This is to certify that the

dissertation entitled

BOUNDED-CELL-LOSS-RATIO FLOW
CONTROL AND RELIABLE ABR MULTICAST

presented by

Wei-Kuo Liao

has been accepted towards fulfillment
of the requirements for

Doctoral degree in Computer Science
& Engineering

Major professor

Date 12/16/99

BOUNDED-CELL-LOSS-RATIO FLOW CONTROL AND RELIABLE ABR MULTICAST

By

Wei-Kuo Liao

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science and Engineering

1999

ABSTRACT

BOUNDED-CELL-LOSS-RATIO FLOW CONTROL AND RELIABLE ABR MULTICAST

By

Wei-Kuo Liao

In this dissertation, we focus on the development for *Available-Variable-Bit* (ABR) flow control in ATM networks. We give a sufficient condition of *max-min fairness* convergence for ER-based ABR flow control. The tractability of this sufficient condition is verified by deriving three switch algorithms.

Bounding the loss ratio is another critical issue in flow control design. We use a learning model called *Hedge Boosting* and recursive least-squares estimation to capture the long-range dependence. With the on-line prediction, the cell loss ratio is bounded below a given ratio by reducing the available bandwidth with a number calculated under the Gaussion process assumption.

We study how to extend error recovery for ABR multicast (one-to-many communication). We use the backward *resource management* cells for ABR flow control to carry error information, and design an error control algorithm for multicast. In most of time, the algorithm will forward the retransmission cells only to the destinations requesting those cells.

At last, we prove that with max-min fairness guarantee, for a multicast session with two link-disjoint connections to support fault tolerance, a third connection atop these two existing connections could be available for the multicast session.

ACKNOWLEDGMENTS

I would like to have my deepest gratitude to Dr. Lionel M. Ni, a mentor and my advisor, who has given me constant and strong help, advises and support during my study. My special thank is to Dr. Abdol H. Esfahanian, my project co-advisor, who kept encouraging me and elevated me to have the graph-theoretic thinking. Dr. Raoul LePage, a member of my Ph.D. program committee, always inspired me by his special and precious view on various probability models. I am very lucky to have him as my committee member. I am also grateful to Dr. Matt W. Mutka, one of my committee members, as well as all faculty members, staffs, and colleagues in the Department of Computer Science and Engineering in Michigan State University. Moreover, I would like to take this chance to thank my former advisor during the master study, Dr. Chung-Ta King.

From the bottom of my heart, I would like to thank my wife, Yi-Hsuan Lai. Her pleasant personality, kindness, and creative thoughts made my study easy and interesting. At last, I would like to dedicate this dissertation to my most wonderful family – my parents, sister and brother-in-law, who gave me generous and invaluable support in all respects throughout the whole study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Many real-time applications for B-ISDN, such as teleconferencing, movie broadcasting, and multi-party chattering, must use *multicast communication* (or one-to-many communication). Extensive studies, [1–8] have shown that with efficient multicast communication, both network throughput and connection setup latency [9] can be improved. While the Mbone has been put into practice for multicast communication in the Internet, how to efficiently implement multicast communication for different classes of services in B-ISDN is still an open issue, i.e., there is no consensus on the multicast routing, dimensioning, flow control, error control, and support for fault tolerance, under the current situation. We will study the development for flow control, error control, and support for fault tolerance.

Available-Bit-Rate (ABR) service has been defined in ATM forum. The aim of ABR service is to adapt and highly utilize the available bandwidth left by *Constant-Bit-Rate* (CBR) and *Variable-Bit-Rate* (VBR) traffic. The network uses closed-loop feedback control for each ABR session to carry out the *max-min* fairness and prevent users from injecting traffic which may degrade network throughput. The ABR service can be used to convey the data stream and multimedia stream [12]. In the study of ABR flow control, the *cell loss ratio* (CLR) is always very low by assumption. However, up to date, no upper bound has ever been given by any work.

A guideline of the flow control for ABR multicast (point-to-multipoint connection) has also been specified by [13]. Basically, the guideline extends the *bottleneck*

flow control [14] from the point-to-point communication for multicast. Different implementation techniques based on the specified guideline have been developed in [15–17].

Lossless data delivery for multicast communication (*reliable multicast*) is necessary for some applications, such as distributing the information of stock markets. In this thesis, we will propose a flow control scheme called bounded-cell-loss-ratio (BCLR) flow control which will guarantee the maximum *cell loss ratio* (CLR). The BCLR flow control is developed with following the standard for ABR service defined in ATM forum. With this flow control, we show how to extend the standard to support reliable multicast.

The remainder of this thesis is organized as follows: In Chapter 1, we will give a survey for the background of this study. In Chapter 2, we will develop a set of sufficient conditions for the max-min fairness convergence guarantee. In addition, we will derive a set of algorithms based on these rules. In Chapter 3, the bounded cell loss ratio flow control is described. The error control scheme for the multicast will be discussed in Chapter 4. Chapter 5 gives the flow control for a multicast scheme called *dispersity multicast*, which will provide the fault-tolerance property. We draw the conclusion in Chapter 6.

Chapter 1

Background

In this chapter, we first consider the traffic modeling and the prediction using the corresponding model. We show the prediction error process can be treated as i.i.d. process even when the underlying process has long range dependence. Second, we do the literature review for the loss probability when the queue length process behaves asymptotically as Brownian motion process with negative linear drift in [18]. We give a brief background for ER-based flow control. A graph model is proposed and based on this model, the ER-based multicast is described. Finally, we survey some criteria for reliable multicast.

1.1 Traffic Modeling and Prediction

The traffic characteristics in the modern networks appear to be self-similar or long-range dependent [19, 20]. To explore the characteristics, from the viewpoint of modeling, the fractional Brownian motion (FBM) and fractional autoregressive

integrated moving average model (FARIMA), etc. have been proposed. A variety of actions, such as control signal tracking or multi-step predictions, can be performed based on these models. In [21], a bibliographical survey for the related works for probability models and estimations on self-similar process are given.

We select the FARIMA model to model the traffic due to more freedom to explore the short range dependence. A process W_t is said to be FARIMA(p, d, q) if it is the solution of the following equation

$$\theta(B)\nabla^d W_t = \phi(B)Z_t,$$

where B is the backward shift operator, $\nabla^d := (I - B)^d$, $d \in (-1, 1)$ is the degree of this FARIMA process, p and q are the orders for the polynomials $\theta(z)$ and $\phi(z)$, respectively. The binomial expansion of $(1 - B)^d$ is as follows:

$$(1 - B)^d = \sum_{j=0}^{\infty} \pi_j(d) B^j.$$

where the coefficients $\pi_j(d)$ is

$$\pi_j(d) \;=\; \prod_{k=1}^{j} \frac{k - d - 1}{k}.$$

The process Z_t is an i.i.d. random process with zero mean. It is known that when $d \in (-0.5, 0.5)$, W_t has the stationary solution [22] and has autocorrelation function

$\rho(h)$ with

$$\rho(h) \sim Ch^{2d-1} \qquad \text{as } h \to \infty,$$

where C is some constant, h is the time lag between two random variables and two functions f and g have the relation $f(x) \sim g(x)$ as $x \to \infty$ if $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.

Suppose we have estimators $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_p$, $\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_q$, for the coefficients $\theta_1, \theta_2, ..., \theta_p$, $\phi_1, \phi_2, ..., \phi_q$, respectively and we have the degree d. For the 1-step prediction, we consider the innovation predictor

$$\hat{U}_{t+1} := \sum_{j=1}^{p} \hat{\theta}_j U_{t-j} + \sum_{j=1}^{q} \hat{\phi}_j \hat{Z}_{t-j},$$

where the prediction error $\hat{Z}_t = U_t - \hat{U}_t$, $U_t = \nabla^d W_t$, and then we define

$$\hat{W}_{t+1} = \hat{U}_{t+1} - \sum_{j=0}^{\infty} \pi_{j+1} W_{t-j}. \tag{1.1}$$

Note that

$$W_{t+1} - \hat{W}_{t+1} \;=\; U_{t+1} - \hat{U}_{t+1} = \hat{Z}_t.$$

In addition, if $\forall i \in \{1, 2, ...p\}, j \in \{1, 2, ..., q\}$, $\hat{\theta}_i = \theta_i$ and $\hat{\phi}_j = \phi_j$, then $\{\hat{Z}_t\} = \{Z_t\}$, which is an i.i.d. process.

1.2 Loss Probability Approximation

Currently a large body of literatures investigate the loss ratio where the service rate is deterministic and the arrival process has long memory property (e.g., see [18,23–25]). These studies focus on how to guarantee the maximum CLR for VBR traffic. Our work is based on the result obtained in [18] which uses the technique *Extreme Value Theory* for the Gaussian process with negative linear drift. Up to date, there is no identification that the aggregated traffic, either the arrival process or the departure process for a buffer in the switch, will converge to a continuous-time Gaussian process[1]. We will empirically show how close the loss ratio in [18] can be under Gaussian process assumption when the ER-based flow control is considered.

Consider the queuing model as shown in Figure 1.1, where A_t and S_t is the arrival fluid process and service fluid process, respectively, and $\mathbb{E}S_t - \mathbb{E}A_t = \kappa t$, where kappa κ is a positive constant. Consider the process $X_t := A_t - S_t$. Then

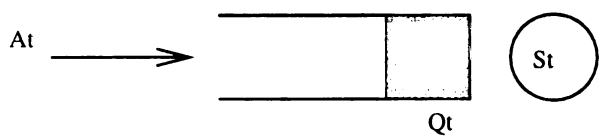


Figure 1.1: The queuing model.

[1]In [26], a large amount of independent binary renewal processes has been proved to converge to a fractional Brownian Motion. However, the convergence takes place when the number of processes and the time lag go to infinity. In [18], the author mentioned that the arrival process is Gaussian process by their empirical studies.

the queuing process Q_t can be expressed as follows:

$$Q_t = \sup_{0 \le s \le t} (X_t - X_s)$$

Suppose the process $\{X_t\}$ has stationary increments, i.e., the distribution $X_{t+\tau} - X_t$ depends only on τ, then

$$\begin{aligned} \mathbb{P}(Q_t > x) &= \mathbb{P}\left(\sup_{0 \le s \le t} (X_t - X_s) > x \right) \\ &= \mathbb{P}\left(\sup_{-t \le s \le 0} (X_t - X_s) > x \right) \\ &\to \mathbb{P}\left(\sup_{s \ge 0} (X_0 - X_{-s}) > x \right) \text{ as } t \to \infty, \end{aligned}$$

where the second equality is because X_t has stationary increments, and the convergence is due to monotone convergence theorem (see [18] for detail).

Suppose $X_0 - X_{-s} = \sigma B_s - \kappa s$ where $\sigma > 0$ and $\{B_s\}$ is standard Brownian motion process with $B_0 \equiv 0$. Then

$$\mathbb{P}(Q_t > x) = \mathbb{P}(\sup_{s>0} \sigma B_s - \kappa s > x) = \exp\left[-\frac{2\kappa x}{\sigma^2} \right],$$

(see page 190 in [27]). Indeed, if $\{X_t\}$ is a Gaussian process, i.e., for each finite sequence $t_1, t_2, ...t_n$, $(X_{t_1}, X_{t_2}, ..., X_{t_n})$ has multi-variate normal distribution, $Var(X_0 - X_{-t}) \sim \sigma^2 t$ as $t \to \infty$ and $\mathbb{E}(X_0 - X_{-t}) = -\kappa t$, then under some condi-

tions,

$$\mathbb{P}(Q_t > x) \approx \exp\left[-\frac{2\kappa x}{\sigma^2}\right], \tag{1.2}$$

as $x \to \infty$ (e.g., see (29) in [18]). The approximation (1.2) uses the inequality as follows:

$$\mathbb{P}(\sup_{s>0} X_0 - X_{-s} > x) \geq \sup_{s>0} \mathbb{P}(X_0 - X_{-s} > x).$$

In addition, let $t_x \in \arg\sup_{s>0} \mathbb{P}(X_0 - X_{-s} > x)$, then

$$t_x \sim \frac{x}{\kappa} \text{ as } x \to \infty, \tag{1.3}$$

(see Proposition 1 in [18]).

Furthermore, we have the following proposition:

Proposition 1 *If* $\{X_t\}$ *and* $\{X'_t\}$ *are Gaussian processes with continuous path almost surely and* $\forall h \geq 0, \forall t, \mathbb{E}(X'_{t+h} - X'_t) \leq \mathbb{E}(X_{t+h} - X_t)$, *and* $\forall t, s, Cov(X'_t, X'_s) = Cov(X_t, X_s)$, *then*

$$\mathbb{P}(\sup_{s>0} X_0 - X_{-s} > x) \geq \mathbb{P}(\sup_{s>0} X'_0 - X'_{-s} > x).$$

Sketch of the proof:

Let Z_t be the Gaussian process with above covariance and mean zero. Let $\tilde{X}_t :=$

$Z_t + \mathbb{E}X_t$ and $\tilde{X}'_t := Z_t + \mathbb{E}X'_t$. Thus $\sup_{s>0} \tilde{X}_0 - \tilde{X}_{-s} \geq \sup_{s>0} \tilde{X}'_0 - \tilde{X}'_{-s}$ everywhere. Then $\tilde{X}_t$ and $\tilde{X}'_t$ have the same distributions of X_t and X'_t, respectively. Therefore, $\sup_{s>0} \tilde{X}_0 - \tilde{X}_{-s} = \sup_{s>0} X_0 - X_{-s}$ in distribution and $\sup_{s>0} \tilde{X}_0 - \tilde{X}_{-s} = \sup_{s>0} X_0 - X_{-s}$ in distribution. The proposition follows. □

1.3 Max-Min Fairness and ER-Based ABR Flow Control

A main criterion of ER-Based ABR flow control is max-min fairness. In this section, we will discuss the definition of max-min fairness, the ER-based ABR flow control, a survey of the literatures on convergence of max-min fairness, and assumptions on ABR flow control we will use for the bounded cell loss ratio flow control.

1.3.1 Fairness Criteria and Definition of Max-Min Fairness

The definitions for discussing the max-min fairness are in Table 1.1. The major

parameter	description
$\mathcal{L}$	the set of links in the network
Ψ'	the set of existing connections in the network
$\mathcal{L}_s$	the set of links used by connection s
Ψ_l	the set of connections using link l
u_s	minimum cell rate (MCR), which is non-negative, for connection s
K_l	the static available bandwidth of link l
η_l	a non-negative parameter associated with the link l

*$\sum_{s \in \Psi_l} u_s \leq K_l$.

Table 1.1: The definitions of parameters.

fairness criteria with minimum cell rate (MCR) guarantee specified in [13] can be

interpreted by η_l as in Table 1.2. The hardest fairness criterion to achieve will be

fairness criterion	The bandwidth of link l for connection s
equal share	η_l
equal share plus MCR	$\eta_l + u_s$
maximum of MCR and equal share	$\max\{\eta_l, u_s\}$

*There are three other weighted fairness criteria defined in [13]. We leave the study of weighted fairness criteria as future work.

Table 1.2: The fairness criteria.

the maximum of MCR and equal share, and we will focus on how to fulfill this fairness criterion. In this dissertation, we have the uniform assumption on fairness, i.e., each link in the network is associated with the same fairness criterion. Let the sending rate for the connection s, denote by a_s, be as follows:

$$a_s = \max\left\{u_s, \min_{l \in \mathcal{L}_s} \eta_l\right\}. \tag{1.4}$$

Equation (1.4) will be referred to as the flow assignment equation. The *max-min fairness* is then an assignment of $\eta_l, \forall l \in \mathcal{L}$, such that $\forall l \in \mathcal{L}$, $\sum_{i \in \Psi_l} a_i \leq K_l$, i.e., *feasibility constraint*, and every connection has a *bottleneck link*, namely, $\forall s \in \Psi'$, there exists a bottleneck link $l \in \mathcal{L}_s$ such that $\sum_{i \in \Psi_l} a_i = K_l$ and $a_s = \max\{u_s, \eta_l\}$, i.e., *bottleneck constraint*. Such an assignment of $\eta_l, \forall l \in \mathcal{L}$, is referred to as *an assignment resulting in max-min fairness flow.*

Example 1 *Figure 1.2 shows the example network. The link set* $\mathcal{L} = \{l_1, l_2, l_3\}$. *The connection set* $\Psi' = \{s_1, s_2, s_3\}$. *We have the link parameters* η_1, η_2, η_3 *associated with links* l_1, l_2, l_3, *respectively. The capacity of* l_1, l_2, l_3 *are* $15, 10, 2$, *respectively. The MCRs for the connections* s_1, s_2, s_3 *are* $2, 0, 5$. *Figure 1.3 shows*

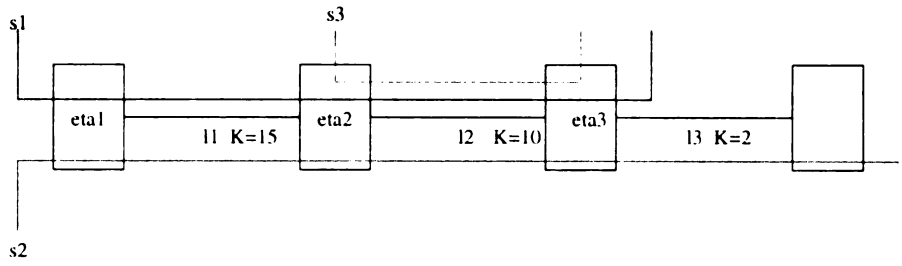


Figure 1.2: The example network.

the assignment to achieve max-min fairness. The max-min flow $(a_{s_1}, a_{s_2}, a_{s_3})$

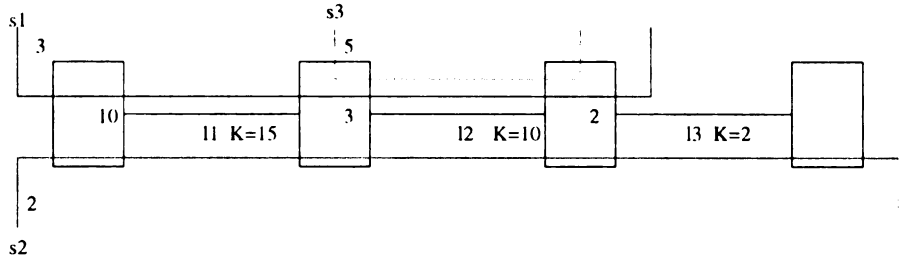


Figure 1.3: The assignment.

is $(3, 2, 5)$. *A possible assignment* (η_1, η_2, η_3) *resulting in max-min fairness flow is* $(x, 3, 2)$, *where* $x \geq 3$. *The satisfaction of feasibility constraint is easy to check. For the bottleneck constraint, the link* l_2 *is the bottleneck link for connections* s_1 *and* s_3 *and the link* l_3 *is the bottleneck link for connection* s_2.

1.3.2 Determining a_s in ABR Flow Control

In ER-based ABR flow control defined by ATM forum, *resource management* (RM) cells are used to convey the control information. We will focus on a special field in RM cell called ER. Whenever there is a fixed number of data cells being sent, an RM cell will be sent from the source. Initially, ER in the RM cell will be set to be *peak cell rate* (PCR). On receiving an RM cell, the destination will adjust the field ER in the cell according to its maximum allowed receiving capacity and

then return to the source along the previous path in the backward direction. The RM cell sent from the source to the destination is in the forward direction and will be called *FRM cell.* Reversely, the RM cell will be in the backward direction and called *BRM cell.* On receiving the BRM cell coming from the link l, the switch will update the field ER in the cell by $\min\{\text{ER}, \eta_l\}$. The BRM cell will then be passed up to the source. Therefore, as the source receives the BRM cell, the field ER in the cell already contains the minimum of all feedback rates, i.e., for connection s, $\text{ER} = \min_{l \in \mathcal{L}_s} \eta_l$. The source then adjusts its sending rate according to the field ER and its MCR. In mathematical formulation, let the η_l be time varying, thus we denote η_l at time t by $\eta_l(t)$. Therefore, for the connection s, the sending rate a_s at time t, denoted by $a_s(t)$, is determined as follows:

$$a_s(t) = \max\{u_s, \min_{l \in \mathcal{L}_s} \eta_l(t - \tau_l^s(t))\},$$

where $\forall l \in \mathcal{L}_s \forall t \geq 0, \tau_l^s(t) > 0$.

1.3.3 Survey of Literatures on Max-Min Fairness Convergence

Suppose when determining η_l for each link l, we only have the information Ψ_l, and $r_s(t), u_s, \forall s \in \Psi_l$ as well as the time-varying available link capacity $K_l(t)$ at time t, where $r_s(t)$ is the arrival rate to the link l at time t for the traffic belonging

to connection s.[2] In what follows, we will drop the subscript l if there is no ambiguous. Notice that $r_s(t) = a_s(t - \tau(t))$, where $\forall t \geq 0, \tau(t) > 0$ and thus $r_s(t)$ is time varying. To determine each link parameter η, i.e, each coordinate of the assignment resulting in max-min fairness flow, in [28], Abraham and Kumar consider the following update rule:

$$\eta(t+1) \leftarrow \eta(t) + \alpha(t)\frac{K(t) - R(t)}{|\Psi|},$$

where $\{\alpha(t)\}$ is any sequence which is squared summable, but not summable, and $R(t) := \sum_{i \in \Psi} r_i(t)$. When $\{K(t)\}$ is a bounded i.i.d. process, then the max-min fairness is proved (to be fulfilled and the total utilization will converge to $\mathbb{E}K(t)$ (Theorem IV.1 in [28]. To implement their rule, as stated in Section V.B of [28], the sequence $\{\alpha(t)\}$ needs to be reset if $\mathbb{E}K(t)$ has changed. In addition, in the proof of Theorem IV.1, the updates for all the rates in each link should happen simultaneously.

In [29], Hou, Tzeng, and Panwar has proposed an algorithm (Algorithm 4 in the paper) which can converge to the max-min fairness within finite time. They use the notion *marking consistency* to calculate the *advertise rate*. We will refer their algorithm as to *HTP algorithm*. The switch algorithm needs to sort the MCRs and thus leads to high computation complexity.

[2]This problem setting is more interesting since in real networks, the parameter is preferred to be obtained in distributed way so that there is no need for extra handshaking.

In [30], they propose the update rule:

$$\eta(t+1) \leftarrow \min\left\{K(t), \frac{K(t)\eta(t)}{R(t)}\right\}.$$

The rule is very simple. However, to guarantee the max-min fairness, the interval between two consecutive updates should be greater than the maximum round trip delay (see Section 3.4 in [30]), which is considered too long for the convergence. Usually, the update interval should be as small as possible subjective to the computation limitation or whenever an FRM cell arrives, the update will be fired. Moreover, they did not prove that their update rule can obtain an assignment resulting in max-min fairness flow using the criterion "maximum of MCR and equal share" (see Table 1.2).

In [31], Tsai and Kim modified HTP algorithm to have faster max-min fairness convergence. However, the algorithm, called CPG protocol, should use an unspecified bit called "RM.state" which is not supported in current traffic management specification by ATM forum [13].

In Chapter 3, we will propose a set of general rules to check the max-min fairness guarantee. With these rules, we also derive three viable switch algorithms conforming to the ATM traffic management specification [13] with no delay information needed.

1.3.4 Assumptions

Let Ψ_l^u denote the set of sessions s using link l such that rate $r_s \geq \eta_l$. We call such a session to be *unconstrained* for link l. It is trivial to see that for each link $l \in L_s$,

$$\sum_{s \in \Psi_l^u} \max\{u_s, \min_{l' \in L_s} \eta_{l'}\} \leq \sum_{s \in \Psi_l^u} \max\{u_s, \eta_l\} \tag{1.5}$$

The inequality (1.5) simply indicates that the total feedback rate for those unconstrained sessions for link l will be bounded by a number known locally. Note that this inequality is independent of the update rule. For those constrained sessions for link l, their rates are limited by the data generating rate in the source end, receiving capacity in the destination, or the bottlenecks elsewhere. Therefore, we have the following assumption:

Assumption 1 *The rate of the constrained session for each link l is independent of change of available bandwidth for l and the rate of unconstrained session for l.*

We also have the following assumption:

Assumption 2 *The change of the total arrival rate of all constrained sessions for each link l is relatively small to the change of available bandwidth for l.*

Beside updating and passing the BRM cell, the ATM traffic management specification 4.0 [13] allows the switch generating and returning a BRM cell to the source in case that the feedback rate η_l is reduced. The operation is called backward congestion notification (BCN) by switch. In this dissertation, we assume the

switch will bypass BRM cell to the source without letting it enter into the ordinary ABR buffer. In this way, we have next assumption:

Assumption 3 *The time interval for a BRM cell traveling from a switch to the source is constant.*

In this dissertation, we assume that when a call for establishing a connection for the ABR session arrives, the network uses MCR and the network status to decide whether to establish the connection for the ABR session. Therefore, for each link l, if $i \in \Psi_l$, then u_i is known in the controller for link l.

In ATM traffic management specification 4.0 [13], the ABR session is only allowed to use the data cell with zero *cell loss priority* (CLP). For the future application of ABR service, such as hierarchically encoded video streams, ABR service must be applicable to the data cells with CLP =1. In this dissertation, we will consider how to restrict the cell loss ratio for cells with CLP = 0 while there exist cells with CLP = 1 flowing through the same buffer.

1.4 The Model

The network is modeled as the directed graph $G(V, A)$, where $V(G)$ and $A(G)$ are the vertex set and arc set, respectively. Each arc e in $A(G)$ has a head vertex $h(e)$ and a tail vertex $t(e)$. A vertex models a switch, or a source, or a destination, and an arc models a link. We write a directed graph G as $G(V, A, C)$ if necessary, where $C : A(G) \rightarrow [0, \infty)$. For each arc $e \in A(G), C(e)$ denotes the available bandwidth of that arc. A path is a sequence of distinct arcs $e_1, e_2, ..., e_s$ where $s \geq 1$ such that

$\forall j, k \in \{1, 2, ..., s\}$, $t(e_j) = h(e_k)$ if $j + 1 = k \leq s$, and $t(e_j) \neq h(e_k)$ otherwise. In addition, $\min_{e \in A(P)} C(e)$ is called ***path bandwidth*** of path P.

A rooted tree $T(V, A, C)$ rooted at r is a directed graph where $\forall v \in V(T) - \{r\}$, there exists a unique path from r to v; in addition, $\forall e \in A(T), h(e) \neq r$. A vertex v is said to be a leaf of the rooted tree T if $v \in V(T)$ and $\forall e \in A(T), t(e) \neq v$. The set of leaves of T is denoted as $L(T)$. Moreover, $\min_{e \in A(T)} C(e)$ is called *connection bandwidth* of T. An arc $e \in A(T)$ is called a *branch* of a vertex v if $t(e) = v$. The two arc-disjoint subtrees (or underlying connections) for dispersity multicasting, denoted by T_1 and T_2, are rooted trees rooted at the same vertex, $L(T_1) = L(T_2)$, and $A(T_1) \cap A(T_2) = \emptyset$.

In this dissertation, for a rooted tree T rooted at r, we restrict that r models the source, $\forall d \in L(T)$, d models a destination and $\forall v \in V(T) - L(T) - \{r\}$, v models a switch. We also restrict that $\forall i \in \{1, 2\}$, $|\{e \in A(T_i) : t(e) = r\}| = 1$, where r is the root of T_i. Note that the general definition of subtree for multicasting can be easily transformed to the subtree with the above restrictions. In this dissertation, we always let the index $i \in \{1, 2\}$.

1.5 ABR Multicast

Consider the ABR multicasting with a single connection $T(V, A, C)$. As a simplified version of the fourth algorithms in [15,16], to obtain the connection bandwidth, the source generates a forward RM (FRM) cell with the field *explicit rate* (ER) being *peak cell rate* (PCR). Upon receiving an FRM cell, the switch multicasts the cell

For every destination:
  Upon receiving an FRM cell,
    return the cell as BRM cell to source.
For every switch v**:**
  Upon receiving a BRM cell with ER from branch e,
    MER $\leftarrow \min\{$MER,ER$\}$;
    $a_e \leftarrow 1$;
    if for each branch $e', a_{e'} = 1$, then
      pass the BRM cell to source with ER$\leftarrow$
        $\min\{$MER,$\min\{C(e') : e'$ is a branch of $v\}\}$;
      MER$\leftarrow$ PCR;
      for each branch $e', a_{e'} \leftarrow 0$;
    else discard the BRM cell.

Figure 1.4: The definition of $\Lambda_1(T)$, where T is the rooted tree for backward RM (BRM) cells traversing backwards, MER, being PCR initially, is a register in the switch and a_e, being zero initially, is a binary variable associated with the branch e.

to all its branches. Then an operation $\Lambda_1(T)$, defined in Figure 1.4, is performed. Upon receiving a backward RM (BRM) cell, the source adjusts its *allowed cell rate* (ACR) according to ER in the cell.

1.6 Criteria of Reliable Multicast

The criteria of great importance in the design for reliable multicast transport protocol are fast error detection, reduction of feedback messages and retransmission traffic, and reducing the impact of the number of receivers on the sending rate. The criteria with the approaches to solve it are listed as follows:

• **fast error detection and reduction of feedback messages:**

approach: To enable fast error detection, a negative acknowledgment (NAK) will be sent back to the sender if the receiver detects an error. The receiver needs a timer for ensuring that the sender receives the NAK. The sender has no knowl-

edge about the data arrived at all the destination correctly; therefore, the sender should periodically inquire the receiver status. In addition, when the number of receivers is large, the NAK will not be sent after a random period to avoid the overflow of buffers near the sender. The analytic study in [32] indicates that the multicast transport protocol using NAK will balance the load between the sender and receivers and thus will result in better scalability.

• **reducing the number of retransmitted data:**

approach: By partitioning the set of receivers to several local groups and electing a local group server for each receiver subset, the server can only multicast retransmitted data to the receivers in the local group [33,34]. Each receiver needs to know its local group server. Another approach counts the number of receivers requesting the same packet and selectively unicasts or multicasts a retransmission [33]. The sender needs to maintain the information for each receivers.

• **reducing the effect of the number of receivers on sending rate:**

approach: In TCP, a packet loss indicates congestion and thus the sending rate should be reduced. Suppose there are $k \geq 1$ receivers and the packet error probability associated with each receiver is $p_e > 0$ and let these error probability distributions among all the receivers be independent. Therefore, for each packet, the sender will receive an NAK with probability $p_n = 1-(1-p_e)^k > 1-e^{-kp_e}$. In a large-scale multicast communication, it is possible that $kp_e \approx 1$, e.g., $k = 100, p_e = 0.01$ and thus $p_n > 0.63$. Thus, the local group size should be small. Another approach takes advantage of bandwidth reservation and thus congestion will never take place for the multicast session [35].

Chapter 2

Convergence of Max-Min Fairness for Single-ER-based Flow Control

There are a large body of literatures (e.g., [28, 36–41]) on the ER-based flow controls. Basically, they can be divided into two categories: *single-ER* (SER) flow control and *multi-ER* (MER) flow control. In SER flow control, at a time point, the feedback rate for a link will be the same for all the ABR sessions through the link. On the other hand, the feedback rate for a link in MER flow control might be different at the same time for the different ABR sessions through the link. Examples for SER flow control are NIST [39], UT [38] and EPRICA [13]. The popular flow control algorithm ERICA [40] is considered to be in the category of MER flow control. Basically, SER flow control has the advantage of simplicity and thus its properties are easy to derived. In this paper, we study the SER flow control algorithm.

In [43], they proposed a sufficient condition *marking consistency* for max-min

fairness convergence with the fairness criterion "equal share" (or called "max-min share" in page 79, [13]). Based on the marking consistency, literatures [29, 31] propose algorithms for the fairness criterion "MCR plus equal share." The advantage of the technique marking consistency is fast convergence. However, as stated in Subsection 1.3.3. extending from marking consistency appears leading to high computation complexity or inconformity to the ATM forum traffic management specification [13]. Instead of marking consistency, based on the insight in [42], we will try to set up a simple sufficient condition which has convergence guarantee for max-min fairness with the fairness criterion "maximum of MCR and equal share." We will use this sufficient condition to generate three SER flow control algorithms conforming to ATM traffic specification [13]. As the analysis and simulation show, though in general, the transient behavior of the algorithm with higher complexity will behave better, there exists a possibility for an algorithm with computation complexity $O(N)$, where N stands for number of connections via the link, but the transient behavior is close to the one with computation complexity $O(N^2)$, which is considered as the optimal one.

2.1 Centralized Algorithms

For the description of notation we will use, please refer to Table 1.1. Given a network with link set $\mathcal{L} = \{l_1, l_2, ..., l_n\}$, a set of existing connections Ψ', we can develop a simple centralized algorithm to find an assignment $\underline{\eta'} := (\eta'_{l_1}, \eta'_{l_2}, ..., \eta'_{l_n})$ resulting in max-min fairness flow. To avoid the trivial case, we have the following

assumptions:

Assumption 4 *Throughout the whole dissertation,* $\forall l \in \mathcal{L}, \exists s \in \Psi'$ *such that* $l \in \mathcal{L}_s$.

Assumption 5 *Throughout the whole dissertation,* $\forall s \in \Psi', \emptyset \neq \mathcal{L}_s \subseteq \mathcal{L}$.

Let the function $g_{c,l}$ with $l \in \mathcal{L}$ be defined as follows:

$$g_{c,l}(\eta) := \max\{\eta, u_{j^*}\} + \sum_{i \in \Psi_l - \{j^*\}} \Big(\max\{\eta, u_i\} \mathbb{I}(a_i \geq \eta) + a_i \mathbb{I}(a_i < \eta) \Big), \qquad (2.1)$$

where $j^* \in \arg\max_{i \in \Psi_l} a_i$, and the indicator function $\mathbb{I}(\cdot) = 1$ if $(\cdot)$ is true; otherwise, $\mathbb{I}(\cdot) = 0$. Moreover, by (1.4), $\forall i \in \Psi_l, 0 \leq u_i \leq a_i$.

Remark 1 *First,* $g_{c,l}(0) = \sum_{i \in \Psi_l} u_i$. *Second,* $g_{c,l}$ *is continuously increasing to infinity.*

Remark 2 *Suppose we use* η_l^* *such that* $g_{c,l}(\eta_l^*) = K_l$ *as the link parameter for link* l. *After the flow assignment, e.g, (1.4), we have* $\sum_{i \in \Psi_l} a_i \leq K_l$. *This is because for each item corresponding to an index* $i \in \Psi_l$, a_i *is less than or equal to the item.*

Lemma 1 *Let* $\Psi_l = \{1, 2, ..., k_l\}$. *Suppose that there is a sequence* $a'_1, a'_2, ..., a'_{k_l}$ *such that* $\forall j, 1 \leq j \leq k_l, a_j \leq a'_j$. *Define function* $\hat{g}_{c,l}$ *as follows:*

$$\hat{g}_{c,l}(\eta) := \max\{\eta, u_{\hat{j}^*}\} + \sum_{i \in \Psi_l - \{\hat{j}^*\}} \Big(\max\{\eta, u_i\} \mathbb{I}(a'_i \geq \eta) + a'_i \mathbb{I}(a'_i < \eta) \Big),$$

where $\hat{j}^* \in \arg\max_{i \in \Psi_l} a'_i$. *Then* $\forall \eta \in [0, \infty), g_{c,l}(\eta) \leq \hat{g}_{c,l}(\eta)$.

Proof: Since $\forall i \in \Psi_l$, if $a_i' \geq \eta$, then

$$\begin{aligned} & \max\{\eta, u_i\}\mathbb{I}(a_i' \geq \eta) + a_i'\mathbb{I}(a_i' < \eta) \\ = \; & \max\{\eta, u_i\} \\ \geq \; & \max\{\eta, u_i\}\mathbb{I}(a_i \geq \eta) + a_i\mathbb{I}(a_i < \eta); \end{aligned}$$

on the other hand, if $a_i' < \eta$, we have

$$\begin{aligned} & \max\{\eta, u_i\}\mathbb{I}(a_i' \geq \eta) + a_i'\mathbb{I}(a_i' < \eta) \\ = \; & a_i' \\ \geq \; & a_i = \max\{\eta, u_i\}\mathbb{I}(a_i \geq \eta) + a_i\mathbb{I}(a_i < \eta). \end{aligned}$$

Therefore, we have $\forall i \in \Psi_l$,

$$\max\{\eta, u_i\}\mathbb{I}(a_i' \geq \eta) + a_i'\mathbb{I}(a_i' < \eta) \geq \max\{\eta, u_i\}\mathbb{I}(a_i \geq \eta) + a_i\mathbb{I}(a_i < \eta). \qquad (2.2)$$

In addition, suppose $j^* \neq \hat{j}^*$. Notice that $a_{\hat{j}\cdot} \leq a_{j^*} \leq a_{j^*}'$. If $a_{\hat{j}\cdot} \geq \eta$, then

$$\begin{aligned} & \max\{\eta, u_{j^*}\} + \max\{\eta, u_{\hat{j}\cdot}\}\mathbb{I}(a_{\hat{j}\cdot} \geq \eta) + a_{\hat{j}\cdot}\mathbb{I}(a_{\hat{j}\cdot} < \eta) \\ = \; & \max\{\eta, u_{j^*}\} + \max\{\eta, u_{\hat{j}\cdot}\} \\ = \; & \max\{\eta, u_{\hat{j}\cdot}\} + \max\{\eta, u_{j^*}\}\mathbb{I}(a_{j^*}' \geq \eta) + a_{j^*}'\mathbb{I}(a_{j^*}' < \eta). \end{aligned}$$

If $a_{\hat{j}\cdot} < \eta \leq a_{j\cdot}$, then

$$\begin{aligned}
& \max\{\eta, u_{j\cdot}\} + \max\{\eta, u_{\hat{j}\cdot}\}\mathbb{I}(a_{\hat{j}\cdot} \geq \eta) + a_{\hat{j}\cdot}\mathbb{I}(a_{\hat{j}\cdot} < \eta) \\
= \; & \max\{\eta, u_{j\cdot}\} + a_{\hat{j}\cdot} \\
< \; & \eta + \max\{\eta, u_{j\cdot}\} \\
\leq \; & \max\{\eta, u_{\hat{j}\cdot}\} + \max\{\eta, u_{j\cdot}\}\mathbb{I}(a'_{j\cdot} \geq \eta) + a'_{j\cdot}\mathbb{I}(a'_{j\cdot} < \eta).
\end{aligned}$$

If $a_{\hat{j}\cdot} \leq a_{j\cdot} < \eta \leq a'_{j\cdot}$, then by $\forall j \in \Psi_l, 0 \leq u_j \leq a_j$,

$$\begin{aligned}
& \max\{\eta, u_{j\cdot}\} + \max\{\eta, u_{\hat{j}\cdot}\}\mathbb{I}(a_{\hat{j}\cdot} \geq \eta) + a_{\hat{j}\cdot}\mathbb{I}(a_{\hat{j}\cdot} < \eta) \\
= \; & \eta + a_{\hat{j}\cdot} \\
< \; & \eta + \eta \\
= \; & \max\{\eta, u_{\hat{j}\cdot}\} + \max\{\eta, u_{j\cdot}\}\mathbb{I}(a'_{j\cdot} \geq \eta) + a'_{j\cdot}\mathbb{I}(a'_{j\cdot} < \eta).
\end{aligned}$$

If $a_{\hat{j}\cdot} \leq a_{j\cdot} \leq a'_{j\cdot} < \eta$, then

$$\begin{aligned}
& \max\{\eta, u_{j\cdot}\} + \max\{\eta, u_{\hat{j}\cdot}\}\mathbb{I}(a_{\hat{j}\cdot} \geq \eta) + a_{\hat{j}\cdot}\mathbb{I}(a_{\hat{j}\cdot} < \eta) \\
= \; & \eta + a_{\hat{j}\cdot} \\
\leq \; & \eta + a'_{j\cdot} \\
= \; & \max\{\eta, u_{\hat{j}\cdot}\} + \max\{\eta, u_{j\cdot}\}\mathbb{I}(a'_{j\cdot} \geq \eta) + a'_{j\cdot}\mathbb{I}(a'_{j\cdot} < \eta).
\end{aligned}$$

Therefore,

$$\begin{aligned} &\max\{\eta, u_{j^*}\} + \max\{\eta, u_{\hat{j}^*}\}\mathbb{I}(a_{\hat{j}^*} \geq \eta) + a_{\hat{j}^*}\mathbb{I}(a_{\hat{j}^*} < \eta) \\ &\leq \max\{\eta, u_{\hat{j}^*}\} + \max\{\eta, u_{j^*}\}\mathbb{I}(a'_{j^*} \geq \eta) + a'_{j^*}\mathbb{I}(a'_{j^*} < \eta). \end{aligned} \tag{2.3}$$

Hence, if $j^* = \hat{j}^*$, then by inequality (2.2),

$$\begin{aligned} g_{c,l}(\eta) &= \max\{\eta, u_{j^*}\} + \sum_{i \in \Psi_l - \{j^*\}} \Big(\max\{\eta, u_i\}\mathbb{I}(a_i \geq \eta) + a_i\mathbb{I}(a_i < \eta) \Big) \\ &\leq \max\{\eta, u_{\hat{j}^*}\} + \sum_{i \in \Psi_l - \{\hat{j}^*\}} \Big(\max\{\eta, u_i\}\mathbb{I}(a'_i \geq \eta) + a'_i\mathbb{I}(a'_i < \eta) \Big) \\ &= \hat{g}_{c,l}(\eta). \end{aligned}$$

If $j^* \neq \hat{j}^*$, then

$$\begin{aligned} g_{c,l}(\eta) &= \max\{\eta, u_{j^*}\} + \sum_{i \in \Psi_l - \{j^*\}} \Big(\max\{\eta, u_i\}\mathbb{I}(a_i \geq \eta) + a_i\mathbb{I}(a_i < \eta) \Big) \\ &= \max\{\eta, u_{j^*}\} + \max\{\eta, u_{\hat{j}^*}\}\mathbb{I}(a_{\hat{j}^*} \geq \eta) + a_{\hat{j}^*}\mathbb{I}(a_{\hat{j}^*} < \eta) \\ &\quad + \sum_{i \in \Psi_l - \{j^*, \hat{j}^*\}} \Big(\max\{\eta, u_i\}\mathbb{I}(a_i \geq \eta) + a_i\mathbb{I}(a_i < \eta) \Big) \\ &\leq \max\{\eta, u_{j^*}\} + \max\{\eta, u_{\hat{j}^*}\}\mathbb{I}(a_{\hat{j}^*} \geq \eta) + a_{\hat{j}^*}\mathbb{I}(a_{\hat{j}^*} < \eta) \\ &\quad + \sum_{i \in \Psi_l - \{j^*, \hat{j}^*\}} \Big(\max\{\eta, u_i\}\mathbb{I}(a'_i \geq \eta) + a'_i\mathbb{I}(a'_i < \eta) \Big) \\ &\leq \max\{\eta, u_{\hat{j}^*}\} + \max\{\eta, u_{j^*}\}\mathbb{I}(a'_{j^*} \geq \eta) + a'_{j^*}\mathbb{I}(a'_{j^*} < \eta) \\ &\quad + \sum_{i \in \Psi_l - \{j^*, \hat{j}^*\}} \Big(\max\{\eta, u_i\}\mathbb{I}(a'_i \geq \eta) + a'_i\mathbb{I}(a'_i < \eta) \Big) \\ &= \hat{g}_{c,l}(\eta), \end{aligned}$$

where the first inequality is from (2.2) and the second inequality is from (2.3). □

Then we have the greedy[1] algorithm shown in Figure 2.1.

```
1  L̂ ← ∅; ∀s ∈ Ψ', a_s ← ∞; ∀l ∈ L, η'_l ← ∞;
2  while L ≠ L̂, do
3     for each l ∈ L − L̂, do
4        η*_l ← min{η : g_{c,l}(η) = K_l};
5     end for;
6     choose a link l̂ ∈ arg min_{l∈L−L̂} η*_l;
7     η'_l̂ ← η*_l̂;
8     for each s ∈ Ψ_l̂, do
9        a_s ← max{u_s, min_{l∈L_s} η'_l};
10    end for;
11    L̂ ← L̂ ∪ {l̂};
12 end while;
13 output η';
```

Figure 2.1: The centralized algorithm to find an assignment resulting in max-min fairness flow.

Let $g^k_{c,l}$ be the function we use in kth while-loop iteration in line 4 for link l. Moreover, let ζ_i be the $\eta_{\hat{l}}$ at line 7 in the ith while-loop iteration in Figure 2.1, where $1 \leq i \leq n$. We have the following lemma:

Lemma 2 *If* $\exists m, 1 \leq m \leq n, \zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_m$, *then* $\forall s \in \Psi', a_s$ *has been changed at most once before* $(m+1)$*th while-loop iteration.*

Proof: If $m = 1$, the assertion is trivial. Therefore, let $m > 1$. Suppose that by the end of kth while-loop iteration with $k < m$, $\forall s \in \Psi', a_s$ has been changed at most once. For those a_s being changed, i.e, $a_s < \infty$, by line 9, we have $a_s \leq \zeta_k$. Therefore, before line 9 in the $(k+1)$th while-loop iteration, $\forall s \in \Psi_{\hat{l}}$, if $a_s < \infty$,

[1]The notion greedy here does not mean that we can obtain each parameter η_l for link l independently. Instead, the notion indicates that in each step, we can obtain one coordinate of the assignment resulting in max-min fairness flow.

i.e., a_s is changed at k_sth while-loop iteration where $k_s \leq k$, then by line 9 in the $(k+1)$th iteration,

$$a_s = \max\{u_s, \min_{l \in \mathcal{L}_s} \eta'_l\} = \max\{u_s, \zeta_{k_s}\},$$

due to that $\zeta_{k_s} \leq \zeta_{k_s+1} \leq \cdots \leq \zeta_k \leq \zeta_{k+1}$ and $\forall l \in \mathcal{L}_s$, either $\exists k', k_s \leq k' \leq k, \eta'_l = \zeta_{k'}$, or $\eta'_l = \infty$. Hence, a_s will not be changed at $(k+1)$th iteration. By induction, we have the proof. □

Theorem 1 *The output $\underline{\eta'}$ obtained by the centralized algorithm in Figure 2.1 is an assignment resulting in max-min fairness flow.*

Proof: First, note that if there is an η'_l being changed, then $\forall s \in \Psi_l, a_s$ is updated at line 9, which is the flow assignment equation (1.4). Therefore, except line 7 – 10, during each while-loop iteration, $\forall s \in \Psi', a_s$ will satisfy the flow assignment equation.

Second, we claim that $\zeta_1 \leq \zeta_2 \leq ... \leq \zeta_n$. If so, by Lemma 2, $\forall s \in \Psi', a_s$ will be changed once. Notice that if l' is chosen in the kth while-loop iteration, $K_{l'} = g_{c,l'}(\eta'_{l'}) \geq \sum_{i \in \Psi_{l'}} a_i$ after kth iteration, and thus the feasibility constraint will not be violated.

To prove the claim, suppose we already have $\zeta_1 \leq \zeta_2 \leq ... \leq \zeta_{j-1}$. Therefore, by Lemma 2, a_s has been changed at most once before jth while-loop iteration. Let l' be $\hat{l}$ at line 6 at jth while-loop iteration. Let $A^k_{l'} \subseteq \Psi_{l'}$ be the set such that $\forall s \in A^k_{l'}, a_s$ is finite in line 3 during the kth while-loop iteration. Hence, $A^{j-1}_{l'} \subseteq A^j_{l'}$.

Therefore, we have $\forall \eta \in [0, \infty), g_{c,l'}^{j-1}(\eta) \geq g_{c,l'}^{j}(\eta)$ since a_s will be changed at most once, $A_{l'}^{j-1} \subseteq A_{l'}^{j}$ and by Lemma 1. (Apply $g_{c,l'}^{j-1}$ as $\hat{g}_{c,l}$ and $g_{c,l'}^{j}$ as $g_{c,l}$.) Since $g_{c,l'}^{k}$ is increasing, therefore, we have $\eta_{l',j-1}^{*} \leq \zeta_j$, where $\eta_{l',j-1}^{*}$ is the $\eta_{l'}^{*}$ obtained at line 4 during the $(j-1)$th while-loop iteration. Since ζ_{j-1} is the minimum among those $\eta_l^*, l \in \mathcal{L} - \mathcal{L}'$ in $(j-1)$th while-loop iteration, therefore, we have $\zeta_{j-1} \leq \eta_{l',j-1}^{*} \leq \zeta_j$. Thus by induction, $\zeta_1 \leq \zeta_2 \leq ... \leq \zeta_n$.

Third, if in some while-loop iteration, l' is chosen at line 6, then, if $\exists s$ such that $a_s = \infty$, then at line 11 in the same while-loop iteration, $a_s = \max\{u_s, \eta_{l'}'\}$ and $\sum_{i \in \Psi_{l'}} a_i = g_{c,l'}(\eta_{l'}') = K_{l'}$. Therefore, l' is the bottleneck link for s. Since $\forall s$, a_s will be from infinite to finite eventually by Assumption 5, therefore, $\forall s$, there exists a bottleneck link for s. □

As stated in the proof above, $\forall s \in \Psi', a_s$ will be changed once. Therefore, we can replace line 8 and 9 with the following statement:

8' for each $s \in \Psi_l$ such that $a_s = \infty$, do
9' $a_s \leftarrow \max\{u_s, \eta_l'\}$;

In addition, we can omit the initialization of η_l' at line 1.

In the centralized algorithm, we need to find the smallest η_l^* to satisfy $g_{c,l}(\eta_l^*) = K_l$ at line 4. The question will be as follows:

Can we simply choose any η_l^* such that $g_{c,l}(\eta_l^*) = K_l$?

Suppose we are at line 4 in jth while-loop iteration. We choose any η_l^* such that $g_{c,l}(\eta_l^*)$. If l is chosen in line 6 as $\hat{l}$, in the analysis, instead of using η_l^* as ζ_j,

we can use ζ_j redefined as follows:

$$\zeta_j := \max\{\eta_l^*, \zeta_{j-1}\}.$$

where $\eta_0 = 0$. Notice that if we can prove that $g_{c,l}^j(\zeta_j) \leq K_l$, then $g_{c,l}^j(\zeta_j) = K_l$ since $g_{c,l}^j$ is increasing and $g_{c,l}^j(\eta_l^*) = K_l$. We have the following lemma.

Lemma 3 *At line 7 in kth while-loop iteration with* $1 \leq k \leq n$, $\forall l \in \mathcal{L} - \hat{\mathcal{L}}^k, g_{c,l}^k(\zeta_j) \leq K_l$, *where* $\hat{\mathcal{L}}^k$ *is the* $\hat{\mathcal{L}}$ *at line 3 - 10 in kth while-loop iteration. Moreover, if* $\eta_1 \neq \eta_2$ *and* $g_{c,l}^k(\eta_1) = g_{c,l}^k(\eta_2) = K_l$, *then* $\forall s \in \Psi_l$, a_s *at line 11 in the same while-loop iteration can be in one of the following cases:*

case 1: If $s = j^*$, *then* $u_s \geq \max\{\eta_1, \eta_2\}$ *and,*

case 2: if $s \neq j^*$, *then* $u_s \geq \max\{\eta_1, \eta_2\}$, *or* $\min\{\eta_1, \eta_2\} \leq u_s = a_s \leq \max\{\eta_1, \eta_2\}$, *or* $a_s \leq \min\{\eta_1, \eta_2\}$.

Therefore, $\forall s \in \Psi_l$, a_s *will be the same when* η_1 *or* η_2 *is used as the parameter for link* l.

Proof: Suppose for some j with $1 \leq j < n$, the first statement holds true when $k = j$. Therefore, if l' is chosen as $\hat{l}$ at line 6 in $(j+1)$th while-loop iteration, in this iteration after line 5, we have $\forall l \in \mathcal{L} - \hat{\mathcal{L}}^j, g_{c,l}^j(\zeta_j) \leq K_l$ and $g_{c,l}^{j+1}(\eta_{l'}^*) \leq g_{c,l}^{j+1}(\eta_l^*) \leq K_l$ (see the third paragraph in the proof of Theorem 1). Since $g_{c,l}^{j+1}(\zeta_j) \leq g_{c,l}^j(\zeta_j) \leq K_l$ and $\zeta_{j+1} = \max\{\zeta_j, \eta_{l'}^*\}$, the first statement holds true when $k = j+1$. By induction, we prove the first statement.

For the second statement, without loss of generality, suppose $\eta_1 < \eta_2$. Consider the following cases:

For case 1, $u_s < \eta_2$ and $s = j^*$: We have

$$\max\{u_s, \eta_1\} < \eta_2 = \max\{u_s, \eta_2\};$$

for case 2: $u_s < \eta_2$, $u_s < a_s$, $a_s > \eta_1$, and $s \neq j^*$: we have

$$\begin{aligned} \max\{\eta_1, u_s\}\mathbb{I}(a_s \geq \eta_1) \quad &+ \quad a_s\mathbb{I}(a_s < \eta_1) \\ &\leq \quad \max\{\eta_1, u_s\} \\ &\leq \quad \min\{\eta_2, a_s\} \\ &= \quad \max\{\eta_2, u_s\}\mathbb{I}(a_s \geq \eta_2) + a_s\mathbb{I}(a_s < \eta_2). \end{aligned}$$

If $\exists s \in \Psi_l$ such that one of the above two cases holds, then due to that each item corresponding to an $s \in \Psi_l$ in $g_{c,l}$ is increasing with respect to η, and thus $g_{c,l}(\eta_1) < g_{c,1}(\eta_2)$. We have a contradiction.

For the third statement, if $s = j^*$, then by case 1, after the flow assignment at line 9 in the same while-loop iteration, $a_s = u_s$ by using $\eta'_l = \eta_1$ or $\eta'_l = \eta_2$. If $s \neq j^*$ and $a_s = \infty$ before line 8, then by case 1, after the flow assignment at line 9 in the same while-loop iteration, $a_s = u_s$ by using $\eta'_l = \eta_1$ or $\eta'_l = \eta_2$. If $a_s < \infty$ and $s \neq j^*$, then a_s will not be changed since $\exists l' \in \mathcal{L}_s, \eta'_{l'}$ has been defined in the previous while-loop iteration with $\eta'_{l'} \leq \eta_1$, or $a_s = u_s$. □

Therefore, in the proof of Theorem 1, the sequence $\zeta_1, \zeta_2, ..., \zeta_n$ is still increasing

and resulting in the same flow sequence as $\{a_s\}$ just before the algorithm finishing. Hence, the statement of Theorem 1 still holds. Therefore, we can replace line 4 with

$$4' \qquad \text{find } \eta_l^* \text{ satisfying } g_{c,l}(\eta_l^*) = K_l;$$

The revised centralized algorithm is shown in Figure 2.2.

```
1'  L̂ ← ∅; ∀s ∈ Ψ', a_s ← ∞;
2   while L ≠ L̂, do
3      for each l ∈ L − L̂, do
4'        find η_l* satisfying g_{c,l}(η_l*) = K_l;
5      end for;
6      choose a link l̂ ∈ arg min_{l∈L−L̂} η_l*;
7      η'_l̂ ← η_l̂*;
8'     for each s ∈ Ψ_l̂ such that a_s = ∞, do
9'        a_s ← max{u_s, η'_l̂};
10     end for;
11     L̂ ← L̂ ∪ {l̂};
12  end while;
13  output η';
```

Figure 2.2: The revised centralized algorithm to find an assignment resulting in max-min fairness flow.

To reduce the time spent in obtaining $g_{c,l}(\eta_l)$, for each link l, we keep a variable initially zero. If for some s, a_s is changed from ∞ to a finite number, then add this finite number to the variable. That is, we *mark* s for l so that in the following steps, we do not cope with s anymore except the variable kept for l. This *mark* operation will be used in some distributed algorithm.

It will be noteworthy that though a_s has been changed at most once, we cannot get η_l^* in line 4' by $\tilde{\eta}_l := \frac{K_l - \sum_{i\in\Psi_l} a_i \mathbb{I}(a_i<\infty)}{\sum_{i\in\Psi_l} \mathbb{I}(a_i=\infty)}$. This is due to that there may exist some $s \in \Psi_l$ such that $a_s = \infty, u_s > \tilde{\eta}_l$ and thus $g_{c,l}(\tilde{\eta}_l) > K_l$.

2.2 Sufficient Conditions for Max-Min Fairness Convergence

We will drop the subscript l if there exists no ambiguity. We have developed a centralized algorithm in Figure 2.2 to find an assignment $\underline{\eta'}$ resulting in max-min fairness flow in Section 2.3. In the following sections, we will focus on the development of distributed algorithms to find the assignment resulting in max-min fairness flow.

Consider the following problem:

Problem 1 *Given $K > 0$, an index set Ψ, and sequences of non-negative numbers $(r_i)_{i\in\Psi}$, $(u_i)_{i\in\Psi}$, and a subset $\{j^*\} \subseteq \Psi^u \subseteq \Psi$ with $j^* \in \arg\max_{i\in\Psi} \max\{u_i, r_i\}$, find an η^* such that*

$$g(\eta^*) = K.$$

where

$$\begin{aligned} g(\eta) := \sum_{i\in\Psi^u} \max\{\eta, u_i\} + \sum_{i\in\Psi-\Psi^u} \Big(& \max\{\eta, u_i\}\mathbb{I}(\max\{r_i, u_i\} \geq \eta) \\ & + \max\{r_i, u_i\}\mathbb{I}(\max\{r_i, u_i\} < \eta) \Big), \end{aligned} \tag{2.4}$$

***where the indicator function** $\mathbb{I}(\cdot) = 1$ **if** $(\cdot)$ **is true; otherwise,** $\mathbb{I}(\cdot) = 0$.*

Remark 3 *If we let $a_i = \max\{r_i, u_i\}$, then for link l, $\forall \eta \in [0, \infty), g(\eta) = g_{c,l}(\eta)$, where $g_{c,l}$ is defined in (2.1).*

Remark 4 *If we use the solution η^* of Problem 1 as the feedback rate and currently network is in the equilibrium, i.e., for each link, the vectors $(r_i)_{i\in\Psi}, (u_i)_{i\in\Psi}$ and the available bandwidth K are time-invariant, then $\sum_{i\in\Psi} r_i \leq K$. That is, $\forall i \in \Psi$, if $r_i > \eta^*$, then $r_i \leq \max\{u_i, \eta^*\}$ due to the flow assignment equation (1.4) and $r_i = a_i$.*

Remark 5 *The solution of Problem 1 is not unique, e.g., when $\sum_{i\in\Psi} u_i = K$, $\forall \eta \in [0, \min_{i\in\Psi} u_i]$, η is a solution.*

Remark 6 *Note that the first summand in (2.4) is continuously increasing to infinity with respect to η since $|\Psi^u| \geq 1$. The second summand in (2.4) is continuously increasing with respect to η. Therefore, $g(\eta)$ is continuously increasing to infinity. Hence, if $g(0) = \sum_{i=1}^{N} u_i \leq K$, then by the intermediate value theorem, the problem has a solution. Indeed, g is a continuously piecewise linear function.*

Remark 7 *It is usually assuming $u_s \leq r_s$. However, we will not stress this inequality throughout our analysis. In fact, in some case, $u_s > r_s$ if we refer r_s as to $\min_{l\in\mathcal{L}_s-\{l'\}} \eta_l$ when considering the arrival rate of connection s at link l'. That is, when a source sends an FRM cell, the ER field will be $\min\{PCR, C_s\}$ initially, where C_s is the maximum sending rate user currently is allowed to requested. On receiving an FRM cell belonging to connection s, the switch updates the register r_s^f by the ER field in this cell and then updates the ER field by $\min\{ER, \eta\}$. on receiving a BRM cell belonging to connection s, the switch updates the register r_s^b by the ER field in this cell and then updates the*

ER field by $\min\{ER, \eta\}$. *Then assign* $r_s = \min\{r_s^f, r_s^b\}$. *On receiving a BRM cell, the destination updates ER field in the cell by* $\min\{PCR, C_d\}$ *where* C_d *is the maximum capacity allowed in the destination. We refer the calculation of* r_s *by using ER field in both FRM cell and BRM cell to as advanced calculation.*

Example 2 *Figure 2.3 shows the plot for* g *vs.* η *when*

- $\Psi = \{1, 2, 3, 4\}$,
- $(r_1, r_2, r_3, r_4) = (4, 2, 5, 2)$,
- $(u_1, u_2, u_3, u_4) = (3, 2, 4, 1)$, *and*
- $\Psi^u = \{1\}$.

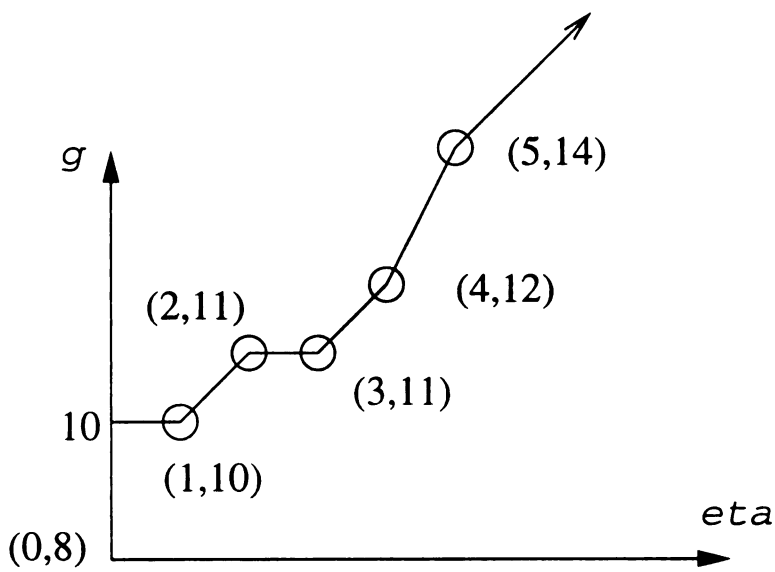


Figure 2.3: An example of g.

We have an important property referred to as *lower bound potential* as follows:

Lemma 4 *Consider the function* $g_{c,l}$ *defined in (2.1). Let* $\eta' := \min\{\eta : g_{c,l}(\eta) = K_l\}$ *and let* $\hat{\Psi} = \{i \in \Psi_l : a_i \geq \eta'\}$. *There exists a sequence* $\{r_i\}_{(i \in \Psi)}$ *such that*

$\forall i \in \hat{\Psi}, \max\{u_i, r_i\} \geq \max\{u_i, \eta'\}$ ***and*** $\forall i \in \Psi_l - \hat{\Psi}, r_i = a_i$. ***Define*** $\hat{g}_l$ ***as***

$$\hat{g}_l(\eta) := \sum_{i \in \hat{\Psi}_l^u} \max\{\eta, u_i\} + \sum_{i \in \Psi_l - \hat{\Psi}_l^u} \Big(\max\{\eta, u_i\} \mathbb{I}(\max\{r_i, u_i\} \geq \eta) + \max\{r_i, u_i\} \mathbb{I}(\max\{r_i, u_i\} < \eta) \Big),$$

where $\{\hat{j}^*\} \subseteq \hat{\Psi}_l^u \subseteq \hat{\Psi} \cup \{\hat{j}^*\}$ ***and*** $\hat{j}^* \in \arg\max_{i \in \Psi_l}(\max\{u_i, r_i\})$. ***We have*** $\eta' = \min\{\eta : \hat{g}_l(\eta) = K_l\}$.

Proof: Recall $j^* \in \arg\max_{i \in \Psi_l} a_i$ in (2.1). If $j^* = \hat{j}^*$, then $\forall \eta \leq \eta'$, we have

$$\begin{aligned}
\hat{g}_l(\eta) &= \max\{\eta, u_{\hat{j}^*}\} + \sum_{i \in \hat{\Psi}_l^u - \{\hat{j}^*\}} \max\{\eta, u_i\} \\
&\quad + \sum_{i \in \Psi_l - \hat{\Psi}_l^u - \{\hat{j}^*\}} \Big(\max\{\eta, u_i\} \mathbb{I}(\max\{r_i, u_i\} \geq \eta) \\
&\quad + \max\{r_i, u_i\} \mathbb{I}(\max\{r_i, u_i\} < \eta) \Big) \\
&= \max\{\eta, u_{j^*}\} + \sum_{i \in \hat{\Psi}_l^u - \{j^*\}} \max\{\eta, u_i\} \\
&\quad + \sum_{i \in \Psi_l - \hat{\Psi}_l^u - \{j^*\}} \max\{\eta, u_i\} \mathbb{I}(a_i \geq \eta) + a_i \mathbb{I}(a_i < \eta) \\
&= g_{c,l}(\eta).
\end{aligned}$$

If $j^* \neq \hat{j}^*$, then either 1) $\max\{r_{\hat{j}^*}, u_{\hat{j}^*}\} < \eta'$ and $a_{j^*} < \eta'$, or 2) $\max\{r_{\hat{j}^*}, u_{\hat{j}^*}\} \geq \eta'$ and $a_{j^*} \geq \eta'$. If it is the case 1), then $\hat{\Psi}_l^u = \{\hat{j}^*\}$, and $a_{j^*} = \max\{u_{\hat{j}^*}, r_{\hat{j}^*}\} = a_{\hat{j}^*}$.

Therefore, $\forall \eta, a_{j^*} < \eta \leq \eta'$,

$$\begin{aligned}
\hat{g}_l(\eta) &= \max\{\eta, u_{\hat{j}^*}\} + \sum_{i \in \Psi_l - \{\hat{j}^*\}} \max\{r_i, u_i\} \\
&= \eta + a_{j^*} + \sum_{i \in \Psi_l - \{\hat{j}^*, j^*\}} a_i \\
&= \eta + a_{\hat{j}^*} + \sum_{i \in \Psi_l - \{\hat{j}^*, j^*\}} a_i \\
&= \max\{\eta, u_{j^*}\} + \sum_{i \in \Psi_l - \{j^*\}} a_i \\
&= g_{c,l}(\eta)
\end{aligned}$$

If it is the case 2), then $\forall \eta \leq \eta'$,

$$\begin{aligned}
\hat{g}_l(\eta) &= \max\{\eta, u_{\hat{j}^*}\} + \max\{\eta, u_{j^*}\} + \sum_{i \in \hat{\Psi}^u - \{\hat{j}^*, j^*\}} \max\{\eta, u_i\} \\
&\quad + \sum_{i \in \Psi_l - \hat{\Psi}_l^u - \{j^*\}} \Big(\max\{\eta, u_i\} \mathbb{I}(\max\{r_i, u_i\} \geq \eta) \\
&\quad + \max\{r_i, u_i\} \mathbb{I}(\max\{r_i, u_i\} < \eta) \Big) \\
&= \max\{\eta, u_{j^*}\} + \sum_{i \in \hat{\Psi}_l^u - \{j^*\}} \max\{\eta, u_i\} \\
&\quad + \sum_{i \in \Psi_l - \hat{\Psi}_l^u - \{j^*\}} \max\{\eta, u_i\} \mathbb{I}(a_i \geq \eta) + a_i \mathbb{I}(a_i < \eta) \\
&= g_{c,l}(\eta)
\end{aligned}$$

The lemma follows. □

Lemma 5 *Consider Problem 1. If $A \subseteq B \subseteq \Psi$, then $\forall \eta \in [0, \infty), g_A(\eta) \leq g_B(\eta)$, where g_A and g_B are the function g using $\Psi^u = A$ and $\Psi^u = B$, respectively.*

Proof: For all $i \in B - A$, either $r_i \leq u_i$ or $r_i > u_i$. In either case,

$$\max\{\eta, u_i\}\mathbb{I}(\max\{r_i, u_i\} \geq \eta) + \max\{r_i, u_i\}\mathbb{I}(\max\{r_i, u_i\} < \eta) \leq \max\{\eta, u_i\}.$$

Therefore, $g_A(\eta) \leq g_B(\eta)$. □

Let $\mathcal{L} = \{l_1, l_2, ..., l_n\}$. Suppose under the persistent environment, we use the centralized algorithm in Figure 2.1. In what follows, we will use the notations in Table 2.1.

parameter	description
$r_{s,l}(t)$	r_s at time t for link l
g_l^t	the function g for link l at time t
$\Psi_l^u(t)$	as Ψ^u in g_l^t
$j_l^*(t)$	in $\arg\max_{i\in\Psi_l}(\max\{r_{i,l}(t), u_i\})$
$j_{l,k}^*$	j^* for link l at line 4 in kth while-loop iteration of algorithm in Figure 2.1
$\Psi_{c,l}^{\infty,k}$	$\{s \in \Psi_l : a_s = \infty$ at line 4 in kth while-loop iteration of algorithm in Figure 2.1$\}$
$g_{c,l}^k$	the function $g_{c,l}$ at line 4 in kth while-loop iteration of algorithm in Figure 2.1
$\eta_l(t)$	parameter at time t for link l
$\underline{\eta}(t)$	$(\eta_{l_1}(t), \eta_{l_2}(t), ..., \eta_{l_n}(t))$
η_l'	the assignment for link l by the algorithm in Figure 2.1
$\underline{\eta'}$	$(\eta_{l_1}', \eta_{l_2}', ..., \eta_{l_n}')$
a_s'	the flow for connection s resulted from $\underline{\eta'}$
$\Psi_{c,l}^{u,k}$	$\{j_{l,k}^*\} \cup \Psi_{c,l}^{\infty,k} \cup \{s \in \Psi_l : a_s' \geq \eta_k'\}$
$g_l^{t,k}$	g using $\Psi_l, \{r_{i,l}(t)\}_{(i\in\Psi_l)}, \{u_i\}_{(i\in\Psi_l)}, \Psi_{c,l}^{u,k}$ as $\Psi, \{r_i\}_{(i\in\Psi)}, \{u_i\}_{(i\in\Psi)}, \Psi^u$
τ_d	maximum round trip delay among each pair of switch and source
τ_u	maximum interval between two consecutive updates for each link parameter

Table 2.1: The notations used in the proofs.

Lemma 6 *Given* $k, 1 \leq k \leq n$ *and* $i, k \leq i \leq n$, *let* $\Psi = \Psi_{l_i}$, $\Psi^u = \Psi_{c,l_i}^{u,k}$, *and*

$\forall s \in \Psi - \Psi_{c,l_i}^{\infty,k}, \max\{u_s, r_s\} = a'_s$. *Therefore,* $\forall \eta \in [0, \eta'_{l_k}]$, $g(\eta) = g_{c,l_i}^k(\eta)$.

Proof: For all $\eta \in [0, \eta'_{l_k}]$, we have

$$\begin{aligned} g(\eta) &= \sum_{j \in \Psi_{c,l_i}^{u,k}} \max\{\eta, u_j\} + \sum_{j \in \Psi_{l_i} - \Psi_{c,l_i}^{u,k}} \Big(\\ &\quad \max\{\eta, u_j\} \mathbb{I}(\max\{r_j, u_j\} \geq \eta) + \max\{r_j, u_j\} \mathbb{I}(\max\{r_j, u_j\} < \eta) \Big) \\ &= \max\{\eta, u_{j^*_{l_i,k}}\} + \sum_{j \in \Psi_{c,l_i}^{u,k} - \{j^*_{l_i,k}\}} \max\{\eta, u_j\} + \sum_{j \in \Psi_{l_i} - \Psi_{c,l_i}^{u,k} - \{j^*_{l_i,k}\}} \Big(\\ &\quad \max\{\eta, u_j\} \mathbb{I}(a'_j \geq \eta) + a'_j \mathbb{I}(a'_j < \eta) \Big) \\ &= g_{c,l_i}^k(\eta) \end{aligned}$$

□

Without loss of generality, we assume $\forall i, 1 \leq i \leq n, l'_i$ is chosen to be $\hat{l}$ at line 7 in ith while-loop iteration of Figure 2.1 and thus $\eta'_{l_1} \leq \eta'_{l_2} \leq \cdots \leq \eta'_{l_n}$.

Theorem 2 First Version of Global Convergence:

Under the persistent environment, suppose we use $\eta_l(t) = \min\{\eta : g_l^t(\eta) = K_l\}$, *as the parameter for link* l *at time* t, *where* $\{j_l^*(t)\} \subseteq \Psi_l^u(t) \subseteq \{j_l^*(t)\} \cup \Psi'_l(t)$, *and*

$$\Psi'_l(t) := \left\{ i \in \Psi_l : \max\{r_i(t), u_i\} \geq \min_{s \in [t-\tau, t]} \eta_l(s) \right\},$$

and τ *is a non-negative constant. Then* $\exists T < \infty$ *such that* $\forall t > T, \underline{\eta}(t) = \underline{\eta}'$.

Proof: Let $t_0 = 0$ and $\forall i > 0$, define $t_i := t_{i-1} + T_i\tau + 2\tau_d + (T_i+1)\tau_u$, where T_i is a finite positive integer. We claim that $\forall k, 1 \le k \le n, \forall t \ge t_k, \forall i, 1 \le i \le k, \eta_{l_i}(t) = \eta'_{l_i}$ and $\forall i, k < i \le n, \eta_{l_i}(t) \ge \eta'_{l_k}$.

For all $t \ge \tau_u, \forall i, 1 \le i \le n$, $\forall \eta \in [0, \eta'_1], g^t_{l_i}(\eta) \le g^{t,k}_{l_i}(\eta) = g^1_{c,l_i}(\eta)$ by $\Psi^{\infty,1}_{c,l_i} = \Psi_{l_i}$, Lemma 5 and Lemma 6, and thus $\eta_{l_i}(t) \ge \eta'_1$. In addition, $\forall t \ge \tau_d + \tau_u, \forall s \in \Psi_{l_1}, r_{s,l_1}(t) \ge \max\{u_s, \eta'_1\}$ by the flow assignment equation (1.4) and thus by Lemma 4, $\forall t \ge \tau_d + 2\tau_u, \eta_{l_1}(t) = \eta'_{l_1}$.

Suppose our claim holds true for some $k < n$. Therefore, $\forall t \ge t_k + \tau_d, \forall i, k < i \le n, \forall s \in \Psi_{l_i} - \Psi^{\infty,k+1}_{c,l_i}$, $r_{s,l_i}(t) = a'_s$, and $\forall s \in \Psi^{\infty,k+1}_{c,l_i}, r_{s,l_i}(t) \ge \max\{u_s, \eta'_{l_k}\}$ by flow assignment equation (1.4). The following two cases need to be considered:

- $\eta'_{l_{k+1}} = \eta'_{l_k}$: By Lemma 4, $\forall t \ge t_k + \tau_d + \tau_u, \eta_{l_{k+1}}(t) = \eta'_{l_{k+1}}$.

- $\eta'_{l_{k+1}} > \eta'_{l_k}$: At line 6 in kth while loop, $\forall i, k < i \le n, g^k_{c,l_i}(\eta'_{l_k}) \le g^{k+1}_{c,l_i}(\eta'_{l_k}) < K_{l_i}$ by Lemma 1, and thus by Lemma 5 and Lemma 6, $\forall t \ge t_k + \tau_d + \tau_u, g^t_{l_i}(\eta'_{l_k}) \le g^{t,k}_{l}(\eta'_{l_k}) = g^k_{c,l_i}(\eta'_{l_k}) < K_{l_i}$, and hence $\eta_{l_i}(t) > \eta'_{l_k}$. Therefore, $\forall i, k < i \le n$,

  - if $\Psi^u_{l_i}(t) \subseteq \Psi^{u,k+1}_{c,l_i}$, then $\forall t \ge t_k + \tau_d + \tau + \tau_u$, by Lemma 5, $\forall \eta \in [0, \infty), g^t_{l_i}(\eta) \le g^{t,k+1}_{l_i}(\eta)$;

  - if $\Psi^{\infty,k+1}_{c,l_i} = \emptyset$ and $\max\{r_{j^*_l(t)}, u_{j^*_l}\} = a'_{j^*_{l,k}} \le \eta'_{l_k}$, where $\Psi^u_{l_i}(t) = \{j^*_l(t)\}$, and thus $\forall \eta \in [\eta'_{l_k}, \infty), g^t_{l_i}(\eta) = g^{t,k+1}_{l_i}(\eta)$;

  - if $\exists s \in \Psi_{l_i} - \Psi^{\infty,k+1}_{c,l_i}, \eta'_{l_k} < \max\{r_s(t), u_s\} = a'_s < \eta_{l'_{k+1}}$; let $s \in \arg\min_{j \in \Psi^u_{l_i}(t)} a'_s$ and $|\Psi^u_{l_i}(t)| > 1$; $g^t_{l_i}(a'_s) = g^{k+1}_{c,l_i}(a'_s) < K_{l_i}$ and thus $\forall t \ge t_k + \tau_d + \tau + \tau_u, \eta_{l_i}(t) > a'_s$, and hence either $\Psi^u_{l_i}(t) = \{s\}$ or $s \notin \Psi^u_{l_i}(t)$; if it is the former case, $\forall t \ge t_k + \tau_d + \tau + \tau_u, \forall \eta \in [a'_s, \infty), g^t_{l_i}(\eta) \le g^{t,k+1}_{l_i}(\eta)$;

if it is the latter case, we can repeat the process above till $\Psi^u_{l_i}(t) \subseteq \Psi^{u,k+1}_{c,l_i}$ or $|\Psi^u_{l_i}(t)| = 1$, and have that $\forall t \geq t_k + \tau_d + T_{k+1}(\tau + \tau_u)$, where T_{k+1} is a finite positive integer, $\exists \epsilon > 0, \forall \eta \in [\eta'_{l_{k+1}} - \epsilon, \infty), g^t_{l_i}(\eta) \leq g^{t,k+1}_{l_i}(\eta)$;

by Lemma 6, $\forall t \geq t_k + \tau_d + T_{k+1}(\tau + \tau_u), \forall \eta \in [0, \eta'_{l_{k+1}}], g^{t,k+1}_{l_i}(\eta) = g^{k+1}_{c,l_i}(\eta)$, and hence $\exists \epsilon > 0, \forall \eta \in [\eta'_{l_{k+1}} - \epsilon, \eta'_{l_{k+1}}], g^t_{l_i}(\eta) \leq g^{k+1}_{c,l_i}(\eta)$ and $\eta_{l_i}(t) \geq \eta'_{l_{k+1}}$ due to $\eta'_{l_{k+1}} \leq \min\{\eta : g^{k+1}_{c,l_i}(\eta) = K_{l_i}\}$. Thus $\forall t \geq t_k + T_{k+1}\tau_u + 2\tau_d + T_{k+1}\tau, \forall s \in \Psi_{l_{k+1}} - \Psi^{u,k+1}_{c,l}, r_{s,l_{k+1}}(t) \geq \max\{u_s, \eta'_{l_{k+1}}\}$ and thus by Lemma 4, $\forall t \geq t_{k+1}, \eta_{l_{k+1}}(t) = \eta'_{l_{k+1}}$.

By induction, we prove the theorem. □

Remark 8 *One interpretation of the theorem of Global Convergence is that we aggressively mark the unconstrained connections but passively unmark the unconstrained connections. That is, if it is an unconstrained connection with respect to current η, then it should be recognized immediately when calculating the new η. If it is a constrained connection with respect to current η, then it is fine if it is miss-recognized to be a constrained connection within finite time except the one with maximum arrival rate.*

In the followings, we will consider how to reduce the computation complexity for finding the solution of $g(\eta) = K$.

Definition 1 $\Lambda(\gamma)$-convergence:

Given a sequence $b_1, b_2, \ldots$ with $\lim_{k\to\infty} b_k = b^*$ and a constant γ, $0 < \gamma < 1$, let $j = \inf\{i : b_i \geq b^*\}$. If

- $j = 1$ *or* $b_1, b_2, ..., b_{j-1}$ *is increasing but below* b^*, *and*

- $j = \infty$ *or* $b_j, b_{j+1}, ...$ *is decreasing, and*

- $\forall i > 0, i \neq j-1, |b_{i+1} - b^*| \leq \gamma |b_i - b^*|$,

then we say $\{b_k\}$ *is* $\Lambda(\gamma)$*-convergent to* b^*, *and denote the convergence by* $b_k \rightarrow_{\Lambda(\gamma)} b^*$ *as* $k \rightarrow \infty$.

Definition 2 $\Lambda(\gamma, \delta^{\pm})$-convergence:

Given a sequence $b_1, b_2, ...$ *with* $b_k \rightarrow_{\Lambda(\gamma)} b^*$ *as* $k \rightarrow \infty$ *and constants* $\delta^+ > 0, \delta^- > 0$, *let* $T = \inf\{i : b_i \in [b^* - \delta^-, b^* + \delta^+]\}$. *If* $T < \infty$ *and* $\forall k > T, b_k = b^*$, *then we say* $\{b_k\}$ *is* $\Lambda(\gamma, \delta^{\pm})$*-convergent to* b^* *and denote the convergence by* $b_k \rightarrow_{\Lambda(\gamma, \delta^{\pm})} b^*$.

For any $\Psi, 0 < |\Psi| < \infty, (r_i)_{i \in \Psi}$, $(u_i)_{i \in \Psi}$, where $\forall i \in \Psi, r_i \geq 0, u_i \geq 0$ and any constant K with $K \geq \sum_{i \in \Psi} u_i$, we want to find an η^* such that $g(\eta^*) = K$, where g is defined in (2.4) with $\Psi^u = \{j^*\}$ and $j^* \in \arg\max_{i \in \Psi} \max\{u_i, r_i\}$. Since the function g might not be concave due to the non-zero MCRs, therefore, it is reasonable to find the solution iteratively to avoid the computation overhead. Table 2.2 has the description of notations which will be used in the following analysis.

If the starting point $\eta(0) \in [\eta^M_{g,K}, \min \Psi^1_{g,K}]$ or $\eta(0) \in [\max\{0, \max \Psi^2_{g,K}\}, \eta^m_{g,K}]$, then we can simply use the slope information

$$\begin{aligned} h_{g,K}(\eta) := & \left|\{i \in \Psi : u_i < \eta \leq r_i\}\right| \mathbb{I}(g(\eta) < K) \\ & + \left|\{i \in \Psi : u_i \leq \eta < r_i\}\right| \mathbb{I}(g(\eta) > K) \end{aligned} \tag{2.5}$$

parameter	description
$\eta^M_{g,K}$	$\max\{\eta : g(\eta) = K\}$
$\eta^m_{g,K}$	$\min\{\eta : g(\eta) = K\}$
$\Psi^1_{g,K}$	$\left\{\max\{r_i, u_i\} : i \in \Psi, \max\{r_i, u_i\} > \eta^M_{g,K}\right\}$
$\Psi^2_{g,K}$	$\left\{\max\{r_i, u_i\} : i \in \Psi, \max\{r_i, u_i\} < \eta^m_{g,K}\right\}$
$\delta^+_{g,K}$	$\min \Psi^1_{g,K} - \eta^M_{g,K}$
$\delta^-_{g,K}$	$\eta^m_{g,K} - \max \Psi^2_{g,K}$
$\delta^+_l(t)$	$\min \Psi^1_{g^t_l,K_l} - \eta^M_{g^t_l,K_l}$
$\delta^-_l(t)$	$\eta^m_{g^t_l,K_l} - \max \Psi^2_{g^t_l,K_l}$
$\eta_l(t^-)$	parameter for link l after the last update before time t
$l_{\hat{\mathcal{L}}}(t)$	in $\arg\min_{l\in\hat{\mathcal{L}}} \eta_l(t)$
$\eta^M_{l_k}$	$\max\{\eta : g^k_{c,l_k}(\eta) = K_{l_k}\}$
$\hat{\mathcal{L}}_i$	$\{l \in \mathcal{L} : \eta^M_l \le \eta^M_i\}$ initially
$\eta^m_l(t)$	$\min\{\eta : g^t_l(\eta) = K_l\}$

*We let $\min\emptyset = \infty$ and $\max\emptyset = -\infty$.

Table 2.2: The notations used for analysis of Local Convergence.

and $\eta(0) + \frac{K - g(\eta(0))}{\max\{h_{g,K}(\eta(0)),1\}}$ will be the solution due to g is continuously piece-wise linear. We have the following theorem.

Theorem 3 Local Convergence:

Consider an update rule. For any $\Psi, 0 < |\Psi| < \infty, (r_i)_{i\in\Psi}$, $(u_i)_{i\in\Psi}$, *where* $\forall i \in \Psi, r_i \ge 0, u_i \ge 0$ *and any constant* K *with* $K \ge \sum_{i\in\Psi} u_i$, *the update rule starts from a point* $\eta(0)$ *and generates updated sequence* $\{\eta(k)\}$, *where* $\forall j > 0$, *if* $g(\eta(j)) = K$, *then* $\eta(j+1) = \eta(j)$, *to find a solution for the equation* $g(\eta) = K$ *and* g *is defined in (2.4) with* $\Psi^u = \{j^*\}$ *and* $j^* \in \arg\max_{i\in\Psi} \max\{u_i, r_i\}$.

- *If* $\exists\gamma, 0 < \gamma < 1$ *and* $\eta^*, g(\eta^*) = K$ *such that* $\eta(k) \to_{\Lambda(\gamma)} \eta^*$ *as* $k \to \infty$, *then* $\underline{\eta}(t) \to \underline{\tilde{\eta}'}$ *as* $t \to \infty$, *and*

- *in addition, if* $\eta(k) \to_{\Lambda(\gamma,\delta^{\pm}_{g,K})} \eta^*$, *then* $\exists T < \infty, \forall t \ge T, \underline{\eta}(t) = \underline{\hat{\eta}'}$,

where $\forall s \in \Psi', \hat{a}_s = \tilde{a}_s = a'_s$, *and* $\hat{a}_s$ *and* $\tilde{a}_s$ *are the flows resulted from the assignments* $\underline{\hat{\eta}'}$ *and* $\underline{\tilde{\eta}'}$, *respectively.*

Before proving the theorem, recall that $\eta'_{l_k} = \min\{\eta : g^k_{c,l_k}(\eta) = K_{l_k}\}$ and consider the following lemma:

Lemma 7 *If* $\forall k, 1 \leq k \leq n, \forall i, 1 \leq i \leq k, \forall j, k \leq j \leq n, \eta_{l_i} \geq \eta'_{l_i}$, $g_{c,l_i}(\eta_{l_i}) = K_{l_i}$, *and* $\eta_{l_j} \geq \eta'_{l_k}$, *then* $\forall s \in \Psi_{l_i}, a''_s = a'_s$, *where* a''_s *is the flow resulted from the assignment* $\underline{\eta} := (\eta_{l_1}, \eta_{l_2}, ..., \eta_{l_n})$.

Proof: Let $k = 1$. Since $\forall s \in \Psi_{l_1}, a''_s \geq \max\{u_s, \eta'_{l_1}\} = a'_s$, therefore, $\forall \eta \in [0, \eta^M_{l_1}]$, it is easy to see $g_{c,l_1}(\eta) = g^1_{c,l_1}(\eta)$. Hence, if $\exists s \in \Psi_{l_1}, a''_s > a'_s$, then $\forall \epsilon > 0, g_{c,l_1}(\eta^M_{l_1} + \epsilon) > K_{l_1}$ and thus $\eta_{l_1} \leq \eta^M_{l_1}$. Therefore, $a''_s = \max\{u_s, \eta'_{l_1}\} = \max\{u_s, \eta^M_{l_1}\} = a'_s$ by Lemma 3 (apply η'_{l_1} as η_1 and $\eta^M_{l_1}$ as η_2).

Suppose the statement holds true for some k with $1 \leq k < n$. Therefore, $\forall s \in \Psi_{l_i}, a''_s = a'_s$, where $1 \leq i \leq k$. For l_{k+1}, since $\forall s \in \Psi_{l_{k+1}} - \bigcup_{i \leq k} \Psi_{l_i}, a''_s \geq \max\{u_s, \eta'_{l_{k+1}}\} = a'_s$ and $\forall s \in \Psi_{l_{k+1}} \cap \bigcup_{i \leq k} \Psi_{l_i}, a''_s = a'_s$, therefore, $\forall \eta \in [0, \eta^M_{l_{k+1}}]$, it is easy to see $g_{c,l_{k+1}}(\eta) = g^{k+1}_{c,l_{k+1}}(\eta)$. If $\exists s \in \Psi_{l_{k+1}}, a''_s > a'_s$, then $\forall \epsilon > 0, g_{c,l_{k+1}}(\eta^M_{l_{k+1}} + \epsilon) > K_{l_{k+1}}$ and thus $\eta_{l_{k+1}} \leq \eta^M_{l_{k+1}}$. Hence $a''_s = \max\{u_s, \eta'_{l_{k+1}}\} = \max\{u_s, \eta^M_{l_{k+1}}\} = a'_s$ by Lemma 3 (apply $\eta'_{l_{k+1}}$ as η_1 and $\eta^M_{l_{k+1}}$ as η_2). Therefore, by induction, we prove the lemma. □

Sketch of proof for Theorem 3: Since the proof will mainly follow the proof of Theorem 2, we only show the sketch of the proof. In addition, we will show the part for $\Lambda(\gamma, \delta^{\pm}_{g,K})$-convergence. The part for $\Lambda(\gamma)$-convergence can be done in the

similar way.

Let $t_0 = 0$ and $\forall i, 1 \leq i \leq n$, define $t_i := t_{i-1} + T_i'$, where T_i' is a finite number. We claim that $\forall k, 1 \leq k \leq n, \forall t \geq t_k, \forall i, 1 \leq i \leq k, \eta_{l_i}(t) \geq \eta'_{l_i} g^t_{l_i}(\eta_{l_i}(t)) = K_{l_i}$ and $\forall i, k < i \leq n, \eta_{l_i}(t) \geq \eta'_{l_k}$.

For all $t \geq \tau_u, \forall i, 1 \leq i \leq n$, by the second paragraph in the proof of Theorem 2, $\eta^m_{l_i}(t) \geq \eta'_1$; we need to consider the following two cases:

- if $\eta_{l_i}(t^-) < \eta^m_{l_i}(t)$, then $\eta_{l_i}(t) \geq \eta^m_{l_i}(t)$ or $\eta^m_{l_i}(t) - \eta_{l_i}(t) < \gamma(\eta^m_{l_i}(t) - \eta_{l_i}(t^-))$, since $\eta_{l_i}(t)$ will be updated in the way of $\Lambda(\gamma)$-convergence to $\eta^*_{l_i}(t)$ where $\eta^*_{l_i}(t) = \eta^m_{l_i}(t)$. Therefore, let $\exists s \in \Psi_{l_i}, \eta_{l_i}(t^-) \leq \max\{r_{s,l_i}, u_s\} < \eta^m_{l_i}(t)$, then after some finite interval $T^1_{l_i}$, $\eta_{l_i}(t + T^1_{l_i}) > \max\{r_{s,l_i}, u_s\}$ and thus we have have $\eta^m_{l_i}(t + T_{l_i}) - \delta^-_{l_i}(t + T_{l_i}) \leq \eta_{l_i}(t + T^1_{l_i}) < \eta^m_i(t + T^1_{l_i})$ and we have $\eta_{l_i}(t + T^1_{l_i} + t_u) = \eta^m_{l_i}(t + T^1_{l_i} + t_u) \geq \eta'_{l_i}$ due to $\Lambda(\gamma, \delta^{\pm}_{l_i}(t + T^1_{l_i} + t_u))$-convergence. Therefore, $\exists T^1_1 < \infty, \forall t \geq T^1_1, \eta_{l_i}(t) \geq \eta'_{l_1}$.

- By the above statement, we have $\forall i, 1 \leq i \leq n, \forall t \geq T^1_1 + \tau_d, \forall s \in \Psi_{l_i}, r_{s,l_i}(t) \geq \eta'_{l_1}$. Therefore, if $\exists s \in \Psi_{l_1}, r_{s,l_1}(t) > \max\{\eta^M_1, u_s\}$, then $\forall \epsilon > 0, g^t_{l_1}(\eta^M_{l_1} + \epsilon) > K_{l_1}$ and thus $0 \leq (\eta_{l_1}(t + \tau_u) - \eta^M_{l_1}) < \gamma(\eta_{l_1}(t) - \eta^M_{l_1})$. Hence, $\exists T^2_1, \forall t > T^2_1$, let $l = l_{\hat{\mathcal{L}}_1}(t)$ and $\eta_l(t) \leq \eta^M_{l_1}$ due to $\Lambda(\gamma, \delta_l(t)^{\pm})$-convergence. Let $\hat{\mathcal{L}}_1 \leftarrow \hat{\mathcal{L}}_1 - \{l\}$ and repeat the process, we have $\exists T'_1 < \infty, \forall t \geq T'_1, g^t_{l_1}(\eta_{l_1}(t)) = K_{l_1}$.

By the similar argument above, the third paragraph in the proof of Theorem 2 and by Lemma 7, we can prove our claim holds true for every $k \leq n$. By induction, we prove the theorem. □

Remark 9 *One interpretation of Local Convergence is as follows: If current*

link parameter η is less than any η^* such that $g(\eta^*) = K$, then in the next update, the link parameter should be increased by at least $\gamma(\eta^* - \eta)$, where γ is a constant and $0 < \gamma < 1$. However, if current link parameter η is greater than any η^* such that $g(\eta^*) = K$, then in the next update, the link parameter should be decreased by at least $\gamma(\eta - \eta^*)$, but the new η cannot be smaller than η^*.

We consider an update rule which does not satisfy the condition of Local Convergence and leads to the *thrashing hazard.*

Example 3 *Given $\forall i \in \Psi, u_i = 0$, consider the update rule*

$$\eta(k+1) \leftarrow \begin{cases} \frac{K - \sum_{i \in \Psi} r_i \mathbb{I}(r_i < \eta(k))}{\sum_{i \in \Psi} \mathbb{I}(r_i \geq \eta(k))} & \text{if } \sum_{i \in \Psi} \mathbb{I}(r_i \geq \eta(k)) > 0, \\ K - \sum_{i \in \Psi} \mathbb{I}(r_i < \eta(k)) + r_j. & \text{otherwise.} \end{cases} \tag{2.6}$$

To see that the sequence $\{\eta(k)\}$ given by the above update rule is not Λ-convergent to η^* where $g(\eta^*) = K$ with $\Psi^u = \{j^*\}$, consider the network in the Figure 2.4. Consider the link parameter η_2 for link l_2. Initially, $\eta_2(1) = 4.3$. Notice that $\Psi^u = \{s_3\}$ and $\eta_2^* = 2.05$. If each arrival rate keeps constant, then the sequence generated by the update rule will be $4.3, 0.9, 1.775, 2.0333, 2.05, 2.05, 2.05, \ldots\ldots$ Since the sequence bounces across 2.05 in downwards direction, therefore, it is not Λ-convergence. There exists a thrashing hazard for this update rule when the arrival rate is changing. For example, before obtaining $\eta_2(3)$, it is possible that all arrival rates are 0.9 because of $\eta_2(2) = 0.9$ and flow assignment equation (1.4). There-

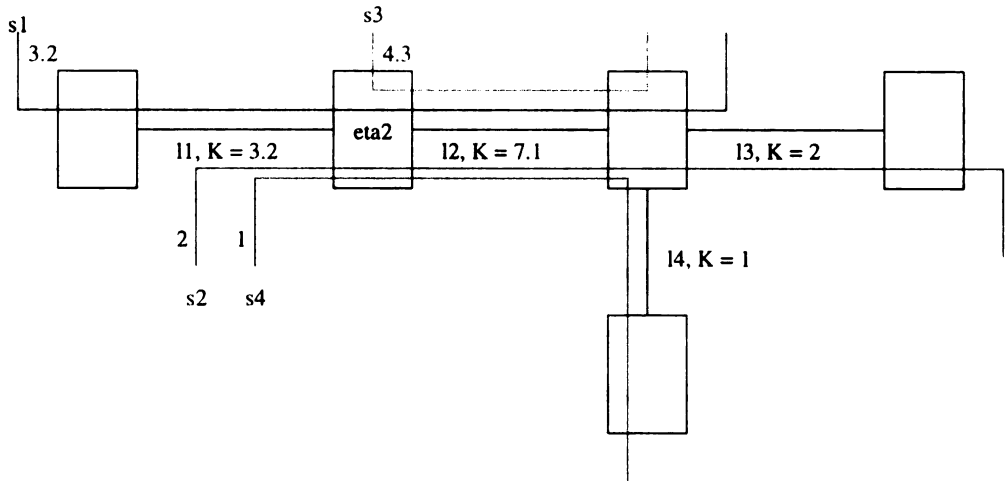


Figure 2.4: The network for the counter-example.

fore, $\eta_2(3) = 4.3$. *When determining* $\eta_2(4)$, *all arrival rates are returned to the original values by the flow assignment equation. Therefore,* $\eta_2(4)$ *will bounce to* 0.9 *again and thus it is possible that the sequence* $\{\eta_2(k)\}$ *becomes* $4.3, 0.9, 4.3, , 0.9, ..., 4.3, 0.9, ...$

Theorem 3 gives a sufficient condition for convergence to max-min fairness. The condition in Theorem 3 is intuitive and will be used to generate a class of update rules. In addition, we have the following corollary from Theorem 3.

Corollary 1 Second Version of Global Convergence:

Under the persistent environment, let $\eta_l(t)$ *defined as follows:*

$$\eta_l(t) = \begin{cases} \eta_l(t^-) & \text{if } g_l^t(\eta_l(t^-)) = K_l, \\ \eta : g_l^t(\eta) = K_l & \text{otherwise,} \end{cases}$$

where $\Psi_l^u(t) = \{j_l^*(t)\}$ *(see Table 2.1). Suppose we use* $\eta_l(t)$ *as the parameter for link* l *at time* t, *and* $\Psi_l^u(t)$ *is defined as in the Theorem 2. Then* $\exists T < \infty$

such that $\forall t > T, \underline{\eta}(t) = \underline{\eta}''$, *where* $\forall s \in \Psi', a''_s = a'_s$ *and* a''_s *is the flow resulted from the assignment* $\underline{\eta}''$.

2.3 Distributed Update Rules

In the following examples, we will let $j := \arg\max_{i \in \Psi} \max\{r_i, u_i\}$ and $\Psi^u = \{j\}$. The further discussion about the usage of Ψ^u will be in Chapter 4. In addition, we let $\eta = \min_{i \in \Psi} u_i$ when $\sum_{i \in \Psi} u_i = K$.

2.3.1 Slow Update Rule (SUR)

We consider the following update rule:

$$\eta(k) \leftarrow \eta(k-1) + \frac{K - g(\eta(k-1))}{|\Psi|}$$

To see this update rule satisfying the condition of local convergence (Theorem 3), suppose $\eta(0)$ and η^* are lying on the same linear segment of function g, where $g(\eta^*) = K$. If $g(\eta(0)) = K$, then $\eta(0)$ is what we want. If $g(\eta(0)) \neq K$, then let $\Delta = g(\eta^*) - g(\eta(0))$. Since $|g(\eta^*) - g(\eta(0))| \leq |\Psi| \times |\eta^* - \eta(0)|$, therefore, $\exists \gamma, 0 < \gamma \leq 1$ such that $|\eta^* - \eta(k)| = \Delta(1-\gamma)^k$. Therefore, $\eta(k) \rightarrow_{\Lambda(\gamma)} \eta^*$ as $k \rightarrow \infty$. Suppose there is no η^* such that it lies on the same linear segment as $\eta(0)$. We only need to prove that $\exists M < \infty$ such that $\eta(M)$ will across the end point of the linear segment, say η', which is nearer to η^*. Suppose $g(\eta(0)) < g(\eta^*)$. Therefore, $\eta(0) \leq \eta' < \eta^*$. Let $\epsilon := g(\eta^*) - g(\eta')$. If $g(\eta(0)) = g(\eta')$, then $\exists \Delta > 0$

such that $\eta(k) - \eta(k-1) = \Delta$ and it is easily to seen that $\exists M < \infty$ such that $\eta(M) \geq \eta'$. If $g(\eta(0)) < g(\eta')$, then let $\Delta = g(\eta') - g(\eta(0))$ and we have $\forall k$ such that $\eta(k) < \eta'$ and $0 < \gamma < 1$ satisfying that $|\eta^* - \eta(k)| = (\Delta + \epsilon)(1-\gamma)^k$. Therefore, $\exists M < \infty$ such that $\eta(M) \geq \eta'$. For the part $g(\eta(0)) > g(\eta^*)$, we can apply the same technique. Note that in all cases, $\{\eta(k)\}$ is monotone sequence and $\eta(k) \rightarrow_{\Lambda(\gamma)} \eta^*$ as $k \rightarrow \infty$.

2.3.2 Modified Slow Update Rule (MoSUR)

We can accelerate the Slow Update Rule by using the following update rule:

$$\eta(k) \leftarrow \begin{cases} \eta(k-1) + \frac{K - g(\eta(k-1))}{\max\{1, h_{g,K}(\eta(k-1))\}} & \text{if } g\left(\eta(k-1) + \frac{K - g(\eta(k-1))}{\max\{1, h_{g,K}(\eta(k-1))\}}\right) = K, \\ \eta(k-1) + \frac{K - g(\eta(k-1))}{|\Psi|} & \text{otherwise,} \end{cases}$$

where $h_{g,K}$ is defined in (2.5). It is easy to see that the sequence generated by MoSUR update rule is $\Lambda(\gamma, \delta^{\pm}_{g,K})$-convergent to some η^* where γ is defined in SUR and $g(\eta^*) = K$.

2.3.3 Algorithm SHARE

The last update rule that we consider is simply obtaining the solution of $g(\eta) = K$. We use an iterative algorithm as shown in Figure 2.5. The part from line 1 to 7 is to find a new Ψ^u satisfying $\exists \eta$ such that $\forall i \in \Psi - \Psi^u, \max\{u_i, r_i\} < \eta$ and $\sum_{i \in \Psi^u} \max\{u_i, \eta\} = K - \sum_{i \in \Psi - \Psi^u} \max\{u_i, r_i\}$. It is obvious that if letting $\eta' = \max_{i \in \Psi - \Psi^u} \max\{u_i, r_i\}$ and $\sum_{i \in \Psi^u} \max\{u_i, \eta'\} > K - \sum_{i \in \Psi - \Psi^u} \max\{u_i, r_i\}$, then Ψ^u

is what we want. If the above condition cannot be satisfied, we need to put all i such that $i \in \arg\max_{i \in \Psi - \Psi^u} \max\{u_i, r_i\}$ in Ψ^u.

Input: a sequence of non-negative numbers $(r_i)_{i \in \Psi}$,
a decreasing sequence with non-negative numbers $(u_i)_{i \in \Psi}$,
a subset $\Psi^u \subseteq \Psi$ and a constant $K \geq \sum_{i \in \Psi} u_i$.
and previous η;
// Obtain new Ψ^u
1 $K' \leftarrow K - \sum_{i \in \Psi - \Psi^u} \max\{r_i, u_i\}$; $j \leftarrow -1$;
2 while ($\sum_{i \in \Psi^u} \max\{\eta, u_i\} \geq K'$ and $\Psi^u \neq \Psi$) or $\Psi^u = \emptyset$ do
3 if $j \geq 0$, then $\Psi^u \leftarrow \Psi^u \cup \{j\}$;
4 $j \leftarrow \arg\max_{i \in \Psi - \Psi^u} \max\{r_i, u_i\}$;
5 $\eta \leftarrow \max\{r_j, u_j\}$;
6 $K' \leftarrow K' + \eta$;
7 end while;
// Obtain η
8 $j \leftarrow \arg\min\{u_i : i \in \Psi^u\}$;
9 $U \leftarrow \Psi^u$;
10 $K_L \leftarrow K' - \sum_{i \in \Psi^u} u_i$;
11 do
12 $U \leftarrow U - \{j\}$;
13 $K_L \leftarrow K_L + u_j$;
14 $\eta \leftarrow \frac{K_L}{|\Psi^u - U|}$;
15 $j \leftarrow \arg\min\{u_i : i \in U\}$;
16 while $\eta > u_j$;
17 return η.

Figure 2.5: The iterative procedure SHARE.

Once the new Ψ^u is obtained, we can focus on finding η^* satisfying

$$\sum_{i \in \Psi^u} \max\{\eta^*, u_i\} = K - \sum_{i \in \Psi - \Psi^u} \max\{u_i, r_i\}.$$

The right hand side of the above equality is denoted as K'. Since it is clear that $\eta^* \leq K' - \sum_{i \in \Psi^u} u_i + \min \Psi^u$, we will set the initial value of η to be the right hand side of the inequality. As in the first part, we partition Ψ^u into two sets, i.e., U

and $\Psi^u - U$, where $U := \{i : u_i \geq \eta, i \in \Psi^u\}$. If $\eta \leq \min U$ and $\eta = \frac{K_l}{|\Psi^u - U|}$, where $K_L = K - \sum_{i \in \Psi - \Psi^u} \max\{u_i, r_i\}$, then η is the solution. If the above condition is not true, then η must be larger than the solution η^*. We then take the element, which is minimum of U, out of U. This operation will cause the next η be smaller than the current η but larger than the value of the element taken out. We will repeat the procedure until the solution is found. Note that η^* may not be unique only when $\sum_{i \in \Psi^u} u_i = K'$. Therefore, in this case, we will select $\eta^* = \min\{u_i : i \in \Psi^u\}$ to validate the equality (2.4). The part of the algorithm to obtain η^* is shown in line 8 – 17 in Figure 2.5.

2.4 Simulation

We implement our flow control algorithm on the NIST ATM simulator and conduct our simulation on a multi-broadband-link network which is similar to that used in [28] for simulation. Before starting the ABR services, we let the trace flowing through the links for fifteen minutes to stabilize the predictor. For each ABR session, we set MCR be zero. Each ABR session is best-effort. Furthermore, in each ABR source, we set RIF to be 0.0625 and peak cell rate to be 149.76 Mbps. Figure 2.6 shows the multi-broadband-link network system. Each link has capacity 149.76 Mbps and the same background trace as mentioned in [44], with different starting point runs through each link 'BBi'. The trace has mean around 126 Mbps. In addition, the output buffer size in each switch is 2000 cells.

To decide K_l, let $Y_0, Y_{\frac{1}{2}}, Y_1, Y_{\frac{3}{2}}, ...$ be the time series of VBR and CBR traffic.

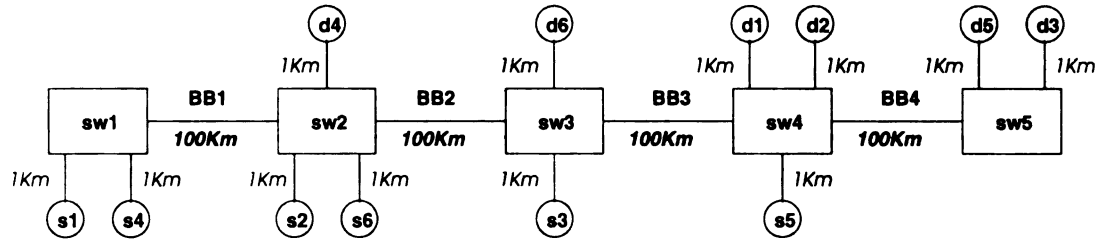
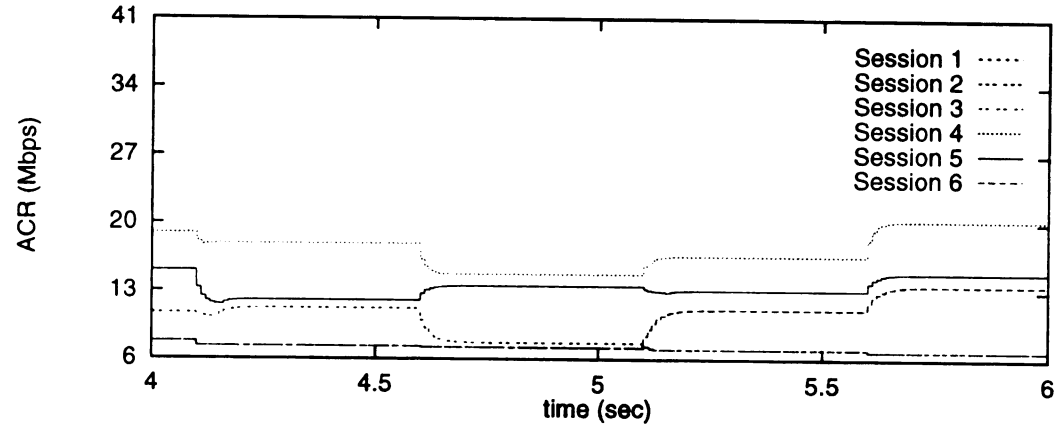
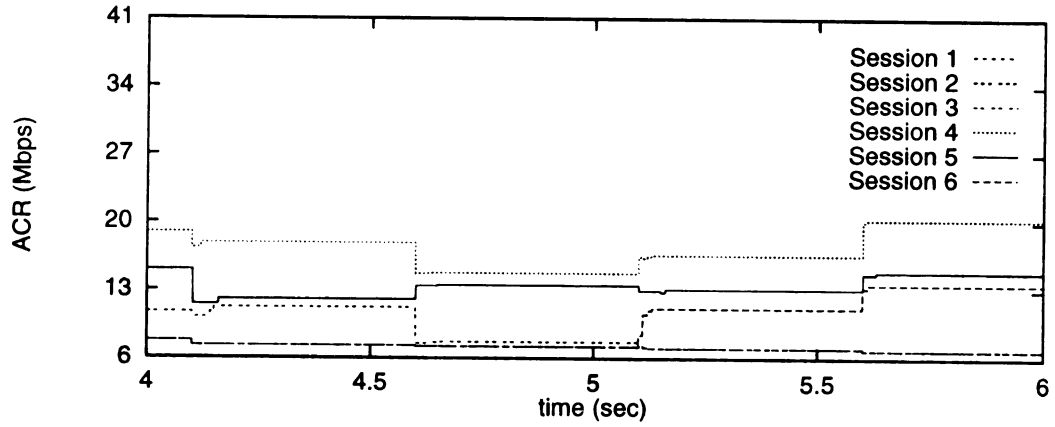


Figure 2.6: The network topology. The session i has source s_i and destination d_i.

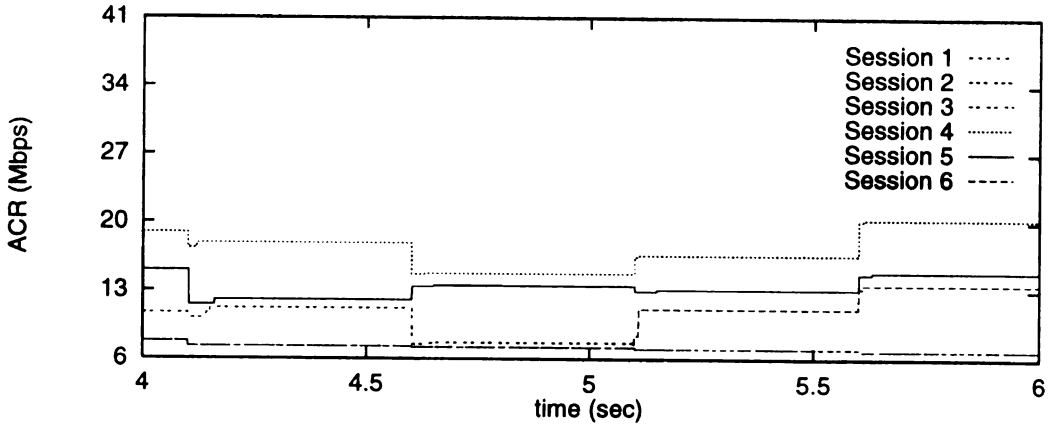
Let $Y_0 \equiv 0$. The variable $Y_{\frac{k}{2}}$ is the number of bits in the VBR and CBR traffic arriving to the link l during the time period $[\frac{k-1}{2}, \frac{k}{2})$. Therefore, at time $t \in [\frac{v}{2}, \frac{v+1}{2})$ where v is a non-negative integer, $K_l = C_l - 2Y_{\frac{v}{2}} - \kappa$, where $C_l = 149.76$ Mbps. We arbitrarily set $\kappa = 1$ Mbps. The interval between each two updates for feedback rate is set to be 10ms. Figure 2.7 shows the simulation results. As the simulation results show, the algorithm SHARE has the fastest transient response, and the update rule MoSUR is very similar to SHARE except at time 5.1 second, the response of ACRs of session 4 and 6 is a little bit slow. In most of time, the session 1 and 2 have the minimum ACR.



(a) ACRs when using SUR.



(b) ACRs when using MoSUR.



(c) ACRs when using SHARE.

Figure 2.7: The simulation results for multi-broadband-link network system.

Chapter 3

Bounded-Cell-Loss-Ratio Flow Control

Transmitting data with low data loss ratio is a basic requirement for future B-ISDN networks. Likewise, the capability to support a large number of destinations for point-to-multipoint data transmission is strongly dependent of the packet loss ratio. The loss ratio also plays a major role in the multimedia transmission with the *quality of service* (QoS) guarantee. More precisely, in real-time application, the buffer size in a switch should be restricted to bound the transmission delay. This leads to that the loss ratio will be increased if the data sending rate is not well controlled and thus the quality of voices or pictures will be degraded. Moreover, from the viewpoint of the network system, the loss ratio always has great impact on the network effective throughput and thus on the network routing and dimensioning.

There are several ways to carry out the low loss ratio property. First, when a

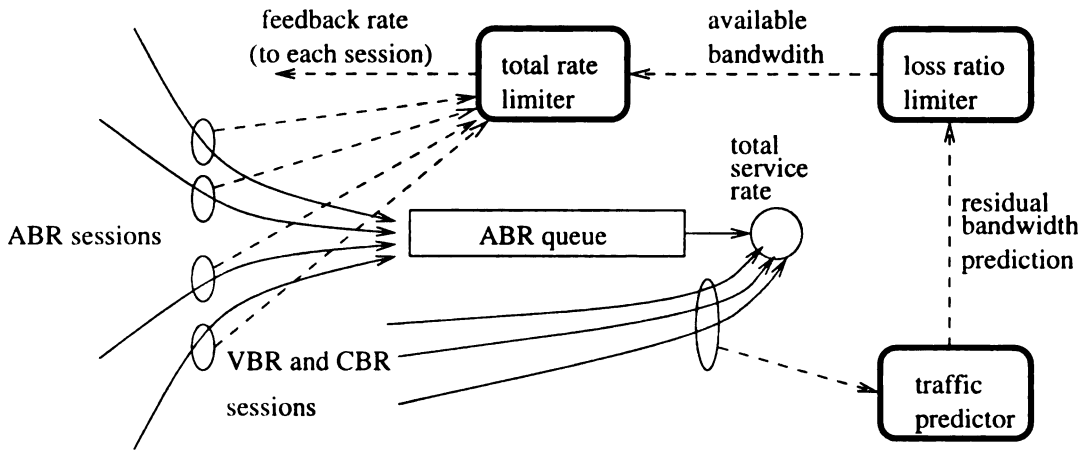


Figure 3.1: The Components for BLCR flow control.

call arrives at the boundary of the network, the network will allocate the buffers and bandwidths along the chosen path according to the traffic description. A call allowed to be set up will be associated with a mechanism in the network entry to police the traffic. In each switch along the route of the call, a traffic shaper in front of the link in the route can be employed [45]. Second, the application can adapt to the network current status to adjust the sending rate, i.e., use flow control, so that the loss ratio can be reduced. The former approach is usually dedicated to the higher priority traffic, and the latter one is designed to enable high network utilization. In this chapter, we study how to bound loss ratio by the flow control.

3.1 Framework

In this section, we give the framework about how the flow control algorithm functions. In Figure 3.1, we show the relation between each component which we will discuss as a subsection in the followings. Table 3.1 gives the description of notations which we will use in the followings.

Label	Description	Units
$Y_t + C_Y t$	fluid process for VBR and CBR traffic	cells
W_k	$Y_{\frac{k}{2}} - Y_{\frac{k-1}{2}}$	cells
$\hat{W}_k$	prediction of W_k	cells
U_k	$\nabla^d W_k$	cells
$\hat{U}_k$	prediction of U_k	cells
$\tilde{U}_{d,k}$	truncated version of U_k with degree d	cells
$\{Z_k\}$	an i.i.d. process	cells
$\hat{Z}_k$	prediction error defined as $W_k - \hat{W}_k$	cells
S_t	service fluid process for queue	cells
A_t	arrival fluid process for queue	cells
κ	amount of negative drift	cells per second*
X_t	$S_t - A_t - \kappa t$	cells
Q_t	queue length process	cells
x	buffer size	cells
K	total feedback rate	cells per second
η	feedback rate to each session	cells per second*
u_i	MCR for session i	cells per second
r_i	CCR for session i	cells per second
ς_l	current service rate for link l	cells per second
R_l	summation of CCRs of sessions via l	cells per second
Ψ_l	index set of ABR sessions through link l	–
Ψ_l^u	index set of unconstrained sessions for link l	–

* (Mbps in simulation)

3.1.1 Bandwidth Predictor: On-line Estimation for FARIMA model

Let $Y_t + C_Y t$ be the fluid process for VBR and CBR traffic, where $\forall t, \mathbb{E}Y_t = 0$, and $W_k = Y_{\frac{k}{2}} - Y_{\frac{k-1}{2}}$, i.e., the aggregated process per 0.5 second. We treat $\{W_k\}$ is an FARIMA$(1, d, 0)$ process and use a simple method for on-line estimating and prediction for the FARIMA$(1, d, 0)$ model. Let D be a finite set $\{\frac{i}{\Upsilon} : i \in \{0, 1, 2, ..., \Upsilon\}\}$. With each value in D, an expert is associated and the expert uses this value as its degree d in FARIMA$(p, d, 0)$ model. Each expert obtains truncated

U_k, denoted as $\tilde{U}_k$, as follows

$$\tilde{U}_{d,k} = \sum_{j=0}^{100} \pi_j(d) W_{k-j},$$

and then treats $\tilde{U}_{d,k}$ as an AR(1) process and uses the ordinary weighted least squares estimation [46] for the parameters. Each expert will generate its prediction according to (1.1) with replacing U_k by $\tilde{U}_{d,k}$. To conclude the predictions by these experts, we use a technique called *Hedge Boosting* [47] with $\beta = 0.1$.

Figure 3.2 shows the autocorrelation function for the trace which we will use in our simulation. As shown in figure, the autocorrelation function for the trace will not decay to zero until the time lag is large. The trace consists of twenty different thirty-minute MPEG traces [1] and each of them will be repeatedly and sequentially read to generate the final trace for thirteen times but with different starting point. When reading the last frame in an MPEG trace, we will wrap around and start to read the first frame. The total expected bandwidth needed for the traces is about 126 Mbps.

In Table 3.1, we show the average squared prediction error by each expert. Each expert aggregates twelve frames as its W_t. The best degree d is around 0.4. The Hedge boosting appears to be a better predictor since the best d will change during the trace running. We also show the squared prediction error when treating the underlying process $\{W_k\}$ as a random walk process. Its squared error

[1]The traces are fetched from the site ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG. Each item of data in each trace stands for the number of bits per frame.

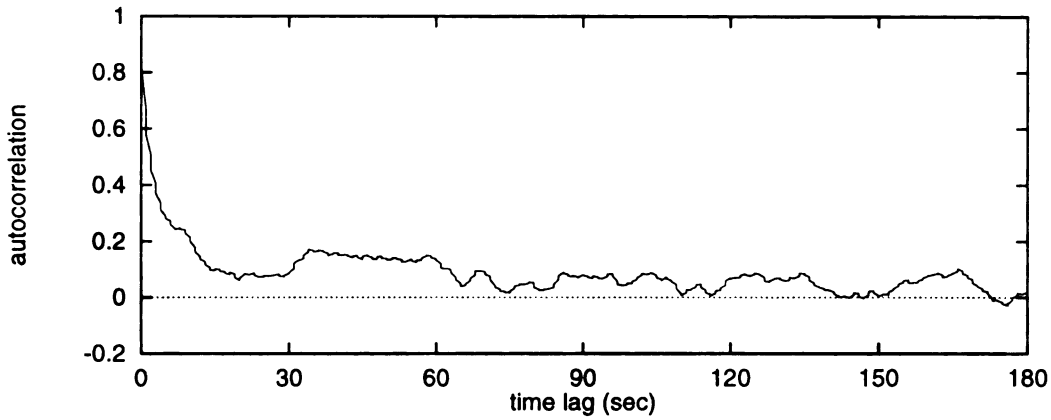


Figure 3.2: The autocorrelation function for the trace.

is 8% more than that of Hedge boosting. As we will show in the next subsection, the squared error is directly related to the bandwidth utilization. We also show the squared prediction error produced by the best linear predictor generated by the *Durbin-Levinson* algorithm [22], which needs the autocorrelation function in advance. It indicates that there is still a lot of space to improve the on-line predictor. The last item in Table 3.1 is the average squared error if using the mean as the predictor. The error is almost three times larger than any other predictor. Let $\hat{W}_k$ be the prediction generated by Hedge boosting for W_k. Figure 3.3 shows the autocorrelation function for the prediction error process $\{\hat{Z}_k\} = \{W_k - \hat{W}_k\}$. As shown in the figure, $\{\hat{Z}_k\}$ is uncorrelated. In the following analysis, we will assume $\{\hat{Z}_k\}$ is i.i.d.

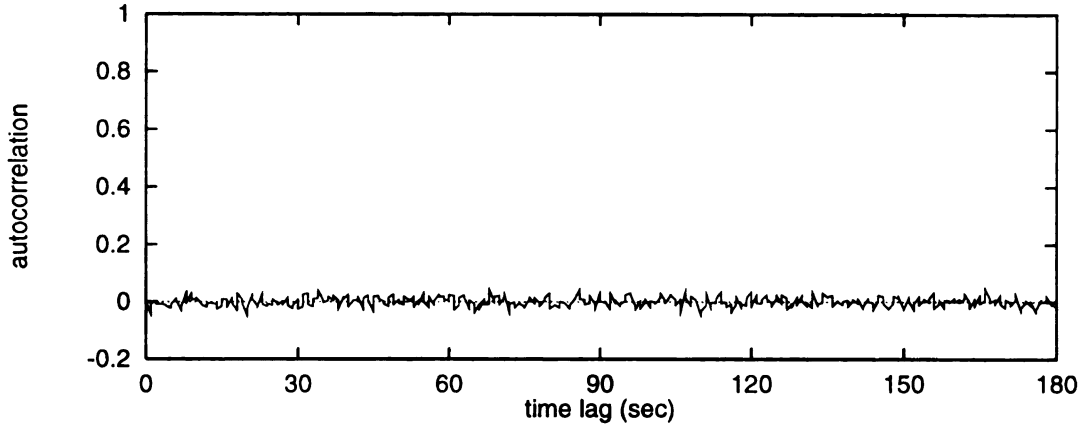


Figure 3.3: The autocorrelation function for the prediction error process by Hedge boosting.

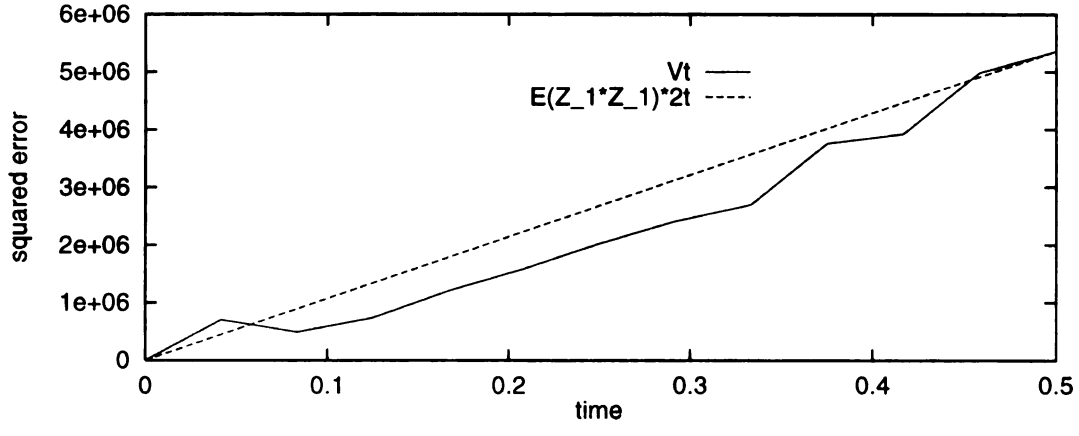


Figure 3.4: The plot for $\mathbb{E}V_t^2$ vs. time lag t.

3.1.2 Loss Ratio Limiter: The Decision of Linear Drift κ

Let C_l stand for the link capacity. Therefore, we have the relation for service fluid process S_t as follows:

$$S_t - S_0 = (C_l - C_Y)t - Y_t.$$

In addition, we let $\forall t, \mathbb{E}Y_t^2 < \infty$.

Let the total feedback rate for ABR sessions at time t to be $K_t := C_A - 2\hat{W}_M$, where $M = \lfloor 2t \rfloor$. Therefore, we have the relation for the arrival fluid process A_t as

predictor	average squared error
d = 0.0	5441007
d = 0.1	5431205
d = 0.2	5387331
d = 0.3	5371501
d = 0.4	5365298
d = 0.5	5380291
d = 0.6	5459099
d = 0.7	5542284
d = 0.8	5642833
d = 0.9	5826337
d = 1.0	5957649
Hedge boosting	5361959
random walk	5919789
Durbin Levinson (off-line)	4431095
mean	16652184

Table 3.1: Squared errors for each predictor.

follows:

$$\begin{aligned} A_t - A_0 &= \int_0^t K_s ds + \sum_{i\in\Psi} \xi_t^{(i)} \\ &= -2\hat{W}_{M+1}(t - \frac{M}{2}) - \sum_{k=1}^{M} \hat{W}_k + \sum_{i\in\Psi} \xi_t^{(i)} + C_A t, \end{aligned}$$

where $\forall i \in \Psi, \mathbb{E}\xi_t^{(i)} = 0$. We have the following relation for X_t:

$$\begin{aligned} X_t - X_0 &= A_t - A_0 - S_t + S_0 \\ &= Y_t - Y_0 - 2\hat{W}_{M+1}(t - \frac{M}{2}) - \sum_{k=1}^{M} \hat{W}_k + \sum_{i\in\Psi} \xi_t^{(i)} + (C_A - C_Y + C_l)t. \end{aligned}$$

Assume X_t is Gaussian and has stationary increments. Let $V_t := Y_t - Y_{\frac{M}{2}} - 2\hat{W}_{M+1}(t - \frac{M}{2})$. As Figure 3.4 shows, we can assume that $\mathbb{E}V_t^2 \approx 2(t - \frac{M}{2})E\hat{Z}_1^2$. We then have the following theorem:

Theorem 4 *Assume that the following condition holds*

$$\exists \tau_{max} > 0, \forall t, \quad \mathbb{E}(\sum_{i \in \Psi^u} \xi_t^{(i)})^2 << \mathbb{E}\hat{Z}_1^2 \tau_{max}. \tag{3.1}$$

and assume that V_t is uncorrelated with $\hat{Z}_1, \hat{Z}_2, ..., \hat{Z}_M$. We have

$$Var(X_0 - X_{-t}) \approx 2\mathbb{E}\hat{Z}_1^2 t, \qquad \forall t \geq \tau_{max}. \tag{3.2}$$

Proof:

$$\begin{aligned}
Var(X_0 - X_{-t}) &= Var(X_t - X_0) \\
&= \mathbb{E}\Big[Y_t - Y_0 - 2\hat{W}_{M+1}(t - \frac{M}{2}) - \sum_{k=1}^{M} \hat{W}_k + \sum_{i \in \Psi} \xi_t^{(i)} - \sum_{i \in \Psi} \xi_0^{(i)}\Big]^2 \\
&= \mathbb{E}\Big[Y_t - Y_{\frac{M}{2}} - 2\hat{W}_{M+1}(t - \frac{M}{2}) + \sum_{k=1}^{M}(W_k - \hat{W}_k) + \sum_{i \in \Psi} \xi_t^{(i)} \\
&\quad - \sum_{i \in \Psi} \xi_0^{(i)}\Big]^2 \\
&= \mathbb{E}(V_t + \sum_{k=1}^{M} \hat{Z}_k)^2 - 2\mathbb{E}\Big[(V_t + \sum_{k=1}^{M} \hat{Z}_k)(\sum_{i \in \Psi} \xi_t^{(i)} - \xi_0^{(i)})\Big] \\
&\quad + \mathbb{E}(\sum_{i \in \Psi} \xi_t^{(i)} - \xi_0^{(i)})^2 \\
&= \mathbb{E}V_t^2 + M\mathbb{E}\hat{Z}_1^2 - 2\mathbb{E}\Big[(V_t + \sum_{k=1}^{M} \hat{Z}_k)(\sum_{i \in \Psi} \xi_t^{(i)} - \xi_0^{(i)})\Big] \\
&\quad + \mathbb{E}(\sum_{i \in \Psi} \xi_t^{(i)} - \xi_0^{(i)})^2 \\
&= \mathbb{E}V_t^2 + M\mathbb{E}\hat{Z}_1^2 - 2\mathbb{E}\Big[(V_t + \sum_{k=1}^{M} \hat{Z}_k)(\sum_{i \in \Psi^u} \xi_t^{(i)} - \xi_0^{(i)})\Big] \\
&\quad + \mathbb{E}(\sum_{i \in \Psi^u} \xi_t^{(i)} - \xi_0^{(i)})^2 + \mathbb{E}(\sum_{i \in \Psi - \Psi^u} \xi_t^{(i)} - \xi_0^{(i)})^2,
\end{aligned}$$

by the assumption that V_t is uncorrelated with $\hat{Z}_1, \hat{Z}_2, ..., \hat{Z}_M$ and the Assumption 1. Then by Schwartz's inequality, (3.1) and Assumption 2, we have

$$\begin{aligned} Var(X_0 - X_{-t}) &\leq \mathbb{E}V_t^2 + M\mathbb{E}\hat{Z}_1^2 + 2\left[(\mathbb{E}V_t^2 + M\mathbb{E}\hat{Z}_1^2)\mathbb{E}(\sum_{i\in\Psi^u} \xi_t^{(i)} - \xi_0^{(i)})^2\right]^{\frac{1}{2}} \\ &\quad + \mathbb{E}(\sum_{i\in\Psi^u} \xi_{t-\tau_i}^{(i)} - \xi_{-\tau_i}^{(i)})^2 + \mathbb{E}(\sum_{i\in\Psi-\Psi^u} \xi_t^{(i)} - \xi_0^{(i)})^2 \\ &\approx 2\mathbb{E}\hat{Z}_1^2 t, \qquad \forall t \geq \tau_{max}. \end{aligned}$$

□

If we want to bound the loss ratio above by δ, then, by (1.2) and (3.2), we have the following:

$$\exp\left[-\frac{\kappa x}{\mathbb{E}\hat{Z}_1^2}\right] \lessapprox \delta. \tag{3.3}$$

Our duty is to obtain the smallest κ to satisfy the above inequality. Therefore, we should let $C_A = C_l - C_Y - \kappa$ and let the total feedback rate $K := C_A - 2\hat{W}_M$ at time t.

Recall that $t_x \sim \frac{x}{\kappa}$ in (1.3). Therefore, in order to have (3.3), we needs

$$\kappa \leq \frac{x}{\tau_{max}}. \tag{3.4}$$

We will assume τ_{max} as the maximum round trip delay. The reason why we choose τ_{max} in this way is simply by observing the experiment results and the proof of Theorem 5 in the next subsection. In practice, we set $\tau_{max} = 0.1$ for the interactive

application. Suppose $x = 2000$ cells, then $\kappa \leq 20000$ cells per second (8.48 Mbps). In our simulation, we will see in general it is the case.

3.1.3 Bounding Total Rate

To bound the total rate, we first need to distinguish whether a session is an unconstrained one. Then we use the algorithm called SHARE, introduced in the previous chapter, which can bound the expected total arrival rates by a given number. In addition, under the persistent environment, i.e., all the sessions are best-effort and the available link capacity is constant, we prove that the algorithm will converge and the utilization of the bottleneck link will reach the given number.

Marking Unconstrained Sessions

In previous section, we have discuss the partition of the set of sessions Ψ_l for link l according to the location of bottleneck for each session. That is, if the session's bottleneck located in the link, then we claim it is a unconstrained for link l; otherwise, it is constrained for l. We aggressively detect the session to be a unconstrained one. That is, for a fixed number J, e.g., the maximum round trip delay between each pair of source and switch, if $r_s(t) \geq \min_{t-J \leq k \leq t-1} \eta_l(k)$, then the session s will be recognized as being unconstrained for l. Otherwise, session s will be constrained. An easy way to implement this concept will be using a shift register $\hat{\eta}_l$ for η_l. At the step k, register $\hat{\eta}_l$ will keep the values $\eta_l(k), \eta_l(k-1), ..., \eta_l(k - \frac{J}{t_m})$ where t_m is the time between two steps. Let $\min \hat{\eta}_l$ denote the minimum value in $\hat{\eta}_l$. If $r_s(t) \geq \hat{\eta}_l^*$, then session s is unconstrained for l at time t; otherwise, it is

constrained.

Theorem 5 Bounding Total Arrival Rate:

Suppose we deal with a single switch network system and there exist N ABR sessions which are all unconstrained. Given K_t at time t, we use η_t^* , the solution in Problem 1 with replacing K by K_t, as the feedback rate, then with Assumption 3 and assuming that for each session, the delay for a cell traveling from the source to the switch is bounded, then $\exists \epsilon > 0$ such that

$$\int_0^t K_s ds - \epsilon \leq A_t - A_0 \leq \int_0^t K_s ds + \epsilon$$

where A_s is the arrival fluid process.

Proof: Let $\zeta_t^{(i)}$ denote the sending rate for session i at time t. Then $\sum_{i=1}^N \zeta_{t+\tau^{(b,i)}}^{(i)} = K_t$, where $\tau^{(b,i)}$ is the delay for sending a BRM cell from the switch to the source for session i. By Assumption 3, $\tau^{(b,i)}$ is constant. Therefore,

$$\int_0^t \sum_{i=1}^N \zeta_s^{(i)} ds + \sum_{i=1}^N \left(\int_t^{t+\tau^{(b,i)}} \zeta_s ds - \int_0^{\tau^{(b,i)}} \zeta_s^{(i)} ds \right) = \int_0^t K_s ds$$

Since if the delay for a cell from the source to the switch is bounded, say above by τ'_{max}, we have

$$\int_{\tau'_{max}}^{t-\tau'_{max}} \sum_{i=1}^N \zeta_s^{(i)} ds \leq A_t - A_0 \leq \int_{-\tau'_{max}}^{t+\tau'_{max}} \sum_{i=1}^N \zeta_s^{(i)} ds. \tag{3.5}$$

For the lower bound of (3.5), we have

$$\begin{aligned}\int_{\tau'_{max}}^{t-\tau'_{max}} \sum_{i=1}^{N} \zeta_s^{(i)} ds &= \int_0^t \sum_{i=1}^{N} \zeta_s^{(i)} ds - \int_0^{\tau'_{max}} \sum_{i=1}^{N} \zeta_s^{(i)} ds - \int_{t-\tau'_{max}}^{t} \sum_{i=1}^{N} \zeta_s^{(i)} ds \\ &= \int_0^t K_s ds - \sum_{i=1}^{N} \Big(\int_t^{t+\tau^{(b,i)}} \zeta_s ds + \int_0^{\tau^{(b,i)}} \zeta_s^{(i)} ds + \int_0^{\tau'_{max}} \zeta_s^{(i)} ds \\ &\quad + \int_{t-\tau'_{max}}^{t} \zeta_s^{(i)} ds \Big)\end{aligned}$$

Since $\forall i, \forall t, \zeta_t^{(i)}$ is bounded, and we repeat the similar procedure for the upper bound, therefore, the theorem is followed. □

Remark 10 *It is easy to see that Theorem 5 implies (3.1).*

Remark 11 *Suppose the network has multiple switches. For the unconstrained sessions for a link, we can still have the same relation as (5) but with replacing K_s by $K_s - \sum_{i \in \Psi_s - \Psi_s^u} \max\{r_i, u_i\}$, where Ψ_s and Ψ_s^u are the index sets for ABR sessions and unconstrained ABR sessions for the link at time s, respectively. Therefore, we can have (3.1) and obtain the result in Theorem 4. In this way, the negative linear drift κ can be derived to bound CLR. Using this κ, we can still have the upper bound for CLR even when $\forall s$, the actual total feedback rate $K'_s \leq K_s - \sum_{i \in \Psi_s - \Psi_s^u} \max\{r_i, u_i\}$, i.e., resulting from (1.5), because of Proposition 1.*

Remark 12 *Theorem 5 holds even when η_t^* will not converge as $t \to \infty$.*

The algorithm SHARE has the worst-case time complexity $O(|\Psi|^2)$ due to the summations inside the loop. If the time complexity is really an issue, then one can relax the resolution of η for finding the solution of (2.4) by binary search. However, the approximated solution η^* should satisfy $g(\eta^*) \leq K$. The number of search steps can be set to whatever the processor can handle. In the following analysis and our simulation, we use the algorithm SHARE.

Backward Congestion Notification by Switch

When the available bandwidth decreases, the frequency of the feedback rate back to the source. In this way, the source will keep sending the cells with higher rate for a longer time and thus the buffer will be saturated. To avoid this problem, whenever feedback rate decreases, a timer is set. If the timer is expired and the session has no BRM cell flowing through during the timer ticking down, the switch will generate a BRM cell sent back to the source.

3.1.4 Overall Flow Control Algorithm

The overall flow control algorithm is listed in Figure 3.5.

3.2 Two Loss Priority Traffic Stream

Suppose a session has traffic with two loss priorities. For example, a traffic stream with the same virtual circuit identifier has cells with different loss priorities. In [13], either the relative-rate-based or ER-based flow control is on the data cells with low

End of Estimation Interval (e.g., every 0.5 sec)
1 Obtain K according to the previous subsection
End of Measurement Interval (e.g., every 10 milli-sec)
2 if $\eta_1 < \eta$, then
3 for each $VC \in M - R$ do
4 generate BRM cell for VC; *// BCN*
5 end for;
6 end if;
// detect constrained and unconstrained sessions
7 for each $i \in \mathcal{C}$ do *// $\mathcal{C}$: the set of total sessions*
8 if $\max\{r_i, u_i\} < \min \hat{\eta}$, then $\Psi^u \leftarrow \Psi^u - \{i\}$;
9 else $\Psi^u \leftarrow \Psi^u \cup \{i\}$;
10 end for;
// obtain η
11 $\Psi \leftarrow \Psi^u \cup M \cup G$; *// $G := \{i : u_i > 0\}$*
12 $\eta_1 \leftarrow \eta$;
13 if $K \geq \sum_{i \in G} u_i$, then
14 $\eta \leftarrow$ **SHARE**$(\Psi^u, \{r_i\}_{(i \in \Psi)}, \{u_i\}_{(i \in \Psi)}, K, \eta)$;
15 else $\eta \leftarrow 0$;
16 end if;
17 push η to $\hat{\eta}$;
18 $M \leftarrow \emptyset$; $R \leftarrow \emptyset$;
When a cell with VC is received
19 $M \leftarrow M \cup \{VC\}$; *// session VC is effective*
When an BRM cell with VC is received:
20 ER_in_RM_Cell $\leftarrow \min\{$ER_in_RM_Cell, $\eta\}$;
21 $R \leftarrow R \cup \{VC\}$;
When a FRM cell with VC is received:
22 $r_{VC} \leftarrow$ CCR_in_RM_Cell.

Figure 3.5: The flow control algorithm in the switch.

loss priority (CLP = 0), i.e., the *in-rate* cells. The data cell with high loss priority (CLP = 1), i.e., the *out-of-rate* cells, is not allowed. Here we propose that ER-based flow control and relative-rate-based flow control will be on the cells with low loss priority and high loss priority, respectively.

We use a simple buffer management to bound the loss ratio of low-loss-priority traffic. Consider two buffer management schemes in Figure 3.6. Let $b + h$ denote the total buffer size where $b, h > 0$. If a cell with CLP = 0 arrives, then the cell is

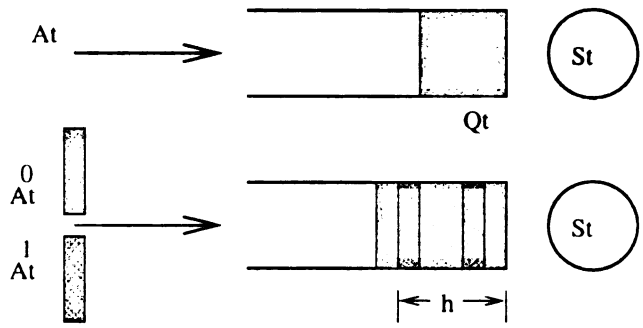


Figure 3.6: Two buffer management schemes.

allowed to enter into the buffer if there is an empty space in the buffer. However, a cell with CLP $= 1$ is allowed to enter into the buffer only if the buffer length is less than h.

To bound the loss ratio for the cells with CLP $= 0$, we can simply apply the algorithm in Figure 3.5 with only considering the cell with CLP $= 0$. For the flow control of cells with CLP $= 1$, we can use the binary fields NI and CI. That is, if the buffer length is larger than a positive constant h_1 which is less than h, then NI $= 1$ in the BRM cell. Moreover, if the buffer length is larger than h, then CI $= 1$ in the BRM cell. The source will simply check NI and CI fields in the BRM cell and follows the operation specified in [13] for the cells with CLP $= 1$. Then we can bound CLR for cells with CLP $= 0$ by the following theorem:

Theorem 6 *Consider two buffer management schemes in Figure 3.6. If* $A_t^0 \equiv A_t$ *and* $\forall t \leq 0, A_t \equiv 0$ *and* $A_t^1 \equiv 0$, *then*

$$\mathbb{P}(Q_t^1 > x) \geq \mathbb{P}(Q_t^2 > x + h),$$

where Q_t^1 *and* Q_t^2 *are queue length processes for buffer management scheme 1*

and 2, respectively.

Proof: We claim that $Q_t^1 \geq Q_t^2 - h$.

First, when $Q_t^2 \leq h$, then there is nothing to prove. Since $Q_0^1 = 0$ and $Q_0^2 = 0$, therefore, let t_0 defined as follows:

$$t_0 := \inf\{t : Q_t^1 \geq Q_t^2 - h = 1, Q_{t^-}^1 \geq Q_{t^-}^2 - h = 0\}.$$

If $t_0 = \infty$, then there is nothing to prove since either $Q_t^2 - h \leq 0$ always. Suppose $t_0 < \infty$, then let t_1 defined as follows:

$$t_1 := \inf\{t : Q_{t^-}^1 \geq Q_{t^-}^2 - h = 1, Q_t^1 \geq Q_t^2 - h = 0\}.$$

Note that t_1 could be infinity. Let $k_0 = Q_{t_0}^1 - Q_{t_0}^2 + h$. Then $\forall t \in [t_0, t_1], Q_t^1 - Q_{t_0}^2 + h = k_0$. If $t_m, t_{m+1} < \infty$ where $m \geq 0$ is even, then we define t_{m+2} as follows:

$$t_{m+2} := \inf\{t > t_{m+1} : Q_t^1 \geq Q_t^2 - h = 1, Q_{t^-}^1 \geq Q_{t^-}^2 - h = 0\}.$$

If $t_{m+2} < \infty$, then we define t_{m+3} as follows:

$$t_{m+3} := \inf\{t > t_{m+2} : Q_{t^-}^1 \geq Q_{t^-}^2 - h = 1, Q_t^1 \geq Q_t^2 - h = 0\}.$$

Let $k_{m+2} = Q_{t_{m+2}}^1 - Q_{t_{m+2}}^2 + h$. It is easy to see that $\forall t \in [t_{m+2}, t_{m+3}], Q_t^1 - Q_{t_0}^2 + h = k_{m+2}$. Therefore, we prove that, by induction, during the period $Q_t^2 > h, Q_t^1 \geq Q_t^2 - h$. Therefore, we have the claim and the theorem is

followed. □

To provide the further fairness, we can discard the cell with CLP = 1 when the queue length is larger than h_2, where $h_1 < h_2 < h$, and the arrival rate for cells with CLP = 0 of the session is larger than η. We will discuss the application of this scheme in [48, 49] for fault-tolerant and reliable multicast.

3.3 Simulation

We implement our flow control algorithm on the NIST ATM simulator and conduct our simulation on a one-broadband-link network system and a multi-broadband-link network. In both simulation, before starting the ABR services, we let the trace flowing through the links for fifteen minutes to stabilize the predictor. For each ABR session, we set MCR be zero.

3.3.1 One-Broadband-Link Network

Figure 3.7 shows the one-broadband-link network system. The bandwidth for 'BB1' is 149.76 Mbps. We use the trace, mentioned in Section 3, with mean around 126 Mbps, as the background traffic through 'BB1'. As in [28], we assume the mean of the background traffic is known in advance. Furthermore, in each ABR source, we set RIF to be 0.0625 and peak cell rate to be 149.76 Mbps. The allowed loss ratios will be $1\%, 5\%, 10\%$ and 20%, respectively. Figures 3.8, Figure 3.9, and Figure 3.10 show the simulation results. As the results indicated, CLR is bounded above by

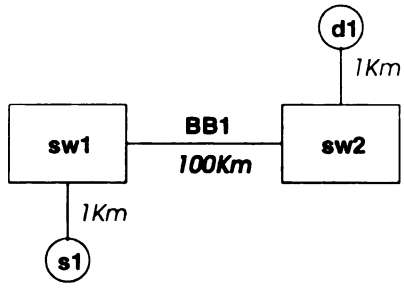


Figure 3.7: The one-broadband-link network system. The session 1 has source s_1 and destination d_1.

the given number with a factor about 20 while the link total bandwidth utilization is above 96% since $\kappa < 5.5$ and by the formula

$$\text{bandwidth utilization} = \frac{149.76 - \kappa}{149.76}.$$

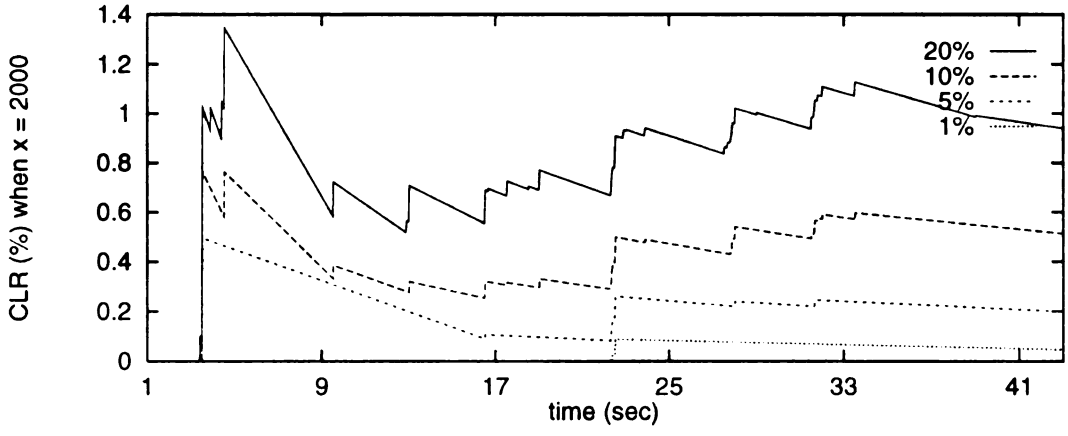


Figure 3.8: The loss ratio when x = 2000 in the one-broadband-link network.

Figure 3.11 and Figure 3.12 show the loss ratios for buffer size being 1000 cells and 5000 cells, respectively. As the results indicate, the CLRs for buffer size being 1000 cells and 5000 cells are similar to that for buffer size being 2000 cells.

3.3.2 Multi-Broadband-Link Network

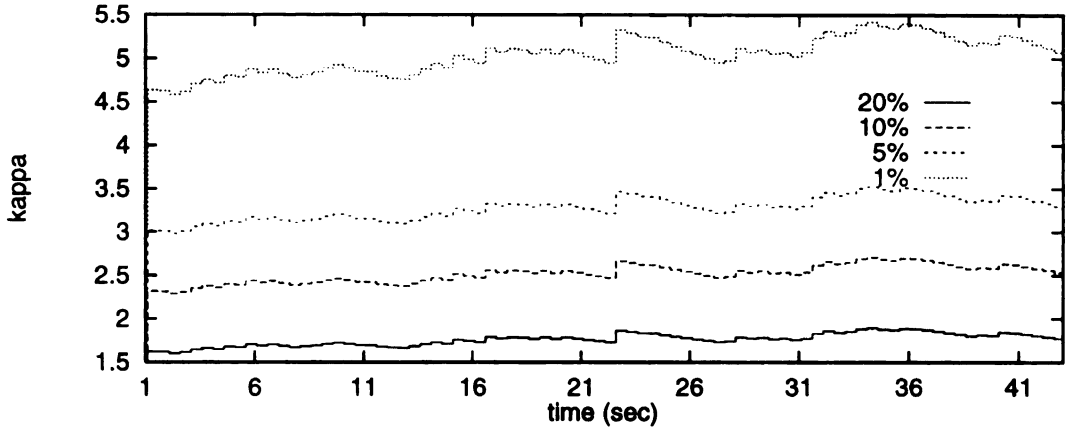


Figure 3.9: The κ when x = 2000 in the one-broadband-link network.

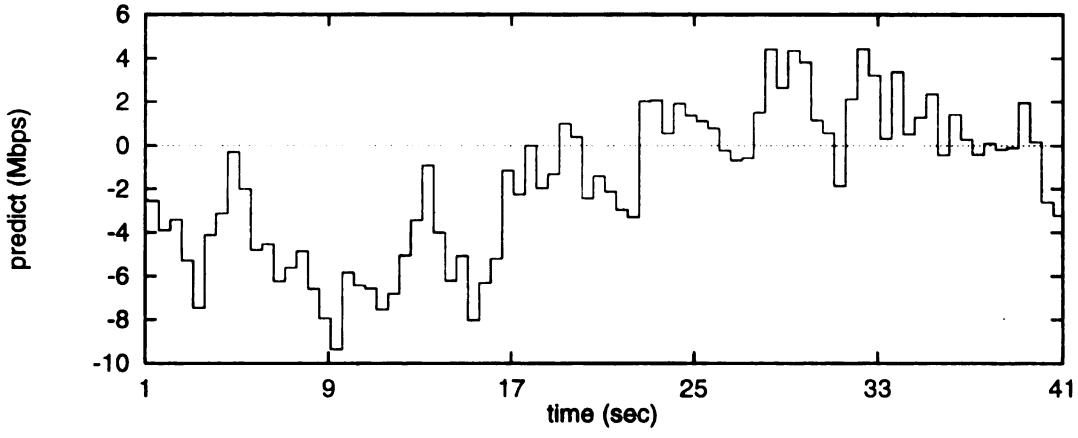


Figure 3.10: The bandwidth prediction for x = 2000 in the one-broadband-link network.

Figure 3.13 shows the multi-broadband-link network system. Each link has capacity 149.76 Mbps and the same background trace with different starting point runs through each link 'BBi'. Each ABR session is best-effort. We set $J = 0.1$ second for marking the unconstrained session. Figure 3.14–3.18 show the simulation results. As the results indicate, CLR is again bounded above by the given upper bound which is 1%. As Figure 3.18 shows, the session 1 and session 2 always have the smallest ACRs. In addition, the convergence to the new equilibrium is very fast. We observe that there exists the rate overshoot. It is due to the rate change of constrained sessions. However, for the minimum feedback rate, there will be no

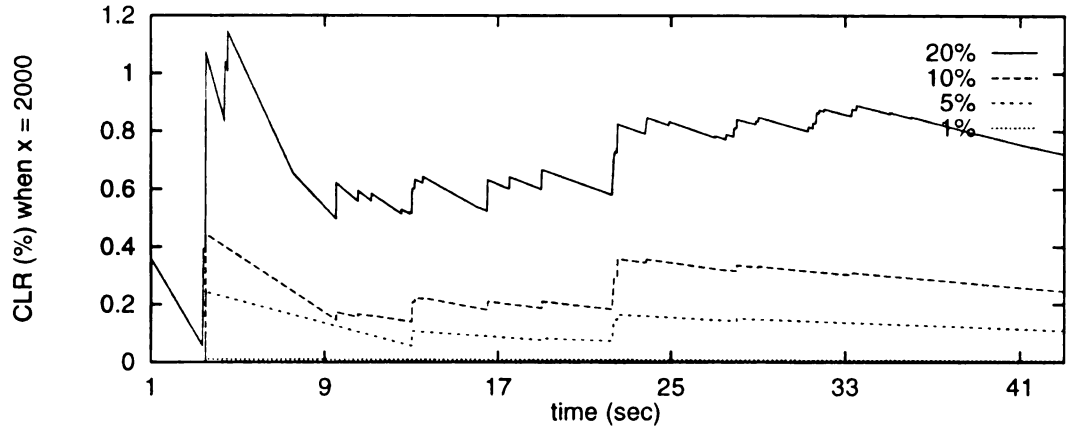


Figure 3.11: The loss ratio when $x = 1000$ in the one-broadband-link network.

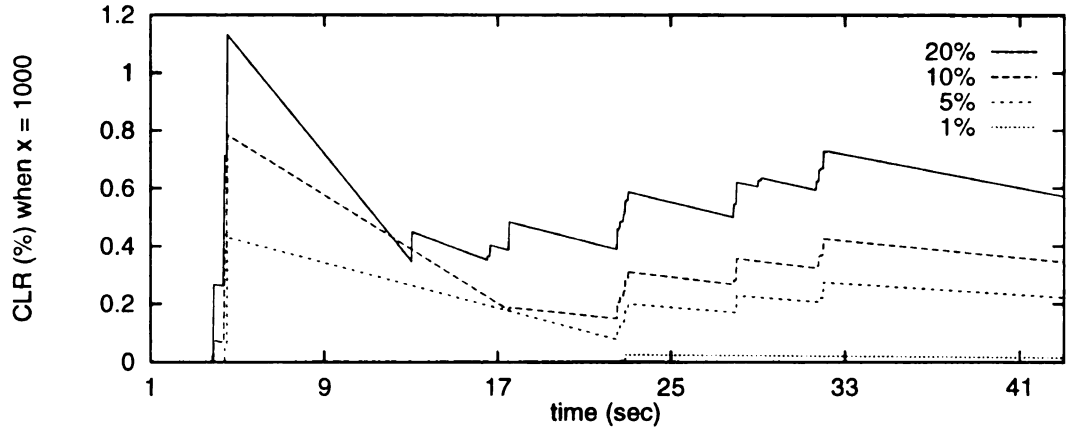


Figure 3.12: The loss ratio when $x = 5000$ in the one-broadband-link network.

rate overshoot. This result is implied by the proof of Theorem 2. Around time 5.3, the session 6 raises its rate since the switch 'sw2' recognizes that the session 1 and 2 are no longer unconstrained for the link 'BB2' from 'sw2'. The current bottlenecks of session 1 and 2 are at link 'BB3' from 'sw3'.

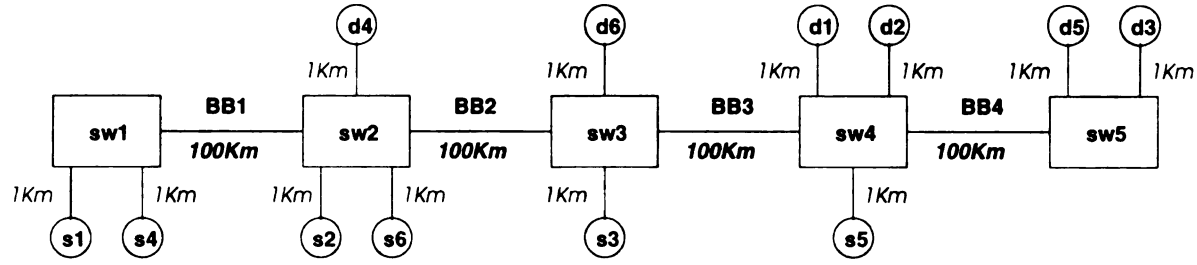


Figure 3.13: The second network topology. The session i has source s_i and destination d_i.

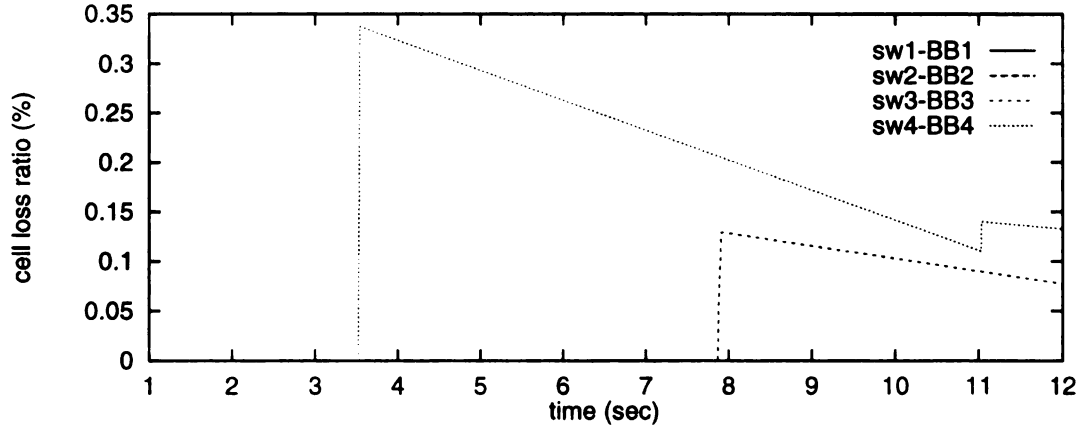


Figure 3.14: The loss ratio for multi-broadband-link network system with x = 2000.

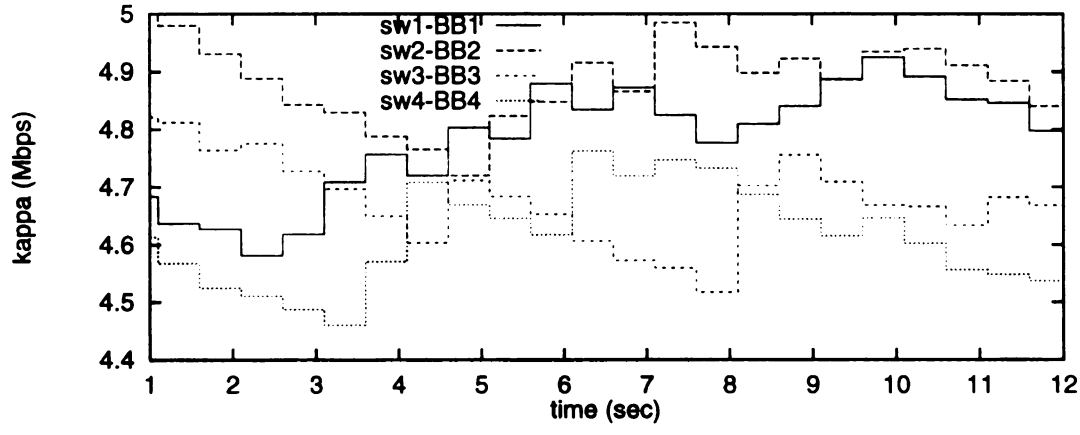


Figure 3.15: The κ for multi-broadband-link network system with x = 2000.

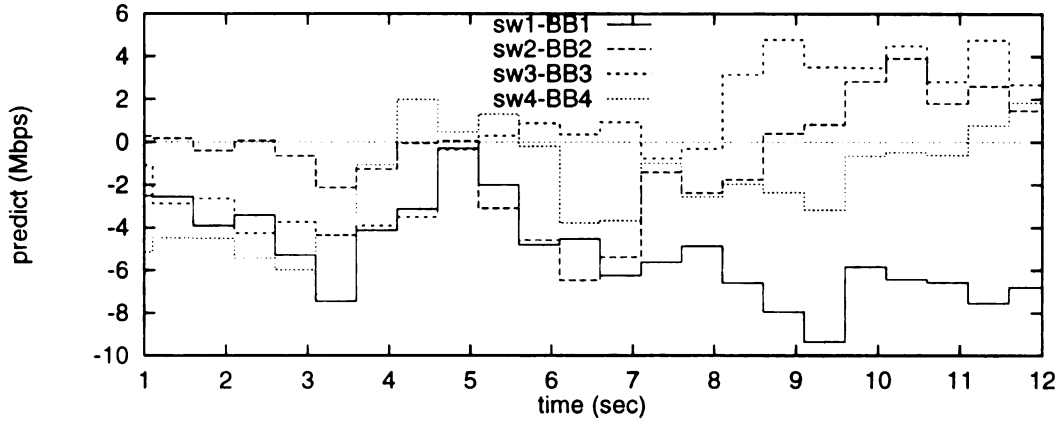


Figure 3.16: The bandwidth prediction for multi-broadband-link network system with $\mathbf{x} = 2000$.

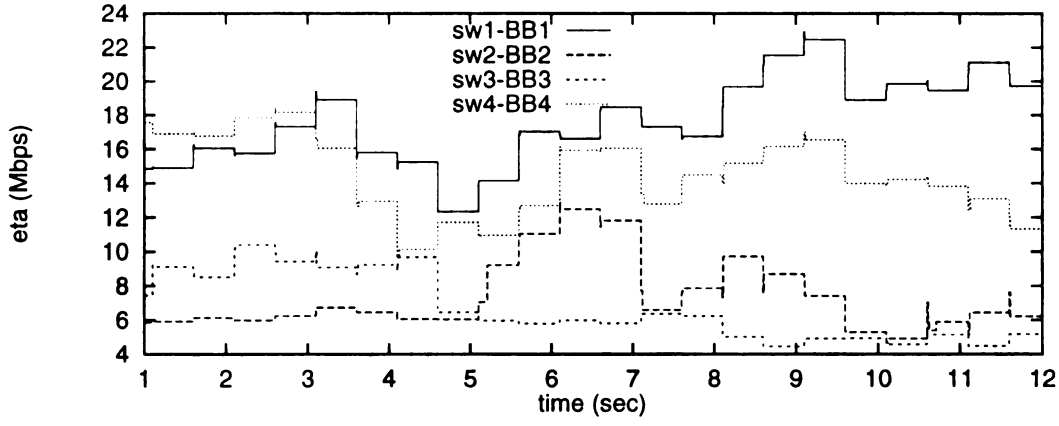


Figure 3.17: The η for multi-broadband-link network system with $\mathbf{x} = 2000$.

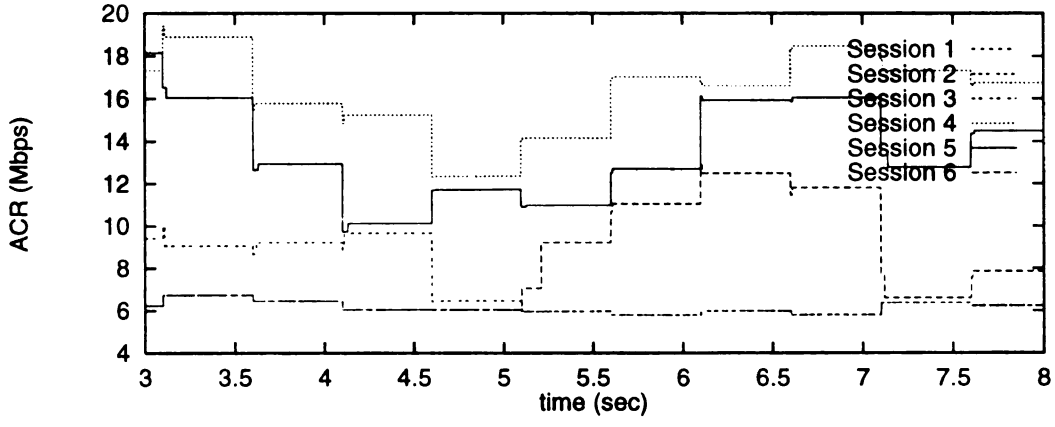


Figure 3.18: The ACRs for multi-broadband-link network system with $\mathbf{x} = 2000$.

Chapter 4

Error Control for ABR Multicast

Lossless data delivery for multicast communication (*reliable multicast*) is necessary for some applications, such as distributing the information of stock markets. In this chapter, we study how to extend the flow control of ABR multicast for reliable multicast. The output buffer for the ABR streams in the switch is implemented by a single queue, such as the buffer in a virtual path [10]. When the queue is full since the total available bandwidth for the whole ABR streams is suddenly reduced, the incoming ABR cells will be dropped, even those belonging to an ABR stream below the fairshare, and thus a loss will be detected in the downstream receiver. Therefore, though the allowed cell rate for each ABR stream will be reduced according to the max-min fairness, the buffer will not be managed in a fair manner. As a result, the cell loss may occur in the link where it is not the bottleneck for the multicast communication. The above observation leads to the idea of using the virtual connection, a partial connection, which is built by the feedback information for error control, atop the existing connection for the data

retransmission. That is, we can retransmit the data only to the receiver requesting it with carefully detecting the status of the link as well as filtering and directing the data by the switch without violating the max-min fairness.

Another advantage of using ABR flow control to carry out reliable multicast is that by allowing the *resource management* (RM) cell, originally dedicated to the flow control, carrying the information for the error control, there is no need to send any other feedback message. In addition, the fault tolerance will be coped with by the flow control algorithm and thus relieving the workload for the error control algorithm.

4.1 The Error Control Algorithm

In ATM networks, the resource management information is carried by resource management (RM) cell. To convey the error information, we let the backward RM (BRM) cell carry a non-negative integer SRX (sequence number of retransmitted packet) and an error indication bit EI. If an error is detected in the receiver, then assign SRX with the smallest sequence number of the packet needed to be retransmitted, and set EI $\leftarrow$ 1. If there is no error, then assign SRX with the number of which the receiver expects to receive next, and set EI $\leftarrow$ 0. The sequence number can be IP packet sequence number. Upon receiving a forward RM (FRM) cell, the destination will assign SRX and EI decided by the associated receiver and then turn the cell back as BRM cell.

The switch error control algorithm is the critical part for reliable multicast.

We assume all switches in the network implement explicit-rate based flow control algorithm.[1] Therefore, the congestion indication (CI) can be used to indicate the current queue situation along the path where error may occur. Upon receiving a BRM cell with EI = 1, the switch error control algorithm should configure the virtual connection accordingly. The algorithm should satisfy the following three criteria to guarantee correctness:

deadlock free: Deadlock will occur if there is a permanent wrong configuration in the virtual connection so that some receivers are always unable to get the packet they requested. For example, the receiver r_1 requests packet p_1 and receiver r_2 requests packet p_2. The virtual connection will not be changed until r_1 has received p_1. The sender will keep retransmitting p_1; however, the virtual connection always directs the packet only to r_2.

starvation free: Starvation could occur when the error information of the lost packet never reaches the source.

livelock free: Livelock could happen if the switch keeps changing the configuration of virtual connection and thus the receiver cannot get the packet it requests within a finite time even under the condition that there is no data loss after the receiver requests the retransmission.

Note that under the assumption of no data lost after the receiver requests the retransmission, if these three criteria can be guaranteed, the receiver will always

[1]The flow control algorithm of large-scale multicast communication cannot only use the binary congestion indication from all links in the connection. Otherwise, due to the unfair buffer management, the allowed cell rate (ACR) of the multicast communication will be around minimum cell rate (MCR).

get the packet it requests eventually. To satisfy above three criteria, we implement the strategy called *smallest first*. That is, the switch always picks up the possible smallest sequence number of the packet requested for retransmission as far as it knows to configure the virtual connection. Therefore, the receiver requests the packet with the smallest sequence number will eventually obtain the packet, and within finite moment, each receiver will have its requested packet with the smallest sequence number. We show a switch error control algorithm in Figure 4.1, which modifies the fourth algorithm in [15] for flow control. The constant γ is determined by the correlation of two packets being lost. The strategy smallest first is implemented in lines 5 and 21. The parameter PCR stands for peak cell rate. MER, MCI, and MNI, being PCR, zero, and zero initially, respectively, are the registers for flow control. MSRX, MEI, MCN (maximum completion number), and CCN (current configuration number), being β, zero, β, and β initially, respectively, are the registers for the error control. We assume the sequence number of any packet is less than β. For each branch e, a_e, which indicates the arrival of BRM cell, and b_e, which is for the filtration, are the binary registers with initial value zeros.

The switch error control algorithm in Figure 4.1 is conservative since it leads to the sender waiting for all receiver's requests before retransmitting the data (lines 16 and 17). Therefore, it will save the retransmission traffic but the delay of the retransmission will depend on the longest round trip time among all the pairs of source and receiver.

For every branch point v:

0 Upon receiving a BRM cell with BN=0 from branch e,
1 MER $\leftarrow \min\{\text{MER}, \text{ER}\}$; MEI $\leftarrow$ (MEI OR EI);
2 if EI $= 1$, then MSRX $\leftarrow \min\{\text{MSRX, SRX}\}$;
3 else MCN $\leftarrow \min\{\text{MCN, SRX}\}$;
4 end if;
5 $a_e \leftarrow 1$; CCN $\leftarrow \min\{\text{CCN,MSRX}\}$;
6 if EI $= 1$ and SRX $\leq$ CCN $+ \gamma$, then
7 $b_e \leftarrow 1$;
8 MCI $\leftarrow$ (MCI OR CI); MNI $\leftarrow$ (MNI OR NI);
9 else $b_e \leftarrow 0$;
10 end if;
11 if for each branch $e', a_{e'} = 1$, then
12 for each branch e' with $b_{e'} = 1$,
13 obtain CI and NI of the output buffer to e';
14 MCI $\leftarrow$ (MCI OR CI); MNI $\leftarrow$ (MNI OR NI);
15 end for;
16 if MCN $<$ MSRX, then
17 MSRX $\leftarrow$ MCN; MEI $\leftarrow 0$;
18 end if;
19 pass the BRM cell to source with
ER $\leftarrow \min\{\text{MER}, \min\{C(e') : t(e') = v\}\}$,
CI $\leftarrow$ MCI, NI $\leftarrow$ MNI, EI $\leftarrow$ MEI, and SRX $\leftarrow$ MSRX;
20 if for each branch $e', b_{e'} = 0$, then CCN $\leftarrow \beta$;
21 else CCN $\leftarrow$ MSRX;
22 end if;
23 MER $\leftarrow$ PCR; MCI $\leftarrow 0$; MNI $\leftarrow 0$;
24 MEI $\leftarrow 0$; MSRX $\leftarrow \beta$; MCN $\leftarrow \beta$;
25 for each branch $e', a_{e'} \leftarrow 0$;
26 else discard the BRM cell;
27 end if.

Figure 4.1: The switch error and flow control algorithm.

```
1  ACR'_u ← ACR_u;
2  if ACR_u > ER, then
3     ACR_u ← ACR_u + AIR;
4  end if;
5  ACR_u ← max{min{ACR_u, ER}, MCR};
6  if CI = 1, then
7     ACR_v ← ACR_v − ACR_v × RDF;
8  else
9     if q_r is empty, then
10       ACR_v ← 0;
11    else
12       ACR_v ← ACR_v + AIR − max{ACR_u − ACR'_u, 0};
13       ACR_v ← min{ACR_v, PCR − ACR_u};
14    end if;
15 end if.
```

Figure 4.2: The decision rules for ACR_u and ACR_v when a BRM cell is received. The parameters RDF and AIR stand for rate decreasing factor and ACR increasing rate, respectively.

Upon receiving a data cell, the switch with single out-going branch will send it via the branch. If the switch is a branch point, then the switch multicasts the data cell to each branch e with $b_e = 1$ if it is a virtual cell, and multicasts it to all its branches, otherwise. The destination implements two queues for reassembling, one is for virtual cells and one for the other data cells.

Upon receiving a BRM cell, the source will determine the ACR for virtual connection (ACR_v) and ACR for underlying connection (ACR_u). The goal for assigning ACR_v and ACR_u is to fully utilize the possible available bandwidth subjective to satisfying the source behavior given in [13]. If there no cells in the retransmitted queue q_r, then $ACR_v = 0$ always. The decision rules for ACR_v and ACR_u as shown in Figure 4.2. When the received SRX in the source is larger than the previous one, the source returns this number to the sender along with EI. When sending an FRM cell, the source always assigns the field *current cell rate*

(CCR) with ACR_u.

Upon receiving SRX and EI from the source, the outstanding packets with sequence number smaller than SRX in the sender will be taken out from the list in the sender. In addition, when EI = 1, then cells corresponding to packets with sequence number in [SRX,SRX+γ] will be put into q_r in the source if the packet with sequence number SRX has not been retransmitted. These packets are called *retransmitting packets.* Before putting the cells corresponding to transmitting packets to q_r, the timer will be reset and old cells in q_r will be discarded first. When a timeout occurs, if the last EI = 1 and SRX is larger than the sequence number of the first retransmitting packet at the point that the timer was reset in the last time, then the cells corresponding to each packet with sequence number in [SRX,SRX+γ] will be inserted into q_r. Otherwise, the cells corresponding to each packet with number in [SRX,SRX+γ] will be inserted into q_n. In both cases, the timer will be reset again.

We allow a retransmitted packet waiting in the q_r being preempted to the normal queue q_n if there is no cells in q_n. The preemptive operation is not allowed in q_n. The cells corresponding to the packets in q_r and those in q_n will be sent via the virtual connection and underlying connection, respectively. In addition, there is a timer associated with q_r. If the length of q_r jumps from zero, then the timer is reset. When timeout occurs, every cells in q_r will be discarded. Therefore, if network keeps congested, it is highly possible that many receivers will not receive the corresponding packets and using underlying connection will be better than using virtual connection. We show the service model in the source in Figure 4.3.

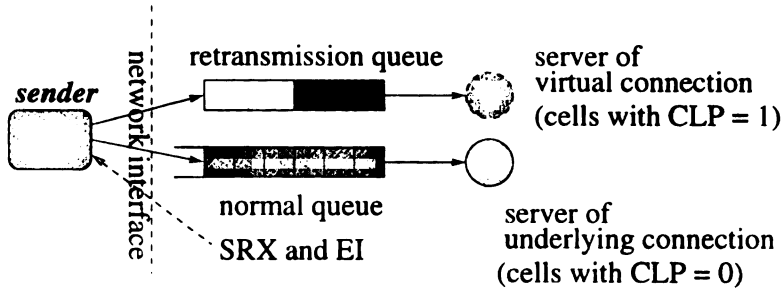


Figure 4.3: The service model in the source.

parameter	value	parameter	value
PCR	149.76Mpbs	MCR	0.1 Mpbs
RIF	1/16	RDF	1/16
buffer size	3000 (cells)	Nrm	32 (cells)
packet size	23 (cells)	LT	1000 (cells)
ADTF	200ms	HT	2500 (cells)

Table 4.1: The important parameters for the simulation. When there are thirty-two cells in underlying connection being sent after the last RM cell, a new RM cell will be sent. We use ADTF (ACR Decrease Time Factor) as the sleep time of the timer in both source and sender.

4.2 Simulation

We modify the NIST switch algorithm [39] in the NIST simulator 3.0 to support reliable ABR multicast. The important parameters for our simulation are listed in Table 4.1. Figure 4.4 shows the network configuration for the simulation. The pair

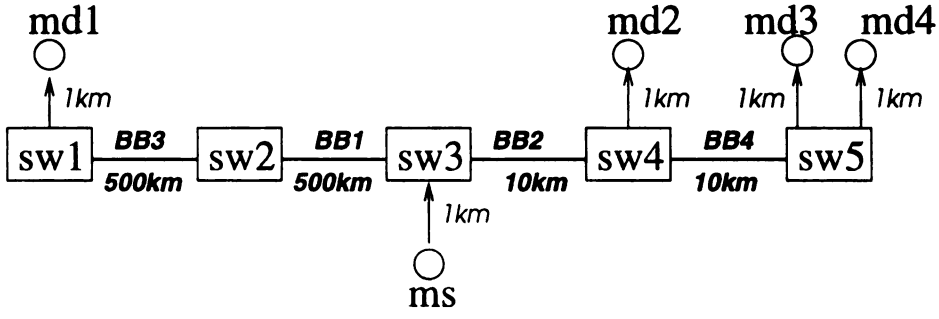


Figure 4.4: The network configuration. The multicast session has source ms and three destinations $md1, md2, md3$. In addition, for each broadband link BBj, there exists another ABR session sj sharing the same link with the multicast session.

"ms" to "md1" will have the longest round trip time. For each link BBj, where $j \in \{1, 2, 3, 4\}$, the capacity is 149.76 Mbps and a VBR traffic with mean 80 Mpbs and timegrain 2 milliseconds, as well as another ABR stream will flow through it. Therefore, the capacity of each will be almost fully utilized.

We compare two different retransmission mechanisms. One uses q_r, q_n, virtual and underlying connections for retransmission, and the other only uses q_n, i.e., always let the length of $q_r = 0$ and multicasts the retransmission cells to all the receivers via underlying connection. When $\gamma = 8$ and 16, the retransmission at one time consists of nine and seventeen packets, respectively. The error control protocol in the receiver implements the selective repeats and each lost packet detected in the receiver will be requested for retransmission. In addition, let *completion number* (CN) be the last SRX observed in the sender and *progress number* (PN) be the largest sequence number of the outstanding packet in the sender. When retransmission is needed and PN > CN + W, we always send the retransmission data via underlying connection. This is because if the round trip time is long, the cells in virtual connection tend to be discarded due to the congestion, and thus using underlying connection for retransmission is better.

We show the simulation results when using the BCLR flow control. The size of each output buffer is 3000 cells. The parameters HT and LT are 2500 cells and 1000 cells, respectively. If the current queue length in an output port of a switch is not smaller than HT or the number of cells with CLP =1 is not smaller than LT, then CI in the BRM cell will be set to one. The switch algorithm and decision rules for ACR_u and ACR_v remain the same. When a cell with CLP = 1 arrives,

it will be allowed to enter into the buffer in the output port of a switch if there is some empty space and the number of cells with CLP =1 is less than LT. The cell with CLP = 0 will enter into the buffer if there is some empty space. The upper bound of cell loss ratio is set to be 20% and by the experience, the exact upper bound will be reduced to 1%. Figure 4.5, 4.6, 4.7, 4.8 show the results. We see that when W is small, the virtual connection is not effective. In addition, when W is large, the virtual connection will affect the loss ratio (the CLR for sw3 in Figure 4.8). This effect can be reduced by decreasing the value of HT.

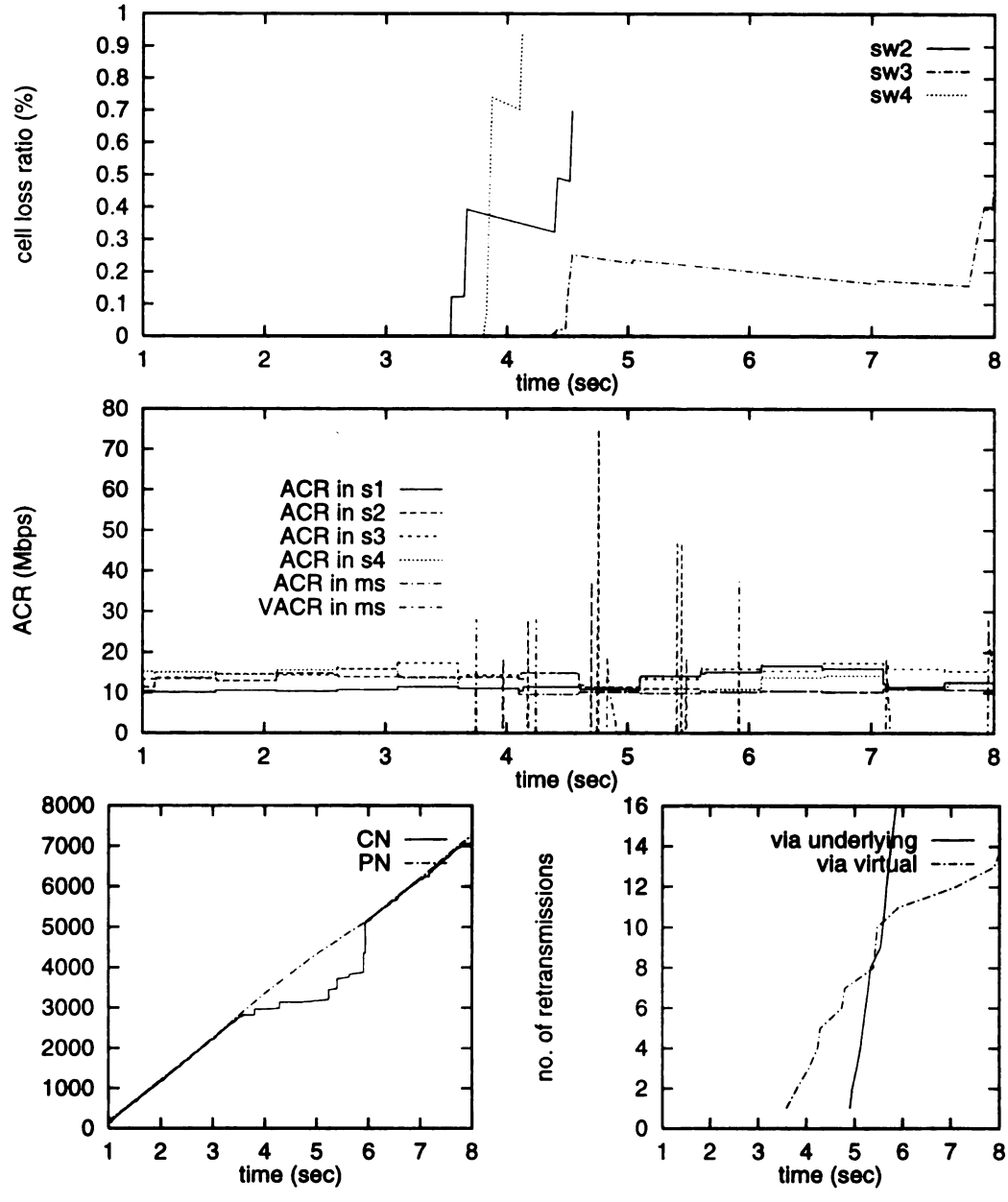


Figure 4.5: The simulation results using BCLR flow control when $\gamma = 8$ and $W = 100$.

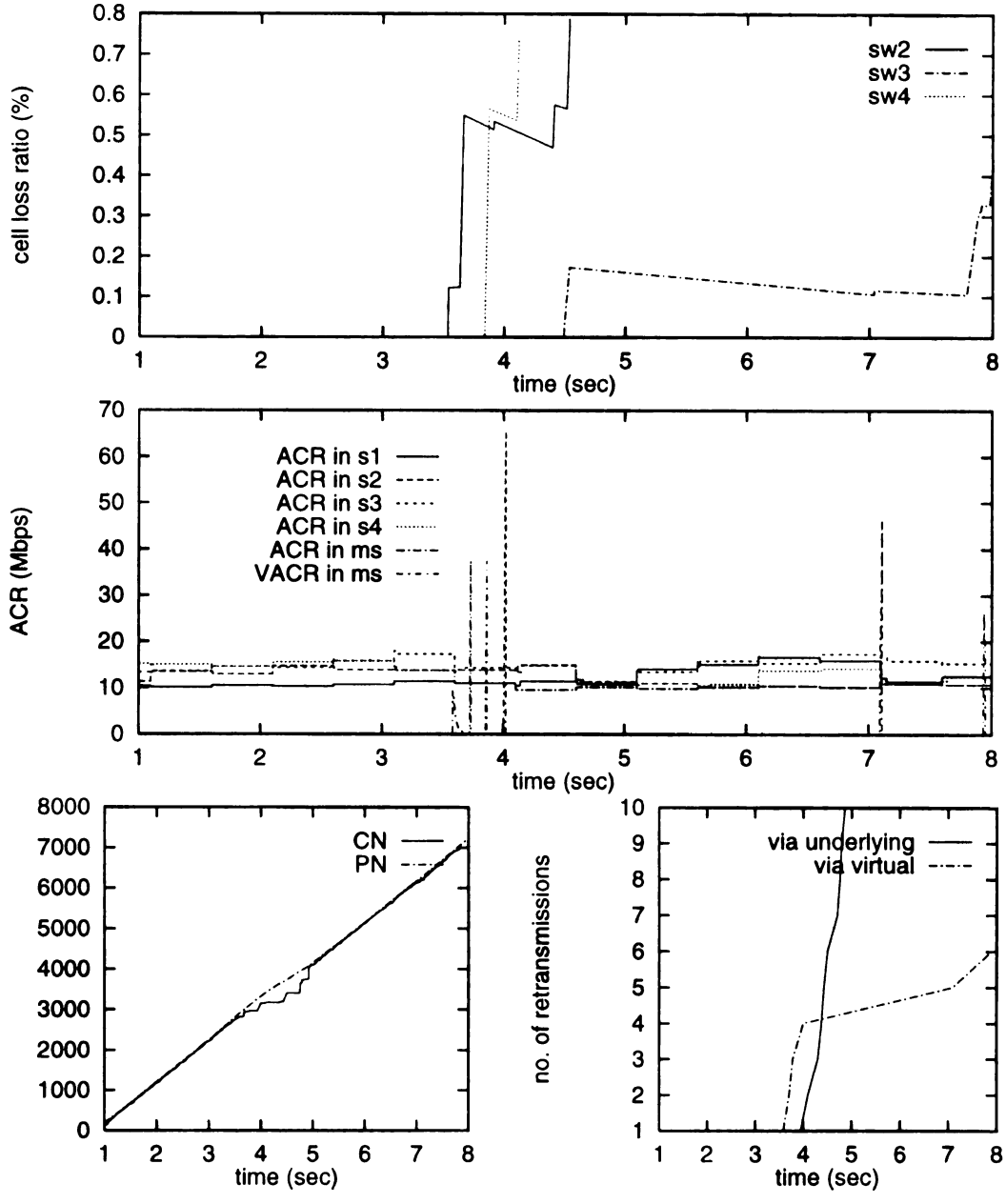


Figure 4.6: The simulation results using BCLR flow control when $\gamma = 16$ and $W = 100$.

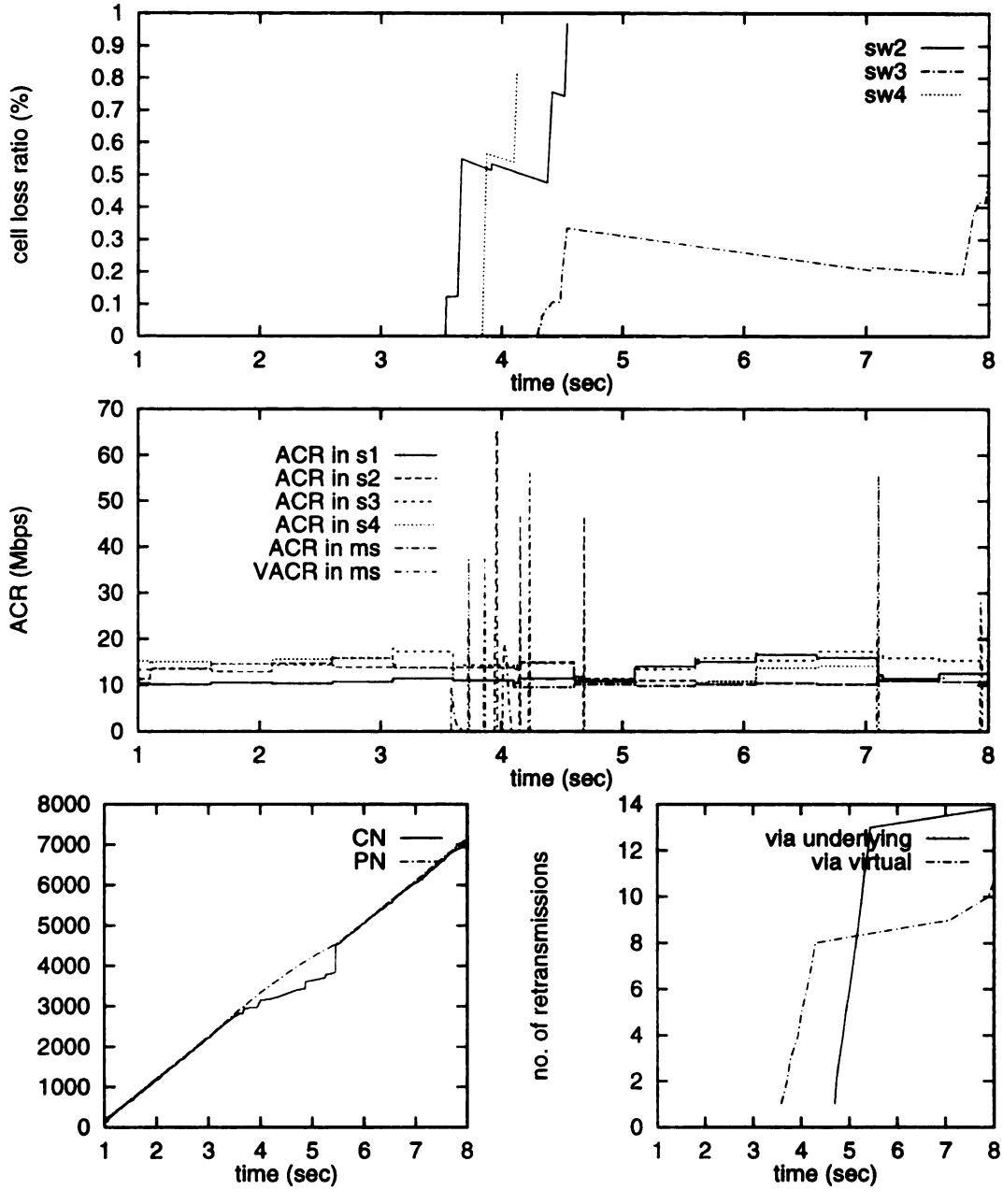


Figure 4.7: The simulation results using BCLR flow control when $\gamma = 16$ and $W = 500$.

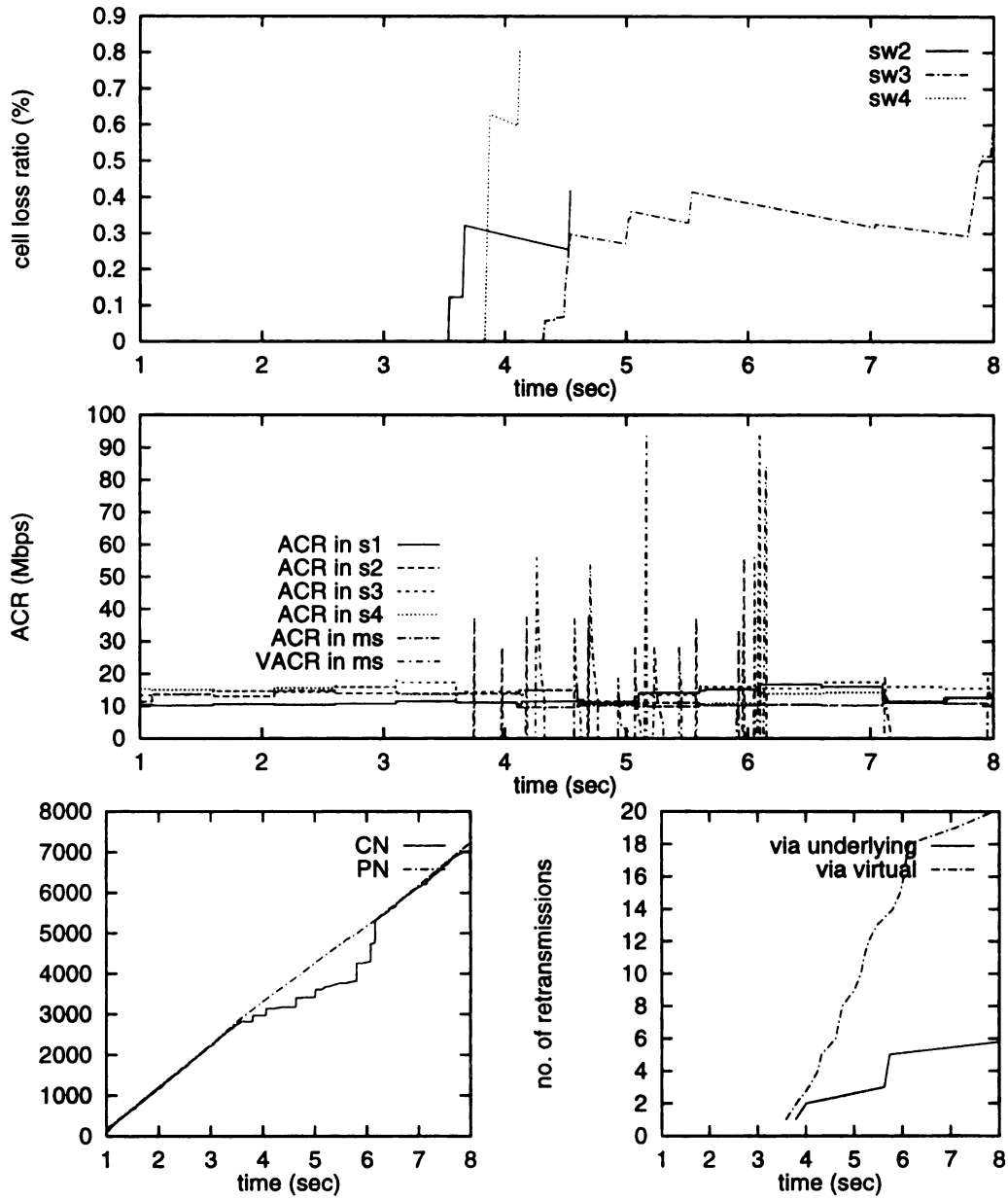


Figure 4.8: The simulation results using BCLR flow control when $\gamma = 16$ and $W = 1000$.

Chapter 5

Flow Control for ABR Dispersity Multicasting

Dispersity routing was first defined by Maxemchuk [50,51]. In dispersity routing, a message is divided into a number of submessages, which are transmitted in parallel over m link-disjoint paths. Due to using link-disjoint paths, fault tolerance is then an inherent property. In addition, as the studies indicated, dispersity routing would essentially equalize the network load and increase the overall network utilization. A survey can be found in [52].

Dispersity multicasting is an extension of dispersity routing to *multicast* (point-to-multipoint) communication. A multicast communication usually uses a subtree with the source as the root and connecting all the destinations to carry the messages. Thus in dispersity multicasting, we consider m arc-disjoint subtrees. We call each subtree as a connection.

Consider the dispersity multicasting for the ABR service (*ABR dispersity mul-*

ticasting) with two arc-disjoint connections shown in Figure 5.1. Each of these two arc-disjoint connections is referred to as an *underlying connection.* As shown in Figure 5.1.(a), links ab, bd, be belong to one underlying connection and links ac, cd, ce belong to the other. In addition, the available bandwidth of each link is shown along with the link. Suppose the bottleneck flow control for multicast communication described above is applied to each underlying connection. Thus, sending rates 6 and 8 are granted for the underlying connections, respectively. For each link, the sending rate of the corresponding underlying connection will be consumed by the connection, as shown in Figure 5.1.(b), and let the *residual bandwidth* be the result of substracting available bandwidth from the sending rate for the corresponding connection. Let the *virtual network* be the network induced by all the links in the underlying connections. Moreover, let the available bandwidth of each link in the virtual network be the residual bandwidth of that link. Another connection, i.e., a subtree, called *virtual connection*, can possibly be established on the virtual network. Figure 5.1.(c) shows a virtual connection using links ab, be, ac, cd with sending rate being one. Figure 5.1.(d) shows the final configuration. The concept of virtual connection can be extended for m underlying connections and n virtual connections for multicast communication where $m \geq 2$ and $n \geq 1$.

In this chapter, we study the ABR dispersity multicasting. A flow control algorithm is extended from the fourth algorithms in [15, 16] to extract a virtual connection for the case in which we have two underlying connections. We also discuss the issues to make the algorithm viable on ATM networks. Simulation

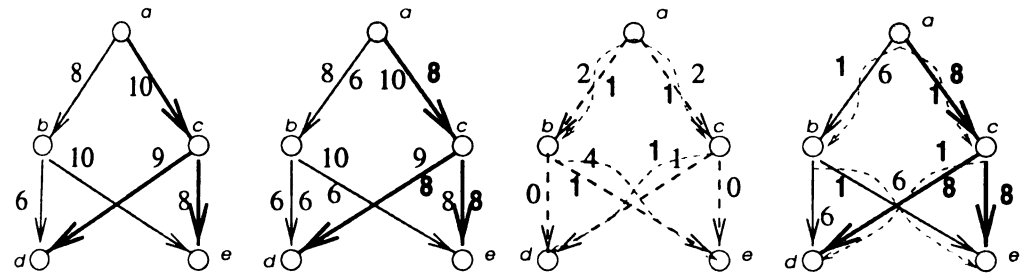


Figure 5.1: The concept of virtual connection.

studies show that in general, our flow control algorithm enhances the throughput of dispersity multicasting.

5.1 Flow Control Algorithm

In ATM networks, the management information is carried by *resource management* (RM) cells [13]. Consider the ABR multicasting with a single connection $T(V, A, C)$. As a simplified version of the fourth algorithms in [15, 16], to obtain the connection bandwidth, the source generates a forward RM (FRM) cell with the field *explicit rate* (ER) being *peak cell rate* (PCR). Upon receiving an FRM cell, the switch multicasts the cell to all its branches. Then an operation $\Lambda_1(T)$, defined in Figure 5.2, is performed. Upon receiving a backward RM (BRM) cell, the source adjusts its *allowed cell rate* (ACR) according to ER in the cell.

Consider the ABR dispersity multicasting with two underlying connections and one virtual connection. Each underlying connection has its own virtual channel identifier (VCI). Cells in the virtual connection use the VCI assigned to the underlying connection where these cells will traverse. Thus, each path from the source to

For every destination:
  Upon receiving an FRM cell,
    return the cell as BRM cell to source.
For every switch v**:**
  Upon receiving a BRM cell with ER from branch e,
    MER $\leftarrow$ min{MER,ER};
    $a_e \leftarrow 1$;
    if for each branch e', $a_{e'} = 1$, then
      pass the BRM cell to source with ER$\leftarrow$
        min{MER,min$\{C(e') : e'$ is a branch of $v\}$};
      MER$\leftarrow$ PCR;
      for each branch e', $a_{e'} \leftarrow 0$;
    else discard the BRM cell.

Figure 5.2: The defintion of $\Lambda_1(T)$, where T is the rooted tree for backward RM (BRM) cells traversing backwards, MER, being PCR initially, is a register in the switch and a_e, being zero initially, is a binary variable associated with the branch e.

field	representation
CCR	ACR parameter of the underlying connection
ER	the path bandwidth
$\mathcal{ER}_t$	relating to the total bandwidth of the underlying and virtual connections
$\mathcal{ER}_u$	relating to bandwidth of underlying connection

Table 5.1: The representation of fields in RM cells.

a destination in the virtual connection must be in the same underlying connection. The FRM and BRM cells carry $\langle$CCR,ER$\rangle$ and $\underline{ER} = \langle \mathcal{ER}_t, \mathcal{ER}_u \rangle$, respectively.[1] In the followings, the flow control for the virtual connection is only driven by the BRM cell which is turned around by the destination. That is, the BN field of the BRM cell should be zero. Table 5.1 shows the representation of each field.

The idea is letting the destination choose the path in the virtual connection and the switch treat the underlying connections independently. For each underlying

[1]In next section, we will discuss the issue when CCR and ER in the BRM cell are used for $\mathcal{ER}_t$ and $\mathcal{ER}_u$, respectively.

connection T_i, FRM cells are constantly sent from the source with CCR and ER being ACR for T_i and PCR, respectively, via T_i to all destinations. Upon receiving an FRM cell via T_i, the switch multicasts the cell to all its branches in T_i. Before delivering the copy of the FRM cell to the branch e, the switch updates ER by $\min\{\mathrm{ER}, C(e)\}$ in the copy of the cell.

For each i, the destination maintains $\underline{y^i} = \langle \epsilon^i, ER^i \rangle$, where ϵ^i and ER^i denote the path bandwidths from the source via T_i to the destination in the virtual network and original network, respectively. Upon receiving an FRM cell via T_i, the destination updates $\underline{y^i}$ by the rule:

$$\langle \epsilon^i, ER^i \rangle \leftarrow \langle (\text{ER-CCR})^+, \text{ER} \rangle , \tag{5.1}$$

where $(\cdot)^+ \equiv \max\{0, \cdot\}$. An operation $\Lambda_2(T_i)$ defined in Figure 5.3, which is extended from Λ_1, is then performed.

The descriptions of terms used in Figure 5.3 are as follows: T_i is the rooted tree for BRM cells traversing backwards. N is a number which is larger than PCR. The indicator $I_{(\cdot)} = 1$ if $(\cdot)$ is true or $I_{(\cdot)} = 0$ otherwise. $\underline{MER} = \langle M\mathcal{ER}_t, M\mathcal{ER}_u \rangle$ being $\langle N, \text{PCR} \rangle$ initially, is a register for T_i in the switch. The register a_e and b_e are binary variables associated with the branch e with initial value zeros, respectively.

Upon receiving a BRM cell from T_i, the decision of the ACR for T_i (ACR^i) follows the source behavior of an ordinary ABR service by replacing ER with $\mathcal{ER}_u$. In this section, we let $ACR^i \leftarrow \mathcal{ER}_u$. Then the ACR for the virtual connection

For every destination:
Upon updating y^i by Equation 5.1,
if $i = 1$, then return a BRM cell with $\underline{ER} \leftarrow$
$\langle ER^1 \times I_{(\epsilon^1 \geq \epsilon^2)} + N \times I_{(\epsilon^1 < \epsilon^2)}, ER^1 \rangle$;
if $i = 2$, then return a BRM cell with $\underline{ER} \leftarrow$
$\langle N \times I_{(\epsilon^1 \geq \epsilon^2)} + ER^2 \times I_{(\epsilon^1 < \epsilon^2)}, ER^2 \rangle$.
//If $\mathcal{ER}_t = N$, the path from the source to the destination
//via T_i is chosen not to be in the virtual connection.
For every switch v:
Upon receiving a BRM cell from branch e in T_i,
$\underline{MER} \leftarrow \langle \min\{M\mathcal{ER}_t, \mathcal{ER}_t\}, \min\{M\mathcal{ER}_u, \mathcal{ER}_u\} \rangle$;
$a_e \leftarrow 1$;
$b_e \leftarrow I_{(\mathcal{ER}_t \leq \text{PCR})}$;
if for each branch e' in T_i, $a_{e'} = 1$, then
pass the BRM cell to source with $\underline{ER} \leftarrow$
$\langle M\mathcal{ER}_t, \min\{M\mathcal{ER}_u, \min\{C(e') : e' \text{ is a branch of } v \text{ in } T_i\}\} \rangle$;
$\underline{MER} \leftarrow \langle N, \text{PCR} \rangle$;
for each branch e' in T_i, $a_{e'} \leftarrow 0$;
else discard the BRM cell.

Figure 5.3: The defintion of $\Lambda_2(T_i)$.

(ACR_v) is decided by

$$ACR_v \leftarrow \phi(ACR_v, \min_{i \in \{1,2\}} (\mathcal{ER}_t^i - ACR^i)^+), \tag{5.2}$$

where $\mathcal{ER}_t^i$ stores $\mathcal{ER}_t$ in the most recent BRM cell from T_i, and $\phi : [0, \infty) \times [0, \infty) \rightarrow [0, \infty)$. In this section, we use the function $\phi(a, b) = b, \forall a, b \geq 0$.

A file to be transferred is dynamically configured to three groups which will be sent via two underlying connections and one virtual connection, respectively, and call data cells sent via the latter one *virtual cells*. Upon receiving a data cell via T_i, the switch performs the *filtration* by multicasting the cell to every branch e in T_i with $b_e = 1$, if it is a virtual cell; otherwise, the switch multicasts the cell to all its branches in T_i.

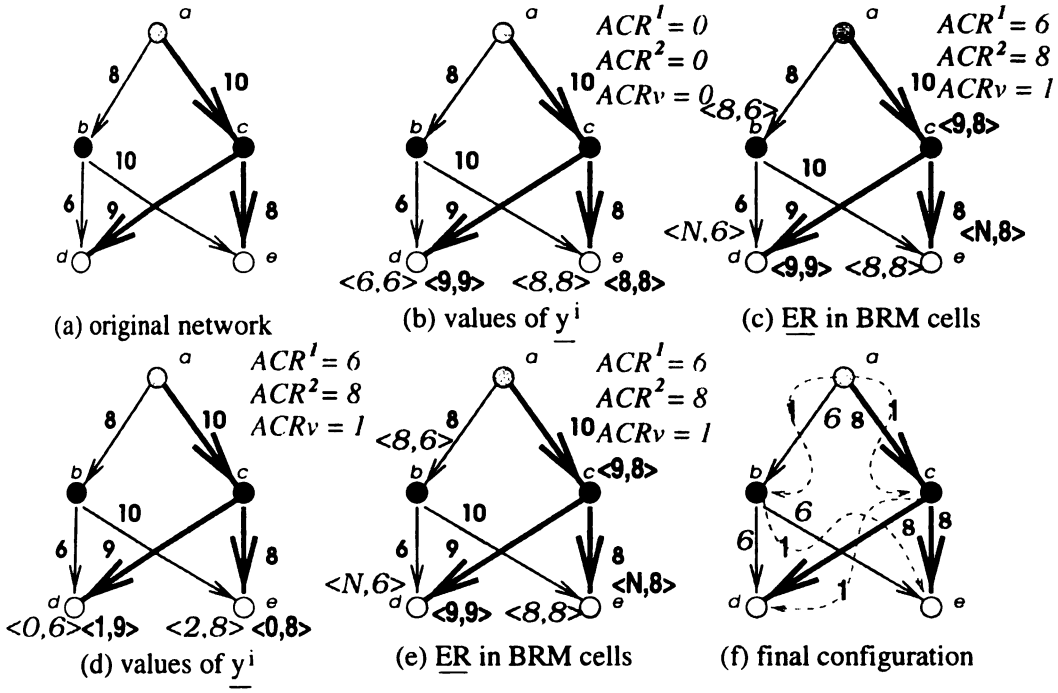


Figure 5.4: The example illustrating the algorithm.

5.1.1 Example

Figure 5.4 shows an example. In part (a), let T_1 and T_2 be the rooted trees induced by the arc sets $\{ab, bd, be\}$ and $\{ac, cd, ce\}$, respectively. In part (b), FRM cells have arrived at each destination via T_1 and T_2, and $\forall i, \underline{y^i}$ in each destination has been updated by Equation 5.1. In part (c), paths ab, be and ac, cd are chosen in the virtual connection and $\forall i, \Lambda_2(T_i)$ is performed. By Equation 5.2, $ACR_v = 1$. In part (d), another FRM cells have arrived at each destination via T_1 and T_2, and $\forall i$, a new $\underline{y^i}$ in each destination has been obtained by Equation 5.1. In part (e), again, ab, be and ac, cd are chosen as paths in the virtual connection and $\forall i, \Lambda_2(T_i)$ is performed. By Equation 5.2, $ACR_v = 1$. Part (f) shows the final configuration.

5.1.2 Correctness of Algorithm

A sequence of times $S_n, n \geq 0$ is defined by $S_0 = 0$ and when $n \geq 1$, S_n is the smallest time at which $\forall i$, the first FRM cell sent after S_{n-1}, from the source via T_i, has arrived at all the destinations and the corresponding operation $\Lambda_2(T_i)$ has been performed completely.

Theorem 7 *Suppose the available bandwidth of each link is static, and the durations for source and destination operations can be ignored. Then we have 1) at S_1, the bandwidth of underlying connections will be obtained; 2) the bandwidth of the virtual connection will be obtained at S_3; and 3) after S_3, all bandwidths obtained will not be changed.* □

Proof: The statement 1 is true since the operation for obtaining bandwidths for each underlying connections is the reduction operation $R(T(V, A, C), ER_u, MER_u, PCR)$ for each underlying connection T where PCR is the constant function with value PCR. Note that $\forall e \in A(T), C(e)$ is the available bandwidth along e in the original network. The statement 2 can be easily verified since after S_1, the bandwidths for both underlying connections are known in the source, and before S_2, the information of CCRs will be obtained in the destination. Then correct ER_t^1 and ER_t^2 in the destination can be obtained by (5.1).

Suppose the correct bandwidth for the virtual connection is $\beta \geq 0$. After S_2, there exists an underlying connection T, by the reduction operation $R(T(V, A, C), ER_t, MER_t, \Gamma)$ where $\forall e \in A(T), C(e) = L$, and Γ stands for the

results obtained by (5.1), $\beta + ACR_u$ will be obtained where ACR_u is the ACR for the underlying connection. The result for the other connection should be $\beta' + ACR'_u$ where $\beta' \geq \beta$ and ACR'_u is the ACR for the underlying connection. Therefore, by the source operation specified in Equation 5.2, the correct bandwidth will be ACR_v in the source. This operation will be completed at S_3. The statement 4 can be easily checked. □

5.2 Implementation Issues

To support our algorithm described in the last section on the ATM networks, the switch must distinguish the data cell to be virtual cell or not, so that the filtration can only be applied to virtual cells. Here we propose that the CLP field in the data cell for the ABR connection will be used to provide the filtration. That is, in our case, virtual cells are with CLP = 1. Therefore, the switch can filter out virtual cells if needed.

For the BRM cell, CCR and ER could be used for $\mathcal{ER}_t$ and $\mathcal{ER}_u$, respectively. However, CCR in the BRM cell is not allowed to change by the destination or switch in the current ATM traffic management specification [13]. It is noteworthy that if *ABR multipoint-to-point connections* are also supported, fairness cannot be achieved if CCR in the BRM cell is used to calculate the available bandwdith [53]. Thus, we expect in the future a more flexible usage of CCR in the BRM cell will be defined.

The frequency of RM cells has an impact on the cell loss ratio, bandwidth utilization, and fairness. In our current approach, the frequency of the RM cells is only decided by the parameters, such as the number of cells, for the underlying connection. Thus the frequency of the RM cells is consistent for the underlying connection.

5.3 Simulation

We modify the NIST switch algorithm [39] in the NIST simulator 3.0 to support ABR dispersity multicasting. In addition, upon receiving a BRM cell, the switch does MCI $\leftarrow$ MCI OR CI and MNI $\leftarrow$ MNI OR NI, where CI (*congestion indication bit*) and NI (*no increase bit*) are the fields in BRM cell, and MCI and MNI, initially zeros, are the registers in the switch. If the BRM cell will be passed to the source, then let CI and NI be MCI and MNI, respectively, and reset MCI and MNI to be zeros.

Figure 5.5 shows the network configuration for the first simulation. The distance of BBj for $j \in \{1,2,3,4\}$ is 500 km and the distance of any other link is 10 km. Each link is associated with 155 Mbps bandwidth. The source of the ABR dispersity multicasting, "ms"", has two destination ends "$md1$" and "$md2$". The first underlying connection uses links $BB2$ and $BB3$ and the other one uses links $BB1$ and $BB4$. For each i, The decision of ACR^i follows a normal ABR source behavior and the decision of ACR_v uses the function $\phi(a,b) = \min\{a+\text{RIF}\times\text{PCR}, b\}I_{(a<b)} + \min\{a(1-\text{RDF}), b\}I_{(a\geq b)}, \forall a,b \geq 0$, where

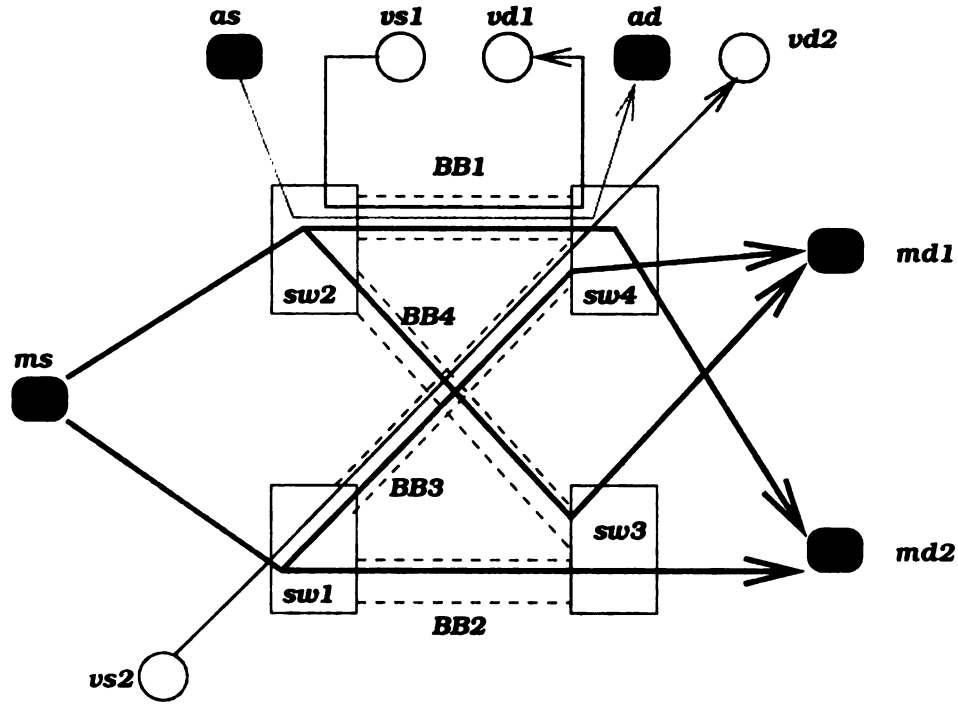


Figure 5.5: The first network configuration.

parameters RIF and RDF for all ABR sources are 0.0625, and PCR = 155 Mbps. In addition, $ACR_v = 0$ initially. Another ABR connection with source "as" and destination "ad" uses the link $BB1$. Two VBR connections, "$vs1$" through $BB1$ to "$vd1$" and "$vs2$" through $BB3$ to "$vd2$" also participate in the simulation. Each VBR traffic has mean rate 80 Mbps, timegrain 1 millisecond, and the *Hurst parameter*, the intensity of the long-memory dependence, 0.7. The virtual connection is expected to use links $BB1$ and $BB3$.

In Figure 5.6 and Figure 5.7, the bandwidths of $BB1$ and $BB3$, where there is no virtual cell flowing, are highly utilized. In Figure 5.8 which shows the bandwidth utilization of $BB2$ and $BB4$, we observe the maximum of them will be close to the link capacity. This indicates that the virtual connection will highly utilize the residual bandwidth left by the underlying connection. In this simulation, no cell

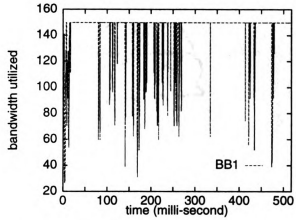


Figure 5.6: The bandwidth utilization of $BB1$ in the first simulation.

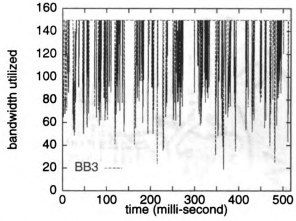


Figure 5.7: The bandwidth utilization of $BB3$ in the first simulation.

loss occurs in each switch and the virtual connection is not dynamically changed. It is noteworthy that in Figure 5.9, the bandwidth utilization of link from "$sw4$" to "ad" is almost equivalent to that from "$sw4$" to "$md2$". This indicates that the fairness of the link outside the virtual connection ($BB1$ in this case) will not be affected.

After adding another VBR traffic through link $BB2$ to Figure 5.5, Figure 5.10 shows the network configuration for our second simulation. In Λ_2, the destination always choose the path in the virtual connection with larger ϵ^i. In this simulation, the destination swaps from current path, say via T_i, to the path via T_j only when

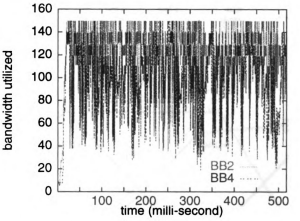


Figure 5.8: The bandwidth utilization of $BB2$ and $BB4$ in the first simulation.

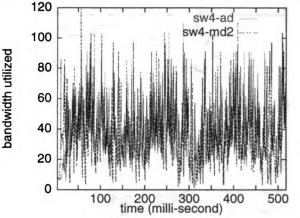


Figure 5.9: The bandwidth utilization of links "$sw4$"–"ad" and "$sw4$"–"$md2$" in the first simulation.

$\epsilon^j > \epsilon^i + \tau$. We expect that there is no path swap for "$md1$" since the virtual connection will always use link $BB4$. We check if the cell is consequently received in "$md2$". If not, a jump is noted. As shown in Table 5.2, as τ is increased, the number of path swaps and the number of jumps are decreased. Note that we can also decrease the number of jumps by returning a BRM cell with $\mathcal{ER}_t = 0$ via the currently chosen path in the destination when a path swap is needed, and then Λ_2 is performed upon receiving next set of FRM cells. As a result, $ACR_v = 0$ and then ACR_v will be slowly increased. Our future work will study the action and

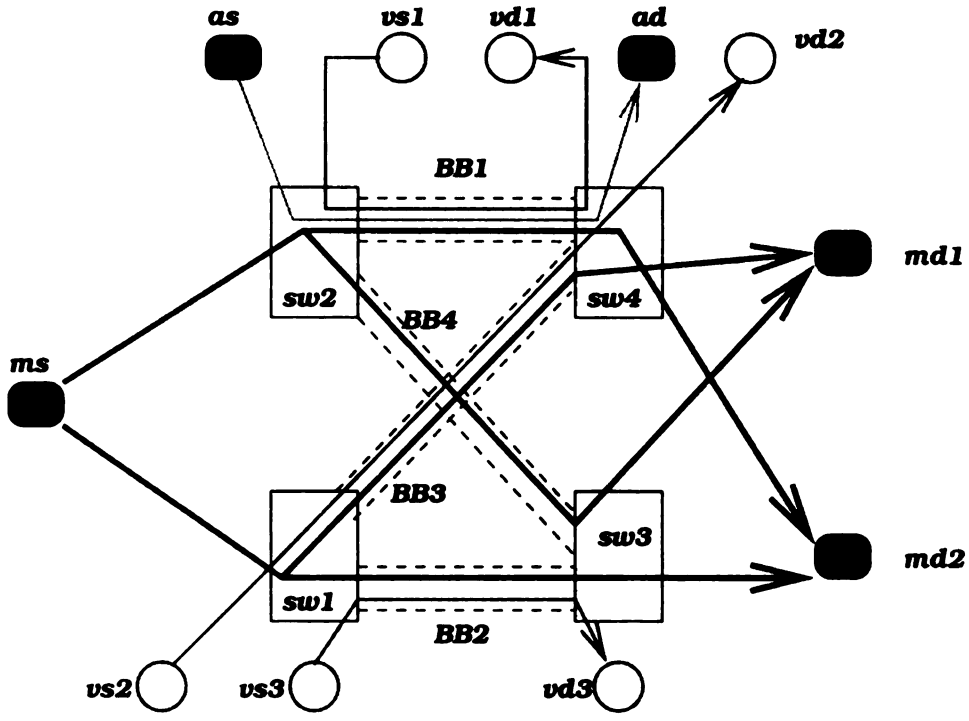


Figure 5.10: The second network configuration.

scheme	received cells	virtual cells	jumps	path swaps
w/o VC	507340	-	-	-
$\tau = 0$	569310	83562	318	613
$\tau = 2$Mbps	563877	76208	291	451
$\tau = 4$Mbps	565949	76252	244	419
$\tau = 8$Mbps	572198	82861	166	324

Table 5.2: The results in "$md2$" in the second simulation where the simulation time is 3 seconds.

condition for path swapping in a more rigorous way.

Figure 5.11 shows the network configuration for our third simulation. In this simulation, we add another VBR traffic to link $BB4$. We expect both "$md1$" and "$md2$" will have path swaps for the virtual connection. The simulation results for $\tau = 8$ Mbps listed in Table 5.3 show that we still can have more than 8% performance improvement by using virtual connection. Hence, though for each destination, the expected available bandwidths of all paths via one of the under-

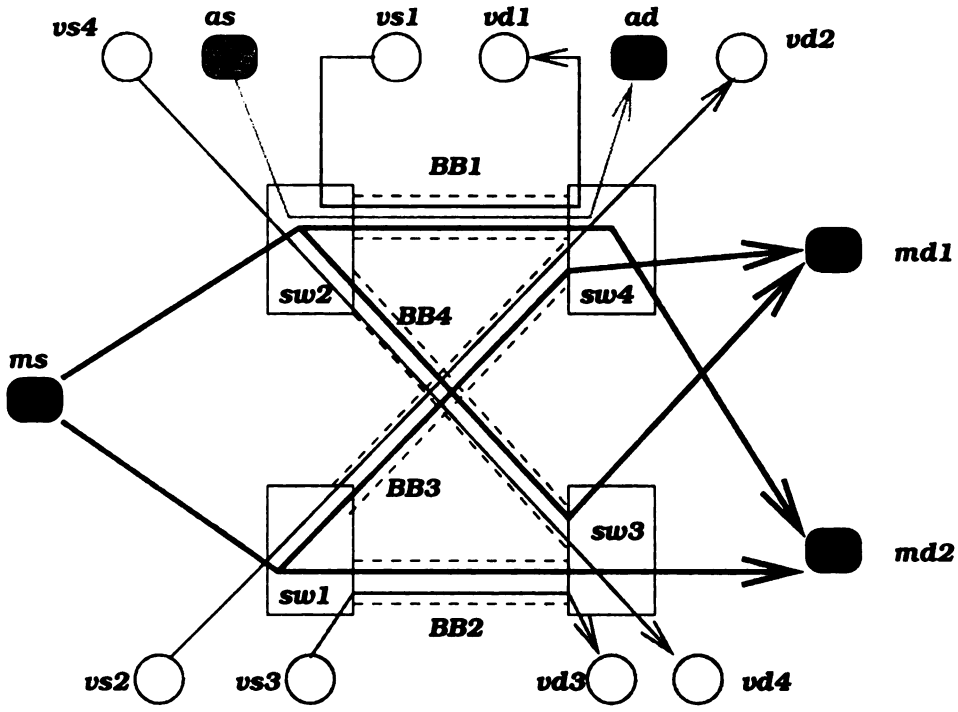


Figure 5.11: The third network configuration.

	$\tau = 8$ Mbps		w/o VC	
	$md1$	$md2$	$md1$	$md2$
cells received	518201	513991	473139	473028
virtual cells	62166	58275	-	-
jumps	155	181	-	-
path swaps	230	280	-	-

Table 5.3: The results in the third simulation where simulation time is 3 seconds.

lying connections are the same (75Mbps in this case), the virtual connection can still benefit the dispersity multicasting.

Chapter 6

Conclusion

In Chapter 3, we focus on the max-min fairness convergence when using ER-based ABR flow control. In designing a flow control mechanism, the max-min fairness convergence is a critical constraint. Up to date, it is an open issue to develop an efficient switch algorithm conforming to the standard defined by the ATM Forum to guarantee the convergence of max-min fairness with the fairness criterion "maximum of MCR and equal share." We give some sufficient conditions for max-min fairness convergence. The tractability of this sufficient conditions are verified by giving three switch algorithms, one of which has finite-time max-min fairness convergence with computation complexity $O(N)$. One of the future work will be extending our sufficient conditions for the weighted fairness criteria. Another future work is to prove or disprove that there exists an algorithm with computation complexity $O(1)$ which has finite-time max-min fairness convergence.

We give an algorithm in Figure 3.5 to bound CLR for cells with CLP $= 0$ under the Gaussian process assumption. The convergence is proved and a buffer

management scheme for BCLR flow control with two loss priorities is proposed. Our simulation shows that our algorithm bounds CLR successfully, and has good transient response and high link utilization.

From our simulation, the ratio of our given upper bound for CLR and actual CLR is around twenty. Therefore, there is some rich information in the traffic we have not explored yet. It will be an interesting topic for our future study. Another important work is to study the loss probability of a cell if its preceding cell is lost. Thus, the QoS will be improved if we can bound this conditional loss probability. In addition, this work will help us to develop an algorithm to bound packet loss ratio.

In Chapter 5, we extend the flow control of ABR multicast for error control. By using the virtual connection, a dynamically configured partial connection by error control algorithm, we can reduce the retransmission traffic. In addition, the throughput of the multicast communication can be enhanced.

The error control algorithm can be improved under the criterion of reducing the feedback delay. That is, though the receiver with longer round trip time has not informed the switch about its most recent status, the switch can still advise the sender to retransmit the packet according to the current information. Therefore, the receiver with shorter round trip time will recover its lost packets first and if the receiver with longer round trip time requests the packets with the same numbers, the sender will retransmit them again. The prospect of this approach depends on the insight that the less the number of destinations is, the more the available bandwidth tends to be granted. We will study the improved error control algorithm

as the future work.

For the fault tolerance, we propose a flow control algorithm for ABR dispersity multicasting with two underlying connections and one virtual connection. The simulation results show that our flow control algorithm can enhance the throughput of ABR dispersity multicasting under a variety of conditions. The current flow control algorithm can be improved in many respects, e.g., the generalization of the concept of virtual connection to *virtual source/virtual destination* (VS/VD) control [13], the prediction of the path bandwidth, and the conditions for path swapping. We will examine the improved algorithm by the criteria of fairness, scalability, stability, and transient response.

Bibliography

[1] B. Waxman, "Performance evaluation of multipoint routing algorithm," in *IEEE INFOCOM'93*, (San Francisco), pp. 980–986, 1993.

[2] P. Hwang and Y. Tanaka, "Multicast routing based on predicted traffic statistics," *IEICE Trans. Commun.*, vol. E77-B, Oct. 1994.

[3] M. J. Donahoo and E. W. Zegura, "Core migration for dynamic multicast routing," in *ICCCN'96*, Sept. 1996.

[4] R.-H. Hwang, "Adaptive multicast routing in single rate loss networks," in *IEEE INFOCOM'95*, (Boston), pp. 571–578, 1995.

[5] S.-B. Kim, "An optimal VP-based multicast routing in ATM networks," in *IEEE INFOCOM'96*, (San Francisco), pp. 1302–1309, 1996.

[6] M. Ammar, S. Cheung, and C. Scoglio, "Routing multipoint connections using virtual paths in an ATM network," in *IEEE INFOCOM'93*, (San Francisco), pp. 98–105, 1993.

[7] B. Waxman, "Routing of multipoint connection," *IEEE Journal on Sel. Areas in Comm.*, vol. 9, pp. 1617–1622, Dec. 1988.

[8] M. Grossglauser and K. Ramakrishnan, "SEAM: Scalable and efficient ATM multicast," in *IEEE INFOCOM'97*, (Kobe, Japan), 1997.

[9] M. Veeraraghavan, T. L. Porta, and W. Lai, "An alternative approach to call/connection control in broadband switching systems," *IEEE Comm. Mag.*, vol. 33, pp. 90–96, Nov. 1995.

[10] K. Liu, D. Petr, and V. Frost, "Desing and analysis of a bandwidth management framework for ATM-based Broadband ISDN," *IEEE Comm. Mag.*, vol. 35, May 1997.

[11] G. Ash and et al., "Special issues in dynamic routing in telecommunications networks," *IEEE Comm. Mag.*, vol. 33, July 1995.

[12] B. Vandalore and et. al., "QoS and multipoint support for multimedia applications over the ATM ABR service," *IEEE Comm. Mag.*, vol. 37, pp. 53–57, Jan. 1999.

[13] The ATM Forum, "The ATM Forum traffic management specification version 4.0," Apr. 1996.

[14] J. Jaffe, "Bottleneck flow control," *IEEE Trans. Comm.*, vol. 29, pp. 954–962, July 1981.

[15] S. Fahmy and et al., "Feedback consolidation algorithms for ABR point-to-multipoint connections in ATM networks," in *INFOCOM'98*, pp. 1104–1113, 1998.

[16] W. Ren, K.-Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM," in *IEEE ATM'96 Workshop*, Aug. 1996.

[17] K.-Y. Siu and H.-Y. Tzeng, "Congestion control for multicast service in ATM networks," in *IEEE GLOBECOM'95*, pp. 310–314, 1995.

[18] J. Choe and N. B. Shroff, "Supremum distribution of Gaussian processes and queueing analysis including long-range dependence and self-similarity (a short version of this paper is published in INFOCOM'99)," *submitted to Stochastic Model*, 1999.

[19] W. Willinger, V. Paxson, and M. S. Taqqu, "Self-similarity and heavy tails: Structural modeling of network traffic," in *A Practical Guide to Heavy Tails* (R. J. Adler, R. E. Feldman, and M. S. Taqqu, eds.), pp. 27–53, Birkhauser, 1998.

[20] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *ACM SigComm*, Sept. 1994.

[21] W. Willinger, M. S. Taqqu, and A. Erramilli, "A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks," in *Stochastic Networks: Theory and Applications* (F. Kelley, S. Zachary, and I. Ziedins, eds.), 1996.

[22] P. Brockwell and R. Davis, *Time Series: Theory and Method.* Springer-Verlag, 1991.

[23] P. W. Glynn and W. Whitt, "Logarithmic asymptotics for steady-state tail probabilities in a single-server queue," *Applied Probability*, pp. 131–156, 1994.

[24] P. R. Jelenkovic, "Long-tailed loss rates in a single server queue," in *INFOCOM'98*, pp. 1462–1469, 1998.

[25] N. Likhanov and B. Tsybakov, "Analysis of an ATM buffer with self-similar ("fractal") input traffic," in *INFOCOM'95*, pp. 985–993, 1995.

[26] M. Taqqu, W. Willinger, and R. Sherman, "Proof of fundamental result in self-similar traffic modeling," *Computer Communication Review*, pp. 5–23, 1997.

[27] S. Ross, *Stochastic processes.* New York: John Wiley & Son, 1983.

[28] S. Abraham and A. Kumar, "A stochastic approximation approach for max-min fair adaptive rate control of ABR sessions with MCRs," in *INFOCOM'98*, pp. 1358–1365, 1998.

[29] Y. Hou, H.-Y. Tzeng, and S. Panwar, "A generalized max-min rate allocation policy and its distributed implementation using the ABR flow control mechanism," in *INFOCOM'98*, pp. 1366–1375, 1998.

[30] T. Lee and G. Veciana, "A decentralized framework to acheive max-min fair bandwidth allocation for ATM networks," in *Globecom'98*, 1998.

[31] W. Tasi and Y. Kim, "Re-examing maxmin protocols: A funcamental study on convergence, complexity, variations, and performance," in *INFOCOM'99*, pp. 811–818, 1999.

[32] D. Towsley, J. Kurose, and S. Pingali, "A comparison of sender-initiated and receiver-initiated reliable multicast protocols," *IEEE Journal on Sel. Areas in Comm.*, vol. 15, no. 3, pp. 398–406, 1997.

[33] J. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," in *INFOCOM'96*, pp. 1414–1424, Mar. 1996.

[34] S. Singhal, H. Holbrook, and D. Cheriton, "Log-based receiver-reliable multicast for interactive simulation," in *ACM SIGCOMM'95*, pp. 328–341, Oct. 1995.

[35] S. Floyd and et al., "A reliable multicast framework for light-weight sessions and application-level framing," in *ACM SIGCOMM'95*, pp. 342–365, Oct. 1995.

[36] Y. Zhao, S. Li, and S. Sigarto, "A linear dynamic model for design of stable explicit-rate ABR control schemes," in *INFOCOM'97*, Apr. 1997.

[37] C. Rohrs and R. Berry, "A linear control approach to explicit rate feedback in ATM networks," in *INFOCOM'97*, Apr. 1997.

[38] C. Fulton, S.-Q. Li, and C. Lim, "UT: ABR feedback control with tracking," in *INFOCOM'97*, 1997.

[39] N. Golmie, Y. Chang, and D. Su, "NIST ER switch mechanism (an example)," *ATM Forum/95-0695*, June 1995.

[40] S. Kalyanaraman and et. al., "The ERICA switch algorithm for ABR traffic management in ATM networks," Nov. 1997.

[41] M. Ritter, "Network buffer requirement of the rate-based control mechanism for ABR services," in *IEEE INFOCOM'96*, pp. 1090–1097, 1996.

[42] S. Abraham and A. Kumar, "Max-min fair rate control of ABR connections with nonzero MCRs," in *Globecom'97*, pp. 498–502, 1997.

[43] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *ICC'95*, pp. 1954–1963, 1995.

[44] W.-K. Liao, A. H. Esfahanian, L. M. Ni, and R. LePage, "Bounded-cell-loss-ratio flow control," *Technical Report*, 1999.

[45] C.-S. Chang, "On deteriministic traffic regulation and service guarantees: A systematic approach by filtering.," *IEEE Trans. Information Theory.*, pp. 1097–1110, 1998.

[46] E. Hannan, "The convergence of some recursions," *Annals of Statistics*, pp. 1258–1270, 1976.

[47] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[48] W.-K. Liao, A. H. Esfahanian, and L. M. Ni, "Reliable ABR multicast," *Technical Report*, 1999.

[49] W.-K. Liao, A. H. Esfahanian, and L. M. Ni, "Flow control for dispersity multicast," *Technical Report*, 1999.

[50] N. Maxemchuk, "Dispersity routing," in *ICC'75*, pp. 41–10–41–13, June 1975.

[51] N. Maxemchuk, "Dispersity routing on ATM networks," in *INFOCOM'93*, pp. 347–357, 1993.

[52] E. Gustafsson and G. Karlsson, "A Literature Survey on Traffic Dispersion," *IEEE Network*, pp. 28–36, March/April 1997.

[53] S.Fahmy and et al., "A switch algorithm for ABR multipoint-to-point connections," *ATM Forum/97-1085*, Dec. 1997.