Efficient Traffic Management Based on Deterministically Constrained Traffic Flows

by

Ching-fong Su, B.S., M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 1998

Efficient Traffic Management Based on Deterministically Constrained Traffic Flows

Publication No. _____

Ching-fong Su, Ph.D.
The University of Texas at Austin, 1998

Supervisor: Gustavo de Veciana

Given the Quality of Service (QoS) requirements and various traffic characteristics in multiservice networks, traffic management is critical to the success of network operations. In general, traffic management is designed towards providing QoS guarantees as well as maximizing system utilization. In this dissertation we first analyze statistical multiplexing of deterministically constrained traffic and design a conservative call admission scheme which guarantees the QoS of established connections. In principle, flow control can improve system utilization which otherwise might be low when fixed amounts of network resources are reserved for the traffic with fluctuating requirements. We propose a novel explicit rate flow control algorithm for Available Bit Rate (ABR) service subject to cell loss and fairness constraints. We argue that the combination of conservative admission control and adaptive ABR service can achieve an adequate system utilization and provide robust QoS guarantees. Moreover, overall network efficiency can be further improved by careful route selections. We also consider the routing in a Virtual Path (VP) network and pinpoint problems which arise due to statistical multiplexing and traffic heterogeneity.

Contents

Abstract ii

List of Tables v

List of Figures vi

Chapter 1 Introduction 1
1.1 On Integrated Network and Traffic Management 1
1.2 Chapter 2: Statistical Multiplexing of Deterministically Constrained Traffic 3
1.3 Chapter 3: Virtual Paths—Resource Allocation and Routing 3
1.4 Chapter 4: Explicit Rate Flow Control of ABR Traffic 4
1.5 Chapter 5: Conclusions 5

Chapter 2 Statistical Multiplexing of Deterministically Constrained Traffic 6
2.1 Introduction 6
2.2 Traffic Description 7
2.3 On the Statistical Multiplexing 8
2.3.1 Large Deviations Results 8
2.3.2 Upper Bound on the Overflow Probability 9
2.3.3 Uniqueness of Overflow Time Scale 9
2.3.4 An Example: Leaky Bucket Constrained Traffic 10
2.3.5 Multiplexing Heterogeneous Traffic 11
2.4 In Search of the Worst Case Traffic 11
2.5 Simulation Results and Summary 14
2.6 Appendix 16
2.6.1 Proof of Lemma 2.3.1 16
2.6.2 Proof of Lemma 2.3.2 17

Chapter 3 Virtual Paths—Resource Allocation and Routing 18
3.1 Introduction 18
3.2 Traffic Mix and Admissible Region 20
3.2.1 Characteristics of the Admissible Region 20
3.2.2 Linearization of the Admissible Region 22
3.3 VP Integration 24
3.3.1 Integration or Segregation? 24

3.3.2 The Benefit of VP Integration . . . 27
3.4 Routing and Traffic Mix . . . 28
3.4.1 Routing of Permanent VCs . . . 28
3.4.2 Routing and Linearization of Admissible Region . . . 31
3.4.3 Routing with Multiple VPs and Traffic Types . . . 33
3.4.4 Routing with Dynamic Call Arrivals . . . 36
3.5 Simulation Results . . . 37
3.5.1 Two Traffic Types and Two VPs . . . 37
3.5.2 Three Traffic Types and Three VPs . . . 39
3.5.3 Robustness & Linearization . . . 41
3.6 Summary . . . 41
3.7 Appendix . . . 43
3.7.1 Proof of Lemma 3.4.1 . . . 43
3.7.2 Proof of Lemma 3.4.2 . . . 43

Chapter 4 Explicit Rate Flow Control of ABR Traffic 45
4.1 Introduction . . . 45
4.2 Explicit Rate Flow Control—a Fluid Model . . . 47
4.3 Guaranteeing No Loss and Positive $e(t)$ . . . 49
4.4 Asymptotic Stability . . . 52
4.4.1 Linear Feedback . . . 52
4.4.2 Nonlinear Feedback . . . 53
4.5 Steady State and Fairness . . . 54
4.5.1 Steady State Characteristics of Greedy Sources . . . 54
4.5.2 Impact of Constrained Flows . . . 54
4.5.3 Sessions with Minimum Cell Rate Guarantees . . . 54
4.6 ABR Call Admission and Statistical Multiplexing . . . 55
4.7 Implementation and Design Issues . . . 56
4.7.1 Protocols and Complexity . . . 57
4.7.2 Estimation of Link Status . . . 58
4.7.3 Design Parameters . . . 58
4.8 Simulation and Performance Evaluation . . . 60
4.9 Summary . . . 62
4.10 Appendix . . . 64
4.10.1 Proof of Lemma 4.4.1 . . . 64
4.10.2 Proof of Asymptotic Stability . . . 64
4.10.3 Pseudo-code for Sources and Bottleneck Links . . . 67

Chapter 5 Conclusions 68

Bibliography 70

List of Tables

2.1 Comparison of admissible numbers of connections. 15

3.1 Number of VPs carrying homogeneous traffic. 35
3.2 The comparison of blocking probabilities. 39
3.3 Capacity and the effective bandwidths. 40
3.4 Comparison of blocking probabilities. 40
3.5 Routing sequence under various traffic loads. 41

4.1 Comparison of admissible numbers. 62

List of Figures

2.1 Contour plot of $K(\cdot,\cdot)$ and locus of (x,y). 10
2.2 $e(t)$ and locus of (x,y) on the contour plot of $K(\cdot,\cdot)$. 11
2.3 A periodic On/Off traffic constrained by $e(t) = \min[pt, \rho t + \sigma]$. 12
2.4 The proposed traffic pattern. 14
2.5 $\Lambda_t^*(ct+b)$ for various t and T_m. 15
2.6 Simulated overflow probability(with 95% confidence interval) and the upper bound. 16

3.1 An admissible region and its linear approximations. 22
3.2 Another admissible region. 22
3.3 Admissible regions for various δ. 23
3.4 Linearization of the admissible region's boundary. 23
3.5 Traffic mix vs. $\frac{\sigma_1}{\sigma_2}$. 25
3.6 Integration and segregation regions. 26
3.7 The difference in bandwidth requirements. 27
3.8 Integration and segregation regions. 27
3.9 Two VPs and two traffic types. 28
3.10 $F(\alpha)$. 29
3.11 The joint admissible region. 30
3.12 Examples of $\overline{ST}$ with maximum length. 31
3.13 Examples of optimum $\overline{ST}$ locations. 32
3.14 Parallel boundaries. 32
3.15 Non-parallel boundaries. 33
3.16 Problem set-up. 34
3.17 Four routing algorithms. 38
3.18 Determining the length of $\overline{ST}$. 43

4.1 Network bottleneck model. 47
4.2 Source model characteristics. 48
4.3 An upper bound on the queue length. 51
4.4 The control system model and its equivalent. 52
4.5 The non-linear model with constraints. 53
4.6 Average and maximum bandwidth increments for aggregated VBR sources. 57
4.7 The estimation of $\hat{n}(t)$. 59

4.8 A saturated function $f(\cdot)$. 59
4.9 A bottleneck link shared by ABR and VBR connections. 61
4.10 Queue dynamics for different drift function gains k. 61
4.11 Queue length and ACR in a changing environment. 62
4.12 45 VBR and 3 ABR connections. 63
4.13 $k(x)$ lies in a sector. 65
4.14 Nyquist plot of $G(j\omega)$ and the circle. 66

Chapter 1

Introduction

This chapter provides an overview of problems in managing integrated services networks as well as a brief introduction to this dissertation. We begin with a review of the key features of integrated services networks and the new design and management challenges that result from heterogeneity in both traffic characteristics and service requirements. Subsequent sections provide an overview of the chapters of the dissertation.

1.1 On Integrated Network and Traffic Management

Traditionally there have been separate communication networks for carrying specific types of traffic. For example, voice traffic was carried by public circuit-switched networks, while data was usually transmitted over packet-switched computer networks. However, due to the significant advances in technology, including signal processing and computer engineering, communication networks are evolving and merging in a very dramatic fashion. Recent examples include Internet phone, and data service over CATV and satellite networks.

The deployment of optical fiber and high speed switching electronics have brought greater bandwidths to both circuit-switched and packet-switched networks. With bit rates as high as gigabit per second becoming available, it is reasonable to ask whether it is possible to merge conventionally separated networks and provide a single infrastructure upon which various current and future communication services can be efficiently supported. The motivations for this are various, including the obvious desirability of sharing network resources, thus avoiding unnecessary duplication of network infrastructure and improving flexibility. The need for flexible networks is particularly important in light of the wide spectrum of applications evolving in current computer networks with both heterogeneous traffic characteristics and quality of service (QoS) requirements.

Asynchronous Transfer Mode (ATM) has been developed and promoted as the technology of choice to support Broadband Integrated Services Digital Networks (BISDN) [51, 44, 14]. ATM is based on statistical multiplexing and switching of fixed size cells—53-byte packets. The service is connection-oriented with cells transported on Virtual Channel Connections (VCCs). Most importantly, QoS such as delay or cell loss are defined as part of the attributes which distinguish among various service types. Given such QoS requirements and the diversity in traffic characteristics, traffic management is critical to en-

sure the success of network operations. In particular, the promise of ATM lies in providing "managed bandwidth" in an efficient and predictable manner.

In general, the objective of network management is to provide QoS guarantees as well as to maximize system utilization. With a careful management of traffic and resources, the network can carry more connections while keeping its QoS commitments, or provide better service without additional hardware. In the following we discuss three principal components of network management.

- **Admission control.** This is the mechanism by which the network determines whether a connection request should be accepted, and is a basic tool for ensuring the QoS guarantees.

- **Routing.** When a network considers accepting a new connection, it must account for the impact the new connection might have on the rest of network. For example, an appropriate route or path must be selected in order to avoid congestion when the connection traverses the network.

- **Flow and congestion controls.** These are used to slow down a source's transmission so as to prevent network congestion, or speed up the transmission in order to take advantage of the available capacity inside the networks.

These three control methods constitute a basis for traffic management. Admission control ensures there are enough network resources to support the QoS of established connections. Flow control can in principle improve system utilization which otherwise might be low when fixed amounts of network resources are reserved for the traffic with fluctuating requirements. Overall network efficiency is further improved by careful route selections.

In this dissertation we consider these management issues in the context of multiservice networks. We begin by considering admission control for deterministically constrained traffic in Chapter 2. Unlike telephone networks, call admission in an multiservice network is not straightforward. A number of problems arise in performing admission control. For example, the estimation of resource requirements for new connections requires information such as the traffic characteristics and current system status, both of which are not readily available.

To reduce management and signaling costs, Virtual Paths (VPs) have been proposed to allow for joint handling or bundling of connections in ATM networks. In Chapter 3 we consider the routing in a VP network, and pinpoint problems which arise due to statistical multiplexing and traffic heterogeneity. In Chapter 4 we propose an explicit rate flow control algorithm for Available Bit Rate (ABR) service, which provides connections with dynamically varying capacities. We discuss the QoS issue for ABR service and identify the key parameters which may affect performance. Finally we provide some design principles and conclude in Chapter 5.

1.2 Chapter 2: Statistical Multiplexing of Deterministically Constrained Traffic

The QoS (e.g., cell loss and delay) that a traffic flow experiences when it passes through network links depends on several factors. Indeed, the cell rates, the link capacity, and the buffer size, may all affect the resulting queuing delay and the amount of possible cell loss. Because prior information about the traffic characteristics is essential for networks to reserve an appropriate amount of resource and guarantee QoS, a "description" (or model) of the traffic is needed.

One can model the traffic by a stochastic process, such as Gaussian process, or Markov Modulated Poisson Process, see e.g.,[22, 35], and then analytically evaluate the performance of the system. However, stochastic traffic descriptors are difficult to enforce and/or verify. Moreover, the predicted performance is then sensitive to modeling errors and vulnerable to mis-behaving users.

By contrast, traffic could be modeled using deterministic descriptors. Various researchers have used deterministic traffic models for which the worst case traffic behavior can be determined and bounds on delay and buffer requirements are established, see e.g., [13]. However, the analysis is usually limited to *deterministic* QoS such as worst case end-to-end delay, because multiplexing of connections is typically not considered. In [34] *statistical* performance is analyzed for deterministically constrained traffic, but the analysis therein is based on a Gaussian approximation approach.

In Chapter 2 we consider a deterministic approach to modeling traffic, where the maximum arrivals of each connection over any time interval t are bounded. Given such traffic, we analyze statistical multiplexing in a buffered link and derive an upper bound on the overflow probability by using the Large Deviations results. The upper bound is computed based on the traffic descriptor as well as the link capacity and buffer size. The dependence of multiplexing performance on system parameters is shown to have a significant impact on performance and thus on admission control.

Given such deterministic descriptors, it is also reasonable to ask what would be the worst case conforming traffic pattern which results in the worst overflow probability upon being multiplexed in a buffered link. Although this question is critical, it is an open problem and the solution is not yet available. Nevertheless, we present some observations about this problem and discuss the characteristics of possible solutions.

1.3 Chapter 3: Virtual Paths—Resource Allocation and Routing

ATM technology is based on multiplexing and switching cells transported on virtual channel connections (VCCs). Virtual path connections (VPCs) in turn allow for joint handling of connections and, among other reasons, have been proposed to reduce the management costs. The VP layer is likely to serve as an intermediate resource management layer, wherein key resource allocation decisions are made on a somewhat slower time scale than typical connection times. Indeed, one can use the VP layer to simplify call admission control,

routing, and to segregate traffic based on QoS, traffic characteristics, or service classes. In Chapter 3 we consider a variety of problems concerning the traffic management on VP layered networks.

Due to statistical multiplexing, the bandwidth requirement of a connection on various resources across the network is not fixed. Indeed, it would depend on the interfering traffic with which the connection is multiplexed. Motivated by this fact, we first consider the following problem. Given two traffic types with different quality of service requirements, should one segregate such flows on their own VPs, or is it to the network's advantage to multiplex the flows on a single VP guaranteeing the most stringent QoS requirement? The answer to this problem is not straightforward. One needs to assess the QoS requirement as well as traffic characteristics and mix in order to chose the best policy.

Next we consider the problem of routing multiple traffic types with a pre-defined traffic mix onto multiple VPs between a source destination pair. We show that it is not advantageous to let each VP carry every traffic type. In fact, the optimum solution to this problem, perhaps surprisingly, suggests that only a small number of traffic types, or even homogeneous traffic should be present on each VP.

These results show that it is critical to account for the statistical multiplexing and traffic mix in making routing decisions and resource allocations. We propose a simple routing algorithm which accounts for these factors and results in a much better blocking probability than Least Loaded Routing—itself often claimed to be "optimal."

1.4 Chapter 4: Explicit Rate Flow Control of ABR Traffic

The rationale for including ABR service, in ATM networks, is to provide an economical and flexible way to carry data traffic. From the service provider's point of view, ABR traffic promises to enhance utilization by directing sources to make the most of the network's available capacity subject to minimum cell rate and cell loss guarantees.

In Chapter 4 we propose a novel explicit rate flow control algorithm intended for ABR service on an ATM network subject to loss and fairness constraints. The goal is to guarantee low cell loss in order to avoid throughput collapse due to retransmission by higher level protocols. The mechanism draws on measuring the current queue length, bandwidth availability, as well as tracking the current number of active sessions contending for capacity, to adjust an explicit bound on the source transmission rates.

We identify the factors that might affect the queue overflows, e.g., round trip delay of sessions and a constraint on the aggregate bandwidth variability, and then propose simple design rules aimed at achieving transmission with controlled loss in a dynamic environment. One might argue that the feedback control mechanism of ABR would be ineffective for individual bursty sessions. To improve the system utilization, we also consider the role of statistically multiplexing bursty ABR sources and show that this allows for an increase of the acceptable number of concurrent sessions.

1.5 Chapter 5: Conclusions

In Chapter 5 we review the results that have been discussed in this dissertation. We review our findings in the context of designing multiservice networks and discuss some directions for the future research.

Chapter 2

Statistical Multiplexing of Deterministically Constrained Traffic

2.1 Introduction

In order to carry heterogeneous traffic on the same network and meet their Quality of Service (QoS) requirements, several changes need to be made to current networks. Techniques for traffic modeling, routing, Connection Admission Control (CAC) and scheduling are required and intensively being investigated. Because of the high speed of the network and the stringent service requirements, it is difficult to keep the service commitments by flow control. Hence CAC plays an important role in meeting the QoS guarantee.

The core of the CAC mechanism is based on the ability to estimate performance, such as overflow probability, given the information about the current link capacity, buffer sizes, number of connections, and traffic characteristics. One approach to estimating performance is to generate statistical models for traffic and then analytically evaluate the steady state performance of the queue, see, e.g.,[22, 35]. Unfortunately, the enforcement of such source models and the sensitivity of predicted performance to modeling errors are two concerns. Even if traffic can be adequately modeled, the problem of actually estimating the performance of statistically multiplexed traffic streams is generally difficult.

An alternative approach is to determine the worst case congestion behavior based on a worst case traffic specification, see, e.g., [16, 15, 34, 38]. Given some traffic parameters, such as mean rate and/or maximum cumulative arrivals within a time interval $(0,t]$, a traffic stream satisfying the constraints is said to be the worst behaving traffic when it causes the "worst" performance upon being multiplexed. Thus, an upper bound on the overflow probability or the worst case performance of the system can be computed by multiplexing the associated worst behaving traffic streams.

In this chapter we refine the second approach, by further considering the time-scale problem. We consider a set up where N streams are multiplexed at a buffered link and consider the probability that the aggregated arrivals within a fixed time interval may overcome the link's potential capacity and further exceed the buffer capacity. The traffic is described by a traffic constraint function $e(t)$, and the cumulative arrivals of the traffic stream within an interval of length t, $A(\tau,\tau+t]$, are bounded by $e(t)$ uniformly in τ [13, 52]. For off-line

traffic, $e(t)$ can be found by computing the empirical envelope of the cumulative arrivals from the traffic's trace, while for real-time traffic, it can be enforced by a traffic policer at the entrance to the network.

For example, the Generic Cell Rate Algorithm[1] for Usage Parameter Control (UPC) in an ATM network is a typical deterministic traffic descriptor or policer [18] . The main functionality of UPC is to ensure the conformance of every connection to its pre-negotiated traffic characteristics, such that the QoS of sessions sharing the network will not be deteriorated by misbehaving traffic. The leaky bucket algorithm is usually considered one of the simplest traffic policing algorithms.

In a leaky bucket algorithm, a counter representing the number of tokens is maintained for each connection. The counter value increases at a pre-defined rate ρ, i.e., the token arrival rate, but its maximum value is clamped by the bucket size σ. Each time a cell departs from a leaky bucket, a token is consumed and the counter decreases by 1. A cell can not depart when the counter is 0. In principle, the leaky bucket can be viewed as regulating the traffic to a rate ρ, but allowing a burst of σ cells. In addition, a leaky bucket is usually coupled with a peak rate regulator to ensure that the cell rate is bounded when the counter value is positive. For example, if a traffic stream is policed by a leaky bucket (σ,ρ) and peak rate regulator p, then $A(\tau,\tau+t] \leq \min[pt, \rho t+\sigma]$.

This chapter is organized as follows: In §2.2 we discuss the deterministic traffic descriptor which will be used in this chapter. An upper bound on the overflow probability is derived in §2.3. In §2.4 the worst case traffic *pattern* is considered. Simulation results are presented in §2.5 and followed by a summary.

2.2 Traffic Description

Let $a(0,t]$, $\forall t \in [0,T]$, denote the cumulative arrivals of a traffic stream with duration T, e.g., a video stream which is stored off-line. Note that this corresponds to a fixed sample path of arrivals, i.e., stored data with a pre-defined transmission schedule. We define a periodic extension of $a(0,t]$ as $\tilde{a}(0,t+nT] \equiv a(0,t] + na(0,T], \forall t \in [0,T],\ n \in \mathbb{N}$. With the extended cumulative arrivals $\tilde{a}(0,t]$, one can define an *empirical envelope* $e(t)$ by

$$e(t) = \max_{\tau} \tilde{a}(\tau,\tau+t]. \tag{2.1}$$

We will use $e(t)$ as a deterministic traffic descriptor since it bounds the maximum arrivals over any arbitrary interval of length t.

Suppose we randomize the "phase" Θ of the arrivals uniformly over $[0,T]$; then the resulting periodically extended arrival process is a stationary and ergodic random process. Let us define $\mu = a(0,T]/T$ as the mean rate of arrivals for the off-line traffic stream. It can be shown that $E[\tilde{a}(\Theta,\Theta+t]] = \mu t$. We will show that the random variable $\tilde{a}(\Theta,\Theta+t]$ plays an important role in considering the performance of statistically multiplexing N such $e(t)$-constrained streams.

In the case of real-time traffic, we shall assume a similar traffic description exists. Suppose the arrival process is policed so that it is deterministically constrained by a concave

[1] Also referred to as the Virtual Scheduling or continuous-state Leaky Bucket algorithm.

function $e(t)$, i.e., the cumulative arrivals $A(\tau,\tau+t]$ over any interval of length t satisfy $A(\tau,\tau+t] \le e(t)$. Thus, $e(t)$ plays the same role as an empirical envelope for off-line traffic. For example, if the traffic is policed by a leaky bucket with parameters (σ,ρ) and a peak rate regulator p, then the deterministic constraint on the traffic is $e(t) = \min[pt, \rho t + \sigma]$.

An equivalent way to describe a deterministically constrained traffic flow is by determining its impact on a buffer with a fixed service rate c, e.g., the maximum buffer occupancy or burstiness curves $b(c)$, see, e.g.,[36]. The traffic description $b(c)$ corresponds to the worst case buffer occupancy for a stream served at rate c. On one hand, for a traffic stream with envelope $e(t)$, its maximum buffer occupancy is given by $b(c) = \sup_{t>0}[e(t) - ct]$. On the other hand, given $b(c)$, one can bound the traffic arrivals by $A(0,t] \le ct + b(c)$, $\forall c$. Since the choice of c is arbitrary, $\hat{e}(t) = \inf_{c>0}[ct + b(c)]$ is another envelope function for $A(0,t]$. It can be shown that $\hat{e}(t) \ge e(t)$ with equality when $e(t)$ is concave. In fact, $\hat{e}(t)$ is the smallest concave function which is greater than or equal to $e(t)$ for all $t > 0$.

From the definition of the empirical envelope $e(t)$ in (2.1), we know it is increasing but not necessarily concave. However, a concave $e(t)$ results in nice structural properties as shown in §2.3.3. Thus we will upper-bound $e(t)$ with $\hat{e}(t)$ when $e(t)$ is not concave. In the sequel, we simply assume $e(t)$ is concave.

2.3 On the Statistical Multiplexing

2.3.1 Large Deviations Results

Suppose we are given N stationary i.i.d. traffic streams with arrivals $A_i(0,t]$ which are constrained by the same empirical envelope $e(t)$. We assume that $\lim_{t\to\infty} \frac{e(t)}{t} = \mu_e$, i.e., all traffic streams constrained by $e(t)$ have finite mean cell rates. Let μ be the mean rate of the streams $A_i(0,t]$, it follows that $\mu_e \ge \mu$. Next we consider the overflow probability when statistically multiplexing N such streams in a buffer of size Nb with capacity Nc. Let A_t^N be the aggregated arrivals of N streams over an interval of length t, i.e., $A_t^N = \sum_{i=1}^N A_i(0,t]$, where all $A_i(0,t]$ are i.i.d. random variables. Using the large deviations results in [9, 7], we know that for large N the probability that A_t^N exceeds $N(ct+b)$ at a particular time scale t is given by $\mathbb{P}(A_t^N > N(ct+b)) \approx \exp[-N\Lambda_t^*(ct+b)]$, where $\Lambda_t^*(\alpha) = \sup_\theta(\theta\alpha - \log M(\theta))$ and $M(\theta) = \mathrm{E}\exp[\theta A_i(0,t]] = \mathrm{E}\exp[\theta A(0,t]]$.[2]

Since the traffic processes are stationary, the steady state queue length can be written as $Q^N = \sup_{t>0}[A_t^N - Nct]$. Thus the steady state overflow probability can be approximated by [7]

$$\mathbb{P}(Q^N > Nb) \approx \sup_{t>0} \mathbb{P}(A_t^N > N(ct+b)) \approx \exp[-N \inf_{t>0} \Lambda_t^*(ct+b)]. \tag{2.2}$$

Note that such large deviations results can be improved by including the Bahadhur-Rao leading term [39]. Intuitively (2.2) suggests that the overflow probability $\mathbb{P}(Q^N > Nb)$ essentially corresponds to the largest probability of A_t^N exceeding $N(ct+b)$ over any fixed time interval t. A time scale t which achieves the inf in (2.2) is called an *overflow* time

[2] We drop the subscript i since the random variables $A_i(0,t]$ are i.i.d.

scale, which is not necessarily unique. If the distribution of $A(0,t]$ is known, $M(\theta)$ can be computed and $\mathbb{P}(A_t^N > N(ct+b))$ can be estimated. However, the distribution of $A(0,t]$ is usually unavailable, and the overflow time scale is unknown. Herein we assume only knowing that $A(0,t]$ is bounded by $e(t)$ and $\mathbb{E}[A(0,t]] = \mu t$.

2.3.2 Upper Bound on the Overflow Probability

Since the same peak $e(t)$ and mean μt constraints can be satisfied by various distributions, there may be multiple rate functions $\Lambda_t^*(\alpha)$ associated with different *conforming* traffic streams, which results in different estimation of the overflow probabilities. One approach to overcome this difficulty is to determine the "worst" conforming distribution and obtain an upper bound on the overflow probability.

Since the maximum mean rate which an $e(t)$-constrained traffic can achieve is $\lim_{t\to\infty}\frac{e(t)}{t} = \mu_e$, we assume that μ is equal to μ_e in order to obtain the largest amount of traffic allowed by $e(t)$. In addition for a fixed t, it can be proved that when $A(0,t]$ is Bernoulli distributed with peak $e(t)$ and mean μt, this leads to the largest bound $\exp[-N\Lambda_t^*(ct+b)]$ [24, 54, 38] over all distributions with the same peak and mean. Note that a stationary traffic flow in conformance with $e(t)$ would typically be unable to achieve a Bernoulli distributed "marginal" $A(0,t]$; this means that the bound would not be tight. However, the upper bound computed from a Bernoulli distributed $A(0,t]$ is still shown to be useful by simulations. In §2.4 we will further discuss the relationship between $e(t)$ and the resulting distribution of $A(0,t]$.

Given a Bernoulli random variable with mean μt and peak $e(t)$, the rate function is given by [3][p.148]

$$\Lambda_t^*(ct+b) = K(\frac{ct+b}{e(t)}, \frac{\mu t}{e(t)}), \text{ where } K(x,y) = \begin{cases} x\log\frac{x}{y} + (1-x)\log\frac{1-x}{1-y} & \text{if } 0 \le y < x \le 1 \\ \infty & \text{otherwise.} \end{cases}$$

Thus from (2.2), the steady state overflow probability is bounded by

$$\mathbb{P}(Q^N > Nb) \le \exp[-N \inf_{t>0} K(\frac{ct+b}{e(t)}, \frac{\mu t}{e(t)})].$$

2.3.3 Uniqueness of Overflow Time Scale

Next we investigate the characteristics of $K(\frac{ct+b}{e(t)}, \frac{\mu t}{e(t)})$ as a function of t in order to compute $\inf_{t>0} K(\frac{ct+b}{e(t)}, \frac{\mu t}{e(t)})$. For an overflow to occur within a time scale t, it is clear that $e(t) > ct+b$ since otherwise $K(\cdot) \to \infty$, i.e., overflows are not going to occur. Given the concavity of $e(t)$, it can be shown that the region of interest is the interval, $[t_1, t_2] = \{t \in \mathbf{R} \mid 0 \le \frac{ct+b}{e(t)} \le 1\}$. In the following we show that $K(\frac{ct+b}{e(t)}, \frac{\mu t}{e(t)})$ has an unique minimizer t^* in this interval. The proof of the following lemmas can be found in §2.6.

Lemma 2.3.1 *For $0 \le y < x \le 1$, $K(x,y)$ is strictly convex in the pair (x,y).*

Lemma 2.3.2 *Let $x(t) = \frac{ct+b}{e(t)}$, $y(t) = \frac{\mu t}{e(t)}$ and $[t_1,t_2]$ be the interval where $0 \le \frac{ct+b}{e(t)} \le 1$, then $A = \{(\lambda, y(t)) \in \mathbf{R}\times\mathbf{R} \mid t \in [t_1,t_2],\ \lambda \ge x(t)\}$ is convex.*

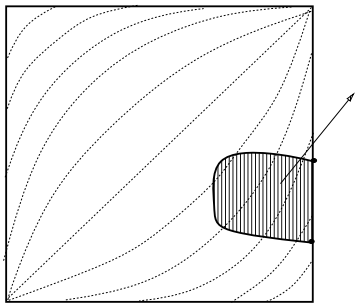

Figure 2.1: Contour plot of $K(\cdot,\cdot)$ and locus of (x,y).

Theorem 2.3.1 *$K(\frac{ct+b}{e(t)}, \frac{\mu t}{e(t)})$ has a unique minimizer on the interval $[t_1,t_2]$.*

Proof: By Lemma 2.3.1 and Lemma 2.3.2 it is clear that there exists a unique minimizer of $K(\cdot,\cdot)$ on the set $A \cap [0,1] \times [0,1]$, see Fig.2.1. In addition, since $K(\cdot,\cdot)$ is increasing in x for any fixed y, the minimizer lies on the boundary of A, i.e., the locus of $(x(t),y(t))$ for $t \in [t_1,t_2]$. ■

We have shown that there is a t^* minimizing $K(\cdot,\cdot)$ and maximizing the overflow probability. For any traffic conforming to $e(t)$, the overflow probability is thus upper-bounded by

$$\mathbb{P}(Q^N > Nb) \approx \sup_{t>0} \mathbb{P}(A_t^N > N(ct+b)) \leq \exp[-NK(\frac{ct^*+b}{e(t^*)}, \frac{\mu t^*}{e(t^*)})]. \qquad (2.3)$$

2.3.4 An Example: Leaky Bucket Constrained Traffic

Next we consider in detail the multiplexing of the traffic streams which are policed by leaky buckets. Consider a stationary, ergodic, and leaky bucket policed traffic, with maximum arrivals on interval $(0,t]$ given by $A(0,t] \leq e(t) = \min[pt, \rho t + \sigma]$ and $\mathbb{E}A(0,t] \leq \rho t$. By applying (2.3), we can obtain an upper bound on the overflow probability. Our objective is to find the minimum K^* and minimizer (x^*,y^*) of $K(\cdot,\cdot)$. The envelope $e(t)$ and the contour plot of $K(\cdot,\cdot)$, as well as with the locus of $(x(t) = \frac{ct+b}{e(t)}, y(t) = \frac{\rho t}{e(t)})$ for $t \in [t_1,t_2]$ are shown in Fig.2.2.

In Fig.2.2, $\overline{MNO}$ is the locus of $(x(t),y(t))$ for $t \in [t_1,t_2]$, i.e., the set of possible overflow time scales. Since $K(\cdot,\cdot)$ is increasing in x for any fixed y, it is clear that (x^*,y^*) would fall in $\overline{NO}$ instead of $\overline{MN}$. It is worth noticing that the key parameters in Fig.2.2 affecting the overflow probability are the ratios of capacity to mean arrival rate $\frac{c}{\rho}$, buffer size to burstiness $\frac{b}{\sigma}$, and mean rate to peak rate $\frac{\rho}{p}$. More insight can be obtained by varying these parameters. For example, if $\frac{b}{\sigma}$ increases, the point L is moving to the right and K^* would increase. It leads to a smaller overflow probability, which is expected for less bursty traffic streams. Similarly, if we increase $\frac{c}{\rho}$, we will get larger K^*, resulting in a smaller overflow probability.

However, the changes as we adjust $\frac{\rho}{p}$ are trickier. Given points O,L on the contour plot of $K(\cdot,\cdot)$, it can be shown that $K(\cdot,\cdot)$ is convex along the line $\overline{OL}$. Depending on the

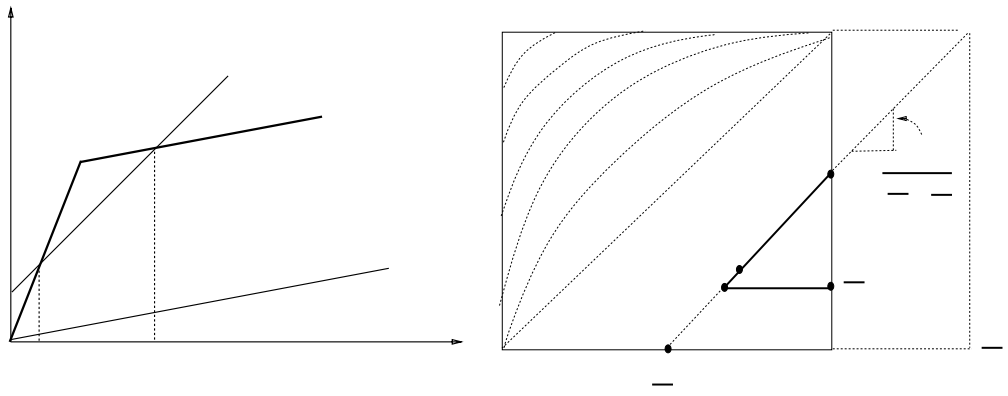

Figure 2.2: $e(t)$ and locus of (x,y) on the contour plot of $K(\cdot,\cdot)$.

position of N, (x^*,y^*) might be located at point N or somewhere between N and O. This introduces an interesting threshold phenomenon. If we increase $\frac{\rho}{p}$ from 0, it does not affect the K^*(or the overflow probability) at first; after passing some point, (x^*,y^*) is located at N and K^* starts to increase as $\frac{\rho}{p}$ increases. Since we know that $K(\cdot,\cdot)$ is strictly convex on $\overline{NO}$, the minimizer and minimum are both unique and easy to compute.

2.3.5 Multiplexing Heterogeneous Traffic

When the N connections carried by a buffered link are not constrained by the same envelope function, some modifications are needed in order to find an upper bound on the overflow probability. Suppose there are two types of independent connections multiplexed on the same link and they are constrained by envelope functions $e_1(t)$ and $e_2(t)$ respectively. Let Nf_1 and Nf_2 be the numbers of connections for each type, where $f_1 + f_2 = 1$. For fixed t, the exponent $\Lambda_t^*(\alpha)$ from large deviations results is given by

$$N * \Lambda_t^*(\alpha) = \sup_\theta[N\theta\alpha - Nf_1 \log M_1(\theta) - Nf_2 \log M_2(\theta)] = N * \sup_\theta[\theta\alpha - \log(M_1^{f_1}(\theta)M_2^{f_2}(\theta))],$$

where $M_1(\theta)$ and $M_2(\theta)$ are the Moment Generating Functions (MGF) of the cumulative arrivals associated with different traffic types. Therefore, this problem can be transformed into another one with N *homogeneous* connections by constructing a *"typical"* traffic type to represent the mix of different connections. The *"typical"* traffic will have an envelope function $e_{mix}(t)$ associated with it, such that the resulting MGF of cumulative arrivals $M_{mix}(\theta) = M_1^{f_1}(\theta) * M_2^{f_2}(\theta)$. Unfortunately, an expression of $e_{mix}(t)$ in terms of $e_1(t)$ and $e_2(t)$ is not available at this point in time. An approximation is given by $e_{mix}(t) = f_1 e_1(t) + f_2 e_2(t)$. However, this approximation can not guarantee staying on the conservative side. That is, it may be quite optimistic.

2.4 In Search of the Worst Case Traffic

In §2.3.2 we used the MGF of a Bernoulli random variable to upper-bound the MGF of $A_i(0,t]$ in order to estimate the overflow probabilities based on the Chernoff's bound.

However, a stationary traffic conforming to $e(t)$ would be unable to achieve a Bernoulli-distributed $A_i(0,t]$. Hence the upper bound on the overflow probability is not associated with a worst traffic *pattern*, i.e., is not tight.

Let us consider the worst case $e(t)$-constrained traffic using fluid model for two scenarios: bufferless and buffered links. In a bufferless link, the overflow occurs *instantaneously* whenever the aggregate cell rates exceed the link capacity. Hence to find a worst case traffic, one only need to consider the MGF of the *marginal* cell arrival rates. For $e(t)$-constrained traffic, the largest mean rate is $\lim_{t\to\infty}\frac{e(t)}{t}=\mu_e$ and the largest peak rate is $\lim_{t\to 0^+}\frac{e(t)}{t}=p_e$.[3] Therefore, it can be shown that a traffic with cell rates alternating between p_e and 0 subject to the envelope constraint $e(t)$ is a worst case traffic pattern for a bufferless link.

For example, various researchers have shown by different approaches that a periodic On/Off traffic source is *a* worst case leaky bucket constrained traffic pattern, see Fig. 2.3. It causes the worst overflow probability when large numbers of such streams are multiplexed in a bufferless link, see e.g.,[38, 15, 54]. However, the worst case traffic pattern for a buffered link is still an open problem. Simulation results and simple computations suggest that the periodic On/Off traffic is in fact not the worst case traffic if buffering is allowed, see e.g.,[15, 53].

Figure 2.3: A periodic On/Off traffic constrained by $e(t)=\min[pt,\rho t+\sigma]$.

In a buffered link, an overflow occurs when the aggregate arrivals over some time interval exceed the link's potential capacity, which causes the queue length to grow and eventually exceed the buffer size. Hence the MGF of cumulative arrivals $A(0,t]$ are critical in assessing the overflow probability. However, it is not clear how the distribution of $A(0,t]$ is affected by the envelope constraint $e(t)$, which is to be satisfied by all time shifted $A(0,t]$.

Consider stationary and ergodic arrivals processes $A_i(0,t]$ which are constrained by $e(t)$ and have a mean rate μ_e. Suppose that the traffic is carried by a link with capacity Nc and buffer size Nb. As shown in §2.3.2, the overflow probability can be approximated by

$$\mathbb{P}(Q^N>Nb)\approx\sup_{t>0}\mathbb{P}(A_t^N>N(ct+b))\approx\exp[-N\inf_{t>0}\Lambda_t^*(ct+b)].$$

Our objective is to find a traffic pattern which would lead to the smallest rate function $\inf_{t>0}\Lambda_t^*(ct+b)$, and in order to do so we resort to finding a traffic pattern which results in the largest MGF, i.e., $\mathbb{E}[\exp(\theta A_i(0,t])]$.

Let $a(0,t]$ be a sample path of $A(0,t]$. The MGF of $A(0,t]$ is given by

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T\exp(\theta a(\tau,\tau+t])d\tau=\mathbb{E}[\exp(\theta A(0,t])]\quad\text{a.s.}$$

[3] For the $e(t)$ of interest, both limits exist.

To explore the nature of the worst case traffic patterns, we consider the following optimization problem:

$$\begin{aligned} \max_{a(0,t]} \quad & \tfrac{1}{T}\int_0^T \exp(\theta a(\tau,\tau+t])d\tau \\ \text{s.t.} \quad & 0 \le a(\tau,\tau+t] \le e(t), \qquad \forall\,\tau,\ t>0 \end{aligned} \tag{2.4}$$

Note that in doing so we are considering finite duration *functions* $a(0,t]$ which are uniformly constrained by $e(t)$ for any τ and t. It can be shown that the set of functions satisfying the constraints constitute a convex feasible set. Since the objective is a convex *functional*, a solution to this optimization problem will be an "extremal" point of the feasible set. However, the *uniform* constraints imposed by $e(t)$ over all τ and t result in a complicated set of feasible arrivals functions. Nevertheless an exploration of the extremal functions within the set provides some insight.

An example: leaky bucket constrained traffic. Next we consider the worst case leaky bucket constrained traffic. To overcome the complexity of the feasible set resulting from the constraints imposed by $e(t)$, we will relate a sample path $a(0,t]$ of cell arrivals to the token status of the leaky bucket which polices $a(0,t]$. By doing so, we formulate (2.4) from a different perspective and discuss the characteristics of possible solutions based on the "extremal" property of the feasible set.

Let $w(\tau)$ be the number of tokens in the leaky bucket at time τ. Suppose no tokens are lost, so the maximum mean cell rate ρ can be achieved. The cell process $a(\tau,\tau+t]$ and the token number $w(\tau)$ can be related as follows:

$$a(\tau,\tau+t] = \rho t + w(\tau) - w(\tau+t), \forall\,\tau,\ t.$$

Now we consider the constraints on the number of tokens $w(\tau)$. First, since the number of tokens is non-negative and the bucket size is σ, it follows that $0 \le w(\tau) \le \sigma$. Second, the upper bound on $w'_+(\tau)$,[4] i.e., the rate of increases of token number, is ρ, when no tokens are consumed (or cell rate is zero). Similarly $w'_+(\tau)$ is lower-bounded by $\rho - p$ when tokens are consumed at rate p, i.e., the cell rate is p.

Therefore (2.4) can be written as:

$$\begin{aligned} \max_{w(\cdot)} \quad & \tfrac{1}{T}\int_0^T \exp(\theta(\rho t + (w(\tau) - w(\tau+t)))d\tau \\ \text{s.t.} \quad & 0 \le w(\tau) \le \sigma, && \forall \tau > 0 \\ & \rho - p \le w'_+(\tau) \le \rho, && \forall \tau > 0. \end{aligned} \tag{2.5}$$

A maximizer $w^*(\tau)$ is necessarily an "extremal" point of the convex feasible set of (2.5). Let us say that a traffic is "on" when its cell rate is positive. Otherwise it is said to be "off" when cell rate is zero. Since maximizers are extremal, a worst case leaky bucket constrained traffic has the following properties:

1. cell rates are either ρ or p over an "on" period.

[4] Since $w(\tau)$ is continuous, we assume that its *right* derivative exists for all τ.

2. cell rates can not be equal to ρ exclusively over an "on" period.

3. cell rates are unimodal over an "on" period.

Based on the aforementioned properties for the "extremal" traffic, we consider a family of possible traffic patterns which we believe to be the candidate worst case traffic for the buffered scenario. The traffic is periodic and has symmetric cell arrival rates over each "on" state, see Fig. 2.4. This family of traffic patterns is parameterized by the length of mean rate "burst" which surrounds the peak rate "burst" of length $\frac{\sigma}{p-\rho}$ over each "on" state. It is clear that the duration of the "off" state is $\frac{\sigma}{\rho}$ in order to achieve maximum allowable mean cell rate ρ. Note that a special case of $T_m = 0$, i.e, no mean rate "burst", is indeed a periodic On/Off traffic shown in Fig. 2.3, which is a worst traffic pattern for the bufferless case.

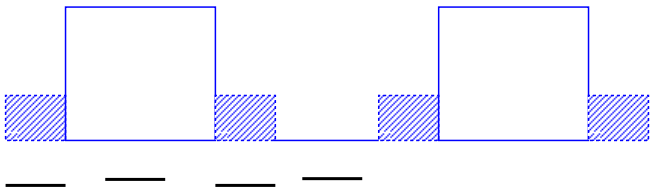

Figure 2.4: The proposed traffic pattern.

We will numerically compute the exponent $\Lambda_t^*(ct+b)$ associated with randomizing the phases of these periodic arrivals for various T_m, t and then determine $\inf_{T_m, t>0} \Lambda_t^*(ct+b)$, where t is the likely time scale for an overflow to occur. The parameters are as follows: $e(t) = \min[120t, 80t+400]$, $c = 100$ and $b = 160$. Fig. 2.5 shows that $\inf_{T_m, t>0} \Lambda_t^*(ct+b)$ (i.e., the lowest point on the surface) is achieved at $t = 9.9, T_m = 6.5$. This plot suggests that a periodic On/Off traffic ($T_m = 0$) is far from being a worst traffic in a buffered scenario. Because the conventional periodic On/Off traffic does not result in the worst cast overflow probability upon being multiplexed, resource reservation based on multiplexing them may not be able to guarantee the overflow probability requirement for the connections on a buffered link.

2.5 Simulation Results and Summary

In order to verify the effectiveness in using the proposed upper bound on the overflow probability to perform CAC, we multiplex off-line video traffic streams in a single multiplexer with a fixed total capacity and buffer size. We obtain an envelope function $e(t)$ by periodically extending the video traces. The overflow probability is then obtained from simulation results and compared with the analytical upper bound given by $\exp[-NK(\frac{ct^*+b}{e(t^*)}, \frac{\mu t^*}{e(t^*)})]$, see Fig.2.6.

As shown in Fig.2.6, the simulated overflow probability is indeed upper-bounded by the one predicted by our analysis. Since the estimation of the overflow probability is likely to be used for resource reservation and connection admission control, we also consider the

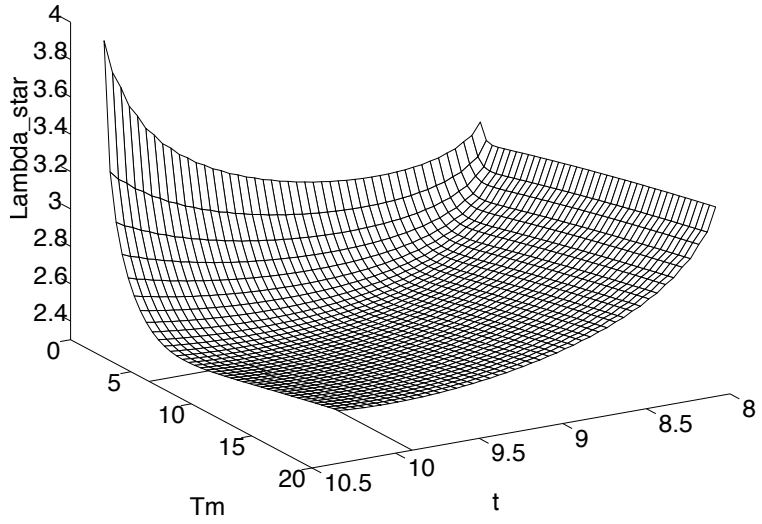

Figure 2.5: $\Lambda_t^*(ct+b)$ for various t and T_m.

Overflow prob.	3.564e-02	1.403e-02	5.244e-03	1.858e-03	5.747e-04	9.329e-05	3.359e-05
No. simulated	59	57	55	53	51	49	47
No. predicted	50	46	42	40	38	32	30
No. pk-rate allocation	15	15	15	15	15	15	15

Table 2.1: Comparison of admissible numbers of connections.

effectiveness of the proposed upper bound from this perspective. Given a fixed link capacity and buffer size, we compute the admissible numbers of streams predicted by our approach, by a peak rate allocation scheme, and observed from simulation, see Table.2.1.

Our approach leads to a conservative number of admissible connections due to overestimation of the overflow probability. This is not unexpected, since our analysis gives an upper bound on the overflow probability for any traffic conforming to the same envelope and the video trace used in the simulation is not necessarily the worst case traffic. However, without making any assumption on the traffic's statistics, the admissible number predicted by our approach improves significantly over that made by a peak-rate allocation scheme. For example, for an overflow probability $3.36 \cdot 10^{-5}$, there is 100% improvement in the admissible number of connections.

To make the upper-bound approach less conservative, one can combine it with online measurement. By doing so, we can estimate the distribution of $A_i(0,t]$ rather than assuming it is Bernoulli. For example, when a multiplexer considers accepting a new connection, the descriptors of its envelope function (e.g., leaky bucket parameters) can be sent to the multiplexer. The multiplexer then uses the Bernoulli assumption for the new connections, while measurement data are used for the existing connections.

In a network carrying heterogeneous traffic, under-utilization because of overestimation of overflow probabilities is an issue. However, the spare capacity could in principle

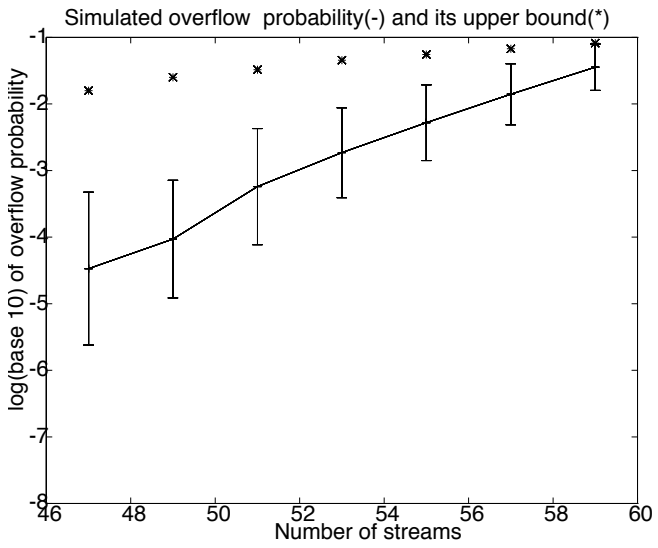

Figure 2.6: Simulated overflow probability(with 95% confidence interval) and the upper bound.

be used by Available Bit Rate traffic, whose cell rates can be controlled dynamically to increase the network's utilization. In Chapter 4 we will discuss the management of ABR traffic. The advantage of our approach is that it uses only the deterministic traffic descriptors which the network can enforce, rather than the traffic statistics. Hence it can lead to a robust, simple connection admission control mechanism.

2.6 Appendix

2.6.1 Proof of Lemma 2.3.1

Lemma 2.3.1 *For $0 \leq y < x \leq 1$, $K(x,y)$ is strictly convex in the pair (x,y).*
Proof: Consider $\lambda \in (0,1)$,

$$\begin{aligned}
& K(\lambda x_1 + (1-\lambda)x_2, \lambda y_1 + (1-\lambda)y_2) \\
= \; & (\lambda x_1 + (1-\lambda)x_2) \log \frac{\lambda x_1 + (1-\lambda)x_2}{\lambda y_1 + (1-\lambda)y_2} \\
& \quad + [1 - (\lambda x_1 + (1-\lambda)x_2)] \log \frac{1 - (\lambda x_1 + (1-\lambda)x_2)}{1 - (\lambda y_1 + (1-\lambda)y_2)} \\
= \; & (\lambda x_1 + (1-\lambda)x_2) \log \frac{\lambda x_1 + (1-\lambda)x_2}{\lambda y_1 + (1-\lambda)y_2} \\
& \quad + [\lambda(1-x_1) + (1-\lambda)(1-x_2)] \log \frac{\lambda(1-x_1) + (1-\lambda)(1-x_2)}{\lambda(1-y_1) + (1-\lambda)(1-y_2)}.
\end{aligned}$$

By Theorem 2.7.1(log sum inequality) [12][p.29],

$$\begin{aligned}
&K(\lambda x_1 + (1-\lambda)x_2, \lambda y_1 + (1-\lambda)y_2) \\
&\leq \lambda x_1 \log \frac{\lambda x_1}{\lambda y_1} + ((1-\lambda)x_2) \log \frac{(1-\lambda)x_2}{(1-\lambda)y_2} \\
&\qquad + \lambda(1-x_1)\log\frac{\lambda(1-x_1)}{\lambda(1-y_1)} + (1-\lambda)(1-x_2)\log\frac{(1-\lambda)(1-x_2)}{(1-\lambda)(1-y_2)} \\
&= \lambda K(x_1,y_1) + (1-\lambda)K(x_2,y_2).
\end{aligned}$$

The equality holds when $\frac{x_1}{y_1} = \frac{x_2}{y_2}$ and $\frac{1-x_1}{1-y_2} = \frac{1-x_2}{1-y_2}$, or equivalently $x_1 = y_1$. Therefore, $K(x,y)$ is strictly convex in pair (x,y), for $0 \leq y < x \leq 1$. ■

2.6.2 Proof of Lemma 2.3.2

We will need the following two lemmas in order to prove Lemma 2.3.2.

Lemma 2.6.1 *If $e(t)$ is differentiable everywhere in $[t_1,t_2]$, $\frac{e^{'}(t)}{e(t)-te^{'}(t)}$ is decreasing in t.*

Proof: If $e(t)$ is differentiable everywhere in $[t_1,t_2]$, it follows that $e^{'}(t) > 0$ and $e^{''}(t) < 0$ since $e(t)$ is an increasing concave function. Thus $e^{'}(t)$ is decreasing in t and $(e(t) - te^{'}(t))$ is increasing in t, We can conclude that $\frac{e^{'}(t)}{e(t)-te^{'}(t)}$ is decreasing in t. ■

Lemma 2.6.2 *If $e(t)$ is not differentiable at some $t \in [t_1,t_2]$, $\frac{e^{'}_{+}(t)}{e(t)-te^{'}_{+}(t)} \leq \frac{e^{'}_{-}(t)}{e(t)-te^{'}_{-}(t)}$.*

Proof: If $e(t)$ is not differentiable at t, then $e^{'}_{+}(t) < e^{'}_{-}(t)$ because $e(t)$ is a continuous, concave function. This lemma follows directly. ■

Lemma 2.3.2 *Let $x = \frac{ct+b}{e(t)}$, $y = \frac{\mu t}{e(t)}$ and $[t_1,t_2]$ be the interval where $0 \leq \frac{ct+b}{e(t)} \leq 1$. Then $A = \{(\lambda, y(t)) \in \mathbf{R}\times\mathbf{R} \mid t \in [t_1,t_2],\ \lambda \geq x(t)\}$ is convex.*

Proof: Consider a function $f(y(t)) = x(t), t \in [t_1,t_2]$. Note that A is the *epigraph* epi(f) of f. By Theorem[5.10] [49], the convexity of f corresponds to the convexity of epi(f).

If $e(t)$ is differentiable everywhere in $[t_1,t_2]$, the derivative of f is

$$\frac{dx}{dy} = \frac{\frac{dx}{dt}}{\frac{dy}{dt}} = \frac{\frac{ce(t)-(ct+b)e^{'}(t)}{e^2(t)}}{\frac{\mu e(t)-\mu t e^{'}(t)}{e^2(t)}} = \frac{ce(t)-(ct+b)e^{'}(t)}{\mu(e(t)-te^{'}(t))} = \frac{c-b\frac{e^{'}(t)}{e(t)-te^{'}(t)}}{\mu}.$$

By Lemma 2.6.1, $\frac{e^{'}(t)}{e(t)-te^{'}(t)}$ is decreasing in t, so the derivative of f is increasing.

If $e(t)$ is not differentiable at some $t \in [t_1,t_2]$, by Lemma 2.6.2, we still could show $f^{'}_{+}(x) \geq f^{'}_{-}(x)$. This proves the convexity of f, and hence of A. ■

Chapter 3

Virtual Paths—Resource Allocation and Routing

3.1 Introduction

Traditionally networks have been designed to carry specific types of traffic, e.g., the telephone network for voice and computer networks, such as the Internet, for data traffic. Current integrated-services networks aim to provide a single infrastructure upon which various current and future communication services can be efficiently supported. The motivations for this are various, including the obvious desirability of sharing network resources, thus avoiding unnecessary duplication of network infrastructure and improving flexibility. The need for flexible networks is particularly important in light of the wide spectrum of applications evolving in current computer networks with both heterogeneous traffic characteristics and quality of service (QoS) requirements. Such *diversity* will be an important feature in future broadband networks so network design and management may have to carefully account for it [45].

ATM has been designed to meet the possible needs of integrated broadband communication networks. This technology is based on multiplexing and switching cells transported on virtual channel connections (VCCs). Virtual path connections (VPCs) allow for joint handling of bundled VCCs and can serve as an effective way of reducing complex signaling and management tasks in a core network. The VP layer is in fact likely to serve as an intermediate resource management layer, wherein key resource allocation decisions are made on a somewhat slower time scale than typical connection times. Indeed, one can use the VP layer to simplify call admission control, routing, and to segregate traffic based on QoS, traffic characteristics, or service classes. This chapter addresses the question of whether or not segregating heterogeneous traffic with different QoS requirements on separate VPs is desirable. We shall see that traffic heterogeneity plays a critical role in multiplexing, and careful allocation of traffic mixes is essential to achieving good performance.

Similar care is needed in making routing decisions in a heterogeneous environment. There is much research and experience with routing policies in circuit-switched networks but one might ask if these principles will extend to multiservice networks. In circuit-switched networks, there exists a clear separation among connections, and the reserved

resources for each connection are well defined from the start. Routing schemes selecting the "least-loaded path" can be useful in single service networks because they tend to balance the traffic load across the network and minimize the blocking probability [23]. However, in a multiservice network, it has been suggested that a "most-loaded path" strategy might be preferable in order to leave room on the "least-loaded path" for future connections with high bandwidth requirements [47]. Various other routing policies, such as trunk reservation, have been investigated, and these are likely to also play a role in integrated services networks, see e.g., [31].

In this chapter we consider the impact that heterogeneity and statistical multiplexing might have on the design and performance of simple routing decisions, such as selecting among many VPCs that have already been dimensioned. Although we assume capacity has been partitioned among VPCs one would nevertheless hope that a moderate degree of statistical multiplexing can be achieved by sharing the resources allocated to VPCs. One difficulty in considering the role of statistical multiplexing in such systems is that the "effective bandwidth" required for each traffic stream may depend on the current load and capacity of the system, see e.g., [47]. A simple example can illustrate this and show how it might in turn impact routing policies. Consider a bufferless link shared by two types of traffic whose cell arrival rates are for simplicity modeled by Gaussian random variables with means m_1, m_2 and variances σ_1^2, σ_2^2 respectively. Suppose the link currently has n_1 and n_2 ongoing connections of each type. It can be shown that the capacity requirement is then roughly given by

$$c(n_1, n_2) = (n_1 m_1 + n_2 m_2) + k\sqrt{(n_1\sigma_1^2 + n_2\sigma_2^2)},$$

where k is an overall QoS parameter related to a link overflow probability. The bandwidth required for an additional connection of Type 1 can be approximated by[1]

$$\frac{\partial c}{\partial n_1} \approx m_1 + \frac{1}{2}\, k\, \sigma_1^2\, (n_1\sigma_1^2 + n_2\sigma_2^2)^{-\frac{1}{2}}. \tag{3.1}$$

Note that the marginal bandwidth needed for an additional connection depends on traffic characteristics, mix, and load. For example, given the mean m and the variance σ^2 of a connection's cell arrival rate, the key factor determining the marginal bandwidth requirement is the variance of the aggregate traffic $n_1\sigma_1^2 + n_2\sigma_2^2$ currently on the links it traverses. Based on this example we conclude that, it may be "cheaper" (consuming less additional bandwidth) to route a new connection through a link whose current aggregate variance is large. In turn by selecting routes with minimum marginal bandwidth requirements one might make more resources available to incoming connections or other types of services.

This argument is in sharp contrast to typical routing policies that try to balance the loads on the network. Indeed a naive interpretation of (3.1) suggests that we might want to generate imbalances on various routes because they may result in reduced resource requirements, and are thus more "economical." In the usual circuit-switched environment, the bandwidth requirement for each traffic stream is fixed and independent of other traffic

[1]This approximation is based on the derivative of $c(n_1, n_2)$ with respect to a continuous variable n_1. It can be shown to be accurate when the aggregate variance is high.

currently sharing the links. By contrast, in packet-switched networks statistical multiplexing and the traffic mix on the link will affect the bandwidth requirements and thus judicious routing of connections may improve the system performance.

Suppose there are multiple traffic types carried on the same link. In this case, varying bandwidth requirements complicate admission decisions since the traffic mix in the system needs to be considered upon admitting new connections. Indeed such heterogeneity results in an admissible region with a nonlinear boundary. To simplify call admission, one can approximate the admissible region's boundary by a hyper-plane. By doing so, we linearly approximate the boundary and approximate the bandwidth requirement (per connection) of each type by a constant independent of the traffic mix. These constants are called the "effective bandwidth" for each traffic type, and are often proposed as a means to simplify call admission control. In [6], an argument is made that the boundaries may be linearized without any significant reduction in the potential "revenue" subject to the traffic loads. In fact, unless such linearizations are done carefully, i.e., paying attention to the traffic loads and routing policies to be used, they can result in a loss of revenue.

Indeed, routing in multiservice networks also introduces new challenges. On one hand, a traffic type may have different "effective bandwidth" associated with it on different links because of different linear approximations. Routing decisions may need to account for these differences in "effective bandwidth" in order to improve overall resource efficiency. On the other hand, the linearization of each link's admissible region may need to consider its impact on routing. Hence it may not be straightforward to extend the current "know how" on routing policies in circuit-switched networks to multiservice networks. In this chapter we shall show that routing decisions which account for the impact of statistical multiplexing, and the relative loads of various traffic types can significantly improve performance.

The balance of this chapter is organized as follows: In §3.2 we discuss the role of statistical multiplexing and nonlinear call admission regions associated with heterogeneous traffic. The problem of VP partitioning is analyzed in §3.3. Routing issues are discussed in §3.4 and are followed by simulations and summary.

3.2 Traffic Mix and Admissible Region

3.2.1 Characteristics of the Admissible Region

In general to assess the exact admissible numbers of connections on a given link, one would have to resort to either simulation or a significant amount of computation. In this chapter we will resort to a popular (conservative) approximation based on the Chernoff's bound in the context of bufferless multiplexing. Most of the ideas herein follow from the qualitative characteristics of the admissible region which are captured by this bound, see e.g., [32, 25]. Suppose N i.i.d. traffic streams are carried on a bufferless link, and each stream has a stationary cell arrival rate $A_i, i \in \{1, ..., N\}$. Assume that the link capacity is c and we require that the aggregate arrival rate to the link exceeds the capacity only rarely—with a probability no larger than δ. Based on the Chernoff's bound [3], the overflow probability is

upper-bounded by

$$\mathbb{P}(\sum_{i=1}^{N} A_i > c) \leq \exp(-\sup_{\theta>0}[c\theta - N\Lambda(\theta)]), \tag{3.2}$$

where $\Lambda(\theta) = \log(\mathbb{E}[\exp(\theta A_i)])$ is the logarithm of the moment generating function of A_i.

One can generalize the result to multiple traffic types and analyze statistical multiplexing with heterogeneous traffic. Let

$$A = \sum_{j=1}^{J} \sum_{i=1}^{n_j} A_{ji}$$

be the aggregate arrivals, where A_{ji} are independent random variables denoting the cell arrival rate of i th traffic flow of Type j, and $\Lambda_j(\theta) = \log(\mathbb{E}[\exp(\theta A_{ji})])$. In this case the Chernoff's bound and the overflow constraint give the following requirement:

$$\log \mathbb{P}(A > c) \leq -\sup_{\theta>0}\left[c\theta - \sum_{j=1}^{J} n_j \Lambda_j(\theta)\right] \leq \log \delta. \tag{3.3}$$

We shall define the *admissible region* $A(c)$ for a bufferless link with capacity c as the collection of vectors $\mathbf{n} = (n_1, ..., n_J) \geq 0$ which satisfy (3.3), i.e.,

$$A(c) = \left\{ \mathbf{n} | \mathbf{n} \geq 0 \text{ and } \sum_{j=1}^{J} \alpha_j(\theta^*) n_j - \frac{\log(\delta)}{\theta^*} \leq c, \text{for some } \theta^* > 0 \right\}, \tag{3.4}$$

where $\alpha_j(\theta^*) = \frac{\Lambda_j(\theta^*)}{\theta^*}$ and θ^* depends on the chosen $\mathbf{n}$ and is the argument which achieves the sup in (3.3) for a given $\mathbf{n}$.[2] Note that the θ^* is an implicit function of $\mathbf{n}$. Hence the bandwidth requirement (per connection) for each traffic type is affected by the traffic mix.

It has been shown in [26] that the complement of the admissible set in $\mathbb{R}_+^J$ is a convex region. It was suggested in [32, 25] that a linear approximation could thus be used to conservatively represent the boundary of the admissible region. However, a linear approximation of the admissible region boundary is not always accurate, and may affect performance.

The nonlinearity in the admissible region is called the *diversity cost* of the system and was quantified in [46]. It was pointed out in [16] that the total allocated bandwidth for a single type of aggregate traffic needs to exceed a critical value in order to make the traffic "statistically-multiplexable." Thus a minimum capacity is required to see the "economies of scale." The results in [16] also showed that combining "statistically-multiplexable" and "nonstatistically-multiplexable" traffic will create a nonlinear admissible region which can not be effectively approximated by a linear hyper-plane.

For example, let us consider the admissible region of traffic with Bernoulli cell arrival rate. The peak rate and mean rate are assumed to be p_j and m_j respectively, i.e., $\Lambda_j(\theta) = \log\left[1 + \frac{m_j}{p_j}(e^{\theta p_j} - 1)\right]$. For simplicity let us consider the case with two traffic types, i.e., $J = 2$. In Fig. 3.1 we plot the admissible region for a bufferless link with capacity 25 and two linear approximations. The two traffic types are On/Off with peak rates 1 and

[2] In the cases of interest, here θ^* is finite.

0.5 as well as mean-to-peak ratios 10% and 90% respectively, and $\delta = 10^{-9}$. Fig. 3.1 shows that the admissible region's boundary is convex and that linear approximations can be very inaccurate, suggesting that the effective bandwidth of each type is indeed state-dependent and sensitive to the traffic mix. In Fig. 3.2 we exhibit the admissible region when the capacity is increased 4 times. As shown in the figure, as the link capacity increases, the admissible region appears to become more "linear." Such changes are due to better multiplexing allowed by the higher link capacity.

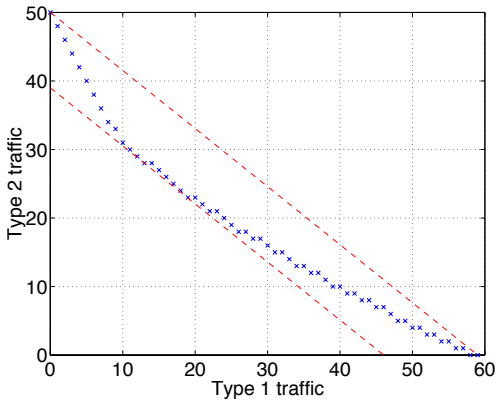

Figure 3.1: An admissible region and its linear approximations.

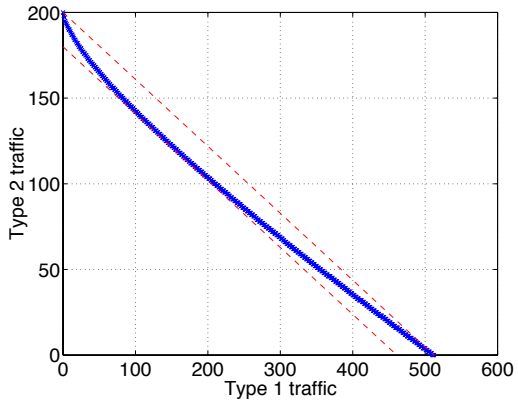

Figure 3.2: Another admissible region.

Next we consider the admissible regions $A(c)$ for various qualities of service $\delta = 10^{-3}, 10^{-6}$ and 10^{-9} in Fig. 3.3. The figure shows that, as the δ decreases, the region not only becomes smaller, but its boundary also becomes more "nonlinear." In fact, the increased "nonlinearity" is caused by less "efficient" multiplexing when the overflow probability requirement becomes more stringent. Note that Type 2 traffic is "nonstatistically-multiplexable" in this set-up [16], so the maximum admissible number of Type 2 traffic stays at 50. By contrast, Type 1 is "statistically-multiplexable", so its maximum admissible number changes as δ varies. As with the link capacity, the QoS requirement δ also affects the the nonlinear characteristics of the admissible region's boundaries.

We can conclude that the traffic mix indeed plays a role in the effectiveness of statistical multiplexing, even though its impact might be neglected when the number of connections, i.e., system capacity, becomes large or the QoS is relaxed. When carrying multiple traffic types in segregated VPs, where the bandwidth and number of connections are both moderate, the admissible region may indeed be nonlinear due to the diversity in burstiness and insufficient statistical multiplexing. In such cases the traffic composition in the VPs will be an important factor in dimensioning their capacities.

3.2.2 Linearization of the Admissible Region

To simplify call admission, it may be advantageous to linearize the admissible region. A linearization of $A(c)$ can be done by fixing an operating point $\mathbf{n}^*$ on the boundary and using the tangent hyper-plane to the admissible region's boundary at $\mathbf{n}^*$ as a linear approximation

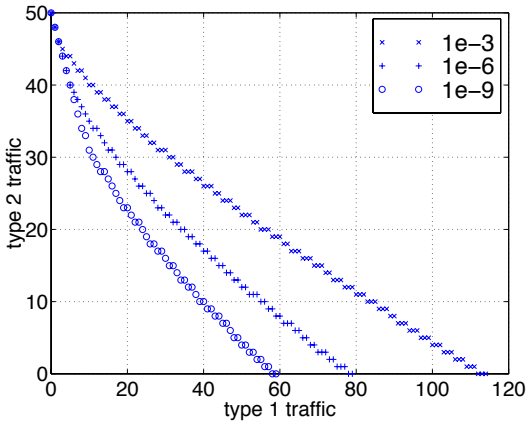

Figure 3.3: Admissible regions for various δ.

to the boundary, see Fig. 3.4. Therefore, θ^* in (3.4) will be determined by $\mathbf{n}^*$ and (3.4) can

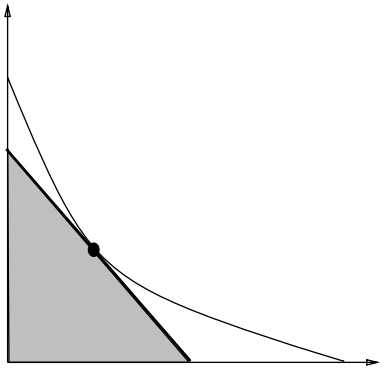

Figure 3.4: Linearization of the admissible region's boundary.

be rewritten as

$$A_{\mathbf{n}^*}(c) = \left\{ \mathbf{n} | \mathbf{n} \geq 0 \text{ and } \sum_{j=1}^{J} \alpha_j n_j \leq c' \right\}, \tag{3.5}$$

where $\alpha_j = \alpha_j(\theta^*)$ is the "effective bandwidth" of Type j traffic and $c' = c + \frac{\log(\delta)}{\theta^*}$ is the link's "effective capacity" with respect to θ^* (i.e., a linearization at $\mathbf{n}^*$.) Given such a linearization, the relative bandwidth requirements of each traffic type become fixed, resulting in a linearly constrained region and making call admission straightforward.

Note that it is essential to select the operating point $\mathbf{n}^*$ carefully. The selection may become more critical when the number of traffic types increases, which makes the admissible region more "nonlinear." As shown in (3.4), the effective bandwidth α_j of each traffic type depends on $\mathbf{n}^*$. Thus in principle we could exploit this dependence so as to change the relative resource requirements of each traffic type to the advantage of network. Note that a desirable operating point $\mathbf{n}^*$ is mainly determined by the offered loads from different traffic types which can be controlled through routing decisions. We will show in

§3.4 that careful selection of operating points could result in a better statistical multiplexing, hence reduces the total bandwidth requirements in the network.

3.3 VP Integration

We consider the problem of carrying multiple traffic types on VPs. It is generally believed that one should segregate traffic with different QoS requirements on their own VPs, see e.g., [23]. The intuition is that if we multiplex traffic with different QoS requirements on the same VP, then the overall QoS for the VP will be the most stringent QoS requirement. By providing a QoS that is more stringent than necessary to some traffic streams, we waste network resources. However, due to the nature of statistical multiplexing, it may still be more "economical" to put all traffic on the same VP. Careful allocation of VPs is essential to achieving high efficiency. In the following we use an example of two traffic types to illustrate the roles of statistical multiplexing and traffic mix.

3.3.1 Integration or Segregation?

Suppose there are N total connections which consist of a fraction f_1 of Type 1 flows and a fraction f_2 of Type 2 flows, where $f_1 + f_2 = 1$.[3] In order to get a qualitative understanding, we shall first resort to Gaussian traffic models for which an explicit expression for bandwidth requirements exists and then consider the popular On/Off traffic model. The cell arrival rates of each traffic type are modeled by Gaussian random variables with mean and variance (m_1, σ_1^2), (m_2, σ_2^2) respectively. We assume that the two traffic types are carried by a bufferless link and require cell loss ratios of 10^{-6} and 10^{-3} respectively. The goal is to decide whether to partition a link into two segregated VPs, or form a single shared VP. If we aggregate the flows on a single VP, then the loss ratio requirement is 10^{-6}. Otherwise we can have different QoS on separate VPs. Forming a single VP and providing a better QoS to Type 2 traffic is not necessarily a bad idea, since multiplexing may be more efficient when all flows are on the same VP. To understand this question, we need to assess the bandwidth requirements for the two options.

The total required capacities for these two cases are:

$$c_1 = N(f_1 m_1 + f_2 m_2) + k_1 \sqrt{N} \sqrt{f_1 \sigma_1^2 + f_2 \sigma_2^2} \tag{3.6}$$

$$c_2 = N(f_1 m_1 + f_2 m_2) + \sqrt{N} \left[k_1 \sqrt{f_1 \sigma_1^2} + k_2 \sqrt{f_2 \sigma_2^2} \right], \tag{3.7}$$

where k_1 and k_2 are the QoS parameters. For the Gaussian model, the tail distribution can be captured by the deviations from mean, and k_1, k_2 correspond to the multiples of standard deviation [41]. The bandwidth requirements of a single shared VP and segregated VPs are shown in (3.6) and (3.7) respectively. Without loss of generality, we assume $k_1 > k_2$. For the aforementioned QoS, k_1 and k_2 are 4.7534 and 3.0902 respectively. We are interested in a condition making $c_1 \leq c_2$, so it is advantageous to form a single VP and give Type 2

[3] For simplicity we assume f_1, f_2 are real numbers even though they should be restricted to multiples of $\frac{1}{N}$ such that Nf_1, Nf_2 are integers.

traffic a better QoS, rather than setting up two VPs. In other words, the benefit of statistical multiplexing outweighs the loss in over-provisioning for a better QoS.

Surprisingly, the condition depends only on $\frac{\sigma_1}{\sigma_2}$ and f_1, where N plays no role. Indeed for $c_1 \leq c_2$, we need that

$$k_1\sqrt{f_1(\frac{\sigma_1}{\sigma_2})^2} + k_2\sqrt{1-f_1} \geq k_1\sqrt{f_1(\frac{\sigma_1}{\sigma_2})^2 + 1 - f_1},$$

which can be rewritten as

$$\frac{\sigma_1}{\sigma_2} \geq \frac{k_1^2 - k_2^2}{2k_1k_2}\sqrt{\frac{1-f_1}{f_1}}. \tag{3.8}$$

Hence to answer the question of whether to integrate two types of traffic on the same VP, one needs to assess if the ratio of their standard deviations exceeds a threshold which depends on their QoS requirements and the traffic mix.

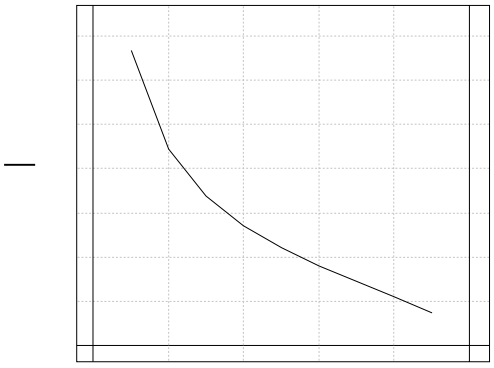

Figure 3.5: Traffic mix vs. $\frac{\sigma_1}{\sigma_2}$.

In Fig. 3.5 we plot the threshold on $\frac{\sigma_1}{\sigma_2}$, i.e., the right side of (3.8), as a function of the traffic mix f_1 with the aforementioned k_1 and k_2. The threshold defines the *integration* and *segregation* regions. For example, for $\frac{\sigma_1}{\sigma_2} = 0.6$, we should form a single VP when the fraction of Type 1 traffic exceeds 0.35. Otherwise, it is more efficient to have two VPs with different QoS.

Based on the regions shown in Fig. 3.5 it is clear that when f_1 is small, the ratio of $\frac{\sigma_1}{\sigma_2}$ needs to be large in order to make integration beneficial. An interpretation for this might be that we waste a larger amount of bandwidth in bringing a better QoS to Type 2 traffic (i.e., integration) when f_1 is small. Thus only when Type 1 traffic is "bursty", i.e., has high variance, can the benefit of better multiplexing outweigh the waste of bandwidth. Therefore, the threshold on $\frac{\sigma_1}{\sigma_2}$ should be larger when f_1 is small, as shown in Fig. 3.5. By contrast when f_1 is large, the Type 2 traffic becomes less significant, so the threshold on $\frac{\sigma_1}{\sigma_2}$ becomes less stringent, i.e., *integration* is desirable. The trade-offs of integration are captured by the curve in Fig. 3.5.

Note that $\frac{k_1^2-k_2^2}{2k_1k_2}$ serves as a scaling factor for the the curve in Fig. 3.5. If k_1 and k_2 are close, then the threshold on $\frac{\sigma_1}{\sigma_2}$ for integration becomes small, which increases the *integra-*

tion region. That is, one is indeed likely to integrate traffic with similar QoS requirements when minimizing the bandwidth reservation.

One can look at the problem of whether to integrate or segregate from a different perspective. Suppose n_1 and n_2 are the numbers of Type1 and Type 2 connections carried by the system. Inequality (3.8) can then be written as:

$$\frac{n_1\sigma_1^2}{n_2\sigma_2^2} \geq \left[\frac{k_1^2 - k_2^2}{2k_1k_2}\right]^2.$$

In Fig. 3.6, we plot the *integration* and *segregation* regions with respect to n_1 and n_2 with aforementioned k_1, k_2 and $\frac{\sigma_1}{\sigma_2} = 0.6$. The figure clearly indicates that the ratio of n_1 and n_2, rather than their magnitudes, determines the decision.

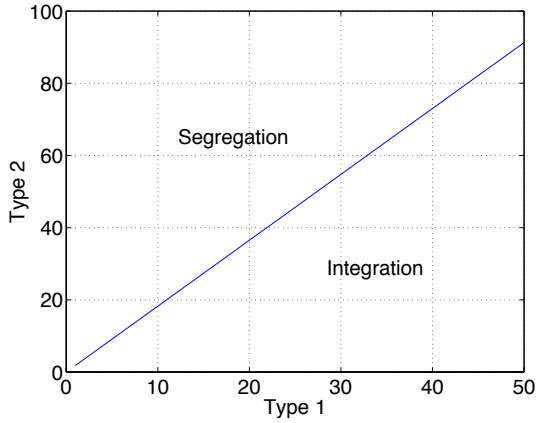

Figure 3.6: Integration and segregation regions.

Next we consider On/Off traffic models. For such models, it is difficult to derive a simple criterion such as (3.8) to determine whether to segregate or integrate heterogeneous traffic. The complexity is mainly due to the lack of a closed-form expression for the bandwidth requirement. Nevertheless we can show the trade-offs of integration numerically. In Fig. 3.7 we plot the difference in bandwidth requirements between integrating and segregating two types of On/Off traffic. The z-axis represents the difference, and a positive value means that segregation requires more bandwidth than integration. Fig. 3.8 indicates where integration or segregation are desirable.

The two types of On/Off traffic have mean rates 0.5 and 0.15 as well as peak rates 0.9 and 0.8. In addition, the overflow probability requirements are 10^{-9} and 10^{-3} for Type 1 and Type 2 respectively. As shown in the figure, there exists *integration* and *segregation* regions which are dependent on the traffic mix (n_1, n_2). As in the Gaussian case, the traffic mix determines whether it is advantageous to integrate two traffic types. An interesting observation is the small "plateau" region around $(n_1 = 0, n_2 = 0)$ in Fig. 3.7. This is because the number of connections is too small to achieve any statistical multiplexing, and the bandwidth reservation is done based on peak-rate allocation. Hence it makes no difference whether the two traffic types are integrated or segregated. Note that for Gaussian traffic, there is no such a threshold in the number of connections for achieving statistical multiplexing.

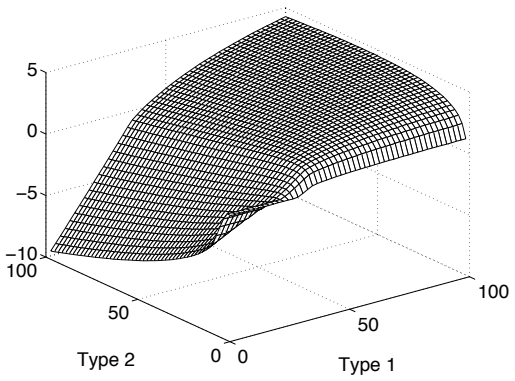

Figure 3.7: The difference in bandwidth requirements.

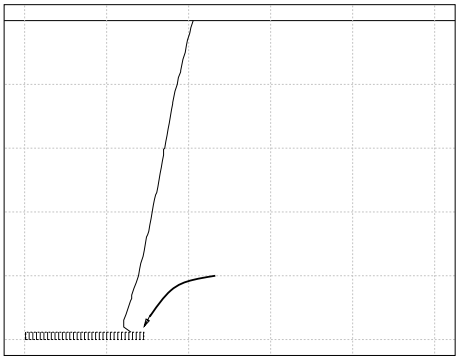

Figure 3.8: Integration and segregation regions.

As with the example of Gaussian traffic, the desirability of VP partitioning for On/Off traffic depends on the traffic mix and traffic characteristics, see Fig. 3.8. These results show that the dimensioning of VP bandwidth is not straightforward and accounting for the efficiency of multiplexing is essential in order to minimize the bandwidth reservation. For cases with more than two traffic types, these observations still hold, but a simple criterion for the best bandwidth partitioning is unlikely because of the increased complexity. Nevertheless optimum VP partitioning of multiple traffic types could be determined numerically or by simulation.

3.3.2 The Benefit of VP Integration

We have shown that it may be more efficient to carry traffic streams with different QoS requirements on the same VP, even though the VP is provisioned for the most stringent QoS requirement. Next we quantify the bandwidth savings as a function of the model's parameters using the Gaussian traffic example— this is given by

$$\begin{aligned} \Delta c &= c_2 - c_1 \\ &= \sqrt{N}\sigma_2 \left[k_1 \left(\sqrt{f_1 (\frac{\sigma_1}{\sigma_2})^2} - \sqrt{f_1 (\frac{\sigma_1}{\sigma_2})^2 + 1 - f_1} \right) + k_2 \sqrt{1 - f_1} \right]. \end{aligned} \quad (3.9)$$

Note that although the number of connections N does not play a role in the condition (3.8) determining whether it is beneficial to form a single VP, the bandwidth savings Δc grows as $\sqrt{N}$ as the number of connections increases.

Let us call Δc divided by $N(f_1 m_1 + f_2 m_2)$ the *normalized* bandwidth savings. It should be clear that the *normalized* bandwidth savings are proportional to $\frac{1}{\sqrt{N}}$, i.e., they become smaller as N increases. This result is not unexpected due to the increased efficiency of statistical multiplexing. Indeed when N is large, the *effective bandwidth* of each traffic stream decreases and approaches its mean rate, so the bandwidth utilization of both the integration and segregation schemes improve. Hence the (normalized) difference between (3.6) and (3.7) becomes smaller as N increases. However, the absolute magnitude of Δc still increases as N becomes large, and since the number of connections on a VP may be

moderate, statistical multiplexing gains may not effectively eliminate the impact of the traffic mix and heterogeneous QoS requirements. Thus a careful partitioning of VPs can lead to improvements in performance that perhaps should not be ignored.

3.4 Routing and Traffic Mix

We have shown that it may be more efficient to carry multiple traffic types on the same VP. With efficient multiplexing, one can minimize the total reserved bandwidth and in turn allow more connections to enter the system. However, using multiple VPs may be preferable (or necessary) in some circumstances. For example, one might choose to use multiple VPs through different links to increase reliability or due to capacity constraints. By doing so, we ensure that if a VP fails, the traffic can be quickly rerouted to other resources and the performance degrades smoothly. Given such requirements, one needs to determine how to route the heterogeneous traffic efficiently through multiple VPs. We will show that routing policies that account for statistical multiplexing characteristics have a significant impact on performance. For simplicity we shall assume that all VPs provide the same aggregate QoS, which are equal to or better than those requested by each traffic type. As shown in §3.3.1, integrating traffic and providing a better QoS on VPs could be advantageous. We first consider the impact of nonlinear admissible region's boundaries by using an example of two traffic types and two VPs. Then we consider a more general set up with linearized admissible region's boundaries on each VP.

3.4.1 Routing of Permanent VCs

Suppose there are two VPs between a given origin-destination pair, see Fig. 3.9. We first consider a simple static network flow problem for partitioning heterogeneous connections onto the VPs, geared at achieving good multiplexing. This problem is indeed an abstraction of routing permanent VCs on a VP network, where the goal is to minimize the total bandwidth reservation in order to maximize free capacity in the network. For simplicity, we again focus on routing two traffic (VC) types modeled by Gaussian distributions, but the solutions are based on general traffic characteristics leading to nonlinear admissible regions.

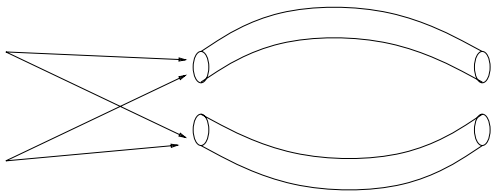

Figure 3.9: Two VPs and two traffic types.

Consider two VPs with bandwidth c_1, c_2 and two types of Gaussian traffic streams as in §3.3.1. Assume there are n_1 Type 1 and n_2 Type 2 streams, and both have the same overflow probability requirement. We will consider two problems: first, whether this load

is admissible, and second how to partition (or route[4]) these connections onto the two VPs in order to minimize the total bandwidth reserved on the two VPs. Note that the remaining bandwidth can be used to admit other traffic or carry best-effort traffic, so it is advantageous to minimize the bandwidth reservation.

Suppose a fraction[5] a of Type 1 and a fraction b of Type 2 traffic are sent to VP 1 and the remaining traffic is sent to VP 2, then the bandwidth requirements r_1, r_2 on each VP must satisfy the following inequalities:

$$c_1 \geq r_1 = n_1 a m_1 + n_2 b m_2 + k\sqrt{n_1 a \sigma_1^2 + n_2 b \sigma_2^2} \quad (3.10)$$

$$c_2 \geq r_2 = n_2(1-a)m_1 + n_2(1-b)m_2 + k\sqrt{n_1(1-a)\sigma_1^2 + n_2(1-b)\sigma_2^2}, \quad (3.11)$$

where k is the QoS parameter. The *admissibility* constraints (3.10) and (3.11) ensure that the bandwidth requirements subject to a pre-defined QoS do not exceed the capacities of the VPs. Let $V_1 = n_1 a\sigma_1^2 + n_2 b\sigma_2^2$ and $V_2 = n_1(1-a)\sigma_1^2 + n_2(1-b)\sigma_2^2$ be the variances of aggregate traffic on VP 1 and VP 2. The total capacity requirement is then given by $r_1 + r_2 = n_1 m_1 + n_2 m_2 + k(\sqrt{V_1} + \sqrt{V_2})$. An optimum partitioning policy is a pair of (a^*, b^*) such that (3.10) and (3.11) are satisfied, and $(\sqrt{V_1} + \sqrt{V_2})$ is minimized (or equivalently $r_1 + r_2$ is minimized).

Note that $V = V_1 + V_2 = n_1\sigma_1^2 + n_2\sigma_2^2$ is a constant representing the total *variance* of the aggregate traffic. We can represent V_1, V_2 as fractions of V, e.g., $V_1 = \alpha V$, $V_2 = (1-\alpha)V, \alpha \in [0,1]$. Since the contribution of m_1 and m_2 to $r_1 + r_2$ is fixed, the total bandwidth requirement is determined by the variance V_1, V_2 on each VP. Hence determining (a^*, b^*) is equivalent to picking V_1, V_2 or alternatively α such that $F(\alpha) = \sqrt{\alpha V} + \sqrt{(1-\alpha)V}$ is minimized. Since $F(\alpha)$ is concave and symmetric, see Fig. 3.10, $F(\alpha)$ is minimized when $\alpha = 1$ or $\alpha = 0$, i.e., send all traffic to one VP or the other.

However, with the admissibility constraints (3.10) and (3.11), sending all traffic to the same VP may not be possible if c_1 or c_2 are not big enough. Nevertheless, to minimize $F(\alpha)$, it is essential to keep α close to 1 or 0. In other words, a partitioning policy should distribute the total variance V in an *unbalanced* fashion so as to improve the efficiency of multiplexing.

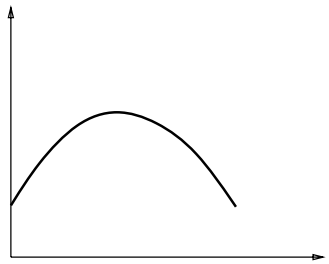

Figure 3.10: $F(\alpha)$.

[4]The terms routing and partitioning are used interchangeably in this discussion.

[5]For simplicity we assume a and b are real numbers, even though they should be restricted to the multiples of $\frac{1}{n_1}$ and $\frac{1}{n_2}$, such that $n_1 a$ and $n_2 b$ are integers.

Joint admissibility. We have shown in §3.2 that a nonlinear admissible region can be obtained from the link's bandwidth and traffic statistics. In Fig. 3.11, we represent the joint admissible regions for the two VPs: VP 1 on the first quadrant and VP 2 on the third quadrant. A point $S = (s_1, s_2)$ in the first quadrant represents a scenario where s_1 Type 1 and s_2 Type 2 streams are carried in VP 1. Similarly a point $T = (-t_1, -t_2)$ in the third quadrant represents t_1 Type 1 and t_2 Type 2 streams in VP 2. A line segment $\overline{ST}$ connected by two points in the two quadrants represents the scenario where $s_1 + t_1$ Type 1 streams and $s_2 + t_2$ Type 2 streams are jointly carried by the two VPs. In addition, a *feasible* $\overline{ST}$ is a line segment with end points S and T which are located on the admissible regions of VP 1 and VP 2 respectively.

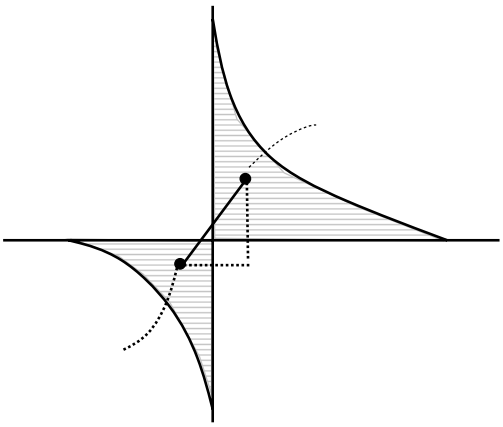

Figure 3.11: The joint admissible region.

Suppose $n_1 = N f_1$ and $n_2 = N f_2$, where $f_1 + f_2 = 1$. To determine whether (n_1, n_2) are *jointly admissible*, i.e., can be carried by two VPs subject to the QoS requirements, one needs to find points S and T such that $\frac{f_2}{f_1} = \frac{s_2+t_2}{s_1+t_1}$ and the length of $\overline{ST}$ is equal to $\sqrt{n_1^2 + n_2^2}$. In fact, every feasible $\overline{ST}$ with slope $\frac{f_2}{f_1}$ is associated with a partitioning policy for traffic with a mix f_1, f_2. There exists such an $\overline{ST}$ (may not be unique) having the largest length K^*. If $K^* \geq \sqrt{n_1^2 + n_2^2}$, then (n_1, n_2) are jointly admissible. Note that K^* depends on f_1, f_2 and, of course, the admissible regions of the two VPs.

Lemma 3.4.1 *With (strictly) convex admissible region's boundaries, in order to accommodate a maximum number of connections with a given mix f_1, f_2, a (unique) allocation exists such that one of the two VPs carries homogeneous traffic, and the other VP carries the remaining traffic.*

The intuition behind this lemma is as follows. Since the admission region's boundaries are convex, the longest $\overline{ST}$ with slope $\frac{f_2}{f_1}$ will contain at least one of the four intercept points of the admissible region's boundaries with the coordinate axis, as shown in Fig. 3.12. That is, one VP, which is determined by the slope of $\overline{ST}$ (i.e., the traffic mix), will carry a single type of traffic, and K^* can be determined accordingly. See §3.7 for a proof.

Fig. 3.12 shows that it is essential to partition the traffic carefully on each VP in order to accommodate a maximum number of traffic streams. For example, if the relative

traffic mixes are kept identical on each VP (i.e., $\overline{ST}$ passes the origin), then the length of $\overline{ST}$ will be much shorter than its optimum value, and a smaller number of traffic streams can be admitted.

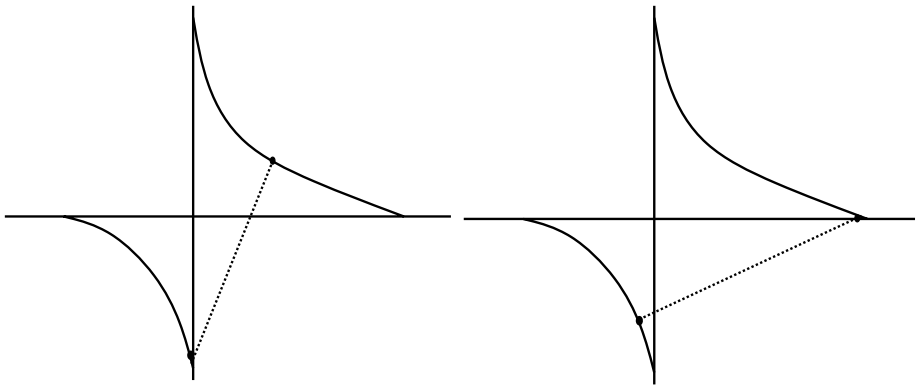

Figure 3.12: Examples of $\overline{ST}$ with maximum length.

Minimizing reserved bandwidth. Suppose $\sqrt{n_1^2+n_2^2} \leq K^*$, so the heterogeneous traffic with combined number of connections (n_1, n_2) are jointly admissible. Now we consider the problem of partitioning them onto two VPs in order to minimize the total reserved bandwidth. For the Gaussian traffic model, we have shown that one needs to distribute the variance V onto the two VPs in an unbalanced fashion in order to minimize the reserved bandwidth. We expect general traffic will have similar characteristics. Assume $c_1 > c_2$, then one needs to determine the location of $\overline{ST}^*$ which will maximize V_1 subject to two constraints: $n_1 = s_1 + t_1$ and $n_2 = s_2 + t_2$. Since $V_1 = s_1\sigma_1^2 + s_2\sigma_2^2$, it is clear that the end point S of $\overline{ST}^*$ will be on the admissible region boundary of VP 1 in order to make V_1 large.

Lemma 3.4.2 *To minimize the bandwidth reservation for the Gaussian traffic model, one would fill the VP of larger bandwidth, and leave idle bandwidth, if any, on the smaller VP. Moreover, one of the following two scenario occurs: (1) the larger VP carries all the traffic, or (2) one of the VPs carries homogeneous traffic.*

Fig. 3.13 shows examples of $\overline{ST}^*$ with different slopes $\frac{f_2}{f_1}$. Note that S is on the admissible region's boundary, and either S or T is located on the coordinate axis. See §3.7 for detailed proof.

Lemma 3.4.2 suggests that traffic streams are packed into the larger VP, leaving the smaller VP partially occupied. The traffic fractions f_1, f_2 affect how traffic streams are packed onto VPs, i.e., the location of $\overline{ST}^*$, in order to minimize reserved bandwidth.

3.4.2 Routing and Linearization of Admissible Region

In §3.2.2 we discussed how a convex admissible region's boundary can be approximated by a tangent hyper-plane at a given point $\mathbf{n}^*$ on the boundary. Such a linearization results in a notion of "effective bandwidth" for each traffic type. Here we discuss the impact that such linearizations might have on the routing policy.

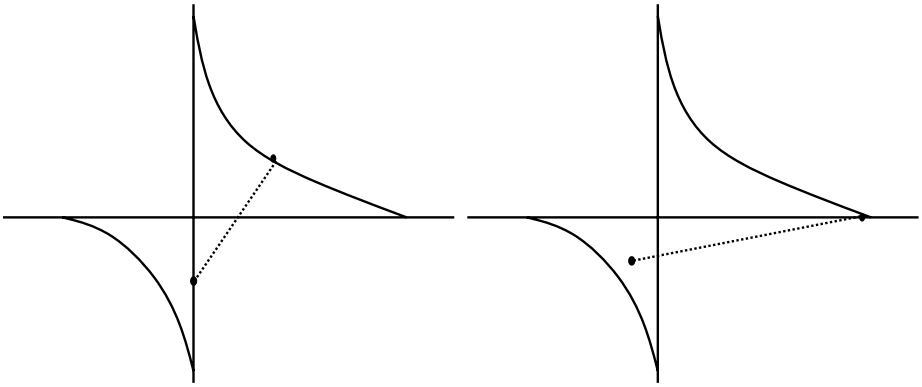

Figure 3.13: Examples of optimum $\overline{ST}$ locations.

Suppose the capacities of VP 1 and VP 2 are c_1, c_2 and $c_1 \geq c_2$. Let s_1^* and s_2^* be the maximum admissible number of homogeneous Type 1 and Type 2 connections which can be carried by VP 1 based on its linearized admissible region. Similarly let t_1^* and t_2^* be the admissible number of the two traffic types on VP 2. The slopes of two linear admissible region's boundaries are given by $-\frac{s_2^*}{s_1^*}$ and $-\frac{t_2^*}{t_1^*}$ respectively.

Without loss of generality, we consider the following two cases: (1) $\frac{s_2^*}{s_1^*} = \frac{t_2^*}{t_1^*}$ and (2) $\frac{s_2^*}{s_1^*} < \frac{t_2^*}{t_1^*}$. Note that the scenario of $\frac{s_2^*}{s_1^*} > \frac{t_2^*}{t_1^*}$ is equivalent to Case (2) since the designations of Type 1 and Type 2 are interchangeable.

Case 1: Two admissible region's boundaries are parallel, see Fig. 3.14.

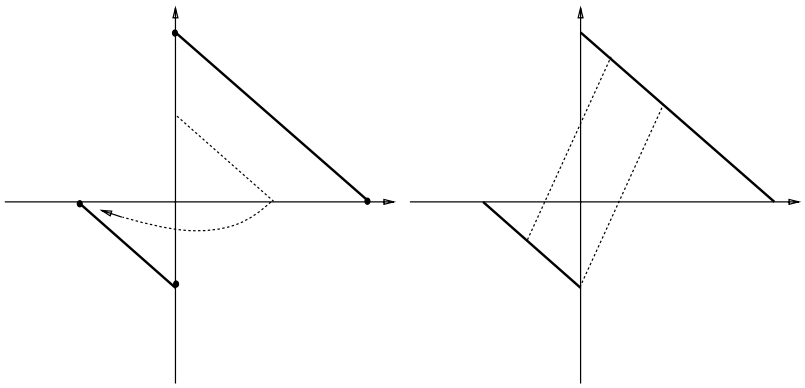

Figure 3.14: Parallel boundaries.

Let us now consider the maximum number of heterogeneous connections which can be carried by the two VPs with fractions f_1 and f_2 of each type. As shown in Fig. 3.14, it is clear that for any feasible $\overline{ST}$ with slope $\frac{f_2}{f_1}$, every $\overline{ST}$ with S and T on the boundaries will have the same length of K^* since the two boundaries are parallel. In other words, the maximum number of permanent VCs can be admitted onto the VPs by multiple $\overline{ST}$. By contrast to Lemma 3.4.1, neither VP has to carry homogeneous traffic in order to admit the maximum number of connections.

Case 2: In this case, the boundaries of admissible regions are no longer parallel, see Fig. 3.15.

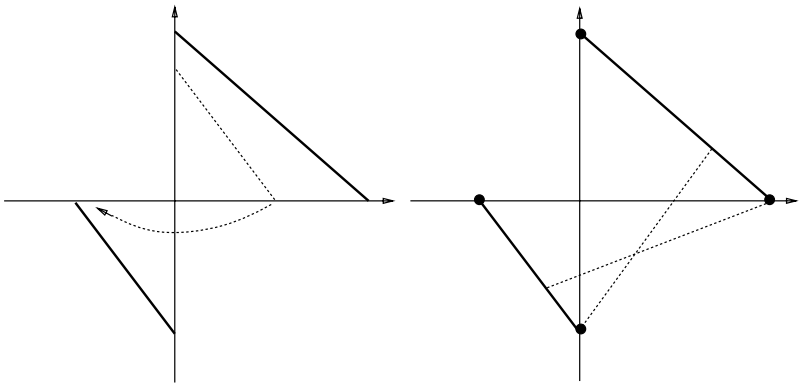

Figure 3.15: Non-parallel boundaries.

Indeed the "distance" between the two boundaries is monotonically increasing (or decreasing) along the boundaries. Based on Fig. 3.15, it is clear that for any feasible $\overline{ST}$ with slope $\frac{f_2}{f_1}$, the maximal allocation $\overline{ST}^*$ will have S or T at points A_1 and/or A_2. As a result, in order to carry the maximum number of heterogeneous connections with fractions f_1 and f_2, either one (or both) of the following two scenarios is possible:

1. VP 1 carries homogeneous Type 1 traffic and VP 2 carries the remaining traffic ($\overline{ST}^*$ passes point A_1);
2. VP 2 carries homogeneous Type 2 traffic and VP 1 carries the remaining traffic ($\overline{ST}^*$ passes point A_2).

Thus there exists a fixed preferable assignment of traffic types to VPs, i.e., Type 1 traffic is mainly carried by VP 1 and Type 2 traffic is mainly carried by VP 2.

Note that such a "preference" is due to the difference in $\frac{s_2^*}{s_1^*}$ and $\frac{t_2^*}{t_1^*}$, which results from the linearizations of admissible regions. Therefore, different linearizations of admissible regions may result in different first choice preference in these VP assignments.

3.4.3 Routing with Multiple VPs and Traffic Types

So far we have analyzed in detail the problem of routing two traffic types onto two VPs. Next we consider the routing problem for the network shown in Fig. 3.16, where the admissible regions of each VP are linearized. For simplicity we shall assume that all VPs provide the same aggregate QoS, which are equal to or better than those requested by each traffic type.

The problem of determining the maximum admissible number of connections can be formulated as a linear programming problem where we relax the integer constraint on the number of connections. We assume that the total number of connections of each type are big enough that the rounding errors to the closest integers are negligible. Thus an optimum solution to the linear programming problem is close to an optimum solution to the *integer* programming problem.

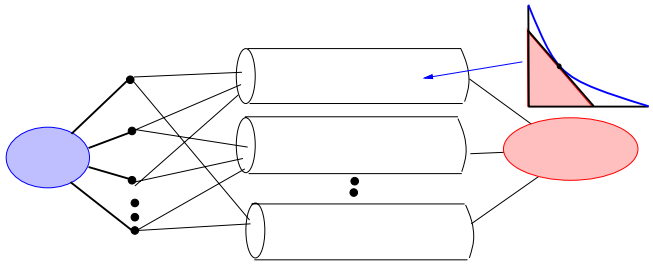

Figure 3.16: Problem set-up.

Suppose there are J traffic types to be routed onto P VPs with linearized admissible regions, where $\alpha_{p,j}$[6] is the "effective bandwidth" of Type j traffic on VP p. We shall assume that every VP can carry any traffic type. Let $n_{p,j}$ denote the number of Type j connections carried by VP p. The capacity constraints are then given by the following inequalities:

$$\sum_{j=1}^{J} \alpha_{p,j} n_{p,j} \leq c_p, \ \forall p, \tag{3.12}$$

where c_p is the effective capacity of VP p, see (3.5).

Next we consider a set of constraints associated with the pre-defined traffic mix. Let $f_j, \ j = 1,..,J$ be the fraction of each traffic type and $\sum_{p=1}^{P} n_{p,j}$ denote the total number of Type j connections carried by all VPs. To ensure the traffic mix constraint is met, we require that

$$\frac{\sum_{p=1}^{P} n_{p,1}}{f_1} = \frac{\sum_{p=1}^{P} n_{p,2}}{f_2} = ... = \frac{\sum_{p=1}^{P} n_{p,J}}{f_J}. \tag{3.13}$$

We can rewrite these as a set of equality constraints[7]:

$$\sum_{p=1}^{P} n_{p,1} - \frac{f_j}{f_1} \sum_{p=1}^{P} n_{p,j} = 0, \forall j \neq 1. \tag{3.14}$$

The goal is to maximize the total number of connections $\sum_{j=1}^{J} \sum_{p=1}^{P} n_{p,j}$. Given the nature of this objective function, the constraints in (3.12) will hold with equality since we have relaxed $n_{p,j}$ to be continuous. The linear programming problem is then formulated as follows:

$$\begin{array}{lll} \max & \sum_{j=1}^{J} \sum_{p=1}^{P} n_{p,j} & \\ \text{subject to} & \sum_{j=1}^{J} \alpha_{p,j} n_{p,j} = c_p, & \forall p \\ & \sum_{p=1}^{P} n_{p,1} - \frac{f_j}{f_1} \sum_{p=1}^{P} n_{p,j} = 0, & \forall j \neq 1 \\ & n_{p,j} \geq 0 & \forall p, j \end{array} \tag{3.15}$$

It is clear that the feasible set is a *polytope* and a solution to this linear programming problem will be a vertex of the polytope [30, 37]. The constraints can be represented in a

[6] For simplicity the argument θ^* is omitted. Nevertheless it should be clear that $\alpha_{p,j}$ depends on θ^* and the linearization points.

[7] We assume that $\sum_{p=1}^{P} n_{p,j}$ are big enough such that the error of rounding $n_{p,j}$ to the closest integers are negligible. Hence the traffic mix can be roughly maintained.

# VP (P)	# types (J)	M	# homogeneous VP
2	2	3	≥ 1
2	3	4	≥ 0
3	2	4	≥ 2
3	3	5	≥ 1
3	4	6	≥ 0

Table 3.1: Number of VPs carrying homogeneous traffic.

compact form as $A\vec{\mathbf{n}} = b, \vec{\mathbf{n}} \geq 0$, where $\vec{\mathbf{n}} = \{n_{p,j} | p = 1,...,P, j = 1,...J\}^T$ is a vector representing the number of connections on every VPs. In addition, we let $\mathbf{n}_p = \{n_{p,j} | j = 1,...J\}^T$ denote the number of connections on VP p.

Note that A is an $M \times N$ matrix, where $M = P + J - 1$ and $N = PJ$.[8] It can be shown that a vertex $\vec{\mathbf{n}}^*$ of the feasible set contains at most rank(A) non-zero components [30] and rank$(A) \leq M$. In other words, there exists a solution to (3.15) which contains at least $N - M$ zero components, where a zero component, i.e., $n_{p,j} = 0$, means that VP p does not carry Type j traffic. This observation sheds light on how the traffic might be carried by the VPs in order to maximize the admissible number of connections subject to a pre-defined mix.

For example, one can determine the number of VPs carrying homogeneous traffic by assessing the number of non-zero components on $\vec{\mathbf{n}}^*$. Since each VP will carry some traffic, there exists at least one $n^*_{p,j} > 0, j = 1,...,J$ for each VP p. Therefore, at most rank$(A) - P$ additional positive $n^*_{p,j}$ can be spread out across P VPs. If rank$(A) < 2P$, then it is clear that some VPs have to carry homogeneous traffic. In Table 3.1, we show the lower bound on the number of homogeneous VPs for various numbers of VPs and traffic types. For instance, when 2 traffic types are routed onto 3 VPs, we find that at least 2 VPs will carry homogeneous traffic.

One can interpret $\min[P, \text{rank}(A) - P]$ as the maximum number of heterogeneous VPs. Given the fact that rank$(A) \leq J + P - 1$, so $\min[1, \frac{J-1}{P}]$ is an upper bound on the *fraction* of VPs carrying heterogeneous traffic. As the ratio $\frac{J}{P}$ decreases, i.e., the number of VPs is greater than the number of traffic types, more VPs are likely to carry homogeneous traffic.

In addition, $1 + \frac{J-1}{P}$ also gives an upper bound on the "average" number of traffic types carried on each VP, suggesting that only a small subset of the J traffic types will be present on each VP when P is large. In particular, when $P > J$, the "average" number of traffic types on each VP is smaller than 2.

The key insight is that it is not advantageous for each VP to carry *all* traffic types in this heterogeneous set-up. To maximize the throughput, only a small number of traffic types, or even homogeneous traffic will be present on each VP. This suggests that, in practice, optimized multiservice networks with sufficient routing diversity might end up looking like multiple logical networks which are segregated by service type.

[8] Here we ignore the case of $P = 1$ or $Q = 1$, thus $N > M$.

Routing with nonlinear admissible region. We have discussed the characteristics of a routing policy for multiple VPs with linearized admissible regions. Next we show that the same characteristics still hold even if the admissible region's boundary of each VP is convex.

Given nonlinear admissible regions, the maximization problem can be similarly formulated but with new capacity constraints. The capacity constraints are now determined by the nonlinear admissible region's boundary and the feasible set F associated with these capacity constraints is no longer a polytope. Nevertheless, there still exists an optimal allocation $\vec{\mathbf{n}}^*$ which maximizes the objective function $f(\vec{\mathbf{n}}) = \sum_{j=1}^{J}\sum_{p=1}^{P} n_{p,j}$ and satisfies the capacity as well as the traffic mix constraints.

Notice that the components of $\vec{\mathbf{n}}^*$ consist of P points (i.e., $\mathbf{n}_p^*$), and each $\mathbf{n}_p^*$ is on the admissible region's boundary of VP p. Therefore one can linearize the admissible region of VP p at the point $\mathbf{n}_p^*$. Based on these linearized admissible regions, we can formulate a linear programming problem similar to (3.15) with a new feasible set $F^{'}$ which is a polytope.

Note that $F^{'}$ is a subset of the original feasible set F because linearization is a conservative approximation of the admissible region of each VP, see e.g., Fig. 3.4. Hence $\vec{\mathbf{n}}^*$ must also be a maximizer to this linear programming problem with respect to $F^{'}$. Indeed, otherwise one finds a contradiction to the optimality of $\vec{\mathbf{n}}^*$ for the nonlinear problem.

In addition, based on basic properties of linear programming, there exists a vertex (might be $\vec{\mathbf{n}}^*$ itself) $\vec{\mathbf{v}} \in F^{'}$ such that $f(\vec{\mathbf{v}}) = f(\vec{\mathbf{n}}^*)$ [37, 30]. Therefore, $\vec{\mathbf{v}}$ is also a maximizer of $f(\cdot)$ with respect to both F and $F^{'}$. Since $\vec{\mathbf{v}}$ is a vertex of polytope $F^{'}$, it also contains at most $P + J - 1$ non-zero components and all the aforementioned characteristics of the routing policy follow exactly, e.g., the characteristics for the number of homogeneous and the "average" number of traffic types on each VP are similar.

3.4.4 Routing with Dynamic Call Arrivals

Suppose that calls of each traffic type arrive as Poisson processes with rates λ_j and each type of call has an arbitrary holding time distribution with mean μ_j^{-1}. We will consider the routing problem with the objective of minimizing blocking probabilities. The results in §3.4.3 suggest that connections need to be allocated to VPs carefully and appropriate traffic mixes need to be maintained on each VP in order to maximize the admissible number of connections. However, it is not always possible to maintain desirable traffic mixes on each VP in a dynamic environment. Based on our observation on static route assignment that each VP carries only a small number of traffic types when the admissible number is maximized, we propose a simple alternate routing algorithm which approximately maintains the traffic mix on each VP around desirable operating points.

Each traffic type will be assigned a *sequence* of potential choices of VPs. The routing algorithm then selects the first VP in the sequence that can carry the connection. Alternate routing usually comes in many flavors which mainly depend on how the routing sequence is chosen [20]. We shall design the routing sequences to account for statistical multiplexing and traffic mixes on the VPs.

Suppose the offered loads for each traffic type are $\rho_j = \frac{\lambda_j}{\mu_j}$. Let $f_j = \frac{\rho_j}{\sum_j \rho_j}$ denote the fraction of Type j traffic. One can formulate a static route assignment problem subject

to these fractions f_j and solve for a maximum allocation $\vec{\mathbf{n}}^*$. The sequence of VPs to be attempted for Type j traffic is chosen based on $\vec{\mathbf{n}}^*$ and the following rules:

- VP p precedes VP q on the sequence if $\vec{\mathbf{n}}^*$ indicates that VP p carries only Type j traffic and VP q carries heterogeneous traffic.
- VP p precedes VP q on the sequence if (1) $n^*_{p,j} > n^*_{q,j}$, or (2) $n^*_{p,j} = n^*_{q,j}$ and $p < q$ (break ties).

For example, consider a three types and three VPs example where the components $n^*_{p,j}$ of $\vec{\mathbf{n}}^*$ are given by $(n^*_{1,1}, n^*_{1,2}, n^*_{1,3}) = (7, 28, 0)$, $(n^*_{2,1}, n^*_{2,2}, n^*_{2,3}) = (28, 0, 0)$, and $(n^*_{3,1}, n^*_{3,2}, n^*_{3,3}) = (0, 7, 36)$. The sequence of VPs to be used by the routing algorithm are $2 \to 1 \to 3$, $1 \to 3 \to 2$, and $3 \to 1 \to 2$ for Type 1, 2, and 3 respectively.

A repacking routing policy. In general the order of call arrivals might affect the routing decision and thus the efficiency of network resource. Ideally, the "optimal" routing policy without favoring any particular traffic type is one that recomputes the best routing decisions whenever the system status changes, e.g., new connections arrive. In particular, such a policy might reroute (repack) connections in order to admit a new request. By doing so, networks can admit as many connections as possible and make the most of network resources. However, repacking may not be feasible for the connections that are already in-progress. Nevertheless, the blocking probabilities of such a policy can be used as a measure of comparison in evaluating the proposed routing algorithm.

3.5 Simulation Results

In this section we use simulations to evaluate the performance of the proposed algorithm under two scenarios: two traffic types and two VPs, and three traffic types and three VPs. We compare the call blocking probabilities achieved by a repacking algorithm to a variety of routing algorithms including Least-Loaded Route (LLR) and Minimum-Resource Route (MRR) algorithms. Although LLR is usually claimed to be efficient for single service networks [23, 27], our simulations showed that in the multiservice context it can not achieve good performance because it fails to account for the impact of traffic mix and statistical multiplexing.

3.5.1 Two Traffic Types and Two VPs

We first consider the case of two traffic types and two VPs. As shown in Table 3.1, we found that at least one VP carries homogeneous traffic. This property leads to different routing sequences of VPs for each traffic type, i.e., a sequence $1 \to 2$ for one type and $2 \to 1$ for the other type. As a result, each type is assigned a separate primary VP and a connection will be sent to its primary VP if it is available. If the primary VP is unavailable, the other VP is tried. The connection will be blocked if the second trial fails.

The proposed algorithm is compared with other algorithms under various traffic loads which are denoted by ρ_1 and ρ_2 respectively. In the simulations, both traffic types are

assumed to be On/Off with peak rates 1 and 0.5 as well as mean-to-peak ratios 10% and 90% respectively. The two VPs have an identical capacity of 25, with nonlinear admissible regions. The routing algorithms we compare are shown in Fig. 3.17 and explained below.

A. Balanced-load scheme without re-trial. Connections of each type are sent to each VP with equal probability. The loads sent to each VP are $\frac{1}{2}(\rho_1 + \rho_2)$.

B. Balanced-load scheme. This is the same as Algorithm A except that if the selected VP is unavailable, the other VP is tried. That is, each connection is assigned a routing sequence $1 \rightarrow 2$ or $2 \rightarrow 1$ with equal probability regardless of its type.

C. Aggregated-load scheme. A VP will be assigned as the primary VP for *both* traffic types. If the selected VP is unavailable, the other VP is tried. For example, the routing sequence $1 \rightarrow 2$ is always attempted by new connections regardless of type.

D. Proposed algorithm. Different traffic types are assigned different routing sequences: $1 \rightarrow 2$ or $2 \rightarrow 1$.

E. Repacking scheme. Connections are repacked if this will permit a new connection to be admitted.

In addition to the above routing algorithms, we also simulated two dynamic routing algorithms: LLR and MRR. In LLR, a newly arriving connection is sent to the VP which has the largest free capacity[9]. By contrast, in MRR, a connection is sent to the VP where minimum additional bandwidth is required to carry the new connection given the current load and statistical multiplexing. If the selected VP is unavailable, the other VP is tried.

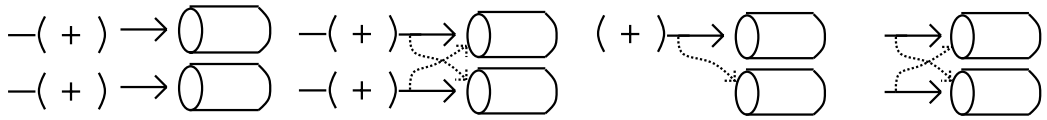

Figure 3.17: Four routing algorithms.

Comparison of blocking probabilities. The simulation results is shown in Table 3.2, where p_1 and p_2 are the blocking probabilities of Type 1 and Type 2 traffic respectively. Table 3.2 shows that Algorithm A has the worst overall blocking probabilities. This is because Algorithm A uses static load sharing without retrying the other path if the selected path is unavailable. Hence a connection may be blocked unnecessarily. The difference in blocking probabilities of Algorithms A and B strongly suggests that re-trial is worthwhile in order to improve the performance. Excluding the repacking algorithm (E), Algorithm D has the smallest blocking probabilities for both traffic types under all traffic load combinations, which echos the observation found in the static route assignment problem, that unbalanced traffic mixes on the VPs will improve the efficiency of statistical multiplexing and the usage

[9]The free capacity is the difference between VP's capacity and the minimum bandwidth required to carry the connections already on the VP.

Loads (ρ_1, ρ_2)	A	B	C	D	E	LLR	MRR
(40,40)	p1=0.0597 p2=0.0837	p1=0.03099 p2=0.05027	p1=0.03120 p2=0.04743	p1=0.00221 p2=0.00222	p1=0.00081 p2=0.00158	p1=0.03024 p2=0.05149	p1=0.02891 p2=0.04283
(45,30)	p1=0.0269 p2=0.0420	p1=0.00791 p2=0.01477	p1=0.00804 p2=0.01406	p1=0.00044 p2=0.00048	p1=0.00013 p2=0.00014	p1=0.00760 p2=0.01580	p1=0.00720 p2=0.01212
(30,45)	p1=0.0498 p2=0.0627	p1=0.02280 p2=0.03149	p1=0.02289 p2=0.03043	p1=0.00148 p2=0.00149	p1=0.00076 p2=0.00137	p1=0.02258 p2=0.03201	p2=0.02257 p2=0.02767
(50,25)	p1=0.0197 p2=0.0326	p1=0.00462 p2=0.00913	p1=0.00465 p2=0.00878	p1=0.00069 p2=0.00095	p1=0.00033 p2=0.00037	p1=0.00438 p2=0.00981	p1=0.00419 p2=0.00765
(25,50)	p1=0.0542 p2=0.0655	p1=0.02796 p2=0.03405	p1=0.02853 p2=0.03282	p1=0.00289 p2=0.00290	p1=0.00169 p2=0.00279	p1=0.02867 p2=0.03458	p1=0.02971 p2=0.02990

Table 3.2: The comparison of blocking probabilities.

of VP capacities. Notice that the call blocking probabilities of Algorithm D are close to those achieved by the repacking algorithm (E).

Intuitively an efficient multiplexing reduces the bandwidth reservation on each VP, which in turn reduces the chance of blocking. This is why the proposed algorithm has the best performance. Algorithms B and C have roughly the same blocking probabilities due to the fact that the VPs end up having similar traffic mixes. The traffic load ratios on each VP of Algorithm B and C are roughly equal to the original offered loads ratio $\frac{\rho_1}{\rho_2}$, so multiplexing is not as efficient as that in Algorithm D. For example, when the traffic load is $(45, 30)$, the blocking probabilities of Algorithm D are 0.00044 and 0.00048, which are more than a order of magnitude smaller than those achieved by Algorithm B.

We find that LLR and MRR are inferior to the proposed algorithm, and their blocking probabilities are close to those of Algorithm B for both types of traffic. Since LLR and MRR fail to account for the traffic mix in making routing decisions, the traffic load ratios are roughly equal to the original offered load ratio $\frac{\rho_1}{\rho_2}$ on each VP. Therefore, LLR and MRR can not achieve as efficient multiplexing as the proposed algorithm does, which leads to higher blocking probabilities. These observations provide some insight into how one should extend the routing policies in circuit-switched networks to multiservice packet-switched networks. Obviously the effect of traffic mix and multiplexing on the selected routes should be accounted for when routing connections.

3.5.2 Three Traffic Types and Three VPs

Next we compare two alternate routing algorithms with the repacking scheme for the problem of routing three traffic types onto three VPs. For simplicity we assume the admissible regions of VPs are linearized. The "effective bandwidth" and the "effective capacity" of VPs are shown in Table 3.3.

E. Repacking scheme. Connections are repacked if this will permit a new connection to be admitted.

F. Balanced-load scheme. Three sequences of possible VP choices, $1 \rightarrow 2 \rightarrow 3$, $2 \rightarrow 3 \rightarrow 1$, and $3 \rightarrow 1 \rightarrow 2$ are randomly assigned to each connection with equal probability regardless of the traffic type. Hence the VPs are attempted in a rotary fashion.

VP	Capacity	Effective Bandwidth		
		Type 1	Type 2	Type 3
1	100	2	2.5	3
2	100	3	4	5
5	100	2	2	2

Table 3.3: Capacity and the effective bandwidths.

Traffic loads (ρ_1, ρ_2, ρ_3)	E	F	G
(33,33,33)	p1=0.00150 p2=0.00190 p3=0.00190	p1=0.00868 p2=0.01180 p3=0.01418	p1=0.00174 p2=0.00238 p3=0.00306
(42,42,15)	p1=0.00080 p2=0.00110 p3=0.00110	p1=0.00461 p2=0.00613 p3=0.00748	p1=0.00129 p2=0.00179 p3=0.00223
(60,24,24)	p1=0.00340 p2=0.00430 p3=0.00430	p1=0.01773 p2=0.02329 p3=0.02825	p1=0.00528 p2=0.00724 p3=0.00891
(50,30,15)	p1=0.00035 p2=0.00045 p3=0.00035	p1=0.00321 p2=0.00429 p3=0.00531	p1=0.00055 p2=0.00077 p3=0.00094
(90, 9, 9)	p1=0.00240 p2=0.00240 p3=0.00240	p1=0.00637 p2=0.00818 p3=0.00983	p1=0.00226 p2=0.00301 p3=0.00370
(3,45,45)	p1=0.00270 p2=0.00370 p3=0.00440	p1=0.00692 p2=0.01027 p3=0.01265	p1=0.00266 p2=0.00425 p3=0.00547

Table 3.4: Comparison of blocking probabilities.

G. The proposed algorithm. Each connection is assigned a sequence of possible VP choices based on its type. The sequences are computed using the rules discussed in §3.4.4.

The comparison of blocking probabilities are shown in Table 3.4. The blocking probabilities of the proposed algorithm (G) are pretty close to those of Algorithm E. In fact, the routing sequences of the proposed algorithm are intended to approximate the effect of "repacking." These results show that "appropriate" traffic mixes on each VP were indeed achieved by carefully selecting the routing sequence for each traffic type.

The load-balancing algorithm (F) resulted in worse blocking probabilities than the proposed algorithm. For example, when the traffic loads are $(54, 30, 15)$, the p_1 of Algorithms E and F are 0.00035 and 0.00048 respectively, but p_1 of Algorithm G is 0.00321. The difference is almost a order of magnitude. The discrepancy in blocking probability is due to the traffic mix which in turn affects the usage of VP capacities.

Traffic loads (ρ_1, ρ_2, ρ_3)	Routing Sequence		
	Type 1	Type 2	Type 3
(15,30,60)	$2 \to 1 \to 3$	$1 \to 2 \to 3$	$3 \to 1 \to 2$
(18,30,60)	$2 \to 1 \to 3$	$1 \to 2 \to 3$	$3 \to 1 \to 2$
(15,30,72)	$2 \to 1 \to 3$	$1 \to 2 \to 3$	$3 \to 1 \to 2$
(15,30,48)	$2 \to 1 \to 3$	$1 \to 2 \to 3$	$3 \to 1 \to 2$
(18,24,60)	$2 \to 1 \to 3$	$1 \to 2 \to 3$	$3 \to 1 \to 2$
(12,24,72)	$2 \to 1 \to 3$	$2 \to 1 \to 3$	$3 \to 1 \to 2$

Table 3.5: Routing sequence under various traffic loads.

3.5.3 Robustness & Linearization

In our proposed algorithm, the routing sequences of each traffic type depend on the *relative* magnitudes of the components on $\vec{\mathbf{n}}^*$. When the offered traffic loads (mix) are perturbed, the sequences are usually preserved. Hence the proposed routing algorithms are robust against gentle fluctuations of traffic loads or measurement errors. For example, when the traffic loads in §3.5.2 are perturbed around an operational point $(15, 30, 60)$ by 20%, most of the computed routing sequences stay unchanged, see Table 3.5.

Suppose in the simulations of §3.5.2 the linearization of admissible regions are identical on all the VPs, i.e., the effective bandwidths of each traffic type are identical across the VPs, then Algorithms E, F and G will have the same performance. In fact, the choice of routing sequence becomes irrelevant. In this scenario, there exists a *resource pooling* effect, see e.g., [33] and three VPs "look" like a big VP of larger capacity with the same effective bandwidth for each traffic type. Hence it might be advantageous to linearize the admissible regions identically on all VPs in order to simplify the routing algorithms.

In [6], an argument was made that the boundaries of admissible regions can be linearized without significantly reducing the achievable "revenue" in high-capacity networks. However, unless the linearization points are chosen carefully and appropriate routing policies are used to keep the networks around the desired operational regime, linearization can lead to a loss in overall revenue. Note that a desirable linearization point $\mathbf{n}_p^*$ is mainly determined by the offered loads from different traffic types which can be controlled through routing decisions.

3.6 Summary

In this chapter we have attempted to clarify problems related to resource allocation and routing in integrated services networks. In particular, we were motivated by questions that arise in managing heterogeneous traffic types with possibly different QoS requirements using VPCs as an intermediate resource management layer.

The first natural question that arises is whether heterogeneous traffic with different QoS requirements should be segregated on distinct VPs or aggregated on a single VP but given the most stringent QoS requirement. Based on a simple model our analysis shows

that the answer is certainly not straightforward, i.e., in some cases, it is advantageous to aggregate while in others it is better to segregate. For this model a criterion for making such decisions is derived which depends on the traffic characteristics, traffic mix, and QoS requirements. Similar behavior is likely to hold for more general setups, where the essential tradeoff is between achieving improved statistical multiplexing by aggregating but losing efficiency due to provisioning for the most stringent QoS.

Given a QoS requirement, such as cell loss at a link, the set of admissible numbers of connections of various types is very likely to have a nonlinear boundary. This reflects the role that the traffic mix plays in determining the effectiveness of statistical multiplexing of such traffic. We argue that although such nonlinearities disappear as the link bandwidths become larger, they are nevertheless present in systems multiplexing moderate numbers of connections, as might be expected when network resources are partitioned using VPs. The second natural question is to consider the impact of statistical multiplexing and relative traffic mix in routing connections through the network.

Indeed we argue that statistical multiplexing, might encourage "users" to route connections along heavily loaded links or VPs, since increased loads are likely to reduce the marginal bandwidth requirements for the new connection. This observation motivated us to consider whether this approach might also be beneficial from the network's point of view. In a network with heterogeneous traffic types, we found that an aggressive strategy seeking the most loaded resource is however not as effective as a more careful allocation of resources that accounts for the traffic characteristics and mix. Indeed we show that in both a static and a dynamic routing model with heterogeneous traffic types, careful allocations or decisions can lead to a significant decrease in the required bandwidth or the blocking probability that connections will experience.

In summary, we have obtained following observations and insight:

- The "marginal" bandwidth requirement for an additional connection depends on the current load and the mix of traffic on the path. Judicious route selections may lead to a better system performance.

- VP dimensioning depends on both the traffic characteristics and the QoS requirements. Integrating traffic with different QoS requirements on the same VP is not necessarily a bad idea, since the increased benefit of multiplexing across traffic types may outweigh the possible loss caused by providing a QoS that is more stringent than necessary to some traffic.

- The "effective bandwidth" for each traffic type is determined by the linearization of admissible region's boundaries. The routing decisions may be affected by the possibly different "effective bandwidths" associated with the same traffic type on different links. Hence the linearization can affect the routing decision. Alternatively, the routing decisions may change the traffic loads on each link, which in turn could also affect the linearization.

- It is not always advantageous for the VPs connecting a given source-destination pair to carry *all* traffic types. To maximize the throughput, only a small number of traffic

types, or even homogeneous traffic are present on each VP. This suggests that in practice multiservice networks might end up looking like multiple logical networks which are segregated by service types.

- A simple alternate routing algorithm which accounts for traffic mix can achieve a better blocking probability than LLR.

3.7 Appendix

3.7.1 Proof of Lemma 3.4.1

Proof: To determine the maximum length of $\overline{ST}$ with slope $\frac{f_2}{f_1}$, we first draw an auxiliary line passing the origin O with a slope $-\frac{f_1}{f_2}$, see Fig. 3.18. Let $\|\overline{ST}\|$ denote the length of $\overline{ST}$. It is obvious that $\|\overline{ST}\| = \|\overline{SM}\| + \|\overline{MT}\|$. Suppose $\|\overline{OM}\| = x$, both $\|\overline{SM}\|$ and $\|\overline{MT}\|$ are (strictly) convex functions of x since the admissible region's boundary is (strictly) convex. In turn, $\|\overline{ST}\|$ is also a (strictly) convex function of x. Hence the problem is indeed to maximize a (strictly) convex function of x and the (unique) maximizer will occur at the boundary of feasible interval of x, namely the optimum $\overline{ST}$ contains intercept points of the joint admissible region's boundaries with the coordinate axis. That is, one VP will carry a single type of traffic. ■

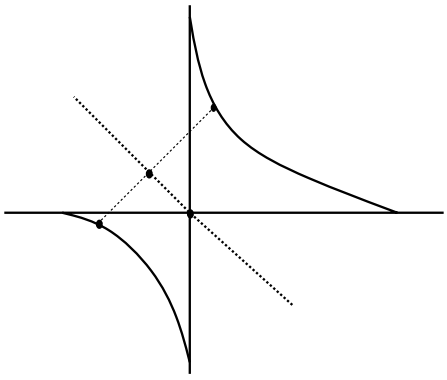

Figure 3.18: Determining the length of $\overline{ST}$.

3.7.2 Proof of Lemma 3.4.2

Proof: It is clear that when VP 1 is large enough to carry all the traffic, the total bandwidth reservation is minimized if all the traffic are sent to VP 1. Below we consider what happens when this is not the case. It is easy to see that $S = (s_1, s_2)$ will be located on the admissible region's boundary in order to maximize $V_1 = s_1\sigma_1^2 + s_2\sigma_2^2$. Along the boundary, s_2 is a convex function of s_1. It follows that $V_1 = s_1\sigma_1^2 + s_2\sigma_2^2$ is also a convex function of s_1 along the admissible region's boundary, and V_1 is maximized when S is at both ends of the boundary. However, recall that n_1, n_2 are fixed, namely the slope and length of $\overline{ST}$ are fixed, so the location of $\overline{ST}$ might be constrained by the admissibility of point T on the third

quadrant, see Fig. 3.13 for examples. Hence, V_1 is maximized when $\overline{ST}$ is located on the boundary of its feasible set, i.e., at least one of the two points S and T is on the coordinate axis. That is, one VP will carry a single type of traffic. ■

Chapter 4

Explicit Rate Flow Control of ABR Traffic

4.1 Introduction

Asynchronous Transfer Mode (ATM) networks are geared towards supporting and integrating a variety of communication services which might broadly be divided into those based on *reservation,* e.g., Constant and Variable Bit Rate (CBR,VBR) services and *best effort* services, such as Unspecified Bit Rate (UBR) and to some extent Available Bit Rate (ABR) services. Among the latter, ABR service promises to play an important role in supporting high bandwidth data as well as Internet traffic, such as TCP traffic. The rationale for including ABR is to provide an economical and flexible way to carry data traffic, as might be needed to simplify adoption of ATM to support delay adaptive real-time applications [4]. From the service provider's point of view, ABR traffic can be used to enhance utilization by directing sources to make the most of the network's available capacity subject to minimum cell rate and cell loss guarantees.

It has been shown [42, 43] that TCP performs poorly over ATM networks when there is congestion and ATM cells from multiple TCP packets are dropped. Significant performance degradation results from "corrupted" TCP packets since they 1) waste network bandwidth because they are useless to TCP upon arriving at their destination hosts, and 2) trigger retransmission from the sources to make the effective throughput even lower. Several algorithms have been proposed for dealing with this throughput collapse problem, such as packet discard strategies and tuning of the TCP flow control mechanism [43]. However, instead of tuning the concurrent feedback loops of TCP and ABR, we believe that it is important to ensure low cell loss inside the networks, so as to avoid the throughput collapse. This chapter aims to analyze resource requirements and proposes simple design rules to provide ABR service with controlled loss in a dynamic environment.

ABR service is likely to use rate-based feedback flow control[1], i.e., adjusting the transmission rates of sources based on the current network state. Feedback control in the context of wide and even local area networks is plagued by the potentially large source

[1]Rate-based feedback flow control had been chosen by ATM Forum in 1995.

transmission rates relative to the propagation (as well as processing and queuing) delays in the system, making the responsiveness of such mechanisms sluggish and typically requiring large buffers to absorb traffic fluctuations. In general, when ABR sessions have relatively long bursts of traffic to send, one might hope to have enough time to properly adapt their transmission rates—such sessions are said to be *greedy.* By contrast, traffic with small burst sizes relative to the control time scales, e.g., some WWW connections, are said to be *bursty* sessions. Feedback control would typically be ineffective for individual bursty connections though it might still work reasonably well on an aggregated basis.

There are two types of rate control mechanisms. The network can determine and enforce a bound on the transmission rate for each ABR session based on the current state of the system or may rely on exchanging minimal (binary) congestion indications to incrementally adjust source transmission rates; for a survey see [4, 40, 28] and for a representative analysis see [5]. These two mechanisms are not incompatible and in fact future networks might use a natural combination of binary feedback adjustments with explicit rate bounds to adjust source transmission rates, e.g., the Proportional Rate Control Algorithm (PRCA) and Enhanced PRCA (EPRCA) discussed in [4]. In the PRCA the source continuously decreases its cell rate in a multiplicative fashion—proportional to its current cell rate. It increases its cell rate linearly only after it receives a *positive* feedback which indicates the network is not congested. If the feedback is *negative* or lost, the source will keep decreasing its cell rate. EPRCA, an improved version of the PRCA, provides an Explicit Rate (ER) feedback as a dynamic upper bound on the cell rate calculated by the PRCA. In other words, the new cell rate will be the minimum of the calculated rate, based on single bit feedback, and the most recent explicit rate received from the network.

There are other variations of the PRCA, aimed at enhancing its performance, e.g., with respect to fairness. For example, the network can send congestion indications selectively to particular sources rather than all sources [48]. However, slow adaptation to the network's state and instability are two problems with algorithms using single bit feedback. In [5] it was shown that such control mechanisms result in an oscillating queue and traffic flows. By contrast, using explicit rate feedback allows switches to specify a desirable traffic rate, so sources can rapidly adapt their traffic.

Several algorithms have been suggested for computing the explicit rate. In general, the computation is based on the queue length, see e.g., [11, 17, 2] and/or the arrival rates, see e.g., [48, 29, 10]. The former uses the difference between queue length and a target queue threshold to adjust the explicit rate. Algorithms using arrival rates to compute the explicit rate do so by dividing the capacity among sessions in a "fair" manner without considering the queue length. In order to divide capacity "fairly" among sessions, the switches need to maintain rate/state information for each session. The computational complexity incurred by the per-source accounting is an issue in implementation.

This chapter extends an approach first proposed in [8], which was inspired from [19]. In §4.2 we propose a computationally efficient algorithm for computing the explicit rate by considering queue length, source activity, and available capacity. In practice one might expect a mix of traffic with various burst scale properties to use ABR service, and it is of interest to understand the impact that both the unpredictable nature of source transmis-

sions and spare capacity in the network will have on flow control. In §4.2 we also consider a simple model for a flow control mechanism which accounts for such fluctuations. In §4.3 we show that the lossless guarantee can be met by reserving a minimum capacity and buffer in the bottleneck node. We also show in §4.4 that when sources are greedy, the queue length and explicit rate will be asymptotically stable around the equilibrium points. In principle one might argue that by statistically multiplexing a large number of *bursty* ABR sessions on a given link, one can achieve relatively high utilizations. In §4.6 we articulate this point of view and suggest how one might hope to optimistically use this to deal with bursty ABR sessions. The balance of this chapter includes further discussion of the proposed flow control mechanism, preliminary simulations, and consideration of implementation requirements in the context of ABR service.

4.2 Explicit Rate Flow Control—a Fluid Model

In this section we consider a "fluid model" wherein for simplicity we assume that the instantaneous transmission rates of sources and links are well-defined. In practice these correspond to windowed estimates of the cell rates in the system. We further assume the Minimum Cell Rates (MCR) of ABR sessions are zero. The case of positive MCR will be discussed in §4.5.3.

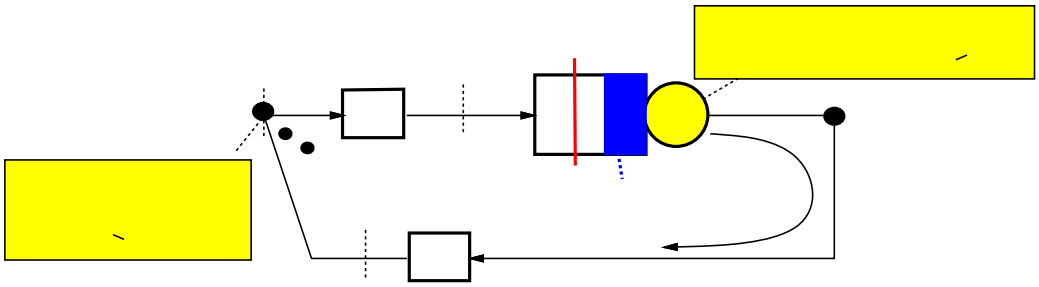

Figure 4.1: Network bottleneck model.

Network model. We simplify our analysis of the network dynamics by considering a single "bottleneck" buffered link shared by at most n_{max} concurrent ABR sessions. We shall assume that the worst case delay, including both propagation and queuing time, from the j^{th} source to the bottleneck is τ_j^f and then back is τ_j^b for a total round trip delay of τ_j. Let $\tau = \max_{j=1}^{n_{max}} \tau_j$ and $\Delta\tau = \max_{j=1}^{n_{max}} [\tau - \tau_j]$ be the worst case round trip delay and worst case delay discrepancy respectively.

The bottleneck model is shown in Fig. 4.1, where $c(t)$ denotes the instantaneous capacity available at a bottleneck link. We further assume that the rate at which the available capacity can decrease is lower-bounded, i.e., $dc(t)/dt \geq -\rho$. Changes in $c(t)$ are primarily due to fluctuations in the aggregate bandwidth requirements of current reserved services, e.g., VBR sessions sharing the link, as well as changes in the *number* of such sessions. As the number of sessions sharing high capacity network links becomes large, one might hope that statistical averaging would result in slow fluctuations in the aggregate bandwidth

requirement of reserved services relative to the link capacity. Drops in the available capacity due to sudden increases in the number of VBR connections may be controlled by exerting flow control on call admissions. Both of these mechanisms affect the magnitude of ρ and are further discussed in §4.6 and §4.7.

Source model. Let $r_j(t)$ denote the instantaneous transmission rate for the j^{th} session at time t. Throughout its lifetime a source's transmission rate can never exceed the most current explicit rate indication $e(t)$ [2] received from the network, i.e., $r_j(t) \leq e(t-\tau_j^b)$. We introduce a threshold r^* to discriminate among sources with different "activity" levels.

- If $e(t-\tau_j^b) \geq r^*$, we say a session is "on" if its current transmission rate exceeds r^*, i.e., $r_j(t) > r^*$, otherwise the session is said to be "off." Sessions which are "on" are contending for available capacity.
- If $e(t-\tau_j^b) < r^*$, then the available capacity of the link is low, that is, the link appears to be congested and all sessions are considered to be "on."

Moreover we will assume that once a session's transmission rate exceeds r^*, at most a linear rate of increase, g, can be supported. Fig. 4.2 shows the characteristics of the source transmission rate as discussed above.

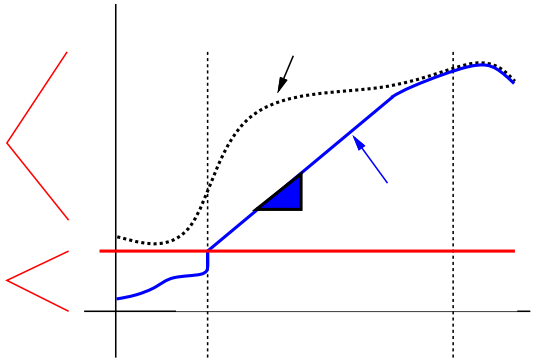

Figure 4.2: Source model characteristics.

This mechanism captures a possibly desirable initial cell rate wherein sources can typically jump start their transmission up to a rate r^* after being idle and may thereafter ramp up linearly. Note that sources desiring to transmit at a rate below the threshold r^* may do so freely, which should expedite short bursty transmissions. By contrast a persistent session wishing to transmit at high rate may certainly do so but must give the network time to detect that it is becoming a major contender for capacity in the network, hence the ramp-up above r^* is constrained. In addition, such linear ramp-up constraints might be desirable in order to integrate single-bit and explicit-rate flows control mechanisms in a heterogeneous environment.

[2] Here $e(t)$ means the explicit rate , not the envelope function of a deterministically constrained traffic.

Flow control mechanism. The dynamics of the bottleneck queue are given by :

$$\dot{q}(t) = \begin{cases} \sum_{j=1}^{n_{max}} r_j(t-\tau_j^f) - c(t), & q(t) > 0, \\ \left[\sum_{j=1}^{n_{max}} r_j(t-\tau_j^f) - c(t)\right]^+, & q(t) = 0, \end{cases} \tag{4.1}$$

with $[x]^+ = \max[x, 0]$ and where the transmission rate of each session is bounded by the latest explicit rate received, i.e., $r_j(t) \leq e(t-\tau_j^b)$.

The explicit rate $e(t)$ is computed based on the current network state which includes the queue length, $q(t)$, the current available capacity, $c(t)$, and *delayed information* about the current number of sources that are "on." The number of "on" sources is given by

$$\hat{n}(t) = \sum_{j=1}^{n_{max}} 1\{r_j(t-\tau_j^f) \geq r^*\} + 1\{e(t-\tau_j) < r^*\}\ , \tag{4.2}$$

where $1\{\cdot\}$ is the indicator function. The first term corresponds to the sources with transmission rates exceeding r^*, while the second term corresponds to the scenario where the bottleneck node appeared to be congested, indicated by $e(t-\tau_j) < r^*$. In that scenario a session was assumed to be "on" regardless of its rate.

The explicit rate is computed so that the net input into the network approximately tracks a delay-free reference model for the queue dynamics given by

$$\dot{q}(t) = f(q(t)) \qquad \text{e.g., } f(q(t)) = -k(q(t)-q^*),$$

where the drift $f(q(t))$ is selected to drive the reference queue towards a target level q^*. The bottleneck queue computes $e(t)$ so as to approximate this drift assuming the sources that were "on" (with at least one being on, i.e., $\hat{n}(t) \vee 1$) will transmit at this new rate, i.e.,

$$f(q(t)) = e(t)(\hat{n}(t) \vee 1) - c(t) \ \Rightarrow\ \boxed{e(t) = \tfrac{f(q(t))+c(t)}{\hat{n}(t)\vee 1}.} \tag{4.3}$$

Thus the explicit rate is based on the available capacity as well as the queue's state, which as suggested in [1] is necessary to ensure stability. Note that a single $e(t)$ is computed for all ABR sessions carried by the bottleneck link, which significantly reduces the implementation complexity of this algorithm.

4.3 Guaranteeing No Loss and Positive $e(t)$

In this section we show a minimum amount of buffer b_{min} must be reserved at the potential bottleneck in order to ensure no loss. In addition, a minimal service rate c_{min} also needs to be reserved in order to guarantee that the explicit rate is non-negative since the transmission rate can not be less than zero. For the remainder of this chapter, we assume the drift function is linear, i.e., $f(q(t)) = -k(q(t)-q^*),$ where $k > 0$, and show that

$$\boxed{b_{min} \ = \ q^* + \tfrac{w}{k} + (kq^* + w)\tau \quad \text{and} \quad c_{min} \ = k(b_{min} - q^*),} \tag{4.4}$$

where $w = \tau\rho + n_{max}[r^* + g\tau]$.

Lemma 4.3.1 *The aggregate arrival rate into the queue at time t is bounded by $f(q(t-\tau))+c(t-\tau)+n_{max}[r^*+g\tau]$.*

Proof: Consider the aggregate traffic rate reaching the bottleneck queue at time t. We can subdivide the contributing sources into the $\hat{n}(t-\tau)$ that were thought to be "on" at time $t-\tau$, and those that were thought to be "off":

$$\begin{aligned}\sum_{j=1}^{n_{max}} r_j(t-\tau_j^f) &= \underbrace{\sum_{j=1}^{n_{max}} 1\{r_j(t-\tau-\tau_j^f)\geq r^* \text{ OR } e(t-\tau-\tau_j)<r^*\}\, r_j(t-\tau_j^f)}_{\text{"on" sources}} \\ &+ \underbrace{\sum_{j=1}^{n_{max}} 1\{r_j(t-\tau-\tau_j^f)<r^* \text{ AND } e(t-\tau-\tau_j)\geq r^*\}\, r_j(t-\tau_j^f)}_{\text{"off" sources}},\end{aligned}$$

where the arguments of indicator functions are based on the two cases captured in (4.2).

Using the explicit rate constraint and the bound on the ramp-up of sources' transmission rates, we can establish that

$$\begin{aligned}r_j(t-\tau_j^f) &\leq r_j(t-\tau+\tau_j^b)+g[(t-\tau_j^f)-(t-\tau+\tau_j^b)] \\ &\leq e(t-\tau)+g\Delta\tau.\end{aligned}$$

Now distinguishing between the sources which were "on" and accounting for the worst case linear growth of "off" sessions from r^*, we get the following bound on the aggregate rate into the queue at time t:

$$\begin{aligned}\sum_{j=1}^{n_{max}} r_j(t-\tau_j^f) &\leq \hat{n}(t-\tau)[e(t-\tau)+g\Delta\tau]+[n_{max}-\hat{n}(t-\tau)][r^*+g\tau] \\ &\leq f(q(t-\tau))+c(t-\tau)+\hat{n}(t-\tau)g\Delta\tau+[n_{max}-\hat{n}(t-\tau)][r^*+g\tau] \\ &\leq f(q(t-\tau))+c(t-\tau)+n_{max}[r^*+g\tau]. \qquad (4.5)\end{aligned}$$

■

Notice that the upper bound on arrival rate includes transient bursts caused by the "off" sessions turning "on" between $t-\tau$ and t. The feedback mechanism guarantees that the arrival rates of such bursting sessions will be regulated no later than τ seconds after the bottleneck node detects them. From the queue's perspective, such bursts will last at most τ seconds and only contain a finite amount of traffic, so the sudden increase in queue length due to such bursts can be upper-bounded.

Using (4.5) and (4.1), as well as the variability constraint on the available capacity, we find the following differential inequality :

$$\begin{aligned}\dot{q}(t) &\leq f(q(t-\tau))+c(t-\tau)-c(t)+n_{max}[r^*+g\tau] \\ &\leq f(q(t-\tau))+\tau\rho+n_{max}[r^*+g\tau] \\ &\leq f(q(t-\tau))+w, \qquad (4.6)\end{aligned}$$

where $w=\tau\rho+n_{max}[r^*+g\tau]$. Note that ρ is the variation of available capacity, or equivalently the burstiness of VBR connections sharing the same link. In the worst case, $\tau\rho$

corresponds to the maximum variation in the arrival rate of VBR traffic over τ seconds. Thus the worst case queue growth is driven by three factors: 1) the drift computed from delayed queue information, 2) the *unexpected* burstiness of VBR connections, and 3) the *unexpected* bursting ABR connections.

The bound in (4.6) is conservative as it is based on the assumption that capacity is dropping by $\rho\tau$ and n_{max} sessions are turning "on" at the same time. Nevertheless, it can be used to derive an upper bound on queue length.

Lemma 4.3.2 *The queue length is upper-bounded by* $q_{max} = q^* + \frac{w}{k} + (kq^* + w)\tau$.

Proof: Note that the upper bound (4.6) on $\dot{q}(t)$ depends on the queue length at $t - \tau$. If $q(t-\tau) \geq q^* + \frac{w}{k}$, it follows that $f(q(t-\tau)) \leq -w$ and $\dot{q}(t) \leq 0$. In other words, the queue length has to stop increasing τ seconds after it exceeds $q^* + \frac{w}{k}$. Consider arbitrary time intervals during which the queue length exceeds $q^* + \frac{w}{k}$; we call such periods "overshooting cycles." We first show an upper bound on the queue length over such "overshooting cycles."

Without loss of generality, let $t = 0$ be the beginning of an "overshooting cycle" and $q(0) = q^* + \frac{w}{k}$. Since the queue length exceeds $q^* + \frac{w}{k}$, we know $\dot{q}(t) \leq 0$, for $t > \tau$ on the "overshooting cycle." To compute the maximum queue length q_{max} over an "overshooting cycle," it suffices to consider the worst case queue growth, see Fig. 4.3, on the interval $[0, \tau]$:

$$\begin{aligned} q(t) &\leq q^* + \frac{w}{k} + \int_0^\tau f(q(t-\tau)) + w \, dt \\ &\leq q^* + \frac{w}{k} + \int_0^\tau (kq^* + w) \, dt \\ &= q^* + \frac{w}{k} + (kq^* + w)\tau = q_{max}. \end{aligned} \tag{4.7}$$

We have shown that q_{max} is an upper bound on the queue length over an "overshooting

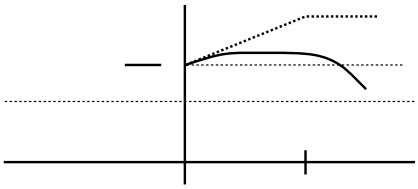

Figure 4.3: An upper bound on the queue length.

cycle." For intervals other than "overshooting cycles," the queue length does not exceed $q^* + \frac{w}{k}$, thus q_{max} is an upper bound on the queue length. ■

In order to guarantee that no loss occurs, we need to reserve a buffer of size

$$b_{min} = q^* + \frac{w}{k} + (kq^* + w)\tau, \text{ where } w = \tau\rho + n_{max}[r^* + g\tau]. \tag{4.8}$$

In addition, we need to reserve a minimum capacity c_{min} to guarantee non-negative $e(t)$. Indeed, to ensure $e(t) \geq 0$, we require that $f(q(t)) + c(t) \geq 0$. Since the minimum of $f(q(t))$ corresponds to the largest queue length, a capacity $c_{min} = -f(q_{max}) = k(b_{min} - q^*)$ is sufficient to ensure non-negative $e(t)$.

By modifying the drift function $f(\cdot)$, we can change the minimum buffer/capacity requirements. This is further discussed in §4.7.3.

4.4 Asymptotic Stability

In this section we discuss the stability of the proposed algorithm in a fixed environment, i.e., the number of sessions and capacity are both fixed and all the sessions are greedy. In particular we analyze the impact of the round-trip delay, drift function $f(\cdot)$, and the ramp-up constraints on the stability of the system. We assume that n greedy sessions share a link with fixed capacity $c(t) = c \geq c_{min}$. In other words, the sources attempt to track the latest explicit rate indications. Hence in the following analysis, r^* plays no role in distinguishing source activity—sources are always assumed to be "on."

4.4.1 Linear Feedback

We shall first relax the ramp-up constraint on the sources and discuss the stability of the system. If we take derivative on both sides of "$e(t) = \frac{f(q(t))+c}{n}$" with respect to t, it follows that $\dot{e}(t) = \frac{-k\dot{q}(t)}{n}$. Now substituting into (4.1), we find that $e(t)$ is governed by a delay-differential equation given by

$$\dot{e}(t) = -\frac{k}{n}[\sum_{i=1}^{n} e(t-\tau_i) - c]. \tag{4.9}$$

An equivalent system for (4.9) is shown in Fig. 4.4, where $G(s) = \frac{1}{s}D(s)$, and $D(s) = \sum_{i=1}^{n} e^{-s\tau_i}$ models the feedback delays. The model shown in Fig. 4.4.(b) is a linear feedback

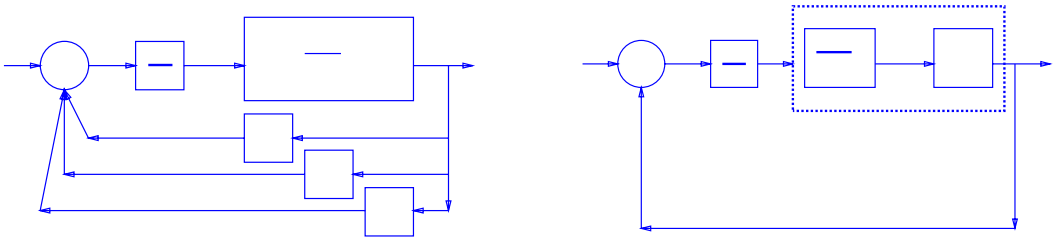

Figure 4.4: The control system model and its equivalent.

control system and its stability can be verified based on the frequency response of $G(s)$ using the Nyquist criterion [50]. By contrast, one can take a transfer function approach and consider the location of the transfer function's poles, i.e., the roots of $1+\frac{k}{n}G(s)$, see e.g., [11, 2, 17]. However, it is usually nontrivial to find the poles of such transfer functions. In the following we consider the stability of this system based on how the frequency response of $G(s)$ encircles the Nyquist point $z = -\frac{n}{k}$. The Nyquist plot of $G(s)$ is determined by

$$G(j\omega) = \sum_{i=1}^{n} \frac{e^{-j\omega\tau_i}}{j\omega} = \sum_{i=1}^{n} [-\frac{\sin(\tau_i\omega)}{\omega} - j\frac{\cos(\tau_i\omega)}{\omega}].$$

Since $\left|\frac{\sin(\omega\tau_i)}{\omega}\right| \leq \tau_i$, it follows $Re(G(j\omega)) \geq -\sum_{i=1}^{n}\tau_i$. Thus the Nyquist plot of $G(j\omega)$ always resides on the right hand side of the vertical line $z = -\sum_{i=1}^{n}\tau_i$ on the complex plane. If we choose the Nyquist point to be $z = -\frac{n}{k} < -\sum_{i=1}^{n}\tau_i$, then $G(j\omega)$ will stay away from circling the Nyquist point at all. As a result, $\frac{k}{n}\sum_{i=1}^{n}\tau_i < 1$ is a sufficient condition to ensure the asymptotic stability of the system.

Since the ramp-up of sources is not constrained, the c_{min} in §4.3 can not be applied to ensure $e(t) \geq 0$. Thus we have a different requirement on c_{min} to ensure that $e(t)$ is well-defined, i.e., does not become negative, and thus prevents our model from assuming negative source rates.

Lemma 4.4.1 *For the system in Fig. 4.4.(a), $c \geq c_{min} = k^2 q^* \tau$ is a sufficient condition to ensure $e(t) \geq 0$.*

Proof: See Appendix 4.10.1.

4.4.2 Nonlinear Feedback

Next we discuss the stability of the proposed algorithm subject to source ramp-up constraints. We shall use the result in §4.4.1 as a stepping stone and take a similar approach. Consider the nonlinear system shown in Fig. 4.5. The system is similar to the one in Fig. 4.4 except the block showing that $\dot{r}(t) = \min[\dot{e}(t), g]$, i.e., the ramp-up of sources is constrained.

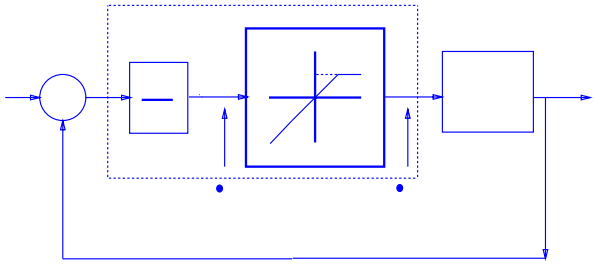

Figure 4.5: The non-linear model with constraints.

Lemma 4.4.2 *For the system in Fig. 4.5, $\frac{k}{n}\sum_{i=1}^{n}\tau_i < 1$ is a sufficient condition to guarantee the asymptotic stability of the system.*

Proof: See Appendix 4.10.2.

We have shown in §4.4.1 that $\frac{k}{n}\sum_{i=1}^{n}\tau_i < 1$ is a sufficient condition for ensuring the asymptotic stability in the linear feedback case. The same condition also holds for our proposed algorithm where the source ramp-up is constrained. Note that the ramp-up constraint g will affect the minimum buffer requirement in §4.3. Hence we can change g to meet different buffer requirements without affecting the stability. Another useful property is that this condition only depends on the summation of all ABR session's round-trip delays (or equivalently the *average* round-trip delays among all sessions), rather than their individual values. This means that variation in each session's round-trip delay can be tolerated as long as the *average* satisfies the stability condition.

4.5 Steady State and Fairness

In the previous section we showed the asymptotic stability of our proposed mechanism under the greedy source assumption. Next we consider its steady state characteristics. We shall focus on the dynamics of bottleneck link in response to our control mechanism.

4.5.1 Steady State Characteristics of Greedy Sources

In steady state the derivatives of system variables go to zero, i.e.,

$$\dot{e}(\infty) = \lim_{t\to\infty} \dot{e}(t) = 0 \text{ and } \dot{q}(\infty) = \lim_{t\to\infty} \dot{q}(t) = 0.$$

If $\dot{e}(\infty) = 0$, then from (4.9) it follows that $ne(\infty) = c$. Hence $f(q(\infty)) = 0$ and $q(\infty) = q^*$. As a result, the system converges to $e(\infty) = \frac{c}{n}$ and $q(\infty) = q^*$. It should be clear that available capacity of a bottleneck link is partitioned fairly among sources which are "on" and greedy.

4.5.2 Impact of Constrained Flows

Next we consider how the bottleneck link responds to the status in the other parts of the network. Suppose some sessions are not able to send traffic at the allocated explicit rate, i.e., are constrained elsewhere in the network. These constrained sessions will cause the aggregate arrival rate to be smaller than expected, thus the queue length will decrease. Nevertheless, the drift function $f(\cdot)$ in (4.3) aims to bring queue length towards the target level, so it will compensate for this queue changes by increasing $e(t)$. The increased $e(t)$ allows other sessions to send traffic at even higher rates, so the available capacity will not wasted. In other words, the unused bandwidth of constrained sessions is re-allocated to the greedy sessions.

Suppose m sessions are constrained by peak rates $p_i > r^*$ elsewhere in the network. In steady state the following equation would hold:

$$\frac{c + f(q(\infty))}{n} * (n-m) = c - \sum_{i=1}^{m} p_i.$$

It follows that $f(q(\infty)) = \frac{m}{n-m}c - \frac{n}{n-m}\sum_{i=1}^{m} p_i$ and $q(\infty) = [0, q^* - \frac{1}{k(n-m)}(mc - n\sum_{i=1}^{m} p_i)]^+$, so the steady state queue length lies between q^* and 0 as a result of constrained traffic. Thus in order to fully re-allocate unused bandwidth, we need q^* to be greater than $\frac{1}{k(n-m)}(mc - n\sum_{i=1}^{m} p_i)$. Also notice that although constrained sessions are allocated higher rates than they really use, this will not cause network instability. Indeed all sessions have to ramp up at a rate g, so if their constraints are suddenly removed, this will give the network time to detect the change.

4.5.3 Sessions with Minimum Cell Rate Guarantees

A further goal in managing ABR sessions is to guarantee each session a pre-negotiated Minimum Cell Rate (MCR) m_j, as well as a fair share of the spare network capacity among

the currently active sources. In order to achieve this, we modify the proposed mechanism as follows.

We reserve the MCR, m_j, for each session and make the worst case assumption that a session will attempt to send traffic at a rate no smaller than its MCR. Moreover, we modify the definition of $c(t)$ and $r_j(t)$ such that $c(t) + \sum_{i=1}^{n} m_j$ and $m_j + r_j(t)$ are the total available capacity and the real transmission rate of session j respectively. In other words, $c(t)$ and $r(t)$ in the previous analysis are the spare capacity and data rate in excess of the reserved MCRs. Therefore, $e(t)$ is the fair share of spare capacity for each session, and the transmission rate of session j is bounded by $m_j + e(t)$. Note that the bottleneck node will still send the same explicit rate $e(t)$ to each session, so the complexity of the algorithm does not change.

The reserved MCRs are constant terms, so they have no impact on the stability results. In this case, $\frac{k}{n} \sum_{i=1}^{n} \tau_i < 1$ is still a sufficient condition for asymptotic stability. The minimum buffer requirement b_{min} stays the same as in (4.4), but now we need to reserve additional capacity to satisfy minimum cell rate requirements of ABR sessions sharing the link i.e., $c_{min} + \sum_{i=1}^{n} m_j$.

4.6 ABR Call Admission and Statistical Multiplexing

Feedback control will be ineffective to control the connections sending small bursts whose durations are shorter than the control time scale. While resources are reserved to account for such bursts, the resource utilization is usually low. In this section we consider the role of bursty ABR sessions and the statistical multiplexing of independent bursts from such sessions. Our goal is to show that the admissible number of concurrent ABR sessions can be increased by relaxing the loss constraint in a controlled manner.

Lossless analysis. From the result in (4.8), one can compute the buffer requirement when at most n_{max} sessions are carried with a lossless guarantee. Moreover, (4.8) can also be used as an admission control threshold to compute the admissible number n'_{max} of ABR sessions subject to a fixed buffer size and lossless guarantee. Note that n'_{max} would then be a function of r^*, g, ρ, and τ. The capacity c does not affect n'_{max}, as long as the rate c_{min} is guaranteed, but the available capacity's fluctuation ρ plays an important role in determining n'_{max}.

The analysis in §4.3 is based on the worst case assumption that n'_{max} ABR sessions are concurrently "bursting" without being detected by the node. This is conservative since it is unlikely that all ABR sessions will "burst" at the same time. Furthermore, ABR sessions can not keep "bursting" without being detected. The node will usually detect such sources and reallocate the explicit rate in a round-trip delay time, τ. Hence we can exploit the "statistical multiplexing" of ABR sessions by modeling their bursting behavior and relaxing the loss requirement, so as to increase the admissible number of sessions from n'_{max} to n^b_{max}. In short, we shall control the probability that n'_{max} sessions out of n^b_{max} will burst within τ seconds.

Controlled loss. Suppose ABR sessions alternate between "off" and "on" modes throughout their lifetime and the distribution of the time that sources stay "off" is modeled by an exponential distribution with a parameter λ. We further assume the distributions of each session's "off" time are identical and independent. We can construct a random variable I_j as follows

$$I_j = \begin{cases} 1 & \text{if stream } j \text{ jumps from "off" to "on" within } \tau, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, a *conservative* admissible number of ABR sessions subject to controlled loss would be

$$n^b_{max} = \max\{n \mid \mathbb{P}(\sum_{j=1}^{n} I_j > n_{max}) < \delta\},$$

where δ is a design parameter which roughly characterizes the desired quality of service at the buffer of the bottleneck link. From the exponential distribution it follows that $\mathbb{P}(I_j = 0) = e^{-\lambda\tau}$, and n^b_{max} can be determined by a Binomial distribution. The additional number of sources $n^b_{max} - n_{max}$ that can be admitted on the link is due to statistical multiplexing of ABR bursts. The effectiveness of such multiplexing depends on the average interval between "bursts", $\frac{1}{\lambda}$, and round-trip delay τ. Some examples are considered in §4.8.

Capacity variability ρ. The analysis in §4.3 showed that the variability ρ in the available capacity is critical to assessing queue fluctuations and thus the losses in the network. Since ABR sessions exploit the unused bandwidth of VBR sessions sharing the same link, the variation of the aggregate VBR bandwidth requirement will affect the available capacity of ABR sessions. For example, an increase in the VBR bandwidth requirement will mean a decrease in the available capacity of ABR sessions. Fig. 4.6 shows what might be the aggregate bandwidth increment for a fixed number of heterogeneous VBR (MPEG 1 video) sessions over various time scales τ. Measurements of the worst case and average ρ show that the variability is on the order of 10's of Mbps/s and highly dependent on the time scale of interest. On the good side, the variability grows in a sub-linear fashion in the number of sources, suggesting that VBR multiplexing will help to reduce the variability. On the bad side, there is a large discrepancy between the average and worst case variability, which makes buffer dimensioning difficult. In practice one might consider the distribution of this quantity, represented by a random variable R, and let ρ satisfy $\mathbb{P}(R > \rho\tau) \leq 10^{-6}$, so that the probability of failure for the control is small, and the quality of service is roughly maintained. Given such a ρ, an appropriate buffer size can be determined.

4.7 Implementation and Design Issues

We briefly discuss the implementation of the proposed control mechanism based on the rate-based flow control framework in [4].

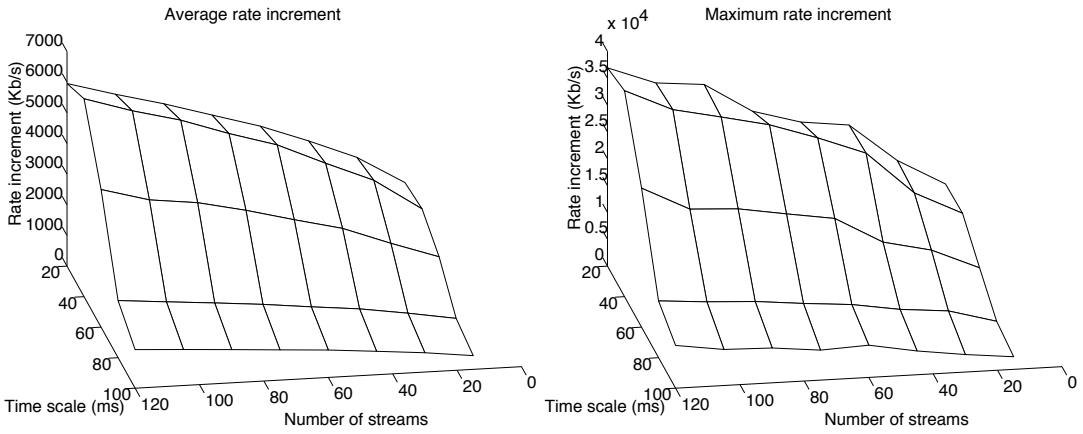

Figure 4.6: Average and maximum bandwidth increments for aggregated VBR sources.

4.7.1 Protocols and Complexity

Feedback rate control depends on regular information exchange between the network and sources. This information is carried in special Resource Management (RM) cells which are periodically (every *Nrm* data cells) generated by the source and sent along the session's route to the destination where they are looped back to the source. The RM cells carry various types of information; of particular interest herein, will be the Current Cell Rate (CCR), the Minimum Cell Rate (MCR), the Explicit Rate (ER), and a Congestion Indication (CI) bit. The explicit rate is initially set to the source's Peak Cell Rate (PCR) and the CCR is equal to the Allowed Cell Rate (ACR) when the RM cell is generated. The ER and CI fields can be modified by properly equipped switches, as the RM cell travels through the network.

In our proposed mechanism the CCR corresponds to the current transmission rate of the source which is constrained by the latest ER message sent to the source from the network. We envisage a setting where explicit rates are computed at some or all of the switches a session traverses, and the minimum of computed ERs is stamped on the returning RM cells. To compute the current ER, the switch needs to determine roughly how many sources are "on." Notice that this assessment could be done at the source/policer end and the results are encoded in RM cells. The switch does not need to monitor the rate/state information for each session, which otherwise could be prohibitively expensive for switches carrying a large number of ABR sessions.

Hence the switch simply tracks the number of "on" sessions by updating a state variable based on the information carried by RM cells, and monitors both the available capacity and queue state. Each source would receive a returned RM cell with an explicit rate, which it would for example add to its reserved MCR to determine its allowed cell rate. This algorithm has the advantage that the computed $e(t)$ is the same for all sources sharing a given link. This significantly reduces the complexity of computing explicit rates and stamping RM cells. Pseudo code of the proposed algorithm can be found in Appendix 4.10.3.

4.7.2 Estimation of Link Status

The proposed algorithm uses the available capacity $c(t)$, the queue length $q(t)$, and the number of "on" ABR sessions $\hat{n}(t)$ to compute $e(t)$. The queue length can be easily obtained by monitoring the ABR buffer of a link, but the available capacity is dependent on other service types sharing the same link, such as VBR. Suppose a given link (e.g., an output port of a switch) is shared by VBR and ABR connections. The available capacity consists of two parts. The first part is the difference between the link capacity and the bandwidth which has been reserved to provide quality of service guarantees to VBR connections. The second part comes from the momentarily unused capacity left from the reserved capacity for VBR. To determine the amount of momentarily unused capacity, we need to measure how much capacity is consumed by VBR.

To estimate $\hat{n}(t)$, one can monitor the rate/state of each ABR connection. However, the complexity of such approach is again a concern in its implementation. We propose an algorithm for estimating $\hat{n}(t)$ without doing per-source accounting. Suppose the i^{th} RM cell arrives at a switch at time t_i, and it carries CCR_i and a status[3] $s_i \in \{0,1\}$, i.e., "off" or "on." The switch monitors the RM cell arrivals in a synchronous fashion over fixed length intervals of l seconds. For the j^{th} interval, the number of "on" sources which send RM cells can be approximated by

$$\delta n_j = \sum_{i \in L_j} \frac{Nrm}{l * CCR_i} s_i, \quad \text{where } L_j = \{i \mid jl < t_i \leq (j+1)l\}.$$

Note that the summation of s_i has been normalized by the inter-arrival time of RM cells and the interval length l. Suppose an "on" source's CCR is fixed, it will send RM cells every $\frac{Nrm}{CCR}$ seconds. Within l seconds, $\frac{l*CCR}{Nrm}$ RM cells are expected to arrive at the bottleneck link, so the associated s_i is normalized by this number to get a correct estimate for the number of active sources. As a result, $\hat{n}(t)$ is a piece-wise step function, which is continuous from right hand side at the points $t = j * l, j \in N$. The recursive estimate is computed as follows

$$\hat{n}((j+1)l) = \hat{n}(jl) * \alpha + \delta n_j * (1-\alpha),$$

where α is an averaging factor. Fig. 4.7 shows an example of estimated $\hat{n}(t)$ and real $n(t)$ when a link carries 100 bursty ABR sessions. Clearly $\hat{n}(t)$ can track the number of "on" sessions quite nicely.

4.7.3 Design Parameters

From the analysis in §4.3, the buffer requirement is a function of several system parameters. In the following we discuss the design trade-offs in selecting these parameters.

Drift function $f(\cdot)$. In §4.3 we considered the linear drift function $f(q(t)) = -k(q(t) - q^*)$ and used the fact that $f(q(t)) \leq kq^*$ to derive an upper-bound on the queue length. In practice one may want to saturate the maximum value of the drift $f(\cdot)$ in order to control

[3]Currently, the on/off status bit is not in the standard definition of RM cell fields.

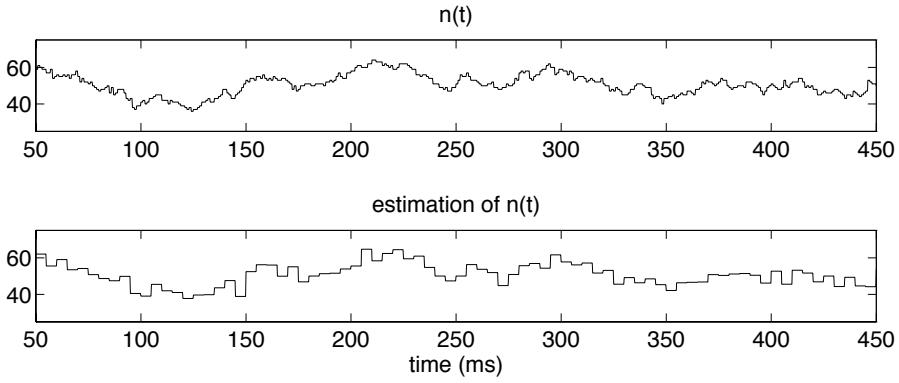

Figure 4.7: The estimation of $\hat{n}(t)$.

b_{min} to a desirable value, see e.g., Fig. 4.8. Moreover, if the minimum value of $f(\cdot)$ is clamped by f_{min} and $-k(b_{min}-q^*) < f_{min} < 0$, the minimum capacity requirement c_{min} will be equal to $-f_{min}$ rather than $k(b_{min}-q^*)$, thus the minimum capacity requirement can be reduced.

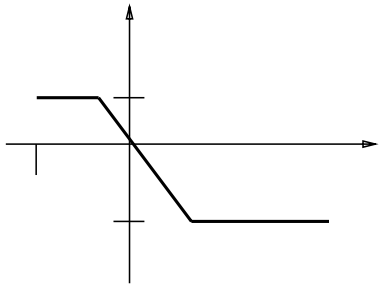

Figure 4.8: A saturated function $f(\cdot)$.

Ramp-up constraint g. In practice source rate adjustments might occur according to the positive feedback CI mechanism using additive rate increase and proportional rate decrease factors (AIR,RDF) discussed in [4]. Note that using this type of proportional rate control mechanism, one can bound a source's growth g by $PCR/Nrm \times AIR$. Thus these parameters can be used to optimize the operation and determine the growth rate that should be assumed in dimensioning of the network resources. Linear ramp-up also facilitates the integration of CI-based and ER-based mechanisms in a heterogeneous environment.

Rate threshold r^*. This parameter allows the sessions some degree of freedom in sending small bursts. It also captures the impact of such bursts on the network operation. The threshold r^* prevents the switch from incorrectly believing a session sending small bursts is becoming a major contender for spare capacity, thus reducing the overhead of doing unnecessary adjustments on the explicit rate computation. Moreover, r^* could be interpreted as the Initial Cell Rate (ICR) in the ABR framework [4], which limits the initial transmission rate after an idle period.

Capacity variation ρ. In addition to the fluctuations in available capacity due to statistical multiplexing of VBR flows, a further contributor to the changes in available capacity would be the admission of new CBR/VBR calls into the system. Let $N(r,s]$ denote the number of *new* connections that are admitted to a particular link during the time interval $(r,s]$. In order to control the magnitude of ρ, it may be necessary to constrain call admissions such that $N(r,s] \leq \gamma(s-r)$. Note that hardware and demand would limit the rate γ of call admissions. During a round trip delay τ, at most $\gamma\tau$ connections are initiated. Assuming they have a peak rate p Mbps, the available bandwidth could decrease at a rate $p\gamma$ Mbps/sec. Overall we believe it is not unreasonable to assume that once the operation regime and traffic on a link is known, the variability ρ can be assessed by combining empirical evaluation of VBR traffic fluctuations and admission control on the connection process.

Queue threshold q^*. The target queue level q^* will determine the overall utilization of the system. Intuitively the larger q^*, the greater the ability of the system to buffer ABR traffic, and thus to exploit available capacity if it suddenly becomes available. However, a larger q^* means a larger queuing delay in steady state, so a trade-off between utilization and delay needs to be made in selecting q^*. In addition, if some ABR sessions are constrained and can not fully utilize the allocated rate, we showed in §4.5.2 that a large enough q^* is necessary to allow reallocating the unused capacity of constrained connections. In essence, q^* determines the "dynamic range" for the explicit rate that the link can support when sessions are constrained elsewhere.

4.8 Simulation and Performance Evaluation

In this section we present some simulation results to verify the analysis in previous sections. Our network configuration, shown in Fig. 4.9, contains 15 ABR connections and aggregated VBR connections sharing a bottleneck link. We are interested in the interaction between ABR feedback control and the rate variation of VBR connections, as well as their impact on the bottleneck node's queue length.

Stability. In §4.4 we proved the bottleneck queue length will converge to the target level q^* in a fixed environment if the drift function gain k is chosen such that $\frac{k}{n}\sum_{i=1}^{n}\tau_i < 1$. We first consider the case where ABR connections are greedy and the VBR connections are off, hence the available capacity and the number of "on" sources are both fixed. The queue threshold q^* is 200 and the largest round-trip delay τ for the ABR connections is set to be 20 ms, which means that the largest k guaranteed to ensure stability is 0.0207 Mb/(Cells*s). The queue dynamics for $k = 0.02$ and $k = 0.06$ are shown in Fig. 4.10, illustrating that $k = 0.06$ may result in instability.

Queue response to varying available capacity. Next we study the bottleneck queue response when the available capacity changes. We feed greedy ABR connections and an on/off VBR connection into the bottleneck link. The arrival rate of the VBR connection, ACR of an ABR connection, and the queue length are shown in Fig. 4.11. The figure shows

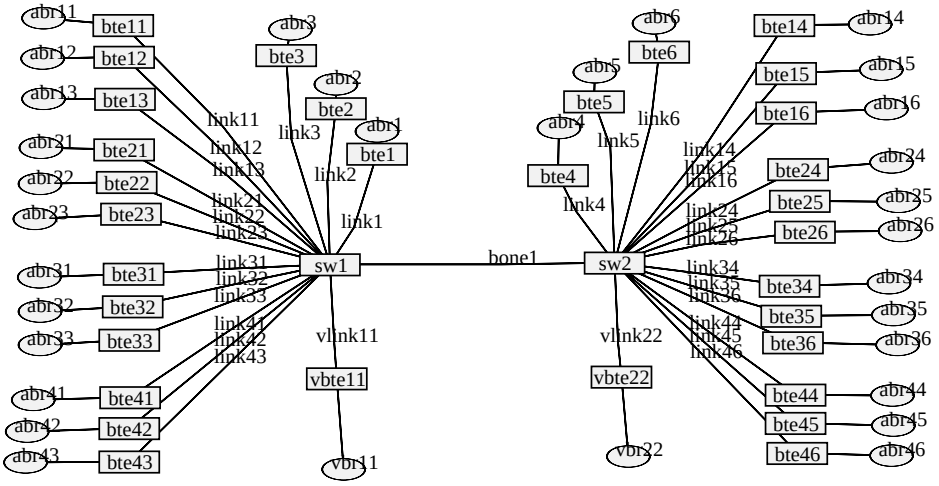

Figure 4.9: A bottleneck link shared by ABR and VBR connections.

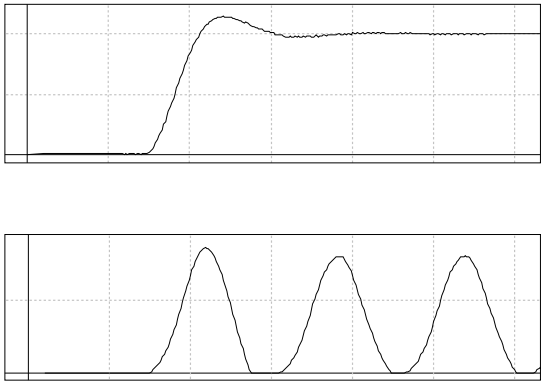

Figure 4.10: Queue dynamics for different drift function gains k.

a jump in the queue length when the VBR flow starts bursting. However, the queue length goes back to the target level and available capacity is reallocated after the bottleneck link responds to the change in the available capacity. Since the queue length is controlled around the target level, the available capacity is fully utilized.

Statistical multiplexing. In the lossless case the queue overshooting contributed from bursting ABR sources is determined by the total number of ABR sessions, n_{max}. In §4.6 this contribution was reconsidered because the effective number of bursting ABR sessions within a round-trip delay time is smaller than the total number of ABR connections due to statistical multiplexing. Hence, in a controlled loss scenario, the buffer requirement can be reduced, or alternatively the admissible number can be increased. We assume the average idle time of ABR connections is 100 ms and use the result in §4.6 to compute the n^b_{max} for different "QoS" when $n_{max} = 30$. The results, shown in Table 4.1, indicate that the admissible number of ABR sessions is increased significantly due to statistical multiplexing

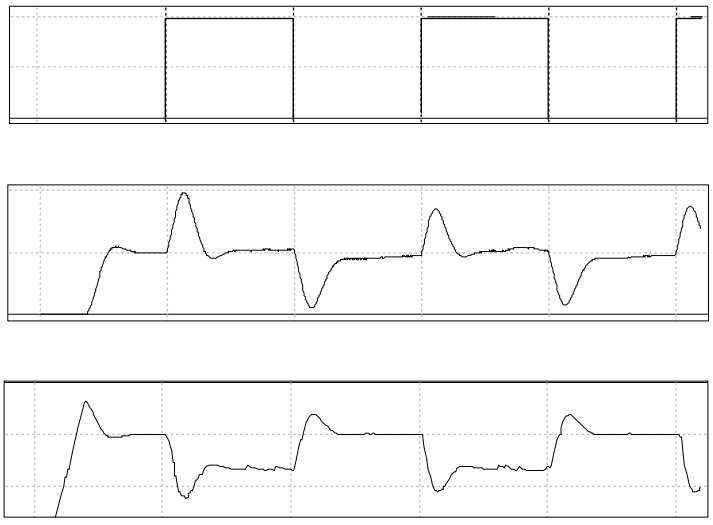

Figure 4.11: Queue length and ACR in a changing environment.

"QoS" (δ)	0	1e-7	1.0e-6	1.0e-5	1.0e-4	1.0e-3
n_{max}	30	30	30	30	30	30
n^b_{max}	30	63	68	75	84	94

Table 4.1: Comparison of admissible numbers.

of their bursts.

Utilization improvement. An important advantage of introducing ABR service is to improve the network utilization. Because of the stringent quality of service requirement and bursty behavior of VBR traffic, the network utilization is usually low if only VBR connections are carried. One can let ABR connections use the momentarily unused bandwidth inside the network, so as to improve utilization. In our final simulation we let the bottleneck link carry 45 VBR connections from the video traces for obtaining Fig. 4.6. In addition, we introduce 3 greedy ABR connections to exploit the unused bandwidth. We found the utilization of the bottleneck link increased from 70% to 95%. The plots of aggregate VBR arrival rate, ACR of an ABR connection, and the queue length are shown in Fig. 4.12. It shows that the ACR of ABR connections are varying according to the changes in the VBR arrival rates.

4.9 Summary

In order to avoid throughput collapse, ABR service will need to be implemented so as to provide some control on cell loss. To achieve this, flow control mechanisms need to be designed, so that by making appropriate resource reservations and performing call admission,

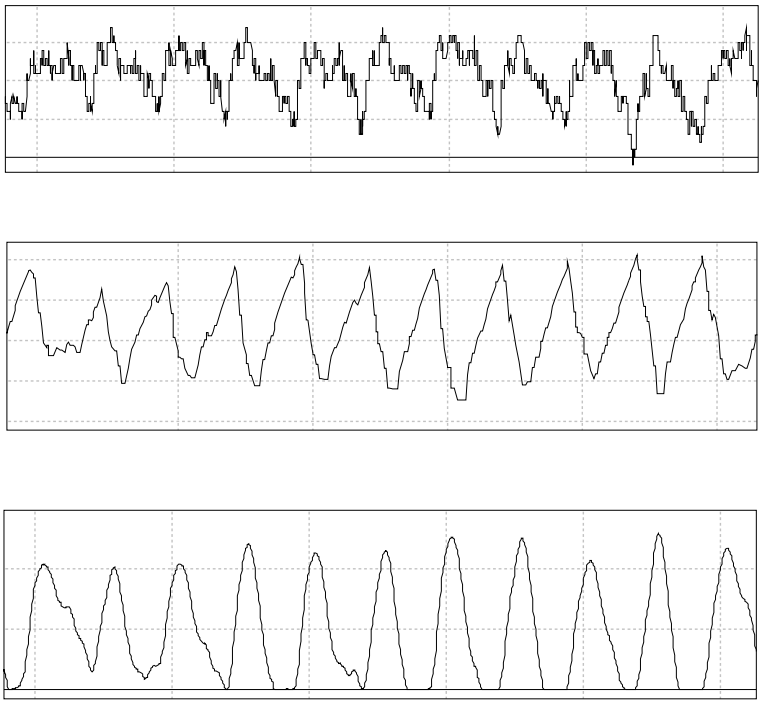

Figure 4.12: 45 VBR and 3 ABR connections.

the network can ensure that losses are low. We have proposed a *simple* algorithm to compute an explicit rate bound on source traffic. Indeed, it substantially reduces the complexity of computing explicit rates, since it is based on estimating the number of ABR sessions currently contending for bandwidth on a bottleneck link, without requiring per-connection rate/state information. By accounting for the rate variability of the interfering (VBR) traffic and the source update behavior, we analyzed the queue dynamics at the bottleneck link and derived the minimum buffer and capacity requirements for guaranteeing lossless service to ABR connections.

We have introduced a threshold r^* to discriminate among sources with different activity levels. Sources are free to send bursts at any rate below the threshold, but must ramp up linearly after exceeding it. In practice this setup would expedite the transmission of short bursts and facilitate the integration of CI-based and ER-based flow control mechanisms. In general, feedback control would typically be ineffective at regulating ABR traffic with small burst sizes relative to the network's round trip delay time. Hence resources need to be reserved to absorb such traffic variability and control loss, but doing so would typically reduce link utilization. By accounting for statistical multiplexing of source bursts, one can reduce the required reservations to achieve the desired quality of service, or alternatively one can allow for a larger number of concurrent ABR connections for a given reservation. We have articulated this point of view and proposed a primitive model to assess the acceptable number of concurrent connections. The effectiveness of such multiplexing depends on the control time scale and the characteristics of bursty traffic.

Finally, we have discussed design parameters for our proposed algorithm in the context of standard ABR rate control mechanisms. We identified the factors that would affect queue overflows, such as the source behavior, variability of available capacity, and of course round trip delays. We believe that our analysis provides some novel insights to dimensioning capacity/buffer requirements for ABR rate control mechanisms.

Acknowledgment. We used the NIST ATM network simulator for the simulation in this chapter. The pseudo codes of switches and ABR sources were implemented based on [21].

4.10 Appendix

4.10.1 Proof of Lemma 4.4.1

To ensure $e(t) \geq 0$, we require that $f(q_{max}) + c \geq 0$, i.e., the maximum queue length will determine the minimum capacity requirement. First notice that

$$\dot{q}(t) = \sum_{i=1}^{n} e(t-\tau_i) - c = \frac{1}{n}\sum_{i=1}^{n}[f(q(t-\tau_i)) + c] - c = \frac{1}{n}\sum_{i=1}^{n} f(q(t-\tau_i)),$$

hence $\dot{q}(t)$ depends on the past queue lengths. If $q(t-\tau) \geq q^*,\ \forall t > 0$, it follows that $f(q(t-\tau_i)) \leq 0,\ \forall i,\ \forall t > 0$ and $\dot{q}(t) \leq 0,\ \forall t > 0$. In other words, the queue length has to stop increasing τ seconds after it exceeds q^*. Consider arbitrary time intervals during which the queue length exceeds q^* and call such periods "overshooting cycles." We show an upper bound on the queue length over an "overshooting cycle."

Without loss of generality, let $t = 0$ be the beginning of an "overshooting cycle" and $q(0) = q^*$. Since the queue length exceeds q^*, we know that $\dot{q}(t) \leq 0,\ \forall t > \tau$ on an "overshooting cycle." To compute the maximum queue length q_{max} of an "overshooting cycle," it suffices to consider the worst case queue growth on the interval $[0, \tau]$:

$$q(t) \leq q^* + \int_0^{\tau} \sum_{i=1}^{n} \frac{f(q(t-\tau_i))}{n}\, dt \leq q^* + \int_0^{\tau} kq^*\, dt = q^* + kq^*\tau = q_{max}.$$

Thus q_{max} is an upper bound on the queue length over an "overshooting cycle." For intervals other than "overshooting cycles," the queue length does not exceed q^*, thus q_{max} is an upper bound on the queue length. Given this bound, in order to ensure non-negative $e(t)$, it suffices that when $c \geq -f(q_{max}) = k^2 q^* \tau$.

■

4.10.2 Proof of Asymptotic Stability

In Fig. 4.5 we have a controller $k(x)$ with input $x = c - \sum_{i=1}^{n} r(t-\tau_i)$ in the system. Since the ramp-up of sources is constrained, i.e., $\dot{r}(t) = \min[\dot{e}(t), g]$, the controller $k(x)$ is nonlinear and $k(x) = \min[\frac{k}{n}x, g]$, see Fig.4.13. A generalized Nyquist criterion—Circle criterion [50][p.344] is useful in determining the stability of a nonlinear system. For reference, an abridged and rephrased version of the theorem is given below.

Theorem 4.10.1 (Circle criterion) *Consider a feedback control system consisting of a nonlinear controller (memoryless gain function) $k(x)$ and a LTI system $G(s)$, e.g, the system in Fig. 4.5. The system is asymptotically stable if*

1. *$k(x)$ lies in the sector $[a,b]$, i.e., $a \le \frac{k(x)}{x} \le b$, $\forall x \neq 0$, where $0 < a < b$,*

2. *$G(s) = G_a(s) + G_r(s)$, where $G_r(s)$ is strictly proper and $G_a(s)$ is the Laplace transform of a function in the space $L_1[0,\infty)$ augmented by delayed impulses,*

and the Nyquist theorem is satisfied for the $G(j\omega)$ locus with respect to the circle with diameter on the negative real axis of the complex plane from $-\frac{1}{a}$ to $-\frac{1}{b}$, i.e., the locus stays away from the circle and encircles it an appropriate number of times [50] according to the Nyquist theorem. ■

We first show that $k(x)$ lies in a sector. Since $r(t)$ is non-negative, hence $x = c - \sum_{i=1}^{n} r(t-\tau_i) \le c$. In addition, we assume that $k(c) = \min[\frac{k}{n}c, g] = g < \frac{k}{n}c$, otherwise $k(x)$ is a linear function and the stability of such system can be considered based on the Nyquist criterion, see §4.4.1. We consider the possible values of $\frac{k(x)}{x}$ as follows.

- Case 1: $0 < x \le c$

$$\frac{k(x)}{x} = \begin{cases} \frac{k}{n}, & \text{if } 0 < x < g\frac{n}{k}, \\ \frac{g}{x} \ge \frac{g}{c}, & \text{if } g\frac{n}{k} \le x \le c. \end{cases}$$

- Case 2: $x < 0$

$$\frac{k(x)}{x} = \frac{\min[\frac{k}{n}x, g]}{x} = \frac{k}{n}.$$

Note that $\frac{g}{c} < \frac{k}{n}$ because $k(c) = g < \frac{kc}{n}$. Thus we can conclude that $\frac{g}{c} \le \frac{k(x)}{x} \le \frac{k}{n}$.

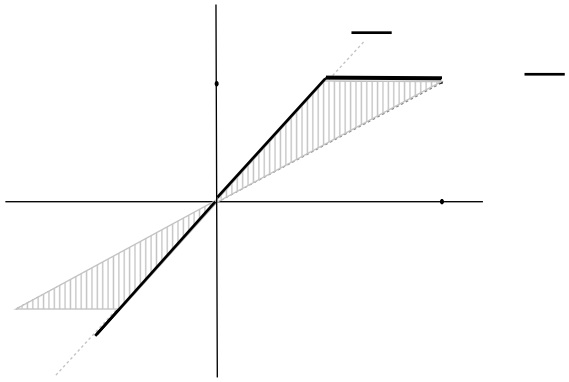

Figure 4.13: $k(x)$ lies in a sector.

Next we verify that whether $G(s)$ can be decomposed in the form of $G_a(s) + G_r(s)$. Let us choose a single term $\frac{1}{s}e^{-\tau_1 s}$ in $G(s)$ as an example. We can decompose $\frac{1}{s}e^{-\tau_1 s}$ in the following way,

$$\frac{e^{-\tau_1 s}}{s} = \frac{e^{-\tau_1 s} - 1}{s} + \frac{1}{s}.$$

The second term is strictly proper, which satisfies the condition in Theorem 4.10.1. The inverse Laplace transform of the first term is $u[t-\tau_1] - u[t]$, where $u[t]$ is a unit step function, thus it is clearly in $L_1[0,\infty)$.

Finally, let us consider the Nyquist plot of $G(j\omega)$. As shown in §4.4.1, the real part of $G(j\omega)$ always resides on the right hand side of the vertical line $z = -\sum_{i=1}^{n} \tau_i$ on the complex plane. If $\frac{k}{n}\sum_{i=1}^{n} \tau_i < 1$, the Nyquist plot of $G(j\omega)$ will stay away from the circle determined by $-\frac{n}{k}$ and $-\frac{c}{g}$, see Fig. 4.14. Therefore, $\frac{k}{n}\sum_{i=1}^{n} \tau_i < 1$ is a sufficient condition to guarantee the stability of the system in Fig .4.5.

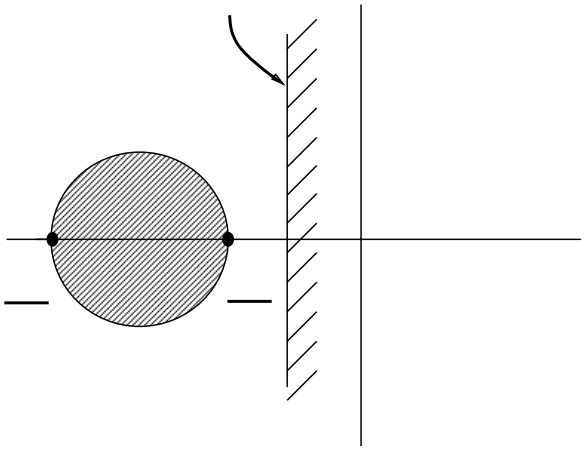

Figure 4.14: Nyquist plot of $G(j\omega)$ and the circle.

4.10.3 Pseudo-code for Sources and Bottleneck Links

```
/* ABR SOURCE */

/* initialization */
 count=0;
 acr = MCR;

/* upon receiving a returned RM */

  er  = cell(ER);                     /* get the new feedback from Network */

/* send cells */

 if(now() >= scheduled_cell_time)
  {
      if (count==0)                              /* time for sending RM cells */
      {                                          /* and updating acr          */
          acr = acr + g*(now() - last_RM);
          acr = min(acr, er + MCR);
          acr = min(acr, PCR);
          send_RM_cell();
          last_RM = now();
          count++;
      }
      else
      {
          send_data_cell();
          count = (count + 1) mod Nrm;
      }

      scheduled_cell_time = now() + 1/acr;
  }
```

```
/* SWITCH */

/* upon receiving a forward RM cell */

 n_on  = update( RM_cell(status) );     /* update the number of ON sources */
 drift = -k*(q_length - q_target);
 er    = (link_rate + drift - measured_VBR_rate) /(min(1,n_on));
                                                    /* compute the new ER */
/* upon receiving a backward RM cell */

 RM_cell(ER) = min(er,RM_cell(ER));         /* stamp ER on the backward RM */
```

Chapter 5

Conclusions

In this dissertation we have discussed three important issues in the management of integrated services networks. Below we summarize our findings.

- **Admission control.** We analyzed statistical multiplexing of deterministically constrained traffic. The rationale for using deterministic traffic descriptors is that such parameters can be enforced and verified by networks. An upper bound on the overflow probability in a buffered link is derived based on the traffic descriptor, link capacity, and buffer size. Using this upper bound, one can design a conservative call admission scheme which guarantees the QoS of established connections.

- **Routing.** We proposed a routing scheme which accounts for traffic mix and the efficiency of multiplexing in a multiservice network. Given the nature of statistical multiplexing, the resource requirements of a connection are state-dependent across network resources. Thus in principle we can exploit this dependence to achieve better overall network efficiency. For example, we found that it is not always advantageous for the VPs connecting a given source-destination pair to carry *all* traffic types. To maximize the throughput, only a small number of traffic types, or homogeneous traffic should be present on each VP. This suggests that in practice multiservice networks might end up looking like multiple logical networks which are segregated by the service types.

- **Flow control.** We proposed an explicit rate flow control algorithm for ABR which draws on measuring the current queue length, bandwidth availability, as well as tracking the current number of active sessions contending for capacity. Because the number of active connections are estimated by a simple scheme without using the per-connection information, the complexity of the proposed algorithm is minimized. We also considered the role that statistical multiplexing might have in managing bursty ABR sessions.

Design issues. ATM networks are geared towards supporting and integrating a variety of communication services which might broadly be divided into those based on *reservation*, e.g., Constant and Variable Bit Rate (CBR,VBR) services and *best effort* services, such as

Unspecified Bit Rate (UBR) and to some extent ABR services. To guarantee QoS as well as to maximize efficiency, a general approach might be as follows. The aforementioned call admission scheme can be used to determine a connection's resource requirements and to control the number of established connections in order to ensure the QoS of reservation-based services. Since resource reservation is based on conservative deterministic traffic descriptors, there will be a significant but varying amount of bandwidth for adaptive services such as ABR. In principle a flow control scheme can direct best-effort traffic to exploit the available capacity, leading to an overall improvement in system utilization.

Whether this combination of conservative CAC with ABR services can achieve an adequate utilization depends on the speed (bandwidth) and extent (i.e., propagation delays) in the network. Indeed, consider a Wide Area Network having high capacity links but large round trip propagation times. For such networks flow control may be sluggish and ineffective. However, due to the large capacity links our conservative CAC scheme based on crude traffic descriptors can achieve relatively high efficiencies. By contrast in a Local Area Network with low round trip delays and perhaps lower link rates, the multiplexing of reservation-based services would be less efficient. However, in this case flow control mechanism can be very effective at exploiting spare capacity in the system. Thus we argue that by considering the combined utilization that can be achieved in multiservice networks, relatively crude CAC can be used without compromising efficiency.

Future work. There are some issues discussed in this dissertation that deserve extension and further investigation. It is natural to ask how the proposed call admission scheme can be written as a simple formula in terms of the UPC parameters such as the peak cell rate, sustainable cell rate, and burst tolerance etc. A simple formula is desirable because it would greatly streamline the procedure of call admission. In addition, such a formula is also helpful in resource reservation.

We obtained some qualitative understanding on the routing issues in Chapter 3. However, it is desirable to "quantify" these findings. For example, it is useful to be able to analytically estimate the blocking probability based on traffic loads and link capacities across the networks. Based on such an estimation, one can allocate appropriate VP capacities in order to maintain desirable system performance.

A further open problem is to find the worst case traffic pattern subject to a deterministic traffic descriptor. In Chapter 2 we considered an optimization problem and obtained some properties of the possible solutions. It would be interesting to solve this problem since this would provide a tight worst case bound on multiplexing performance.

Bibliography

[1] E. Altman, F. Baccelli, and J. C. Bolot. Discrete-time analysis of adaptive rate control mechanisms. *Proc. 5th Int. Conference on Data and Communications*, pages 121–40, 1993.

[2] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching network: The case of a single congested node. *IEEE Trans. Networking*, Vol. 1:694–708, 1993.

[3] P. Billingsley. *Probability and Measure*. John Willey and Sons, New York, 1986.

[4] F. Bonomi and K. W. Fendick. The rate-based flow control framework for the available bit rate ATM service. *IEEE Network Mag.*, Vol. 9, No. 2:25–39, 1995.

[5] F. Bonomi, D. Mitra, and J. B. Serry. Adaptive algorithms for feedback-based flow control in high-speed wide-are ATM networks. *IEEE JSAC*, Vol. 13, No. 7:1267–83, 1995.

[6] S. Borst and D. Mitra. Asymptotically achievable performance in ATM networks. *To appear in Advanced Applied Probability*.

[7] D.D. Botvich and N.G. Duffield. Large deviations, the shape of the loss curve, and economies of scale in large multiplexers. *Technical Report DIAS-APG-94-12, Dublin Institute for Advanced Studies*, 1994.

[8] J-Y. Le Boudec, G. de Veciana, and J. Walrand. QoS in ATM: theory and practice. *35th IEEE CDC*, pages 773–778, 1996.

[9] J.A. Bucklew. *Large Deviation Techniques in Decision, Simulation and Estimation*. John Wiley and Sons, New York, NY, 1990.

[10] A. Charny, K. K. Ramakrishnan, and A. Lauck. Time scale analysis and scalability issue for explicit rate allocation in ATM networks. *IEEE Trans. Networking*, Vol. 4:569–581, 1996.

[11] S. Chong, R. Nagarajan, and Y.T. Wang. First-order rate-based flow control with dynamic queue threshold for high-speed wide-area atm networks. *Proceedings of SPIE Conference on Performance and Control of Network Systems*, 3231:259–270, 1997.

[12] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[13] R. L. Cruz. A calculus for network delay, Part 1: Network elements in isolation. *IEEE Trans. Inform. Theory*, 37:114–131, 1991.

[14] M. de Prycker. *Asynchronous Transfer Mode Solution for Broadband ISDN*. Prentice-Hall, 1996.

[15] B. T. Doshi. Deterministic rule based traffic descriptors for broadband ISDN: Worst case behavior and connection acceptance control. *Proc. 14th Int. Teletraffic Cong., 6-10 June 1994 North-Holland Elsevier Science B.V.*, 1:591–600, 1994.

[16] A. Elwalid, D. Mitra, and R.H. Wentworth. A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node. *IEEE JSAC*, Vol. 13, No. 6:1115–1127, 1995.

[17] A. I. Elwalid. Analysis of adaptive rate-based congestion control for high-speed wide-area networks. *IEEE ICC'95*.

[18] The ATM Forum. *ATM User-Network Interface Specification Version 3.1*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[19] C. Fulton and S.-Q. Li. UT: ABR feedback control with tracking. *IEEE Infocom'97*.

[20] A. Girard. *Routing and Dimensioning in Circuit-Switched Networks*. Addison-Wesley, 1990.

[21] N. Golmie, A. Koenig, and D. Su. *The NIST ATM Network Simulator Operation and Programming*. NIST, 1995.

[22] R. Guérin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE JSAC*, Vol. 9, No. 7:968–981, 1991.

[23] S. Gupta, K. W. Ross, and M. El Zarki. *Routing in Communications Networks*, chapter 2. Prentice Hall, ed. M. Steenstrup, 1995.

[24] Ivy Hsu. *Admission Control and Resource Management for Multi-Service ATM Networks*. PhD thesis, Dept. of EECS, University of California, Berkeley, 1995.

[25] J.Y. Hui. Resource allocation for broadband networks. *IEEE JSAC*, 6:1598–1608, 1988.

[26] J.Y. Hui. *Switching and Traffic Theory for Integrated Broadband Networks*. Kluwer Acad. Publ., Boston, 1990.

[27] R.-H. Hwang. LLR routing in homogeneous VP-based ATM networks. *IEEE Infocom 95*, pages 587–593, 1995.

[28] R. Jain. Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, Oct., 1996.

[29] L. Kalampoukas and A. Varma. Dynamics of an explicit rate allocation algorithm for available bit rate service in ATM networks. Technical report, UCSS-CRL-95-54, 1995.

[30] Howard Karloff. *Linear Programming*. Birkhauser, Boston, 1991.

[31] F.P. Kelly. Routing and capacity allocation in networks with trunk reservation. *Mathematics of Operations Research*, Vol. 15, No. 4, 1990.

[32] F.P. Kelly. Effective bandwidths of multi-class queues. *Queueing Systems*, Vol. 9, No. 1:5–16, 1991.

[33] F.P. Kelly and C.N. Laws. Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing Systems*, 13:47–86, 1993.

[34] E. W. Knightly. H-bind: A new approach to providing statistical performance guarantees to VBR traffic. *IEEE INFOCOM'96 Proc.*, Vol. 3:1091–99, 1996.

[35] S.Q. Li and C.L. Hwang. Queue response to input correlation functions: Discrete spectral analysis. *IEEE/ACM Trans. Networking*, 1(5):522–533, 1993.

[36] S. Low. *Traffic Management of ATM Networks: Service Provisioning, Routing, and Traffic Shaping*. PhD thesis, Dept. of EECS, University of California, Berkeley, 1992.

[37] David G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Menlo Park, CA, 1984.

[38] D. Mitra and J. A. Morrison. Multiple time scale regulation and worst case processes for ATM network control. *Proc. of the 34th Conf. on Decision & Control*, pages 353–357, 1995.

[39] M. Montgomery and G. de Veciana. On the relevance of time scales in performance oriented traffic characterizations. *IEEE INFOCOM'96 Proc.*, Vol. 2:513–520, 1996.

[40] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara. Rate-based congestion control for ATM networks. *ACM SIGCOMM*, pages 60–72, 1995.

[41] A. Papoulis. *Probability & Statistics*. Prentice-Hall, 1990.

[42] A. Romanrow. TCP over ATM: Some performance results. *ATM Forum/93-784*, 1993.

[43] A. Romanrow and S. Floyd. The dynamics of TCP traffic over ATM networks. *IEEE JSAC*, 13, May 1995.

[44] M. Schwartz. *Broadband Integrated Networks*. Prentice-Hall, 1996.

[45] S. Shenker. Fundamental design issues for the future Internet. *IEEE JSAC*, Vol. 13 No. 7:1176–88, 1995.

[46] I. Sidhu and S. Jordan. Multiplexing gains in bit stream multiplexors. *IEEE Trans. Networking*, 3:785–797, 1995.

[47] R. Siebenhaar. Multiservice call blocking approximations for virtual path based ATM networks with CBR and VBR traffic. *IEEE Infocom 95*, pages 321–329, 1995.

[48] K. Y. Siu and H. Y. Tzeng. Intelligent congestion control for ABR services in ATM networks. *Computer Communication Review*, 24(5):81–106, 1994.

[49] Jan van Tiel. *Convex Analysis: An introductory text*. John Wiley and Sons, New York, 1984.

[50] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, 1993.

[51] J. Walrand and P. Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann, San Francisco, 1996.

[52] D. E. Wrege and J. Liebeherr. Video traffic characterization for multimedia network with a deterministic service. *IEEE INFOCOM '96*, vol 2/3:537–544, 1996.

[53] N. Yamanaka, Y. Sato, and K.I. Sato. Performance limitations of leaky bucket algorithm for usage parameter control of bandwidth allocation methods. *IEICE Trans. Commun.*, Vol. E75-B, No. 2:82–86, 1992.

[54] Z. Zhang, J. Kurose, J. Salehi, and D. Towsley. Smoothing statistical multiplexing and call admission control for stored video. *IEEE JSAC*, August 1997.