Traffic Management and Design of Multiservice Networks: the Internet and ATM Networks

by

Tae-Jin Lee, B.S., M.S., M.S.E.

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 1999

Traffic Management and Design of Multiservice Networks: the Internet and ATM Networks

Approved by
Dissertation Committee:

_____

_____

_____

_____

_____

To my parents and Younsuk

Acknowledgments

First of all, I am very grateful to my advisor Dr. Gustavo de Veciana for his invaluable advice and guidance throughout my work. Without his encouragement, patience and commitment, this work would not have been possible. I would also like to thank Dr. Ari Arapostathis, Dr. San-qi Li, Dr. Takis Konstantopoulos and Dr. Patrick Jaillet for serving on my committee and providing wonderful suggestions.

I am indebted to my colleagues Ching-fong Su, Michael Montgomery, Jay Yang, Xun Su, Steven Weber, Philip Girolami, John Stine, Wei-Lin Yang, Trevor Sosebee, and Jian-Huei Guo for their helpful comments and discussions. I also thank Sangyoub Kim, Yetik Serbest, Roberto Vargas, Garret Okamoto, Murat Torlak, Adnan Kavak, Liang Dong, Weidong Yang, JoonHyuk Kang, and office mates of ENS 419, who made our environment friendly and alive.

I express my thanks to Dr. Semyon Meerkov and Dr. William Ribbens who generously guided my research at my early stage of graduate study in University of Michigan, Ann Arbor. I am thankful to Xi Zhang and Steven Bieser who discussed many things and helped me as friends. I would like to thank Dr. Dae-Hee Youn in Yonsei University for his invaluable inspiration, advice and support since my undergraduate years. Thanks are due to the faculties of the Dept. of Electronics Engineering in Yonsei University. I also benefited from the members of ASSP lab.

My thanks go to my friends in Austin, Korea, and other places for reaching their hands and giving their support. Especially, I wish to thank youth members of St. Andrew Taegon Kim Korean Catholic Church and Hwa-Gok-Bon-Dong Catholic Parish. They

generously provided emotional support and helped me go through the years of my study.

Most of all, I would like to thank my Lord, who have been with me always and lighted my way with His love. I would like to express my special thanks to my parents, wife Younsuk, sisters Mi-Sun, Hye-Jin and Bo-Young, parents-in-law and brothers-in-law for their prayers, encouragement, support, and love. This dissertation is dedicated to them.

TAE-JIN LEE

The University of Texas at Austin

May 1999

Traffic Management and Design of Multiservice Networks: the Internet and ATM Networks

Publication No. _____

Tae-Jin Lee, Ph.D.
The University of Texas at Austin, 1999

Supervisor: Gustavo de Veciana

This work starts by considering flow control mechanisms for rate-adaptive services in networks with a static number of connections. It spans performance and design of dynamic networks supporting rate-adaptive services, and culminates in a collection of tools and methods for designing multiservice networks. These results lead to some guidelines for the traffic management and design of networks.

We consider a flow control algorithm to allocate bandwidth for rate-adaptive services in a network with a 'fixed' number of connections subject to throughput and fairness constraints. Our algorithm achieves a max-min fair rate allocation among contending users, and has desirable properties in that it can operate in a decentralized and asynchronous manner. The algorithm is simple in that the network links make local measurements of capacity and calculate local 'explicit rates' without requiring knowledge of the number of ongoing connections. Connections will receive a bandwidth determined by the minimum explicit rate along their respective routes. We discuss its stability, convergence, and feasibility issues related to fair allocation and rate-based flow control. We also consider the role of sessions with priorities under weighted max-min fair allocation of bandwidth, and its use

for ‘ABR flow control’ in ATM networks.

We next consider the stability and performance of a model for ‘dynamic’ networks supporting rate-adaptive services. In our model connection arrivals are stochastic and have a random amount of data to send, so the number of connections in the system changes over time. In turn bandwidth allocated to connections also may change over time due to feedback control, *e.g.,* max-min fair or proportionally fair allocation of bandwidth, that reacts to congestion and implicitly to the number of ongoing connections. We prove the stability of such networks when the offered load on each link does not exceed its capacity. Simulations are used to investigate the performance, in terms of average connection delays, for various types of bandwidth allocation. Our model can be used to investigate connection level stability and performance of networks supporting rate-adaptive services. We also discuss design issues and possible methods to guarantee delay quality of service requirements to dynamic connections, as required by interactive services.

We then consider multiservice ATM networks, in which both rate-adaptive ABR and CBR services, with dynamic arrivals and departures, share a single node. This is modeled by two-dimensional Markov chain, and a matrix-geometric equation is solved to yield performance estimates for ABR connections, *i.e.,* average delay and available bandwidth. By a “separation of time scales” between ABR and CBR services, we propose an approximate solution for the steady state performance of the above Markov chain. These performance results enable joint design of networks supporting multiple services. These results are partially extended to large-scale networks to compute available bandwidth for ABR connections in a dynamically changing environment. We find an upper bound on the average minimum throughput for ABR services and show that the bound is asymptotically achieved in large-capacity networks. To further increase efficiency, we consider adjustments via network level priority by way of weighted max-min fair allocation of bandwidth.

Contents

List of Tables

List of Figures

List of Notation

$\mathcal{L}$	set of links
$\mathcal{S}$	set of sessions
$\mathcal{R}$	set of routes
$\mathcal{L}_s$	set of links a session s traverses
$\mathcal{S}_\ell$	set of sessions going through link ℓ
c_ℓ	available bandwidth or capacity of link ℓ
$e_\ell(t)$	explicit rate at link ℓ at time t
$f_\ell(t)$	total flow at link ℓ at time t
$a_s(t)$	rate of session s at time t
m_s	minimum session rate of session s
p_s	peak session rate of session s
w_s	weight to session s
$\mathcal{L}^{(i)}$	set of i^{th} level bottleneck links
$\mathcal{S}^{(i)}$	set of i^{th} level bottleneck sessions
$\mathcal{U}^{(i)}$	cumulative set of bottleneck links from level 1 to i
$\mathcal{V}^{(i)}$	cumulative set of bottleneck sessions from level 1 to i
A	$A = (A_{\ell r}, \ell \in \mathcal{L}, r \in \mathcal{R})$, 0-1 matrix describing whether link ℓ is traversed by route r
b	mean number of bits for connections to send (bits)
ν_ℓ	capacity of link ℓ, $\nu_\ell = c_\ell / b$ (connections/sec)
λ_r	arrival rate of connections on route r (connections/sec)

n_r, x_r	number of connections on route r
N_r, X_r	number of connections on route r (random variable)
a_r	bandwidth for a connection on route r
w_r	weight assigned to route r
μ_r^m	bandwidth allocated to route r under max-min fair rate allocation
μ_r^w	bandwidth allocated to route r under weighted max-min fair rate allocation
μ_r^p	bandwidth allocated to route r under proportionally fair rate allocation
$f_\ell^{(i)}$	fair share at link ℓ at i^{th} level of bottleneck hierarchy
$f^{(i)}$	fair share in the i^{th} level of bottleneck hierarchy
$q(i,j)$	transition rate from state i to j
Q	infinitesimal generator (rate matrix)
V, W	Lyapunov functions
$\Delta V(x)$	$(\partial V(x)/\partial x_r, r \in \mathcal{R})$
$\Delta^2 V(x)$	$(\frac{\partial^2 V}{\partial x_r \partial x_s}(x), r, s \in \mathcal{R})$
$\pi(i,j)$	stationary distribution of i CBR and j ABR connections
r	reserved bandwidth for ABR connections
Π_j	$(\pi(0,j)\, \pi(1,j)\, \cdots\, \pi(C,j))$, vector of stationary distribution with j ABR connections
η_i	effective service rate of ABR connections under i CBR connections
ν	arrival rate of CBR connections
$1/\mu$	mean connection holding time of CBR connections
λ	arrival rate of ABR connections
$1/m$	mean amount of bits for ABR connections to send
$u_s(\cdot)$	utility function for connection s
a_s, r_s	rate of connection s
b_ℓ	available bandwidth at link ℓ
B_ℓ	available bandwidth at link ℓ (random variable)
ν_r	arrival rate of connections on CBR route r

$1/\mu_r$	mean connection holding time of CBR connections on route r
L_r	loss probability of CBR connections on route r
ξ_r	throughput of CBR connections on route r
E_ℓ	blocking probability of connections at link ℓ
ρ_ℓ	effective call arrival rate on link ℓ

Chapter 1

Introduction

Telephone networks, Cable TV (CATV), and the Internet have historically provided single types of service. However, these networks are quickly becoming "multiservice" networks partly driven by user demand for new services, by technological advances, and by economic factors. Some examples include internet phone, web-TV, and internet service on Asymmetric Digital Subscriber Line (ADSL).

Multiservice networks will carry various traffic types such as video, voice and data which require different qualities of service (QoS). Moreover, such networks will transfer huge volumes of traffic which will grow rapidly as the number of users increases and data intensive services are provided. The challenge to network providers is to manage high capacity networks carrying heterogeneous traffic while meeting QoS requirements, *e.g.*, bandwidth, delay and loss rate. The objective of this dissertation is to consider some aspects of this problem and to present guidelines which might be used in network design.

Both conventional loss-networks, *e.g.*, telephone networks, and packet-networks, *e.g.*, the Internet, are limited in their ability to carry heterogeneous traffic subject to various service requirements. In an attempt to envision multiservice networks with QoS guarantees, two major directions are being considered: Asynchronous Transfer Mode (ATM) networks and the Internet with differentiated services. In this dissertation, we explore both approaches, specifically as they relate to adaptive services.

ATM networks are essentially packet networks using small fixed size packets, in order to 1) reduce packetization delay with a view on delay constrained traffic such as voice/video and 2) allow the construction of large high capacity switching fabrics. However by contrast to traditional packet networks, the ATM standard sets up virtual circuits, *i.e.,* fixed paths which the cells associated with a given connection will follow. Thus ATM networks exhibit the same character as that of telephone networks. In designing the ATM standard, much attention was paid to efficiently managing bandwidth and providing QoS guarantees to users [12, 14, 29]. However, replacing current networks with new ATM network infrastructure and protocols may take time and be costly.

Another direction being pursued by researchers is to upgrade current Internet infrastructure and protocols (TCP/IP) by increasing capacity and introducing service differentiation enabling QoS guarantees [37, 15, 48, 16, 6, 36]. In this approach, it is important to understand the limitations of the current transport and IP service models, and how they might be enhanced at low cost. This approach looks promising since whole new networks and protocols need not to be built. However, it is questionable whether only minor changes to the Internet can deliver the promised differentiated services.

In both approaches, traffic management is the key element. Networks should allocate proper bandwidth to connections so as to prevent or alleviate congestion while maximizing network utilization and meeting service requirements. They should also ensure that users/connections are treated "fairly" when there is contention for bandwidth. In this context, flow control, QoS guarantees, and fairness provisioning are often closely related to one another.

We first consider a service class aimed at efficiently utilizing varying available bandwidth resulting from sharing of resources with variable rate traffic. Connections using the service class adapt their transmission rate via a flow control mechanism based on either explicit or implicit indications of congestion. Typically, applications using this service class require less stringent QoS guarantees, *e.g.,* range of bandwidth. In Chapter 2, we present a flow control mechanism for this type of service.

Typical analysis of rate adaptive services assume a fixed number of connections in the system and investigate convergence. In practice the number of connections using the network resources is in constant flux. For example, users can establish World Wide Web (WWW) connections at any time, and close the connections at will. Viewing the Internet from the transport level, we see TCP connections adapting transmission dynamically based on congestion status which would in turn reflect dynamic changes in the number of connections. Similarly in ATM networks, Available Bit Rate (ABR) service would adapt to varying available bandwidth due to changes in the number of connections as well as changes in available bandwidth for such connections. In Chapter 3, we will consider the stability of a stochastic network model which captures both the rate adaptation as well as the dynamic nature of the environment.

Next we consider the performance of networks supporting adaptive services. In particular in Chapter 4, we develop methods to control the average delay connections will experience. It is increasingly important to guarantee delays for interactive or delay-intolerant applications to be carried by the adaptive services. For example, users may withdraw when delay response is more than a few seconds in WWW applications.

The next step is to consider multiservice networks carrying adaptive services in addition to constant and variable bit rate services. In such networks, connections can be dynamic and have heterogeneous QoS requirements and traffic characteristics. Modeling is important to support design of multiservice networks. In Chapter 5, we analyze the performance and consider the dimensioning of multiservice networks. Finally we conclude and present future research directions in Chapter 6.

Chapter 2

Flow Control of Networks Supporting Adaptive Services

2.1 Introduction

As various types of traffic and QoS are expected to be carried in integrated services networks (*e.g.*, ATM networks), traffic control and management are becoming increasingly important. In this context, flow control is playing a prominent role. Its fundamental roles are congestion control and fairness provisioning. Since network resources are shared by many connections, it is important to decide on a policy dictating how the resources are allocated while achieving high utilization of network resources.

The question of whether flow control mechanisms should (or would) achieve a 'fair' allocation of resources among users sharing a network, has been the focus of both intensive research and debate [12]. There are currently two major views on the meaning of fairness, leading to alternative approaches to network control. The first, called *max-min fairness*, attempts to make the network transparent to users, *i.e.*, resources are allocated so as to maximize the minimum throughput of users contending for network resources [8]. More general definitions of this type of fairness, might give priority, or weights to users, but have essentially the same structure [23]. The second approach, is economic in nature, and at-

tempts to allocate resources so as to maximize the sum of the user's utilities - assuming such utility functions are available. Kelly [30] refers to the associated allocation as being *proportionally fair* and discusses cases where these two criteria coincide. Intuitively, in this case the throughput achieved by various users will in general depend on the number of bottleneck links the connections share. In a sense, max-min fairness attempts to maximize the worst case *individual* user performance, while the second approach maximizes the network's *overall* utility to users at the possible expense of some individuals.

We will focus on the problem of achieving max-min fairness. While there has been much work in this area, we believe that many of the proposed mechanisms are not viable in a large-scale networking environment where there are strong limitations on the complexity of the algorithms that can be implemented, see *e.g.*, [14].

Our starting point is a simple mechanism for flow control proposed in [21]. The rationale for the mechanism is as follows: suppose that n connections share a link with capacity c. If the capacity is to be shared evenly by the connections, then the fair rate $e(t)$

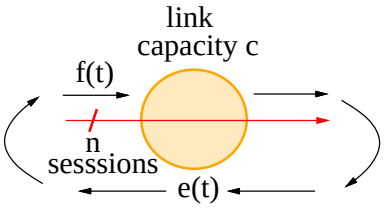

Figure 2.1: A network with one link and n sessions (unconstrained sessions).

for each session, called "explicit rate", should be c/n. Assuming the sessions send traffic at this explicit rate, the link flow (typically measured) will be n times $e(t)$, *i.e.*, $f(t) = ne(t)$. Now, since the number of active connections n may be unknown, we might estimate the number implicitly rather than monitoring it explicitly as other rate-based control schemes do [14, 57]. One can estimate the number of active connections using $\hat{n}(t) = f(t)/e(t)$. The explicit rate is then computed based on the estimated number, *i.e.*, $e(t+1) = c/\hat{n}(t)$. Due to the capacity constraint, it is desirable to ensure that $e(t)$ can not exceed the link capacity c, that is, $e(t+1) = \min[\, c/\hat{n}(t), c\,]$. It may be preferable to limit the $e(t)$ to be

even smaller than c, *e.g.*, peak session rate for any connection in the network. We will see that this surprisingly simple mechanism can be extended to a network setting.

We consider a network consisting of a set of buffered links $\mathcal{L}$ each with a (typically measured) current bandwidth availability $\vec{c} = (c_\ell, \ell \in \mathcal{L})$. Suppose a set $\mathcal{S}$ of sessions share the network, where each session $s \in \mathcal{S}$ has a set of links $\mathcal{L}_s$ associated with it. The set $\mathcal{L}_s$ is intended to define an end-to-end connection through the network. More than one session might share each link, thus we let $\mathcal{S}_\ell$ be the set of sessions crossing link ℓ. [1]

Suppose each link $\ell \in \mathcal{L}$ measures the aggregate flow $f_\ell(t)$ it is currently supporting, and computes a local 'explicit rate' $e_\ell(t)$ based on an estimated effective number of connections. In a scenario with greedy sources the session rates $a_s(t)$ are adjusted to be the smallest among the explicit rates of all links along the route of the session. So the rate adjustment for the sessions and the aggregate flows are captured by the following iterative algorithm, which extends the idea of computing explicit rate in the single link network:

$$\begin{array}{rcll} e_\ell(t+1) & = & \min\left[\, \frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell \,\right], & \ell \in \mathcal{L} \\ f_\ell(t) & = & \sum_{s \in \mathcal{S}_\ell} a_s(t), & \ell \in \mathcal{L} \\ a_s(t) & = & \min_{\ell \in \mathcal{L}_s}[\, e_\ell(t)\,], & s \in \mathcal{S}. \end{array} \tag{2.1}$$

The goal of this type of rate adjustment is to ensure that capacities are fully exploited while achieving max-min fair rate allocation. Note that the explicit rate at each link ℓ is updated in a *decentralized* manner using local information c_ℓ, $e_\ell(t)$ and $f_\ell(t)$ and exchanges of information along each session's path (rate adjustments) rather than requiring exchanges of global states, *e.g.*, whether each session is constrained or not at the link. The algorithm has clear advantages in terms of minimizing the information required to determine the max-min fair allocation in that 1) it need not keep track of the number of active connections and 2) it need not maintain information on which sessions are constrained at each link.

In §2.2 we formally define some notions related to max-min fairness that will be useful in the sequel, and in §2.3 we show that the iterative algorithm (2.1), wherein explicit rate updates are synchronous, has a unique fixed point and it achieves max-min fair

[1] In general one might allow for a multi-point session, say s, by allowing the set $\mathcal{L}_s$ to be a rooted tree on the network.

bandwidth allocation. Moreover we present a totally asynchronous version of the algorithm and its convergence to the same max-min fairness. Next we consider the role of round trip delays between sources and links, and extend the algorithm to one with session priorities leading to the notion of *weighted fairness*. As a specific application of this framework, we consider *rate-based flow control for ABR traffic* in ATM networks in §2.4. We conclude and re-evaluate the issue of fairness in §2.5.

2.2 Max-min Fairness

The main idea underlying max-min fairness can be explained as follows: each connection crossing a link should get as much bandwidth as other such connections unless that session is constrained elsewhere. In other words, available resources are allocated equally among unconstrained sessions. Max-min fairness has the following characteristics:

- each session has a bottleneck link;
- and, unconstrained sessions at a given link are given their equal share of the available capacity.

To formally define max-min fairness, we will use the following bottleneck property [8]:

Definition 2.2.1 (Bottleneck Property) *A session s has a 'bottleneck' link, if there exists a link $\ell \in \mathcal{L}_s$ such that $f_\ell^* = c_\ell$ and $a_s^* \geq a_r^*$ for all sessions $r \in \mathcal{S}_\ell$ traversing ℓ.*

Based on the bottleneck property, max-min fairness can be defined as follows [8]:

Theorem 2.2.1 (Max-min Fairness) *A session rate allocation $\vec{a}^* = (a_s^*, s \in \mathcal{S})$ is 'max-min fair' if for each session $s \in \mathcal{S}$, a_s^* can not be increased without decreasing a_r^* for some session r for which $a_r^* \leq a_s^*$. Equivalently, $\vec{a}^*$ is max-min fair if and only if each session has a bottleneck link. Moreover the max-min fair allocation $\vec{a}^*$ is unique.*

It will be useful to consider the max-min fair allocation in terms of a *hierarchy* of sets of bottleneck links and sessions [23] and *fair shares*. We define the fair share $x_\ell^1 =$

c_ℓ/n_ℓ^1 at a link $\ell \in \mathcal{L}$ as a fair partition of capacity at the link in the 1^{st} level of the hierarchy, where $n_\ell^1 = |\mathcal{S}_\ell|$ is the number of sessions through ℓ. The set of 1^{st} level bottleneck links and sessions is defined as follows:

$$\begin{aligned} \mathcal{L}^{(1)} &= \{\ell \in \mathcal{L} \mid f_\ell^* = c_\ell \text{ and for all } s \in \mathcal{S}_\ell, a_s^* = x^1 = \min_{m \in \mathcal{L}} x_m^1\}, \\ \mathcal{S}^{(1)} &= \{s \in \mathcal{S} \mid s \in \mathcal{S}_\ell \text{ and } \ell \in \mathcal{L}^{(1)}\}. \end{aligned} \tag{2.2}$$

Thus $\mathcal{L}^{(1)}$ is the set of 1^{st} level bottleneck links such that the sessions in $\mathcal{S}^{(1)}$ traversing these links are allocated the minimum bandwidth ('fair share') in the network, *i.e.*, for $s \in \mathcal{S}^{(1)}, a_s^* = \min_{r \in \mathcal{S}} a_r^* = x^1$. These two sets make up the 1^{st} level of the bottleneck hierarchy.

The next level of the hierarchy is obtained by applying the same procedure to a reduced network. The reduced network is obtained by removing the sessions in $\mathcal{S}^{(1)}$. The capacity at each link in $\mathcal{L} \setminus \mathcal{L}^{(1)}$ traversed by sessions in $\mathcal{S}^{(1)}$ is reduced by the bandwidth allocated to these sessions. The bottleneck links $\mathcal{L}^{(1)}$ are also removed from the network. Thus $\mathcal{L}^{(2)}$ and $\mathcal{S}^{(2)}$ are obtained based on a network with fewer links and sessions and adjusted capacities. The set of these bottleneck links and sessions can now be defined as follows using the notion of fair share.

Let $\mathcal{U}^{(i)} = \cup_{j=1}^{i} \mathcal{L}^{(j)}$ and $\mathcal{V}^{(i)} = \cup_{j=1}^{i} \mathcal{S}^{(j)}$ be the cumulative set of bottleneck links and sessions, respectively, in levels 1 to i of the hierarchy. The fair share x_ℓ^i $(i \geq 2)$ of link ℓ in $\ell \in \mathcal{L} \setminus \mathcal{U}^{(i-1)}$ is defined as a fair partition of available capacity at the link in the i^{th} level of the hierarchy:

$$x_\ell^i = \frac{c_\ell - \alpha_\ell^{i*}}{n_\ell^i}, \tag{2.3}$$

where $\alpha_\ell^{i*} = \sum_{s \in \mathcal{S}_\ell \cap \mathcal{V}^{(i-1)}} a_s^*$ is the total flow of sessions through ℓ which are constrained by bottleneck links in $\mathcal{U}^{(i-1)}$, and $n_\ell^i = |\mathcal{S}_\ell \setminus \mathcal{V}^{(i-1)}|$, where $n_\ell^i > 0$, is the number of sessions through ℓ which are unconstrained by the links in $\mathcal{U}^{(i-1)}$ (see Figure 2.2). Based on the fair share, the set of i^{th} level $(i \geq 2)$ bottleneck links and sessions can be defined as:

$$\begin{aligned} \mathcal{L}^{(i)} &= \{\ell \in \mathcal{L} \setminus \mathcal{U}^{(i-1)} \mid f_\ell^* = c_\ell \text{ and for all } s \in \mathcal{S}_\ell, a_s^* = x^i = \min_{m \in \mathcal{L} \setminus \mathcal{U}^{(i-1)}} x_m^i\}, \\ \mathcal{S}^{(i)} &= \{s \in \mathcal{S} \setminus \mathcal{V}^{(i-1)} \mid s \in \mathcal{S}_\ell \text{ and } \ell \in \mathcal{L}^{(i)}\}. \end{aligned} \tag{2.4}$$

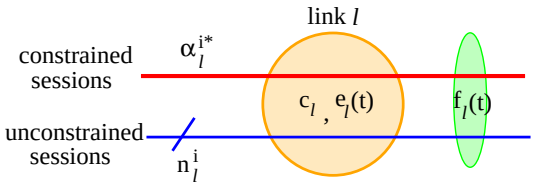

Figure 2.2: A link with constrained and unconstrained sessions in the i^{th} bottleneck level.

Here $\mathcal{L}^{(i)}$ is the set of i^{th} level bottleneck links such that the sessions in $\mathcal{S}^{(i)}$ are allocated the minimum fair share in the reduced network, *i.e.,* for $s \in \mathcal{S}^{(i)}, a_s^* = \min_{r \in \mathcal{S} \setminus \mathcal{V}^{(i-1)}} a_r^* = x^i$. Note that $x^i = x_\ell^i$ for $\ell \in \mathcal{L}^{(i)}$ is the fair share at the bottleneck links at the i^{th} level in the hierarchy.

We repeat this procedure until we exhaust all the links and sessions resulting in a hierarchy of bottleneck links and corresponding sessions $\mathcal{L}^{(1)}, \cdots, \mathcal{L}^{(N)}$ and $\mathcal{S}^{(1)}, \cdots, \mathcal{S}^{(N)}$, which is uniquely defined by (2.3) and (2.4), where N is the number of levels in the hierarchy. We will use the notion of 'bottleneck hierarchy' and 'fair share' in the sequel.

2.3 Analysis of Algorithm

We shall show that the fixed point equation associated with the iterative algorithm (2.1) has a unique solution which is the max-min fair allocation in §2.3.1. The iterative synchronous algorithm is shown to converge geometrically to the fixed point in §2.3.2, and an asynchronous version of the algorithm is also shown to converge in §2.3.3.

In practice, the explicit rate indications $e_\ell(t)$ of links will experience delays while they propagate back to the sources and until they are eventually reflected in the incident flows on the link. We assume in §2.3.2 and §2.3.3 that newly modified explicit rates at time t appear by the time the update is made in the link flow $f_\ell(t)$ without delay. That is the link flow reflects the explicit rates computed at time t. This condition is relaxed in §2.3.4. As a generalization, the algorithm with session priorities is also presented and the issue of *feasibility*, *i.e.,* maintaining link flows not exceeding link capacities is discussed.

We shall assume the following:

Assumption 2.3.1 (Bottleneck Link Assumption) *Each bottleneck link has at least one session for which it is the unique bottleneck link.*

This implies that sessions might have more than one bottleneck link, but if this is the case, each of the bottleneck links should carry at least one session for which it is the unique bottleneck. Assumption 2.3.1 is a little weaker than that in [8], but more generalized than that in [23], wherein it is assumed "single" bottleneck link per session.

2.3.1 Existence and Uniqueness

Define $\vec{e} = (e_\ell, \ell \in \mathcal{L})$ and $\vec{a} = (a_s, s \in \mathcal{S})$. Consider the following fixed point equation derived from the iterative algorithm (2.1)

$$\vec{e} = g(\vec{e}) = (g_\ell(\vec{e}), \ell \in \mathcal{L}) \tag{2.5}$$

where

$$e_\ell = \min\left[\ \frac{c_\ell e_\ell}{f_\ell}, c_\ell\ \right] = g_\ell(\vec{e}) \quad \text{for all} \quad \ell \in \mathcal{L}, \tag{2.6}$$

and where

$$f_\ell = \sum_{s \in \mathcal{S}_\ell} a_s,\ \ell \in \mathcal{L} \qquad \text{and} \qquad a_s = \min_{\ell \in \mathcal{L}_s}[\ e_\ell\ ],\ s \in \mathcal{S}. \tag{2.7}$$

We show the existence and uniqueness of a solution $\vec{e}^*$ to the fixed point equation (2.6), and further establish that the corresponding rate allocation $\vec{a}^*$ obtained by (2.7) is unique and satisfies the max-min fairness criterion.

Theorem 2.3.1 (Existence and Uniqueness) *Suppose Assumption 2.3.1 holds, then the fixed point equation (2.6) has a unique solution $\vec{e}^* = (e^*_\ell, \ell \in \mathcal{L})$. The associated session rates $\vec{a}^* = (a^*_s, s \in \mathcal{S})$ satisfy the max-min fairness criterion, and are thus unique.*

Proof: Let $\vec{0}$ denote zero vector with same dimension as $|\mathcal{L}|$. Since $\mathbb{E} = \{\vec{e} \in \mathbb{R}^{|\mathcal{L}|} |\ \vec{0} \leq \vec{e} \leq \vec{c}\}$ is compact and $g : \mathbb{E} \to \mathbb{E}$ is continuous, it follows by the Brouwer Fixed Point Theorem [9] that (2.6) has at least one solution. It follows from (2.6) that for any link $\ell \in \mathcal{L}$,

$$e^*_\ell = \min\left[\ \frac{c_\ell e^*_\ell}{f^*_\ell}, c_\ell\ \right] \Rightarrow \begin{cases} e^*_\ell = c_\ell & \text{if } f^*_\ell < c_\ell \\ e^*_\ell = c_\ell \frac{e^*_\ell}{f^*_\ell} & \text{if } f^*_\ell = c_\ell, \end{cases}$$

thus we have that

$$f_\ell^* = \sum_{s \in \mathcal{S}_\ell} a_s^*,\ \ell \in \mathcal{L} \qquad \text{and} \qquad a_s^* = \min_{\ell \in \mathcal{L}_s} [\ e_\ell^*\ ],\ s \in \mathcal{S}.$$

We will show that the session rate allocation $\vec{a}^*$ corresponds to a max-min fair allocation. Consider an arbitrary session $s \in \mathcal{S}$, we show that it has at least one bottleneck link. Consider $s \in \mathcal{S}$, then $a_s^* = \min_{\ell \in \mathcal{L}_s} [\ e_\ell^*\ ] = e_{\ell^*}^*$ for some $\ell^* \in \mathcal{L}_s$. Suppose that the link flow $f_{\ell^*}^* < c_{\ell^*}$, but then $a_s^* = e_{\ell^*}^* = c_{\ell^*}$, which contradicts $f_{\ell^*}^* < c_{\ell^*}$. Thus $f_{\ell^*}^* = c_{\ell^*}$. Now consider the sessions through ℓ^*. For each such session $r \in \mathcal{S}_{\ell^*}$, either $a_r^* = e_{\ell^*}^* = a_s^*$ ("constrained" at link ℓ^*) or $a_r^* < e_{\ell^*}^* = a_s^*$ (constrained elsewhere). Thus $a_s^* \geq a_r^*$ for all $r \in \mathcal{S}_{\ell^*}$, whence ℓ^* is a bottleneck link for session s. Therefore, $\vec{a}^*$ is a max-min fair allocation which is unique by Theorem 2.2.1.

Now, consider a solution $\vec{e}^*$. The explicit rate $e_{\ell^*}^*$ at each bottleneck link ℓ^* must be unique since by Assumption 2.3.1 the link is the only bottleneck for at least one session s of which the session rate is unique, *i.e.*, $a_s^* = \min_{\ell \in \mathcal{L}_s} [\ e_\ell^*\ ] = e_{\ell^*}^*$, and the explicit rate of non-bottleneck link is its link capacity c_ℓ which is unique. So the uniqueness of the solution $\vec{e}^*$ follows. ■

2.3.2 Synchronous Iterative Algorithm without Delayed Information

In this subsection, we assume that explicit rate updates and flow adjustments occur synchronously on some discrete time step. In other words, the explicit rates at links are updated exactly at the same time. Based on Assumption 2.3.1, we prove the following result in Appendix 2.6.

Theorem 2.3.2 (Convergence of Synchronous Iterative Algorithm) *Suppose Assumption 2.3.1 holds, then the explicit rates $\vec{e}(t) = (e_\ell(t), \ell \in \mathcal{L})$ in the iteration (2.1) converge geometrically to the fixed point $\vec{e}^*$ and the associated session rates $\vec{a}^*$ achieve the max-min fair rate allocation.*

The proof of Theorem 2.3.2 uses the following ideas. Consider a link ℓ whose flow consists of constrained and unconstrained sessions, see Fig. 2.3. Neither the constrained

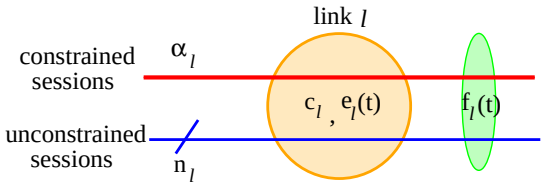

Figure 2.3: Constrained and unconstrained sessions on a link ℓ.

flow α_ℓ nor the number of unconstrained connections n_ℓ are known explicitly. The explicit rate update is given by

$$e_\ell(t+1) = \min\left[\frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell\right] = \min\left[\frac{c_\ell e_\ell(t)}{\alpha_\ell + n_\ell e_\ell(t)}, c_\ell\right] \triangleq g_\ell(e_\ell(t)).$$

It can be shown that $g_\ell(.)$ is a pseudo-contraction [9] and $e_\ell(t+1) = g_\ell(e_\ell(t))$ is a pseudo-contracting iteration converging to e_ℓ^*, the fixed point of $g_\ell(.)$. Note that we do not have a fixed point at zero if we start from non-zero $e_\ell(0)$ since $e_\ell(t+1) \geq e_\ell(t)$ when $e_\ell(t) \leq e_\ell^*$ for all t. Fig. 2.4 shows how the pseudo-contracting property arises. Thus

$$|e_\ell(t+1) - e_\ell^*| \quad \leq \quad \xi_\ell |e_\ell(t) - e_\ell^*|, \qquad 0 < \xi_\ell < 1,$$

where $e_\ell^* = (c_\ell - \alpha_\ell)/n_\ell$ is the fair share of the remaining capacity $(c_\ell - \alpha_\ell)$ to be allocated

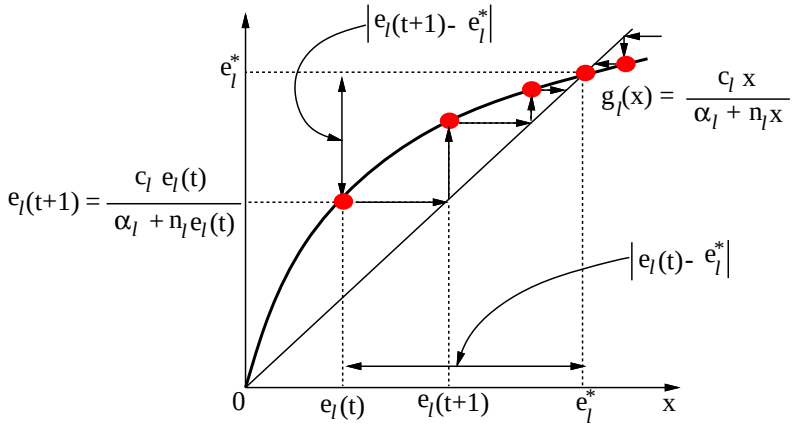

Figure 2.4: A pseudo-contraction of $g_\ell(.)$.

to the n_ℓ unconstrained sessions (see §2.2 for the definition of 'fair share'). We can show

similar pseudo-contraction properties in the network setup using the bottleneck hierarchy.

At each level of the bottleneck hierarchy, the explicit rates of the associated bottleneck links can be shown to eventually have lower and upper bounds,

$$\underline{e}_\ell(t) \leq e_\ell(t) \leq \overline{e}_\ell(t) \text{ for } \ell \in \mathcal{L}^{(i)},$$

such that converge geometrically to the fair share $e^*_\ell = x^i = (c_\ell - \alpha^{i*}_\ell)/n^i_\ell$ for $\ell \in \mathcal{L}^{(i)}$ at the i^{th} bottleneck level. So the explicit rates of i^{th} level bottleneck links converge to e^*_ℓ geometrically. We can show that the algorithm quickly achieves max-min fairness using these properties by induction on the bottleneck hierarchy. Furthermore, the explicit rates of non-bottleneck links $e_m(t)$ converge to the link capacities c_m geometrically.

Based on the previous result, we can construct a box in a space of dimension $|\mathcal{L}|$ at each time t by taking the maximum of geometric converging sequences among all the links, see Lemma 2.6.1. The box shrinks as updates proceed, and it includes all the possible explicit rates at a specific time t, so that any sequence of explicit rates converge to the fair shares or link capacities. These boxes provide the foundation for proving that asynchronous updates will converge as will be discussed in the following subsection.

2.3.3 Asynchronous Iterative Algorithm without Delayed Information

In the synchronous algorithm, updates of the explicit rates at links are assumed to be perfectly synchronized. In practice, this is unlikely to be the case, so next we consider how asynchronism would affect convergence. We use the asynchronous model in [9] to formulate a totally asynchronous version of the algorithm and prove its convergence.

Each link $\ell \in \mathcal{L}$ may not have access to the most recent values of components of $\vec{e}$. That is the flow on link ℓ may reflect old information about other links' states. Let T^ℓ denote a set of times at which e_ℓ is updated. We shall assume that there is a set of times $T = \{0, 1, 2, \cdots\}$ at which one or more components of $\vec{e}(t)$ are updated. An asynchronous

iteration can be described by

$$e_\ell(t+1) = \begin{cases} \min\left[\ \frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell\ \right] \stackrel{\triangle}{=} g_\ell(\vec{e}(t)), & t \in T^\ell \\ e_\ell(t), & \text{otherwise.} \end{cases} \tag{2.8}$$

Note that $f_\ell(t)$ depends on the possibly outdated explicit rates indication in the network, *i.e.,*

$$\begin{aligned} f_\ell(t) &= h_\ell\left(e_1(\tau_1^\ell(t)), e_2(\tau_2^\ell(t)), \cdots, e_\ell(\tau_{|\mathcal{L}|}^\ell(t))\right) \\ &= \sum_{s\in\mathcal{S}_\ell} \min_{m\in\mathcal{L}_s} [\ e_m(\tau_m^\ell(t))\ ], \end{aligned}$$

where $\tau_m^\ell(t)$ is the most recent time for which e_m is known to link ℓ through incident flow $f_\ell(t)$ at the link (see (2.1)), $0 \le \tau_m^\ell(t) \le t$ for all $t \in T$ and $\tau_\ell^\ell(t) = t$ for all $t \in T^\ell$. In the asynchronous iterative algorithm, the explicit rate e_ℓ is updated using the link flow carrying explicit rates $e_m(\tau_m^\ell(t))$ known to ℓ when $t \in T^\ell$, otherwise it remains unchanged. It is assumed here that $\tau_m^\ell(t) \to \infty$ as $t \to \infty$. This assumption implies that every link updates its explicit rate infinitely often as $t \to \infty$. In this case following theorem proven in Appendix 2.7 applies.

Theorem 2.3.3 (Convergence of Asynchronous Iterative Algorithm) *The explicit rates $\vec{e}(t)$ in the asynchronous implementation proposed in (2.8) converge to the fixed point $\vec{e}^*$ of (2.6) and the associated session rates $\vec{a}(t)$ converge to the max-min fair rates $\vec{a}^*$.*

Asynchronous convergence ensures that although links update explicit rates independently, the allocation will converge as in the synchronous algorithm, though it may take longer to do so.

In §2.3.2 and §2.3.3, we considered the convergence of synchronous and asynchronous decentralized updates based on local information. In practice delays will be incurred in the communication between sources and links. We consider the role of the delays in the following subsection.

2.3.4 Iterative Algorithm with Round Trip Delays

In the preceding analysis, the link flow $f_\ell(t)$ was assumed to be the sum of session rates $a_s(t)$ traversing link ℓ. The session rates were in turn assumed to be $a_s(t) = \min_{\ell \in \mathcal{L}_s}[\, e_\ell(t)\,]$, *i.e.*, the incident flows at time t reflect the computed explicit rates at the time t with no delay. In reality, the link flows would be immediately measured at each link, and would depend on *delayed* explicit rate indications computed at links and sent back to sources in order to control the source rates. We will present an example to show the oscillations that arise due to propagation delay.

Consider the network shown in Fig. 2.5 with one link shared by two sessions. Suppose the link capacity is $c_\ell = 1$, the initial explicit rate is $e_\ell(0) = 0.25$, and the *Round*

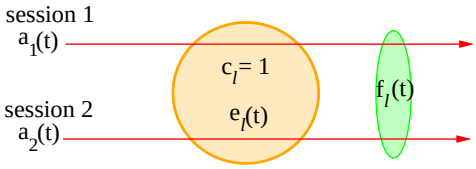

Figure 2.5: Network example with two ABR sessions with round trip delay.

Trip Delay (RTD) is assumed to be 1 time unit for both sessions. Thus the explicit rate takes at most 1 unit of time to propagate back to the sources and be reflected in the incident flow on the link, *i.e.*, $f_\ell(t) = 2e_\ell(t-1)$. The explicit rate update would be

$$e_\ell(t+1) = \min\left[\frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell\right] = \min\left[\frac{e_\ell(t)}{2e_\ell(t-1)}, 1\right],$$

which results in the oscillation shown in Fig. 2.6.

One way of preventing oscillation is to update the explicit rate at each link only after the worst case RTD, D_ℓ has elapsed, where D_ℓ is the worst case RTD of the sessions sharing link ℓ. In other words, explicit rate $e_\ell(t)$ is updated only after the link receives newly modified source rates regulated by the last computed local explicit rate. This scheme can be shown to converge to the same max-min fair allocation. The explicit rate update of link ℓ is then

$$e_\ell(t+1) = \min\left[\frac{c_\ell e_\ell(t-D_\ell)}{f_\ell(t)}, c_\ell\right]. \tag{2.9}$$

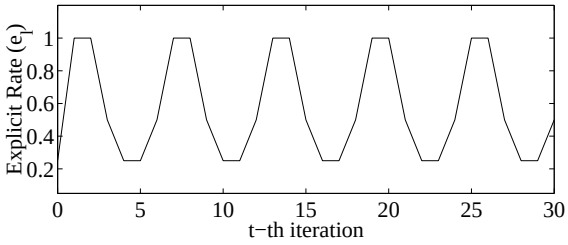

Figure 2.6: Oscillation of explicit rate in the network example without considering RTD.

Thus stability can be achieved by delaying updates, or alternatively as suggested in [21] by damping the measurements and computation. The proof of convergence is the same as that of the synchronous convergence result stated in Theorem 2.3.2.

2.3.5 Feasibility Issue of Rate Control Mechanism

An allocation is said to be *feasible* if the link flows do not exceed the link capacities. In our algorithm, link flow may temporarily exceed capacity causing queue buildups. For example, Fig. 2.7 and 2.8 show a case where a new session 5 is setup after the other sessions in the network have reached the max-min fair allocation. The infeasibility can be mitigated by

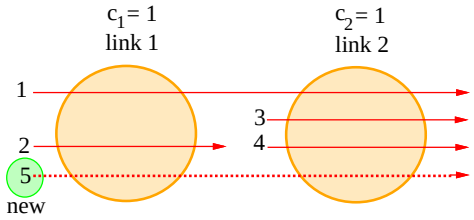

Figure 2.7: A network with a new session 5.

damping the computation of explicit rates. Damping of explicit rates by network adjustment will lessen the abrupt ramp-up or down of rates, and allow sufficient time for the network to adapt to the varying session rates and link flows and presumably prevent from excessive infeasibility. It can be shown that the damped version of the algorithm also converges to the solution of the algorithm without damping by similar steps followed in the proof of

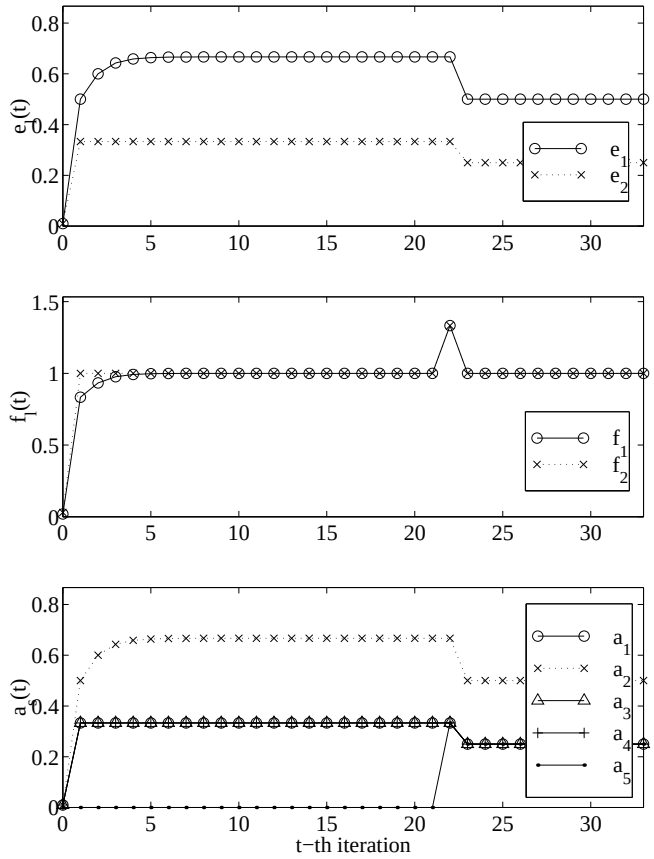

Figure 2.8: Explicit rates, link flows and session rates when a new session 5 is setup: before the new session is setup, it achieves max-min fair allocation $\vec{a}^* = (\frac{1}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3})$, and quickly adjusts to its new max-min fairness $\vec{a}^* = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ after the new session.

Theorem 2.3.2.

Another approach to manage the variability in a dynamic environment is to constrain sources to make slow rate adjustments particularly upon entering and increasing their rates: the session rates can not be increased rapidly when they are admitted to a network, rather they are permitted to increase their rates by only a little amount at a time so that the network will have sufficient time to recognize the number of connections by measuring the flow. This approach can be considered as damping source behavior.

We believe that single bit indication of queue status can be used in conjunction with

our scheme to prevent the excessive queue buildup when link flows exceed the available resource transiently. In the scheme, sources slow down their change of rates if queue starts to build up, otherwise they speed up to achieve the desired max-min fairness quickly. In a sense, the single bit indication scheme can take care of the feasibility while the explicit rate control mechanism can ensure fast convergence to fair rates. The single bit queue indication scheme might be jointly combined with the damping at sources such as linear growth of source rate. While damping of explicit rates at network links and/or damping of session rates at sources manages to keep the queue from growing beforehand, the single bit indication scheme primarily reduces the queue already built-up.

An even more conservative approach would be to employ a safety margin on available capacity. Suppose we have network utilization factor ρ, where $0 < \rho < 1$, and the link capacity to be shared is only $\bar{c}_\ell = \rho c_\ell$. We then have spare capacity $(1-\rho)c_\ell$ to absorb the transient overshoot above virtual capacity $\bar{c}_\ell$ leading to implicit control of queue buildup. The max-min fair allocation of resources would be defined with respect to the new capacities $\bar{c}_\ell, \ell \in \mathcal{L}$.

Combining these ideas, we can significantly improve the feasibility and thus performance in a dynamic network environment. There have been algorithms designed to guarantee feasibility. They, however, might also experience transient infeasibility if the instantaneous available bandwidth is highly variable, which might be typical in integrated services networks, and buffering should be provided to tackle the problem [14, 57].

2.3.6 Iterative Algorithms with Priority

It may be useful to allow sessions to have different priorities. We can formulate an iterative algorithm wherein a priority w_s, where $w_s \geq 1$, of a session s plays a role as follows:

$$\begin{aligned} e_\ell(t+1) &= \min\left[\, \tfrac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell \,\right], & \ell \in \mathcal{L} \\ f_\ell(t) &= \textstyle\sum_{s \in \mathcal{S}_\ell} a_s(t), & \ell \in \mathcal{L} \\ a_s(t) &= w_s \min_{\ell \in \mathcal{L}_s}[\, e_\ell(t)\,], & s \in \mathcal{S}. \end{aligned} \tag{2.10}$$

Note that the computation of explicit rates is still conducted locally in a decentralized manner and the priority w_s is dealt with at the source leading to the same structure of distributed computation as in the preceding algorithms.

We can now define a *bottleneck hierarchy with priority* and a notion of *weighted fair share* following a similar procedure as in §2.2. Let $\overline{\mathcal{U}}^{(i)} = \cup_{j=1}^{i} \overline{\mathcal{L}}^{(j)}$ and $\overline{\mathcal{V}}^{(i)} = \cup_{j=1}^{i} \overline{\mathcal{S}}^{(j)}$ be the cumulative set of bottleneck links and sessions, respectively, in levels 1 to i of the hierarchy with priority. The *weighted fair share* $\overline{x}_\ell^i$ can be defined as a weighted fair partition of available capacity at link ℓ in the i^{th} level of the hierarchy:

$$\overline{x}_\ell^i = \frac{c_\ell - \overline{\alpha}_\ell^{i*}}{\overline{n}_\ell^i}, \tag{2.11}$$

where $\overline{\alpha}_\ell^{i*} = \sum_{s \in \mathcal{S}_\ell \cap \overline{\mathcal{V}}^{(i-1)}} \overline{a}_s^*$ is the total flow of sessions through ℓ constrained by bottleneck links in $\overline{\mathcal{U}}^{(i-1)}$, and $\overline{n}_\ell^i = \sum_{\mathcal{S}_\ell \setminus \overline{\mathcal{V}}^{(i-1)}} w_s$ is the effective number of sessions through ℓ unconstrained by the links in $\overline{\mathcal{U}}^{(i-1)}$. Note that $\overline{\alpha}_\ell^{1*} = 0$ in the 1^{st} level of the hierarchy.

Based on the weighted fair share, the set of i^{th} level bottleneck links and sessions with priority can be defined as:

$$\begin{aligned} \overline{\mathcal{L}}^{(i)} &= \{\ell \in \mathcal{L} \setminus \overline{\mathcal{U}}^{(i-1)} \mid f_\ell^* = c_\ell \text{ and for all } s \in \mathcal{S}_\ell, \frac{\overline{a}_s^*}{w_s} = \overline{x}^i = \min_{m \in \mathcal{L} \setminus \overline{\mathcal{U}}^{(i-1)}} \overline{x}_m^i\}, \\ \overline{\mathcal{S}}^{(i)} &= \{s \in \mathcal{S} \setminus \overline{\mathcal{V}}^{(i-1)} \mid s \in \mathcal{S}_\ell \text{ and } \ell \in \overline{\mathcal{L}}^{(i)}\}. \end{aligned} \tag{2.12}$$

Here $\overline{\mathcal{L}}^{(i)}$ is the set of i^{th} level bottleneck links such that the sessions in $\overline{\mathcal{S}}^{(i)}$ are allocated weighted minimum fair share in the network, *i.e.*, for $s \in \overline{\mathcal{S}}^{(i)}$, $\frac{\overline{a}_s^*}{w_s} = \min_{r \in \mathcal{S} \setminus \overline{\mathcal{V}}^{(i-1)}} \frac{\overline{a}_r^*}{w_r} = \overline{x}^i$, thus each session sharing the link receives bandwidth in proportion to its priority, *i.e.*, $\overline{a}_s^* = w_s \overline{x}^i$. Note that $\overline{x}^i = \overline{x}_\ell^i$ for $\ell \in \overline{\mathcal{L}}^{(i)}$ is the weighted fair share of the bottleneck links in the i^{th} level hierarchy. The construction of the bottleneck hierarchy with priority is analogous to that of §2.2 resulting in set of bottleneck links and sessions with priority $\overline{\mathcal{L}}^{(1)}, \cdots, \overline{\mathcal{L}}^{(N)}$ and $\overline{\mathcal{S}}^{(1)}, \cdots, \overline{\mathcal{S}}^{(N)}$.

One can show that (2.10) will converge and allocate bandwidth to the sessions proportional to the weights w_s, *i.e.*, $\overline{a}_s^* = w_s \overline{x}^i$ for all $s \in \overline{\mathcal{S}}^{(i)}$, which is "weighted fair" leading to *weighted fair allocation* $\vec{\overline{a}}^* = (\overline{a}_s^*, s \in \mathcal{S})$.

2.4 ABR Flow Control

We have provided a simple flow control framework using explicit rate. In this section we discuss how this mechanism might be employed for *ABR flow control*.

Current and future multi-media applications will require high throughputs driving the deployment of high speed networks, *e.g.,* ATM networks, to carry such traffic. ABR service was defined as a new service class for ATM networks to utilize the remaining resources not used by other types of services (*e.g.,* CBR, VBR). There has been much effort devoted to the design of ABR flow control. For a survey on ABR rate-based flow control see [12, 14, 28, 29, 56] and references therein.

Several issues arise in reviewing the rate-based control algorithms:

- The *explicit rate control mechanism* has fast convergence characteristics.
- A *simple algorithm* is preferred to make the complexity of explicit rate control reasonable.
- In a large-scale network environment, a *distributed and asynchronous algorithm* is desirable.
- *Max-min fairness* [12] needs to be provided to treat connections fairly.
- It is desirable to *minimize the amount of information* required (*e.g.,* number of ongoing connections and status of links).

We can adopt the flow control mechanism for the control of ABR traffic, in which the above issues are resolved.

There are several parameters associated with each ABR session $s \in \mathcal{S}$, notably the Minimum Cell Rate (MCR), Allowable Cell Rate (ACR) and Peak Cell Rate (PCR) denoted by m_s, a_s and p_s respectively. Define $\vec{m} = (m_s, s \in \mathcal{S})$ and $\vec{a}, \vec{p}$ similarly. The allowable cell rate may be adjusted by the network/source to ensure good performance as long as $\vec{m} \leq \vec{a} \leq \vec{p}$ component-wise.

We will assume that each session s has a dedicated access link $\ell_s \in \mathcal{L}_s$ with a capacity p_s corresponding to the source's PCR. This will ensure that $a_s \leq p_s$ and make the description of algorithm simple since we consider $a_s(t) = \min_{\ell \in \mathcal{L}_s}[\, e_\ell(t)\,]$ instead of using $a_s(t) = \min_{\ell \in \mathcal{L}_s}[\, p_s, e_\ell(t)\,]$. Moreover we consider persistent greedy sessions, which transmit at their current ACR.

In the flow control mechanism, each link measures the aggregate flow $f_\ell(t)$ and computes a local 'explicit rate' parameter $e_\ell(t)$. Switching devices at links send this information to the sources by stamping Resource Management (RM) cells with the local explicit rate, if the Current Cell Rate (CCR) indication in the packet is higher than the computed explicit rate at this switch. Note the explicit rate in an RM cell is modified either on the forward or backward trip. Thus the source receives the minimum explicit rate for links along its route. The role of each source is to adjust the current ACR $a_s(t)$ so that it does not exceed the explicit rate indication carried back by RM cells, or the session's PCR constraint.

The flow control mechanism achieving max-min fairness can be applied to ABR service exactly as it is conceived when all connections have zero MCR. It has been argued how to define max-min fair allocation of bandwidth with sessions of non-zero MCR [12]. We consider two options to achieve similar notion of max-min fairness with non-zero MCR. In the first approach, we pre-allocate bandwidth corresponding to non-zero m_s to each session s and subtract m_s from the link capacity c_ℓ for $\ell \in \mathcal{L}_s$ resulting in new available capacity c'_ℓ. By applying the flow control algorithm (2.1) to the adjusted capacities, we achieve *max-min fairness above MCR*. This approach is a simple way to handle with MCR and would be formally described as

$$\begin{aligned} c'_\ell &= c_\ell - \textstyle\sum_{s \in \mathcal{S}_\ell} m_s, & \ell \in \mathcal{L} \\ e_\ell(t+1) &= \min\left[\, \tfrac{c'_\ell e_\ell(t)}{f_\ell(t)}, c'_\ell \,\right], & \ell \in \mathcal{L} \\ f_\ell(t) &= \textstyle\sum_{s \in \mathcal{S}_\ell} a_s(t), & \ell \in \mathcal{L} \\ a_s(t) &= \min_{\ell \in \mathcal{L}_s}[\, e_\ell(t)\,], & s \in \mathcal{S}. \end{aligned} \tag{2.13}$$

As another way of handling non-zero MCR, we consider MCR as priority for each session. In this case, the algorithm with priority (2.10) can be employed by replacing w_s by m_s

to weight the bandwidth allocated to sessions depending on MCRs leading to *weighted fairness by MCR*.

2.5 Summary

In this chapter, we have investigated a simple flow control mechanism using explicit rate. We have formulated decentralized iterative algorithm and we have shown that the solution of fixed point equation of the algorithm is unique and the algorithm converges geometrically to the max-min fair allocation of resources. We have proposed asynchronous version of the algorithm leading to the same max-min fairness. These algorithms operate in a distributed manner accounting for the heterogeneity of a large-scale high speed network.

It has been shown that they quickly achieve notion of global max-min fair rate allocation for contending users sharing resources through decentralized adjustment of explicit rates. The algorithms are simple in that they do not require that the links keep track of the number of constrained and unconstrained connections as some rate based flow algorithms did. Hence it has clear scalability advantage in terms of both complexity and state information. We have considered the feasibility issue and extended the algorithms so as to deal with priorities of sessions. As an application example, we have presented ABR rate-based flow control for ATM networks.

It is debatable whether specifying a uniform notion of fairness across a heterogeneous network, including access and backbone facilities makes sense. We believe a more appropriate notion of fairness might allow for a subdivision of the network into domains where resources might be allocated based on local administrative policies. For instance, a domain might want to differentiate among *local* and *transiting* flows, see Fig. 2.9. Indeed it could, for example, decide to give priority to local traffic, because it is critical at the site, or give priority to transit traffic because backbone or access bandwidth is limited and it is of utmost importance to achieve high throughput in connecting to remote locations.

Fig. 2.9 shows an interconnection of Domains 1 and 3 which give priority to transiting traffic and Domain 2 which gives priority to local traffic. We propose to consider

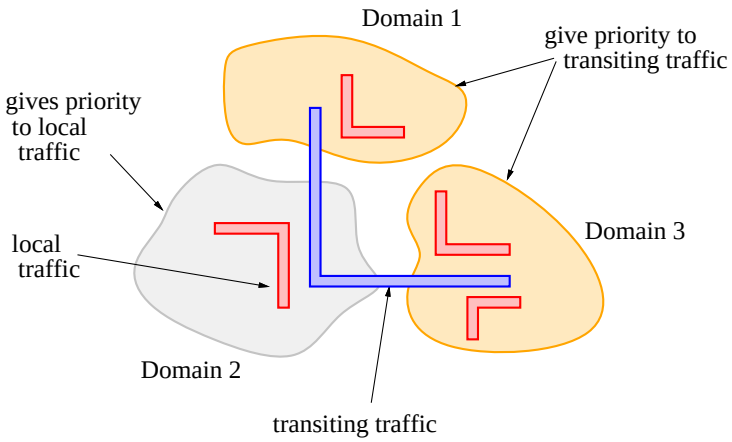

Figure 2.9: Domain fairness policies and network level interaction.

approaches at defining fairness policies within domains, and more importantly to study how interconnected domains would interact. The key issue is to characterize the equilibria, if any, of the the interconnected networks in terms of 'fairness' to users and demand in various domains. Consider for example the performance of a distributed application running over various domains, in principle it would be roughly characterized by the throughput equilibrium of the system which in turn would depend on the fairness policies of the various network components. In summary, a flexible notion of fairness should allow for possibly heterogeneous domains to define their local policies with respect to various types of flows, but would nevertheless achieve a 'consistent' notion of fairness across the internetwork.

Appendix

2.6 Proof of Theorem 2.3.2

Recall the hierarchy of bottleneck links and sessions defined in §2.2. Note that $\mathcal{U}^{(i)}$ is the cumulative set of bottleneck links in levels 1 to i and $\mathcal{V}^{(i)}$ is the cumulative set of bottleneck sessions in levels 1 to i. Theorem 2.3.2 is a consequence of Lemma 2.6.1. ■

Lemma 2.6.1 is the key lemma to show the convergence of the synchronous iterative algorithm (2.1). For that purpose, we need Lemma 2.6.2 and 2.6.3, which presents monotonicity of lower bound $\underline{e}_\ell(t)$ and upper bound $\overline{e}_\ell(t)$ of explicit rate $e_\ell(t)$. In addition, we use Lemma 2.6.4 and 2.6.5 where both lower and upper bound are shown to converge geometrically to e^*_ℓ.

Lemma 2.6.1 (Convergence of Explicit Rates and Session Rates) *Given an initial vector $\vec{e}(0)$, there exists a time t_N, where N is the number of hierarchy levels, such that for all $t \geq t_N$, the explicit rates of bottleneck links $\ell \in \mathcal{U}^{(N)}$ and the associated session rates $s \in \mathcal{V}^{(N)}$ converge geometrically to e^*_ℓ and a^*_s, respectively. Moreover, the explicit rates of non-bottleneck links $\ell \in \mathcal{L} \setminus \mathcal{U}^{(N)}$ also converge geometrically to the corresponding link capacities c_ℓ for $t \geq t_{N+1}$, where $t_{N+1} \geq t_N$, that is*

$$\begin{aligned}
\max_{\ell \in \mathcal{U}^{(N)}} |e_\ell(t) - e^*_\ell| &\leq A_N \gamma_N^t, \qquad 0 < \gamma_N < 1, \\
\max_{s \in \mathcal{V}^{(N)}} |a_s(t) - a^*_s| &\leq B_N \gamma_N^t, \qquad 0 < \gamma_N < 1, \\
\max_{\ell \in \mathcal{L} \setminus \mathcal{U}^{(N)}} |e_\ell(t) - c_\ell| &\leq A_{N+1} \gamma_{N+1}^t, \quad 0 < \gamma_{N+1} < 1.
\end{aligned}$$

Proof : We will prove this lemma by induction on the bottleneck hierarchy.

Step 1: We first show that the explicit rates of 1^{st} level bottleneck links $e_\ell(t)$ for $\ell \in \mathcal{U}^{(1)}$ and the associated session rates of 1^{st} level bottleneck sessions $a_s(t)$ for $s \in \mathcal{V}^{(1)}$ converge geometrically to e^*_ℓ and a^*_s, respectively, for $t \geq t_1$, where $\mathcal{U}^{(1)} = \mathcal{L}^{(1)}$ and $\mathcal{V}^{(1)} = \mathcal{S}^{(1)}$.

Note that for any link $m \in \mathcal{L}$ and for all $t \geq 0$,

$$\begin{aligned} f_m(t) &= \sum_{s \in \mathcal{S}_m} a_s(t) = \sum_{s \in \mathcal{S}_m} \left\{ \min_{k \in \mathcal{L}_s} e_k(t) \right\} \\ &\leq |\mathcal{S}_m| e_m(t) = n_m^1 e_m(t), \end{aligned}$$

where $n_m^1 = |\mathcal{S}_m|$ denotes the number of sessions through link m. It follows that for all $m \in \mathcal{L}$ and for all $t \geq t_1'$,

$$e_m(t+1) = \min \left[\frac{c_m e_m(t)}{f_m(t)}, c_m \right] \geq \frac{c_m}{n_m^1} = x_m^1, \tag{2.14}$$

where x_m^1 is the fair share defined in (2.3) and t_1' is the time after which all $e_m(t)$ have been updated and have achieved at least the fair share x_m^1. Thus, once $e_m(t)$ is updated, the explicit rate for link m is at least the fair share x_m^1. From now on, we consider $t \geq t_1'$.

Consider $\ell \in \mathcal{L}^{(1)}$, then the link flow $f_\ell(t)$ satisfies

$$\begin{aligned} f_\ell(t) &= \sum_{s \in \mathcal{S}_\ell} \left\{ \left(\min_{k \in \mathcal{L}_s \setminus \mathcal{L}^{(1)}} e_k(t) \right) \wedge \left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(1)}} e_k(t) \right) \right\} \\ &\geq \sum_{s \in \mathcal{S}_\ell} \left\{ \left(\min_{k \in \mathcal{L}_s \setminus \mathcal{L}^{(1)}} x_k^1 \right) \wedge \left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(1)}} e_k(t) \right) \right\}, \end{aligned} \tag{2.15}$$

where $a \wedge b$ denotes the minimum of a and b. Note that $x_\ell^1 = x^1$ for all $\ell \in \mathcal{L}^{(1)}$ and let

$$y^1 = \min_{\ell \in \mathcal{L}^{(1)}} y_\ell^1, \qquad \text{where} \qquad y_\ell^1 = \min_{s \in \mathcal{S}_\ell} \left(\min_{k \in \mathcal{L}_s \setminus \mathcal{L}^{(1)}} x_k^1 \right). \tag{2.16}$$

It follows by the bottleneck hierarchy and by the definition of fair share in §2.2 that $y^1 > x^1$ since x^1 is the fair share in the 1^{st} level bottleneck links. So (2.15) results in

$$\begin{aligned} f_\ell(t) &\geq \sum_{s \in \mathcal{S}_\ell} \left\{ y^1 \wedge \left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(1)}} e_k(t) \right) \right\} \\ &\geq (n_\ell^1 - 1) x^1 + y^1 \wedge e_\ell(t), \end{aligned} \tag{2.17}$$

where we use Assumption 2.3.1, noting that among the sessions in $\mathcal{S}_\ell$ there is at least one, say r, for which ℓ is the unique bottleneck, *i.e.*, $\mathcal{L}_r \cap \mathcal{L}^{(1)} = \ell$, and for $k \in \mathcal{L}_s \cap \mathcal{L}^{(1)}$, $e_k(t) \geq x^1$ by (2.14). It follows from (2.17) that

$$x^1 \leq e_\ell(t+1) = \min \left[\frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell \right] \leq \frac{c_\ell e_\ell(t)}{(n_\ell^1 - 1) x^1 + y^1 \wedge e_\ell(t)}.$$

Thus

$$\begin{aligned}
|e_\ell(t+1) - x^1| &= \left|\min\left[\frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell\right] - x^1\right| \le \left|\frac{c_\ell e_\ell(t)}{(n_\ell^1 - 1)x^1 + y^1 \wedge e_\ell(t)} - x^1\right| \\
&\le \frac{(n_\ell^1 - 1)x^1}{(n_\ell^1 - 1)x^1 + y^1 \wedge e_\ell(t)}|e_\ell(t) - x^1| \\
&\le \frac{(n_\ell^1 - 1)x^1}{(n_\ell^1 - 1)x^1 + y^1 \wedge x^1}|e_\ell(t) - x^1| \\
&\le \frac{c_\ell - x^1}{c_\ell}|e_\ell(t) - x^1| \\
&\le \xi_\ell|e_\ell(t) - x^1|, \qquad 0 < \xi_\ell < 1, \qquad (2.18)
\end{aligned}$$

since $c_\ell = n_\ell^1 x^1$. So by (2.18) and since $e_\ell^* = x^1$, the explicit rate $e_\ell(t)$ converges to e_ℓ^* geometrically for $\ell \in \mathcal{U}^{(1)}$, *i.e.,*

$$\max_{\ell \in \mathcal{U}^{(1)}} |e_\ell(t) - e_\ell^*| \le A_1 \gamma_1^t, \qquad 0 < \gamma_1 < 1, \qquad (2.19)$$

where A_1 is some positive constant and $\gamma_1 = \max_{\ell \in \mathcal{U}^{(1)}}[\,\xi_\ell\,]$.

Note that by (2.14) and (2.16) it follows that $e_k(t) \ge y^1$ for $k \notin \mathcal{U}^{(1)}$, and $e_k(t)$ for $k \in \mathcal{L}^{(1)}$ converges to x^1, where $x^1 < y^1$. So there exists $t_1 \ge t_1'$ such that for all $t \ge t_1$, $e_k(t) < y^1$ for $k \in \mathcal{L}^{(1)}$. Thus it follows that

$$\begin{aligned}
|a_s(t) - a_s^*| &= \left|\min_{k \in \mathcal{L}_s} e_k(t) - e_\ell^*\right| \le \left|\left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(1)}} e_k(t) \wedge \min_{k \in \mathcal{L}_s \setminus \mathcal{U}^{(1)}} e_k(t)\right) - e_\ell^*\right| \\
&\le \left|\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(1)}} e_k(t) - e_\ell^*\right| \le \max_{\ell \in \mathcal{U}^{(1)}} |e_\ell(t) - e_\ell^*| \le B_1 \gamma_1^t,
\end{aligned}$$

where B_1 is some positive constant. Hence, for $s \in \mathcal{V}^{(1)}$ and for $t \ge t_1$, the session rate $a_s(t)$ converges to a_s^* geometrically, where $a_s^* = e_\ell^*$, *i.e.,*

$$\max_{s \in \mathcal{V}^{(1)}} |a_s(t) - a_s^*| \le B_1 \gamma_1^t, \qquad 0 < \gamma_1 < 1. \qquad (2.20)$$

So the flows of 1^{st} level bottleneck links rapidly converge leaving the rest of the sessions to sort out their rates.

Step 2: Suppose that the explicit rates $e_\ell(t)$ for $\ell \in \mathcal{U}^{(i-1)}$ and the associated session rates

$a_s(t)$ for $s \in \mathcal{V}^{(i-1)}$ converge geometrically for $t \geq t_{i-1}$, *i.e.*,

$$\max_{\ell \in \mathcal{U}^{(i-1)}} |e_\ell(t) - e^*_\ell| \leq A_{i-1}\gamma^t_{i-1}, \qquad 0 < \gamma_{i-1} < 1, \tag{2.21}$$

$$\max_{s \in \mathcal{V}^{(i-1)}} |a_s(t) - a^*_s| \leq B_{i-1}\gamma^t_{i-1}, \qquad 0 < \gamma_{i-1} < 1, \tag{2.22}$$

where A_{i-1} and B_{i-1} are some positive constants. We will then show that $e_\ell(t)$ for $\ell \in \mathcal{U}^{(i)}$ and $a_s(t)$ for $s \in \mathcal{V}^{(i)}$ converge geometrically for $t \geq t_i$, where $t_i \geq t_{i-1}$.

Consider any link $m \in \mathcal{L} \setminus \mathcal{U}^{(i-1)}$,

$$\begin{aligned} f_m(t) &= \sum_{s \in \mathcal{S}_m \cap \mathcal{V}^{(i-1)}} a_s(t) + \sum_{s \in \mathcal{S}_m \setminus \mathcal{V}^{(i-1)}} a_s(t) \\ &= \sum_{s \in \mathcal{S}_m \cap \mathcal{V}^{(i-1)}} \left\{ \min_{k \in \mathcal{L}_s} e_k(t) \right\} + \sum_{s \in \mathcal{S}_m \setminus \mathcal{V}^{(i-1)}} \left\{ \min_{k \in \mathcal{L}_s} e_k(t) \right\} \\ &\leq \alpha^i_m(t) + |\mathcal{S}_m \setminus \mathcal{V}^{(i-1)}| e_m(t) = \alpha^i_m(t) + n^i_m e_m(t), \end{aligned} \tag{2.23}$$

where $n^i_m = |\mathcal{S}_m \setminus \mathcal{V}^{(i-1)}|$ is the number of sessions unconstrained by links in $\mathcal{U}^{(i-1)}$ and $\alpha^i_m(t)$ is the sum of session rates constrained by links in $\mathcal{U}^{(i-1)}$. Thus

$$e_m(t+1) = \min\left[\frac{c_m e_m(t)}{f_m(t)}, c_m \right] \geq \min\left[\frac{c_m e_m(t)}{\alpha^i_m(t) + n^i_m e_m(t)}, c_m \right] \triangleq T(\alpha^i_m(t), e_m(t)).$$

It follows by Lemma 2.6.2 that for $m \in \mathcal{L} \setminus \mathcal{U}^{(i-1)}$, there exists lower bound $\underline{e}_m(t)$,

$$e_m(t) \geq \underline{e}_m(t). \tag{2.24}$$

By Lemma 2.6.4, $\underline{e}_m(t)$ converges to x^i_m geometrically, where $x^i_m = \frac{c_m - \alpha^{i*}_m}{n^i_m}$ and x^i_m is the fair share, *i.e.*, fair amount of bandwidth among all the remaining unconstrained n^i_m sessions at link m since $(c_m - \alpha^{i*}_m)$ is the available capacity at link m. Note that we are dealing with all the links in $\mathcal{L} \setminus \mathcal{U}^{(i-1)}$ which includes i^{th} level bottleneck links $\mathcal{L}^{(i)}$, so for all $m \in \mathcal{L}^{(i)}$ lower bound $\underline{e}_m(t)$ converges to $e^*_m = x^i$, where $x^i = x^i_m$. Since there are only a finite number of links, there exists $t'_i \geq t_{i-1}$, such that for all $t \geq t'_i$ the lower bounds $\underline{e}_m(t)$ are at least $x^i_m - \varepsilon$ for arbitrarily small ε and for all $m \in \mathcal{L} \setminus \mathcal{U}^{(i-1)}$, *i.e.*,

$$e_m(t) \geq \underline{e}_m(t) \geq x^i_m - \varepsilon. \tag{2.25}$$

From now on, we consider $t \geq t_i'$.

Consider an i^{th} level bottleneck link $\ell \in \mathcal{L}^{(i)}$, by (2.23) and (2.25) we have the following:

$$\begin{aligned} f_\ell(t) &= \sum_{s \in \mathcal{S}_\ell \cap \mathcal{V}^{(i-1)}} \left\{ \min_{k \in \mathcal{L}_s} e_k(t) \right\} \\ &\quad + \sum_{s \in \mathcal{S}_\ell \setminus \mathcal{V}^{(i-1)}} \left\{ \left(\min_{k \in \mathcal{L}_s \setminus \mathcal{L}^{(i)}} e_k(t) \right) \wedge \left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(i)}} e_k(t) \right) \right\} \\ &\geq \alpha_\ell^i(t) + \sum_{s \in \mathcal{S}_\ell \setminus \mathcal{V}^{(i-1)}} \left\{ \left(\min_{k \in \mathcal{L}_s \setminus \mathcal{L}^{(i)}} x_k^i - \varepsilon \right) \wedge \left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(i)}} e_k(t) \right) \right\}. \quad (2.26) \end{aligned}$$

Note that $x_\ell^i = x^i$ for all $\ell \in \mathcal{L}^{(i)}$ and let

$$y^i = \min_{\ell \in \mathcal{L}^{(i)}} y_\ell^i, \qquad \text{where} \qquad y_\ell^i = \min_{s \in \mathcal{S}_\ell \setminus \mathcal{V}^{(i-1)}} \left(\min_{k \in \mathcal{L}_s \setminus \mathcal{L}^{(i)}} x_k^i \right) - \varepsilon. \tag{2.27}$$

It follows by the bottleneck hierarchy and by the definition of fair share in §2.2 that $y^i + \varepsilon > x^i$ since x^i is a fair share in the i^{th} level bottleneck link, and if we choose ε small enough then $y^i > x^i$.

So (2.26) results in

$$\begin{aligned} f_\ell(t) &\geq \alpha_\ell^i(t) + \sum_{s \in \mathcal{S}_\ell \setminus \mathcal{V}^{(i-1)}} \left\{ y^i \wedge \left(\min_{k \in \mathcal{L}_s \cap \mathcal{L}^{(i)}} e_k(t) \right) \right\} \\ &\geq \alpha_\ell^i(t) + (n_\ell^i - 1) \min_{k \in \mathcal{L}^{(i)}} \underline{e}_k(t) + y^i \wedge e_\ell(t) \quad (2.28) \\ &\geq \beta_\ell^i(t) + y^i \wedge e_\ell(t), \end{aligned}$$

where $\beta_\ell^i(t) = \alpha_\ell^i(t) + (n_\ell^i - 1) \min_{k \in \mathcal{L}^{(i)}} \underline{e}_k(t)$. In (2.28), we have used Assumption 2.3.1 that among the sessions in $\mathcal{S}_\ell \setminus \mathcal{V}^{(i-1)}$ there is at least one, say r, for which ℓ is the unique bottleneck, *i.e.*, $\mathcal{L}_r \cap \mathcal{L}^{(i)} = \ell$, and $e_k(t) \geq \underline{e}_k(t)$ for $k \in \mathcal{L}_s \cap \mathcal{L}^{(i)}$ by (2.24). Thus

$$e_\ell(t+1) = \min \left[\frac{c_\ell e_\ell(t)}{f_\ell(t)}, c_\ell \right] \leq \min \left[\frac{c_\ell e_\ell(t)}{\beta_\ell^i(t) + y^i \wedge e_\ell(t)}, c_\ell \right] \triangleq R(\beta_\ell^i(t), e_\ell(t)).$$

It follows by Lemma 2.6.3 that for $t \geq t_i'$ and for $\ell \in \mathcal{L}^{(i)}$, we have

$$e_\ell(t) \leq \overline{e}_\ell(t). \tag{2.29}$$

By Lemma 2.6.4 and Lemma 2.6.5, both bounds $\underline{e}_\ell(t)$ and $\overline{e}_\ell(t)$ in (2.24) and (2.29), respectively, converge to e^*_ℓ geometrically for $\ell \in \mathcal{U}^{(i)}$, *i.e.*,

$$\begin{aligned}\max_{\ell\in\mathcal{U}^{(i)}} |\underline{e}_\ell(t) - e^*_\ell| &\leq \underline{A}_i\underline{\gamma}_i^t, \quad 0 < \underline{\gamma}_i < 1,\\ \max_{\ell\in\mathcal{U}^{(i)}} |\overline{e}_\ell(t) - e^*_\ell| &\leq \overline{A}_i\overline{\gamma}_i^t, \quad 0 < \overline{\gamma}_i < 1,\end{aligned}$$

where $\underline{A}_i$ and $\overline{A}_i$ are some positive constants, so $e_\ell(t)$ converges to e^*_ℓ geometrically for $t \geq t'_i$ and for $\ell \in \mathcal{U}^{(i)}$,

$$\max_{\ell\in\mathcal{U}^{(i)}} |e_\ell(t) - e^*_\ell| \leq A_i\gamma_i^t, \quad 0 < \gamma_i < 1, \tag{2.30}$$

where $A_i = \max[\underline{A}_i, \overline{A}_i]$ and $\gamma_i = \max[\,\underline{\gamma}_i, \overline{\gamma}_i\,]$.

Note that $e_k(t)$ for $k \in \mathcal{U}^{(i-1)}$ converges to e^*_k, where $e^*_k = x^j$, for some j such that $1 \leq j \leq i-1$ by our induction hypothesis (2.21). Also, $e_k(t) \geq y^i$ for $t \geq t'_i$ and for $k \notin \mathcal{U}^{(i)}$ (see (2.25) and (2.27)). Since $e_k(t)$ for $k \in \mathcal{L}^{(i)}$ converges to x^i, where $x^i < y^i$, the explicit rate $e_k(t)$ eventually becomes smaller than y^i for large enough $t \geq t_i$, where $t_i \geq t'_i$, so it follows

$$\begin{aligned}|a_s(t) - a^*_s| &= \left|\min_{k\in\mathcal{L}_s} e_k(t) - e^*_\ell\right| \leq \left|\left(\min_{k\in\mathcal{L}_s\cap\mathcal{L}^{(i)}} e_k(t) \wedge \min_{k\in\mathcal{L}_s\setminus\mathcal{U}^{(i)}} e_k(t)\right) - e^*_\ell\right|\\ &\leq \left|\min_{k\in\mathcal{L}_s\cap\mathcal{L}^{(i)}} e_k(t) - e^*_\ell\right| \leq \max_{\ell\in\mathcal{U}^{(i)}} |e_\ell(t) - e^*_\ell| \leq B_i\gamma_i^t,\end{aligned}$$

Thus for $s \in \mathcal{V}^{(i)}$ and for $t \geq t_i$, the session rate $a_s(t)$ converges to a^*_s geometrically, where $a^*_s = e^*_\ell$, *i.e.*,

$$\max_{s\in\mathcal{V}^{(i)}} |a_s(t) - a^*_s| \leq B_i\gamma_i^t, \quad 0 < \gamma_i < 1. \tag{2.31}$$

Since we have finite number of levels N in the bottleneck hierarchy, the induction terminates at N.

Step 3: Now consider a non-bottleneck link $m \in \mathcal{L} \setminus \mathcal{U}^{(N)}$, then

$$e_m(t+1) = \min\left[\,\frac{c_m e_m(t)}{f_m(t)}, c_m\,\right],$$

where $f_m(t) = \sum_{s\in\mathcal{S}_m} a_s(t)$. It follows that the link flow $f_m(t)$ converges to f^*_m since all session rates converge to a^*_s as shown previously. Note $f^*_m < c_m$, otherwise m would

be a bottleneck link by Assumption 2.3.1. So there exists t_{N+1} such that for all $t \geq t_{N+1}$ the link flow $f_m(t) \leq f_m^* + \varepsilon < c_m$ and $e_m(t+1) \geq \min\left[\frac{c_m e_m(t)}{f_m^*+\varepsilon}, c_m\right]$. Thus, $e_m(t)$ converges to c_m for $t \geq t_{N+1}$ and for $m \in \mathcal{L} \setminus \mathcal{U}^{(N)}$. In fact, $e_m(t)$ is a pseudo-contracting sequence towards $e_m^* = c_m$,

$$|e_m(t+1) - e_m^*| \leq \xi_m |e_m(t) - e_m^*|, \qquad 0 < \xi_m < 1.$$

Thus we have that

$$\max_{m \in \mathcal{L}\setminus\mathcal{U}^{(N)}} |e_m(t) - e_m^*| \leq A_{N+1}\gamma_{N+1}^t, \qquad 0 < \gamma_{N+1} < 1, \tag{2.32}$$

where A_{N+1} is some positive constant and $\gamma_{N+1} = \max_{m\in\mathcal{L}\setminus\mathcal{U}^{(N)}}[\xi_m]$. This completes the proof. ■

Lemma 2.6.2 (Monotonicity of Lower Bounds) *Suppose*

$$e_\ell(t+1) \geq \min\left[\frac{c_\ell e_\ell(t)}{\alpha_\ell^i(t) + n_\ell^i e_\ell(t)}, c_\ell\right] \triangleq T(\alpha_\ell^i(t), e_\ell(t)),$$

then $e_\ell(t) \geq \underline{e}_\ell(t)$ *for* $t \geq t_i$*, where* $\underline{e}_\ell(t_i) = e_\ell(t_i)$ *and* $\underline{e}_\ell(t+1) = T(\alpha_\ell^i(t), \underline{e}_\ell(t))$ *for* $t \geq t_i$.

Proof : Note that $T(\alpha_\ell^i(t), \ \cdot\ )$ is a non-decreasing function in second parameter. Since $e_\ell(t_i+1) \geq T(\alpha_\ell^i(t_i), \underline{e}_\ell(t_i)) = \underline{e}_\ell(t_i+1)$, it follows by monotonicity of $T(\alpha_\ell^i(t), \ \cdot\ )$,

$$e_\ell(t_i+2) \geq T(\alpha_\ell^i(t_i+1), e_\ell(t_i+1)) \geq T(\alpha_\ell^i(t_i+1), \underline{e}_\ell(t_i+1)) = \underline{e}_\ell(t_i+2),$$

so for all $t \geq t_i$,

$$e_\ell(t) \geq \underline{e}_\ell(t).$$

■

Lemma 2.6.3 (Monotonicity of Upper Bounds) *Suppose*

$$e_\ell(t+1) \leq \min\left[\frac{c_\ell e_\ell(t)}{\beta_\ell^i(t) + y^i \wedge e_\ell(t)}, c_\ell\right] \triangleq R(\beta_\ell^i(t), e_\ell(t)),$$

then $e_\ell(t) \leq \overline{e}_\ell(t)$ *for* $t \geq t_i$*, where* $\overline{e}_\ell(t_i) = e_\ell(t_i)$ *and* $\overline{e}_\ell(t+1) = R(\beta_\ell^i(t), \overline{e}_\ell(t))$ *for* $t \geq t_i$.

Proof : Note that $R(\beta_\ell^i(t), \ \centerdot \ )$ is a non-decreasing function in second parameter. Since $e_\ell(t_i+1) \le R(\beta_\ell^i(t_i), \overline{e}_\ell(t_i)) = \overline{e}_\ell(t_i+1)$, it follows by monotonicity of $R(\beta_\ell^i(t), \ \centerdot \ )$,

$$e_\ell(t_i+2) \le R(\beta_\ell^i(t_i+1), e_\ell(t_i+1)) \le R(\beta_\ell^i(t_i+1), \overline{e}_\ell(t_i+1)) = \overline{e}_\ell(t_i+2),$$

thus for all $t \ge t_i$,

$$e_\ell(t) \le \overline{e}_\ell(t).$$

■

Lemma 2.6.4 (Convergence of Lower Bounds) *Suppose for $t \ge t_{i-1}$,*

$$\max_{\ell \in \mathcal{U}^{(i-1)}} |e_\ell(t) - e_\ell^*| \ \le \ A_{i-1}\gamma_{i-1}^t, \qquad 0 < \gamma_{i-1} < 1, \tag{2.33}$$

$$\max_{s \in \mathcal{V}^{(i-1)}} |a_s(t) - a_s^*| \ \le \ B_{i-1}\gamma_{i-1}^t, \qquad 0 < \gamma_{i-1} < 1, \tag{2.34}$$

$$\underline{e}_\ell(t+1) = T(\alpha_\ell^i(t), \underline{e}_\ell(t)) \stackrel{\triangle}{=} \min\left[\frac{c_\ell \underline{e}_\ell(t)}{\alpha_\ell^i(t) + n_\ell^i \underline{e}_\ell(t)}, c_\ell \right],$$

where $\underline{e}_\ell(t_i) = e_\ell(t_i)$ and $t_i \ge t_{i-1}$, and that

$$\underline{e}_\ell(t) \ge x^i - \varepsilon, \tag{2.35}$$

then for some positive constant $\underline{A}_i$,

$$\max_{\ell \in \mathcal{U}^{(i)}} |\underline{e}_\ell(t) - e_\ell^*| \le \underline{A}_i \underline{\gamma}_i^t, \qquad 0 < \underline{\gamma}_i < 1.$$

Proof : Consider $T(\ \centerdot \ , e)$. Note that

$$\max_{\alpha \ge 0} \left| \frac{\partial}{\partial \alpha} T(\alpha, e) \right| = \left. \frac{c_\ell e}{(\alpha + n_\ell^i e)^2} \right|_{\alpha=0} = \frac{c_\ell}{{n_\ell^i}^2 e}.$$

Since by (2.35), $e \ge x^i - \varepsilon$, letting Lipschitz constant $\underline{K} = \frac{c_\ell}{{n_\ell^i}^2 (x^i - \varepsilon)}$, we have

$$|T(\alpha_\ell^i(t), \underline{e}_\ell(t)) - T(\alpha_\ell^{i*}, \underline{e}_\ell(t))| \le \underline{K} |\alpha_\ell^i(t) - \alpha_\ell^{i*}|. \tag{2.36}$$

Furthermore, $T(\alpha_\ell^{i*}, \ \centerdot \ )$ is a pseudo-contraction [9], whose sequence converges towards $T(\alpha_\ell^{i*}, e_\ell^*) = e_\ell^* = \frac{c_\ell - \alpha_\ell^{i*}}{n_\ell^i}$, *i.e.,*

$$\begin{aligned} |T(\alpha_\ell^{i*}, \underline{e}_\ell(t)) - T(\alpha_\ell^{i*}, e_\ell^*)| &\leq \frac{\alpha_\ell^{i*}}{\alpha_\ell^{i*} + n_\ell^i \underline{e}_\ell(t)} |\underline{e}_\ell(t) - e_\ell^*| \\ &\leq \underline{\xi}_\ell |\underline{e}_\ell(t) - e_\ell^*|, \qquad 0 < \underline{\xi}_\ell < 1. \end{aligned} \tag{2.37}$$

Note by (2.34) that for $t \geq t_{i-1}$,

$$\begin{aligned} |\alpha_\ell^i(t) - \alpha_\ell^{i*}| &\leq \left| \sum_{s \in \mathcal{S}_\ell \cap \mathcal{V}^{(i-1)}} a_s(t) - \sum_{s \in \mathcal{S}_\ell \cap \mathcal{V}^{(i-1)}} a_s^* \right| \leq \sum_{s \in \mathcal{S}_\ell \cap \mathcal{V}^{(i-1)}} |a_s(t) - a_s^*| \\ &\leq C_{i-1} \gamma_{i-1}^t, \end{aligned} \tag{2.38}$$

where C_{i-1} is some positive constant. So it follows by (2.36) and (2.37) that

$$\begin{aligned} |\underline{e}_\ell(t+1) - e_\ell^*| &= |T(\alpha_\ell^i(t), \underline{e}_\ell(t)) - T(\alpha_\ell^{i*}, e_\ell^*)| \\ &\leq |T(\alpha_\ell^i(t), \underline{e}_\ell(t)) - T(\alpha_\ell^{i*}, \underline{e}_\ell(t))| + |T(\alpha_\ell^{i*}, \underline{e}_\ell(t)) - T(\alpha_\ell^{i*}, e_\ell^*)| \\ &\leq \underline{K} |\alpha_\ell^i(t) - \alpha_\ell^{i*}| + \underline{\xi}_\ell |\underline{e}_\ell(t) - e_\ell^*|, \qquad 0 < \underline{\xi}_\ell < 1. \end{aligned}$$

Thus we obtain by (2.33) and (2.38) that

$$\max_{\ell \in \mathcal{U}^{(i)}} |\underline{e}_\ell(t) - e_\ell^*| \leq \underline{A}_i \underline{\gamma}_i^t, \qquad 0 < \underline{\gamma}_i < 1,$$

where $\underline{A}_i$ is some positive constant, $\underline{\gamma}_i = \max[\gamma_{i-1}, \underline{\rho}_i]$, and where $\underline{\rho}_i = \max_{\ell \in \mathcal{L}^{(i)}}[\underline{\xi}_\ell]$. ■

Lemma 2.6.5 (Convergence of Upper Bounds) *Suppose for $t \geq t_{i-1}$,*

$$\max_{\ell \in \mathcal{U}^{(i-1)}} |e_\ell(t) - e_\ell^*| \leq A_{i-1} \gamma_{i-1}^t, \qquad 0 < \gamma_{i-1} < 1, \tag{2.39}$$

$$\max_{s \in \mathcal{V}^{(i-1)}} |a_s(t) - a_s^*| \leq B_{i-1} \gamma_{i-1}^t, \qquad 0 < \gamma_{i-1} < 1, \tag{2.40}$$

$$\max_{\ell \in \mathcal{U}^{(i)}} |\underline{e}_\ell(t) - e_\ell^*| \leq \underline{A}_i \underline{\gamma}_i^t, \qquad 0 < \underline{\gamma}_i < 1, \tag{2.41}$$

$$\overline{e}_\ell(t+1) = R(\beta_\ell^i(t), \overline{e}_\ell(t)) \stackrel{\triangle}{=} \min \left[\frac{c_\ell \overline{e}_\ell(t)}{\beta_\ell^i(t) + y^i \wedge \overline{e}_\ell(t)}, c_\ell \right],$$

where $\overline{e}_\ell(t_i) = e_\ell(t_i)$ and $t_i \geq t_{i-1}$, then for some positive constant $\overline{A}_i$,

$$\max_{\ell \in \mathcal{U}^{(i)}} |\overline{e}_\ell(t) - e_\ell^*| \leq \overline{A}_i \overline{\gamma}_i^t, \qquad 0 < \overline{\gamma}_i < 1.$$

Proof : Note by (2.40) and (2.41) that for $t \geq t_i$,

$$\begin{aligned}
|\beta_\ell^i(t) - \beta_\ell^{i*}| &\leq \left| \left(\alpha_\ell^i(t) + (n_\ell^i - 1) \min_{k \in \mathcal{L}^{(i)}} \underline{e}_k(t) \right) - \left(\alpha_\ell^{i*} + (n_\ell^i - 1) e_\ell^* \right) \right| \\
&\leq |\alpha_\ell^i(t) - \alpha_\ell^{i*}| + (n_\ell^i - 1) \left| \min_{k \in \mathcal{L}^{(i)}} \underline{e}_k(t) - e_\ell^* \right| \\
&\leq \sum_{s \in \mathcal{S}_\ell \cap \mathcal{V}^{(i-1)}} |a_s(t) - a_s^*| + (n_\ell^i - 1) |\underline{e}_\ell(t) - e_\ell^*| \\
&\leq \underline{B}_i \underline{\gamma}_i^t, \qquad (2.42)
\end{aligned}$$

where $\underline{B}_i$ is some positive constant. Following the similar steps in Lemma 2.6.4, we have

$$\begin{aligned}
|\overline{e}_\ell(t+1) - e_\ell^*| &= |R(\beta_\ell^i(t), \overline{e}_\ell(t)) - R(\beta_\ell^{i*}, e_\ell^*)| \\
&\leq |R(\beta_\ell^i(t), \overline{e}_\ell(t)) - R(\beta_\ell^{i*}, \overline{e}_\ell(t))| + |R(\beta_\ell^{i*}, \overline{e}_\ell(t)) - R(\beta_\ell^{i*}, e_\ell^*)| \\
&\leq \overline{K} |\beta_\ell^i(t) - \beta_\ell^{i*}| + \overline{\xi}_\ell |\overline{e}_\ell(t) - e_\ell^*|, \qquad 0 < \overline{\xi}_\ell < 1,
\end{aligned}$$

where $\overline{K}$ is a Lipschitz constant and $R(\beta_\ell^{i*}, \ \centerdot \ )$ is a pseudo-contraction whose sequence converges towards e_ℓ^*. Thus by (2.39) and (2.42) it follows

$$\max_{\ell \in \mathcal{U}^{(i)}} |\overline{e}_\ell(t) - e_\ell^*| \leq \overline{A}_i \overline{\gamma}_i^t, \qquad 0 < \overline{\gamma}_i < 1,$$

where $\overline{A}_i$ is some positive constant, $\overline{\gamma}_i = \max[\underline{\gamma}_i, \overline{\rho}_i]$, and where $\overline{\rho}_i = \max_{\ell \in \mathcal{L}^{(i)}} [\overline{\xi}_\ell]$, see Lemma 2.6.4. ■

2.7 Proof of Theorem 2.3.3

From Lemma 2.6.1, let $A = \max[A_N, A_{N+1}]$, $\gamma = \max[\gamma_N, \gamma_{N+1}]$, $C = \max_{\ell \in \mathcal{L}} [c_\ell]$, and

$$E(t) = \begin{cases} \{\vec{e} \mid || \vec{e} - \vec{e}^* ||_\infty \leq A\gamma^t \}, & t \geq t_{N+1}, \\ \{\vec{e} \mid || \vec{e} - \vec{e}^* ||_\infty \leq \max[A\gamma^{t_{N+1}}, C] \}, & t < t_{N+1}, \end{cases}$$

then following conditions hold.

1. We have $E(t+1) \subset E(t)$, and $g(\vec{e}) \in E(t+1)$ for all t and $\vec{e} \in E(t)$. The sequence $\{\vec{e}(t)\}$ converges to $\vec{e}^*$ by Theorem 2.3.2 (Convergence of Synchronous Iterative Algorithm).

2. The set $E(t)$ satisfies the Box Condition [9] for all t. That is there exist sets $E_i(t) \subset E_i(0)$ for all t, such that

$$E(t) = E_1(t) \times E_2(t) \times \cdots \times E_{|\mathcal{L}|}(t).$$

3. Initial explicit rate vector $\vec{e}(0)$ is in the set $E(0)$.

Thus by Asynchronous Convergence Theorem [9], the asynchronous iteration (2.8) converges to $\vec{e}^*$. ■

Chapter 3

Stability of Dynamic Networks Supporting Services with Flow Control

3.1 Introduction

Future communication networks are likely to support *elastic* applications that permit adaptation of the data transmission rate to the available network bandwidth while achieving a graceful degradation in the perceived quality of service [55]. Transport services that match the flexibility of such applications are already supported on the Internet via TCP wherein end-systems adjust their transmissions in response to delayed or lost packets, *i.e.*, implicit indicators of available bandwidth [26]. Available Bit Rate service, defined for ATM networks, draws on both the end-systems and network elements to implement a similar functionality through adaptive rate control mechanisms that strive to allocate the available bandwidth among ongoing connections [12]. Typically such mechanisms represent an efficient way to carry traffic corresponding to elastic applications, ranging from today's file transfers to future rate adaptive voice/video applications.

Since mechanisms to adapt transmission rate typically draw on delayed (implicit

or explicit) feedback from the network, much work has been devoted to establishing their stability, particularly for networks supporting a *fixed* number of connections. Stability is usually interpreted as avoiding queue/delay buildups, and/or somewhat loosely as ensuring that transmission rates converge to an equilibrium corresponding to a bandwidth allocation among ongoing connections, see *e.g.*, [3, 7, 13, 57, 38, 1, 33]. Such equilibria are in turn usually characterized in terms of their 'fairness' to users, such as max-min fairness or proportional fairness [8, 30]. Thus given a fixed number of users and fixed network capacities, one can typically arrange (through an appropriate control mechanism) to achieve an equilibrium which represents, according to some criterion, an equitable allocation of resources among users.

By contrast very little is known about the network's performance when the number of connections in the network is in constant flux. Previous work along these lines has focused on studying transients, *i.e.*, how quickly will the transmission rates reach a new equilibrium. In this chapter we consider a novel model that includes stochastic arrivals and departures. However it abstracts the queueing and rate adaptation that would be taking place in the network by assuming that an equilibrium, and thus appropriate bandwidth allocation is immediately achieved. In essence, this corresponds to assuming a *separation of time scales* between the time scales of connection arrivals and departures and those on which rate control processes converge to equilibria. Our focus is on exploring the stability and performance of this connection-level model for networks using different types of rate control and thus operated under different fairness policies.

Paralleling models used in the circuit switched literature, we assume connection arrivals processes are Poisson and that each connection has a random, exponentially distributed, amount of data to send.[1] In contrast to circuit switched models, the bandwidth allocated to each user will be a function of the global state of the network. Indeed recall that the bandwidth allocated to a user depends on the equilibrium achieved by the rate

[1]This arrivals model is a typical and a reasonable assumption for connections generated by a large population of independent users. The exponential assumption simplifies our analysis but is likely not to be critical for the stability results in this chapter.

control mechanisms and the number of ongoing connections.

In general, one expects work conserving systems to be stable when the offered load to each link (queue) in the network does not exceed its capacity. However given the complex network-wide interactions underlying the bandwidth allocation mechanism, a demonstration of this fact was an open question. Note that our model can be said to be 'non-work conserving' in the sense that a link supporting active connections may not be operating at a full capacity because its connections are 'bottlenecked' elsewhere – a typical sign of a potential for instability. In this chapter we come to terms with this problem by showing the stability of our model when natural conditions are satisfied.

Since ours is a higher layer model, it is logical to consider network-level performance, say in terms of average connection delays. This is important because the goals of fairness and low connection delays may not be compatible, and should be examined prior to committing to a particular architecture for large-scale broadband networks. Moreover network designers might want to dimension capacities to achieve a reasonable responsiveness, say for web browsing, when the network is subject to typical loads. Our preliminary simulations suggest that indeed it may be of interest to examine more carefully the impact of a given fairness criterion and topology on the overall network performance.

Based on our model we point out an insidious architectural problem in networks supporting adaptive services of this type. To achieve connection layer stability we must ensure that connection level loads do not exceed link capacities. Clearly this then requires that the routing layer be aware of the connection level offered loads. However, typical routing algorithms draw on short term link averages of utilization or packet delays. Such metrics reflect the connection level offered loads quite poorly, since connections are adapting their transmission rates depending on link congestion. Loosely speaking, the router is indifferent to the fact that a 90 % link utilization may be due to a single traffic source or a thousand sources transmitting at a thousandth of the latter's rate. Herein lies a possible explanation for the congestion currently experienced on the Internet, *i.e.,* connection level instability.

The Chapter is organized as follows. In §3.2, we present our model and define the

max-min, weighted max-min and proportionally fair bandwidth allocations. Next, in §3.3 we show the stability of the model by constructing appropriate Lyapunov functions. In §3.4 we return to our question concerning possible connection level instabilities in current networks and discuss future work.

3.2 Network Model and Bandwidth Allocation Schemes

Our network model consists of a set of links $\mathcal{L}$ with fixed capacities $c = (c_\ell, \ell \in \mathcal{L})$ in bits/sec shared by a collection of routes $\mathcal{R}$. Routes are undirected and may traverse several links in the network.[2] A 0-1 matrix $A = (A_{\ell r}, \ell \in \mathcal{L}, r \in \mathcal{R})$ indicates which links a route traverses. In other words, $A_{\ell r} = 1$ if route r uses link ℓ and zero otherwise.

The dynamics of the model are as follows. New connections are initiated on route $r \in \mathcal{R}$ at random times forming a Poisson process Π_r with rate λ_r connections/sec. The collection of processes $\Pi = \{\Pi_r, r \in \mathcal{R}\}$, with rates $\lambda = (\lambda_r, r \in \mathcal{R})$ are assumed to be independent. Each connection has a volume of data (in bits) to transmit, which is assumed to be an exponentially distributed random variable with mean b bits. The parameter b is the same for all connections, irrespective of route or arrival time. This assumption simplifies the description of the system state and, consequently, its analysis. The random variables representing connection volumes are thus i.i.d. and also independent of Π. We let $\nu_\ell = c_\ell b^{-1}$ denote the capacity of link ℓ expressed in connections/sec, and let $\nu = (\nu_\ell, \ell \in \mathcal{L})$.

The "state" of the network is denoted by $n = (n_r, r \in \mathcal{R})$ where n_r is the number of connections currently on route r. We assume that the bandwidth allocated to each ongoing connection depends only on the current state n of the system. Let $\mu_r(n)$ denote the total bandwidth allocated to connections on route r when the system state is n, expressed as a service rate in connections/sec. The choice of the functions $\mu = (\mu_r : \mathbb{Z}_+^{\mathcal{R}} \to \mathbb{R}_+, r \in \mathcal{R})$ will be described in the sequel. If the state of the system changes during the sojourn of a connection (*e.g.,* due to the establishment of a new connection or the termination of an existing one), then, there may be a corresponding change (speed-up or slow-down) in its

[2] Our model is at the connection level, so we can assume undirected routes without loss of generality.

service rate. Indeed since no arriving connections are blocked, new connections must be accommodated by changing the bandwidth allocation, whereas bandwidth made available by departing connections is reallocated to the remaining ones. We assume that ongoing connections are *greedy* in the sense that they will use whatever network bandwidth is made available to them. Note that in reality a given connection may have a limit on the rate at which it can transmit, *e.g.,* may be limited by the access network or network interface card. Herein we shall assume that such bottlenecks have been explicitly modeled by incorporating limited capacity access links in the network.

Let $\Pi_r(t)$ denote the number of connections arriving on route r on the time interval $(0,t]$. This is a rate-λ_r Poisson counting process. Let $\Phi_r(t)$ be another independent unit rate Poisson process. Letting $\{N_r(t), t \geq 0\}$ be the random process corresponding to the number of connections on route r, we have

$$N_r(t) = N_r(0) + \Pi_r(t) - \Phi_r\left(\int_0^t \mu_r(N(s))\, ds\right),\ r \in \mathcal{R},\ t \geq 0, \tag{3.1}$$

which captures the state dependent service rates along each route in the network. It should be clear that given an initial state $N_r(0)$, this evolution equation has a unique solution. Moreover, if the initial condition $(N_r(0), r \in \mathcal{R})$ is selected independently of the arrivals and service processes then the $\mathbb{Z}_+^{\mathcal{R}}$–valued process $N(t) = (N_r(t), r \in \mathcal{R})$ is Markovian.

In the sequel, we describe various bandwidth allocation schemes, or, equivalently, various possible functions μ. In particular we will use μ^m, μ^w and μ^p to denote the max-min, weighted max-min and proportionally fair bandwidth allocation functions. Notice that these functions, of the state n, depend on the capacity vector ν, the routing matrix A, and the type of rate control used on the network. By contrast with standard queuing models, which track packets and queues throughout the network, it is through this dependence that the evolution (3.1) models the dynamics of the network. Also note that we have assumed that connections are not rerouted once they are initiated. One could in principle account for rerouting or splitting of flows across the network but this will not be considered here. Finally, and to avoid possible confusion, bandwidth will be measured in units of connections/sec rather than bits/sec – see above discussion.

3.2.1 Max-min Fair Bandwidth Allocation

We first consider max-min fair bandwidth allocation. An allocation is said to be max-min fair if the bandwidth allocated to a connection cannot be increased without also decreasing that of a connection having a less than or equal allocation [8]. For a single link network this translates to giving each connection traversing the link the same amount of bandwidth. In general one first determines what would be the maximum minimum bandwidth one could assign to any connection in the network and allocates it to the most poorly treated connections. One then removes these connections and the allocated bandwidths from the network, and iteratively repeats the process of maximizing the minimum bandwidth allocation for the remaining connections. More formally the max-min fair allocation can be defined in terms of a hierarchy of optimization problems, described in detail in [23], which is easily solved via the above procedure. Below we briefly review how given the state n of the network one determines the max-min fair bandwidth allocations per connection and in turn determines the bandwidth allocations $(\mu_r^m(n), r \in \mathcal{R})$ per route.

Let the vector $a^* = (a_r^*, r \in \mathcal{R})$ be the max-min fair allocation where a_r^* denotes the bandwidth, in connections/sec, allocated to a single connection on route r. Notice that we have suppressed the dependence of a^* on n. All connections on the same route get the same allocation so $\mu_r^m(n) = n_r a_r^*$. We determine a^* as follows. First for all routes $r \in \mathcal{R}$ such that $n_r = 0$ we set $a_r^* = 0$ and thus $\mu_r^m(n) = 0$. Next we solve a hierarchy of optimization problems starting with

$$f^{(1)}(n) := \max_a \{\min_{r\in\mathcal{R}} a_r : \sum_{r\in\mathcal{R}} A_{\ell r} n_r a_r \le \nu_\ell,\ \ell \in \mathcal{L}\}, \tag{3.2}$$

which corresponds to maximizing the minimum bandwidth per connection subject to the link capacity constraints. It can be shown, see [23], that the solution to this problem is given by

$$f^{(1)}(n) = \min_{\ell\in\mathcal{L}} f_\ell^{(1)}(n) \quad \text{with} \quad f_\ell^{(1)}(n) := \frac{\nu_\ell}{\sum_{r\in\mathcal{R}} A_{\ell r} n_r}, \tag{3.3}$$

where $f_\ell^{(1)}(n)$ can be thought of as the *fair share* at link ℓ, *i.e.*, the bandwidth per connection at link ℓ if its capacity were equally divided among the connections traversing the link.

Let $\mathcal{L}^{(1)}$ be the set of links ℓ such that $f_\ell^{(1)}(n) = f^{(1)}(n)$. This is the set of first-level *bottleneck links*. The set of first-level *bottleneck routes* $\mathcal{R}^{(1)}$ is the set of routes traversing a link in $\mathcal{L}^{(1)}$. These two sets make up the first-level of the *bottleneck hierarchy*. Finally, for each route $r \in \mathcal{R}^{(1)}$, let $a_r^* = f^{(1)}(n)$. The remaining, if any, components of a^* are determined by repeating this process on a reduced network as explained next.

In its second step, if it arises, the algorithm replaces the sets $\mathcal{L}$ and $\mathcal{R}$ by $\mathcal{L} \setminus \mathcal{L}^{(1)}$ and $\mathcal{R} \setminus \mathcal{R}^{(1)}$, respectively. The new state of the system is simply the projection $(n_r, r \in \mathcal{R} \setminus \mathcal{R}^{(1)})$, and a new link capacity vector, $\nu^{(1)}$ is defined on $\mathcal{L} \setminus \mathcal{L}^{(1)}$, where ν_ℓ is reduced to

$$\nu_\ell^{(1)} = \nu_\ell - \sum_{r \in \mathcal{R}^{(1)}} A_{\ell r} \mu_r^m(n) = \nu_\ell - f^{(1)}(n) \sum_{r \in \mathcal{R}^{(1)}} A_{\ell r} n_r.$$

From (3.2) and the definition of $\mathcal{L}^{(1)}$, it is clear that the reduced capacities are non-negative. A new problem paralleling (3.2) but on the reduced network (with reduced sets or routes and links, reduced state, and reduced capacities–as described above) is then defined and solved to obtain a new value $f^{(2)}(n)$, and second-level bottleneck sets $\mathcal{L}^{(2)}$ and $\mathcal{R}^{(2)}$. Finally for $r \in \mathcal{R}^{(2)}$ we set $a_r^* = f^{(2)}(n)$. If necessary this process is once again repeated, but, since the sets $\mathcal{R}^{(1)}, \mathcal{R}^{(2)}, \ldots$ are nonempty, it terminates in a finite number of steps, uniquely specifying the vector a^* and thus $\mu^m(n)$.

Notice that in the above procedure n need not be integer valued, hence $\mu^m(n)$ can be easily extended for real-valued arguments. We shall use the same notation to denote the extension of μ^m from $\mathbb{Z}_+^{\mathcal{R}}$ to $\mathbb{R}_+^{\mathcal{R}}$. Some straightforward properties of this function are summarized below.

Proposition 3.2.1 *The function $\mu^m : \mathbb{R}_+^{\mathcal{R}} \to \mathbb{R}_+^{\mathcal{R}}$ is radially homogeneous, in the sense that*

$$\mu^m(\alpha x) = \mu^m(x), \;\; x \in \mathbb{R}_+^{\mathcal{R}}, \;\; \alpha > 0.$$

In the interior of the positive orthant $\mathbb{R}_+^{\mathcal{R}}$, the function μ^m is continuous, and has strictly positive components. Finally, μ^m is bounded.

The proof of this proposition can be shown by induction on the bottleneck hierarchy and considering the dependence on x of the max-min fair bandwidth allocation.

Notice that the bandwidth allocation policy reflected in μ^m satisfies the link capacity constraints, is fair in the max-min fair sense, but the performance, *e.g.*, in terms of connection delays, may be poor. In the next section we discuss the weighted max-min fair bandwidth allocation which allows some latitude in controlling performance by giving different priorities to connections based on their routes.

3.2.2 Weighted Max-min Fair Bandwidth Allocation

Let $w = (w_r, r \in \mathcal{R})$ be positive "weights" associated with each route in the network, and $a^{w*} = (a_r^{w*}, r \in \mathcal{R})$ denote the weighted max-min fair bandwidth allocation vector. For a given state n we determine a^{w*} in a similar fashion to the max-min fair allocation. First for all routes $r \in \mathcal{R}$ such that $n_r = 0$ set $a_r^{w*} = 0$. Next, replace (3.2) with

$$f^{(1),w}(n) := \max_a \{\min_{r\in\mathcal{R}} \{a_r/w_r\} : \sum_{r\in\mathcal{R}} A_{\ell r} n_r a_r \leq \nu_\ell,\ \ell \in \mathcal{L}\},$$

which can again be solved by first defining the *weighted fair share* on link ℓ as

$$f_\ell^{(1),w}(n) := \frac{\nu_\ell}{\sum_{r\in\mathcal{R}} A_{\ell r} w_r n_r} \tag{3.4}$$

and then setting $f^{(1),w}(n) = \min_{\ell\in\mathcal{L}} f_\ell^{(1),w}(n)$. Paralleling the max-min fair case, the first-level bottleneck links and routes, denoted $\mathcal{L}^{(1),w}$ and $\mathcal{R}^{(1),w}$ respectively, can be defined, and one can proceed iteratively to determine the bandwidth allocation for connections on all routes. We will let $\mu^w(n)$ denote the vector of bandwidths allocated to each route where $\mu_r^w(n) = w_r n_r a_r^{w*}$, and let $\mu^w = (\mu_r^w : \mathbb{Z}_+^{\mathcal{R}} \to \mathbb{R}_+, r \in \mathcal{R})$.

One can again extend μ^w for real-valued arguments *i.e.*, from $\mathbb{Z}_+^{\mathcal{R}}$ to $\mathbb{R}_+^{\mathcal{R}}$, and show that

$$\mu^w(x) = \mu^m(Dx), \tag{3.5}$$

where μ^m corresponds to the unweighted max-min fair allocation discussed in the previous section, and $D = \text{diag}(w)$, *i.e.*, a square matrix with components $(w_r, r \in \mathcal{R})$ along its diagonal. Thus one way to view the weighted max-min fair allocation is as a max-min fair

allocation where the "effective number" of ongoing connections is Dx. Moreover one can easily see that the results in Proposition 3.2.1 also apply to μ^w.

A weighted max-min fair allocation can be used to differentiate among connections following different routes and thus give priority based on geographic, administrative, or service requirements by grouping like connections on a route. However specific criteria for the selection of weights need to be developed. In principle one can consider control policies which adjust the weights based on the state of the network – a simple example is briefly considered in §3.3.4

3.2.3 Proportionally Fair Bandwidth Allocation

As a final alternative we consider a framework where utility functions $U_r : \mathbb{R}_+ \to \mathbb{R}, r \in \mathcal{R}$ have been associated with connections following various routes. Here $U_r(a_r)$ is the utility to a user/connection on route r of a bandwidth allocation a_r.[3] A bandwidth allocation policy which maximizes the total network utility when the state is n can be obtained by solving the following optimization problem:

$$\max_a \left\{ \sum_{r\in\mathcal{R}} n_r U_r(a_r) : \sum_{r\in\mathcal{R}} A_{\ell r} n_r a_r \le \nu_\ell, \ell \in \mathcal{L}; a \ge 0 \right\}, \tag{3.6}$$

where we assume that the utility functions are concave and so the optimizer is unique. This approach to allocating bandwidth is pleasing in the sense that it finds an appropriate compromise between the extent to which users value bandwidth and the *overall* user "satisfaction."

In general it is unclear how to select utility functions. However, [30] and others, have considered the case where $U_r(a_r) = \log a_r$ and shown that in this case the maximizer $a^{p*} = (a_r^{p*}, r \in \mathcal{R})$ corresponds to a *proportionally fair* bandwidth allocation in the sense that the vector is feasible, *i.e.,* satisfies the link capacity constraints, and for any other

[3] If there exist connections with different utility functions that follow the same path, one can define several routes carrying connections that share the same utility function.

feasible rate $a' = (a'_r, r \in \mathcal{R})$, the aggregate proportional change is negative, *i.e.*,

$$\sum_{r \in \mathcal{R}} n_r \frac{a'_r - a_r^{p*}}{a_r^{p*}} < 0. \tag{3.7}$$

Determining the maximizer of (3.6) for log utility functions can be done explicitly for simple networks. Alternatively, as with max-min fairness, one can design rate control mechanisms that converge to the associated bandwidth allocation [33]. We will let $\mu_r^p(n) = n_r a_r^{p*}$ denote the total bandwidth allocated to connections along route $r \in \mathcal{R}$ and $\mu^p(n) = (\mu_r^p(n), r \in \mathcal{R})$ be the bandwidth allocations per route when proportional fairness is used. Again μ^p can be easily extended for real-valued arguments. We shall use the same notation to denote the extension of μ^p from $\mathbb{Z}_+^{\mathcal{R}}$ to $\mathbb{R}_+^{\mathcal{R}}$.

Proposition 3.2.2 *The function $\mu^p : \mathbb{R}_+^{\mathcal{R}} \to \mathbb{R}_+^{\mathcal{R}}$ is radially homogeneous, in the sense that*

$$\mu^p(\alpha x) = \mu^p(x), \;\; x \in \mathbb{R}_+^{\mathcal{R}}, \;\; \alpha > 0. \tag{3.8}$$

In the interior of the positive orthant $\mathbb{R}_+^{\mathcal{R}}$, the function μ^p is continuous, and has strictly positive components. Finally, μ^p is bounded.

The continuity of μ^p follows by considering the functional dependence on x of the proportionally fair bandwidth allocation, while radial homogeneity is easily shown by a change of variables $b_r = x_r a_r$. The problem is then

$$\mu^p(x) = \text{argmax}_b \left\{ \sum_{r \in \mathcal{R}} x_r \log(b_r) : Ab \leq \nu; b \geq 0 \right\}, \tag{3.9}$$

where we note that b_r now corresponds to the bandwidth allocated on route r, and thus the maximizing vector corresponds to proportionally fair bandwidth allocation $\mu^p(x)$.

3.3 Stability of the Stochastic Network

In this section we will consider the stability of the stochastic network model defined in §3.2, for various types of bandwidth allocation. Assuming $\{\Pi_r, \Phi_r, \; r \in \mathcal{R}\}$ are independent

Poisson processes on $[0,\infty)$, where Π_r has rate λ_r and Φ_r has rate 1, the evolution equation (3.1) defines a Markov chain on $\mathbb{Z}_+^{\mathcal{R}}$ with transition rates

$$q(n,m) = \begin{cases} \lambda_r, & m = n + e^r,\ r \in \mathcal{R} \\ \mu_r(n), & m = n - e^r,\ r \in \mathcal{R} \\ 0, & \text{otherwise} \end{cases}, \tag{3.10}$$

for $m \neq n$, where $e^r = (e^r_s, s \in \mathcal{R})$, $e^r_s = 1(r = s)$. Thus, when the state is n, route r sees arrivals with rate λ_r and departures with rate $\mu_r(n)$. Note that when $n_r = 0$ we have $\mu_r(n) = 0$, thus $q(n, n - e^r) = 0$, and so the rates are supported on the positive orthant.

We use the notation Q for the infinitesimal generator (viz., rate matrix) of this continuous-time Markov chain. For a function $\varphi : \mathbb{R}_+^{\mathcal{R}} \to \mathbb{R}$, we write[4]

$$Q\varphi(n) := \sum_{m \in \mathbb{Z}_+^{\mathcal{R}}} q(n,m)\varphi(m) = \sum_{m \in \mathbb{Z}_+^{\mathcal{R}}} q(n,m)[\varphi(m) - \varphi(n)], \tag{3.11}$$

where the latter equality follows from the fact that Q is conservative:

$$q(n,n) = -\sum_{m \neq n} q(n,m).$$

Note that $Q\varphi(n)$ can be interpreted as the expected drift, *i.e.*, the change in $\varphi(N(t))$ when $N(t) = n$.

Clearly the Markov chain $\{N(t),\ t \geq 0\}$ is irreducible, and we say that it is stable, iff it is positive recurrent. We will show positive recurrence by constructing a Lyapunov function [44, 19]. For our system, a Lyapunov function is any function $V : \mathbb{Z}_+^{\mathcal{R}} \to \mathbb{R}_+$ with the sole property that there exists a finite set $K \subseteq \mathbb{Z}_+^{\mathcal{R}}$, such that

$$\sup_{n \notin K} QV(n) < 0, \tag{3.12}$$

where QV is defined as in (3.11). Using our formula (3.10) for the transition rates we can rewrite QV as

$$QV(n) = \sum_{r \in \mathcal{R}} \{\lambda_r[V(n + e^r) - V(n)] + \mu_r(n)[V(n - e^r) - V(n)]\}. \tag{3.13}$$

[4]Notice that the sums in (3.11) have a finite number of terms, since the chain has only local transitions, *i.e.*, arrivals and departures for every route, thus there are no restrictions on the function φ for $Q\varphi(n)$ to be well defined.

Intuitively (3.12) means that when the process $N(t)$ lies outside K, it is such that on average $V(N(t))$ is decreasing, *i.e.*, has negative drift.

Searching for such a Lyapunov function can be a tedious procedure, particularly since the transition rates of our Markov chain are defined via the optimization problem associated with the various fairness criteria.

3.3.1 Stability under Max-min Fair Bandwidth Allocation

We first consider the stability of the network when bandwidth is allocated according to the max-min fair criterion and thus the dynamics of the system are captured by (3.1) with μ replaced by μ^m as defined in §3.2.1.

We will begin by considering a *candidate* Lyapunov function, related to the max-min fairness criterion. Let $V(n)$ be the reciprocal of $f^{(1)}(n)$ defined in (3.2) and extend it from $\mathbb{Z}_+^{\mathcal{R}}$ to $\mathbb{R}_+^{\mathcal{R}}$, namely,

$$V(x) = \max_{\ell \in \mathcal{L}} \{\nu_\ell^{-1} \sum_{r \in \mathcal{R}} A_{\ell r} x_r\}, \;\; x \in \mathbb{R}_+^{\mathcal{R}}.$$

For convenience we introduce the vectors

$$\xi^\ell = (\xi_r^\ell, \; r \in \mathcal{R}), \quad \xi_r^\ell := \nu_\ell^{-1} A_{\ell r}, \;\; \ell \in \mathcal{L}. \tag{3.14}$$

and let $\varphi^\ell(x) = \langle \xi^\ell, x \rangle, \ell \in \mathcal{L}$ where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^{\mathcal{R}}$. With this notation we have that

$$V(x) = \max_{\ell \in \mathcal{L}} \; \varphi^\ell(x) = \max_{\ell \in \mathcal{L}} \; \langle \xi^\ell, x \rangle. \tag{3.15}$$

Thus V is a piecewise linear function. Since the vectors ξ^ℓ have non-negative components, the sets $\{x \in \mathbb{R}_+^{\mathcal{R}} : \; V(x) \leq \alpha\}$ are compact polytopes, for all $\alpha \geq 0$. For a fixed x, one or more of the indices ℓ achieve the maximum in (3.15)– these are the first-level bottleneck links defined earlier. We will use $\mathcal{L}^{(1)}(x)$ to denote the dependence of the first-level bottleneck links on x. Similarly $\mathcal{R}^{(1)}(x)$ and $\nu^{(1)}(x)$ will be used to indicate such dependencies in the sequel.

Since for first level bottleneck links the link capacity is fully utilized among ongoing connections, we would expect that, on average, the number of connections on such a link $\sum_{r\in\mathcal{R}} A_{\ell r} n_r$ will decrease as long as the average arrival rate does not exceed the link capacity. The following lemma makes this clear.

Lemma 3.3.1 *Assume that* $A\lambda < \nu$, i.e., $\sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r < \nu_\ell,$ *for all* $\ell \in \mathcal{L}$.[5] *Then, there is a constant* $c > 0$, *such that for all* $x \in \mathbb{R}_+^{\mathcal{R}}$, *and all* $\ell^* \in \text{argmax}_{\ell\in\mathcal{L}}\varphi^\ell(x)$, i.e., *first-level bottleneck links* $\ell^* \in \mathcal{L}^{(1)}(x)$, *we have*

$$Q\varphi^{\ell^*}(x) = \langle \xi^{\ell^*}, \lambda - \mu^m(x)\rangle \le -c. \tag{3.16}$$

Proof: First, using (3.13) and the definition (3.14) of ξ^ℓ we have that

$$Q\varphi^{\ell^*}(x) = \sum_{r\in\mathcal{R}} \xi_r^{\ell^*}(\lambda_r - \mu_r^m(x)) = \langle \xi^{\ell^*}, \lambda - \mu^m(x)\rangle = \sum_{r\in\mathcal{R}} \nu_{\ell^*}^{-1} A_{\ell^* r}(\lambda_r - \mu_r^m(x)).$$

Next, since ℓ^* is a first-level bottleneck link, it follows that for routes r traversing link ℓ^* we have $\mu_r^m(x) = x_r a_r^*$ where a_r^* is given by (3.2). Thus,

$$\begin{aligned} Q\varphi^{\ell^*}(x) &= \nu_{\ell^*}^{-1}\left(\sum_{r\in\mathcal{R}} A_{\ell^* r}\lambda_r - \sum_{r\in\mathcal{R}} A_{\ell^* r}\frac{\nu_{\ell^*} x_r}{\sum_{s\in\mathcal{R}} A_{\ell^* s} x_s}\right) \\ &= \nu_{\ell^*}^{-1}\left(\sum_{r\in\mathcal{R}} A_{\ell^* r}\lambda_r - \nu_{\ell^*}\right) \\ &\le -c, \end{aligned}$$

where $c := \max_{\ell\in\mathcal{L}}\{\nu_\ell^{-1}(\nu_\ell - \sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r)\}$ is positive by the stability condition. ■

Despite the promise of Lemma 3.3.1 it is unclear whether V is an appropriate Lyapunov function. Indeed the lemma only suggests that as long as the state makes transitions on regions having the *same* first level bottleneck links, $V(N(t))$ will experience a negative drift. To make this more precise we will explicitly identify these regions and for clarity present an example in §3.3.3. Let $\mathcal{M}$ be a nonempty subset of $\mathcal{L}$ and let

$$C_{\mathcal{M}} = \{x \in \mathbb{R}_+^{\mathcal{R}} :\ \mathcal{L}^{(1)}(x) = \mathcal{M}\}. \tag{3.17}$$

[5] *In the sequel this will be referred to as the stability condition.*

It is clear that if $\alpha > 0, x \in C_{\mathcal{M}} \Rightarrow \alpha x \in C_{\mathcal{M}}$, *i.e.*, these sets are cones, and that

$$\bigcup_{\mathcal{M} \subseteq \mathcal{L}, \mathcal{M} \neq \emptyset} C_{\mathcal{M}} = \mathbb{R}_+^{\mathcal{R}}.$$

Suppose that $n \in C_{\mathcal{M}}$, for some nonempty $\mathcal{M}$, then the drift $QV(n)$ can easily be computed (see (3.13)), provided $n + e^r, n - e^r \in C_{\mathcal{M}}$, for all $r \in \mathcal{R}$. In this case, with ℓ any element of $\mathcal{M}$, we have $QV(n) = \langle \xi^\ell, \lambda - \mu^m(n) \rangle \leq -c$, by Lemma 3.3.1. However when n and $n + e^r$ or $n - e^r$ belong to different cones an explicit verification of the negative drift requirement becomes difficult. Indeed when this is the case a transition causes a change in the bottleneck links – alternatively we are "crossing of a boundary" of one of the cones. Intuitively we may argue that this effect is negligible, since it occurs at a relatively small fraction of points in the state space.

To make this intuition into a rigorous statement observe that Lemma 3.3.1 also implies that there is a $c > 0$, such that

$$\langle \nabla V(x), \lambda - \mu^m(x) \rangle \leq -c, \tag{3.18}$$

for all x at which the gradient $\nabla V(x) := (\partial V(x)/\partial x_r,\ r \in \mathcal{R})$ exists. It is easy to see that this gradient exists almost everywhere, and, when it exists, it equals ξ^ℓ, for some ℓ. We will start by showing that there exists a smoothened version W of the function V that satisfies a drift condition in the sense of (3.18) for all $x \in \mathbb{R}_+^{\mathcal{R}}$.

Lemma 3.3.2 ([18]) *If $A\lambda < \nu$, then there is a non-negative function W, defined on $\mathbb{R}_+^{\mathcal{R}} \setminus \{0\}$, that is at least twice-continuously differentiable, has a Hessian*[6]*, $\nabla^2 W(x)$, such that $\nabla^2 W(x) \to 0$, as $|x| \to \infty$, and which satisfies the following drift condition: there is a $d > 0$, such that*

$$\langle \nabla W(x), \lambda - \mu^m(x) \rangle \leq -d$$

for all $x \neq 0$.

For completeness we have included a proof of the lemma to the appendix. Next we show that the network is indeed positive recurrent.

[6] *Here $\nabla^2 W(x)$ denotes the $|\mathcal{R}| \times |\mathcal{R}|$ matrix with entries $\{\frac{\partial^2 W}{\partial x_r \partial x_s}(x), r, s \in \mathcal{R}\}$.*

Theorem 3.3.1 *If $A\lambda < \nu$ then the Markov chain $\{N(t),\ t \geq 0\}$ associated with the max-min fair bandwidth allocation is positive recurrent.*

Proof: Since W is twice differentiable it follows by the Mean Value Theorem that for $n, m \in \mathbb{Z}_+^{\mathcal{R}}$ there exists a $\theta, 0 \leq \theta \leq 1$ such that

$$W(n+m) - W(n) = \langle \nabla W(n), m \rangle + \frac{1}{2} m^T \nabla^2 W(n+\theta m) m := \langle \nabla W(n), m \rangle + \beta(n, m).$$

Recall that $\nabla^2 W(n) \to 0$ and thus $\beta(n, z) \to 0$ as $|n| \to \infty$. Now, using this approximation to compute QW, as in (3.13), we have

$$QW(n) = \langle \nabla W(n), \lambda - \mu^m(n) \rangle + \sum_m q(n, m)\beta(n, m - n).$$

It follows by Lemma 3.3.2 that the first term is at most $-d$. The second term, is a sum of a finite number of terms, and can be made smaller than $d/2$ for all $|n|$ sufficiently large. Thus noting that $\sup_{|n|>\gamma} QW(n) < 0$, for sufficiently large γ, and letting $K = \{n : |n| \leq \gamma\}$ we satisfy the drift condition (3.12) which as discussed earlier implies positive recurrence. ■

3.3.2 Stability under Weighted Max-min Fair Bandwidth Allocation

While the previous result is intuitive, in that the number of connections on bottleneck links must be decreasing, it is not easily extended it to show the stability of networks under weighted max-min fair bandwidth allocation. Thus, we develop an alternative approach which, instead of focusing links, focuses on the relative states of each route. Suppose that a set of weights w is selected and the network is operated subject to the bandwidth allocation function μ^w defined in §3.2.2. We will let $\varphi^r(x) = \lambda_r^{-1} w_r x_r, r \in \mathcal{R}$ and consider the candidate function

$$V(x) = \max_{r \in \mathcal{R}} \varphi^r(x) = \max_{r \in \mathcal{R}} \{\lambda_r^{-1} w_r x_r\}. \tag{3.19}$$

The following lemma shows why this particular function is useful.

Lemma 3.3.3 *Assume that $A\lambda < \nu$ then there is a constant $c > 0$, such that for all $x \in \mathbb{R}_+^{\mathcal{R}}$ and for all $r^* \in \text{argmax}_{r\in\mathcal{R}}\varphi^r(x)$ we have*

$$Q\varphi^{r^*}(x) = \lambda_{r^*}^{-1} w_{r^*}(\lambda_{r^*} - \mu_{r^*}^w(x)) \leq -c.$$

Proof: Suppose the the network state is x and let $r^* \in \text{argmax}_{r\in\mathcal{R}}\{\lambda_r^{-1} w_r x_r\}$ then for all $r \in \mathcal{R}$, we have that

$$\frac{w_r x_r}{\lambda_r} \leq \frac{w_{r^*} x_{r^*}}{\lambda_{r^*}}, \tag{3.20}$$

or equivalently that $\lambda_{r^*} w_r x_r \leq \lambda_r w_{r^*} x_{r^*}$. Now summing over all routes traversing a link $\ell \in \mathcal{L}$ we have that

$$\lambda_{r^*} \sum_{r\in\mathcal{R}} A_{\ell r} w_r x_r \leq w_{r^*} x_{r^*} \sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r,$$

which one can rearrange to show that

$$\begin{aligned}
\lambda_{r^*} &\leq \frac{w_{r^*} x_{r^*}}{\sum_{r\in\mathcal{R}} A_{\ell r} w_r x_r} \sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r \\
&= \frac{w_{r^*} x_{r^*}}{\sum_{r\in\mathcal{R}} A_{\ell r} w_r x_r}\nu_\ell - \frac{w_{r^*} x_{r^*}}{\sum_{r\in\mathcal{R}} A_{\ell r} w_r x_r}(\nu_\ell - \sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r).
\end{aligned}$$

Given (3.20) and the stability condition one can easily show the existence of a positive lower bound, $\varepsilon > 0$, for the term on the right-hand side :

$$\frac{w_{r^*} x_{r^*}}{\sum_{r\in\mathcal{R}} A_{\ell r} w_r x_r}(\nu_\ell - \sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r) \geq \min_{\ell\in\mathcal{L}} \min_{r\in\mathcal{R}} \left\{ \frac{A_{\ell r}\lambda_r}{\sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r}(\nu_\ell - \sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r) \right\} = \varepsilon.$$

Thus we have that

$$\lambda_{r^*} \leq w_{r^*} x_{r^*} f_\ell^{(1),w}(x) - \varepsilon$$

where we recognize a term corresponding to the fair share $f_\ell^{(1),w}(x)$ at link ℓ, see (3.4). Moreover since this is true for all ℓ we have that

$$\lambda_{r^*} \leq w_{r^*} x_{r^*} f^{(1),w}(x) - \varepsilon \tag{3.21}$$

where $f^{(1),w}(x) = \min_{\ell\in\mathcal{L}} f_\ell^{(1),w}(x)$ is the fair share at first level bottleneck links $\mathcal{L}^{(1),w}(x)$.

Now if r^* is a first level bottleneck route, *i.e.*, $r^* \in \mathcal{R}^{(1),w}(x)$, then $\mu_{r^*}^w(x) = w_{r^*} x_{r^*} f^{(1),w}(x)$, and it follows by (3.21) that $\lambda_{r^*} - \mu_{r^*}^w(x) \leq -\varepsilon$. If r^* is not a first level

bottleneck route, we will show that its bandwidth allocation must exceed $w_{r^*}x_{r^*}f^{(1),w}(x)$ and so again by (3.21) we have that $\lambda_{r^*} - \mu^w_{r^*}(x) \leq -\varepsilon$.

We begin by showing that $f^{(2),w}(x) \geq f^{(1),w}(x)$. Suppose $\ell \in \mathcal{L} \setminus \mathcal{L}^{(1),w}(x)$ and note that

$$\sum_{r\in\mathcal{R}} A_{\ell r}w_r x_r f^{(1),w}(x) \leq \sum_{r\in\mathcal{R}} A_{\ell r}w_r x_r f^{(1),w}_\ell(x) = \nu_\ell$$

so it follows that

$$\nu_\ell - f^{(1),w}(x) \sum_{r\in\mathcal{R}^{(1),w}(x)} A_{\ell r}w_r x_r \geq f^{(1),w}(x) \sum_{r\in\mathcal{R}\setminus\mathcal{R}^{(1),w}(x)} A_{\ell r}w_r x_r.$$

Rearranging terms and recalling the definition of fair share for the links in the second level of the bottleneck hierarchy we have that

$$\begin{aligned} f^{(2),w}_\ell(x) &= \frac{\nu^{(1),w}_\ell(x)}{\sum_{r\in\mathcal{R}\setminus\mathcal{R}^{(1),w}(x)} A_{\ell r}w_r x_r} \\ &= \frac{\nu_\ell - f^{(1),w}(x)\sum_{r\in\mathcal{R}^{(1),w}(x)} A_{\ell r}w_r x_r}{\sum_{r\in\mathcal{R}\setminus\mathcal{R}^{(1),w}(x)} A_{\ell r}w_r x_r} \\ &\geq f^{(1),w}(x). \end{aligned}$$

Thus $f^{(2),w}(x) = \min_{\ell\in\mathcal{L}} f^{(2),w}_\ell(x) \geq f^{(1),w}(x)$. Similarly it follows by induction that $f^{(i+1),w}(x) \geq f^{(i),w}(x)$, until the bottleneck hierarchy is exhausted.

Now since $\mu^w_{r^*}(x) = w_{r^*}x_{r^*}f^{(j),w}(x)$ for some level j in the bottleneck hierarchy, it follows that $\mu^w_{r^*}(x) \geq w_{r^*}x_{r^*}f^{(1),w}(x)$ and so $\lambda_{r^*} - \mu^w_{r^*}(x) \leq -\varepsilon$. The lemma follows by selecting $c = \varepsilon \min_{r\in\mathcal{R}}\{\lambda_r^{-1}w_r\}$. ■

Theorem 3.3.2 *If $A\lambda < \nu$ then Markov chain $\{N(t),\ t \geq 0\}$ associated with weighted max-min fair bandwidth allocation is positive recurrent.*

Proof: Based on Lemma 3.3.3, and the technique used in Lemma 3.3.2, it should be clear that an appropriately smooth Lyapunov function W can be constructed from V in (3.19). Positive recurrence then follows as in Theorem 3.3.1. ■

Note that since max-min fairness is a special case of weighted max-min fairness, Theorem 3.3.2 establishes the stability of both. The two different Lyapunov functions we

have introduced, based on links and routes, may be of interest in further studies of performance. These results establish that $A\lambda < \nu$ is a *sufficient* condition for stability. In fact, it is a *necessary* condition. Say there exists a link ℓ such that $\sum_{r\in\mathcal{R}} A_{\ell r}\lambda_r > \nu_\ell$. Clearly such a link in isolation is unstable, *i.e.*, on average will tend to drift off to infinity. When the link is incorporated within a network, the situation can in fact only get worse, since other links may slow down the departures for connections on ℓ.

3.3.3 Example Network

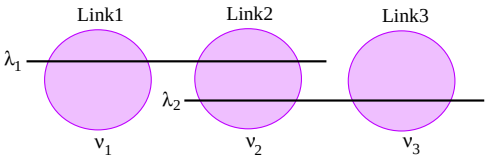

Figure 3.1: Example network with three links and two routes.

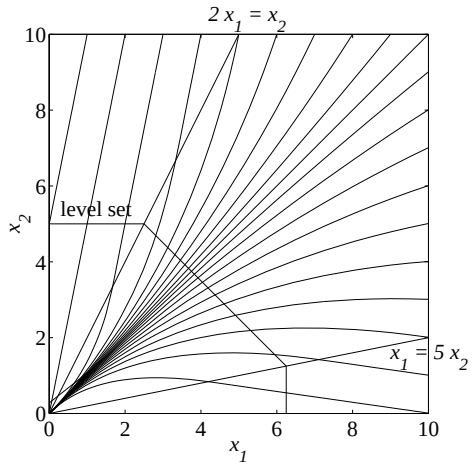

Figure 3.2: A vector field of the example network.

In this example we consider max-min fair bandwidth allocation for the network shown in Fig. 3.1 – it consists of two routes $\mathcal{R} = \{1, 2\}$, three links, $\mathcal{L} = \{1, 2, 3\}$, and routing matrix

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Based on the notion of fair share (3.3), we can define the first and second level bottleneck link and route sets, for any $x \in \mathbb{R}^3_+$. Notice that in this example there are at most 2 levels in the bottleneck hierarchy. The various cases, and corresponding cones $C_{\mathcal{M}}$ (see (3.17)), are

defined below.

- Case 1 (Link 1 is the unique 1st level bottleneck link):

$$\begin{aligned}\mathcal{L}^{(1)}(x) &= \{1\} \text{ if } f_1^{(1)}(x) < f_2^{(1)}(x) \text{ and } f_1^{(1)}(x) < f_3^{(1)}(x) \qquad (3.22)\\ \mathcal{L}^{(2)}(x) &= \{2\} \text{ if } f_2^{(2)}(x) < f_3^{(2)}(x)\\ \mathcal{L}^{(2)}(x) &= \{3\} \text{ if } f_3^{(2)}(x) < f_2^{(2)}(x)\end{aligned}$$

$$C_{\{1\}} = \{x \in \mathbb{R}^3_+ : \ \mathcal{L}^{(1)}(x) = \{1\}\}.$$

- Case 2 (Link 2 is the unique 1st level bottleneck link):

$$\mathcal{L}^{(1)}(x) = \{2\} \text{ if } f_2^{(1)}(x) < f_1^{(1)}(x) \text{ and } f_2^{(1)}(x) < f_3^{(1)}(x) \qquad (3.23)$$

$$C_{\{2\}} = \{x \in \mathbb{R}^3_+ : \ \mathcal{L}^{(1)}(x) = \{2\}\}.$$

- Case 3 (Link 3 is the unique 1st level bottleneck link):

$$\begin{aligned}\mathcal{L}^{(1)}(x) &= \{3\} \text{ if } f_3^{(1)}(x) < f_1^{(1)}(x) \text{ and } f_3^{(1)}(x) < f_2^{(1)}(x) \qquad (3.24)\\ \mathcal{L}^{(2)}(x) &= \{1\} \text{ if } f_1^{(2)}(x) < f_2^{(2)}(x)\\ \mathcal{L}^{(2)}(x) &= \{2\} \text{ if } f_2^{(2)}(x) < f_1^{(2)}(x)\end{aligned}$$

$$C_{\{3\}} = \{x \in \mathbb{R}^3_+ : \ \mathcal{L}^{(1)}(x) = \{3\}\}.$$

The sets of bottleneck links $\mathcal{L}^{(1)}$ or $\mathcal{L}^{(2)}$ could have more than one element if the fair shares were the same on two links, *e.g.,* $\mathcal{L}^{(1)} = \{1, 2\}$ if $f_1^{(1)} = f_2^{(1)} < f_3^{(1)}$, and in this case the cones are defined as $C_{\{1,2\}}$.

Next we consider the piecewise linear function $V(x)$, given in (3.15) :

$$\begin{aligned}V(x) &= \max_{\ell \in \mathcal{L}} \varphi^\ell(x) = \max_{\ell \in \mathcal{L}} \langle \xi^\ell, x \rangle\\ &= \max\{\nu_1^{-1} x_1, \nu_2^{-1}(x_1 + x_2), \nu_3^{-1} x_2\}.\end{aligned}$$

and assume the stability condition holds, *i.e.,*

$$\lambda_1 < \nu_1, \qquad \lambda_1 + \lambda_2 < \nu_2, \qquad \lambda_2 < \nu_3.$$

We can compute the drifts in (3.16) explicitly to obtain,

$$\begin{aligned} Q\varphi^1(x) &= \langle \xi^1, \lambda - \mu^m(x) \rangle = \nu_1^{-1}(\lambda_1 - \nu_1) \leq -c, \qquad x \in C_{\{1\}} \\ Q\varphi^2(x) &= \langle \xi^2, \lambda - \mu^m(x) \rangle = \nu_2^{-1}(\lambda_1 + \lambda_2 - \nu_2) \leq -c, \qquad x \in C_{\{2\}} \\ Q\varphi^3(x) &= \langle \xi^3, \lambda - \mu^m(x) \rangle = \nu_3^{-1}(\lambda_2 - \nu_3) \leq -c, \qquad x \in C_{\{3\}}. \end{aligned}$$

The vector field $\lambda - \mu^m(x)$ corresponding to the max-min bandwidth allocation is shown Fig. 3.2 when $\lambda = (1.5, 1.5)$ and $\nu = (5, 6, 4)$. According to the bottleneck condition, we have three cones coincide at boundaries $x_1 = 5x_2$ and $2x_1 = x_2$, which is obtained by solving (3.22) through (3.24). The cones $C_{\{1\}}$, $C_{\{2\}}$ and $C_{\{3\}}$ correspond to lower part, middle one and upper one, respectively. Also shown on the figure is a level set of the function V. From the figure it is clear that on each cone the network's dynamics push inwards, *i.e.,* have negative drift with respect to V. By smoothing V as in Lemma 3.3.2 we obtain a Lyapunov function W from which the stability of the system follows.

3.3.4 Stability under a State Dependent Weighted Max-min Fair Control Policy

In this section we briefly consider a simple extension to our model with weighted max-min fair allocation, wherein a control policy is implemented by letting the weights depend on the network state. Let $w = (w_r : \mathbb{Z}_+^{\mathcal{R}} \to \mathbb{R}_+, r \in \mathcal{R})$ now denote functions where $w(n) = (w_r(n), r \in \mathcal{R})$ are understood to be the weights associated with each route when the network state is n. Assume that when the system is in state n bandwidth is allocated to routes according to a weighted max-min fair allocation with weights $w(n)$. Let $\mu^{w(n)}(n) = (\mu_r^{w(n)}(n), r \in \mathcal{R})$ denote the bandwidths allocated to each route in the network when the state is n. Our interest in this type of model, was motivated by work on stability of Generalized Processor Sharing networks [53]. Without delving into the details of their model, we remind the reader that in such networks a connection is assigned a weight at each node (representing a queue) which determines the fraction of the available capacity it receives at that node. The authors showed the queue/delay stability of non-acyclic networks

of this type when connections received a *consistent relative treatment*. By analogy here, we will say that a state dependent weight based control policy gives routes a *uniform relative treatment* if $\forall n \in \mathbb{Z}_+^{\mathbb{R}}$ and $r, s \in \mathcal{R}$,

$$\frac{\lambda_s}{\lambda_r} \geq \frac{n_s w_s(n)}{n_r w_r(n)}. \tag{3.25}$$

An example one on such control policy would be $w_r(n) = \lambda_r / n_r$ for $n_r \neq 0$. Thus upon admitting or tearing down a connection along a given route the network controller would need to adjust the weight associated with that route. The following lemma shows that subject to the natural stability condition, a weight based control policy that gives routes uniform relative treatment is stable.

Lemma 3.3.4 *Assume $A\lambda < \nu$ and a weight based control policy $w(\cdot)$ that gives routes a uniform relative treatment is used to allocate bandwidth in the network. Then $\forall n \in \mathbb{Z}_+^{\mathcal{R}}$ and $\forall r \in \mathcal{R}$ such that $n_r > 0$ we have*

$$\lambda_r < \mu_r^{w(n)}(n).$$

It follows that the network is positive recurrent.

The proof of this lemma is almost identical to that of Lemma 3.3.3 and is included in the appendix. Positive recurrence follows since the number of connections on every route has negative drift if it is not empty.

3.3.5 Stability under Proportionally Fair Bandwidth Allocation

Unlike (weighted) max-min fair allocation of bandwidth, proportionally fair bandwidth allocation maximizes the overall utility of the network, rather than focusing on maximizing the worst case individual utility/performance. This is reflected in our choice of Lyapunov function. In particular the property that the aggregate proportional change is negative in proportionally fair allocation as in (3.7) will play a role. We propose the following quadratic Lyapunov function:

$$W(x) = \sum_{r \in \mathcal{R}} \frac{x_r^2}{2\lambda_r}.$$

Note that the function is continuous and twice differentiable, thus there is no need for the smoothing process as we used in the case of max-min fair allocation. In Lemma 3.3.5 and Theorem 3.3.3 below we show that this function satisfies the requirements to show the stability of the proportionally fair allocation.

Lemma 3.3.5 *Assume that $A\lambda < \nu$. Then there exists a constant $d > 0$ such that for all $x \in \mathbb{R}_+^{\mathcal{R}} \setminus \{0\}$ the following holds*

$$< \nabla W(x), \lambda - \mu^p(x) > \quad = \quad \sum_{r \in \mathcal{R}} \lambda_r^{-1} x_r (\lambda_r - \mu_r^p(x)) \leq -d.$$

Proof: Note that $\nabla_r W(x) = \lambda_r^{-1} x_r$, so

$$< \nabla W(x), \lambda - \mu^p(x) > \quad = \quad \sum_{r \in \mathcal{R}} \frac{x_r}{\lambda_r} (\lambda_r - \mu_r^p(x)) \leq \sum_{r \in \mathcal{R}: x_r > 0} \frac{x_r}{\mu_r^p(x)} (\lambda_r - \mu_r^p(x)) \quad (3.26)$$

where the inequality follows from:

$$\begin{aligned} \lambda_r &\geq \mu_r^p(x), \qquad \text{if } \lambda_r - \mu_r^p(x) \geq 0, \\ \lambda_r &< \mu_r^p(x), \qquad \text{if } \lambda_r - \mu_r^p(x) < 0. \end{aligned}$$

Note if $\mu_r^p(x) = 0$ then $x_r = 0$, so in (3.26) the inequality still holds. By noting that $\mu_r^p(x) = x_r a_r^p(x)$, we have

$$\begin{aligned} < \nabla W(x), \lambda - \mu^p(x) > \quad &\leq \quad \sum_{r \in \mathcal{R}} \frac{x_r}{\mu_r^p(x)} (\lambda_r - \mu_r^p(x)) \\ &\leq \quad \sum_{r \in \mathcal{R}} x_r \frac{\lambda_r / x_r - a_r^{p*}(x)}{a_r^{p*}(x)} \qquad (3.27) \\ &\leq \quad 0, \qquad (3.28) \end{aligned}$$

since a^p satisfies the negative aggregate proportional change as given in (3.7), and $a_r{}' = \lambda_r / x_r$ is a feasible vector, *i.e.*, $\sum_{r \in \mathcal{R}} A_{\ell r} x_r a_r{}' \leq \nu_\ell$ is a strict inequality because the problem is strictly concave and we can not have $a_r{}' = \lambda_r / x_r = a_r^{p*}(x)$ for all $r \in \mathcal{R}$ in (3.27). Indeed suppose this were true, *i.e.*, $\lambda_r = x_r a_r^{p*}(x) = \mu_r^p(x)$ for all $r \in \mathcal{R}$, then it implies that for a bottleneck link $\ell \in \mathcal{L}$, $\sum_{r \in \mathcal{R}} A_{\ell r} \lambda_r = \sum_{r \in \mathcal{R}} A_{\ell r} \mu_r^p(x) = \nu_\ell$, noting that

there is at least one bottleneck link per route, which in turn contradicts with our stability condition $A\lambda < \nu$.

So for a given x by continuity of $< \nabla W(x), \lambda - \mu^p(x) >$ in x there exists δ and $d(x)$ such that for all y in a ball $||y-x|| < \delta(x)$, we have $< \nabla W(y), \lambda-\mu^p(y) > \leq -d(x)$. This holds uniformly in a compact subset of $\mathbb{R}_+^{\mathcal{R}}$ containing 0, and can be extended to $\mathbb{R}_+^{\mathcal{R}}$ using the following property with $\alpha > 0$

$$< \nabla W(\alpha x), \lambda - \mu^p(\alpha x) > \quad = \quad \alpha < \nabla W(x), \lambda - \mu^p(x) >$$

since $\nabla W(\alpha x) = \alpha \nabla W(x)$ and $\mu^p(\alpha x) = \mu^p(x)$ by radial homogeneity of μ^p, see (3.8). It follows that

$$< \nabla W(x), \lambda - \mu^p(x) > \quad \leq \quad -d$$

for all x in $\mathbb{R}_+^{\mathcal{R}} \setminus \{0\}$ and a constant $d > 0$. ■

Theorem 3.3.3 *If $A\lambda < \nu$ the Markov chain $\{N(t),\ t \geq 0\}$ associated with proportionally fair bandwidth allocation is positive recurrent.*

Proof: The method of proof for this theorem is analogous to that of our previous results. Since W is twice differentiable it follows by the Mean Value Theorem that for $n, m \in \mathbb{Z}_+^{\mathcal{R}}$ there exists a $\theta, 0 \leq \theta \leq 1$ such that

$$W(n+m) - W(n) = \langle \nabla W(n), m \rangle + \frac{1}{2} m^T \nabla^2 W(n + \theta m) m.$$

Note that the Hessian term yields $\nabla^2 W(x) = \text{diag}(1/2\lambda_r, r \in \mathcal{R})$, so we have from (3.11) noting $m = e^r$

$$QW(n) = \langle \nabla W(n), \lambda - \mu^p(n) \rangle + \langle h, \lambda - \mu^p(n) \rangle,$$

where $h = (1/2\lambda_r, r \in \mathcal{R})$. By radial homogeneity of μ^p and by Lemma 3.3.5, we have

$$\begin{aligned} QW(\alpha n) &= \langle \nabla W(\alpha n), \lambda - \mu^p(\alpha n) \rangle + \langle h, \lambda - \mu^p(\alpha n) \rangle \\ &= \alpha \langle \nabla W(n), \lambda - \mu^p(n) \rangle + \langle h, \lambda - \mu^p(n) \rangle \\ &\leq -\alpha d + \langle h, \lambda \rangle, \\ &\leq -\alpha d + |\mathcal{R}|/2, \end{aligned}$$

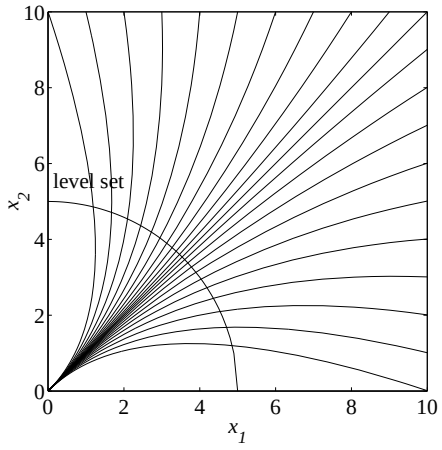

Figure 3.3: A vector field corresponding to proportionally fair bandwidth allocation of the example network.

where $\alpha > 0$ and $|\mathcal{R}|/2$ is a finite constant. Thus for sufficiently large $|n|$, the drift can be made negative. Letting $K = \{n : |n| \leq \gamma\}$ with large enough γ, we have $\sup_{n \notin K} QW(n) < 0$, which satisfies the drift condition (3.12) and implies positive recurrence. ■

The vector field corresponding to proportionally fair bandwidth allocation is shown in Fig. 3.3 when $\lambda = (1.5, 1.5)$ and $\nu = (5, 6, 4)$. Similarly, a weighted proportionally fair allocation of bandwidth can be considered as in [16], where the total weighted network utility is maximized:

$$\max_a \left\{ \sum_{r \in \mathcal{R}} w_r n_r U_r(a_r) : \sum_{r \in \mathcal{R}} A_{\ell r} w_r n_r a_r \leq \nu_\ell, \ell \in \mathcal{L}; a \geq 0 \right\}. \tag{3.29}$$

With the utility function $U_r(a_r) = \log a_r$, the rate allocation $a^{w,p*} = (a_r^{w,p*}, r \in \mathcal{R})$ solving (3.29) is *weighted proportionally fair* bandwidth allocation in the sense that for any other feasible rate $a' = (a'_r, r \in \mathcal{R})$, the aggregate weighted proportional change is negative, *i.e.*,

$$\sum_{r \in \mathcal{R}} w_r n_r \frac{a'_r - a_r^{w,p*}}{a_r^{w,p*}} < 0.$$

One can show the stability of the weighted proportionally fair allocation of bandwidth following a similar procedures as that for proportional fair allocation of bandwidth with the Lyapunov function:

$$W(x) = \sum_{r \in \mathcal{R}} \frac{(w_r x_r)^2}{2\lambda_r}.$$

3.4 Could the Internet be Unstable ?

3.4.1 Modeling of TCP

Internet traffic has been growing dramatically for the last few years. As of January 1999, the number of hosts advertised in the Domain Name Server (DNS) reached more than 43 million [58]. In many places the increase in demand is outpacing resources leading to congestion and degradation in performance. Since performance of Internet traffic is closely linked to the behavior of TCP congestion avoidance algorithm [25, 43], it is crucial to understand the impact of TCP on the macroscopic network level performance.

However, due to complicated interactions of Internet traffic and TCP transport algorithms [54], most research on the performance of TCP has relied on simulations for various TCP mechanisms. In an attempt to quantify throughputs of TCP connections more precisely and predictably, some researchers have started to consider analytical models and throughputs of TCP connections under various operating conditions, see *e.g.,* [25, 20, 43, 51]. Recently, this approach has drawn much attention and relevant work is ongoing.

Mathis et. al. [43] formulate a simple TCP model under the assumptions that (1) TCP is running over lossy path with constant Round Trip Time (RTT), and (2) Packet loss is random with constant probability of p. The TCP throughput, $BW(p)$, is derived as

$$BW(p) = \frac{1}{RTT}\sqrt{\frac{3}{2p}} \quad \text{packets/sec.}$$

The model is shown to match with real traffic when assumptions (1) and (2) hold. This model does not apply in some situations, *e.g.,* when "timeout" behavior is dominant or for the case of short connections which require only a few cycles of congestion avoidance. In

fact, real-life Internet traffic exhibits many timeouts compared with congestion avoidance behavior, *i.e.,* retransmission.

A recently developed model by Padhye et. al. [51] improves upon the previous one. The model captures not only congestion avoidance but also timeout behaviors that many real-life TCP traces exhibit. Moreover their model is shown to fit a wider range of operating conditions, *i.e.,* loss regimes. They assume that packet losses are correlated based on the fact that most current Internet employs drop-tail queueing policy and thus packets are likely to be lost again once previous packets experienced losses due to a full buffer. Their approximate model for TCP [7] throughput as a function of loss rate is

$$BW(p) = \min\left(\frac{W_{max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 \min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)}\right) \quad \text{packets/sec}$$

where W_{max} is the maximum congestion window size, b is the number of packets that are acknowledged by a received ACK, and T_0 is the time interval a sender waits before it starts retransmitting unacknowledged packets when a timeout occurs. Although the model may not fit into the TCP traces under different implementations such as TCP-tahoe or the Linux TCP implementation, it has been shown to match a broad range of real TCP traces and to predict the TCP throughput.

3.4.2 Macroscopic Modeling of the Internet

In this chapter we have considered the stability and performance of an idealized model for a network supporting services that adjust their transmissions to network loads. The model is only a *rough caricature* of the Internet today, in that it assumes TCP operates efficiently by immediately achieving an *average* throughput related to a weighted proportionally fair bandwidth allocation. For a single congested link, weighted max-min or weighted proportionally fair allocation model TCP appropriately [43], [16]. We believe that weighted max-min fair allocation can be adopted as a network model if weights are selected to reflect round-trip delays and TCP dynamics. So a connection's throughput is dictated by a

[7] They model TCP-reno which is the most popular implementation of TCP in the Internet.

weighted allocation of resources at congested or bottleneck links. The average RTT experienced by connections and loss rate can be captured by weights given to connections which in turn impact the equilibrium throughput achieved by TCP connections. This model parallels the one proposed and validated via simulation in [43]. We also assume that packets associated with a given TCP connection typically follow the same route, and connections send data in a greedy manner and depart. Subject to these, perhaps fanciful assumptions, one can show that network stability cannot be guaranteed unless the connection-level offered loads do not exceed the network's link capacities.

While this result is not entirely surprising, it presents an interesting architectural dilemma for future networks. Since routing algorithms on the Internet base their decisions on short term measures, *i.e.*, are not explicitly tracking the long-term averages required to assess the connection level offered loads, there is no reason to believe that the Internet would satisfy a connection level stability requirement. Instability would be perceived by users as an unacceptably low throughput, or inordinate delays, and typically cause them to abandon, thus in some sense solving the problem. To avoid such extremes one might overprovision the network. Unfortunately, this may result in a network which is still unstable, resulting in sporadic long lasting congestion events that are challenging to explain.

Currently we are researching using methodologies similar to those we have used to prove stability, to explore performance issues and consider in more depth the compromises one might make to achieve good performance at the connection level. It would of course be interesting to look at congestion patterns on the Internet today and attempt to explain them in terms of a connection-level instability. However, given the typically non-stationary demands on today's networks and the detailed data that would be required to provide a conclusive answer to this question this appears to be a challenging task.

Appendix

3.5 Proof of Lemma 3.3.2

The argument below is taken from Down and Meyn [18] that uses the property of radial homogeneity. We will construct a smooth W that, just as V, is also radially homogeneous. It then follows that $\nabla^2 W(x) \to 0$, as $|x| \to \infty$.

Recall that $V(x) = \max_{\ell \in \mathcal{L}} \langle \xi^\ell, x \rangle$. The idea is that one can perturb the vectors ξ^ℓ without changing the drift property. To explicitly exhibit the dependence of V on these vectors, define

$$V(\xi, x) = \max_{\ell \in \mathcal{L}} \langle \xi^\ell, x \rangle, \ x \in \mathbb{R}_+^{\mathcal{R}},$$

where $\xi = [\xi^\ell, \ \ell \in \mathcal{L}]$ is a $|\mathcal{R}| \times |\mathcal{L}|$ matrix and let $||\xi||$ be an appropriate matrix norm. By Lemma 3.3.1 we have that

$$\langle \xi^\ell, \lambda - \mu(x) \rangle \leq -c, \ \text{ if } \ell \in \mathcal{L}^{(1)}(\xi, x), \tag{3.30}$$

where $\mathcal{L}^{(1)}(\xi, x)$ denotes set of first-level bottleneck links, or, $\ell \in \mathcal{L}^{(1)}(\xi, x) \Leftrightarrow \langle \xi^\ell, x \rangle \geq \langle \xi^{\ell'}, x \rangle$, for all $\ell' \in \mathcal{L}$.

For a given x by continuity of (3.30) in x and ξ there exist $\delta, \varepsilon, c' > 0$, such that for $||\eta - \xi|| < \delta(x), |y - x| < \varepsilon(x)$ and $\ell \in \mathcal{L}^{(1)}(y, \eta)$ we have $\langle \eta^\ell, \lambda - \mu(y) \rangle \leq -c/2$. In fact this statement can be made to hold uniformly for a compact subset of $\mathbb{R}_+^{\mathcal{R}}$ containing the origin and then extended, using radial homogeneity, to $\mathbb{R}_+^{\mathcal{R}}$.

Now pick a smooth probability density $p(\eta)$ on the set $\{||\eta - \xi|| < \delta\}$ and define

$$W(x) = \int_{||\eta - \xi|| < \delta} V(\eta, x) p(\eta) d\eta.$$

One can see $W(x)$ is smooth on $\mathbb{R}_+^{\mathcal{R}} \setminus \{0\}$, by noting that $V(\eta, x)$ is smooth at x for p almost every η. Moreover it is radially homogeneous, and one can easily show that it satisfies

$$\langle \nabla W(x), \lambda - \mu(x) \rangle \leq -c/2 = -d.$$

■

3.6 Proof of Lemma 3.3.4

We will show the result by induction on the bottleneck hierarchy. Suppose the network state is n, and so the weight vector is $w(n)$ and let $\mathcal{L}^{(1),w(n)}(n)$ and $\mathcal{R}^{(1),w(n)}(n)$ denote the first level bottleneck links and routes associated with the corresponding weighted max-min fair problem. By (3.25) for any two routes $r, s \in \mathcal{R}$ traversing a given link $\ell \in \mathcal{L}$ we have

$$\lambda_s \geq \frac{\lambda_r}{n_r w_r(n)} n_s w_s(n).$$

Now summing over all routes traversing ℓ we have that

$$\sum_{s \in \mathcal{R}} A_{\ell s} \lambda_s \geq \frac{\lambda_r}{n_r w_r(n)} \sum_{s \in \mathcal{R}} A_{\ell s} n_s w_s(n).$$

This in turn means that

$$\lambda_r \leq \frac{n_r w_r(n)}{\sum_{s \in \mathcal{R}} A_{\ell s} n_s w_s(n)} \sum_{s \in \mathcal{R}} A_{\ell s} \lambda_s < \frac{n_r w_r(n)}{\sum_{s \in \mathcal{R}} A_{\ell s} n_s w_s(n)} \nu_\ell = n_r w_r(n) f_\ell^{(1),w(n)}(n),$$

where the strict inequality follows from the stability condition, and we recognize weighted fair share $f_\ell^{(1),w(n)}(n)$ at link ℓ, see (3.4). Suppose that $\ell \in \mathcal{L}^{(1),w(n)}(n)$, then $f_\ell^{(1),w(n)}(n) = f^{(1),w(n)}(n)$, and the right hand term corresponds to the bandwidth allocated to route r whence $\lambda_r < \mu_r^{w(n)}(n)$, $\forall r \in \mathcal{R}^{(1),w(n)}(n)$.

To continue by induction we need only show that the reduced problem also satisfies a stability condition *i.e.,* for all $\ell \in \mathcal{L} \setminus \mathcal{L}^{(1),w(n)}(n)$ we have that

$$\sum_{r \in \mathcal{R} \setminus \mathcal{R}^{(1),w(n)}(n)} A_{\ell r} \lambda_r < \nu_\ell^{(1),w(n)} = \nu_\ell - f^{(1),w(n)}(n) \sum_{r \in \mathcal{R}^{(1),w(n)}(n)} A_{\ell r} w_r(n) n_r.$$

By noting that $\sum_{r \in \mathcal{R}} A_{\ell r} n_r w_r(n) f_\ell^{(1),w(n)}(n) = \nu_\ell$ and $\lambda_r < n_r w_r(n) f_\ell^{(1),w(n)}(n)$, the above is easily shown, so it follows that

$$\begin{aligned}
\sum_{r \in \mathcal{R} \setminus \mathcal{R}^{(1),w(n)}(n)} A_{\ell r} \lambda_r &< \sum_{r \in \mathcal{R} \setminus \mathcal{R}^{(1),w(n)}(n)} A_{\ell r} n_r w_r(n) f_\ell^{(1),w(n)}(n) \\
&= \nu_\ell - f_\ell^{(1),w(n)}(n) \sum_{r \in \mathcal{R}^{(1),w(n)}(n)} A_{\ell r} w_r(n) n_r \\
&< \nu_\ell^{(1),w(n)}
\end{aligned}$$

since $f_\ell^{(1),w(n)}(n) > f^{(1),w(n)}(n)$ if $\ell \in \mathcal{L} \setminus \mathcal{L}^{(1),w(n)}(n)$. ■

Chapter 4

Performance and Design of Dynamic Networks Supporting Services with Flow Control

4.1 Introduction

In an effort to achieve efficient network utilization and to support elastic applications, adaptive services have been adopted, *e.g.*, ABR service in ATM networks and TCP in the Internet. Connections using this service class adapt their transmissions, which are controlled implicitly or explicitly based on congestion status and resource availability. Recently, adaptive services such as TCP and ABR service are drawing increased attention [42, 2, 4, 46, 43, 51].

Most research on adaptive services has focused on stability and transient analysis of flow control mechanisms of networks with "fixed" numbers of connections. However, users establish connections, transmit and receive possibly random amount of data, and disconnect. Thus connection arrivals and departures are stochastic in nature, which results in dynamic allocation of available resources. Although it is increasingly important to understand the behavior of dynamic networks supporting adaptive services, very little is known about their stability and performance.

In Chapter 3, networks with "dynamically" varying number of connections under dynamic rate allocations have been modeled via a Markov chain. [1] It has been shown that a natural condition is necessary for the stability of the dynamic model: the total load on any link does not exceed the link's capacity. However, the performance of dynamic networks is not readily known due to the "global" interactions underlying in dynamic rate allocation mechanisms. In this context, it is challenging to characterize exact performance (*e.g.*, average throughput or connection delays). Extensive simulations will be used to investigate the behavior of dynamic networks.

In dynamic networks, constrained (bottleneck) links and thus bandwidth allocation are dynamically changing over time. So it is questionable whether dynamic networks, operating under fair bandwidth allocation mechanisms (max-min or proportionally fair allocation), can be designed to meet delay performance requirements. Intuitively, one might dimension such networks by determining the bandwidth required for each route in 'isolation' in order to meet an average delay constraint on connections. Then by allowing routes to share these resources one would expect the overall average delays on the network to improve. Contrary to our expectations, this sharing of resources, can lead to degraded performance. In other words, although max-min and proportionally fair bandwidth allocation maximize individual throughput and overall network utility, [2] respectively, it is challenging to meet delay guarantees in a dynamic network.

In this chapter, we first consider using a *state-dependent weighted max-min fair allocation of bandwidth* in order to guarantee delay requirement of each route. Under the policy, a network can be designed to meet delay requirements by controlling weights of routes. This design method, however, has limitations in implementation since the rate allocation requires global information, *i.e.*, current number of ongoing connections. So we next propose a design method based on Generalized Processor Sharing (GPS) rate allocation and show that it indeed satisfies delay QoS requirements. The design method was motivated from bandwidth allocation in networks with fixed number of connections [52, 53]. We believe

[1] An earlier version of Chapter 3 can be found in [17].

[2] A user's utility is specified as a logarithmic function of the user's throughput.

that it provides a basis to network planning and service provisioning for dynamic network environments, *e.g.*, bandwidth allocation of Virtual Path (VP).

We shall explore connection level performance of dynamic networks operating under various bandwidth allocations, and design of dynamic networks guaranteeing delay performance. This chapter is organized as follows. Simulations are conducted in §4.2 in order to examine actual performance. The design of networks with dynamic connections to guarantee delay QoS requirements is presented in §4.3. Finally we summarize results in §4.4.

4.2 Simulations

As discussed earlier, it is challenging to fully quantify the performance of dynamic networks supporting services with adaptive allocations due to the complicated interactions among routes. Hence we shall resort to extensive simulations in an effort to further investigate the behavior of such networks. The objectives of simulations are 1) to understand the actual performance that we may expect to get, 2) to find how service policies affect the performance, and 3) to provide a basis to the design of dynamic networks guaranteeing connection-level delay QoS requirements.

We shall focus on average connection delay as our performance metric. This type of metric is of interest in dimensioning networks to provide a reasonable call-level quality of service. One might also wish to design network control mechanisms to assign priorities (weights) to routes, or to spread call level loads across the network in a manner that improves the individual or overall delays.

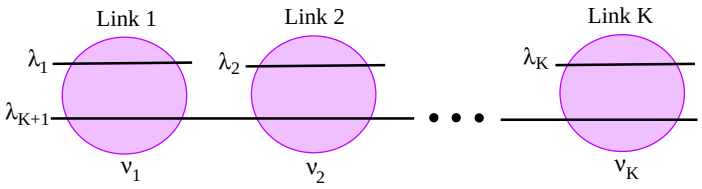

Figure 4.1: A network for simulations.

We shall consider a network consisting of K links in series, see Fig. 4.1. A long route traverses each link in the network, while short single-link routes, model "cross traffic." This network was adopted in order to consider how short (local) and long (transit) traffic interact each other. To investigate the degradation in performance as connections traverse an increasing number of links we simulated several configurations where $K = 2, 3, 4$ and 5. We simulated max-min, weighted max-min, and proportionally fair bandwidth allocation mechanisms in order to assess their impact on connection delays. In the case of weighted max-min fairness, short and long connections were given weights 1 and 2 respectively, *i.e.,* $w_r = 1, r = 1, \cdots, K$ and $w_{K+1} = 2$. Thus priority was given to connections traversing several links as they are likely to experience the poorest performance. Several symmetric and asymmetric load conditions were simulated to explore the impact of various load conditions on the network performance.

4.2.1 Symmetric Load

We first consider symmetric load conditions wherein long and short routes have the same traffic loads, *i.e.,* $\lambda_r = \lambda_s, \forall r, s \in \mathcal{R}$. The load conditions are summarized in Table 4.1.

Load conditions	λ_r, $r = 1, \cdots, K+1$	ν_ℓ, $\ell = 1, \cdots, K$
Light load	0.2	2.4
Moderate load	2.0	6.0
Heavy load	20.0	42.0

Table 4.1: Simulation environment (symmetric loads on all routes).

Both arrival rates and link capacities are in connections/sec. For each of scenario, the average overall connection delays as well as those on short and long routes, under max-min, weighted max-min, and proportionally fair allocation are measured as the number of links K increases, see Fig. 4.2 - Fig. 4.10. In general, as traffic load becomes heavier, and long routes traverse a larger number of links, average overall connection delay becomes large, regardless of the bandwidth allocation policy.

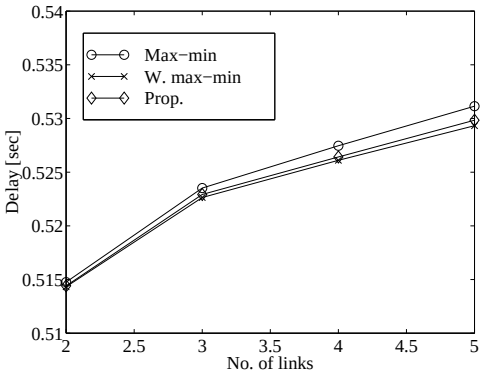

Figure 4.2: Average overall delay (ight load).

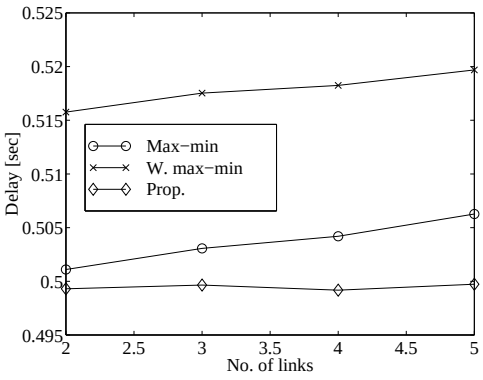

Figure 4.3: Average delay on short routes (light load).

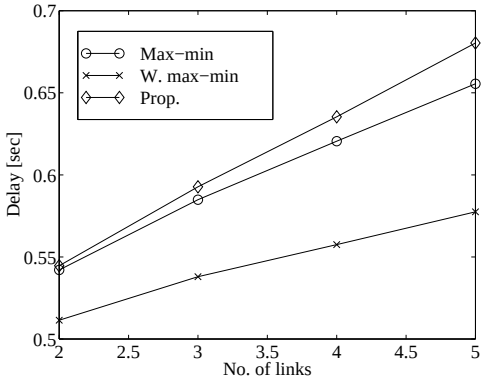

Figure 4.4: Average delay on long routes (light load).

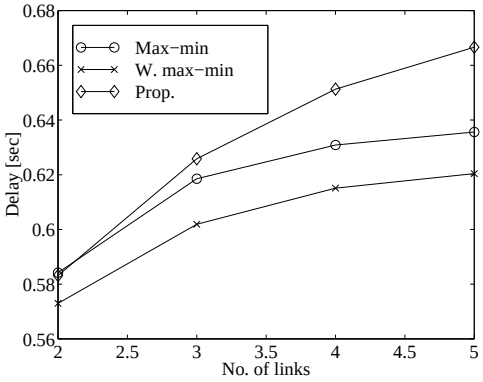

Figure 4.5: Average overall delay (moderate load).

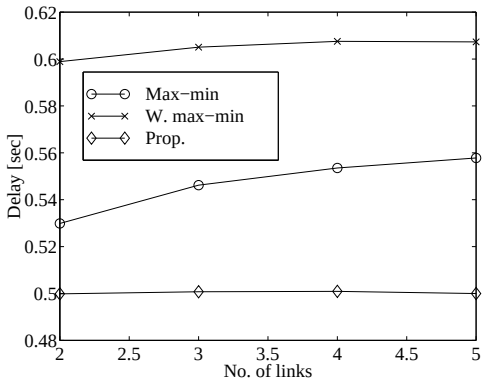

Figure 4.6: Average delay on short routes (moderate load).

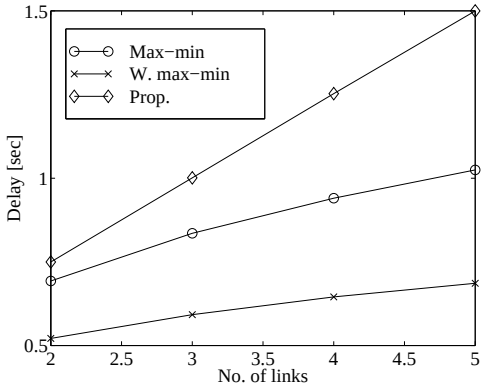

Figure 4.7: Average delay on long routes (moderate load).

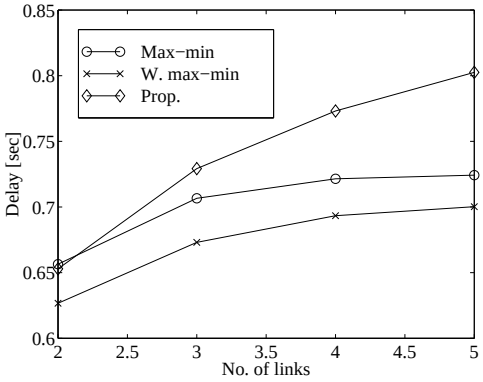

Figure 4.8: Average overall delay (heavy load).

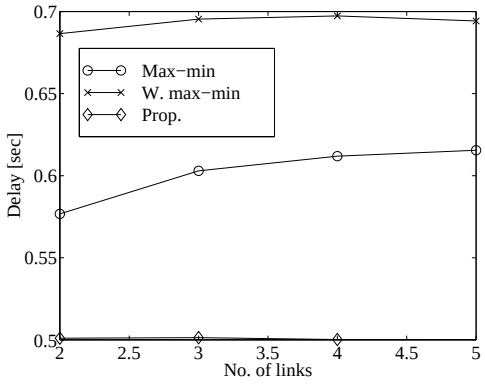

Figure 4.9: Average delay on short routes (heavy load).

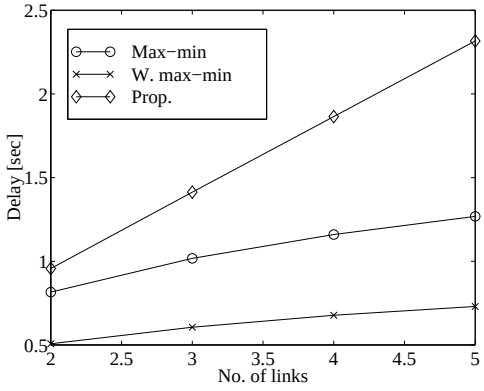

Figure 4.10: Average delay on long routes (heavy load).

We first contrast the performance of max-min fair bandwidth allocation, which strives to maximize the worst case individual performance versus proportional fairness, which strives to maximize the overall network utility. The latter tends to give more bandwidth to connections crossing a small number of links, as they are more efficient in terms of their resource requirements. As a result long routes may linger in the network possibly degrading the overall performance. For example, for $K = 5$ and moderate load, the relative change in delays for proportional versus max-min fair bandwidth allocation is -10 % on short routes, +46 % on long routes, and +5% overall, see Fig. 4.11 - Fig. 4.13. This effect is aggravated as the number of links in the network increases. For heavy load and when $K = 2, 3, 4,$ and 5, the relative change in delays for long routes is +17.3 %, +38.9 %, +60.8 %, and +82.7 %, respectively. Moreover as traffic load becomes heavier, the relative difference in delays for proportional versus max-min fair bandwidth allocation increases. For example, when $K = 5$, change in delays on long routes is +3.8 %, +46.4 % and +82.7 % for light, moderate, and heavy load, respectively.

This result demonstrates that the max-min outperforms the proportionally fair allocation in terms of delays on long routes and overall delays. Moreover, the change in delays for proportional versus max-min becomes larger as the size of network grows and the load of traffic becomes heavier. This suggests that maximizing overall utilities, which is a function of throughputs, is not necessarily compatible with minimizing connection delays. Note that as the number of links increases, proportional fairness leads to a surprisingly flat average delay on short routes, while long routes see a linear growth in average delay, [3] see Figs. 4.6 and 4.7. [4] This suggests that proportional fairness may provide a clean performance differentiation among routes that have different lengths.

Next, we consider the impact that using a weighted max-min fair bandwidth allocation will have on delays, if weights are selected so as to expedite connections on long routes. Clearly, weighted max-min fair allocation can provide a flexibility in allocating

[3] This phenomenon is believed to be a result of the specific network topology.

[4] The overall delay is not linear since it is an average of delays on short and long routes. Since the relative total load on short versus long routes is increasing with K, the overall delay behavior is not linear.

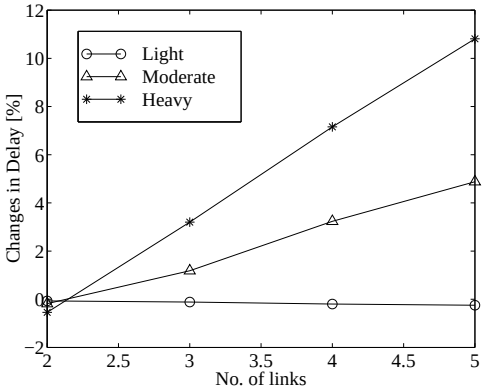

Figure 4.11: Change in delays, prop. over max-min (overall) - Symmetric loads.

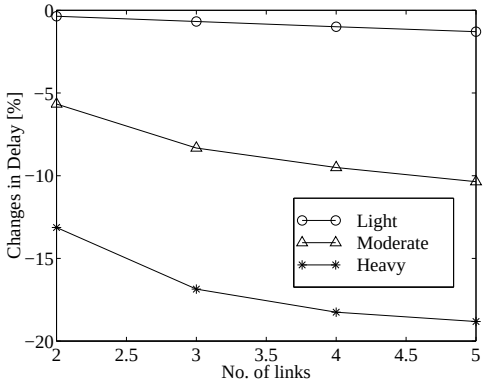

Figure 4.12: Change in delays, prop. over max-min (short routes) - Symmetric loads.

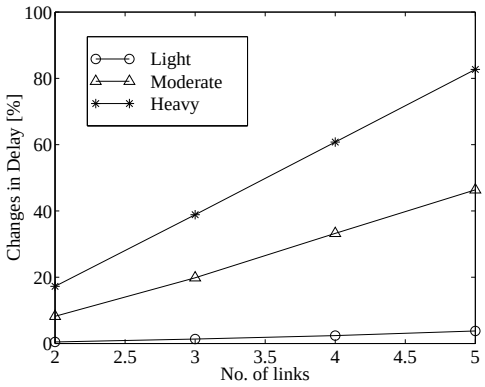

Figure 4.13: Change in delays, prop. over max-min (long routes) - Symmetric loads.

bandwidth over max-min fair allocation. Continuing with our example, when $K = 5$ and load is moderate, the relative change in delays for the weighted versus the max-min fair bandwidth allocation is +9 % on short routes, -33 % on long routes, and -2 % overall (Fig. 4.14 - 4.16). Thus, one can not only dramatically improve the delays experienced on long routes, but also marginally improve the overall performance.

However, for weighted max-min rate allocation, delays do not vary much with the length of long route and the intensity of traffic load, see Fig 4.14 - 4.16. When the load is heavy, the change in delays for weighted max-min versus max-min fair allocation on long routes is -37.9 %, -40.4 %, -41.6 %, and -42.4 % as $K = 2, 3, 4$, and 5, respectively. When $K = 5$, the change in delays is -11.9 %, -33.1 %, and -42.4 % for light, moderate, and heavy load, respectively. This result suggests that weights can be selected based on load conditions and lengths of routes, in order to improve network performance.

Hence we have measured the performance of a network with fixed K and load condition as weights for long routes vary. It turns out that overall performance is not continuously improved in proportion with the increase of weights given to long routes, although average delay on long routes decreases. For the moderate load condition and $K = 2$, performance is illustrated in Fig. 4.17 - 4.19. The overall delay is minimum when the weight $w_{K+1} = 3$, and then degrades as the weight w_{K+1} increases. This result shows that there is a tradeoff between improving individual delay performance and maximizing overall delay performance, which can be achieved by selecting weights (priorities).

In order to see how weights assigned to long routes impact on the performance, we assign weights for long routes to be the number of links long routes traverse, *i.e.*, $w_{K+1} = K$. The performance is shown in Fig. 4.20 - 4.22. "W. max-min2" represents this type of allocation. By this weighted max-min fair rate rate allocation ($w_{K+1} = K$), overall performance degrades as the size of network grows compared with that of the other weighted max-min fair rate rate allocation ($w_{K+1} = 2$). The reason is that short routes start to suffer from insufficient bandwidth allocation due to the priority given to long routes as the length of long routes increases. This result indicates that merely giving high priorities

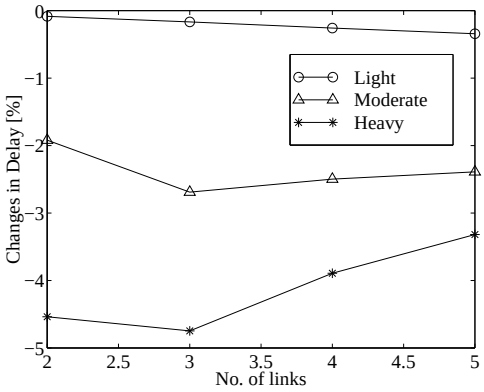

Figure 4.14: Change in delays, weighted max-min over max-min (overall) - Symmetric loads.

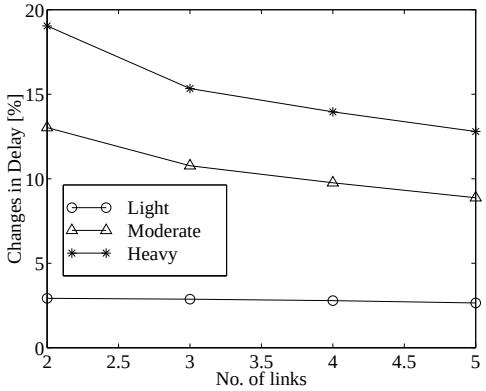

Figure 4.15: Change in delays, weighted max-min over max-min (short routes) - Symmetric loads.

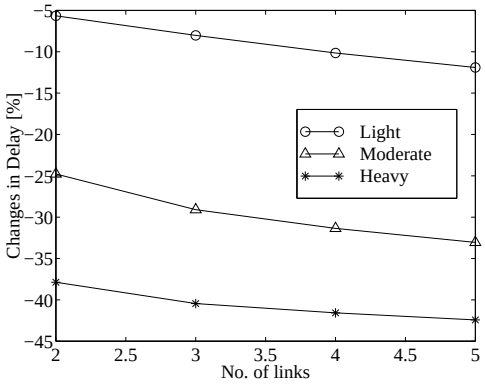

Figure 4.16: Change in delays, weighted max-min over max-min (long routes) - Symmetric loads.

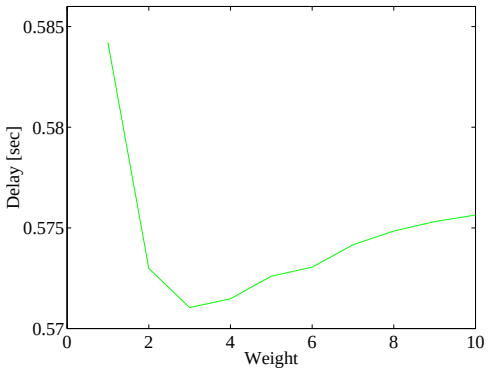

Figure 4.17: Overall delay as the weight on a long route increases (symmetric moderate load, $K = 2$).

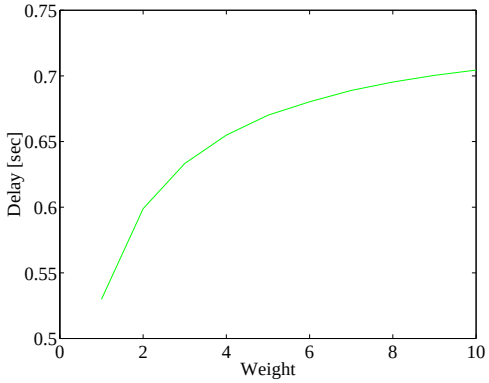

Figure 4.18: Average delay on short routes as the weight on a long route increases (symmetric moderate load, $K = 2$).

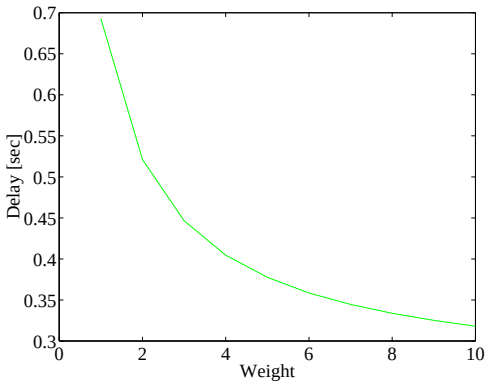

Figure 4.19: Average delay on long routes as the weight on a long route increases (symmetric moderate load, $K = 2$).

to long routes in proportion to the lengths of the routes does not guarantee an overall performance improvement. Herein lies a dilemma that a network designer will have to address if they truly wish to optimize average delays.

4.2.2 Asymmetric Load

We have also investigated asymmetric load conditions: 1) heavy load on long routes and 2) heavy load on short routes in order to examine the impact of "asymmetric" loads on the performance interactions between long and short routes. A simulation environment is

Load conditions	λ_r, $r = 1, \cdots, K$	λ_{K+1}	ν_ℓ, $\ell = 1, \cdots, K$
Asymmetric load 1	2	20	24
Asymmetric load 2	20	2	24

Table 4.2: Simulation environment (asymmetric loads).

summarized in Table 4.2. Performance for these asymmetric load conditions is illustrated in Fig. 4.23 to Fig. 4.28 as the number of links K increases.

As expected, the impact of heavy loads on long routes (asymmetric load 1) is much greater than that on short routes (asymmetric load 2) as shown in Fig. 4.29. For example, when $K = 5$, the change in average delay for proportional versus max-min fair rate allocation is +25.1 % and +0.4 % for asymmetric load 1 and asymmetric load 2, respectively. Moreover, the change in average delay for asymmetric load 1 increases as the number of links increases, *i.e.*, +2 %, +8.7 %, +16.7 %, and +25.1 % for K = 2,3,4, and 5. This result confirms that max-min outperforms proportionally fair rate allocation, and the difference becomes larger as the number of links and the amount of traffic load on long routes increase. As for the impact of weighted max-min fair allocation, the more the load on long routes is, the change in delays for weighted max-min versus max-min becomes greater, see Fig. 4.32 - 4.34. When $K = 5$, the change in delays of weighted max-min over max-min fair rate allocation is -14.4 % and -0.4 % for asymmetric load 1 and asymmetric load 2,

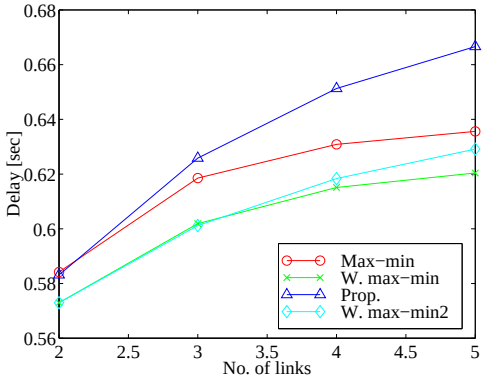

Figure 4.20: Overall delay when $w_{K+1} = K$ (symmetric moderate load).

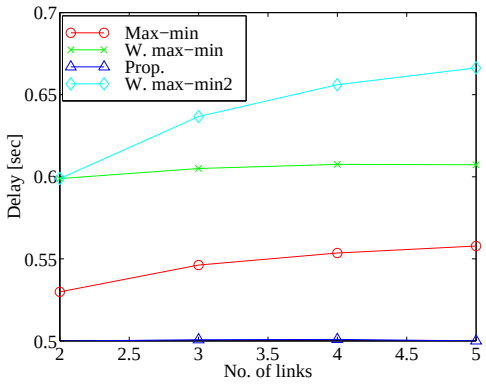

Figure 4.21: Average delay on short routes when $w_{K+1} = K$ (symmetric moderate load).

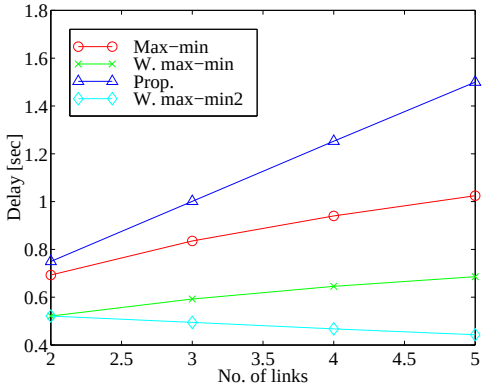

Figure 4.22: Average delay on long routes when $w_{K+1} = K$ (symmetric moderate load).

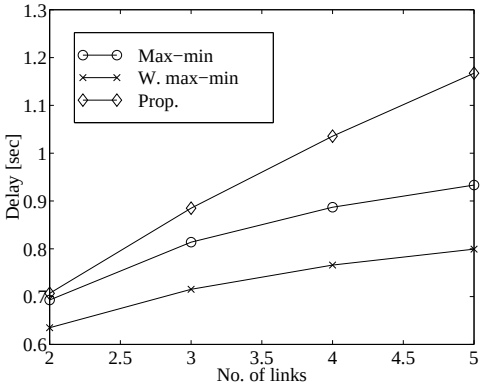

Figure 4.23: Average overall delay (asymmetric load 1).

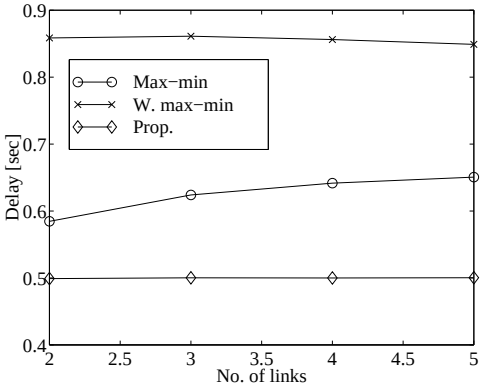

Figure 4.24: Average delay on short routes (asymmetric load 1).

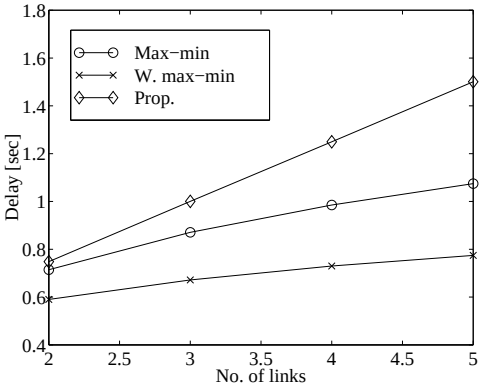

Figure 4.25: Average delay on long routes (asymmetric load 1).

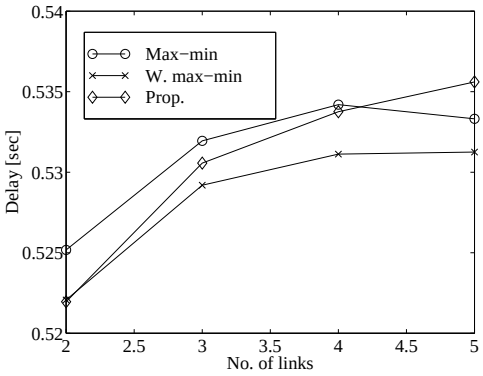

Figure 4.26: Average overall delay (asymmetric load 2).

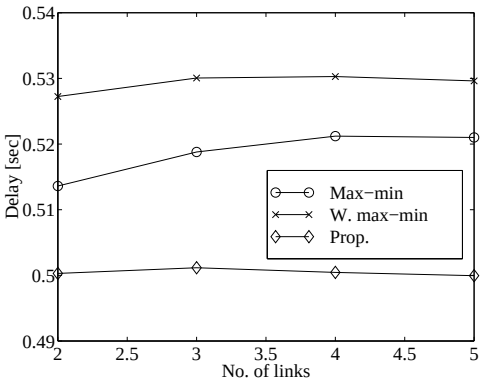

Figure 4.27: Average delay on short routes (asymmetric load 2).

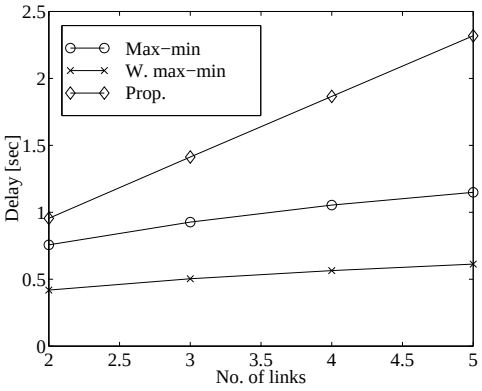

Figure 4.28: Average delay on long routes (asymmetric load 2).

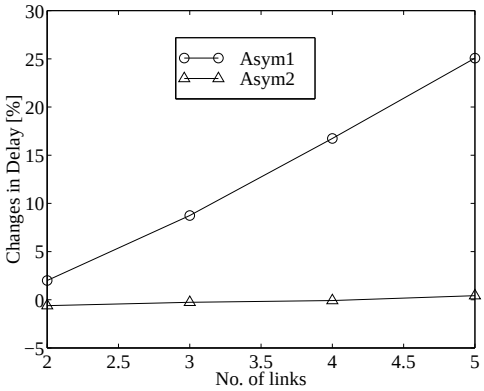

Figure 4.29: Change in delays for prop. versus max-min (overall) - Asymmetric loads.

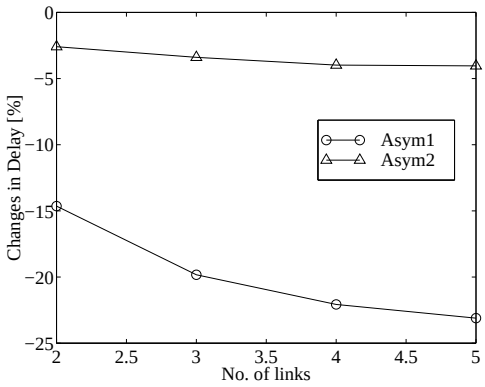

Figure 4.30: Change in delays for prop. versus max-min (short routes) - Asymmetric loads.

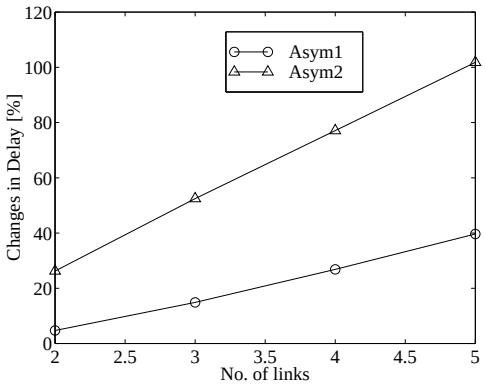

Figure 4.31: Change in delays for prop. versus max-min (long routes) - Asymmetric loads.

respectively. As the load becomes heavier and the number of links increases, the effect of using the weighted max-min rate allocation becomes even more noticeable, which is not the case in symmetric load conditions.

Although the total load on any link is the same for both of asymmetric load conditions, the average overall delay of a network with heavy load on long routes (asymmetric load 1) is much higher than that on short routes (asymmetric load 2). Furthermore, the average delay performance degrades much faster as the number of links increases in the case of asymmetric load 1 than that of asymmetric load 2. Therefore, heavy loads on long routes have much greater impact on the overall and individual delay performance, since long routes take more network resources for which connections are competing.

These results exhibit the potential impact that a fairness criterion selected by designers may have on network performance. However, a better characterization of network performance and tools to 'optimally' select weights, or route connections, will need to be developed if a call level quality of service such as that considered here is deemed important in future networks. Also note that one could in theory introduce weights on a proportionally fair allocation in order to also enhance the performance seen on long connections. Hence our results do not suggest that a particular mechanism is best, we merely suggest that a consideration of these issues might be warranted.

4.3 Design Problem

Based on our observations of the performance results, next we consider how one can design dynamic networks guaranteeing average delay requirements. More specifically,

- given a network topology, a load λ_r, and average delay requirements d_r^* on route r, how can one dimension link capacities ν_ℓ, to meet the delay requirements ?

We will consider this design problem in the sequel.

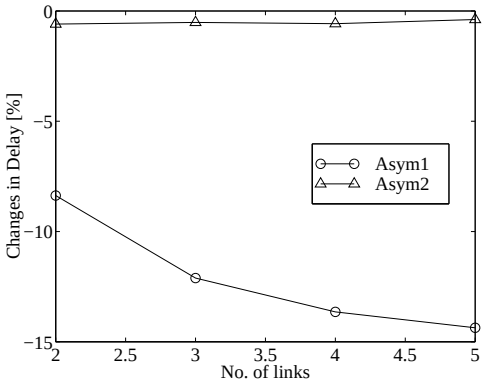

Figure 4.32: Change in delays for weighted max-min versus max-min (overall) - Asymmetric loads.

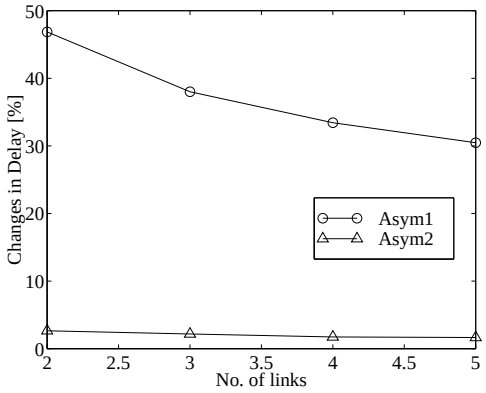

Figure 4.33: Change in delays for weighted max-min versus max-min (short routes) - Asymmetric loads.

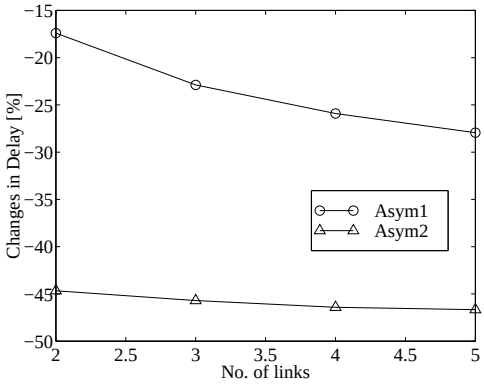

Figure 4.34: Change in delays for weighted max-min versus max-min (long routes) - Asymmetric loads.

4.3.1 Design by Separation of Routes

Let's first consider the network shown in Fig. 4.35. We are given demands λ_r, and delay

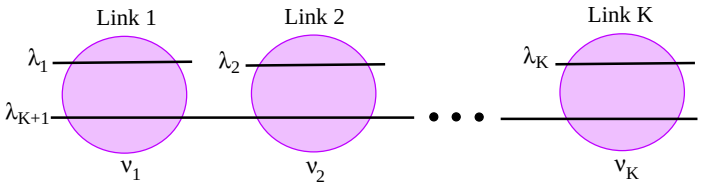

Figure 4.35: A network for design.

requirements d_r^* for the connections on route r. Suppose that a bandwidth μ_r is dedicated to each route, $r \in \mathcal{R}$, and there is no sharing across routes. Then each route would behave as an M/GI/1-PS (Processor Sharing) queue and the delays experienced $\mathbb{E}[D_r]$ are easily computed, see Fig. 4.36. Thus one can design the link capacity μ_r so as to satisfy the delay

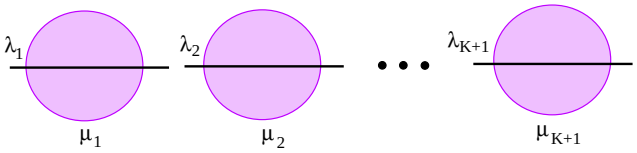

Figure 4.36: Separate links for design.

requirement d_r^* of each route from

$$\mathbb{E}[D_r] = \frac{1}{\mu_r - \lambda_r} \leq d_r^*, \qquad r \in \mathcal{R}.$$

We let the capacity $\mu_r = \lambda_r + \frac{1}{d_r^*}$. Now we add the bandwidth associated with each route traversing each link in the original network to decide the total capacities ν_ℓ, *e.g.*, $\nu_1 = \mu_1 + \mu_{K+1}$, as shown in Fig. 4.37. For example, when $\lambda_r = 2$ connections/sec and $d_r^* = 1$ sec, the bandwidth μ_r should be at least 3 connections/sec. Thus the capacity of each link is designed to be at least $\nu_\ell = 6$ connections/sec. Following this procedure, we expect that connections on all routes would meet their delay requirements d_r^* for max-min fair bandwidth allocation mechanisms. Contrary to our expectations, simulations reveal that average delay on the routes in the designed network operating under max-min or proportionally fair bandwidth allocation could in fact exceed the delay requirements.

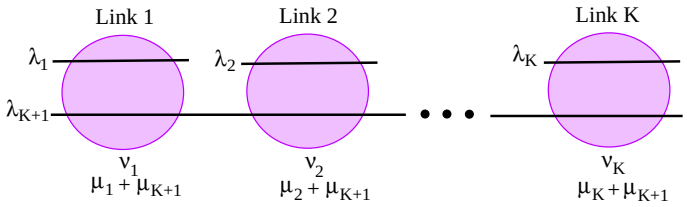

Figure 4.37: Designed network.

Indeed we designed networks for various load conditions subject to delay requirements $d_r^* = 1$ sec. The average delays experienced by the connections on short and long routes for each load condition and bandwidth allocation policy are tabulated in Table 4.3. Boldfaced numbers in the table indicate that the delay requirements are violated. For $K \geq 3$

Load condition	Policy		K=2	K=3	K=4	K=5
Light ($\lambda_r = 0.2, \nu_\ell = 2.4$)	Max-min	Short routes	0.501	0.503	0.504	0.506
		Long routes	0.542	0.585	0.621	0.656
	Prop.	Short routes	0.499	0.500	0.499	0.500
		Long routes	0.545	0.593	0.635	0.680
Moderate ($\lambda_r = 2, \nu_\ell = 6$)	Max-min	Short routes	0.530	0.546	0.554	0.558
		Long routes	0.693	0.835	0.940	**1.025**
	Prop.	Short routes	0.500	0.501	0.501	0.500
		Long routes	0.750	**1.001**	**1.253**	**1.500**
Heavy ($\lambda_r = 20, \nu_\ell = 42$)	Max-min	Short routes	0.577	0.603	0.612	0.616
		Long routes	0.816	**1.018**	**1.160**	**1.268**
	Prop.	Short routes	0.501	0.501	0.501	0.500
		Long routes	0.957	**1.413**	**1.865**	**2.317**

Table 4.3: Average connection delays on routes in the designed network.

and both moderate and heavy load conditions, all the delays on long routes under proportionally fair bandwidth allocation violate the delay requirement, $d_r^* = 1$ sec. For max-min fair allocation, connections on long routes experience longer delays than 1 sec, when $K = 5$ and load is moderate. This surprising result suggests that even individual route is designed to have enough bandwidth to meet delay QoS requirement, fair bandwidth alloca-

tion mechanisms may allocate insufficient bandwidth to the long routes when they compete for bandwidth. Thus even a "conservative" approach to design that ignores the benefits in sharing bandwidth across route fails to meet the desired delay requirements.

4.3.2 Design by State-dependent Weighted Max-min Fair Rate Allocation

In Chapter 3, a dynamic network model under state-dependent weighted max-min fair allocation of bandwidth is shown to be positive recurrent. We propose a design method to guarantee delay requirements of connections under state-dependent weighted max-min fair allocation of bandwidth.

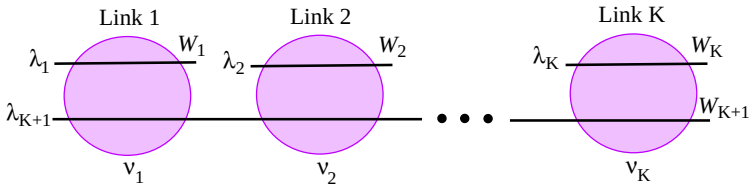

Figure 4.38: Network example for the design by state-dependent weighted max-min fair allocation of bandwidth.

Consider the network shown in Fig. 4.38. Our goal is to determine the link capacities ν_ℓ in such a way that the rate allocation allocates sufficient bandwidth to the connections on routes to meet delay requirement, *i.e.*, $\mathbb{E}[D_r] \leq d_r^*$. Consider a set of networks consisting of single link with capacity μ_r^* and its associated route. The routes experience the same loads as those of the original network as shown in Fig. 4.39. One can decide the capacity

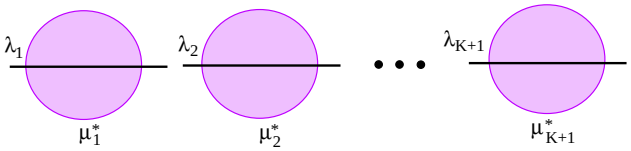

Figure 4.39: Set of networks for design.

of each link μ_r^* to satisfy the delay requirement d_r^* by

$$\mathbb{E}[D_r] = \frac{1}{\mu_r^* - \lambda_r} \leq d_r^*.$$

Thus when capacity of each link is at least

$$\mu_r^* = \lambda_r + \frac{1}{d_r^*}, \qquad r \in \mathcal{R},$$

the delay requirements are satisfied.

In the network of Fig. 4.38, the weight w_r can be designed to meet the delay QoS requirements as follows. If weights w_r are selected to be

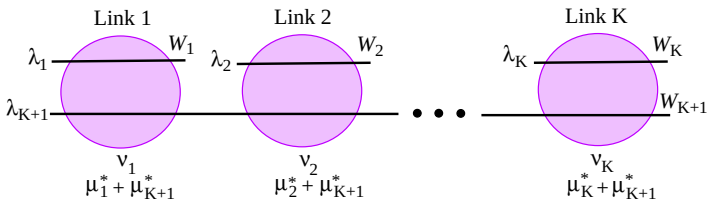

Figure 4.40: Designed network for state-dependent weighted max-min fair allocation of bandwidth.

$$w_r = \frac{\mu_r^*}{n_r}, \qquad r \in \mathcal{R},$$

then the actual bandwidth allocated to routes, μ_r, satisfies

$$\mu_r \geq \mu_r^*, \qquad r \in \mathcal{R},$$

and thus meets the average delay experienced by connections will meet the requirements on each route, *i.e.*,

$$\begin{aligned} \mathbb{E}[D_r] &= \frac{1}{\mu_r - \lambda_r} \\ &\leq \frac{1}{\mu_r^* - \lambda_r} \\ &\leq d_r^*, \qquad r \in \mathcal{R}. \end{aligned}$$

Therefore in the designed network shown in Fig. 4.40, average delay on each route is guaranteed to meet delay requirement.

In order to design a network in this way, weights will have to be adjusted at each new arrival and departure thus requiring additional control load on network switches. Finally,

we consider bandwidth allocation when switches support generalized processor sharing scheduling disciplines [52, 53], in which the number of connections does not need to be known and delay requirements are satisfied.

4.3.3 Design for Networks Supporting GPS Nodes

Again for our dynamic network model, weights for the routes at each node are to be designed so that delay QoS requirements d_r^* are met (see Fig. 4.38). First, consider each route separately from the others and associate it with a link as shown in Fig. 4.39. We have a delay requirement d_r^* for route r,

$$\mathbb{E}[D_r^*] = \frac{1}{\mu_r^* - \lambda_r} \leq d_r^*, \qquad r \in \mathcal{R}.$$

Let

$$\mu_r^* = \lambda_r + \frac{1}{d_r^*}. \tag{4.1}$$

Thus if the bandwidth allocated to each route, μ_r, satisfies

$$\mu_r \geq \mu_r^*, \qquad r \in \mathcal{R},$$

then the delay requirement will be met on each route.

Now let w_r be the weight assigned to route r and $\tilde{\mu}_r$ be minimum service rate for route r. By GPS rate allocation, the minimum service rate $\tilde{\mu}_r$ will be

$$\begin{aligned}
\tilde{\mu}_1 &= \frac{w_1}{w_1 + w_{K+1}}\nu_1, \\
\tilde{\mu}_2 &= \frac{w_2}{w_2 + w_{K+1}}\nu_2, \\
&\vdots \\
\tilde{\mu}_K &= \frac{w_K}{w_K + w_{K+1}}\nu_K, \\
\tilde{\mu}_{K+1} &= \min_{\ell}\left[\frac{w_{K+1}}{w_\ell + w_{K+1}}\nu_\ell\right].
\end{aligned} \tag{4.2}$$

We select weights and capacities to be

$$w_r = \mu_r^*, \qquad r = 1, 2, \cdots, K+1 \tag{4.3}$$

$$\nu_\ell = \mu_\ell^* + \mu_{K+1}^*, \qquad \ell = 1, 2, \cdots, K. \tag{4.4}$$

Then from (4.2) - (4.4), $\tilde{\mu}_r = \mu_r^*$, *i.e.*, the minimum service rate $\tilde{\mu}_r$ becomes the bandwidth required for the connections on route r, μ_r^*. So the bandwidth allocated to route r, μ_r, is at least μ_r^*:

$$\mu_r \geq \mu_r^* = \tilde{\mu}_r, \qquad r \in \mathcal{R}.$$

Hence sufficient bandwidth is allocated to each route and thus guarantees the delay QoS requirement:

$$\begin{aligned} \mathbb{E}[D_r] &= \frac{1}{\mu_r - \lambda_r} \\ &\leq \frac{1}{\mu_r^* - \lambda_r} \\ &\leq d_r^*, \qquad r \in \mathcal{R}. \end{aligned}$$

By (4.1) and (4.4), one can design

$$\nu_\ell = (\lambda_\ell + \frac{1}{d_\ell^*}) + (\lambda_{K+1} + \frac{1}{d_{K+1}^*}), \qquad \ell = 1, \cdots, K.$$

4.3.4 Comparison of the Designs

We have proposed several design methods in the preceding subsections. Next question of interest is how much excess bandwidth needs to be provided to guarantee the delay requirements. We assume the same loads and delay QoS requirements for all the routes, *i.e.*, $\lambda_r = \lambda$ and $d_r^* = d^*$, $r \in \mathcal{R}$, see Fig. 4.41.

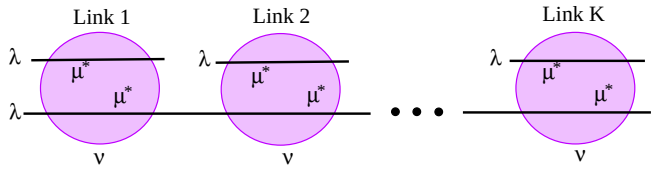

Figure 4.41: A network with the same loads and capacities.

Let's compare the total capacities required in the network.

- Design for networks supporting GPS nodes

  From the delay requirement

$$\mathbb{E}[D_r] = \frac{1}{\mu^* - \lambda} \leq d^*, \qquad r \in \mathcal{R},$$

bandwidth allocated to each route must be at least

$$\mu^* = \lambda + \frac{1}{d^*}.$$

For the network to meet delay requirements, required total capacity is

$$K(2\mu^*) = 2K(\lambda + \frac{1}{d^*}), \tag{4.5}$$

since each link needs capacity of $2\mu^*$. Consider, for example, a network with $K = 10$, wherein $\lambda = 10$ connections/sec, and mean number of bits that connections bring in is 2 Mbits. If delay requirement d^* is 2 sec, total capacity needed should be 210 Mbps. It increases to 1.2 Gbps if connections require stringent delay requirements of 20 msec.

- Optimal allocation

  Consider an optimal allocation, where bandwidth at each link is shared among all the connections traversing the link. In this allocation, the average delay of connections on each route should meet

  $$\mathbb{E}[D_r] = \frac{1}{\nu - 2\lambda} \leq d^*, \qquad r \in \mathcal{R}.$$

  So we let

  $$\nu = 2\lambda + \frac{1}{d^*}.$$

  Total capacity required in the network is

  $$K\nu = K(2\lambda + \frac{1}{d^*}). \tag{4.6}$$

Now we compare total capacities by computing relative gain from (4.5) and (4.6). The relative gain of optimal allocation over design by GPS can be computed as

$$\frac{2K(\lambda + \frac{1}{d^*}) - K(2\lambda + \frac{1}{d^*})}{K(2\lambda + \frac{1}{d^*})} \times 100 = \frac{\frac{1}{d^*}}{2\lambda + \frac{1}{d^*}} \times 100\%.$$

This relative gain corresponds to relative amount of overprovisioning of design by GPS over optimal allocation. Note that as traffic load λ increases, total capacity needed by

design by GPS approaches to that of optimal allocation. For example, when d^* = 1 sec and λ = 2 connections/sec, the relative gain is 20 % while it is only 2.4 % when $\lambda = 20$ connections/sec.

4.4 Summary

In this chapter we have explored performance of dynamic networks operating under max-min, weighted max-min or proportionally fair rate allocation. Simulations have revealed the impact of dynamic rate allocations. We have shown that long routes greatly affect overall and individual delay performance, which is aggravated as the loads on long routes become heavier. We have drawn an interesting conclusion that proportionally fair rate allocation does not necessarily maximize overall delay performance although it maximizes the overall benefit given by a logarithmic function of the throughput. With a reasonable selection of weights, *i.e.,* high priorities to long routes, one can improve the performance of the network over plain max-min fair rate allocation. However, overall delay can be degraded due to the insufficient bandwidth allocation to short routes although delay performance on long routes continuously improves. This observation suggests that weights need to be optimized in order to minimize overall delay.

In dynamic networks under dynamic rate allocations, we have shown that throughput, *e.g.,* individual throughput or overall utility, and delay guarantees may not be achieved simultaneously in general. Thus designing dynamic networks to meet delay requirements under fair rate allocations is challenging. We believe that it is originated from the global interactions in dynamic rate allocations. That may lead to an alternative rate allocation taking delay guarantees into account rather than throughput guarantees. In an effort to provide design methods guaranteeing delay requirements in dynamic networks, we have proposed GPS type bandwidth allocations, which will be useful in designing networks with dynamic connections subject to delay constraints of connections.

Chapter 5

Performance and Design of Multiservice Networks

5.1 Analysis and Design of Multiservice ATM Networks: Single Node

In current networks, various traffic types (*e.g.,* voice, data) are typically carried through logically separate networks (*e.g.,* telephone networks and the Internet). Service differentiations to efficiently carry various traffic types with different QoS via single network entity are envisioned for future broadband networks. ATM networks are expected to be a good candidate to provide this infrastructure.

As part of the efforts to deploy ATM networks, the ATM forum has defined several service classes including ABR for various conceivable applications [24]. ABR service is aimed to utilize the remaining bandwidth after other service classes, *i.e.,* VBR (Variable Bit Rate) or CBR (Constant Bit Rate) service, have been assigned their requested bandwidth. Such networks can provide more efficient network utilization from the network providers point of view, and let users economically share available resources with loose QoS requests on the other hand.

It is expected that heterogeneous service classes will be carried through ATM networks simultaneously. In this environment, it is important to understand the overall performance both for network design and analysis. Research has been conducted on the performance using simulations [50, 27, 21, 22, 40, 49, 59, 39], however, little attention has been paid to the analytical study of network performance analysis under such environments.

Our purpose in this chapter is to model ABR and CBR traffic, and analyze the performance, where first a single node case is considered. As bandwidth occupied by CBR connections changes dynamically, the available bandwidth for ABR service also varies. So it can be considered that ABR is operating under randomly varying bandwidth environment due to the CBR traffic. We model this by a two dimensional Markov chain, and formulate QBD (Quasi-Birth-Death) process and matrix-geometric equation [47]. By solving the equation, we can identify the performance ABR services will see.

Furthermore, when CBR and ABR are operating on different time scales, one can use quasi-stationary approximations to evaluate system performance. We observe that typically such approximations closely follow the exact performances as long as there is indeed a separation of time scales.

We, then, apply this performance analysis to the design of a network carrying both types of traffic. For example, we can estimate how many ABR connections we will observe on average under throughput of ABR connections subject to a fair share service policy and average delay constraints.

In §5.1.1, we present the model, and in §5.1.2 we formulate a Markov chain for the model. Next we derive a matrix-geometric equation to provide exact average performance of ABR connections. In §5.1.3, we approximate the performance based on the time scale separation of two services. Design issues are considered in §5.1.5.

5.1.1 Model

Consider a single link with capacity C shared by CBR and ABR connections (see Fig. 5.1). CBR connections arrive at the link with Poisson arrival rate ν and have a connection

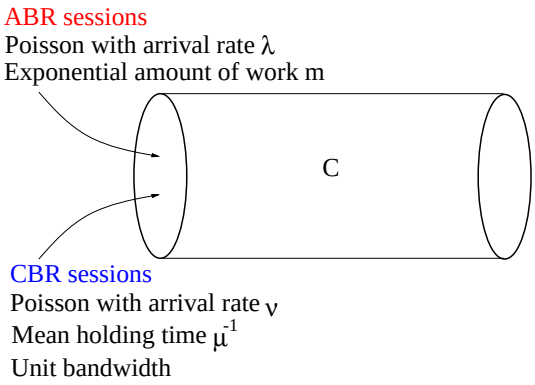

Figure 5.1: Model of ABR and CBR connections to a link.

holding time with exponential distribution and mean μ^{-1}. CBR connections are assumed to require unit bandwidth. So the number of CBR connections that the link can admit is no more than C [1]. In addition, ABR sessions enter the link with Poisson arrival rate λ, and the amount of work (bits) to be done is exponentially distributed with parameter m. Thus the average volume of bits per ABR session will be $1/m$. We assume that ABR sessions share the available resources fairly in the max-min sense, and the available resources are those not allocated to CBR connections.

5.1.2 Analysis

We can formulate a two dimensional Markov chain for the model above, in which a state (i, j) denotes a number i of CBR connections and j ABR connections, see Fig 5.2. Let $\pi(i, j)$ denote the stationary distribution for the numbers of CBR and ABR connections. The transition rate from (i, j) to $(i + 1, j)$ is ν, the rate from $(i + 1, j)$ to (i, j) is $(i + 1)\mu$, the rate from (i, j) to $(i, j + 1)$ is λ, and the rate from $(i, j + 1)$ to (i, j) is η_i, where $\eta_i = ((C - i) + r)/(1/m)$ is the effective service rate of ABR sessions when i CBR connections are present. We introduce a reserved bandwidth r with $r > \rho$ and $\rho = \lambda/m$ to guarantee the connection-level stability of ABR connections. Let $\Pi_j = [\pi(0, j)\, \pi(1, j)\, \cdots\, \pi(C, j)]$

[1] In case CBR requires bandwidth b instead, the number of possible CBR connections to the link will be C/b.

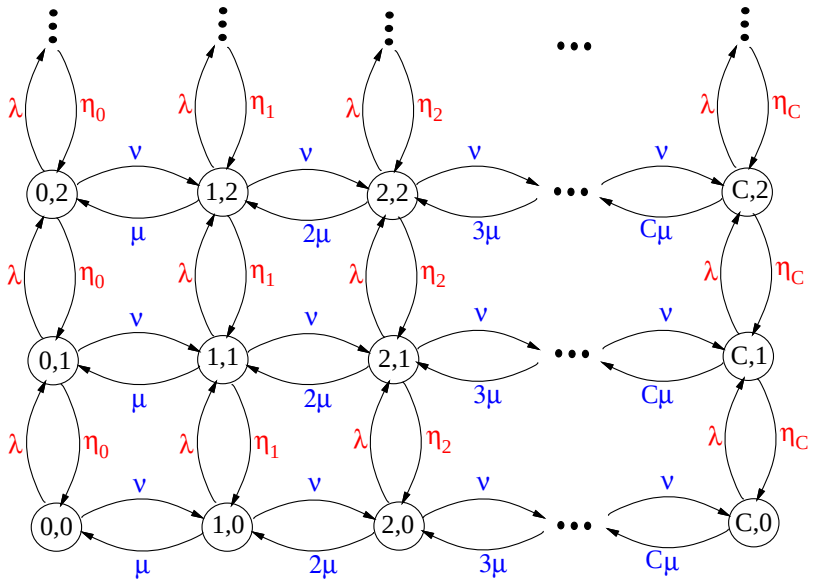

Figure 5.2: Markov chain for the model.

be the vector of stationary distribution with j ABR connections.

Then we have a following balance equation:

$$\nvDash \mathbb{Q} = \vec{0}, \tag{5.1}$$

where

$$\nvDash = [\Pi_0 \, \Pi_1 \, \cdots]$$

and

$$\mathbb{Q} = \begin{bmatrix} A - \Delta(\lambda) & \Delta(\lambda) & 0 & \ddots & \ddots \\ \Delta(\eta) & A - \Delta(\lambda + \eta) & \Delta(\lambda) & 0 & \ddots \\ 0 & \Delta(\eta) & A - \Delta(\lambda + \eta) & \Delta(\lambda) & \ddots \\ \ddots & 0 & \Delta(\eta) & A - \Delta(\lambda + \eta) & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

and where

$$A = \begin{bmatrix} -\nu & \nu & 0 & \ddots & \ddots \\ \mu & -(\mu+\nu) & \nu & 0 & \ddots \\ 0 & 2\mu & -(2\mu+\nu) & \nu & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & 0 & C\mu & -(C\mu+\nu) & \nu \end{bmatrix},$$

$$\begin{aligned} \Delta(\lambda) &= \text{diag}[\lambda\ \lambda \cdots \lambda], \\ \Delta(\eta) &= \text{diag}[\eta_0\ \eta_1 \cdots \eta_C], \\ \Delta(\lambda+\eta) &= \text{diag}[\lambda+\eta_0\ \lambda+\eta_1 \cdots \lambda+\eta_C]. \end{aligned}$$

Note that the matrix $\mathbb{Q}$ constitutes QBD process [47]. The equation has a matrix-geometric solution given by

$$\Pi_k = \Pi_0 R^k = \vec{\pi}(I-R)R^k, \tag{5.2}$$

where $\vec{\pi}A = \vec{0}$ with $\vec{\pi} = [\pi(0)\,\pi(1) \cdots \pi(C)]$, which is a balance equation of M/M/C/C queue, and R is the minimum non-negative solution to the following equation

$$R^2\Delta(\eta) + R(A - \Delta(\lambda+\eta)) + \Delta(\lambda) = \vec{0} \tag{5.3}$$

with boundary conditions for Π_0

$$\Pi_0(R\Delta(\eta) + (A - \Delta(\lambda))) = \vec{0} \tag{5.4}$$

$$\Pi_0(I-R)^{-1}\vec{e}^T = 1, \tag{5.5}$$

where $\vec{e} = [1\,1\,\cdots]$.

In general, it is difficult to find a closed form solution of Eq. (5.3). One could solve the equation numerically and find the matrix R. In this case average number of ABR connections is given by

$$\mathbb{E}[N_{ABR}] = \sum_k k\Pi_k = \vec{\pi}R(I-R)^{-1}\vec{e}^T, \tag{5.6}$$

and by Little's law, the average delay experienced by ABR connections is

$$\mathbb{E}[D_{ABR}] = \frac{1}{\lambda}\mathbb{E}[N_{ABR}]. \tag{5.7}$$

5.1.3 Approximation

If the connection set-up and tear-down of ABR is fast relative to that of CBR, *i.e.*, ABR is operating in fast time scale while CBR is in slow time scale, we expect that both time scales can be separated. In fact, we can establish the following theorem.

Theorem 5.1.1 *Suppose $\lambda, m \to \infty$ with $\rho = \lambda/m$ fixed, then*

$$\pi(i,j) = \pi^1(i)\pi^2(j|i), \tag{5.8}$$

where $\pi^2(j|i) = \rho_2(i)^j(1-\rho_2(i))$ with $\rho_2(i) = \frac{\lambda}{\eta_i}$, and $\pi^1(i) = G^{-1}\frac{\rho_1^i}{i!}$ with $G = \sum_{k=0}^{C}\frac{\rho_1^k}{k!}$ and $\rho_1 = \frac{\nu}{\mu}$.

Proof: By rearranging terms in (5.3), we have

$$(I-R)(R-\Delta(\lambda)\Delta(\eta)^{-1}) = RA\Delta(\eta)^{-1}.$$

Thus

$$\begin{aligned} R &= \Delta(\lambda)\Delta(\eta)^{-1} + (I-R)^{-1}RA\Delta(\eta)^{-1} \\ &= \Delta(\lambda)\Delta(\eta)^{-1} + RA\Delta(\eta)^{-1} + (I-R)^{-1}R^2A\Delta(\eta)^{-1} \\ &= \Delta(\lambda)\Delta(\eta)^{-1} + \Delta(\lambda)\Delta(\eta)^{-1}A\Delta(\eta)^{-1} + \\ &\quad (I-R)^{-1}RA\Delta(\eta)^{-1}A\Delta(\eta)^{-1} + (I-R)^{-1}R^2A\Delta(\eta)^{-1} \\ &= \Delta(\lambda)\Delta(\eta)^{-1} + O(\Delta(\eta)^{-2}). \end{aligned} \tag{5.9}$$

Note that $\eta_i = ((C-i)+r)/(1/m)$ becomes large since we assume that $m \to \infty$. Thus the second term $O(\Delta(\eta)^{-2})$ in (5.9) becomes small. So R can be approximated by $R \approx \Delta(\lambda)\Delta(\eta)^{-1}$, which is independent of CBR matrix A. Thus by (5.2),

$$\begin{aligned} \Pi_j &= \vec{\pi}(I-R)R^j \\ &\approx \vec{\pi}(I-\Delta(\rho))\Delta(\rho)^j, \end{aligned} \tag{5.10}$$

where $\Delta(\rho) = \Delta(\lambda)\Delta(\eta)^{-1}$ and we have

$$\pi(i,j) = \pi^1(i)\pi^2(j|i), \tag{5.11}$$

with $\pi^2(j|i) = \rho_2(i)^j(1-\rho_2(i))$ and $\rho_2(i) = \frac{\lambda}{\eta_i}$, and $\pi^1(i) = G^{-1}\frac{\rho_1^i}{i!}$ with $G = \sum_{k=0}^{C}\frac{\rho_1^k}{k!}$ and $\rho_1 = \frac{\nu}{\mu}$. ■

The theorem implies that as the arrival rate of ABR λ increases and the mean amount of work to be done $1/m$ decreases, the stationary distribution of ABR and CBR can be separated and expressed as product of each individual distribution. Note that $\pi^1(i)$ is a stationary distribution of M/M/C/C queue and $\pi^2(j|i)$ is a stationary distribution of M/M/1 queue conditioned on CBR is in state i.

Based on the approximation $R \approx \Delta(\lambda)\Delta(\eta)^{-1} = \Delta(\rho)$ and by (5.6), the approximate average number of ABR connections will be

$$\begin{aligned}\mathbb{E}[N_{ABRapprox}] &= \vec{\pi}(I-\Delta(\rho))^{-1}\Delta(\rho)\vec{e}^T \\ &= \sum_i \frac{\rho_2(i)}{1-\rho_2(i)}\pi^1(i) \\ &= \sum_i \frac{\lambda}{\eta_i-\lambda}\pi^1(i). \end{aligned} \tag{5.12}$$

Note that in the approximation ABR service constitutes M/M/1 queue conditioned that the number of CBR connections i is fixed. The approximate average delay experienced by ABR connections is given by

$$\mathbb{E}[D_{ABRapprox}] = \frac{1}{\lambda}\mathbb{E}[N_{ABRapprox}]. \tag{5.13}$$

5.1.4 Example

Based on §5.1.2 and §5.1.3, we present an example in this section. The simulation environment is summarized in Table 5.1. We consider an OC3 link with capacity 150 Mbps and CBR sessions requiring 2 Mbps bandwidth corresponding to MPEG sources.

The proposed approximation is based on the idea that ABR connections might come and go much faster than CBR connections. Thus conditioning on a given number of CBR connections, we might assume the distribution of ABR connections has reached "steady state." In order for this to be the case, the time scale on which the number of CBR connections in the system changes should be slow relative to that on which the ABR "queue"

Parameter		Value	
	C	150	Mbps
CBR	ν	10 - 200	connections/hour
	μ	1	connections/hour
	b (bandwidth/CBR)	2	Mbps
ABR	λ	10 - 200	connections/hour
	$\rho = \lambda/m$	8	kbps
	$1/m$	2.88M - 144k	bits/connection
	r	9, 10	kbps

Table 5.1: Parameters for an example.

reaches steady state.

To determine when this is the case, we compute a ratio between these time scales. Since the average number of CBR connections is $\kappa := \mathbb{E}[N_{CBR}]$, [2] we can approximate average time until a CBR connection arrives or leaves by

$$\text{Time-scale}_{CBR} = (\nu + \kappa\mu)^{-1}. \tag{5.14}$$

For ABR connections, the link behaves like M/GI/1-Processor Sharing queue which has the same characteristics as M/M/1 queue. For this queue, the service capacity is typically $(C - \kappa b) + r$. Since the mean number of bits for ABR connections is $1/m$, the effective service rate for ABR connections would typically be about

$$\sigma = \frac{(C - \kappa b) + r}{1/m}.$$

We now compute the *relaxation time* for this queue, *i.e.,* an approximate time to reach steady state [5]. It has been shown, see [5], that the relaxation time for such a system is given by

$$\begin{aligned} \text{Time-scale}_{ABR} &= (\sqrt{\lambda} - \sqrt{\sigma})^{-2} \\ &= (\lambda + \sigma - 2\sqrt{\lambda\sigma}\,)^{-1}. \end{aligned} \tag{5.15}$$

[2] Assuming CBR connections experience small blocking probability, one can also approximate the average number of CBR connections $\mathbb{E}[N_{CBR}] \approx \rho_{CBR} = \nu/\mu$.

Hence, from (5.14) and (5.15), the ratio between the time scales is given by

$$\begin{aligned} \text{Ratio} &= \frac{\text{Time-scale}_{ABR}}{\text{Time-scale}_{CBR}} \\ &= \frac{(\lambda+\sigma-2\sqrt{\lambda\sigma}\,)^{-1}}{(\nu+\kappa\mu)^{-1}} \\ &= \frac{\nu+\kappa\mu}{\lambda+\sigma-2\sqrt{\lambda\sigma}}. \end{aligned}$$

Based on this ratio we can approximately see when our approximation might hold. For example, if the ratio is small enough, say 0.05, then we could say that the time scales are separated. Fig. 5.3 illustrates how this ratio varies as CBR and ABR arrival rates change for the operating condition given in Table 5.1. From the figure, we note that the ratio becomes

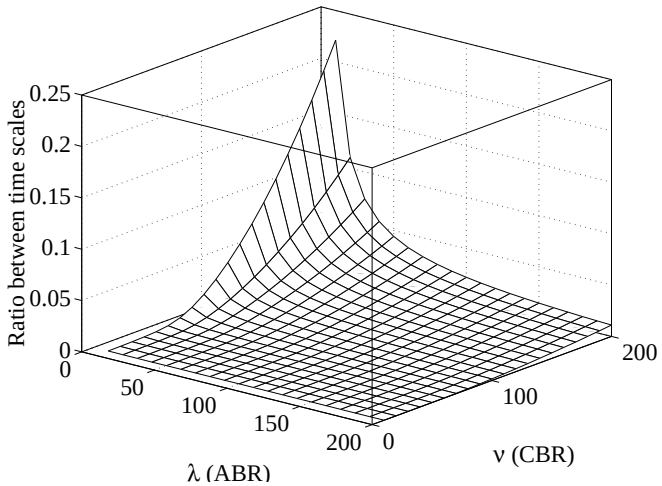

Figure 5.3: Ratio between time scales when $r = 9$ kbps.

less than 0.05 if $\lambda = 10$ conn./hour and $\nu \leq 120$ conn./hour. Thus, when ABR sessions are slow, *e.g.*, $\lambda = 10$ and $\nu > 120$, we should observe a noticeable difference between the true value and the approximation since the ratio exceeds 0.05 - Fig. 5.4 shows that this is indeed the case. When $\lambda \geq 50$ conn./hour and $\nu \in [10, 200]$ conn./hour, the ratio is shown to be less than 0.05. Thus in this case, we might expect a "separation of time scales" to occur and the approximation to be good, see *e.g.*, Fig. 5.5 when $\lambda = 200$ and $\nu \in [10, 200]$. As the reserved bandwidth r for ABR sessions increases from 9 kbps to 10 kbps, the time scales

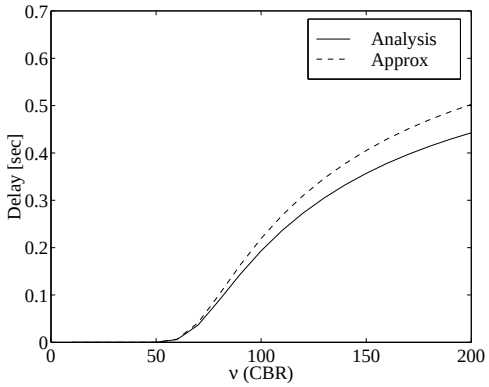

Figure 5.4: Average delay as ν (CBR) changes for a given $\lambda = 10$ (ABR) with r = 9 kbps.

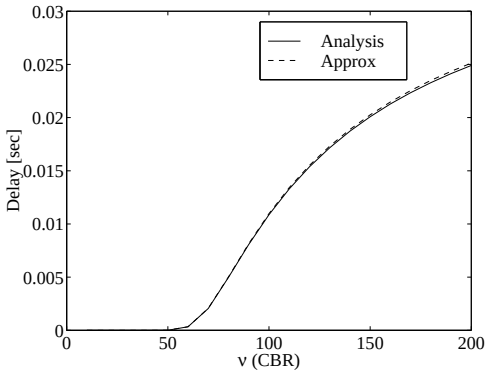

Figure 5.5: Average delay as ν (CBR) changes for a given $\lambda = 200$ (ABR) with r = 9 kbps.

are further separated, which results in even higher approximation accuracy, see Figs. 5.6 and 5.7. For example, when $\lambda = 10$ and $\nu = 100$, the relative difference between the true

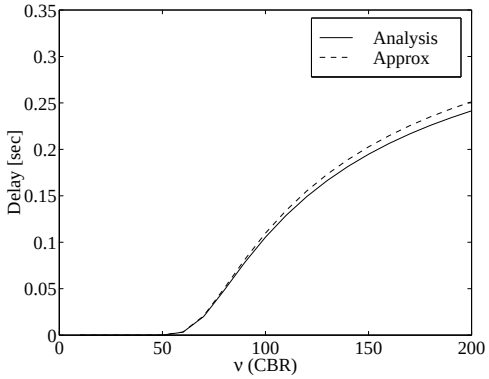

Figure 5.6: Average delay as ν (CBR) changes for a given $\lambda = 10$ (ABR) with r = 10 kbps.

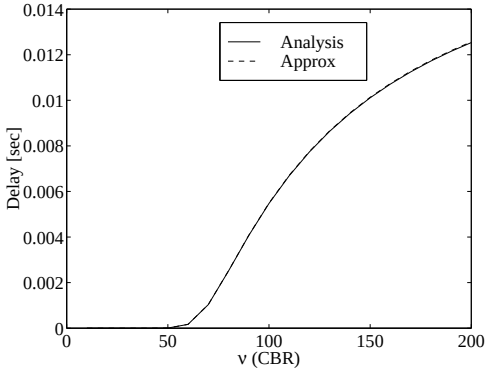

Figure 5.7: Average delay as ν (CBR) changes for a given $\lambda = 200$ (ABR) with r = 10 kbps.

value and the approximation is 14 % and 4 % for $r = 9$ kbps and $r = 10$ kbps, respectively.

This suggests that the ratio of time scales can serve as a guide for when the approximation can be used for the purpose of performance analysis. One can also use this ratio to see the impact of other parameters such as reserved bandwidth r or mean number of bits

$1/m$.

5.1.5 Design

Previous analysis will provide a guide to the design of a multiservice network. For the design of network, we consider following design examples.

- **ABR performance:**

  Suppose we know apriori demands of CBR and ABR sessions, and want to find ABR performance, *e.g.*, average delay and number of ABR sessions, in a randomly varying environment due to CBR sessions. We can decide how many such ABR connections are in the link and how much delay they will experience through the link based on the previous analysis. The values of the parameters are in Table 5.2, which models CBR video calls. Note the mean amount of bits per ABR connection $1/m$ = 32 kbits. If the

Parameter		Value	
	C	150	Mbps
CBR	ν	60,100	connections/hour
	μ	1	connections/hour
	b (bandwidth/CBR)	2	Mbps
ABR	λ	7200	connections/hour
	$\rho = \lambda/m$	64	kbps
	$1/m$	32	kbits/connection
	r	100	kbps

Table 5.2: Parameters for the design example supporting video.

  CBR demand is 100 conn./hour, the average number of CBR connections is 72.60 and the blocking probability is 0.274. The resulting average number of ABR connections, then, will be 0.50 and the average delay will be 0.069 msec. Varying the demand of CBR we obtain the blocking probability of CBR and the corresponding performance of ABR. If the CBR demand is lowered to 60 conn./hour, the average number of CBR connections is 59.50 and the resulting blocking probability will then be 0.0083,

i.e., less than 1 %. The corresponding average number of ABR connections will be 0.018 with average delay of 0.002 msec. Thus given the CBR characteristics, ABR arrival rate and required average bandwidth for ABR, the CBR blocking probability and ABR performance can be found.

- **Design of link capacity:**

  As another design example, the link capacity can be decided to accommodate desired CBR and ABR requirements. Suppose CBR carries voice calls having 64 kbps bandwidth and data is transmitted via ABR sessions with 4 kbits per session on average. The parameters are summarized in Table 5.3. In this scenario, the link bandwidth to guarantee the QoS (delay performance of ABR $\leq$ 5 msec) should be at least 400 kbps (see Fig. 5.8). We may further impose strict blocking probability for CBR, which will require more capacity. For example, if the blocking probability of CBR should be less than 1 %, the bandwidth requirement will be at least 800 kbps (see Fig. 5.9).

Parameter		Value	
CBR	ν	6	connections/min
	μ	1	connections/min
	b (bandwidth/CBR)	64	kbps
ABR	λ	240	connections/min
	$\rho = \lambda/m$	32	kbps
	$1/m$	8	kbits/connection
	r	40	kbps
	$\mathbb{E}[D_{ABR}]$	5	msec

Table 5.3: Parameters for the design example supporting voice calls.

5.2 Dimensioning of Multiservice Networks

Various types of traffic requiring different QoS are expected to be carried by integrated services networks. This trend is pushing technology towards connection-oriented packet-

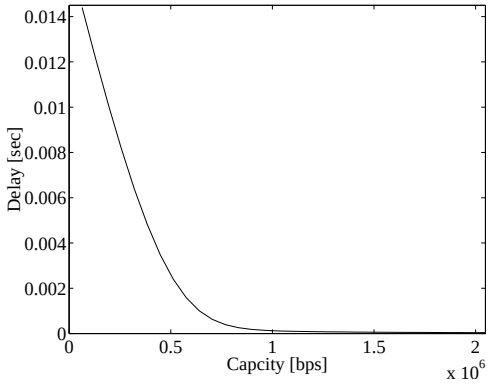

Figure 5.8: Average delay of ABR as link capacity changes.

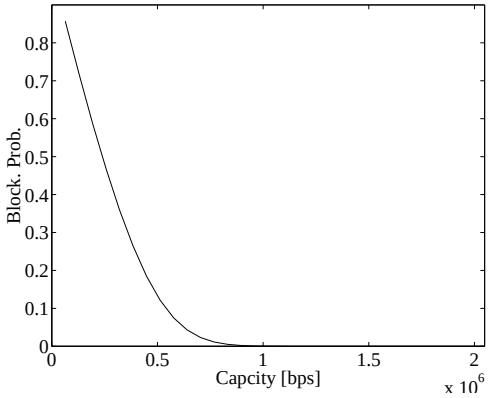

Figure 5.9: Blocking Probability of CBR as link capacity changes.

switched networks which can guarantee heterogeneous QoS to various traffic types. In some cases the available resources will change dynamically and it is important to *fairly* allocate them among contending users. In this chapter we consider approaches to dimension multiservice networks subject to overall performance constraints.

ABR service is a service type defined for ATM networks [12]. The key idea underlying ABR service is to utilize the excess bandwidth when different types of services (*e.g.*, CBR and/or VBR) are also in use. The available bandwidth for ABR traffic in a network varies dynamically depending on the presence of other service types so one needs to assess bandwidth availability and allocate it to ABR service users in an adaptive and efficient way. A network carrying CBR and/or VBR traffic can be modeled using a loss network model, *i.e.*, circuit-switched network, via the concept of *effective bandwidth* [45]. With the aid of this model, it is possible to in turn capture the characteristics of available bandwidth for ABR traffic.

The performance that will be achieved for ABR services is difficult to assess, not only because of the random environment, *i.e.*, available capacities, but also because of the resource allocation policies that are currently being considered. Indeed, the, so called, max-min fair allocation, or alternatively a revenue maximization strategy, are complex functions of the available capacity. Our approach herein is to consider the average performance one

might expect.

Suppose a set $\mathcal{S}$ of sessions share the network, where each session $s \in \mathcal{S}$ has a set of links $\mathcal{L}_s$ associated with it. The set $\mathcal{L}_s$ is intended to define an end-to-end connection through the network. More than one session might share each link, thus we let $\mathcal{S}_\ell$ be the set of sessions crossing link ℓ. Let $\vec{a} = (a_s, s \in \mathcal{S})$ be the vector of session rates and b_ℓ be the capacity of link ℓ. Consider the problem of maximizing the minimum utility of users , *i.e.,*

$$\max_{\vec{a}} \left\{ \min_{s \in \mathcal{S}} u_s(a_s) \;\middle|\; \sum_{s \in \mathcal{S}_\ell} a_s \leq b_\ell, a_s \geq 0 \right\}, \tag{5.16}$$

where $u_s(a_s)$ is a utility for a_s units of bandwidth on session s. One can solve this optimization problem to find a set of sessions with minimum utilities. After removing those sessions and adjusting capacities by those session rates, we obtain a reduced network. If we formulate the same optimization problem to this reduced network, we have a set of sessions whose utilities are the second smallest. Repeating this procedure until we exhaust all the sessions and the links, results in a hierarchy of sessions with the associated utilities resulting in session rate allocation $\vec{a}^{\,*}$. When $u_s(a_s) = a_s$, the allocation is said to be *max-min fair allocation* [38], in the sense that it maximizes the minimum throughput.

Max-min fairness can be viewed as a hierarchical optimization (allocation) problem with resource constraints. If the available bandwidth is changing as it would be on a multiservice network, it may not be easy to assess the performance of a given session. In order to find the average throughputs under the max-min fairness policy, we shall establish an upper bound on the average minimum throughput. In practice it would be more desirable to have a lower bound whence in the sequel we will show that this upper bound is achieved in large-capacity network environment where the capacities and call arrival rates are scaled.

If a goal is to maximize total utility of a network, the objective function of the bandwidth allocation policy might be the sum of the user utility functions, resulting in the following optimization problem:

$$\max_{\vec{r}} \left\{ \sum_{s \in \mathcal{S}} u_s(r_s) \;\middle|\; \sum_{s \in \mathcal{S}_\ell} r_s \leq b_\ell, r_s \geq 0 \right\}, \tag{5.17}$$

where $\vec{r} = (r_s, s \in \mathcal{S})$ denotes the vector of session rates. Solving this optimization, we obtain session rate allocation $\vec{r}^{\,*}$. By contrast with max-min fair allocations, in this case, the emphasis is on social welfare (total utility) at the expense of individual users' performance. Since max-min fairness may not maximize total throughput, one could try to improve overall throughput by introducing priorities to sessions. We consider this issue and provide a foundation to the network design and management to achieve increased performance.

The organization of this section is as follows. We consider max-min fair bandwidth allocation to ABR sessions in a multiservice network and find an upper bound on the average minimum throughput for ABR sessions in §5.2.3. We show that the upper bound is achieved in large-capacity networks in §5.2.4. The average available bandwidth for ABR sessions is given by §5.2.5. We employ circuit-switched network framework to model bandwidth availability for ABR traffic. In addition possible approaches to increase the total throughput (*i.e.,* network efficiency) are presented in §5.2.6. We summarize in §5.3.

5.2.1 ABR and CBR Services

Consider a network consisting of a set of links $\mathcal{L}$ with capacity $\vec{c} = (c_\ell, \ell \in \mathcal{L})$. Suppose a set of routes for CBR connections $\mathcal{R}$ share the network, where each route $r \in \mathcal{R}$ traverses a set of links, $\mathcal{L}_r$. More than one CBR route might share each link, thus we let $\mathcal{R}_\ell$ be the set of CBR routes crossing link ℓ. Suppose that CBR sessions $r \in \mathcal{R}$ arrive as Poisson process with rate ν_r and that the connection holding times have unit mean, *i.e.,* $\mu_r^{-1} = 1$. We assume that holding times are independent of each other and of earlier call arrival times. Suppose for simplicity that each CBR connection requires an unit of bandwidth.

We shall estimate available bandwidth for ABR traffic after CBR traffic grabs certain amount of bandwidth among link capacities. We denote $\vec{B} = (B_\ell, \ell \in \mathcal{L})$ by the available bandwidth for ABR traffic. Note that $\vec{B}$ is a random vector. We shall distinguish the available bandwidth $\vec{B}$ with the capacity $\vec{c}$. Suppose a set $\mathcal{S}$ of ABR sessions share the network, where each session $s \in \mathcal{S}$ has a set of links $\mathcal{L}_s$ associated with it. The set $\mathcal{L}_t$

is intended to define links for an end-to-end ABR connection (CBR route) t through the network. More than one ABR session may share a link, thus we let $\mathcal{S}_\ell$ be the set of ABR sessions crossing link ℓ. We assume that ABR connections are fixed as might be the case, when there exist permanent end-to-end LAN connections.

Since resources are limited incoming CBR connections may be blocked, or will affect the throughput of ABR sessions. In this context, two design criteria are reasonable, a constraint on blocking probability for CBR connections, and an average throughput requirement for ABR sessions.

5.2.2 Distribution of Number of CBR Circuits

We will use the framework of loss network model [32] to find the stationary distribution of CBR connections in the network. The stationary distribution of number of CBR connections $\pi(\vec{n})$, where $\vec{n} = (n_r, r \in \mathcal{R})$, and where n_r is number of CBR calls on route r, is given by

$$\pi(\vec{n}) = G(\vec{c}\,)^{-1} \prod_{r \in \mathcal{R}} \frac{\nu_r^{n_r}}{n_r!}, \qquad \vec{n} \in \mathcal{S}(\vec{c}\,) \tag{5.18}$$

where

$$\begin{aligned} \mathcal{S}(\vec{c}\,) &= \left\{ \vec{n} \in \mathbb{Z}_+^{|\mathcal{R}|} \mid A\vec{n} \leq \vec{c} \right\}, \\ G(\vec{c}\,) &= \left(\sum_{\vec{n} \in \mathcal{S}(\vec{c}\,)} \prod_{r \in \mathcal{R}} \frac{\nu_r^{n_r}}{n_r!} \right), \end{aligned}$$

where $A = (A_{\ell r}, \ell \in \mathcal{L}, r \in \mathcal{R})$ is a 0,1 matrix describing whether route r traverses link ℓ or not. If $A_{\ell r}$ is 1, then a route r occupies a unit circuit on link ℓ. Let $\vec{N}$ be a random variable with distribution π. The available bandwidth for ABR traffic is then given by

$$B_\ell = c_\ell - \sum_{r \in \mathcal{R}} N_r, \qquad \ell \in \mathcal{L}. \tag{5.19}$$

Note that B_ℓ is a random variable due to the dynamic N_r. We will investigate the impact of this B_ℓ on average throughput for ABR sessions subject to a given bandwidth allocation policy.

5.2.3 Average Throughput under a Bandwidth Allocation Policy

Consider max-min fair allocation policy. As mentioned previously this allocation corresponds to a hierarchical optimization problem. Let $\vec{B} = (B_\ell, \ell \in \mathcal{L})$ be the available capacity vector. A bound on the minimum throughput, when $\vec{B} = \vec{b}$, is given by

$$W(\vec{b}) = \max_{\vec{a}} \left\{ \min_{s \in \mathcal{S}} a_s \;\middle|\; \sum_{s \in \mathcal{S}_\ell} a_s \leq b_\ell, a_s \geq 0 \right\}. \tag{5.20}$$

Observing that $f(\vec{a}) = \min_{s \in \mathcal{S}} a_s$ is concave and $g_\ell(\vec{a}) = \sum_{s \in \mathcal{S}_\ell} a_s$ is convex, it follows that the minimum throughput $W(\vec{b})$ is concave in $\vec{b}$ by the Strong Duality Theorem [41]. So the upper bound on average minimum throughput is given by the following theorem.

Theorem 5.2.1 (Bound on Average Throughput)

$$\mathbb{E}[W(\vec{B})] \leq W(\mathbb{E}[\vec{B}]). \tag{5.21}$$

Proof: Since $W(\vec{b})$ is concave the result follows by Jensen's inequality. ■

A question arises whether this property still holds for the next level of hierarchy. Note, however, that the second smallest throughput may not be concave in the original capacity $\vec{b}$. This is explained by the fact that max-min fairness tries to maximize the minimum throughput $W(\vec{b})$ at the possible expense of next level throughput. This coincides the fact that max-min fairness may not maximize the overall network throughput or total utility.

5.2.4 Asymptotic Average Throughput in Loss Networks

We will show that the upper bound (5.21) on the minimum average throughput for ABR session is achieved asymptotically in large-capacity networks. Consider a sequence of networks wherein both the arrival rates $\vec{\nu}^{(n)} = (\nu_r^{(n)}, r \in \mathcal{R})$ and capacities $\vec{c}^{\,(n)} = (c_\ell^{(n)}, \ell \in \mathcal{L})$ are jointly scaled. Let $\vec{B}^{(n)} = (B_\ell^{(n)}, \ell \in \mathcal{L})$ with $B_\ell^{(n)} = c_\ell^{(n)} - \sum_{r \in \mathcal{R}_\ell} N_r^{(n)}$.

Let $\bar{x}_r = \nu_r \prod_{\ell \in \mathcal{L}_r}(1 - D_\ell)$ be the solution to the following primal optimization problem, given in [31], which determines the most likely state $\vec{\bar{x}} = (\bar{x}_r, r \in \mathcal{R})$ under the stationary probability distribution (5.18)

$$\begin{array}{ll} \text{Maximize} & \displaystyle\sum_{r \in \mathcal{R}} (x_r \log \nu_r - x_r \log x_r + x_r) \\ \text{subject to} & A\vec{x} \leq \vec{c}, \qquad \vec{x} \geq 0. \end{array} \tag{5.22}$$

Similarly let $\bar{x}_r^{(n)}$ be the solution to the optimization problem applied to the n^{th} network. The Lagrangian for (5.22) is

$$L(\vec{x}, \vec{z}; \vec{y}) = \sum_{r \in \mathcal{R}} (x_r \log \nu_r - x_r \log x_r + x_r) + \sum_{\ell \in \mathcal{L}} y_\ell \left(c_\ell - \sum_{r \in \mathcal{R}} A_{\ell r} x_r - z_\ell \right)$$

where $\vec{z}$ is a vector of slack variables and $\vec{y}$ is a vector of Lagrange multipliers. The dual problem of the primal optimization can be formulated as

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{r \in \mathcal{R}} \nu_r \exp\left(- \sum_{\ell \in \mathcal{L}} y_\ell A_{\ell r} \right) + \sum_{\ell \in \mathcal{L}} y_\ell c_\ell \\ \text{subject to} & \vec{y} \geq 0, \end{array} \tag{5.23}$$

which has the solution $\bar{y}_\ell$ where $1 - D_\ell = \exp(-\bar{y}_\ell)$. Furthermore let

$$U_r^{(n)} = n^{-1/2}(N_r^{(n)} - \bar{x}_r^{(n)}) \tag{5.24}$$

for the n^{th} network.

Suppose that the scaling satisfies the following as $n \to \infty$

$$\frac{\vec{\nu}^{(n)}}{n} \to \vec{\nu}, \qquad \frac{\vec{c}^{\,(n)}}{n} \to \vec{c}. \tag{5.25}$$

We will consider two regimes: critically loaded networks, where the difference between capacity and offered load is order $O(N^{1/2})$, and underloaded networks, where this difference is order $O(N)$.

For the critically loaded networks, it was shown by Kelly [31] that as $n \to \infty$

$$\frac{\bar{x}_r^{(n)}}{n} \to \bar{x}_r, \tag{5.26}$$

and

$$U_r^{(n)} \to U_r \sim N(0, \bar{x}_r) \qquad \text{in distribution.} \tag{5.27}$$

Using these results we establish the following theorem which claims that the upper bound is asymptotically achieved.

Theorem 5.2.2 (Asymptotic Average Throughput) *Consider a sequence of networks with demands and capacities are scaled according to the assumption (5.25), then the average normalized throughput is asymptotically given by*

$$\lim_{n\to\infty} \mathbb{E}\left[\frac{1}{n} W(\vec{B}^{(n)})\right] = W(\vec{\gamma}),$$

where $\vec{\gamma}$ depends on the loading regime according to:

1. *underloaded case:*

$$\gamma_\ell = c_\ell - \sum_{r\in\mathcal{R}_\ell} \nu_r$$

2. *critically loaded case:*

$$\gamma_\ell = c_\ell - \sum_{r\in\mathcal{R}_\ell} \nu_r \prod_{\ell\in\mathcal{L}_r} (1 - D_\ell).$$

Proof : We shall first consider the critically loaded case. Note that

$$\lim_{n\to\infty} \mathbb{E}\left[\frac{1}{n} W(\vec{B}^{(n)})\right] \;=\; \lim_{n\to\infty} \mathbb{E}\left[W\left(\frac{\vec{B}^{(n)}}{n}\right)\right].$$

Also

$$\begin{aligned}
\lim_{n\to\infty} \frac{B_\ell^{(n)}}{n} &= \lim_{n\to\infty} \left(\frac{c_\ell^{(n)}}{n} - \frac{\sum_{r\in\mathcal{R}_\ell} N_r^{(n)}}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{c_\ell^{(n)}}{n} - \frac{\sum_{r\in\mathcal{R}_\ell} (\bar{x}_r^{(n)} + n^{1/2} U_r^{(n)})}{n}\right) && \text{by (5.24)} \\
&= c_\ell - \sum_{r\in\mathcal{R}_\ell} \bar{x}_r - \lim_{n\to\infty} \sum_{r\in\mathcal{R}_\ell} \frac{U_r^{(n)}}{n^{1/2}} && \text{by (5.25) and (5.26)} \\
&= c_\ell - \sum_{r\in\mathcal{R}_\ell} \bar{x}_r && \text{by (5.27)} \\
&= c_\ell - \sum_{r\in\mathcal{R}_\ell} \nu_r \prod_{\ell\in\mathcal{L}_r} (1 - D_\ell).
\end{aligned}$$

For underloaded networks, the blocking probabilities become asymptotically small. So it follows that

$$\lim_{n\to\infty} \mathbb{E}\left[\frac{1}{n}W^1(\vec{B}^{(n)})\right] \quad = \quad W(\vec{\gamma}).$$

■

The above asymptotic result has the following interpretation: for large n,

$$\mathbb{E}[W(\vec{B}^{(n)})] \cong W(\mathbb{E}[\vec{B}^{(n)}]).$$

In this sense we will achieve the upper bound on average minimum throughput in large-capacity networks.

5.2.5 Average Bandwidth for ABR Traffic

Since the upper bound may be useful in large capacity networks we next consider how to compute the average available bandwidth $\mathbb{E}[\vec{B}]$. Note that even in a small state space with dimension $|\mathcal{R}|$, computing $G(\vec{c}\,)$ in the stationary distribution (5.18) easily becomes intractable and so does computation of number of calls to be carried. A natural approach one might follow is to find actual carried traffic or average number of connections $\mathbb{E}[\vec{N}]$ approximately and in a manageable way. *Erlang fixed point equation* [32] provides an approach to computing the mean bandwidth availability.

Let E_ℓ denote the blocking probability on link $\ell \in \mathcal{L}$ and assume that calls on link ℓ are independent of each other and of those on other links. Let ρ_ℓ be the effective call arrival rate on link ℓ, then

$$\begin{aligned} E_\ell &= E(\rho_\ell, c_\ell) \\ &= \left(\sum_{n=0}^{c_\ell} \frac{\rho_\ell^n}{n!}\right)^{-1} \frac{\rho_\ell^{c_\ell}}{c_\ell!}, \qquad \ell \in \mathcal{L}, \end{aligned} \tag{5.28}$$

where

$$\rho_\ell = \sum_{r\in\mathcal{R}_\ell} \nu_r \prod_{m\in\mathcal{L}_r\setminus\{\ell\}} (1 - E_m), \qquad \ell \in \mathcal{L}. \tag{5.29}$$

This can be viewed as a "thinning" process where the offered traffic is thinned by $1 - E_m$ at each link $m \in \mathcal{L}_r \setminus \{\ell\}$ before being offered to link ℓ. Eqs. (5.28) and (5.29) comprise a set of fixed point equations with a unique solution [32]. The actual throughput ξ_r on route r is, then, given by

$$\xi_r = \nu_r(1 - L_r) \cong \nu_r \prod_{\ell \in \mathcal{L}_r} (1 - E_\ell), \qquad r \in \mathcal{R}, \tag{5.30}$$

where L_r is the loss probability on route r.

Given the throughput ξ_r it follows by Little's law and unit mean holding time assumption that the expected number of calls on route r will be

$$\begin{aligned} \mathbb{E}[N_r] &= \xi_r = \nu_r(1 - L_r) \\ &\cong \nu_r \prod_{\ell \in \mathcal{L}_r} (1 - E_\ell), \qquad r \in \mathcal{R}. \end{aligned} \tag{5.31}$$

Kelly shows in [31] that as the capacities and call arrival rates grow together, the Erlang fixed point equation provides an accurate estimate.

Next we determine the average bandwidth available for ABR traffic on each link. It is given by

$$\mathbb{E}[B_\ell] = c_\ell - \sum_{r \in \mathcal{R}_\ell} \mathbb{E}[N_r], \qquad \ell \in \mathcal{L} \tag{5.32}$$

where $\mathbb{E}[N_r]$ can be estimated via (5.31).

5.2.6 Bandwidth Allocation to Increase Efficiency

The max-min fair bandwidth allocation may not maximize total throughput. From the system's point of view, one may want to increase throughput by introducing priorities to ABR sessions so as to maximize the weighted minimum throughput. Consider a weighted max-min fair allocation of bandwidth, associated with

$$W_w(\vec{b}) = \max_{\vec{a}} \left\{ \min_{s \in \mathcal{S}} a_s / w_s \;\middle|\; \sum_{s \in \mathcal{S}_\ell} a_s \le b_\ell, \quad a_s \ge 0, \quad w_s \ge 1 \right\}, \tag{5.33}$$

where w_s is the priority of session s for the purpose of weighting session s. Noting $f(\vec{a}) = \min_{s\in\mathcal{S}} a_s/w_s$ is concave, $g_\ell(\vec{a}) = \sum_{s\in\mathcal{S}_\ell} a_s$ is convex, and $W_w(\vec{b})$ is concave, we again have a bound on average weighted minimum throughput:

$$\mathbb{E}[W_w(\vec{B})] \le W_w(\mathbb{E}[\vec{B}]). \tag{5.34}$$

The upper bound can be achieved for large-capacity networks in a similar way as in Section 5.2.4.

If the main objective of network design is to maximize total utility or revenue, the bandwidth allocation is formulated as follows:

$$W_m(\vec{b}) = \max_{\vec{a}} \left\{ \sum_{s\in\mathcal{S}} u_s(a_s) \;\middle|\; \sum_{s\in\mathcal{S}_\ell} a_s \le b_\ell, a_s \ge 0 \right\}, \tag{5.35}$$

where $u_s(\cdot)$ is a concave utility function for each session s. Note that $f(\vec{a}) = \sum_{s\in\mathcal{S}} u_s(a_s)$ is concave, $g_\ell(\vec{a}) = \sum_{s\in\mathcal{S}_\ell} a_s$ is convex, and $W_m(\vec{b})$ is concave. So the average overall utility is bounded by

$$\mathbb{E}[W_m(\vec{B})] \le W_m(\mathbb{E}[\vec{B}]). \tag{5.36}$$

The upper bound can also be achieved for large-capacity networks similar fashion.

5.3 Summary

We have considered a single link model supporting dynamic multiservice (CBR/ABR) in ATM networks. By formulating a two-dimensional Markov chain and solving matrix-geometric equation, we have decided throughput (delay for ABR service). The joint distribution of ABR and CBR sessions has been shown to be separated as product terms when both services operate in a different time scale, and the approximation is derived for the simple analysis of network throughput. We have presented design examples (video and voice services) incorporating the analysis result. It has been shown to provide fundamentals of network performance analysis and design for multiservice ATM networks.

We have also considered techniques for network dimensioning for static ABR traffic when other types of dynamic traffic (*e.g.*, CBR) are ongoing. The average minimum

throughput for ABR sessions under max-min fair allocation is shown to be bounded above by the minimum throughput obtained from the mean available bandwidth. Moreover we have shown that the upper bound is achievable when we consider large-capacity networks. The mean available bandwidth is computed from the loss network framework.

In terms of network design, max-min fairness may not be an appropriate policy from the network provider's point of view, since it only manages that minimum throughput is maximized from each individual user's point of view. As an approach to increase total performance, one might introduce priorities or utility functions for ABR sessions so that the total throughput is improved or total revenue is maximized. In the case of max-min fair allocation with priority, the goal would be two-fold: guaranteeing the individual user's performance in the form of weighted fair allocation while achieving higher overall performance within a network.

Chapter 6

Conclusions and Future Work

In this chapter, we summarize our conclusions and insights on the work we have considered throughout the dissertation and present future research directions.

6.1 Summary of Results

Flow Control of Networks Supporting Rate-adaptive Services

- Adaptive services using available bandwidth require flow control mechanisms enabling efficient utilization of network resources and fair allocation of bandwidth.
- We have considered a flow control algorithm for adaptive services, in which rate allocation achieves a notion of *max-min fair* allocation of bandwidth. It provides fair allocation of available capacities to contending connections. It is advantageous in large-scale networks in the sense that it is *simple* and operates in a *decentralized* manner. We have shown that rate allocation converges to the max-min fair allocation of bandwidth both for synchronous and asynchronous implementations.
- Priorities to connections can be given in *weighted max-min fair* allocation of bandwidth. Network providers can differentiate users using rate-adaptive service by weights (priorities), which will impact on performance and design of networks supporting the

service.

Modeling and Stability of Dynamic Networks Supporting Services with Flow Control

- Dynamic networks supporting adaptive services under fair rate allocation mechanisms can be modeled by Markov chains. Based on this dynamic connection level model, one can understand the macroscopic behavior of adaptive services such as TCP in the Internet and ABR service in ATM networks.

- Using a piecewise-linear and quadratic Lyapunov functions we have shown the stability of networks subject to (weighted) max-min and proportionally fair bandwidth allocation policies, respectively. A natural stability condition is required: The total load on each link should not exceed the link capacity.

- Internet (TCP/IP) traffic is believed to be crudely captured by this type of dynamic model with reasonable assumptions. We suggest that congestion phenomena on the Internet might be due to connection-level instabilities. Moreover since the routing mechanism in the Internet is not aware of connection level load and there are network-level interactions, one can not solve the problem without judicious overprovisioning.

Performance and Design of Dynamic Networks Supporting Services with Flow Control

- It is difficult to characterize the performance of dynamic networks due to global interactions arising from dynamic rate allocation mechanisms. In order to observe the realistic impact on the performance, extensive simulations were conducted.

- Connections traversing a larger number of hops have more adverse effect on overall performance, which is aggravated as demand and network size grow. Moreover networks under proportionally fair rate allocation may experience poor performance since the policy tends to deemphasize long-path connections which consume more

network resources. A weighted max-min fair rate allocation can provide flexibility in bandwidth allocation over max-min fair rate allocation. Weights can be selected to improve performance although overall and individual performance may not be compatible with each other.

- We have shown that dynamic networks under fair allocations of bandwidth may not minimize the overall or individual connection delays. In this sense, a question arises as to whether max-min/proportionally fair rate allocation is an appropriate bandwidth allocation mechanism in terms of delay performance. One might want to evaluate such bandwidth allocation mechanisms or new mechanisms based on the average connection delays that are experienced rather than the instantaneous throughput, see *e.g.,* [42].

- We have proposed a design method for dynamic networks supporting GPS nodes, which guarantees average connection level delay to connections on fixed routes. Our design method can provide a basis to the design of VP networks to guarantee delay requirements.

Performance and Design of Multiservice Networks

- Multiservice (dynamic CBR and ABR connections) in a single link can be modeled by a two-dimensional Markov chain. This model provides a stepping-stone to derive the performance of such services, *i.e.,* available bandwidth and average delay for ABR connections.

- When both ABR and CBR connections are dynamic, performance of ABR is derived analytically by solving matrix-geometric equations in a single link. Approximation can also be computed assuming they operate in different time scales, *i.e.,* ABR connections are operating faster than CBR connections. Given CBR demand and/or QoS (delay) requirement of ABR service, a network can be designed based on the performance result.

- In an attempt to understand dynamic networks with static ABR and dynamic CBR connections, we derive an upper bound on the average minimum throughput ABR connections would achieve. Using asymptotics, we showed that the bound becomes tight as network size and demand become large. Thus multiservice networks could be dimensioned based on this bound as the demands/capacities become large.

6.2 Future Work

For various flow control mechanisms achieving weighted max-min or proportionally fair allocation of bandwidth, it would be desirable to understand how to assign weights to connections according to various performance goals. It is also interesting to understand how network domains with possibly different fair bandwidth allocation policies would interact with one another and how overall performance would be affected.

Simulations of the performance of networks supporting dynamic connections with "proportionally" fair allocation of bandwidth require an extensive amount of computation. When a new event occurs, *i.e.*, the arrival of a new connection or the departure of an existing connection, *constrained optimization* needs to be solved for a proportionally fair rate allocation. The number of events to be simulated and constraints in the optimization grows exponentially with the size of network. Nevertheless, unless better analytic tools are developed, simulations for arbitrary large-scale networks are essential since they alone can be used to estimate network performance.

Our dynamic network model assumes that the amount of work connections bring in is exponentially distributed. Simulations on connections with arbitrary distribution would provide further understanding of the impacts of various traffic characteristics on the performance. We conjecture that connections with heavy-tail distribution stay and grab network resources longer, which results in further degradation of performance.

Performance bounds can provide a basis for designing dynamic networks, see *e.g.*, [35, 34, 11, 10] for methods to establish performance bounds in some queueing networks. In this sense, design based on tight bounds, especially upper bounds on delay performance,

would lead to more efficient utilization of network resources. Finally, it is our hope that the model can be further improved as a network design tool for network designers/operators, which provides guidelines to design future networks.

Bibliography

[1] S. P. Abraham and A. Kumar. A stochastic approximation approach for max-min fair adaptive rate control of ABR sessions with MCRs. In *Proc. IEEE INFOCOM*, 1998.

[2] O. Ait-Hellal and E. Altman. Perfroamnce evaluation of congestion phenomena in the rate based flow control mechanism for ABR. In *Proc. IEEE INFOCOM*, 1999.

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

[4] F. M. Anjum and L. Tassiulas. Fair bandwidth sharing among adaptive and non adaptive flows in the Internet. In *Proc. IEEE INFOCOM*, 1999.

[5] S. Asmussen. *Applied Probability and Queues*. John Wiley and Sons, 1987.

[6] L. Benmohamed, S. Dravida, P. Harshavardhana, W. Lau, and A. Mittal. Desiging IP networks with performance guarantees. *Preprint*, 1999.

[7] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching networks the case of a single congested node. *IEEE/ACM Trans. on Networking*, 1(6):693–709, Dec. 1993.

[8] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[9] D. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical methods*. Prentice Hall, 1989.

[10] D. Bertsimas, D. Gamarnik, and J. N. Tsitsiklis. Geometric bounds for stationary distributions of infinite Markov chains via Lyapunov functions. Technical Report LIDS-P-2426, Laboratory for Information and Decision Systems, MIT, Sep. 1998.

[11] D. Bertsimas, I. Ch. Paschalidis, and J. N. Tsitsiklis. Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance. *Ann. Appl. Prob.*, 4:43–75, 1994.

[12] F. Bonomi and K. W. Fendick. The rate-based flow control framework for the Available Bit Rate ATM service. *IEEE Network Mag.*, pages 25–39, Mar/Apr. 1995.

[13] F. Bonomi, D. Mitra, and J. B. Seery. Adaptive algorithms for feedback-based flow control in high-speed wide-area ATM networks. *IEEE J. Select. Areas Commun.*, 13(7):1267–83, Sept. 1995.

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

[15] D. Clark and J. Wroclawski. An approach to service allocation in the Internet. *Internet Draft draft-clark-diff-svc-alloc-00.txt, http://diffserv.lcs.mit.edu/Drafts/draft-clark-diff-svc-alloc-00.pdf*, pages 152–164, 1997.

[16] J. Crowcroft and P. Oechslin. Differentiated end-to-end internet services using a weighted proportional fair sharing TCP. *Computer Communication Review*, 28(3):53–67, 1998.

[17] G. de Veciana, T.-J. Lee, and T. Konstantopoulos. Stability and performance analysis of networks supporting services with rate control – could the Internet be unstable? In *Proc. IEEE INFOCOM*, 1999.

[18] D. Down and S. P. Meyn. Piecewise linear test functions for stability and instability of queueing networks. *Queueing Systems*, 27:205–226, 1997.

[19] G. Fayolle, V. A. Mayshev, and M. V. Menshikov. *Topics in Constructive Theory of Countable Markov Chains*. Cambridge University Press, Cambridge and New York, 1995.

[20] S. Floyd. Connections with multiple congested gateways in packet-switched networks, Part 1: One-way traffic. *Computer Communication Review*, 21(5):30–47, 1991.

[21] C. Fulton, S.-Q. Li, and C. S. Lim. An ABR feedback control scheme with tracking. In *Proc. IEEE INFOCOM*, pages 806–815, 1997.

[22] C. Fulton, S.-Q. Li, and C. S. Lim. UT: ABR feedback control with tracking. *ATM Forum, 97-0239*, 1997.

[23] E. M. Gafni and D. Bertsekas. Dynamic control of session input rates in communication networks. *IEEE Trans. Automatic Control*, 29:1009–16, 1984.

[24] ATM Forum's Traffic Management Working Group. ATM forum traffic management specification version 4.0. Technical report, 1995.

[25] V. Jacobson. Congestion avoidance and control. In *Proc. ACM SIGCOM*, pages 314–329, Aug. 1988.

[26] V. Jacobson and M. Karels. Congestion avoidance and control. In *Proc. ACM SIGCOM*, pages 314–29, Aug. 1988.

[27] R. Jain, S. Kalyanaraman, and R. Goyal. Simulation results for ERICA switch algorithm with VBR + ABR traffic. *ATM Forum, 95-0467*, Apr. 1995.

[28] R. Jain, S. Kalyanaraman, and R. Viswanathan. The OSU scheme for congestion avoidance using explicit rate indication. *ATM Forum, 94-0883*, Sep. 1994.

[29] R. Jain, S. Kalyanaraman, and R. Viswanathan. Simulation results : The EPRCA+ scheme. *ATM Forum, 94-0988*, Oct. 1994.

[30] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[31] F. P. Kelly. Blocking probabilities in large circuit switched networks. *Ann. Appl. Prob.*, 18:473–505, 1986.

[32] F. P. Kelly. Loss networks. *Ann. Appl. Prob.*, 1:317–378, 1991.

[33] F. P. Kelly, A. K. Mauloo, and D. K. H. Tan. Rate control in communication networks shadow prices, proportional fairness and stability. *Journal Oper. Res. Soc.*, 15(49):237–55, 1998.

[34] P. R. Kumar and S. P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. Auto. Control*, 40(2):251–261, 1995.

[35] S. Kumar and P. P. Kumar. Performance bounds for queueing networks and scheduling policies. *IEEE Trans. Automatic Control*, 39:1600–11, 1994.

[36] V. P. Kumar, T. V. Lakshman, and D. Stiliadis. Beyond best effort: Router architectures for the differentiated services of tomorrow's internet. *IEEE Comm. Magazine*, 36(5):152–164, 1998.

[37] T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. on Networking*, 5(3):336–350, Dec. 1997.

[38] T.-J. Lee and G. de Veciana. A decentralized framework to achieve max-min fair bandwidth allocation for ATM networks. In *Proc. IEEE GLOBECOM*, 1998.

[39] J. Liebeherr, I. F. Akyildiz, and A. Tai. Multi-level explicit rate control scheme for ABR traffic with heterogeneous service. In *Proc. IEEE ICDCS*, May 1996.

[40] T.-L. Ling and N. Shroff. Novel flow control mechanism for ABR traffic in ATM networks. In *Proc. IEEE ICC*, Jun. 1997.

[41] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison Wesley, 1989.

[42] L. Massoulié and J. Roberts. Bandwidth sharing: Objectives and algorithms. In *Proc. IEEE INFOCOM*, 1999.

[43] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3):67–82, 1997.

[44] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Springer-Verlag, Berlin, 1993.

[45] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan. ATM network design and optimization: A multirate loss network framework. *IEEE/ACM Trans. on Networking*, 4(4):531–543, Aug. 1996.

[46] J. Mo, R. J. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and Vegas. In *Proc. IEEE INFOCOM*, 1999.

[47] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. The Johns Hopkins University Press, 1981.

[48] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the Internet. *Internet Draft draft-nichols-diff-svc-arch-00.txt, http://diffserv.lcs.mit.edu/Drafts/draft-nichols-diff-svc-arch-00.pdf,* pages 152–164, 1997.

[49] H. Ohsaki, M. Murata, and H. Miyahara. Robustness of rate-based congestion control algorithm for ABR service class in ATM networks. In *Proc. IEEE GLOBECOM*, Nov. 1996.

[50] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara. Rate-based congestion control for ATM networks. *ACM SIGCOM/ Computer Commun. Rev.*, pages 60–72, 1995.

[51] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proc. of ACM SIGCOM*, pages 303–314, 1998.

[52] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Trans. Networking*, 1(3):344–57, 1993.

[53] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Trans. Networking*, July 1994.

[54] V. Paxson and S. Floyd. Why we don't know how to simulate the Internet. In *Proc. of 1997 Winter Simulation Conference*, pages 1037–1044, 1997.

[55] S. Shenker. Fundamental design issues for the future internet. *IEEE J. Select. Areas Commun.*, 13(7):1176–88, Sept. 1995.

[56] K.-Y. Siu and H.-Y. Tzeng. Adaptive proportional rate control for ABR service in ATM networks. Technical report, ECE Dept. U.C. Irvine, Jul. 1994.

[57] C.-F. Su, G. de Veciana, and J. Walrand. Explicit rate flow control for ABR services in ATM networks. *IEEE/ACM Trans. Networking*, 8(3):350–361, Jun. 2000.

[58] Network Wizards. Internet domain survey. *http://www.nw.com*, 2000.

[59] Y. Zhao and S.-Q. Li. Feedback control of multiloop ABR traffic transmission. In *Proc. IEEE ICC*, Jun. 1996.

Vita

Tae-Jin Lee was born in Yongin, Korea on July 4, 1966, the son of Chae-Ok Lee and Kyo-Jung Lee. He graduated from Dong-Sung High School, Seoul, Korea in February 1985 and received his B.S. degree in Electronics Engineering at Yonsei University, Seoul, Korea in February 1989. He began his graduate studies and completed his M.S. degree in Electronics Engineering at Yonsei University in February 1991 and served military duty as a second-lieutenant in the Republic of Korean Army from August 1991 to February 1992. He then received his M.S.E. degree in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, MI, in December 1995. He entered the Department of Electrical and Computer Engineering at the University of Texas at Austin in January 1996. He married Younsuk Kim in December 1998 and finished his Ph.D. in May 1999.

Permanent Address: 409-17 Hwagok8-Dong, Kangseo-Ku

Seoul, Korea 157-018

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin.