_i, i = 2 \ldots 4$ are generated in the range $[min, max]$. Then setting $r_1 = 1$ and $\sum_i r_i m_1 = 1$ we obtain $m_i = r_i m_1$. The variance of per-destination traffic fraction v_j is then computed as a fixed fraction of the mean.

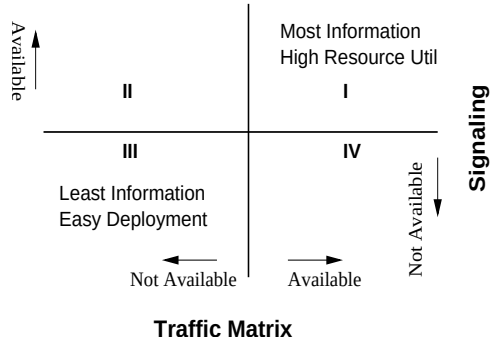


Figure 7.5: Higher resource utilization can be achieved by requiring more information and network support

The range $[min, max]$ decides the *bias* toward a subset of destinations in the set. If the range is small and around 1, traffic is equably directed to all nodes in the set. Higher the value of max greater the spread of the load distribution among destinations. The dual-leaky-bucket regulator parameters for all VPNs was set at ($0.5\ Mbps$, $0.15\ Mbps$, $20\ kb$). The link capacities were set to $100\ Mbps$ and their delay was chosen to be $10\ ms$.

Although we are employing a specific topology for our experiments the manner in which the VPNs have been generated allows us to draw sufficiently general conclusions. The size of the VPN bandwidth demands is very small compared to the link capacities in the core of the topology. This implies that observations regarding statistical multiplexing gains are largely independent of this particular core network topology.

7.4.2 Experiment Roadmap

In the following sections we intend to study the role of the parameters introduced in §7.3. We examine the change in number of VPNs admitted when the parameter is used in the admission decision process.

The schematic in Figure (7.5) summarizes the various options at a designer's disposal. Clearly, the maximum resource utilization is achieved when there is signaling support from the network for per-hop admission control and there is traffic matrix information. The trade-off is implementation complexity. The goal of a designer would be to achieve performance close to that offered by Quadrant I while

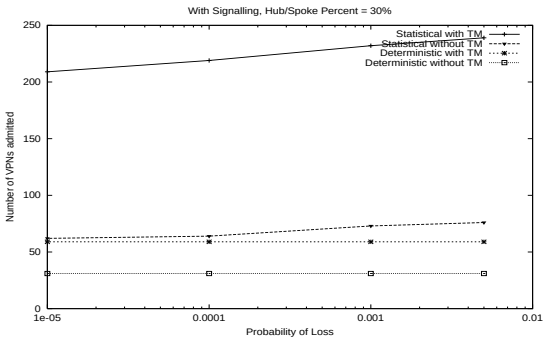


Figure 7.6: Number of Admitted VPNs in the presence of signaling-based per-hop admission control with 30% of the generated VPNs being of the Hub/Spoke type

incurring implementation costs that are typical of Quadrant III. That is, one needs to build mechanisms which approximate the performance provided by more information and network support while remaining inexpensive to implement.

The structure of the VPN is a parameter in addition to the two dimensions noted in Figure (7.5). In the rest of the subsections we first treat Quadrants I and II and then examine III and IV.

7.4.3 Traffic Matrix

In §7.3.4 we specified traffic matrix information in terms of the mean and variance of the random variable representing the per-destination traffic fraction. Thus, if A were the aggregate traffic from an endpoint and A_j were the traffic directed toward destination j, we introduced a random variable $p_j \in [0, 1]$ so that $A_j = p_j A$.

In the presence of traffic matrix information, the admission control decision for a link need only account for the fraction of traffic that is likely to be directed along this link. In particular, we evaluate the admission criterion considering a dual leaky-bucket specification derived using (m_j, v_j) and (π, ρ, σ). In the absence of such information, the admission decision assumes (π, ρ, σ) as the specification of traffic toward every destination.

In discussions in the rest of this subsection we retain signaling-based admission control and concentrate only traffic matrix. The plots in Figures (7.6) and (7.7) show the benefits of collecting traffic matrix information. The salient points to be noted

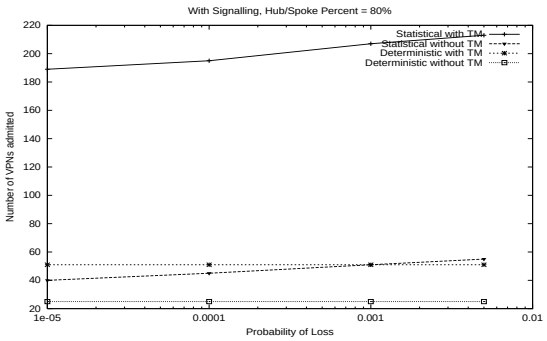


Figure 7.7: Number of Admitted VPNs in the presence of signaling-based per-hop admission control with 80% of the generated VPNs being of the Hub/Spoke type

are:

- If the VPN structure is exploited, importance of traffic matrix information reduces as the fraction of VPNs of the Hub/Spoke type grows (the gap between the curve at the top and the ones at the middle reduces). This is intuitively clear since knowing that a VPN is a hub/spoke VPN implies that we know most of its traffic matrix.

- A statistical admission control algorithm without traffic matrix information does as well as a conservative admission control scheme that exploits traffic matrix information. Intuitively, with more information about traffic matrix and VPN structure, the admission scheme can be simpler for the same resource utilization gain.

Thus if a majority of the VPNs being serviced are of the Hub/Spoke nature and no traffic matrix information is available, a simple deterministic admission can be a good choice if it can be enhanced with information about what nodes are spokes and which node is a hub. Figure (7.8) illustrates this reduction in gain of a statistical scheme over a deterministic one, if there is no traffic matrix information.

7.4.4 Signaling-based admission

In §7.4.3 we retained signaling-based admission control and examined the role of traffic matrix information. We now consider the gains of having signaling-based admission control, with respect to availability of traffic matrix information.

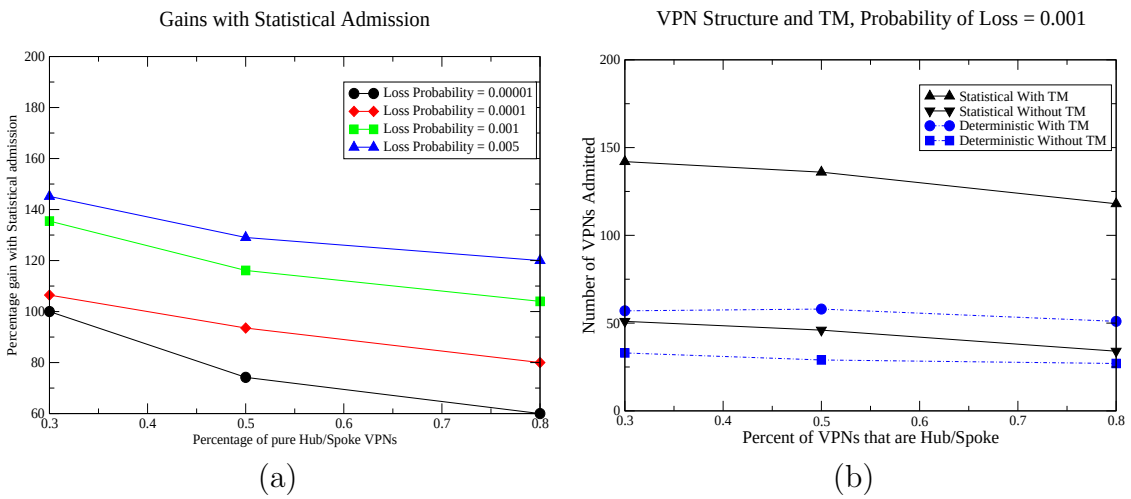


Figure 7.8: The utility of statistical admission control reduces with higher number of Hub/Spoke VPNs

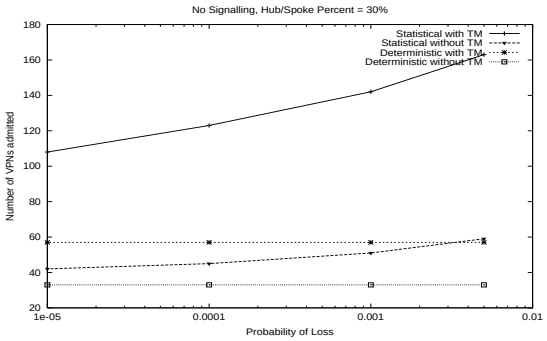


Figure 7.9: Number of admitted VPNs falls in the absence of signaling-based admission control (percentage Hub/Spoke VPNs = 30%)

In the absence of signaling-based admission control, we have to make admission control decisions at the entry of the network. In §7.3.3 we discussed the options available in the absence of signaling. We proposed a simple and static path capacity computation algorithm so that admission control decisions are made consider the ingress-to-egress path as a virtual link with capacity derived from Algorithm 6. Here we employ this strategy to evaluate the value added by signaling-based admission mechanisms.

Figure (7.9) shows the result of such an edge-based strategy for different loss probabilities. Comparing with Figure (7.6) we can clearly see the reduction in the number of admitted VPNs. The plot in Figure (7.10) confirms this inference and

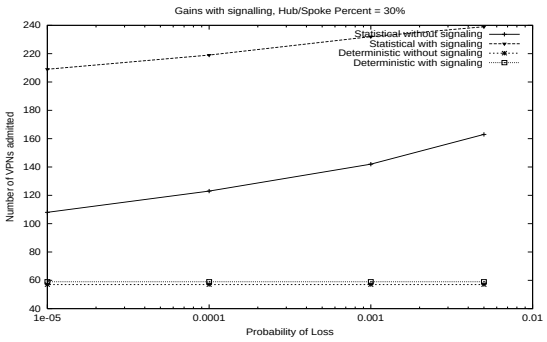


Figure 7.10: Signaling gains in the presence of traffic matrix information

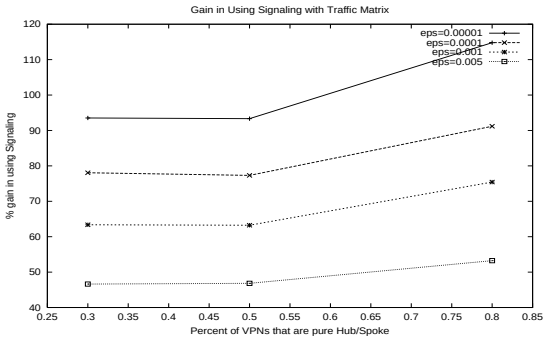


Figure 7.11: Signaling-based admission control is superior irrespective of the percentage of Hub/Spoke VPNs

shows the gains with signaling. Figures (7.11) and (7.12) present this aspect across varying nature of generated VPNs. The following observations are in order:

- The trends indicate that signaling yields consistent gains irrespective of the structure constitution of VPNs and the availability of traffic matrix.
- While the path capacity algorithm is simple and enables edge-based admission decisions, it does not perform as well as an algorithm that exploits signaling.

We would certainly desire to retain the simplicity of the edge-based admission scheme while obtaining performance comparable to the signaling-based mechanism. In §7.5 we examine strategies to bridge the gap in performance via an improved algorithm.

7.4.5 Effect of structure of VPNs

Thus far the structural characteristics of VPNs have been captured in terms of the percentage of VPNs that are of hub/spoke type. In §7.3.1 we introduced a

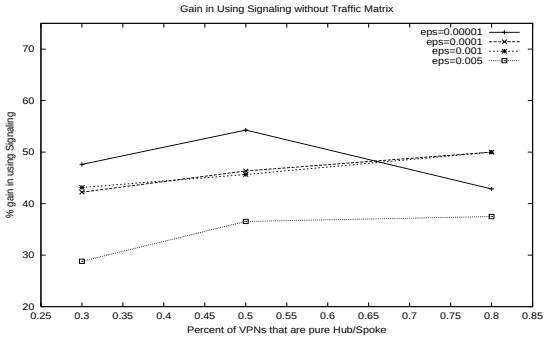


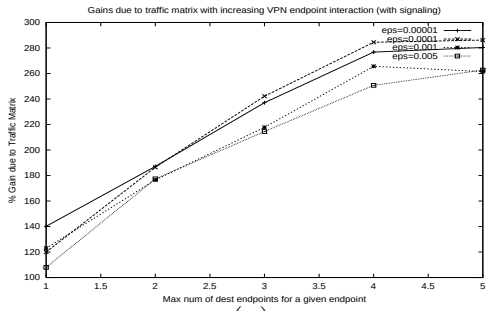
Figure 7.12: Even in the absence of traffic matrix information Signaling-based admission control is superior irrespective of the percentage of Hub/Spoke VPNs

parameter η to represent the maximum number of destinations that an endpoint communicates with. A VPN with higher η has more complex interactions among its endpoints. With reference to the traffic emanating from an endpoint, the fraction of traffic directed to a destination can vary more. Thus one would expect that understanding the traffic matrix when η is higher would yield resource utilization gains. The experiments in this section support this intuition.

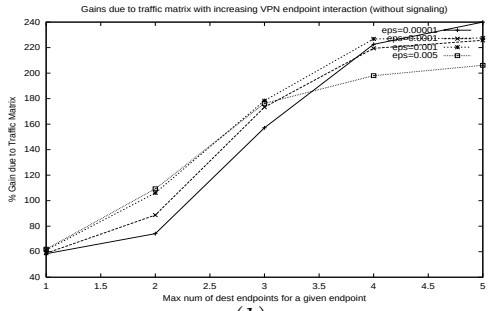
Figure (7.13a) and Figure (7.13b) depict the importance of traffic matrix with increasing complexity in VPN endpoint interactions. The service provider can get a significant benefit in terms of resource utilization (due to the ability to admit a larger number of VPNs) by taking advantage of the traffic matrix characteristics. This is particularly true as we go away from a simple hub and spoke VPN structure (when $\eta = 1$) to a VPN with more peer-to-peer communication Similarly signaling gains (Figure (7.14a) and Figure (7.14b)) become significant with higher η when there is no traffic matrix information.

The results presented till now confirmed and quantified the intuition that complicated VPN structures imply significant costs in resource allocation if we do not take advantage of the benefits of the traffic matrix or signaling. Fortunately, we can devise strategies to exploit such information without elaborate changes to the network:

1. Recent research (e.g.,[133]) has led to efficient means of estimating large traffic matrices using long-term SNMP link statistics.



(a)



(b)

Figure 7.13: With increase in the number of endpoints with which a node communicates (higher value of η) gains due to traffic matrix become more pronounced.

2. Improved algorithms to manage edge-based path capacity allocations dynamically (as discussed in §7.3.3) can lead to performance that compares well with signaling-based admission control schemes.

We elaborate on the second point in the following section by improving the path capacity algorithm presented in Algorithm 6. In summary, we examined the different parameters that affect resource utilization and quantified their importance. We found that learning more about the nature of VPNs to be served (e.g., are they hub and spoke) allows us to exploit attractive trade-offs (e.g., simple deterministic schemes in presence of majority hub and spoke type VPNs).

7.5 Dynamic Path Capacity

In §3.4 we introduced a simple path capacity computation algorithm in order to substitute the function of signaling-based admission. The algorithm statically

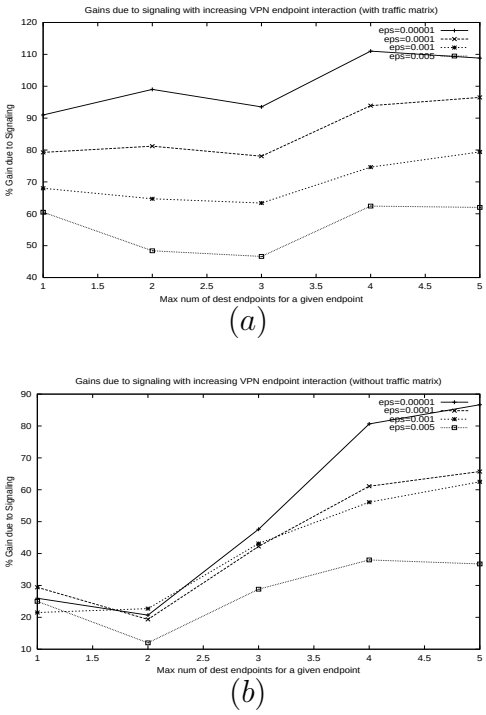


Figure 7.14: (a) With traffic matrix available, gains due to signaling hold steady across multiple values of η; (b) In the absence of traffic matrix, signaling gains become significant as η grows

divided link capacities among different source-destination paths traversing the link. There are some notable disadvantages to this algorithm:

1. It does not consider the fact that some paths carry more traffic than others. A static link sharing scheme does not reassign bandwidth to other paths which might be seeing higher demand.

2. It assumes that routing and topology are fixed. The capacities are computed assuming that the links along a source-destination path are known.

3. It requires the network edge to process routing control information and compute path details.

In this section we attempt to remedy these drawbacks and describe an improved algorithm. In doing so, we increase the resource allocation gains while avoiding signaling-based mechanisms.

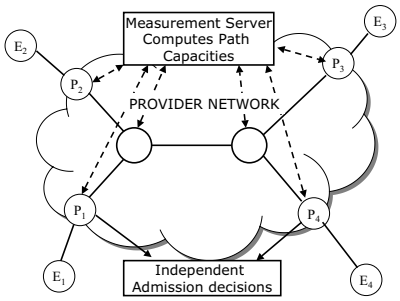


Figure 7.15: The network edges can be decoupled from routing and topology changes if they communicate a central measurement server which provides path capacity information

7.5.1 Distributed Admission, Centralized Measurement

In order to remedy the drawbacks of the aforementioned algorithm, we envisage decoupling the functions of computing the path capacities and making the admission control decision. The former involves processing routing information for topology and capacity details. The latter involves computing an admission control criterion given traffic characteristics. Thus these are separable tasks. Figure (7.15) demonstrates such an architecture. A central "measurement server" receives routing updates so that it has a snapshot of the topology. It processes this data to compute path capacity values for all ingress-egress pairs. The network edge evaluates the admission test using the capacity information obtained from this central server.

The clear advantage with this setup is that routing and topology changes are shielded from the network edges which make admission decisions. The edge is periodically notified of the path capacity that is available toward any destination edge it might want to reach.

With this architecture in mind we devise an improved path capacity assignment algorithm (Algorithm 7). This algorithm uses a parameter $\beta \in [0, 1]$ which indicates the fraction of link capacities that are statically pre-assigned to each path according to Algorithm 6. Lower the value of β higher the flexibility to the allocation algorithm in assigning path capacities. If the value of β is too low, it might cause some network edges to refuse admission even when there is capacity available. Thus this parameter needs to be tuned to obtain acceptable behavior.

The algorithm at the network edge now becomes much simpler. As specified

Algorithm 7 Dynamic Path Capacity Computation

Precondition: Apportion $\beta C_l, \beta \in [0,1]$ equally among all ingress-egress pairs.
Input: Current routing and topology state
Input: Request from an edge for additional path capacity
Input: A parameter C^* representing a capacity increment.
if Unused capacity exists **then**
  Accept request and increase source-destination path capacity by C^*
end if
Return

in Algorithm 8 it evaluates the admission criterion to arrive at a probability of loss. If the probability is less than a pre-determined threshold ϵ, the request is admitted. Further, if the probability of loss is within a factor, α of the threshold (i.e., we are low on available capacity) it requests for additional capacity from the central measurement server. If the admission test fails, the edge requests the central server for more capacity.

Algorithm 8 Admission Control at an edge

Input: A point-to-multipoint service request from an endpoint E_1 toward a set of egresses P_1, P_2,. . . ,P_n
Input: Capacity available on path between E_1 and P_i obtained from Measurement Server
Input: Parameter $\alpha \in (0,1)$ that decides when a request for additional capacity is made.
for each path (E_1,P_i) **do**
  Compute probability of loss P_{loss} as part of admission criterion
  if $P_{loss} > \epsilon$ **then**
    Request measurement server for additional path capacity
    Reject admission request
    Return
  end if
  if $P_{loss} > \alpha\epsilon$ **then**
    Request measurement server for additional path capacity
  end if
end for
Accept admission request
Return

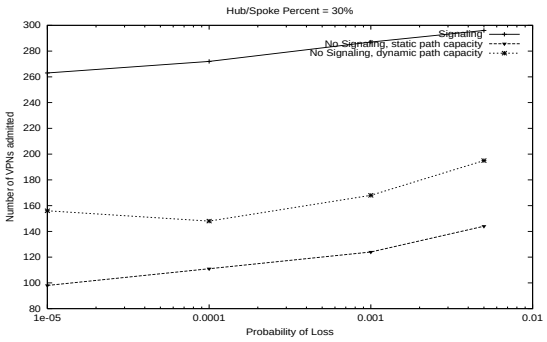


Figure 7.16: The Dynamic path capacity allocation considerably improves the performance of the static link sharing scheme

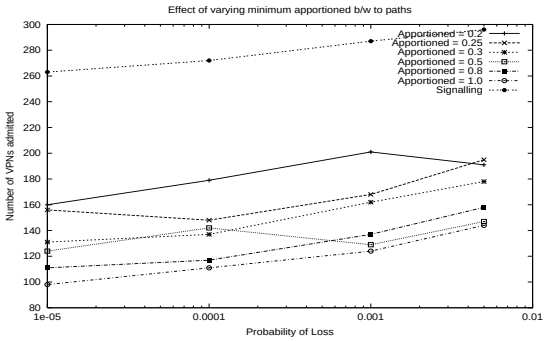


Figure 7.17: With lower values of β we have more flexibility in allocating path capacity where there is demand and hence more gain

7.5.2 Results

We now evaluate the algorithms presented in the previous section. Figure (7.16) demonstrates the gains in dynamically apportioning bandwidth versus a static algorithm. For these experiments we set $\alpha = 0.01$. Next we present the effect of varying β from 0.2 to 1.0 (equivalent to the static apportioning algorithm). As expected reducing β provides more flexibility in allocating path capacity and allows for higher number of admitted contracts.

In summary, the improved path capacity algorithm provides for higher gain and compares better with signaling-based mechanisms as compared to the static path sharing scheme.

7.6 Summary and Conclusions

In this paper we examined the parameters that influence resource allocation in Virtual Private Networks and quantified their role. We presented a set of variables that affect achievable resource utilization, namely, the admission control strategy, availability of traffic matrix, information about VPN structure and support for signaling based admission control. The various options with each of these variables was articulated.

Having laid down an evaluation framework to measure the relative importance of these parameters, we conducted experiments with the MCI backbone topology. We considered statistical and deterministic admission control strategies in a variety of scenarios. Our experiments to understand the interplay of these factors led to some important conclusions:

- When traffic matrix information is available it has a dominant effect on the resource utilization gains. As a consequence, structure of the VPN becomes important when it has an influence on the traffic matrix (e.g., knowing a VPN is of the Hub/Spoke type implies knowing most of the traffic matrix).

- Signaling-based admission control can vastly improve resource utilization. In the absence of signaling, we can use dynamic path capacity allocation algorithms to obtain comparable performance.

- With increasing complexity in the way endpoints in a VPN interact, importance of understanding the traffic matrix increases.

Thus it is important to estimate the traffic matrix for VPNs. In the absence of signaling-based admission control mechanisms it is advisable to build a dynamic path allocation architecture as described here. In the next chapter we examine techniques to obtain traffic matrices from SNMP measurement information.

CHAPTER 8
Traffic Matrix Estimation

8.1 Summary

Recognizing the importance of traffic matrices in realizing an adaptively provisioned point-to-set architecture, we explore techniques to estimate VPN traffic matrices. We find that existing techniques are not sufficient and develop new techniques in line with VPN measurement realities. The rest of the chapter progresses thus:

- Discussion of existing techniques
- Distinct characteristics of the VPN problem that require a different formulation.
- Validation of proposed algorithm
- A Study of VPN structure and temporal characteristics of traffic.
- Relevance of the information to efficient network operation.

8.2 Introduction

In the preceding chapters we formulated an edge-based model to build a VPN architecture that provides QoS assurances while allowing providers to exploit utilization gains by leveraging a common core network. The provider can honor SLAs while pursuing multiplexing gains only if the customer traffic profile and the state of the network is well understood. In the presence of accurate information about customer profile and available network resources, a provider can make accurate provisioning decisions while ensuring the network is never overloaded. The results of the cost-benefit analysis in Chapter 7 support the conclusion that with increasing complexity in customer iterations, the presence of traffic matrix information can mean significant resource utilization gains.

In addition to the Point-to-Set architecture, extant models like Hose Model rely on adaptive provisioning strategies that can be effectively deployed only if we understand factors that impact service quality and efficiency. Such factors include the structure of VPNs (the way various endpoints communicate with each other), traffic matrices and their variation over time. This entails continual measurement-based learning to fine-tune and guide provider decisions. Additionally, it is necessary to devise simple algorithms that leverage existing measurement infrastructure to achieve these goals.

In this chapter we present scalable techniques to infer traffic matrices and structure of VPNs using SNMP link-level measurements from a large IP/VPN service provider. Using these techniques we study the temporal and spatial properties of VPNs. We then place this information in the context of realizing an adaptive provisioning architecture as outlined in §7.5.

For provisioning the capacity of links in both the access as well as the core, the provider needs two pieces of information: a) a traffic matrix that provides statistics about traffic exchanged between any two customer endpoints (CEs); b) the capacity available between any two provider edges (PEs). Recent advances in traffic matrix estimation techniques [134] provide a starting point. We adapt the Entropy-based traffic matrix estimation techniques and exploit structural properties of VPNs to build a computationally feasible formulation. Given the structure and traffic matrix of VPNs on a provider network, the potential bandwidth demand due to existing VPNs can be deduced. Combining this information with customer-specified peak rate information for newly arriving VPNs, the provider obtains an estimate of available resources in the network and prepares for possible link resizing requirements.

Our study leads to the following important observations:

- We can deduce important structural and temporal characteristics of VPNs from simple SNMP-based measurement data.
- Both measurement trends and simulation results indicate that appreciable gains in provisioning can be obtained by exploiting the structure of VPNs and their traffic matrices.

- Temporal properties of traffic matrices indicate that adaptive provisioning of core capacity to suit customer demands is feasible and beneficial.

We outline a simple measurement-based framework to realize an adaptively provisioned VPN architecture that can exploit statistical characteristics of customer traffic.

We begin by analyzing measurement data to gain insight into properties of VPNs and present a traffic matrix estimation technique §8.4.

8.3 Related Work

Traffic matrix provides the volume of traffic between source-destination pairs in a network. Such matrices have been computed at varying levels of detail for IP networks [84]: between ISP Points-of-Presence (PoPs) [85], routers [133], IP prefixes [36] etc. The problem of estimating traffic matrices is ill-posed: for a network with N source-destination pairs we need N^2 demands to be estimated. However the number of pieces of information available is typically much lesser (the order of number of links in the network). For large N the problem becomes massively under-constrained. Such problems have been solved in many fields of engineering and science: seismology, astronomy, geophysics etc. [5, 23, 87, 93, 122]. These studies have indicated that some kind of *side information* must be brought in while solving such linear systems. Many such proposals solve the following minimization problem:

$$\min_{\mathbf{x}} ||\mathbf{y} - A\mathbf{x}||_2^2 + \lambda^2 J(\mathbf{x})$$

where $||.||_2$ denotes the L_2 norm, $\lambda > 0$ is a regularization parameter, and $J(\mathbf{x})$ is a penalization functional. These approaches are generally called *strategies for regularization of ill-posed problems*. The regularization strategy (the choice of $J(\mathbf{x})$) guides the optimization problem in its choice of the traffic matrix that might provide a good solution to the problem. One approach is to model the variables $\mathbf{x}$ as belonging to a known distribution and maximizing the posterior probability density $p(\mathbf{x}|\mathbf{y})$. This approach is known as the Bayesian regularization method.

Zhang et al [134] develop a regularization method tailored for traffic matrix

estimation. Their method incorporates the gravity model solution so that the optimization simultaneously attempts to minimize the error from observed link counts and gravity estimate. They demonstrate that the gravity model estimate for the traffic matrix provides a good starting point and hence propose to opt for the Kullback-Leibler divergence of the gravity estimate from $\mathbf{x}$ as the regularization functional. We shall specify this formulation and explain the underlying intuition in further detail in §8.4.3.

The problem treated here is closest to [134] in that, we adopt the same regularization technique. However, compared to the Border Router (BR) traffic matrix obtained in [134] the scale of the VPN problem is much larger. The computational expense prevents us from solving for a single network-wide problem (which is the case with BR traffic matrices). Instead we evolve approximation techniques that exploit the structure of VPNs and break the problem down to many per-VPN problems. In addition to problems with scale, the measurement information available with VPNs is aggregated across all VPNs and per-VPN information is very often unavailable (in contrast, the BR traffic matrices can exploit fine-grain NetFlow data). Hence it is not straightforward to guage the correctness of the traffic matrix estimates in the case of VPNs. We evolve a set of guidelines to help understand the applicability of the estimates and demonstrate how to obtain the most out of the coarse-grain information available in the case of VPNs.

The value in incorporating traffic matrix information in provisioning was demonstrated in Chapter 7. In that context, the succeeding sections present easily deployed techniques that can help in significantly improving operational efficiencies inspite of coarse-grain measurement information and the prohibitive scale of the problem.

8.4 Traffic Matrix Estimation and Classification

There are multiple uncertainties to overcome while provisioning the network for the aggregate capacity needed for the VPN service. Some of the factors we do not know precisely, a-priori, are:

- The amount of traffic generated by any given source of the VPN. We may only have available the peak rate specification.

- The proportion of the source (hose) traffic that any given link in the network receives.

Often, a new VPN may be admitted as and when the customer request arrives, with very little information being provided by the customer other than peak access capacity requirements. To guarantee the SLAs requested, there is a need to ensure that adequate resources are available. Understanding the "structure" of the VPN helps us in more efficiently provisioning the capacity in the network, and adapting the capacity to changing VPN requirements. By structure, we mean the spatial distribution of the traffic flows between the different source and destinations of the VPN. For example, knowing if there is a hub-and-spoke structure helps in appropriately provisioning capacity in the network since an end-point that is a spoke in a pure "hub-and-spoke" VPN would require capacity primarily between the hub and spoke. However, this information is rarely provided (or known) by the customer at the time when the VPN is admitted. As a result, provisioning without knowledge of the VPN structure could result in a substantial amount of wasted resources.

To infer the structure of a VPN and to achieve efficiencies through adaptive provisioning, we need to examine the way customer endpoints communicate with each other. In other words, we need good estimates of the VPN traffic matrix. We examine measurement data from a large VPN service provider to demonstrate how the structure of VPNs may be deduced and to show the viability of adaptive provisioning strategies.

8.4.1 Measurement Information

In the succeeding sections we present results from our study of measurement information from a large VPN service. Here, we provide a brief description of what data was available from the service. In addition to helping understand the results in the next few sections, this is also meant to be representative of the kind of information that is typically at the the disposal of today's service providers.

Figure (8.1) shows the points in the network where SNMP measurement information is available. Aggregate byte counts over one hour intervals for each PE to PE link are collected by SNMP. This count represents the number of bytes transmitted

Table 8.1: Details of SNMP Information

SNMP Information	CE-PE and PE-PE traffic
SNMP Aggregation Interval	1 hour
VPN Size Range	10s to 100s CEs
Number of PE-PE Links	≈ 6000
Duration of data examined	5 months

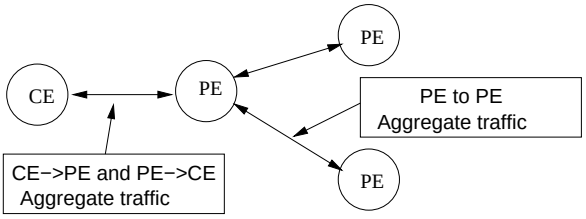


Figure 8.1: Schematic showing available SNMP measurement information

on the PE-PE link due to *all* VPN customers sharing that link. By PE-PE link, we mean a logical link like an MPLS tunnel. In the current dataset there was SNMP information for such logical links for every pair of PEs. The other set of SNMP data available is for the traffic for each Customer Endpoint (CE) to PE link in the form of aggregate byte counts over 15 minute intervals. The CE-PE link is the dedicated access link for the VPN customer and the traffic observed on that link is due only to that customer endpoint.

As one would expect, the SNMP characteristics demonstrate weekly cycles. Figures 8.2 depicts VPNs of different sizes and examines the daily mean of bytes leaving the CE toward the PE and bytes coming to the CE from the network. In some of the VPNs, there is a increase in the magnitude over the months indicating a growth in the VPN. But there is a mean about which the variations of magnitude are seen indicating that there is a certain amount of predictability in the traffic. An additional observation that indicates stable trends is the sensitivity to time-of-day. When the trends are observed separately for mornings, evenings etc, we see the repetition in patterns more clearly - traffic at noon and evening consistently higher, those at nights always low etc. In the succeeding sections, we employ this dataset to evolve a traffic matrix estimation scheme.

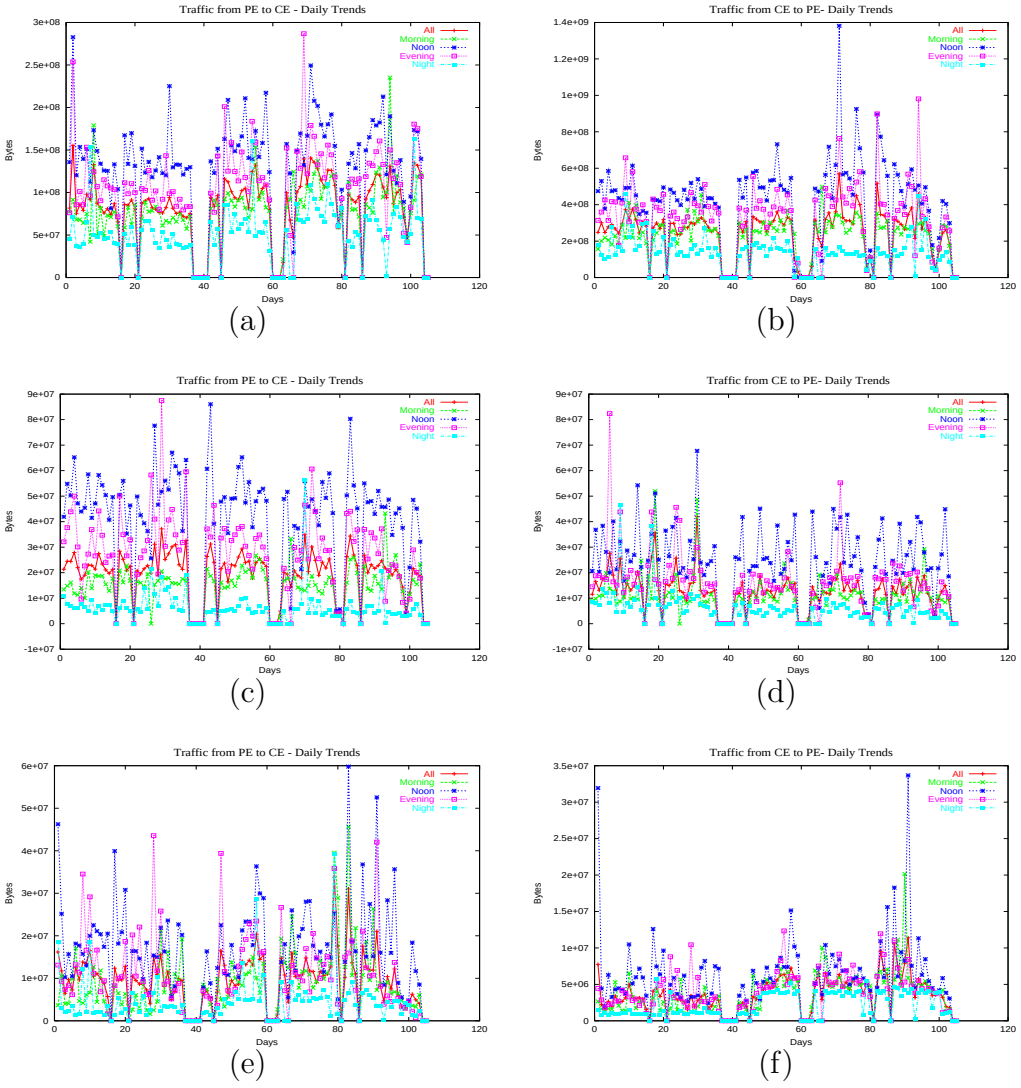


Figure 8.2: Aggregate bytes entering a CE and leaving a CE over 5 months for VPNs of sizes: (a,b) 20; (c,d) 40; (e,f) 79

8.4.2 Cleaning the dataset

Given the large-scale nature of the data that is being handled, it is natural to expect errors and inconsistencies in the collection process. It is very important to remove samples that are manifestation of such errors so that we can understand the performance of our algorithms clearly. In order to clean the dataset, we take recourse to certain properties a valid dataset must satisfy:

1. In any given VPN, the total bytes received by the CEs should be less than

or equal to the total bytes transmitted by the CEs in the same VPN. Noting that CEs in a VPN receive traffic only from other CEs in the same VPN, we observe that whatever data any CE receives must come from a member of the VPN.

2. For any given PE, the total bytes transmitted into the network should be less than or equal to the total bytes offered by all the CEs (of various VPNs) attached to the PE. This is considering the fact that a PE does not generate data.

3. For any given PE, the total bytes transmitted toward CEs attached to it should be less than or equal to the bytes it received from the network. That is, whatever number of bytes the CEs attached to a PE receive that should match the bytes sent to this PE from other PEs.

In reality a large fraction of the dataset does not conform to these rules. We have to relax the rules so that we have a good number of samples to work with. One of the strategies we use is to allow a range of error. For example, we specify that the number of bytes received by CEs in a VPN should be within 10% of the bytes transmitted by all CEs in the same VPN. This means, sometimes, the total output is allowed to be greater than total input. The causes for such cases include data sources not covered by the measurement infrastructure. Typically BGP route information and other control information is transmitted to customers from inside the network, but is not covered by the SNMP data.

Additionally, each VPN is a geographically spread out entity. This means that measurements are usually not time synchronized and sometimes absent due to problems with polling and dropped packets. Some error cases are handled by the measurement modules and a "-1" is indicated in the data. Such samples are discarded. But the majority of the samples are subject to these tests and depending on the objectives of the analytical exercise, the threshold for tolerance can be set.

8.4.3 Estimation techniques

Traffic matrix estimation is an ill-posed problem: with N nodes in a network, the number of traffic demands to be estimated is N^2 while the number of equations we have is only proportional to the number of links. As discussed in §8.3 there are several approaches to solving such under-constrained problems. We mention two popular approaches - the gravity model and the information theoretic approach - and employ both.

Denote the total traffic entering an endpoint s_i by $N^{in}(s_i)$ and the traffic leaving it by $N^{out}(s_i)$. Each element of the traffic matrix indicates the amount of traffic from endpoint s_i toward d_j, denoted by $N(s_i, d_j)$. Thus some portion of $N^{out}(s_i)$ is contributing to $N^{in}(d_j)$. The gravity model attributes a portion of $N^{in}(d_j)$ to each source s_k that transmits to d_j in proportion to the size of $N^{out}(s_k)$. The underlying assumption is that the amount of traffic generated by s_i is independent of that generated by d_j. Thus the following relationship is used: [133]

$$N(s_i, d_j) = \frac{N^{out}(s_i)N^{in}(d_j)}{\sum_{k \neq j} N^{out}(s_k)} \tag{8.1}$$

While the gravity model is simple, it is known to be less accurate in the presence of additional information. One of the methods recently proposed [134] exploits what is generally termed *strategies for regularization of ill-posed problems*. Accordingly a penalized least-squares approach is formulated as:

$$min_x \left\{ ||\mathbf{y} - A\mathbf{x}||^2 + \lambda^2 \sum_{k:g_k>0} \frac{x_k}{T} log\left(\frac{x_k}{g_k}\right) \right\} \tag{8.2}$$

Here, $\mathbf{x}$ is a vector with each x_i representing the variable $N(s_i, d_j)$, the traffic from s_i toward d_j, with the constraint that $x_i \geq 0$. Each element y_i in $\mathbf{y}$ represents the traffic measured for link i, T is the total traffic in the network, and g_k, the gravity estimate for x_k is obtained using Equation (8.1). A is the routing matrix which relates the appropriate variables x_i.

In the present context, s_i and d_j would correspond to the VPN customer endpoints. The set of variables $N(s_i, d_j)$ would be defined for each (s_i, d_j) that are

part of the same VPN, since an endpoint communicates with another endpoint only if it is a part of the same VPN. For example, denote $N(s_1, d_1)$ and $N(s_1, d_2)$ by x_1 and x_2 respectively. If d_1 and d_2 are the only nodes with which s_1 communicates we have the equation $N^{out}(s_1) = x_1 + x_2$. Enumerating such equations for all VPNs in the network would give us the equations denoted by Equation (8.2). Thus the following would be set of equations forming the system:

1. For each source s_i, $N^{out}(s_i) = \sum_j x_j$ where x_j indicates the variables for traffic from s_i to d_j.

2. For each source s_i, $N^{in}(s_i) = \sum_j x_j$ where x_j indicates the variables for traffic from d_j to s_i.

3. For each PE-PE link, $N(PE_{ij}) = \sum_j x_j$ where x_j indicates the variables for all such (s_i, d_j) pairs that transmit on the link PE_{ij}.

In reality, the problem described in Equation (8.2) is too big and computationally expensive to solve. For instance, for the measurement data analyzed here, we have a sparse routing matrix (A in Equation (8.2)) of dimensions $(18 \times 10^3, 950 \times 10^3)$ approximately, with about 2.8×10^6 non-zero elements. In this paper, we evolve a variant of the above estimation techniques to reduce the size of the problem so that the traffic matrices can be quickly computed.

8.4.4 Estimation of VPN Traffic Matrices

Although many VPNs share a common core network, no two endpoints belonging to different VPNs communicate with each other. This lends a kind of separability to our problem and hints at a possible strategy to reduce its size. Instead of solving the problem for all VPNs as part of a single network, we propose to compute the traffic matrices for each VPN independently. In order to do this, we need data on a per-VPN basis to construct the problem as in Equation (8.2). The path from a CE to another CE consists of two segments: a) an access segment (between the PE and the CE) where there is traffic from this VPN alone, b) a core network segment (link between two PEs) which carries traffic multiplexed across multiple VPNs. Typically, we have aggregate SNMP information for each of these segments. Thus we

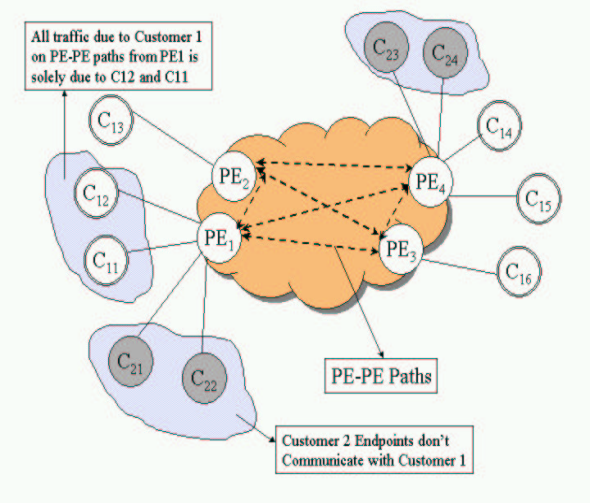


Figure 8.3: Schematic indicating the structural aspects of VPNs that lead to additional equations in the Traffic Matrix estimation problem

need to infer the part of the PE-PE aggregate traffic that is attributable to the VPN being solved for at each step. But there is not enough information to deduce this quantity. Instead, we introduce a bound on the contribution of a particular VPN to the measured PE-PE link traffic.

Figure 8.3 depicts the constraints we evolve by exploiting the structure of VPNs. We consider the set of all CEs in the VPN that can possibly transmit along a given PE-PE link. For example, in Figure 8.3 for the PE1-to-PE3 link, C_{21} and C_{22} are the only endpoints of Customer 2 that offer traffic. The total output from those CEs provides a loose upper bound on the contribution of that VPN to the PE-PE traffic.

Thus, for every PE-PE link which is used by the VPN, we introduce an additional equation as follows:

$$T_l = \sum_{\{(i,j)\in S\}} N(s_i, d_j) + v_l$$

where S is a set of CE pairs which could possibly transmit on the PE-PE link l and v_l is a dummy variable indicative of the contribution of all the other VPNs to the observed PE-PE traffic T_l. We can substitute T_l by $\sum_{\{(i,j)\in S\}} N^{out}(s_i)$ as a loose

upper bound as discussed above. For the example in Figure 8.3, this would indicate the sum of $N^{out}(C_{21})$ and $N^{out}(C_{22})$. Now, we have:

$$\sum_{\{(i,j)\in S\}} N^{out}(s_i) = \sum_{\{(i,j)\in S\}} N(s_i, d_j) + v_l \tag{8.3}$$

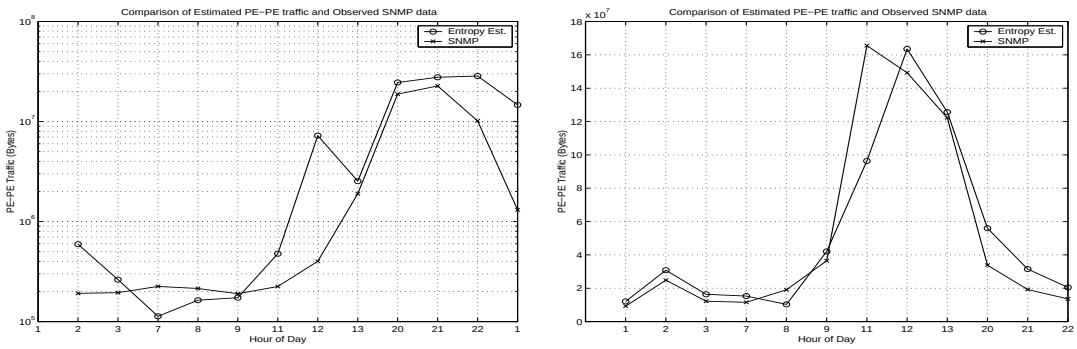
Here v_l is still a dummy variable, but now indicates the fraction of traffic from sources $\{i : \forall k\,(i,k) \in S\}$ that does not go to destinations $\{j : \forall k\,(k,j) \in S\}$. Here we have only used the CE-PE traffic information and not the PE-PE information. Observe that the LHS of Equation (8.3) is the sum of the contributions of all CEs of the VPN attached to this PE. Thus this traffic is intended toward CEs attached to many other PEs and it is possible that this is greater than the PE-PE observed traffic. Thus we can incorporate an additional piece of information in the PE-PE traffic to make the LHS tighter (assuming we are writing the equation for PE-PE link (k,l)):

$$min\{N(PE_{kl}), \sum_{\{(i,j)\in S\}} N^{out}(s_i)\} = \sum_{\{(i,j)\in S\}} N(s_i, d_j) + v_l \tag{8.4}$$

We now are in a position to solve the traffic matrix problem for each VPN separately. The introduction of a loose bound instead of the the actual traffic due to the VPN on the PE-PE link will introduce inaccuracies in the estimated matrix. In the succeeding section we show that these inaccuracies are tolerable for purposes of structural study of VPNs and provisioning decisions.

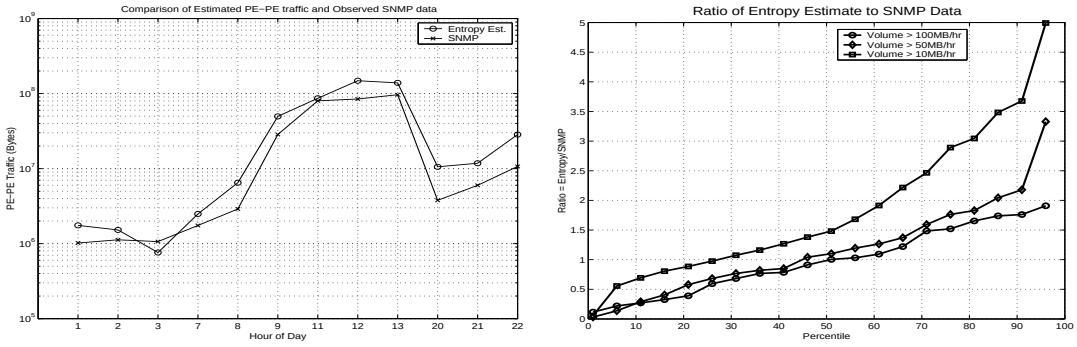
8.4.5 Validation

In order to test the accuracy of the traffic matrices, we compare the traffic on PE-PE links based on the estimated matrices to measured SNMP data for the traffic on these links. As noted previously, the CE to CE path consists of a core network segment where it is shared among multiple VPNs. The traffic matrix solution provides us with values for $N(s_i, d_j)$ for all (s_i, d_j) in the VPN. By summing all such variables that traverse a given PE-PE link, we obtain the estimated contribution of a VPN to a given PE-PE link. We then consider all VPNs that share this link



(a) (b)

Figure 8.4: Estimated traffic vs Observed traffic for two PE-PE links. Accuracy of the estimates is better for PE-PE links with higher traffic. But the estimates mimic the shape and order of the actual traffic in both cases



(a) (b)

Figure 8.5: (a) As seen before Accuracy of the estimates is better for PE-PE links with higher traffic; (b) The error decreases when we look at links with larger volume

and repeat the same procedure. The result is an estimation of the traffic across a large number of PE-PE links, due to all the VPNs that shared the initial PE-PE link chosen.

Figure (8.4) compares the estimated traffic (based on our derived traffic matrices) to the observed SNMP data. Figure (8.4a) depicts a PE-PE link with lower traffic volumes. The accuracy of the traffic matrix estimation reduces for links with lesser traffic. The errors are markedly lesser for a link with higher traffic as depicted in Figures (8.4b) and (8.5a). However, all the graphs shows that the estimated numbers follow the same pattern of variation as the measured numbers in time over a complete day. We conduct the same validation procedure with around 700 links

and obtain Figure (8.4b). This figure indicates the ratio of the TM estimate to the observed PE-PE SNMP traffic for various link volumes. We note the following about various features of this graph:

- For a large number of links, the traffic matrix estimate is within 50% of the SNMP observed traffic and most of the cases it is either close to the SNMP quantity or greater.

- Significant under-estimation happens in around 25% of links considered and these cases were traced to three problems in data for VPNs traversing those links: (a) the PE-CE and CE-PE SNMP data was zero even when the PE-PE data seemed to be significant; (b) there were more bytes transmitted from the PEs to the CEs than what was received by the PE (measurement inconsistency); (c) the total bytes received by CEs was far greater than the total bytes transmitted, i.e., there were external sources not accounted for in the SNMP data.

- Significant over-estimation occurs when the traffic volume on the link being considered is "small". By "small" we mean it is comparable to errors in SNMP data. E.g., if the difference in the number of bytes input to a PE and the number of bytes leaving it is 10MB and the PE-PE traffic volume was 10MB, we would have such a case.

- Although there are very few links which have clean data (less than 10 out of the approximately 700 analyzed) that conforms to rules mentioned in §8.4.2, in all those links, the estimates are in good agreement with the SNMP data.

The validation considered here is not complete since we do not have actual per-VPN traffic data. Due to the scale of the exercise (there are typically hundreds of VPNs and many with more than 100 endpoints) it is unlikely that such data will be available in the near future. In addition, owing to the mission critical and private nature of VPN traffic, there are a lot of administrative hurdles to obtaining access to such traffic. Thus, we need to examine traffic matrix estimates in the current framework and evolve guidelines to guage the reliability of the estimates.

8.4.6 Reliability of Traffic Matrix Estimates

The problem with such estimation techniques is that we have a total number of bytes and some side information with which to arrive at the components that led to that aggregate byte count. Even if the SNMP counts match the estimates, it is not necessary that the individual VPN matrices are correct. But we do not have per-VPN measurement to verify the predictions. So we have to look at other information available in the current data set to help us guage the reliability of the estimates.

Observe that so long as most of the VPNs being analyzed exchange the bulk of their traffic on PE-PE links, the estimates have to be reliable. This is because, the traffic passes three segments: (a) the CE-PE link, (b) the PE-PE link and (c) the PE-CE link on the destination side. Now (a) and (c) are part of the constraints in the optimization formulation. So any solution has the property that the variables sum to the observed access link aggregate counts. Now, consider that the PE-PE link counts match reasonably. If there are a number of CEs of a given VPN attached to a PE, we might still have errors in estimation - we could assign more bytes to a particular CE and still satisfy all constraints of the optimization. But if the number of CEs on a PE is less, then since all the segments of the transit route match with measurement, we must have good estimate. Thus we evolve a set of guidelines to guage the reliability of the estimates.

1. If most of the traffic from customer endpoints must traverse PE-PE links and the number of CEs of a given VPN on the same PE is low, the confidence in the traffic matrix estimate is higher.

2. If most of the VPNs are of the Hub/Spoke kind, since most of the traffic is exchanged with the hub node, reliability of the estimate is high if PE-PE counts match. Hence it is likely that estimates with hub/spoke VPNs is reliable for both structural and provisioning inferences.

3. In the case of hybrid VPNs, structural inferences might be possible only in cases where there is low degree of CE clustering on PEs.

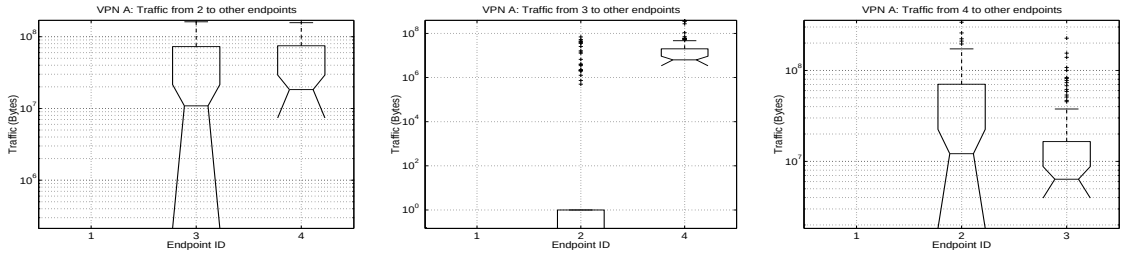


Figure 8.6: Small VPNs have simple structure. The one depicted above has 3 of the 4 nodes in the VPN forming a mesh

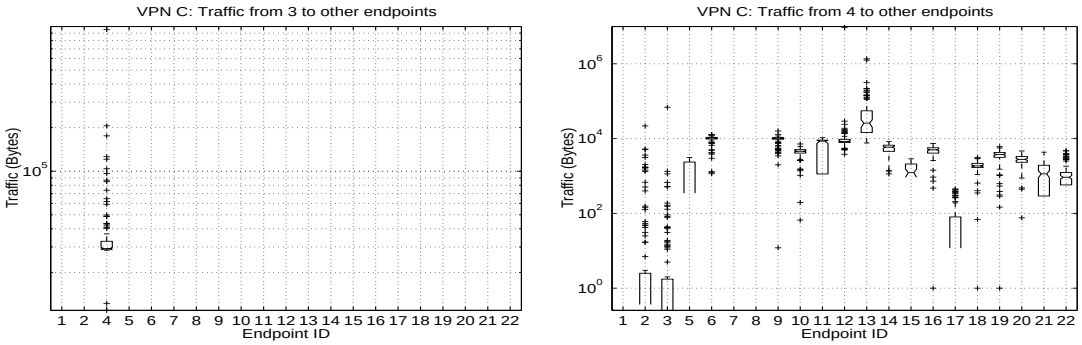


Figure 8.7: Partial Hub/Spoke-like behavior can be seen with some endpoints in VPNs such as above

4. If there are a number of CEs of a given VPN homed into the same PE is high, we cannot draw inferences about structure of the VPNs.

In the current dataset, we have found the number of CEs of a given VPN per PE to be low and that a lot of VPNs are of a hub/spoke nature. Hence we are in a position to study the structural characteristics and temporal characteristics of these VPNs.

8.4.7 Spatial Structure for Classification

Based on the way VPN endpoints communicate with each other (using the derived traffic matrices), three broad categories for the VPN structure can be deduced: (a) Pure hub/spoke, (b) Meshed, and (c) Hybrid VPNs. As the name suggests, a pure hub/spoke VPN features "spoke" nodes that communicate with just one node called the "hub". With meshed VPNs, all endpoints communicate with each other

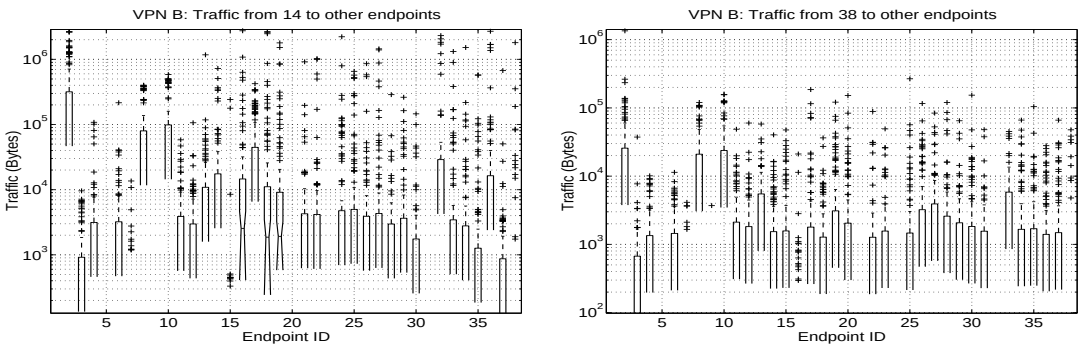


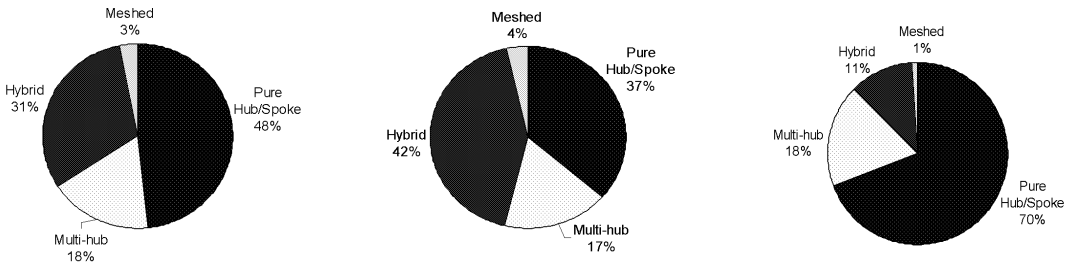
Figure 8.8: A Larger VPN exhibiting complex interactions between various endpoints. There are orders of magnitude difference in the amount of traffic toward different CEs

and the traffic exchanged between any two endpoints is comparable. VPNs where there are multiple hubs or where there is a combination of the first two types can be categorized as hybrid.

Small VPNs (say those with 10 or fewer endpoints) tend to exhibit simple structures like pure hub/spoke or meshed. Figure (8.6) shows the traffic from three endpoints in a VPN of size 4 illustrating a mesh type of communication among the endpoints. Each notched box in the figure represents the range of values for traffic toward a given endpoint. The horizontal lines in each box indicate the upper quartile, median and the lower quartile.

Larger VPNs often exhibit complex structures. VPN C shown in Figure (8.7a) exhibits a partial hub/spoke structure. While some nodes communicate with a single "hub" node, many nodes communicate with multiple endpoints. Figure (8.7a) features traffic flowing from a "spoke" to endpoint 4, which we characterize as a hub. The hub node on the other hand communicates with most other endpoints as shown by Figure (8.7b). Often VPNs cannot be categorized in either of these categories. VPN B featured in Figure (8.8) shows a given endpoint communicating with many other endpoints with orders of magnitude difference in the traffic volumes.

Such structural characteristics are very important to efficiently provision network resources. Once a VPN has been admitted, provisioning can be fine-tuned, exploiting the structure of the VPN. Instead of a mesh of N^2 reservations for a N node VPN, we could tailor allocations depending on the structure of the VPN. There



(a) (b) (c)

Figure 8.9: Structural classification of VPNs: (a) of all sizes; (b) of big VPNs; (c) of small VPNs

is a resultant simplification in provisioning especially in the case of pure hub/spoke and two hub VPNs.

Figure (8.9) depicts the analysis of around 600 VPNs in the dataset. The classification indicates that a significant number of the VPNs are of the hub/spoke nature. Frequently, the VPNs have 2 or 3 hubs for redundancy and load balancing. The classification was carried out with the following thresholding strategy:

1. For each CE, after pruning the bottom 25% of the estimated traffic toward other CEs, we obtain the set of its peers.

2. If a CE communicates with more than 50% of the endpoints in the VPN it is judged to be a hub node.

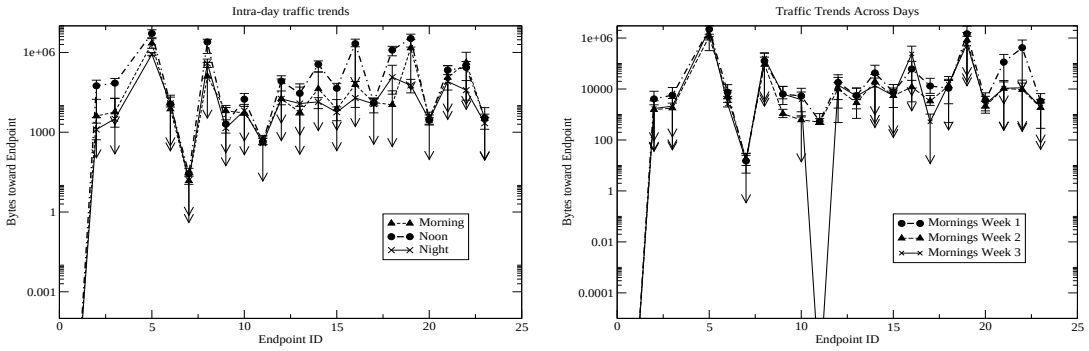
3. If a CE has 1 or 2 peers, it is classified as a spoke.

4. If a CE is in neither of the above categories the VPN is classified as hybrid.

5. After classifying all CEs, we examine the VPN. If the VPN has exactly one hub and all other endpoints are spokes, we classify it as a pure hub/spoke VPN.

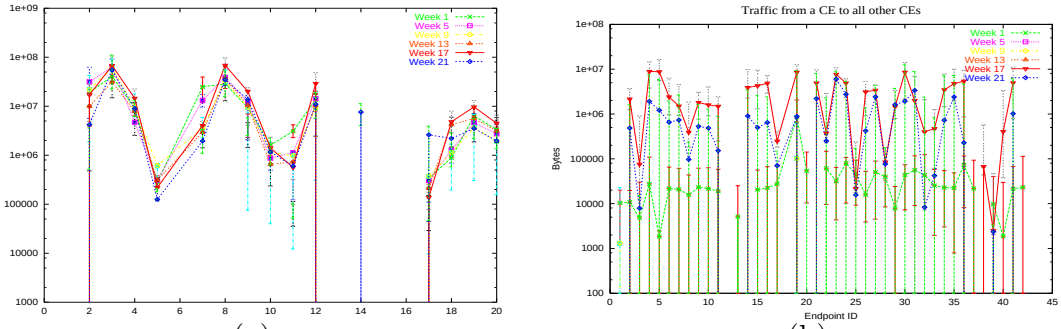
6. If it has more than one hub but the number of hubs is less than 50% of the size of the VPN, we judge the VPN to be of the Multi Hub/Spoke nature.

7. If more than 50% of the nodes in the VPN are hubs, we say the VPN is of the Meshed kind.



(a) (b)

Figure 8.10: (a) An Endpoint communicating with multiple peers; traffic proportions to other endpoints are very similar for different times of day, although the magnitude varies. (b) Traffic trend from an endpoint to others in a VPN remains similar across multiple days.



(a) (b)

Figure 8.11: Additional inter-week trends for traffic from a CE to all other CEs in VPN of higher size.

The classification indicates that with larger VPNs, the structure becomes very complex and there are more of these classified as hybrid. Across various sizes of VPNs, there is a significant fraction that is of the hub/spoke nature (either Pure or Multi Hub/Spoke). This has implications for provisioning and traffic engineering as we shall note in §8.4.9.

8.4.8 Temporal Structure and Provisioning

Once a VPN is admitted, the provider would want to ensure that irrespective of future admissions, the SLAs are met. Further, for a new VPN there is not much information regarding traffic characteristics and hence provisioning has to be

approximate and conservative. Both these points indicate that we need a good understanding of the changes in traffic matrix over time.

Faster changes in traffic characteristics imply that provisioning needs to be more responsive. Links may need to be resized to accommodate existing customers. On the other hand, slower changes to traffic would allow the provider to exploit multiplexing gains and increase the number of customers served. Thus we are interested in studying the changes in traffic matrices over time to judge whether complex dynamic provisioning strategies can yield appreciable gains.

Figure (8.10a) demonstrates the traffic matrix for an endpoint in a hybrid VPN for various times of the day. In this figure, 15 minute SNMP data collected between 6am and 10am are counted for morning traffic, the data from 11am and 2pm is considered as noon and the duration between 8pm-12 midnight is considered as night. Each point in these graphs is the median of the number of bytes seen in those hours, computed using a set of weekdays. The error bars (the vertical lines) indicate the 25^{th} and the 75^{th} percentile values. When the 25^{th} percentile value is too low compared to the median, the error bar is truncated and this is indicated by a downward arrow.

The trend (the proportion of traffic to a given endpoint relative to the others) show a similar shape although there is difference in magnitude indicating that time of day is a distinguishing factor. With this observation, we now consider traffic matrix changes over longer timescales. In Figure (8.10b) we examine traffic trends across multiple weeks for a given endpoint. Each curve shows the median traffic toward an endpoint with the error bars indicating the percentiles as above. Barring one point, the trends for morning traffic across weeks is strikingly similar. This means that the trends do not change too frequently, so that intelligent provisioning schemes have enough time to understand traffic characteristics and act accordingly.

8.4.9 Impact on Provisioning

The discussion in the previous paragraphs lead us to the following important observations:

- **Traffic Engineering:** The traffic matrices provide us an estimate of the size of

the customer aggregate in the core network. This allows us to conduct traffic engineering on a per-customer aggregate basis: we can re-map traffic for a given customer on to a new logical path and have an estimate of the added load and available capacity. Without this information, traffic engineering would have to handle PE-PE aggregates as a whole.

- **Bandwidth Allocation:** Exploiting spatial characteristics can lead to simplified provisioning and efficient resource allocation, especially in the case of endpoints which communicate with just one or two other peers.

- **Customer Differentiation:** Since the traffic matrices provide an estimate of the size of the customer aggregate in the core network, the provider can choose to provide preferential treatment to a selected set of customers more efficiently. For the chosen set of customers, the provider keeps track of the aggregate demands using traffic matrices and makes reservations appropriately. The temporal characteristics of the traffic matrix indicate that the aggregate characteristics vary slowly and can be learnt.

- **Managing network failures:** The additional knowledge of customer traffic can lead elegant management of network failure and maintenance events. E.g., the aggregates leading to a hub node can be mapped on to a new path which has more available capacity.

8.5 Summary and Conclusions

Efficient resource provisioning in VPNs require algorithms that exploit traffic characteristics and VPN structure inferred from measurements. We studied measurement data from a large IP VPN service to evolve techniques to achieve efficient and adaptive provisioning. Typical of such environments, only coarse, aggregate SNMP data is available of traffic on links. The data analyzed here consisted of aggregate information for about 6000 PE-PE links collected over 5 months.

We analyzed this data to get the spatial and temporal characteristics of VPNs. Compared to previous approaches it is harder to get accurate traffic matrices in our case due to the scale of the problem. But, by exploiting the fact that VPN endpoints

only communicate to other endpoints on the same VPN, and arriving at related bounds for the traffic seen on individual links, we are able to get approximate traffic matrices. While approximate, we demonstrated that these were good enough to get a fair idea of VPN structure.

Using the traffic matrices, we were able to identify three broad categories of VPNs: pure hub/spoke, meshed and hybrid VPNs. The matrices for these VPNs showed considerable room for gains via adaptive provisioning. An examination of the temporal properties of traffic matrices showed they are quite stable over the period of a day, and even across days over a period of weeks. Thus, we can use this measurement data to get an idea of the "stable" VPN structure for each customer VPN, and thus, have a reasonable estimate of the demand (as a proportion of the peak bandwidth of the hose) of each VPN customer endpoint on both access and core links.

CHAPTER 9
Conclusions and Future Directions

In this thesis, we presented an edge-based architecture to achieve point-to-multipoint QoS assurances. We began by building a model that specified the nature of the contract with the customer, a parsimonious set of parameters to help providers enforce contracts and achieve multiplexing gain, and algorithms to achieve admission control and provisioning.

We demonstrated that by describing customer traffic using simple dual leaky-bucket shapers, we can evolve an edge-based statistical admission condition. The evaluation of the model in simulation showed that the architecture provides a high degree of multiplexing gain while according control over the trade-off with losses and delays via easily set parameters (ϵ and *Flexibility*). We then proceeded to examine the problem of providing a-priori edge-based delay assurances in the same framework. We demonstrated that a provider can compute meaningful delay bounds given just the limit on the probability of violation and per-path utilization constraints. This means that the provider can compute delay assurances without knowing the individual flow characteristics, but by just enforcing some conditions on the leaky-bucket parameters at the edge of the network.

With an edge-based framework to provide loss and delay we then proceeded to examine two important aspects of the framework: (a) the trade-off in edge-based architectures as compared to one based on signaling; (b) the techniques that would be required in a real network to learn the parameters necessary for the model and deploy it. We found that while signaling based proposals yield higher performance, the gap in performance can be reduced significantly by resorting to adaptive provisioning techniques. This involved estimating and learning traffic matrices of customers. Thus we come to the second aspect of deployability of the framework. Using existing SNMP measurements from a large IP VPN service provider, we evolved a technique to obtain the traffic matrices of VPN customers. We articulated the means to handle lack of measurement information in provider networks and how best to handle

adaptive provisioning within those constraints.

In summary, we presented an edge-based framework for point-to-multipoint QoS, evaluated its costs and benefits and evolved techniques based on SNMP measurement information to aid in deployment.

9.1 Future Directions

The results of the previous chapters demonstrated the benefits of deploying point-to-multipoint QoS with an adaptively provisioned core network. In addition to possible improvements in the statistical admission condition for more efficient admission control, several interesting research directions are opened up.

The ability to decouple delay assurances from customer traffic characteristics allows a provider to evaluate traffic engineering and topology design in terms of their impact on delay. This means a provider can engineer policies to suit a predetermined QoS objective.

While measurement-based techniques allow efficient resource management, they present important questions that need to be answered. Future customer demand can be predicted albeit with finite accuracy. The provider has to brace for situations where demand could overwhelm capacity due to limitations in the adaptive provisioning algorithms. This can be done by quantifying the probability of such events and accordingly maintaining a factor of safety in provisioned resources. In the absence of accurate models for customer traffic such a study is best conducted using measurement-driven simulation experiments. Such a simulation experiment would study the possible violations in service level agreements for realistic workloads.

From the perspective of research, it is interesting to quantify the frequency of such violations as a function of traffic characteristics (e.g., customer size, burstiness of traffic, traffic volume etc.) and develop a set of mechanisms to warn of an impending resource crunch. An engineering view of the problem would be to evolve efficient online techniques that embody the insight gained from research to handle potentially large measurement data on a periodic basis.

One of the important problems facing measurement-based frameworks is the availability and reliability of the measurement data. It is not feasible to instrument

and measure all parts of a network. It is important to identify measurement data that is most valuable for a given algorithm or framework and hence to quantify the relative value of various pieces of information. To rephrase, one needs to answer what parameters to measure and to what granularity. Such a valuation will most likely be specific to the objectives of the measurement-based framework in question.

The second aspect that needs attention is the reliability of the collected data. Provider networks are often large, heterogeneous and geographically distributed. That means data collection is susceptible to a wide-range of problems related to hardware and software. Some of the more routine problems are with respect to time synchronization, loss or corruption of packets carrying measurement data, unresponsive network elements etc. The more subtle problems involve differences in network protocols (e.g., bytes transmitted might involve control packets that should not be attributed to traffic sources), semantics of measured parameters (e.g., dropped bytes might only include queue overflow events and not checksum failures), role of nodes not monitored (e.g., difference between transmitted and received bytes may not indicate drops if there are un-monitored nodes generating data) etc. An essential step while approaching a measurement problem is thus to have a clear description of the semantics of the variables and how they map to the underlying network parameters. Clearly, this is an issue that warrants careful study to help ease deployment of measurement-based architectures.

The findings in this thesis serve both as a baseline implementation and motivation for future endeavors in design of adaptive point-to-multipoint QoS architectures.

LITERATURE CITED

[1] J. Bennett, K. Benson, A. Charny, W. Courtney, and J.-Y. Le Boudec, "Delay jitter bounds and packet scale rate guarantee for expedited forwarding," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 529–540, Aug. 2002.

[2] J. Bennett and H. Zhang, "WF2Q: worst-case fair weighted fair queueing," in *Proc. IEEE INFOCOM'96*, vol. 1, 1996, pp. 120–128.

[3] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," in *Proc. of ACM SIGCOMM '96*, 1996, pp. 143–156.

[4] B. Bensaou, D. Tsang, and K. T. Chan, "Credit-based fair queueing (cbfq): a simple service-scheduling algorithm for packet-switched networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 5, pp. 591–604, Oct. 2001.

[5] M. Bertero, T. Poggio, and V. Torre, "Ill-posed problems in early vision," vol. 76, no. 8, pp. 869–889, Aug. 1988.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2745, Dec. 1998.

[7] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Statistical service assurances for traffic scheduling algorithms," *IEEE J. Select. Areas Commun.*, vol. 18, no. 12, pp. 2651–2664, Dec. 2000.

[8] J.-Y. L. Boudec, "Some properties of variable length packet shapers," *IEEE/ACM Trans. Networking*, vol. 10, no. 3, pp. 329–337, June 2002.

[9] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: An overview," RFC 1633, July 1994.

[10] L. Breslau, "Traffic trace for ns," 2000. [Online]. Available: http://www.research.att.com/ breslau/vint/trace.html

[11] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: architectural issues and performance," in *Proc. of ACM SIGCOMM 2000*, 2000, pp. 57–69.

[12] L. Breslau and S. Shenker, "Best-effort versus reservations: a simple comparative analysis," in *Proc. of ACM SIGCOMM '98*, 1998, pp. 3–16.

[13] C.-S. Chang and J. Thomas, "Effective bandwidth in high-speed digital networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1091–1100, Aug. 1995.

[14] C.-S. Chang, R. Cruz, J.-Y. L. Boudec, and P. Thiran, "A min, + system theory for constrained traffic regulation and dynamic service guarantees," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 805–817, Dec. 2002.

[15] C. Chang, "Stability, queue length, and delay of deterministic stochastic queueing networks," *IEEE Trans. Automat. Contr.*, vol. 39, no. 5, pp. 913–931, May 1994.

[16] C. Chang, W. Song, and Y. Chiu, "On the performance of multiplexing independent regulated inputs," in *Proc. of ACM SIGMETRICS 2001*, 2001, pp. 184–193.

[17] A. Charny and J.-Y. Le Boudec, "Delay bounds in a network with aggregate scheduling," in *Proc. of Quality of Future Internet Services*, Berlin, Germany, Dec. 2000.

[18] G. Chuanxiong, "SRR: An O(1) time complexity packet scheduler for flows in multi-service packet networks," in *Proc. of ACM SIGCOMM 2001*, 2001, pp. 211–222.

[19] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Trans. Networking*, vol. 6, no. 4, pp. 362–373, Aug. 1998.

[20] D. D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," in *Proc. of ACM SIGCOMM '92*, 1992, pp. 14–26.

[21] J. Cobb, "Preserving quality of service guarantees in spite of flow aggregation," *IEEE/ACM Trans. Networking*, vol. 10, no. 1, pp. 43–53, Feb. 2002.

[22] J. Cobb and M. Gouda, "Flow theory," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, pp. 661–674, Oct. 1997.

[23] I. Craig and J. Brown, *Inverse Problems in Astronomy: A Guide to Inversion Strategies for Remotely Sensed Data*. Boston: Adam Hilger, 1986.

[24] R. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[25] ——, "A calculus for network delay, part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 132–141, Jan. 1991.

[26] B. Davie, A. Charny, J. Bennett, K. Benson, J.-Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An expedited forwarding PHB (per-hop behavior)," RFC 3246, Mar. 2002.

[27] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. of ACM SIGCOMM '89*, 1989, pp. 1–12.

[28] B. Doshi, "Deterministic rule based traffic descriptors for broadband isdn: Worst case behavior and its impact on connection acceptance controls," *International Journal of Communication Systems*, March-April 1995.

[29] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: delay differentiation and packet scheduling," in *Proc. of ACM SIGCOMM '99*, 1999, pp. 109–120.

[30] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. van der Merive, "A flexible model for resource management in virtual private networks," in *Proc. of ACM SIGCOMM '99*, 1999, pp. 95–108.

[31] ——, "Resource management with hoses: point-to-cloud services for virtual private networks," *IEEE/ACM Trans. Networking*, vol. 10, no. 5, pp. 679–692, Oct. 2002.

[32] Z. Dziong, M. Juda, and L. Mason, "A framework for bandwidth management in ATM networks-aggregate equivalent bandwidth estimation approach," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 134–147, Feb. 1997.

[33] A. Elwalid, D. Heyman, T. Lakshman, D. Mitra, and A. Weiss, "Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1004–1016, Aug. 1995.

[34] A. Elwalid, D. Mitra, and R. Wentworth, "A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1115–1127, Aug. 1995.

[35] D. Y. Eun and N. Shroff, "A measurement-analytic approach for qos estimation in a network based on the dominant time scale," *IEEE/ACM Trans. Networking*, vol. 11, no. 2, pp. 222–235, Apr. 2003.

[36] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: methodology and experience," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 265–279, June 2001.

[37] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, pp. 368–379, Apr. 1990.

[38] N. R. Figueira and J. Pasquale, "Leave-in-time: a new service discipline for real-time communications in a packet-switching network," in *Proc. of ACM SIGCOMM '95*, 1995, pp. 207–218.

[39] ——, "An upper bound delay for the virtual-clock service discipline," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 399–408, Aug. 1995.

[40] ——, "A schedulability condition for deadline-ordered service disciplines," *IEEE/ACM Trans. Networking*, vol. 5, no. 2, pp. 232–244, Apr. 1997.

[41] V. Firoiu, J. Kurose, and D. Towsley, "Efficient admission control of piecewise linear traffic envelopes at edf schedulers," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 558–570, Oct. 1998.

[42] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z. Zhang, "Theories and models for internet quality of service," in *Proc. of the IEEE*, vol. 90, no. 9, Sept. 2002, pp. 1565–1591.

[43] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 365–386, Aug. 1995.

[44] L. Georgiadis, R. Guérin, and A. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp. 1518–1535, Sept. 1997.

[45] L. Georgiadis, R. Guérin, V. Peris, and R. Rajan, "Efficient support of delay and rate guarantees in an internet," in *Proc. of ACM SIGCOMM '96*, 1996, pp. 106–116.

[46] R. Gibbens, F. Kelly, and P. Key, "A decision-theoretic approach to call admission control in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1101–1114, Aug. 1995.

[47] S. Golestani, "A stop-and-go queueing framework for congestion management," in *Proc. of ACM SIGCOMM '90*, 1990, pp. 8–18.

[48] ——, "A self-clocked fair queueing scheme for broadband applications," in *Proc. IEEE INFOCOM'94*, vol. 1, 1994, pp. 636–646.

[49] ——, "Network delay analysis of a class of fair queueing algorithms," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1057–1070, Aug. 1995.

[50] P. Goyal and H. M. Vin, "Fair airport scheduling algorithms," in *Proc. of NOSSDAV 97*, May 1997, pp. 257–265.

[51] ——, "Generalized guaranteed rate scheduling algorithms: a framework," *IEEE/ACM Trans. Networking*, vol. 5, no. 4, pp. 561–571, Aug. 1997.

[52] P. Goyal, H. M. Vin, and H. Chen, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," in *Proc. of ACM SIGCOMM '96*, 1996, pp. 157–168.

[53] M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," in *Proc. of ACM SIGCOMM '96*, 1996, pp. 15–24.

[54] M. Grossglauser and D. Tse, "A framework for robust measurement-based admission control," in *Proc. of ACM SIGCOMM '97*, 1997, pp. 237–248.

[55] R. Guérin, A. Orda, and D. Williams, "Qos routing mechanisms and ospf extensions," IETF Draft, Nov. 1996. [Online]. Available: http://citeseer.nj.nec.com/guerin96qos.html

[56] A. Gupta, J. M. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, "Provisioning a virtual private network: a network design problem for multicommodity flow," in *ACM Symposium on Theory of Computing*, 2001, pp. 389–398. [Online]. Available: http://citeseer.nj.nec.com/article/gupta01provisioning.html

[57] A. Hung and G. Kesidis, "Bandwidth scheduling for wide-area atm networks using virtual finishing times," *IEEE/ACM Trans. Networking*, vol. 4, no. 1, pp. 49–54, Feb. 1996.

[58] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang, "Delay bounds for a network of guaranteed rate servers with FIFO aggregation," in *Proc. of ACM SIGCOMM '95*, 1995, pp. 2–13.

[59] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks," in *Proc. IEEE GLOBECOM'90*, vol. 1, Dec. 1990, pp. 12–20.

[60] J. Kaur and H. Vin, "Core-stateless guaranteed rate scheduling algorithms," in *Proc. IEEE INFOCOM 2001*, vol. 3, 2001, pp. 1484–1492.

[61] F. Kelly, "Effective bandwidths at multi-type queues," *Queueing Systems*, vol. 9, pp. 5–15, 1991.

[62] F. Kelly, P. Key, and S. Zachary, "Distributed admission control," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 2617–2628, Dec. 2000.

[63] G. Kesidis and T. Konstantopoulos, "Extremal shape-controlled traffic patterns in high-speed networks," *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 813–819, May 2000.

[64] G. Kesidis, J. Walrand, and C.-S. Chang, "Effective bandwidths for multiclass Markov fluids and other atm sources," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 424–428, Aug. 1993.

[65] E. Knightly, "D-BIND, an accurate traffic model for providing qos guarantees to vbr traffic," *IEEE/ACM Trans. Networking*, vol. 5, no. 2, pp. 219–231, Apr. 1997.

[66] ——, "Second moment resource allocation in multi-service networks," in *Proc. of ACM SIGMETRICS '97*, vol. 25, no. 1, 1997, pp. 181–191.

[67] ——, "Enforceable quality of service guarantees for bursty traffic streams," in *Proc. IEEE INFOCOM'98*, vol. 2, 1998, pp. 635–642.

[68] E. Knightly and N. Shroff, "Admission control for statistical qos: theory and practice," *IEEE Network*, vol. 13, no. 2, pp. 20–29, Mar. 1999.

[69] T. Konstantopoulos and V. Anantharam, "Optimal flow control schemes that regulate the burstiness of traffic," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 423–432, Aug. 1995.

[70] K. Kontovasilis and N. Mitrou, "Effective bandwidths for a class of non Markovian fluid sources," in *Proc. of ACM SIGCOMM '97*, 1997, pp. 263–274.

[71] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," in *Proc. of ACM SIGCOMM 2001*, 2001, pp. 135–146.

[72] J. Kurose, "On computing per-session performance bounds in high-speed multi-hop computer networks," in *Proc. of ACM SIGMETRICS '92*, vol. 20, no. 1, 1992, pp. 128–139.

[73] R. Landry and I. Stavrakakis, "Study of delay jitter with and without peak rate enforcement," *IEEE/ACM Trans. Networking*, vol. 5, no. 4, pp. 543–553, Aug. 1997.

[74] J.-Y. Le Boudec and A. Charny, "Packet scale rate guarantee for non-FIFO nodes," in *Proc. IEEE INFOCOM 2002*, vol. 1, 2002, pp. 84–93.

[75] J.-Y. Le Boudec and P. Thiran, *Network Calculus*. Springer-Verlag, 2001. [Online]. Available: http://ica1www.epfl.ch/PS_files/NetCal.htm

[76] H. Lee and J. Mark, "Capacity allocation in statistical multiplexing of ATM sources," *IEEE/ACM Trans. Networking*, vol. 3, no. 2, pp. 139–151, Apr. 1995.

[77] K. Lee, "Performance bounds in communication networks with variable-rate links," in *Proc. of ACM SIGCOMM '95*, 1995, pp. 126–136.

[78] T.-H. Lee, K.-C. Lai, and S.-T. Duann, "Design of a real-time call admission controller for ATM networks," *IEEE/ACM Trans. Networking*, vol. 4, no. 5, pp. 758–765, Oct. 1996.

[79] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[80] C. Li and E. Knightly, "Coordinated multihop scheduling: a framework for end-to-end services," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 776–789, Dec. 2002.

[81] J. Liebeherr, D. Wrege, and D. Ferrari, "Exact admission control for networks with a bounded delay service," *IEEE/ACM Trans. Networking*, vol. 4, no. 6, pp. 885–901, Dec. 1996.

[82] Y. Mansour and B. Patt-Shamir, "Jitter control in qos networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 4, pp. 492–502, Aug. 2001.

[83] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in ATM networks: multiple nodes," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 122–133, Feb. 1997.

[84] A. Medina, C. Farleigh, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "A taxonomy of IP traffic matrices," in *SPIE ITCOM*, Boston, USA, Aug. 2002.

[85] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *Proc. of ACM SIGCOMM 2002*, Pittsburgh, USA, Aug. 2002.

[86] J. Micheel, I. Graham, and N. Brownlee, "The Auckland data set: an access link observed," in *Proc. of the 14th ITC Specialists Seminar on Access Networks and Systems*, vol. 2, Catalonia, Spain, Apr. 2001.

[87] A. Neumaier, "Solving ill-conditioned and singular linear systems: A tutorial on regularization," *SIAM Review*, vol. 40, no. 3, 1998.

[88] NLANR, "Auckland-IV trace archive," 2000. [Online]. Available: http://pma.nlanr.net/Traces/long/auck4.html

[89] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 953–962, Aug. 1995.

[90] I. Norros, J. W. Roberts, A. Simonian, and J. T. Virtamo, "The superposition of variable bit rate sources in an ATM multiplexer," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 378–387, Apr. 1991.

[91] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344–357, June 1993.

[92] ——, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, no. 2, pp. 137–150, Apr. 1994.

[93] R. Parker, *Geophysical Inverse Theory.* Princeton, NJ: Princeton University Press, 1994.

[94] F. L. Presti, Z.-L. Zhang, J. Kurose, and D. Towsley, "Source time scale and optimal buffer/bandwidth tradeoff for heterogeneous regulated traffic in a network node," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 490–501, Aug. 1999.

[95] J. Qiu and E. Knightly, "Measurement-based admission control with aggregate traffic envelopes," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 199–210, Apr. 2001.

[96] S. Raghunath, K. Chandrayana, and S. Kalyanaraman, "Edge-based qos provisioning for point-to-set assured services," in *Proc. of ICC 2002*, vol. 2, Apr. 2002, pp. 1128–1134.

[97] S. Raghunath and S. Kalyanaraman, "Dynamic provisioning for differentiated services on the internet," Rensselaer Polytechnic Institute, NY, Tech. Rep., Jan. 2001. [Online]. Available: http://networks.ecse.rpi.edu/ rsatish/dynaprov.ps

[98] ——, "Statistical Point-to-Set edge-based quality of service provisioning," in *Proc. of QoFIS 2003, Springer Verlag LNCS 2811*, vol. 2, Oct. 2003, pp. 132–141.

[99] S. Raghunath, S. Kalyanaraman, and K. Ramakrishnan, "Quantifying trade-offs in resource allocation for VPNs," in *Proc. of ACM SIGMETRICS 2004, Short Paper*, 2004.

[100] S. Rajagopal, M. Reisslein, and K. Ross, "Packet multiplexers with adversarial regulated traffic," in *Proc. IEEE INFOCOM'98*, vol. 1, 1998, pp. 347–355.

[101] M. Reisslein, K. Ross, and S. Rajagopal, "A framework for guaranteeing statistical QoS," *IEEE/ACM Trans. Networking*, vol. 10, no. 1, pp. 27–42, Feb. 2002.

[102] J. Sahni, P. Goyal, and H. Vin, "Scheduling CBR flows: FIFO or per-flow queuing," in *Proc. of NOSSDAV 99*, 1999.

[103] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, Nov. 1984. [Online]. Available: http://citeseer.nj.nec.com/saltzer84endtoend.html

[104] H. Sariowan, R. Cruz, and G. Polyzos, "SCED: a generalized scheduling policy for guaranteeing quality-of-service," *IEEE/ACM Trans. Networking*, vol. 7, no. 5, pp. 669–684, Oct. 1999.

[105] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," in *Proc. of ACM SIGCOMM '95*, 1995, pp. 231–242.

[106] A. Simonian and J. Guibert, "Large deviations approximation for fluid queues fed by a large number of on/off sources," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1017–1027, Aug. 1995.

[107] V. Sivaraman and F. Chiussi, "Providing end-to-end statistical delay guarantees with earliest deadline first scheduling and per-hop traffic shaping," in *Proc. IEEE INFOCOM 2000*, vol. 2, 2000, pp. 631–640.

[108] ——, "End-to-end statistical delay service under GPS and EDF scheduling: A comparison study," in *Proc. IEEE INFOCOM 2001*, vol. 2, 2001, pp. 1113–1122.

[109] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," in *Proc. of ACM SIGCOMM 2002*, 2002, pp. 133–145.

[110] D. Starobinski and M. Sidi, "Stochastically bounded burstiness for communication networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 206–212, Jan. 2000.

[111] D. Stiliadis and A. Varma, "Efficient fair queueing algorithms for packet-switched networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 175–185, Apr. 1998.

[112] ——, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 6, no. 5, pp. 611–624, Oct. 1998.

[113] ——, "Rate-proportional servers: a design methodology for fair queueing algorithms," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 164–174, Apr. 1998.

[114] I. Stoica and H. Zhang, "LIRA: An approach for service differentiation in the internet," in *Proc. of NOSSDAV 98*, Cambridge, England, July 1998, pp. 115–128.

[115] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high speed networks," in *Proc. of ACM SIGCOMM '98*, 1998, pp. 118–130.

[116] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *Proc. of ACM SIGCOMM '99*, 1999, pp. 81–94. [Online]. Available: http://citeseer.nj.nec.com/article/stoica99providing.html

[117] I. Stoica, H. Zhang, and T. Eugene Ng, "A hierarchical fair service curve algorithm for link-sharing, real-time and priority services," in *Proc. of ACM SIGCOMM '97*, 1997, pp. 249–262.

[118] UCB/LBLN/VINT, "Network simulator - ns (version 2)," 1997. [Online]. Available: http://www.isi.edu/nsnam/ns

[119] S. Valaee, "A recursive estimator of worst-case burstiness," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 211–222, Apr. 2001.

[120] D. Verma, H. Zhang, and D. Ferrari, "Guaranteeing delay jitter bounds in packet switching networks," in *Proc. IEEE TRICOMM '91*, 1991, pp. 35–43.

[121] M. Vojnović and J.-Y. Le Boudec, "Stochastic analysis of some expedited forwarding networks," in *Proc. IEEE INFOCOM 2002*, vol. 2, 2002, pp. 1004–1013.

[122] G. Wahba, *Statistical Decision Theory and Related Topics III*. Academic Press, 1982, vol. 2.

[123] A. Weiss, "An introduction to large deviations for communication networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 938–952, Aug. 1995.

[124] D. Wrege, E. Knightly, H. Zhang, and J. Liebeherr, "Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical trade-offs," *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 352–362, June 1996.

[125] G. Xie and S. Lam, "Delay guarantee of virtual clock server," *IEEE/ACM Trans. Networking*, vol. 3, no. 6, pp. 660–670, Dec. 1995.

[126] J. Xu and R. J. Lipton, "On fundamental tradeoffs between delay bounds and computational complexity in packet scheduling algorithms," in *Proc. of ACM SIGCOMM 2002*, 2002, pp. 279–292.

[127] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 372–385, Jan. 1993.

[128] D. Yates, J. Kurose, D. Towsley, and M. G. Hluchyj, "On per-session end-to-end delay distributions and the call admission problem for real-time applications with qos requirements," in *Proc. of ACM SIGCOMM '93*, 1993, pp. 2–12.

[129] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE INFOCOM'96*, vol. 2, 1996, pp. 594–602.

[130] H. Zhang and D. Ferrari, "Rate-controlled service disciplines," *Journal of High Speed Networks: Special Issue on Quality of Service*, vol. 3, no. 4, 1994.

[131] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines," in *Proc. of ACM SIGCOMM '91*, 1991, pp. 113–121.

[132] L. Zhang, "A new architecture for packet switched network protocols," Ph.D. dissertation, Massachusetts Institute of Technology, July 1989.

[133] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. of ACM SIGMETRICS 2003*, 2003, pp. 206–217.

[134] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proc. of ACM SIGCOMM 2003*, 2003, pp. 301–312.

[135] Z. Zhang, Z. Duan, and Y. Hou, "Fundamental trade-offs in aggregate packet scheduling," in *Proc. of Ninth Intl. Conf. Network Protocols*, 2001, pp. 129–137.

[136] Z.-L. Zhang, Z. Duan, and Y. Hou, "Virtual time reference system: a unifying scheduling framework for scalable support of guaranteed services," *IEEE J. Select. Areas Commun.*, vol. 12, no. 18, pp. 2684–2695, Dec. 2000.

[137] Z.-L. Zhang, Z. Duan, L. Gao, and Y. T. Hou, "Analysis of burstiness and jitter in real-time communications," in *Proc. of ACM SIGCOMM 2000*, 2000, pp. 71–83.