Optimization-Based Flow Control in Multipoint-to-point Communication

by

Xinhua Weng

A thesis submitted to the
Department of Computing and Information Science
in conformity with the requirements for
the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
December 2001

To my wife

Yuhong

and my sons

Eric & Andrew

Abstract

Multipoint-to-point communication allows a group of sources to transfer data to one destination. A major requirement of flow control for such connections is to ensure a fair allocation of resources while maintaining a high level of resource utilization. This work treats multipoint-to-point flow control as a multiple-objective optimization problem and presents a theoretical centralized model as well as a distributed algorithm to compute rate allocations based on this global optimization. Three control objectives have been identified as critical to the flow control of multipoint-to-point connections: *overall network throughput*, *fairness amongst sources*, and *fairness amongst groups*.

The theoretical model is a linearly constrained quadratic programming model with an objective of minimizing weighted sums of individual objective functions. The weighting factors become tuning factors with which decision makers can set their decision preferences. It was shown that the three objectives may indeed be conflicting with each other and, by varying the values of tuning factors, an optimum rate allocation can be achieved to realize many flavors of objective mix.

The distributed algorithm attempts to implement the centralized model in a distributed environment. A resource pricing with an aggregate utility maximization

scheme was used to allocate bandwidth to maximize overall throughput. The algorithm is integrated with an explicit rate indication algorithm that optimizes resource allocation based on the source fairness criteria. It was shown that this algorithm attains similar result to the theoretical model and is also tunable.

This thesis shows that multiple-objective optimization-based rate allocation is feasible, flexible, and powerful, especially in situations where we are able to trade some objectives for others.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Hossam S. Hassanein, for his constant help and constructive support throughout this research. This work would not have been achieved without his full support.

Big thanks to my family for their valuable affection and encouragement: Yuhong, Andrew, Eric, and my parents.

Special thanks go to Dr. Yuting Jia and Guang-Chong Zhu who provided valuable help in the formulation of models and the design of distributed algorithms, and other friends for being supportive.

Finally, I would like to acknowledge the financial support provided to me by Queen's University and the Department of Computing and Information Science.

List of Acronyms

δ	Satisfaction Level
δ_s	Desired Satisfaction Level
δ_{fair}	Fair Satisfaction Level
δ_i	Satisfaction Level at Iteration i
$\widetilde{\delta}$	Preferred Satisfaction Level
ACR	Allowed Cell Rate
ATM	Asynchronous Transfer Mode
ABR	Available Bit Rate
DC	Destination Capacity
EPRCA	Enhanced Proportional Rate Control Algorithm
ERICA	Explicit Rate Indication for Congestion Avoidance
ICR	Initial Cell Rate
ISA	Integrated Service Architecture
LP	Linear Programming
Mbps	Megabits per second
MCR	Minimum Cell Rate
NLP	Non-Linear Programming

PCR	Peak Cell Rate
PD	Propagation Delay
QoS	Quality of Service
QP	Quadratic Programming
RDF	Rate Decrease Factor
RIF	Rate Increase Factor
RM	Resource Management
RSVP	Resource Reservation Protocol
TCP	Transport Control Protocol
VC	Virtual Channel

Contents

List of Figures

Chapter 1

Introduction

Multipoint communications is the exchange of information among multiple sources (senders) and multiple destinations (receivers). Depending on the number of sources and destinations involved, multipoint communication can be classified to several categories. *Unicast* and *multicast* are commonly used to classify traffic originating from a single source. Unicast is understood to be traffic from a single source to a single destination while multicast traffic is understood to be from a single source to multiple destinations. For connectionless traffic, these two terms should be enough to describe all traffic (*broadcast* can be considered as a special case of multicast).

For connection-oriented traffic, there is a need to classify traffic among multiple sources. Hence classification is not based on a single source but on a logical connection topology. There can be four cases (assuming connections to be unidirectional – from source to destination)

- *Point-to-point* connections: Exactly one source communicating over a connection with exactly one destination, e.g., a traditional TCP connection.
- *Point-to-Multipoint* connections: One source communicating with multiple destinations, e.g., a cable TV distribution system. The rate by which each destination receives data may be different.
- *Multipoint-to-Point* connections: Multiple sources communicating with a single destination, e.g., an alarm system with multiple sensors. The rate at which sources send data may be different.
- *Mulitpoint-to-Multipoint* connections: Multiple sources communicating with multiple destinations, e.g., a conference call.

Our research focuses on the case of multipoint-to-point communication. Particularly, we are interested in the *flow control* problem of such connections.

1.1 Multipoint-to-point Connections

Multipoint-to-point communication deals with transferring data from many sources to a single destination across the network. Instead of setting up multiple unicast connections from each source to the destination (in which case unicast connections are independent of each other) only one multipoint-to-point connection is set up to manage all parties. This grouping of logically related communication entities into one manageable unit generates many benefits over other connection methods, thus making

multipoint-to-point communication a promising mode for a variety of applications, such as group communications, adaptive multimedia [1], etc.

Multipoint-to-point communication is usually connection-oriented. Before any data transmission takes place, a *multipoint-to-point connection* must be established for a group of sources that wish to send data to a same destination. Such a multipoint-to-point connection is also called a *multipoint-to-point group* or *multipoint-to-point session*. In a multipoint-to-point connection, data flows from the nodes that wish to send data (the *senders*) to the node that wishes to receive this data (the *receiver*). In this thesis, the senders and the receiver are also referred to as the *sources* and the *destination*, respectively. Due to the nature of multipoint-to-point communications, each multipoint-to-point connection contains only one destination node, which is unchanged throughout the entire multipoint-to-point session. The source membership, however, may change dynamically in a multipoint-to-point session. This means new source may join an existing connection and existing sources may leave the connection. Each source also has an explicit rate *request* that may change dynamically .

In addition to the sources and the destination, each multipoint-to-point connection also contains a number of links and intermediate nodes (also called *routers*). Each link has a finite capacity that may change dynamically. The intermediate nodes forward the data packets from senders to their next router, which forward to its next router. The forwarding continues until the packets reaches the destination node. A router R_B is said to be *downstream* from another router R_A if R_B is on the path from R_A to the destination. Similarly, R_A is said to be *upstream* from R_B. The routers are also called *merge points* because incoming traffic is merged at these points to a single traffic flow which is forwarded to its downstream router.

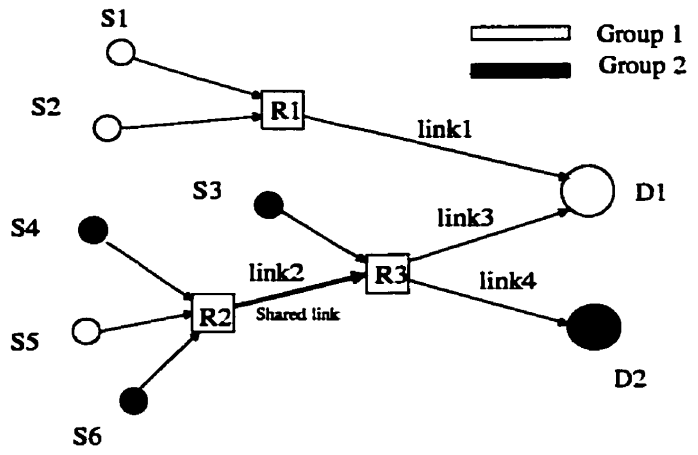

Figure 1.1: Multipoint-to-point connection: an example

Figure 1.1 shows a simple network consisting of two multipoint-to-point groups. Group 1 consists of sources S_1, S_2, S_5 and destination D_1, and Group 2 consists of sources S_3, S_4, S_6 and destination D_2. R_2 is upstream from R_3. D_1, D_2, and R_3 are downstream from R_2. *Link*2 is shared by both groups.

1.2 Thesis Objectives

The main focus of our research is flow control of multipoint-to-point connections. Our primary goal is to allocate bandwidth resources based on *global optimization* of the factors considered critical to the congestion control of this type of communication. In classic point-to-point connections, the major factors considered in flow control are *throughput* and *fairness*. The primary function of flow control can be described as to

ensure good throughput performance while maintaining a fair allocation of network resources to the sources.

Multipoint-to-point communication adds difficulty to this problem because it distinguishes fairness as *source fairness* and *group fairness*. Source fairness and group fairness are measured at the source level and at the group level, respectively. Source fairness requires sources be treated fairly within the group they belong to. Group fairness requires that all multipoint-to-point groups be treated fairly.

Network throughput by itself has not been used to allocate bandwidth in existing flow control schemes. The reason is obvious: allocations that maximize network throughput are often unfair. On the other hand, allocations that satisfy fairness criteria may have poor performance in delivering high network throughput. For example, in some network configurations, a typical "fair" allocation may achieve only about half of the theoretical maximum throughput. We argue that flow control schemes should have a mechanism to allow throughput to play a bigger role in the rate allocation. We also argue that rate allocation should be based on global optimization of multiple objectives. The objectives we have identified to be critical to the rate allocation of multipoint-to-point connections are:

1. Network throughput: We would like to maximize the amount of data transferred from all sources to all destinations in the network. This criterion is obvious because the network service providers normally charge users based on the amount of data transferred.

2. Source fairness (intra-group fairness): For any group, we would like to optimize the source rates so that fairness between individual sources is achieved.

3. Group fairness (inter-group fairness): We would like to optimize the flow assignment so that fairness among different multipoint-to-point groups is achieved.

Apparently, this results in a multiple-objective optimization problem. It is unlikely that an optimum solution exits to optimize all objectives (because the objectives may be conflicting). Instead, decision makers' preferences (bias) is introduced to trade-off objectives to achieve a "balanced" optimality.

1.3 Thesis Outline

The rest of this thesis is organized as follow.

Chapter 2 presents background information and literature review of the related researches. The survey focuses on fairness definitions, rate allocation algorithms, and optimization-based schemes.

Chapter 3 starts by defining fairness as a metric of user satisfaction level. Next, a theoretical multiple-objective optimization model is formulated to solve the flow allocation problems. The model behavior under different parameter settings is analyzed through example configurations.

Chapter 4 presents distributed algorithms to deploy the global optimization model in real networks. First, we present a framework based on which we design our distributed scheme to compute throughput-optimized allocations. Next we present an approximate approach with which we design the distributed source fairness allocation. We then integrate both approaches to achieve the combined objective. Experimental

results are presented for each of the algorithms. Results are compared to those from the theoretical model in Chapter 3 and the results of the max-min based scheme.

Chapter 5 concludes this thesis work and discusses future directions.

Chapter 2

Background and Literature Review

This thesis addresses the problem of flow control in multipoint-to-point network configurations. In multipoint communication, sources do not require a fixed rate of service and can adjust their transmission rates based on the congestion level of the network. Such traffic type is termed *elastic*. Examples of elastic traffic sources include internet traffic sources using TCP, sources using ABR service in ATM networks, and that from Controlled-load Service on Internet [2]. In this chapter, we provide some background information and survey related work in this field. We focus on the fairness definitions and rate control algorithms using optimization approaches. We have observed that most research on multipoint communications was focused on point-to-multipoint communications (multicast), as this is the most visible operation in practice. Relatively fewer researchers have focused on multipoint-to-point connections.

2.1 Fairness of Bandwidth Allocations

Providing fairness is an important requirement to any congestion control algorithm, especially in WAN environments due to high propagation delays and the heterogeneity of sources. This justifies the need to measure fairness upon which the congestion control is based.

2.1.1 Max-min Based Fairness

Fairness can be defined in as many ways as people perceive the word "fair". The most commonly used definition is the *max-min* fairness [3]. Informally, a feasible rate assignment is *max-min* fair if it is not possible to maintain feasibility and increase the rate of a source without decreasing that of any other source which has equal or lower rate. The max-min fair allocation can be computed by the "filling procedure" [4]. Another equivalent algorithm [5] is described below. Given a configuration with n contending sources, suppose the i^{th} source is allocated a bandwidth x_i. The allocation vector $\{x_1, x_2, ..., x_n\}$ is feasible if all link load levels are less than or equal to 100%. The total number of feasible vectors is infinite. Given an allocation vector, the source that is getting the least allocation is, in some sense, the "unhappiest source". We need to find the feasible vectors that give the maximum allocation to this unhappiest source. Now we remove this "unhappiest source" and reduce the problem to that of the remaining $n - 1$ sources operating on a network with reduced link capacities. Again, we find the unhappiest source among these $n - 1$ sources, give that source the maximum allocation and reduce the problem by one source. We keep repeating this process until all sources have been allocated the maximum that they can get.

When max-min fair allocation is achieved the following conditions hold: 1) each connection must pass through at least one saturated link (bottleneck), and 2) on each bottlenecked link, the available bandwidth should be shared evenly by the connections bottlenecked on that link.

Max-min fairness does not have a notion of minimum rate. So it had to be extended for ABR (Available Bit Rate) congestion control in ATM (see Appendix A). The ATM Forum [6] defines a number of alternative fairness definitions. MCR (Minimum Cell Rate) and PCR (Peak Cell Rate) are used to describe traffic for ABR (Available Bit Rate) users. For each link in an ATM network, all connections going through that link are either bottlenecked on that link or bottlenecked on other links (bottlenecked elsewhere).

First, the following parameters need to be defined for a given link:

A = Total available bandwidth for all ABR connections on the given link.

U = Sum of bandwidth of connections bottlenecked elsewhere (including those limited by PCR).

$B = A - U$, excess bandwidth, to be shared by connections bottlenecked on this link.

N = Total number of active connections.

N' = Number of active connections bottlenecked elsewhere.

$n = N - N'$, number of active connections bottlenecked on this link.

M = Sum of MCRs of active connections within n.

MCR_i = MCR of connection i.

x_i =fair allocation for connection i.

w_i =preassigned weight associated with connection i.

If all ABR connections have zero-MCR, the following criteria can be used for allocation:

- *max-min* allocation: equally allocate the bandwidth to all connections bottlenecked on a given link

$$x_i = B/n \tag{2.1}$$

  Assumes all connections are unweighted or equally weighted.

- *Weighted* allocation: if connections have different priorities in getting network resources, the allocation is made proportionally to their weights for all connections bottlenecked on the given link:

$$x_i = B \times (w_i / \sum w_i) \tag{2.2}$$

  where, $\sum w_i$ is the sum of the weights for all contending connections. Apparently, (2.1) is a special case of (2.2).

If connections have non-zero MCR, these MCRs must first be allocated (because MCR is guaranteed by ABR service) and then the leftover bandwidth can be further allocated to connections using one of the following criteria:

- *Proportional to MCR*: if MCR_i is used as the weight w_i, we can have the allocation proportional to the sum MCR of bottlenecked connections:

$$x_i = B \times (MCR_i/M) \tag{2.3}$$

- *MCR plus equal share*: The bandwidth allocation for a connection i is its MCR plus equal share of the bandwidth B with used MCR removed.

$$x_i = MCR_i + (B - M)/n \tag{2.4}$$

- *Maximum of MCR* or *MaxMin share*: the bandwidth allocation for a connection is its MCR or Max-Min share, which ever is larger. The Max-Min share is computed using (2.1).

$$x_i = max(MCR_i, MaxMin\ share) \tag{2.5}$$

- *General Weighted (GW) fairness* [7]: The bandwidth allocation for a connection i is its MCR plus weighted share of the bandwidth B with used MCR removed.

$$x_i = MCR_i + (B - M) \times (w_i / \sum w_i) \tag{2.6}$$

GW fairness is a general form of fairness that can represent (2.1), (2.2), and (2.4). (2.6) is different from (2.2) in that, with (2.6), only the excess bandwidth is allocated proportionally to weights and the allocation always ensures MCR.

2.1.2 Proportional Fairness

The appropriateness of the max-min allocation has been questioned by Kelly [8] who argues that, with max-min fairness, users may remain unaware of physical topology.

He has introduced the alternative notion of *proportional fairness*. With proportional fairness, however, users have an incentive to optimize physical routing, even though they may have no knowledge of physical topology.

Assume S is the set of all sources. A rate allocation $\mathbf{x} = (x_s, s \in S)$ is proportionally fair if it maximizes $\sum_{s \in S} \log x_s$ under the capacity constraints (no link is overloaded). This objective may be interpreted as maximizing the overall utility of rate allocations assuming each source has a logarithmic utility function. In finite networks, the vector of proportional fair rate allocation is unique. It may be characterized as follows. Rate allocation $\mathbf{x}$ is *proportionally fair* if it is feasible and if, for any other feasible allocation $\mathbf{x}^*$, the aggregate of proportional rate changes is zero or negative, i.e.

$$\sum_{s \in S} \frac{x_s^* - x_s}{x_s} \leq 0 \tag{2.7}$$

Proportional fairness is implemented with *resource pricing*. Each user chooses the price per unit time that is paid for that user's flow. At a scarce resource, capacity is shared amongst flows in proportion to prices paid. This resource pricing is termed proportionally fair pricing. Proportional fairness allocates bandwidth with a bias in favour of flows using a smaller number of links. [8, 9, 10] have shown that congestion control based on additive increase and multiplicative decrease tends to share bandwidth according to proportional fairness.

2.1.3 Multipoint-to-point Specific Fairness

For multipoint-to-point connections, the fairness issue becomes more complicated, as fairness needs to be measured at different levels. For examples, we may need to measure fairness at the source level within a group (connection) and/or among different groups (including point-to-point and multipoint-to-point connections).

Fahmy *et al* [11, 12, 13] proposed a set of fairness definitions for multipoint-to-point connection congestion control for ABR service in ATM networks. Fairness can be measured at source level, virtual circuit (VC) level, and the *flow* level. Here, VC (Virtual Circuit) is the multipoint-to-point connection or group. A flow of a switch is the traffic coming on an input port of the switch. Four different fairness definitions were proposed.

Source-based fairness divides bandwidth fairly among active sources as if they were sources in point-to-point connections, ignoring grouping memberships. The allocation vector $\{x_1, x_2, ..., x_n\}$ is determined based on applying the underlying fairness definition for all active sources

Group/Source-based fairness first gives fair bandwidth allocation among different groups, and then fairly allocates the bandwidth of each group among its sources.

Flow-based fairness gives fair allocation for each active flow, where a flow is a VC coming on an input link. The number of flows for an output port is formally defined as the sum of the number of active VCs sending to this output port, for each of the input ports of the switch.

Group/flow-based fairness first gives fair bandwidth allocation among the groups, and then fairly allocates the bandwidth of each group among its flows.

Derived from the different definitions (source, group, and flow), fairness may give very different allocations in some situations. The source-based fairness completely ignores the membership of different sources to connections, and divides the available bandwidth max-min fairly among the sources currently active. If the billing and pricing are based upon sources, it can be argued that this mechanism is good, since allocation is fair among sources. However, if pricing is based on connections (VCs), a VC with 50 concurrent senders should not be allocated 50 times the bandwidth of a point-to-point connection bottlenecked on the same link. VC/source-based fairness is obviously a better choice in this case. The flow-based method is not max-min fair if we view an N-to-one connection as N one-to-one connections, since the same flow can combine more than one source. We can, however, argue that it may be better to favor sources traversing a small number of merge points, since these are more likely to encounter less bottlenecks anyway. Thus, although flow-based fairness may be unfair to sources whose traffic is merged many times with other flows, this might be acceptable in many practical situations. The VC/flow-based fairness is max-min fair with respect to VCs, but within the same VC, it favors sources whose traffic goes through a small number of merge points. Therefore each type of fairness has its own merits and drawbacks and the choice of the type of fairness to adopt relies on the billing and pricing methods used.

Moh and Chen [14] extended and enhanced the *"essential fairness"* concept, which was first proposed to flow control of multicast and unicast TCP traffic in the Internet, to the multipoint-to-point ABR flow control. Let λ_m and λ_μ be the average throughput of a multicast session and of a unicast session, respectively, and let N be the total number of sources in the multicast session. Then a multicast session is essentially fair if λ_m can be bounded by

$$\lambda_\mu \leq \lambda_m \leq N \times \lambda_\mu \tag{2.8}$$

When $\lambda_m = N \times \lambda_\mu$, each multicast source is being treated as a single unicast source (equivalent to source-based fairness); when $\lambda_m = \lambda_\mu$, the entire multicast session is being treated as a single unicast session (equivalent to VC-based fairness). Further more, the essential fairness is represented by bounding the throughput of a unicast session λ_μ by λ_{mss} (the throughput of a single source of a multicast session).

$$\lambda_\mu = w \times \lambda_{mss} \tag{2.9}$$

$$\lambda_m = N \times \lambda_{mss} \tag{2.10}$$

where $1 \leq w \leq N$.

2.2 Flow Control Algorithms

Flow control algorithms implement the target fairness within the network. There exists two broad categories of end-to-end adaptive rate control algorithm: additive increase, multiplicative decrease congestion avoidance and explicit rate calculation. Both rely on the network to provide congestion status feedback.

2.2.1 Additive Increase, Multiplicative Decrease

In the absence of congestion, users with unlimited demands may increase their sending rate linearly. However this may cause congestion and in this case, users begin to decrease the rate multiplicatively. The rate of increase and decrease must be chosen to limit the amplitude of oscillations, which can lead to inefficiencies in link utilization and to ensure rapid convergence when the population of active flows changes.

It is generally recognized in the ATM community that congestion indication is less fair than explicit rate due to the so-called "beat down" effect, in which flows routed over a long path are more often required to reduce their rate than flows on short routes and are consequently unable to compete fairly. However, network economy suggests long paths should be charged more because they use more resources than short paths. The process of additive increase multiplicative decrease leads to proportional fair allocation according to [8, 9] since longer paths tend to be charged more for bandwidth as they may use more congested links.

ATM Binary Feedback Schemes [6] [1]

ATM switches perform two important functions: 1) Detect incipient congestion and 2) provide binary feedback to sources. The basic binary scheme assumes all VCs share a common FIFO queue and queue length is monitored by setting a threshold T. When the queue length exceeds T, congestion is declared and the cells passing through the queue have their EFCI bit set. When the queue length falls below the

[1] See Appendix A for details of ATM ABR traffic congestion control.

threshold (T), the cells are passed without setting their EFCI bits. The destination monitors these indications for a periodical interval and sends an RM cell back to the source. The sources use an additive increase and multiplicative decrease algorithm to adjust their rates. Some proposals use two thresholds, a high threshold T_{High} and a low threshold T_{Low}. When a queue size increases past cross T_{High}, congestion is detected. When the queue starts emptying, the congestion condition is not removed until the queue falls below T_{Low}. Binary feedback schemes where connections may share a common FIFO may sometimes suffer from unfairness problems depending on the network topology and the source and destination behaviors. Given the same level of congestion at all switch connections, packets traversing more hops have a higher probability of having their EFCI bits set than those traversing a smaller number of hops. Therefore, VCs with a long path do not have the same opportunity to increase their rates and consequently their throughputs are starved (this is known as the *beat down problem*).

Potential unfairness problems in binary feedback schemes where all the VCs share a common FIFO can be alleviated by some enhancements to the basic scheme. A separate FIFO queue can be provided for each VC or groups of VCs. This generally results in ensuring fairness among different VCs. Another enhancement is to provide selective feedback or intelligent marking. In this scheme a switch computes a *"fair share"* and if congested sets EFCI bits in cells belonging to only those VCs whose current rates are above the fair share. Alternative ways that a switch can use to compute the *"fair share"* are presented in the following sections on explicit rate feedback.

2.2.2 Explicit Rate Control

The single-bit binary feedback can only inform sources whether they should increase or decrease their rates. The scheme is too slow for rate-based control in high-speed networks [15]. Explicit rate indication [16], on the other hand, would not only be faster, but would offer a better way to control fairness.

The explicit rate scheme in ATM works as follows. Each source puts the rate at which it would like to transmit in the explicit rate (ER) field of the forward RM cell. The value of the ER field is initially set to PCR. Any switch along the path my reduce the ER value to the desired rate that it can support. If the destination is congested, it may also reduce the ER value before returning the RM cell to the source. When the source receives the backward RM cell, the source adjusts its transmission rate so as not to exceed the ER value.

The switch can use many methods to compute its desired rate (fairshare). One attractive way, which is based on the max-min fairness principle, assigns the available bandwidth equally among connections that are bottlenecked on specific links.

A popular rate control algorithm that combines both binary feedback and explicit rate feedback is Enhanced Proportional Rate Control Algorithm (EPRCA). EPRCA allows simple switches supporting only EFCI bit setting to inter-operate with more complex switches that can compute ERs. Switches that implement only EFCI mechanism would ignore the content of the RM cell and would set the EFCI bit to one if the link is congested. Switches that implement the ER scheme may reduce the ER value in the RM cell accordingly if the link is congested. The destinations turn around RM cells, setting the CI bit to one if the last received data cell has the EFCI

bit set to one. When the source receives the backward RM cell, the source set its transmission rate to the minimum value calculated by the binary feedback scheme and the ER value specified in the RM cell.

In EPRCA an ABR source should adhere to the following rules:

1. The source may transmit cells at any rate up to the *allowed cell rate* (ACR). The value of the ACR should be bounded between MCR and PCR.

2. At the call setup time, the source sets ACR to the *initial cell rate* (ICR). The first cell transmitted is an RM cell. When the source has been idle for some time, ACR should also be reduced to ICR.

3. The source should send one RM cell for every $N_{RM} - 1$ data cells or when T_{RM} time (typically set to 100msec) has elapsed.

4. If the backward RM cell does not return, the source should decrease its ACR by ACR*RDF, down to MCR. (RDF is known as the rate decrease factor and is typically set to 1/16).

5. When the source receives a backward RM cell with CI=1, the source should also decrease its ACR by ACR*RDF, down to MCR.

6. When the source receives a backward RM cell with CI=0, the source may increase the ACR by no more than RIF*PCR, up to the PCR. (RIF is known as the rate increase factor and is typically 1/16).

7. When the source receives any backward RM cell, the source should set the ACR to the minimum of the ER value from the RM cell and the ACR computed in 5 and 6.

An ABR destination should adhere to the following rules:

1. The destination should turn around all RM cells so that they can return to the source. The direction bit (DIR) in the RM cell should be set to one to indicate a backward RM cell.

2. If the last received data cell prior to a forward RM cell had an EFCI bit set to one, the destination should set the CI bit in the backward RM cell to one. The destination may also reduce the ER value to whatever it can support.

Finally, an ATM switch supporting ABR congestion control should adhere to the following rules:

1. The switch should implement either EFCI marking or ER marking. With EFCI marking the switch should set the EFCI bit of a data cell to one when the link is congested. With ER marking the switch may reduce the ER field of forward or backward EM cells.

2. The switch may set the CI bit of the backward EM cell to one to prevent the source from increasing its rate.

3. The switch may generate a backward RM cell to make the source respond faster. In such case, the switch should set CI=1 and BN=1 to indicate that the RM cell is not generated by the source.

2.2.3 Hop-by-hop Rate Assignment

Unlike most rate-based schemes which assign bandwidth to sources on an end-to-end basis, Hassanein *et al* [17, 18] presented an assignment procedure that is carried out on a hop-by-hop basis. The simulation results show that a near zero packet loss probability at the destination is achieved while attaining resource utilization and fairness. With the hop-by-hop scheme, the rates are computed and migrated hierarchically as follows. The destination node D computes quotas for it child set $C(D)$ based on the current available bandwidth. Each node $x \in C(D)$ in turn computes a new set of quotas for its own child set $C(x)$. This process is recursively carried out until, eventually, the appropriate quotas reach source nodes.

In the heart of this scheme lies the *quota assignment function* that assigns bandwidth to the children sources as whole. This function has the following characteristics:

1. The quota $Q_{x,i}$ assigned to a source flow i at node x should be at least equal to the aggregated MCR on the incoming flow $\overline{MCR_{x,i}}$. Similarly, no incoming flow should be assigned a quota that exceeds its aggregated PCR, $\overline{PCR_{x,i}}$, or the destination service rate, BW_D.

2. The quota assignment adopts a fairness criteria that achieves the best match between the assigned quotas and the traffic requirements at the individual source. The traffic requirement is represented by the number of cells counted on each input link.

3. The function is adaptive to reflect the most recent traffic measurements on the

links yet incorporating some historical requirements. This is done by exponentially averaging out the traffic data $avg_{x,i}$ over a time window T.

4. The quota assignment function preserves the location-fairness criteria. This can be achieved by setting the monitoring window T to a value larger than twice the maximum propagation delay between the destination and any source ($T > 2 \times PD_{max}$).

The quota assigned by a parent node x to a child link i, $Q_{x,i}$, is computed as follows:

$$Q_{x,i} = min(\overline{PCR_{x,i}}, BW_D, \overline{MCR_{x,i}} + (avg_{i,x}/Avg_x)(Q_x - \overline{MCR_X}))$$

where $Avg_x = \sum_{i \in E(x)} avg_{x,i}$ is the total average traffic requirements at node x, $E(x)$ is the set of links connecting node x to its children node $C(x)$. Q_x is the total quota assigned to node x in cells per second.

The first two terms in $Q_{x,i}$ ensure the assigned quota does not exceed the total PCR and destination service rate BW_D. The last term ensures the MCR must be satisfied and then the left over bandwidth $Q_x - \overline{MCR_X}$ is allocated in proportion to the requirements on each incoming link.

2.3 Optimization-based Flow Control

Flow control facilitates the sharing of network resources amongst competing sources. It often consists of two algorithms: a *link algorithm* executed at routers or switches, and a *source algorithm* executed at edge devices such as host computers. The link

algorithm detects congestion and sends feedback information to sources, and in response, the source algorithm adjusts the rate at which it sends data into the network. The ideal design is to have link and source algorithms work seamlessly to achieve global resource utilization, fairness and stability. This motivated recent approach to flow control schemes based on optimization techniques [8, 19, 20, 21, 22, 23, 24, 25], where the goal is to choose source rates to maximize a global measure of network performance. Different proposals in the literature differ in their choice of objective function, or solution approach, which in turn leads to different link and source algorithm and their implementation.

2.3.1 Max-min Based Algorithms

Max-min fairness is the most accepted fairness definition. It has the advantage of being simple to define at a router. There are many algorithms that can achieve max-min fair allocation in both ATM and TCP network, such as Explicit Rate Indication for Congestion Avoidance (ERICA) [26, 27].

2.3.2 Resource Pricing Algorithms

Resource pricing algorithms often use a *utility function* to measure the amount of "welfare" that a source receives when transmitting data at a certain rate. The utility of a user (source) is a function relating the bandwidth given to the user with a "value" associated to the bandwidth. The utility could be the perceived quality of video or the amount paid by the user for the bandwidth. Flow control should maximize an objective function representing the overall utility of all sources.

According to Kelly [8, 19], the system's objective is to maximize the overall utilities. Consider a network with a set of J of *resources*, and let C_j be the finite capacity of resource $j \in J$. Let a *route* r be a non-empty subset of J, and let R be the set of possible routes. $A = (A_{jr}, i \in J, r \in R)$ is a 0-1 matrix. $A_{jr} = 1$ if $j \in r$. $A_{jr} = 0$ otherwise. Associate a route r with a user, and suppose that if a rate x_r is allocated to user r then this has utility $U_r(x_r)$ to the user (certain conditions are required for the utility function). Let $C = (C_j, j \in J)$. The optimization problem is represented as:

SYSTEM(U,A,C):

$$max \sum_{r \in R} U_r(x_r) \tag{2.11}$$

subject to

$$Ax \leq C \tag{2.12}$$

over

$$x \geq 0 \tag{2.13}$$

Although this problem is mathematically solvable, it involves utilities U that are unlikely to be known by the network. So this problem is decomposed into smaller problems. For each user, suppose that user r may choose an amount to pay per time unit, w_r, and receives in return a flow x_r proportional to w_r, say $x_r = w_r/\lambda_r$, where λ_r could be regarded as a charge per unit flow for user r. The the utility maximization problem for user r is:

$USER_r(U_r;\lambda_r)$:

$$max\ U_r(\frac{w_r}{\lambda_r}) - w_r \tag{2.14}$$

over

$$w_r \geq 0$$

Suppose the network knows the vector $w = (w_r, r \in R)$, and attempts to maximize the function $\sum_r w_r log x_r$. The network optimization problem is then:

$NETWORK(A, C, w)$:

$$max \sum_{r \in R} w_r log x_r$$

subject to

$$Ax \leq C$$

over

$$x \geq 0$$

Under this decomposition, the utility function is no longer required by the network, and only appears in the optimization problem faced by user r. The $NETWORK$ problem can be decentralized by estimating the rate.

Richard Gibbens [20] showed that by appropriately marking packets at overloaded resources and by charging a fixed small amount for each mark received, end-nodes are

provided with the necessary information and the correct incentive to use the network efficiently. An optimum system is achieved when users' choices of charges and the network's choice of allocated rates are in equilibrium.

S. Low [21, 22] presented a framework that also maximizes the aggregate source utility function over their transmission rates. In Kelly's preferred model, a user chooses the charge per unit time that the user is willing to pay; thereafter the user's rate is determined by the network according to a proportional fairness criterion applied to the rate per unit charge. However, in Low's scheme, users decide their rates and pay whatever the network charges. Both schemes have a property that they treats practical flow control schemes as an implementation of a certain optimization algorithm in a distributed computing environment.

Note that, in resource pricing schemes, different sources could have different utility functions. Thus they provide a framework to support heterogeneous users in the system.

2.3.3 Delay-based Optimization

For elastic traffic, delay is the document transfer time. This category of congestion control is based on the minimization of potential delays experienced by all ongoing transfers. A flow's potential delay (transfer time) can be considered equal to the reciprocal of the rate allocation, $1/x_s$.

In [25], Massoulié *et al* proposed a potential delay minimization criteria as an alternative to max-min fairness and proportional fairness. They also show that fixed

window size control can achieve such objectives

2.4 Summary

Fair bandwidth allocation in the context of multipoint-to-point connections is a difficult task. A number of proposals have been presented to solve this problem. A typical proposal usually contains a fairness definition and an algorithm to achieve such "fair" allocation. The fairness definition specifies the way to measure and compare different allocations (e.g., the *max-min* fairness). The algorithms strive to find the "optimum" per source throughput allocation under their respective fairness definitions.

The widely accepted max-min fairness has some basic assumptions. As mentioned in the previous sections, max-min fairness assumes all sources have infinite maximum bandwidth requests and will consume any bandwidth assigned to them (these sources are known as greedy sources). While in reality, sources may only request a bandwidth that is lower than their max-min fair allocation. In other words, if a source is "max-min" fairly assigned a bandwidth which is greater than its request, the unused bandwidth is likely to be wasted. Another concern about max-min is its capability of delivering maximum throughput. Max-min fairness allocation claims to fully utilize link capacity in the sense that any connection goes through at least one saturated link. This is not equivalent to achieving the maximal network throughput (flow). In some network configurations, the throughput delivered by "max-min" fair allocation could be far less than the actual maximum network throughput.

As well, destination nodes are assumed to have infinite capacity in existing fairness

definitions and flow control schemes. This assumption becomes weak in multipoint-to-point connections because a single destination node may receive data from many sources simultaneously. It is necessary to assume that destination nodes have a limited capacity absorbing the data on its incoming links.

Chapter 3

Optimization Model

In this chapter, we first define the network model and assumptions (Section 3.1). Next, in Section 3.2, we present a new way of measuring fairness by defining source fairness and group fairness. Section 3.3 models the multi-objective decision problem as a quadratic programming problem. The model is guaranteed to find the globally optimal solution. Experimental results on sample network configurations are presented in Sections 3.4. In Section 3.5 we discuss the properties of our model and compare it to other methods.

3.1 Network Model

In this thesis, multipoint-to-point communication is assumed to have the following characteristics:

- Each source node may send data to the destination node at an arbitrary rate, which is usually independent of those of other sources. The requested rate for each source may vary from time to time, reflecting the dynamics of the sources. The request variations for different sources are also independent.

- Multipoint-to-point is connection-oriented. A connection from the sources of the communications group to the destination is explicitly required. The multipoint-to-point connection can be modelled by a *tree* with all sources (senders) at the leaf nodes and the destination at the root. Intermediate nodes that are neither destinations nor senders are denoted as *merge points* or *routers*.

- At each intermediate node or merge point, traffic on all incoming flows (from its child nodes) are merged and sent on its outgoing link to its next hop router or the destination. The size of the merged flow is the sum of all incoming flows.

- Links have limited resources available to be shared by multiple connections that use them. The available link resources may change dynamically. Whenever link capacity is shared, it is the flow control's responsibility to allocate available resources to all competing connections. The rules and heuristics for the allocation is an important part of the flow control scheme in the context of multipoint-to-point connections. In this research, we focus only on one type of resource – *bandwidth*.

- The closed-loop flow control mechanism is assumed. Each source indicates its requested transmission rate to the network on a regular basis. The rate assignment mechanism determines rates for each source and sends a feedback message to the sources which adjust their transmission rates to designated values.

Based on the assumptions above, we formally define the network model as follow. A network is defined by (L, G, S, D). $L = (1, ..., L)$ is a set of unidirectional *links* each having a capacity $c_l, l \in L$. The network is shared by a set $S = (1, ..., S)$ of *sources* grouped into a set $G = (1, ..., G)$ of multipoint-to-point *groups* (also called *sessions*). Each group $g \in G$ is associated with a unique *destination* D_g, a set of sources $S(g) \subseteq S$ that comprise the group, and a set of links $L(g) \subseteq L$ that the group uses. The set of links $L(g)$ forms a tree. We assume fixed path routing. So the tree associated with each group is fixed. Each source $s \in S$ is characterized by four parameters $(L(s), m_s, M_s, G(s))$. The path $L(s) \subseteq L$ is a set of links that source s uses to reach its destination, $m_s \geq 0$ and $M_s \leq \infty$ are the minimum and maximum transmission rates, respectively, requested by source s. $G(s) \subseteq G$ is the set of groups that source s belongs to. Throughout this thesis, we limit any source to participate in only one group, $|G(s)| = 1$. For link l, let $S(l) \subseteq S$ be the set of sources that go through this link regardless of their group membership, and let $G(l) \subseteq G$ be the set of groups that use this link. Note that $l \in L(s)$ if and only if $s \in S(l)$. The destinations of all groups form the set of destinations D. Each destination D_g has a destination capacity (receiving capacity) of DC_g.

3.2 Fairness Definition: Satisfaction Level

We adopt a closed-loop explicit rate flow control mechanism. At any given time, each source explicitly indicates its bandwidth request to the network and expects its request be fully met. The flow control mechanism collects such requests and link usage and/or network congestion status. It then makes a decision to allocate the available

bandwidth to all sources such that the fairness criteria are best met. This decision is made periodically. Once the rates are determined, sources adjust to the designated rates until the next decision and adjustment. The rate assignment guarantees the assigned bandwidth never exceeds the requested bandwidth.

Definition 1: *Satisfaction Level* δ is defined as the ratio of assigned bandwidth a to the requested bandwidth r,

$$\delta = a/r \tag{3.1}$$

Apparently, a source should never be assigned a rate greater than its maximum requested rate, i.e., $\delta \leq 1$.

We can extend this definition to sources and groups and thus have the following definitions.

Definition 2: *Source Satisfaction Level* for source $s \in S$ is defined as

$$\delta_s = \frac{a_s}{r_s}$$

where r_s is the bandwidth request for source s and a_s is the allocated bandwidth.

Definition 3: *Group Satisfaction Level* for group $g \in G$ is defined as the ratio of group aggregate allocated rate a_g to group aggregate request r_g

$$\delta_g = \frac{a_g}{r_g} = \frac{\sum_{s \in S(g)} a_s}{\sum_{s \in S(g)} r_s}.$$

Based on the definition of Source Satisfaction Level and Group Satisfaction Level, we define the following fairness measures:

Definition 4: *Group Fairness* (or *Inter-group fairness*) measures the fairness among different groups in the network. The group fairness is defined as

$$f_g = \sum_{\forall\, g_1,\, g_2 \in G} (\delta_{g_1} - \delta_{g_2})^2$$

Group Fairness is used to compare different solutions in terms of group fairness. The smaller the value of f_g the better the group fairness is. For instance, let A_1 and A_2 be two feasible solutions that have their group fairness calculated as $f_g(A_1)$ and $f_g(A_2)$, respectively. If $f_g(A_1) < f_g(A_2)$, we say solution A_1 is better than solution A_2, in terms of group fairness.

Definition 5: *Source Fairness* (or *Intra-group Fairness*) measures the fairness among sources within a single group. The Source Fairness for group $g \in G$ is defined as

$$f_s(g) = \sum_{\forall\, s_1,\, s_2 \in S(g)} (\delta_{s_1} - \delta_{s_2})^2$$

To measure fairness among all sources in the network (not just within a single group), we can use the term $\sum_{\forall\, s_1, s_2 \in S} (\delta_{s_1} - \delta_{s_2})^2$. However, this expression may be computationally complex. To measure fairness between any pair of sources would require $n(n-1)/2$ (n is the number of sources) terms in the function, which may be a large number. So we use a simpler form of calculation as defined below.

Definition 6: *Overall Source Fairness* is the aggregate of source fairness for all groups.

$$f_s = \sum_{g \in G} f_s(g)$$

Similar to group fairness f_g, overall source fairness f_s is used to compare different solution in terms of (overall) source fairness. A solution with a smaller f_s is considered better than solutions with higher f_s values, in terms of source fairness. Single group source fairness should not be used in isolation of other groups. In the rest of this thesis, source fairness will be represented by the overall source fairness (not the single group source fairness in Definition 5).

3.3 The Optimization Model

The basic requirement for traffic management is to maximize the resource utilization while maintaining fairness. This requires the optimization of more than one objective function hence turning the rate allocation into a multi-objective optimization problem. In many cases, it is unlikely that the different objectives would be optimized by the same solution. Hence, some trade-off between the criteria is needed to ensure a satisfactory solution. Next, we will follow a 3-step procedure to formulate the flow allocation optimization model. The three steps are: 1) identify decision variables; 2) specify constraints; and 3) construct objective functions.

Decision variables: In our rate allocation problem, the decision variables are the rates to be allocated to the sources. Let x_s denote the allocated rate for a single source $s \in S$, where S is the set of all sources. The vector $\mathbf{x} = (x_s,\ s \in S)$ is called an *allocation vector* or *solution.*

Constraints: The constraints of the decision variables are:

$$Subject\ to:\ m_s \leq x_s \leq M_s, \quad \forall s \in S \tag{3.2}$$

$$\sum_{s \in S(l)} x_s \leq c_l, \quad \forall l \in L \tag{3.3}$$

$$\sum_{s \in S(g)} x_s \leq DC_g, \quad \forall g \in G \tag{3.4}$$

Equation (3.2) limits source allocated rate to be between its minimum (m_s) and maximum (M_s) rate requests. Because $m_s >= 0$ there is no need for a non-negative constraint for each decision variable. Equation (3.3) ensures that the aggregate rate at any link l does not exceed the link capacity c_l. (A link may be shared by multiple groups.) Equation (3.4) requires the aggregate traffic for any group not exceed the maximum absorbing capacity at their destination node. Equations (3.2)-(3.4) are all linear thus making our rate allocation a linearly constrained optimization problem.

If an allocation vector $\mathbf{x}$ satisfies constraints (3.2)-(3.4), $\mathbf{x}$ is a *feasible allocation* or a *feasible solution.* Otherwise, $\mathbf{x}$ is *infeasible.* All feasible allocations constitute a feasible solution space. The optimum solution is a feasible solution that optimizes the objective function.

Objective functions: The goal of our flow control mechanism is to optimize the rate allocations towards three objectives:

1. *Network throughput (network flow)*

$$\max f_{flow} = \frac{\sum_{s \in S}(x_s)}{\sum_{s \in S}(r_s)} \tag{3.5}$$

This obvious goal is to maximize the network throughput (network flow) to achieve the highest overall bandwidth efficiency. Note that this goal is not the same as maximizing the overall link utilization. There are many cases where some links are fully utilized but the network is delivering a total flow far below the maximal value. Using throughput by itself as an optimization objective has been criticized for the inherent unfairness of the optimal allocation (e.g., some sources getting zero bandwidth). This is why we are using throughput along with other objectives as presented below.

2. *Group fairness (inter-group fairness)*

$$\min f_g = \sum_{\forall g_1, g_2 \in G} (\delta_{g_1} - \delta_{g_2})^2 \tag{3.6}$$

is a measure of fairness among different groups. This measure is based on group-level satisfaction. It views a group as a whole entity and considers only the aggregate request and allotted bandwidth for that group. The optimization goal is to minimize the group fairness. The "ideal" optimal value would be zero, which can be achieved when all groups have identical satisfaction levels.

3. *Overall source fairness (intra-group fairness)*

$$\min f_s = \sum_{g \in G} \sum_{s_1, s_2 \in S(g)} (\delta_{s_1} - \delta_{s_2})^2 \tag{3.7}$$

is to measure the fairness among individual sources in each group. Note that only source pairs in same group participate in the measurement. The optimization direction is minimization. The "ideal" optimal value would be zero, which can be achieved when sources of any one group have identical satisfaction levels.

Intuitively, these objectives may be conflicting with each other in many cases. For example, network throughput can be often increased at the cost of fairness. Group fairness and source fairness may also be conflicting. It is unlikely to find an "optimum" solution that optimizes all three objectives. Instead, a "satisfactory" solution is sought. The technique we use to solve this problem is to combine the multiple objectives into one scalar objective by a method called Minimizing Weighted Sums of Functions, which is to minimize a positively weighted sum of all three objectives.

$$\min \quad Z = -\alpha_1 \cdot f_{flow} + \alpha_2 \cdot f_s + \alpha_3 \cdot f_g \tag{3.8}$$

where, $\alpha_1, \alpha_2, \alpha_3$ are non-negative weights for the three objectives. The negative sign in front of α_1 has the effect of changing the optimization direction for throughput from maximization to minimization. The weights represent the *relative* importance between different objectives. It is the relative values of α's that differentiate the weights of different objectives in Z. For example, $\alpha_1 = \alpha_2$ does not mean f_{flow} and f_g are equally important. But increasing the ratio of $\frac{\alpha_1}{\alpha_2}$ means the objective f_{flow} is having more impact on the final optimal solution than objective f_g. This is also true for f_{flow} vs. f_s and f_g vs. f_s. Therefore, α_1, α_2, and α_3 are also called *tuning factors*, as decision makers may use them to fine-tune the model to reflect their decision preferences.

We will refer equations (3.2)-(3.4) and objective function (3.8) as **Model 1**. This is a linearly constrained quadratic programming, a solvable non-linear programming problem.

Objective function (3.8) requires $\alpha_1 > 0$. When $\alpha_1 = 0$, (3.8) can give an all-0 allocation [1] because an all-0 allocation vector always generates zero group fairness and zero source fairness. To solve this problem, we use the Goal Programming technique, in which throughput is expressed as a goal. For example, instead of maximizing network throughput, we can setup a goal for throughput, say we would like network throughput to be no less than 240. If this goal is met, we look no further in making it better. Thus the solution space is given to optimize other objectives. If the goal can not be met, the problem has no solution. To formulate the model with this strategy, we need to convert the network flow objective to a constraint. The modified objective function and added constraint are given in Equations (3.9) and (3.10), respectively. We will refer the modified model as **Model 1a**.

$$\min \quad Z' = \alpha_2 \cdot f_s + \alpha_3 \cdot f_g \tag{3.9}$$

$$\sum_{\forall s \in S} (x_s) \geq c \tag{3.10}$$

where c is the desired minimum network throughput, $c \leq$ maximal network throughput.

In the experiments in the following section, we will use the model with objective Z (3.8) and constraints (3.2)-(3.4) when $\alpha_1 > 0$. When $\alpha_1 = 0$, we use the model with objective Z' (3.9) and constraints (3.2)-(3.4), and (3.10).

[1] depending on the algorithm.

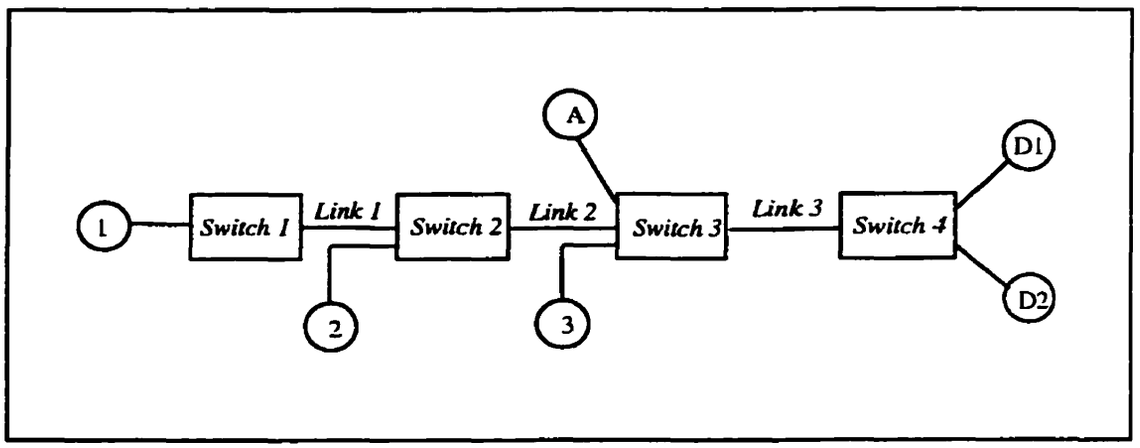

Figure 3.1: Sample network: downstream bottleneck

3.4 Numerical Results

The optimization model presented is guaranteed to find an optimal solution that minimizes the objective function that consolidates our three objectives. Because the problem contains three objectives and three tuning factors, we are going to illustrate their relationship with different network configurations.

3.4.1 Downstream Bottleneck

Figure 3.1 illustrates a configuration with two groups. One of the groups is a multipoint-to-point connection with three sources (S_1, S_2 and S_3) and one destination (D_2). The other group contains only one source (S_A) sending to destination D_1. The requested rates from all sources are $r_A = 90$, $r_1 = 30$, $r_2 = 90$, and $r_3 = 90$, in Mbps. All links are 150Mbps. Destinations have infinite capacity. Apparently, the bottleneck is $Link_3$ between $Switch_3$ and $Switch_4$.

Assume $\alpha_1 = \alpha_2 = \alpha_3 = 1$. The optimal rate allocation is calculated by **Model 1** as $(a_A,\ a_1,\ a_2,\ a_3) = (45,\ 15,\ 45,\ 45)$. The satisfaction levels for all four sources and two groups are all 0.5. In other words, all sources and groups are allocated 50% of what they requested. The solution has accomplished the ideal optimum on both fairness definitions, i.e., $f_g = 0,\ f_s = 0$.

This is indeed a simple configuration since all traffic are bottlenecked on *Link*3, which is the last link connecting to the destinations. No matter what values we choose for source requests and for tuning factors, we are always able to satisfy all sources to the same level.

If we compare this result with the max-min fair allocation (37.5, 37.5, 37.5, 37.5), we will see source S_1 is treated very differently. In max-min allocation, S_1 gets 37.5, while it only needs 30. This then demonstrates the major difference between max-min fairness and the satisfaction level fairness we define here. The differences are 1) satisfaction level fairness assigns rate based on request, and 2) satisfaction level fairness is based on normalized rate (δ), not the actual bandwidth. Source S_1's request is 1/3 of that of other sources, but they all receive 50% of what they requested.

If we allow $\alpha_1 > 0$ and $\alpha_2 = \alpha_3 = 0$, **Model 1** then solves the classic network max-flow problem. This is true for any network configurations. The specific value of α_1 does no matter in this case.

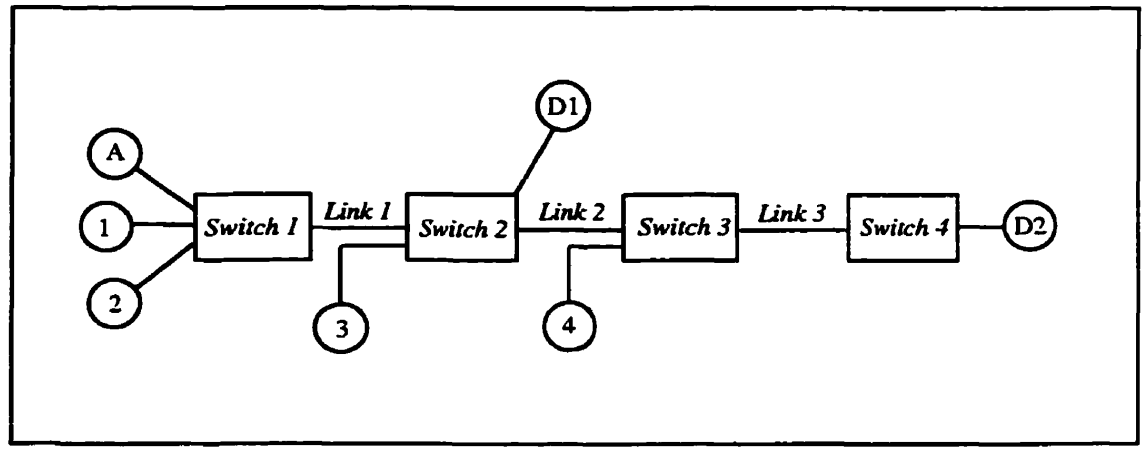

Figure 3.2: Sample network: upstream bottleneck

3.4.2 Upstream Bottleneck

Figure 3.2 shows a more general configuration. It comprises of two group: group 1 has one source S_A sending to destination D_1, group 2 has four sources (S_1,S_2,S_3,S_4) sending to destination D_2. $Link_2$ and $Link_3$ are 150Mbps but $Link_1$ is only 50Mbps. Apparently, both $link_1$ and $link_3$ are bottlenecks. Unless explicitly specified, all sources request a minimal of 0Mbps and a maximum of 200Mbps. The destination capacities for D_1 and D_A are all 150Mbps. We will use this configuration throughout this section. Tuning factors α_1, α_2, and α_3 allow great flexibility in using the proposed models. In this section, we will show different ways that the models can be used.

3.4.2.1 Overall throughput maximization

We choose $\alpha_1 > 0$ and $\alpha_2 = \alpha_3 = 0$. **Model 1** becomes a Linear Programming model to solve the traditional network max-flow problem. Fairness is then not an

issue and is not considered. The maximum network flow of the sample configuration is solved to be 200Mbps. The allocation vector delivering the maximal throughput is $(a_A,\ a_1,\ ,a_2,\ a_3,\ a_4) = (50,\ 0,\ 0,\ a_3,\ a_4)$, where a_3 and a_4 can be any positive numbers as long as $a_3 + a_4 = 150$.

3.4.2.2 Relationship between throughput and group fairness

We choose $\alpha_1 = 0$. As mentioned earlier, **Model 1** cannot be used with α_1 set to 0 because allocating 0 rate to all sources guarantees to minimize either fairness objectives or both. Instead, **Model 1a** should be used where the desired network throughput is expressed as a constraint (rather than an objective).

Assuming $\alpha_2 > 0$ and $\alpha_3 = 0$, **Model 1a** can show the relationship between the throughput and group fairness. Figure 3.3 shows the optimum group fairness (primary y-axis on the left) and corresponding group satisfaction level (the secondary y-axis on the right) for different network throughput expectations (x-axis). Here we modify (3.10) to be an equation in order to clarify the trend. It can be observed from Figure 3.3 that when the expected network throughput is less than or equal to 187.5, **Model 1a** can allocate source rates so that both groups are equally satisfied. Hence the group fairness is 0 (ideal). There is no conflict between throughput and group fairness in this range. When throughput is 187.5, the allocated rates for group 1 and group 2 are 37.5 and 150, respectively. According to our group fairness, the two group have the same satisfaction level 0.1875 (37.5/200 for group 1 and 150/800 for group 2). Only Link3 is saturated at this point. If more throughput is expected, we must move some resources of Link 1 from group 2 to to group 1. As a result, group 1 receives

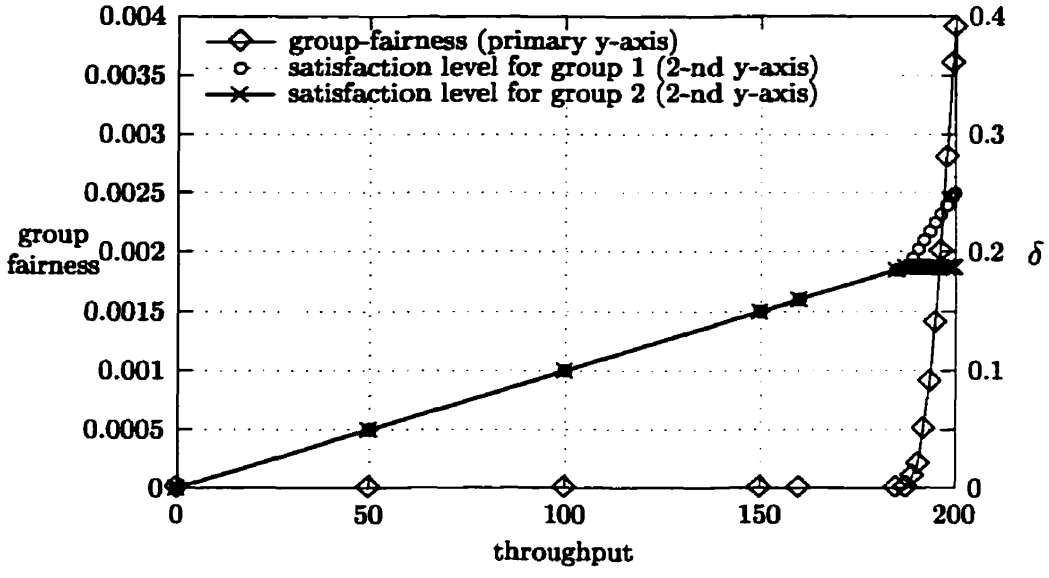

Figure 3.3: Relationship between throughput and group fairness

relatively more resources and has a higher satisfaction level. When all resource on Link1 are given to group 1 the network reaches the maximum throughput, leaving no resources for sources S_1 and S_2. Here maximizing throughput comes at the expense of group fairness.

3.4.2.3 Relationship between throughput and source fairness

Similarly, **Model** 1a can also be used to show the relationship between throughput and source fairness, if we choose $\alpha_2 = 0$ and $\alpha_3 > 0$. The specific value of α_3 is not relevant in this case. In Figure 3.4, the x-axis represents the expected network throughput c, which ranges from 0 to 200. The primary y-axis is the source fairness while the secondary y-axis is the source satisfaction level.

In Figure 3.4, when the expected throughput is less than or equal to 100, **Model 1a**

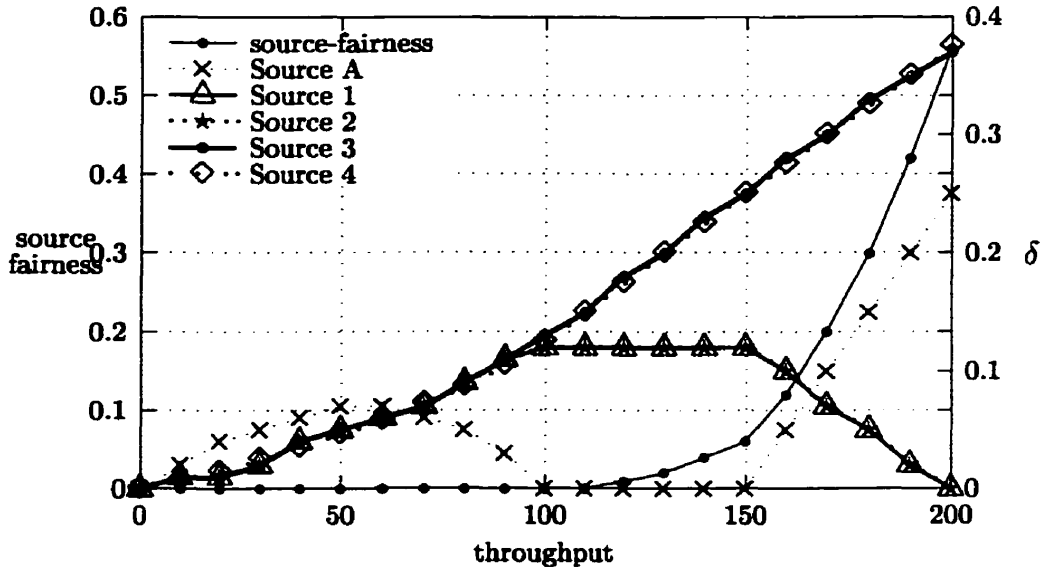

Figure 3.4: Relationship between throughput and source fairness

can allocate source rates so that all sources in group 2 are equally satisfied. Because group 1 only has one member, it does not participate in the source fairness calculation, its source fairness is 0. When throughput is 100, the allocation vector $(S_A, S_1, S_2, S_3, S_4) = (0, 25, 25, 25, 25)$. Only Link1 is saturated at this point. If more throughput is expected, S_3 and S_4 can be allocated higher rates until Link 3 becomes saturated. Then we have the allocation vector $(S_A, S_1, S_2, S_3, S_4) = (0, 25, 25, 50, 50)$, for a total throughput of 150. When the throughput exceeds 150, some bandwidth at Link 1 must be allocated to S_1. As a result, S_2 and S_3 get less and less throughput until all resources on Link 1 are given to S_A and the network reaches the maximal throughput 200. In the course of increasing the throughput, source fairness degrades. This experiment clearly shows the trade off between throughput and source fairness.

3.4.2.4 Relationship between group fairness and source fairness

When choosing $\alpha_2 > 0$ and $\alpha_3 > 0$, **Model 1a** can be used to compute the optimal allocation that achieves both group fairness and source fairness. In order to reveal the relationship between source fairness and group fairness, we assume the network to deliver a throughput of 100 and assume $\alpha_3 + \alpha_2 = 100$. Figure 3.5 shows the trends of group fairness and source fairness when α_2 changes from 0 to 100. It is shown that when we increase the weight of group fairness in the objective function by increasing the value of α_2, the group fairness is improved (becomes smaller). On the other hand, the source fairness deteriorates as it loses its relative weight to group fairness. The results show that group fairness and source fairness are indeed conflicting objectives. Similar results were observed when other values of the throughput were chosen.

Figure 3.6 further shows the source satisfaction levels and the groups' on secondary y-axis, which follows the same trend.

3.4.2.5 Combined throughput and group fairness

In 3.4.2.2, throughput is treated as a constraint, and we require the throughput to be equal to an expected value. Here we use **Model 1** to optimize rate allocation based on both throughput and group fairness. The relative importance of throughput and group fairness is controlled by the selection of α_1 and α_2. We set $\alpha_3 = 0$ to temporarily remove the source fairness from the objective function. Figure 3.7 shows the trends of throughput and group fairness with α_1 changing from 0 to 100, and $\alpha_2 = 100 - \alpha_1$

Compared to the experiment in 3.4.2.2, using **Model 1** gives control of throughput

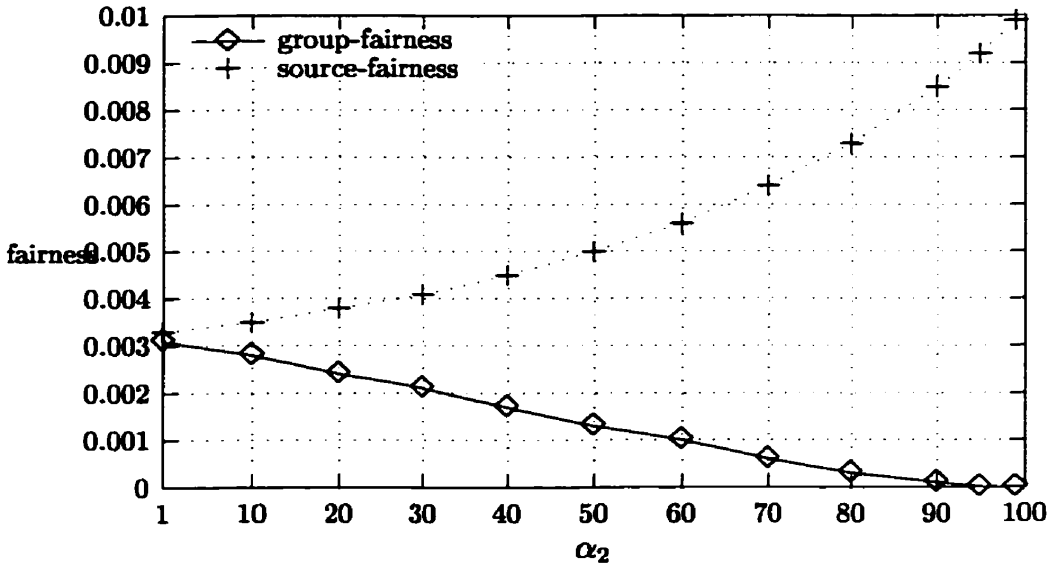

Figure 3.5: Relationship between group fairness and source fairness

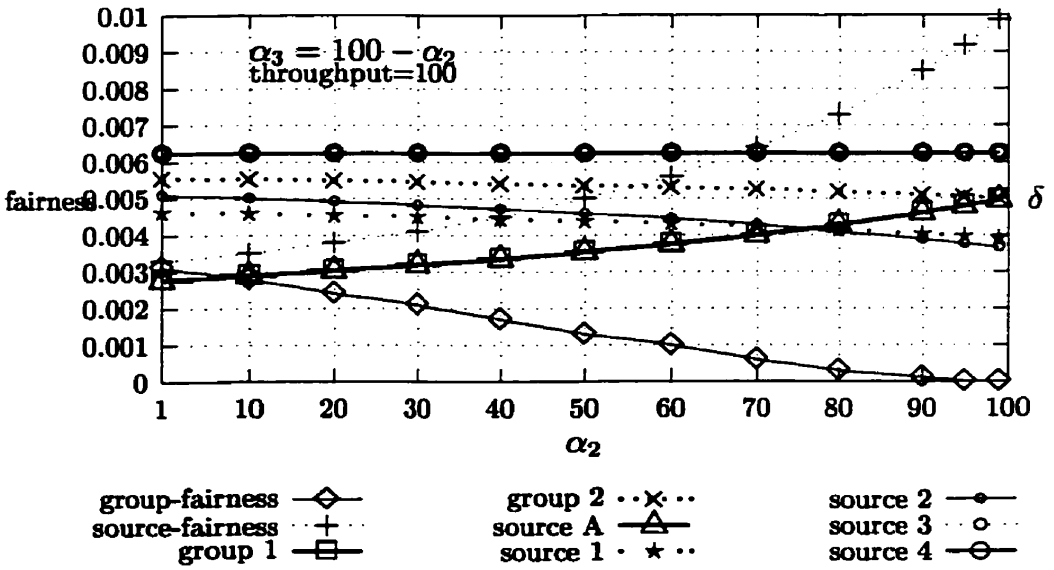

Figure 3.6: Relationship between group fairness and source fairness (with source δ)

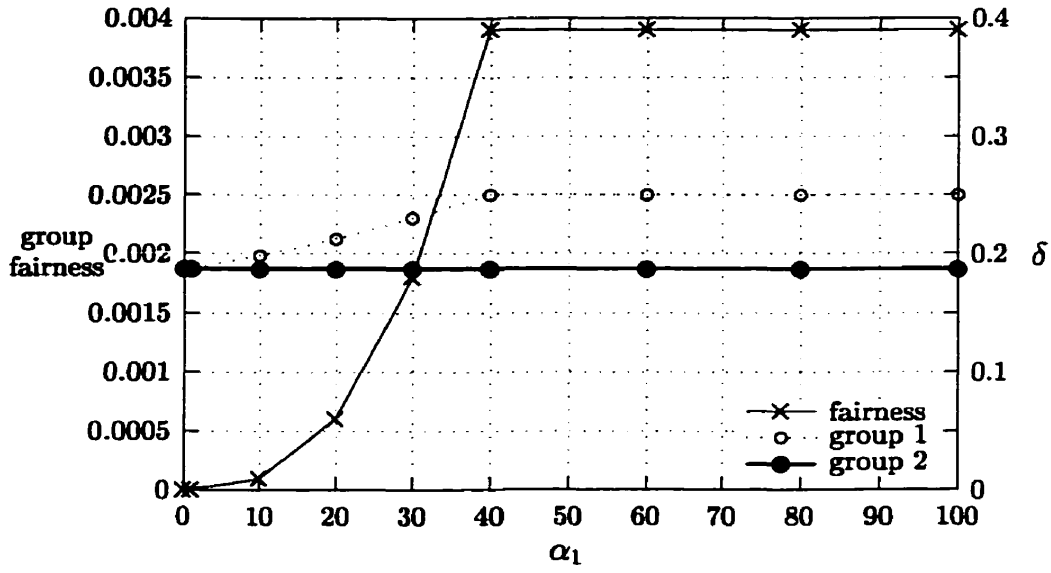

Figure 3.7: Effect of α_1 and α_2 on throughput and group fairness

to the decision maker. By choosing the values of α_1 and α_2, one can always find a solution that best optimizes the combined objectives. The model tries to find an optimal solution that optimizes both objectives. In our case, the network is able to deliver an overall satisfaction level of 0.1875 (the ratio of throughput 187.5Mbps to request 1000) while maintaining zero (ideal) value for group fairness for both groups.

3.4.2.6 Combined throughput and source fairness

Similar to 3.4.2.5, using **Model 1** with $\alpha_1 > 0$, $\alpha_2 = 0$ and $\alpha_3 > 0$ will compute the optimal allocation based on both the throughput and source fairness objectives. Figure 3.8 shows the experimental result when varying α_1 from 0 to 100 with $\alpha_3 = 100 - \alpha_1$. The trend in Figure 3.8 is very similar to that in Figure 3.4. Because both throughput and source fairness are represented in the objective function, the optimal

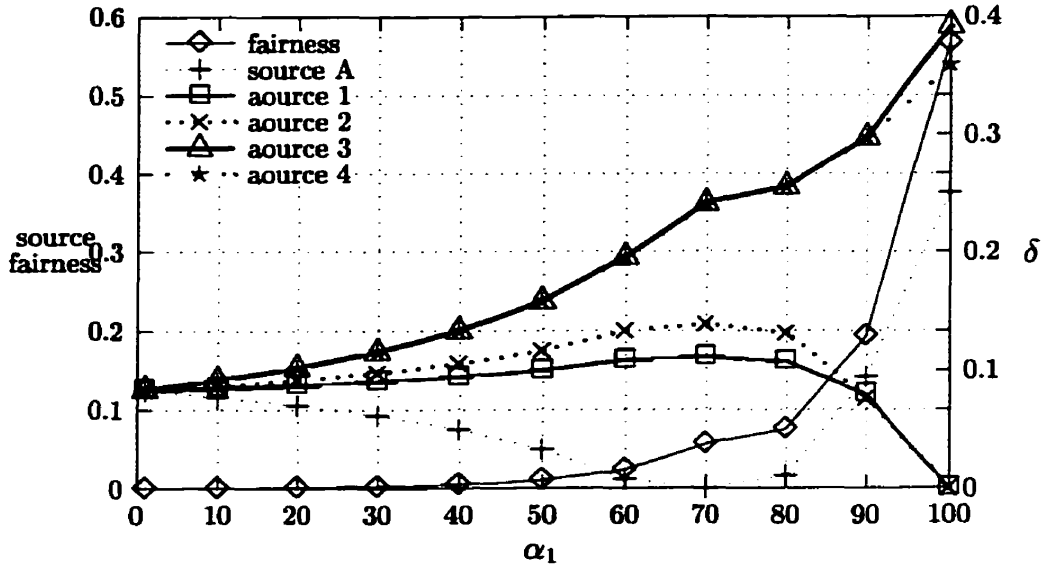

Figure 3.8: Effect of α_1 and α_3 on throughput and source fairness

allocation delivers at least an overall satisfaction level of 0.125 while maintaining zero (ideal) value of source fairness for all sources.

In addition, one can see that when the throughput approaches the maximal value the fairness becomes more sensitive to the increase of throughput. Comparing Figure 3.8 and 3.7 one can see that source fairness may be harder to accomplish than group fairness, especially when throughput approaches its maximum value. This can be explained by the fact that source fairness needs every single source to be satisfied approximately at the same level, while group satisfaction level measures only the aggregate for a group.

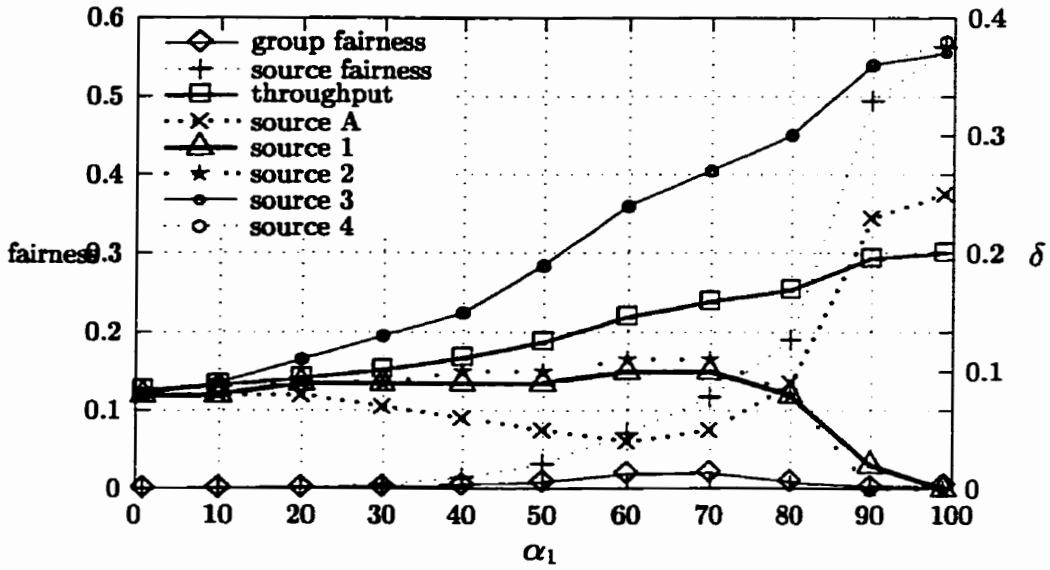

Figure 3.9: Trading throughput for combined fairness

3.4.2.7 Optimization on all three objectives

The real power of **Model 1** can be demonstrated by setting all tuning factors to non-zero values, i.e., $\alpha_1 > 0$, $\alpha_2 > 0$, and $\alpha_3 > 0$. Hence the optimal allocation achieved by **Model 1** would consider all three objectives.

Figure 3.9 illustrates the trend of all three objectives when varying α_1 from 0 to 100. We assume group fairness and source fairness have the identical values $\alpha_2 = \alpha_3 = (100 - \alpha_1)/2$. (Note this does not mean they are equally important to decision makers.)

A number of different experiments were carried out testing different values of α_1, α_2, and α_3. In general, increasing (or decreasing) one tuning factor while keeping all other parameters unchanged increases (or decreases) the weight of the objective

(represented by the changed tuning factor) in the combined objective function.

3.5 Summary

In this chapter, we have presented our fairness definitions and a theoretical optimization model to compute the optimum rate allocation based on multiple objectives. Our flow allocation model possesses some significant characteristics when compared with other rate allocation schemes.

- **Multiple objectives:** We have shown that flow control for multipoint-to-point communication can be based on the optimization of three different objectives: network throughput, group fairness, and source fairness. To the best of our knowledge, no other optimization-based scheme takes this approach. Some schemes (such as max-min fairness and proportional fairness) take into considerations of throughput and fairness. But the optimization is based only on fairness measures.

- **Global optimization:** Our optimization model can achieve global optimization because 1) our fairness definitions are based on source requests, and 2) all objective functions are global functions. It is these global objectives that relate the different links of the network even if they do not directly share any network resources. For instance, source S_1 and S_5 in Figure 1.1 do not share any network resources if the destination D_1 has infinite receiving capacity. In our model, changing the request of S_1 may affect the allocated rate for S_5. This is because the source fairness objective function includes a term for any pair of

sources regardless the network topology.

- **Tuning:** Flexibility is a key advantage of our model. Decision makers can adjust the model by changing the tuning factors so as to reflect their special requirements. By properly choosing the values of α_1, α_2 and/or α_3, one can obtain a full spectrum of optimal solutions, covering extreme interest in particular objectives and balanced interest as well. For group fairness for instance, by varying α_2 from a very small number to a large number, one can get various optimal rate allocations ranging from a very good group fairness to a very poor group fairness (reflecting the decision makers' interest in overall throughput). Existing optimization-based schemes do not have the capability to adjust the algorithm to favor some objectives over others.

Chapter 4

Distributed Rate Optimization Algorithm

To solve the rate allocation problem using the optimization model presented in the previous chapter requires information about the whole network. While theoretically feasible, this method, however, is not practical for wide area networks. This is because no single computer/node can collect all required network information, solve the model, and responds to other nodes in a reasonable time frame. Instead, the model should be implemented in a decentralized way such that the decision problem is decomposed into many smaller ones, which are then solved on different network elements across the network and in a timely manner.

This chapter presents such distributed schemes and discuss potential implementation issues. Section 4.1 presents a distributed optimization framework proposed by S. Low [21]. Using this framework, we are able to compute a rate allocation that

achieves throughput maximization (Section 4.2). Section 4.3 aims to optimize the source fairness objective and proposes a distributed scheme to obtain a sub-optimal (approximate) solution. In Section 4.4, we combine the two algorithms and propose an algorithm to compute allocation that optimizes both throughput and source fairness. Experimental results are presented and compared with those from the theoretical model in Chapter 3.

4.1 A Distributed Optimization Framework

S. Low [21] presents a distributed framework that can compute global optimal rate allocations in a distributed environment. This framework is based on source utility and resource pricing. Source s's utility $U_s(x_s)$ is a function that associates a scalar to the source's transmission rate x_s. The value of the utility function reflects the source's perception of the amount of service (or benefit) it receives when transmitting data at the given rate. Resource pricing allows a link to charge for use of its bandwidth. The objective of the flow control model is to maximize the aggregate source utility $\sum_{s \in S} U_s(x_s)$, where S is the set of all sources. Solving this problem centrally would require not only the knowledge of all utility functions but also the complex coordination among potentially all sources due to coupling of sources through shared links. Instead, an equivalent *dual problem* is solved because the natural structure of the dual problem suggests treating the network links and the sources as processors of a distributed computation system. Appendix C provides detailed steps of how to convert the centralized optimization problem to the decentralized optimization dual problem [21].

The algorithm is an iterative process where information is exchanged between a link algorithm executed at all links and a source algorithm executed at all sources. In each iteration, a source s individually determines its transmission rate by solving a local optimization problem

$$\max \ (U_s(x_s) - x_s p^s) \tag{4.1}$$

where $p^s = \sum_{l \in L(s)} p_l$ is the aggregate link price p_l of links in source s's path to destination, $L(s)$, and $x_s p^s$ is the bandwidth cost for source s. The computed rate is then communicated to all links $l \in L(s)$ in its path to destination.

In turn, link l then updates the link price p_l based on the updated source rates

$$p_l(t+1) \ = \ \max\{0, \ p_l(t) + \gamma(x^l(t) - c_l)\} \tag{4.2}$$

where x^l is the total allocated rate at link l, $\gamma > 0$ is the step-size. t, $t+1$ are time points when link price is reevaluated. The new price is then communicated to sources s. According to (4.2), link price increases or decreases depending on whether the link is overloaded. For under-loaded links, their prices eventually drop to 0. If all parameters are properly selected, the algorithm converges to a point where the rate allocations maximizes the aggregate source utility. See [21] for proof of convergence.

4.2 Maximizing Network Throughput

The framework in Section 4.1 can be used to solve the max-flow problem by properly choosing the source utility function. We first extend the network model described in Section 3.1 using the concept of utility. Source s attains a utility $U_s(x_s)$ when

it transmits at rate x_s that satisfies $m_s \leq x_s \leq M_s$. U_s is assumed increasing and strictly concave (convex upwards) in its argument. Let $I_s = [m_s, M_s]$ denote the range in which source rate x_s must lie.

We can identify a utility function $U(x) = x$ for all sources so that maximizing aggregate source utility would be equivalent to maximizing the overall network throughput $\sum_{s \in S} x_s$. However, this function can not be plugged into the framework because it does not meet the following conditions for utility functions. These conditions are required in order for the algorithm to converge (detailed explanation can be found in [21]).

- C1: On the interval $I_s = [m_s, M_s]$, the utility functions U_s must be increasing, strictly concave, and twice continuously differentiable.
- C2: The curvatures of U_s are bounded away from zero on I_s.

To overcome this issue, we need to define a utility function that meets these conditions yet represents a good approximation of the desired utility function $U(x) = x$. The utility function we have chosen for this purpose is

$$U(x) = r(1 - (1 - \frac{x}{r})^k), \quad k > 1 \tag{4.3}$$

Here we simplify the bandwidth request by assuming $m_s = 0$ and $M_s = r$. Figure 4.1 depicts the function $U(x) = x$ and (4.3) under different values of k. Apparently, k should be chosen as close to 1 as possible in order to get a good approximation .

With the utility function represented by equation (4.3) the subproblem (4.1) for source s becomes

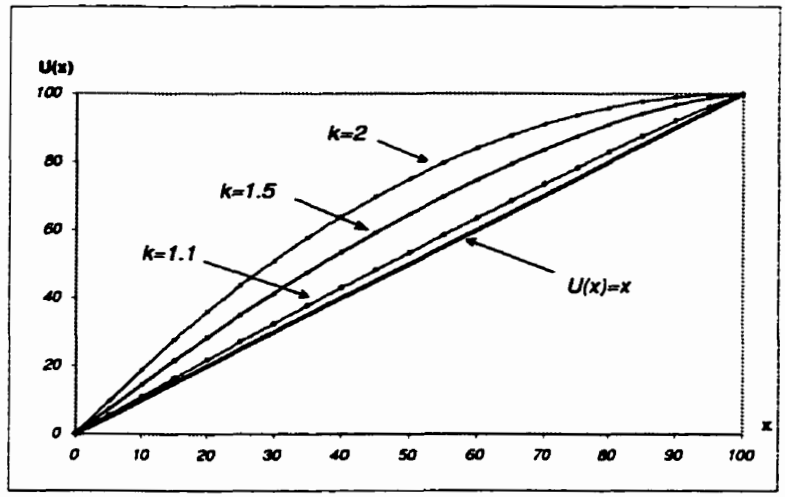

Figure 4.1: Utility function to maximize network throughput (r=100)

$$\max \quad r(1-(1-\frac{x}{r})^k) - x_s p^s, \quad p^s = \sum_{l \in L(s)} p_l \tag{4.4}$$

The optimum solution for source s can be analytically derived as

$$x_s = max(0,\ min(r,\ r(1-(\frac{p}{k})^{\frac{1}{k-1}}))) \tag{4.5}$$

For example, when $k = 2$, $x_s = max(m_s,\ min(M_s,\ r(1-p/2)))$.

The synchronous algorithm at links and sources are described in Figure 4.2. The algorithm is synchronous because it assumes that sources and links change state and exchanges information synchronously. The links state is represented by the link price and source state by the sending rate. At links, new price is computed based on the

Router link algorithm:
At times $t = 0, 1, 2, ...$, link l:

1. Receive rates $x_s(t)$ from all sources $s \in S(l)$, where $S(l)$ is the set of sources that use link l.
2. Compute a new link price using Equation (4.2)
3. Send new price $p_l(t+1)$ to sources $s \in S(l)$.

Source algorithm:
At times $t = 0, 1, 2, ...$, source s:

1. Receive from the network the aggregate link price of all links in s's path $p^s(t) = \sum_{l \in L(s)} p_l(t)$.
2. Compute a new transmission rate $x_s(t+1)$ using Equation (4.1). Source s will send data at this rate until the next calculation at time $t+2$.
3. Send new rate $x_s(t+1)$ to links $l \in L(s)$ in its path.

Figure 4.2: Synchronous algorithm for throughput maximization

previous link price and the current congestion state. A congestion at a link causes the link price to increase while the absence of congestion causes the link price to decrease. At sources, a source always chooses a sending rate that maximizes its net gain (utility - bandwidth cost). A source's new state does not depend on its previous states.

Figures 4.3-4.6 illustrate the convergence process of this synchronous algorithm for the upstream bottleneck configuration in Chapter 3 (Figure 3.2). Variables monitored are link prices of all links (Figure 4.3), source sending rate of all sources (Figure 4.4), link load on all links (Figure 4.5), and bandwidth cost for all sources (Figure 4.6). The x-axis in Figures 4.3-4.6 is the time series ($t = 0, 1, 2, ...$). We have used the utility function given by Equation (4.3) with k=1.1. The step size γ is set to 0.0002.

The algorithm starts with the initial link prices set to 1 and initial source sending rates set to 0. In the first two iterations, S_A and S_4 obtain significant resources because they use only one link to reach their destinations (hence their bandwidth cost is 1). Other sources choose a rate of 0 because their bandwidth cost overwhelms their utility. The sources determine their sending rates without knowing if any link will be overloaded. For instance, $Link1$ has an assigned rate higher than its capacity. It is the link's responsibility to feedback this information to the relevant sources. $Link1$ increases its price to increase the bandwidth cost of the sources sharing the link (S_A). As a result, S_A responds by setting a lower transmission rate until $Link1$ is no longer overloaded. $Link2$ detects no congestion and starts to reduce its price. It continues until its price is 0 because $Link2$ is never a bottleneck for this configuration. However, when $Link2$'s price approaches 0, the bandwidth cost for S_3 becomes so low that S_3 decides to send more data. As $Link2$ continues to drop its price, S_3 increases its sending rate. Again, the participation of S_3 overloads $Link3$ which responds by increasing its price. This further forces S_4 to drop its rate. Because S_3 also uses $Link3$, S_3 is affected by the increased cost from $Link3$. This results in the noticeable oscillation in Figures 4.4 and 4.5. The oscillation stops when the price of $Link2$ reaches 0. At this point all source rates stabilize. The rate allocation vector at the stable point is $(S_A, S_1, S_2, S_3, S_4)$=$(50, 0, 0, 75, 75)$ and is exactly the same as the maximum network throughput (200), obtained in Chapter 3.

The following should be noted:

1. The selection of k does not affect the convergence of the algorithm as long as $k > 1$. However it affects the accuracy of the solution. For better accuracy, we should choose a value for k close to 1.

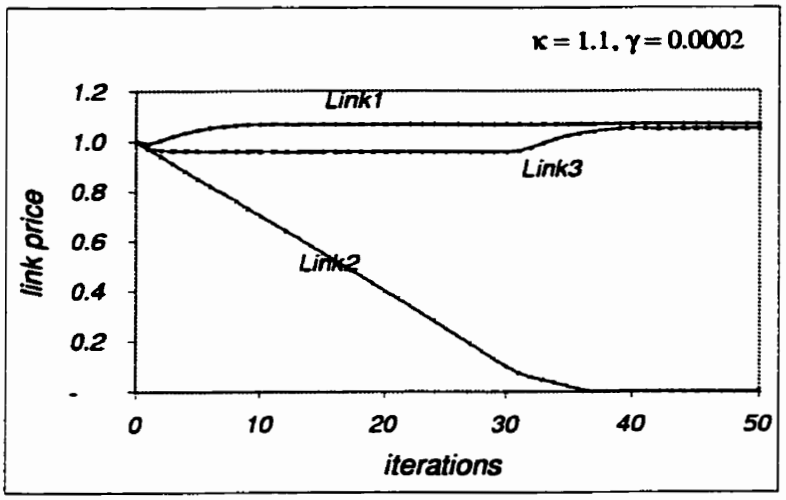

Figure 4.3: Convergence process: link prices

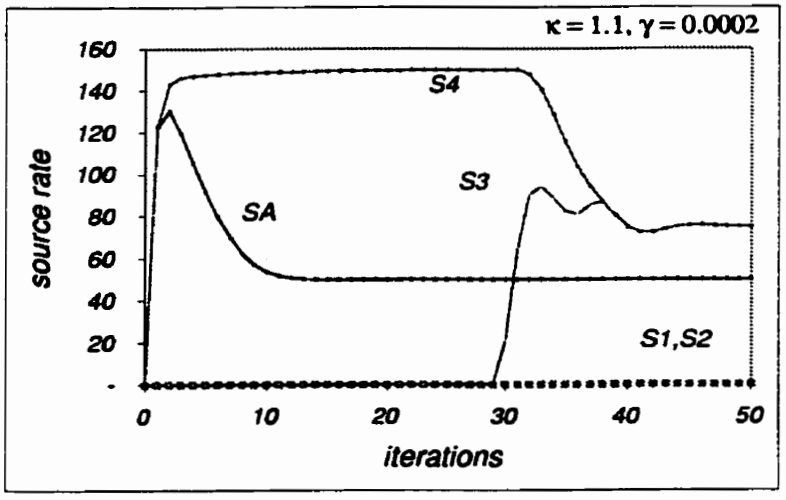

Figure 4.4: Convergence process: source rates

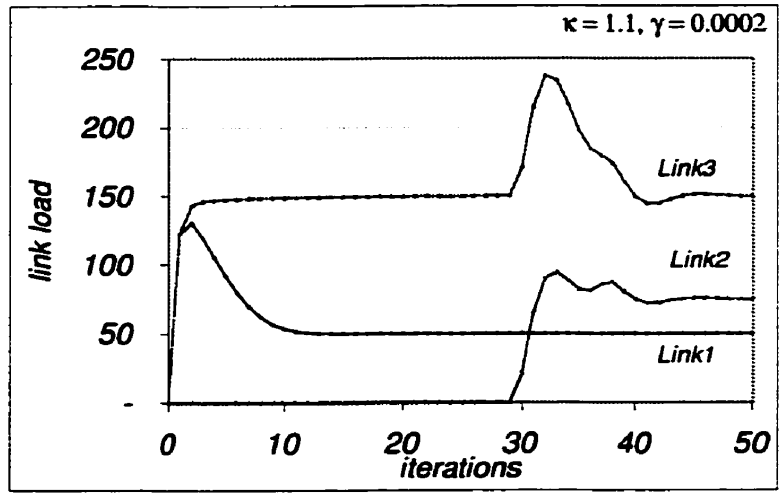

Figure 4.5: Convergence process: link aggregate rate

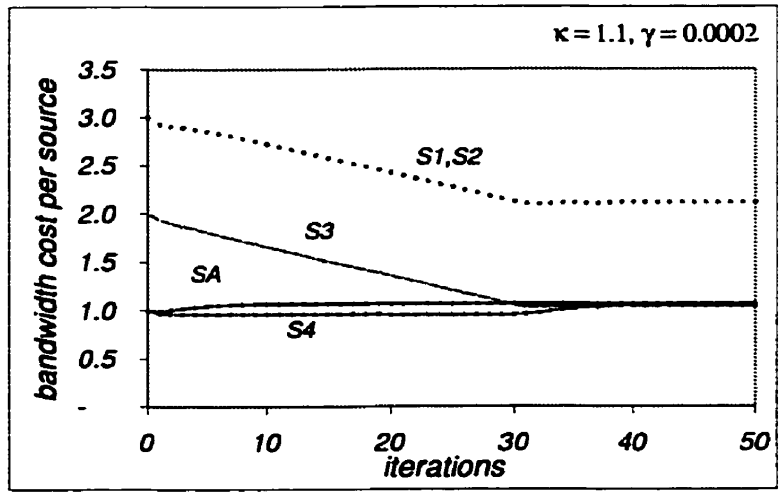

Figure 4.6: Convergence process: bandwidth cost per source

2. Within a certain range, a larger γ leads to faster convergence. For instance for $\gamma = 0.0003$, the algorithm converges after only 30 iterations. However, our experiments show that when γ exceeds a certain value the algorithm becomes unstable.

3. The procedure starts with all link priced at 1 per unit bandwidth (Figure 4.3). Again, the values of the initial link prices do not affect convergence but will affect the rate of convergence (or the number of iterations for the algorithm to stabilize), as it takes more steps for the link price to be adjusted to the desired value.

4.3 Source Fairness Optimization

The algorithm in the previous section is able to compute the maximum throughput (network flow) in a distributed environment. However, it cannot be used to solve the optimization problem for any of the fairness objectives. This is because the fairness objective functions are not separable [1].

In this section, we propose a distributed algorithm to allocate source rates based on the source fairness criteria. The algorithm is an approximate implementation of source fairness defined in Chapter 3. The basic idea is as follows:

Assume the network has a *preferred satisfaction level*, $\widetilde{\delta}$, $0 \leq \widetilde{\delta} \leq 1$ for all sources. The network is seeking a feasible allocation such that all source's satisfaction levels are equal to $\widetilde{\delta}$ (may not be possible in some cases). In the ideal situation, there exists

[1] An objective function is separable if it can be written as $\sum_{i=1}^{n} f_i(x_i)$.

such allocations that all sources are equally satisfied. Thus all sources and groups are treated absolutely equally and both source fairness and group fairness are at the ideal value (0). This can always be achieved by choosing a small $\widetilde{\delta}$. An extreme case would be let $\widetilde{\delta} = 0$. Assume we start with a small value $\widetilde{\delta} = \delta_0$ such that all sources get δ_0 of their request without overloading any link. Then we increase δ by a small value d such that the satisfaction levels of all sources can still be increased by d without overloading any link. We repeat this procedure until reaching a point where further increasing δ would result in at least one link becoming saturated and at least one source's satisfactory level to be less than $\delta + d$. We call the allocation at this point *marginal fair allocation*, and the satisfaction level achieved δ_1. Till now, sources on the upstream side of any saturated link have reached their highest possible rate (δ_1). We then remove the saturated links and sources on their upstream. The process above is repeated until the δ_n is greater than the preferred satisfaction level $\widetilde{\delta}$. Let $\delta_n = \widetilde{\delta}$. The allocation resulting from this procedure is called *equal satisfaction allocation* (it tries to assure equal source satisfaction levels as much as the network allows).

The algorithm of computing *equal satisfaction allocation* does not have to work as described above. In Figure 4.7, we present a more efficient algorithm that achieves equal satisfaction allocation. The algorithm assumes an underlying mechanism to exchange information between sources and routers/destinations.

This Algorithm consists of each source sending a control packet periodically to the destination along the path that data packets travel. The control packet contains the source's request, and a desired satisfaction level (δ_s). The link checks the control packets sent by all upstream sources and computes a "fair" δ (δ_{fair}) for that link. If

a source's desired δ (δ_s) is greater than the fair δ (δ_{fair}) at the link, δ_s in the control packet is reduced to the value of δ_{fair}, and a "reduced bit" is set in the control packet. The control packet is then forwarded to the next router on the path. The destination, when receiving the control packet, turns it around and sends it back to the source, which then adjusts its sending rate based on the value of δ_s. If the reduced bit is clear, the source could demand a higher level of service in the next control packet by setting a higher δ_s. If the bit is set, the source sets δ_s to the current satisfaction level.

This algorithm classifies sources at a given link to sources bottlenecked on this link and sources bottlenecked elsewhere (on other links). Here, the term *bottleneck* link is redefined to be the link that supports the least satisfaction level. A link always makes bandwidth available first to the sources bottlenecked elsewhere and then share the left over bandwidth to the sources bottlenecked on it. The sharing is based on δ_{fair}

δ_{fair} is computed using an iterative procedure described as follows. Initially, δ_{fair} is set to $\frac{link\ bandwidth}{total\ source\ request}$. Sources with δ_s value less than δ_{fair} are obviously not bottlenecked on this link. Hence, their desired δ is granted at the link. The fair δ is then recalculated using (4.6) below.

$$\delta_{fair} = \frac{Link\ Capacity - Total\ bandwidth\ allocated\ to\ sources\ bottlenecked\ elsewhere}{Total\ source\ requests - Total\ request\ of\ sources\ bottlenecked\ elsewhere} \tag{4.6}$$

The updated value of δ_{fair} is compared to δ_s of all sources. Sources with lower δ_s are not bottlenecked at this link. This process continues until all the sources left are bottlenecked on this link or δ_{fair} is already greater than the preferred δ, in which case set δ_{fair} to $\widetilde{\delta}$.

Source algorithm:
Each source stores its current rate request (**Req**), current sending rate (**Rate**), and a none-congestion indicator (**NCI**). The initial value of **Rate** is 0. **NCI** is 0 when the path is congested or 1 otherwise.

1. Periodically send a control packet to the destination along the data path. The control packet contains **Req**, a desired δ, and a reduced bit. The reduced bit is set to 0 for all outgoing control packets. The desired δ is initially set to the global preferred δ ($\widetilde{\delta}$). In later packets, set

$$\delta_s = \min(\widetilde{\delta},\ (\mathbf{Rate/Req} + \mathbf{NCI} \times \mathbf{IF}))$$

   where $\widetilde{\delta}$ is the preferred satisfaction level, **IF** is the increase factor for δ_s.

2. On receiving a returned control packet:

   (a) **Rate** $\leftarrow \delta_s \times$ **Req**

   (b) **NCI** $\leftarrow$ 1 if the reduced bit is clear, 0 otherwise.

Router link algorithm:
A router stores a value for δ_{fair} for each of its links.

1. Periodically update δ_{fair} for its links based on the latest source request and δ_s.

2. On receiving a control packet from a source:

   (a) if $\delta_s > \delta_{fair}$ then $\delta_s \leftarrow \delta_{fair}$, where δ_{fair} is calculated using Equation (4.6).

   (b) Forward the control packet to the next hop router.

Destination algorithm:

1. Upon receiving a control packet, return it to the source.

Figure 4.7: Source fairness rate allocation algorithm

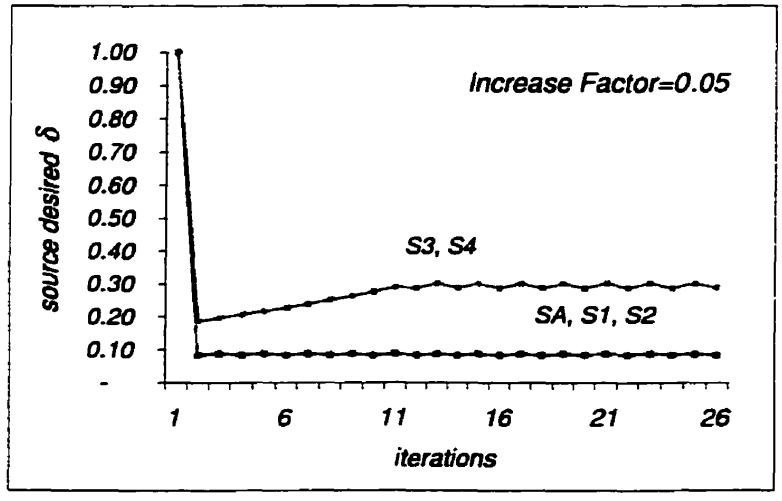

Figure 4.8: Convergence process of source desired δ

Figure 4.8 shows the convergence process of desired δ (δ_s) for all sources in the upstream bottleneck configuration (Figure 3.2). The algorithm starts with all sources setting their δ_s to the value of the global preferred δ ($\widetilde{\delta}$), which is set to 1. For sources S_A, S_1, and S_2, their first returned control packet has δ_s reduced to 0.083 (δ_{fair} of $Link1$) and their δ_s stabilizes. For sources S_3 and S_4, their first returned control packet has the δ_s reduced to 0.1875. In the following iterations, $Link2$ and $Link3$ realize that S_2 and S_3 are bottlenecked elsewhere ($Link1$). The updated δ_{fair} values of $Link2$ and $Link3$ are greater than the δ_s. Receiving no indication of any congestion, source S_3 and S_4 increase their desired δ by 0.05 each time they send a control packet. This continues until congestion occurs at $Link3$. Then the value of δ_s for S_3 and S_4 stabilizes.

Figure 4.9 shows the individual source satisfaction levels (secondary y-axis) and

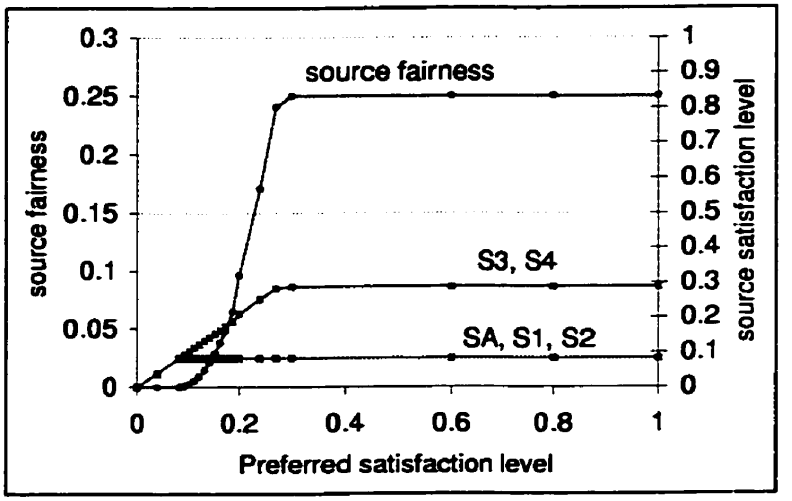

Figure 4.9: Control of source fairness using preferred δ

the corresponding source fairness (primary y-axis) at different levels of the global preferred δ ($\widetilde{\delta}$) (x-axis). When $\widetilde{\delta}$ is less than 0.083, all sources are fully and equally satisfied and the source fairness is 0. At $\widetilde{\delta} = 0.083$, $Link1$ reaches its capacity and S_A, S_1, and S_2 reach their rate limit. Sources S_3 and S_4 can still get fully satisfied if $\widetilde{\delta}$ is further increased. When $\widetilde{\delta}$ reaches 0.2875 $Link3$ becomes saturated. Further increasing the value of the preferred δ has no impact on either the source rates or source fairness. It should be noted that, when $\widetilde{\delta}$ is in the range from 0.083 to 0.2875, increasing $\widetilde{\delta}$ leads to worsening source fairness but improves the overall throughput.

4.4 Optimizing Throughput and Source Fairness

In this section, we attempt to integrate the algorithms presented in the previous two sections into one procedure to compute an optimum rate allocation that addresses both the throughput and source fairness optimization objectives. The key is to properly split the link capacities into two parts one for each of the two objectives. Assume we choose a small value for the preferred δ and run the source fairness algorithm in Figure 4.7. It is likely that link capacities are not fully utilized. The left over capacities can then be re-allocated solely based on the throughput maximization objective. We divide link capacity into two portions:

1. *fairness portion* that is allocated based on source fairness optimization requirement. The algorithm in Section 4.2 is used to compute the optimum allocation, which is called *fairness portion source rate.*

2. *flow portion* that is allocated to maximize the overall throughput. The algorithm in Section 4.3 is used to compute the optimum allocation, which is called *flow portion source rate.*

Similarly, source rate is the sum of flow portion source rate and fairness portion source rate. Depending on the value of the preferred δ, the two portions can be of different sizes. The fairness portion may be 0 if the preferred δ is set to 0. Increasing the preferred δ may increases the fairness portion up to full link capacity.

Because of the iterative nature of both algorithms, the two algorithm can share one control packet to exchange information among sources, links, and destinations.

The contents of the control packet includes: source request, desired δ (for fairness portion allocation), current sending rate (for flow portion allocation), and bandwidth cost (for flow portion allocation). At each link, the fairness portion of link capacity is determined based on the current sending rate of the sources and the fair δ computed using (4.9). The left over capacity is then used to update the link price and is added to the bandwidth cost field in the control packet. Since each link adds its link price to bandwidth cost field. When the control packet is returned to the source. The field value is exactly the aggregate link price for the source.

It should be noted that both algorithms run at the same time. Hence the number of iterations needed to compute an optimum allocation that optimizes source fairness may not increase.

Figure 4.10 and 4.11 present the experimental results of the combined algorithm for the network in Figure 3.2. We vary the value of the preferred δ from 0 to 1. At each value, the algorithm computes a fairness portion and a flow portion of source rates. The overall throughput is the sum of both portions. Figure 4.10 displays the trend of contribution of fairness portion and flow portion throughput. The upper curve represents the overall throughput (marked by "overall throughput"). It is shown that, when the value of the preferred δ is less than 0.2875, the preferred δ effectively controls the relative weights of both objectives in the optimization. Figure 4.11 presents the relationship between the overall throughput and the optimum source fairness.

Comparing these results with those from the theoretical model in Figure 3.8, we can see that they produce similar trend, i.e., increasing overall throughput causes

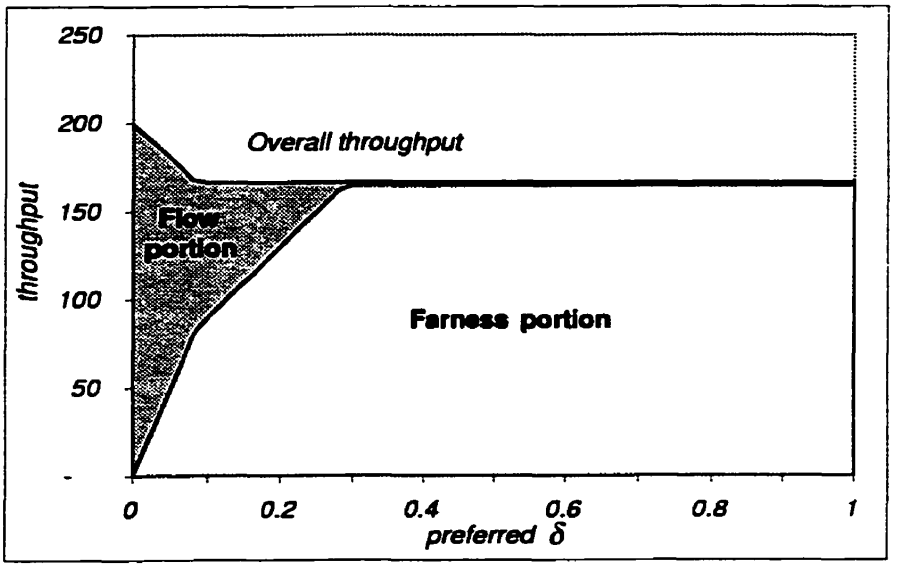

Figure 4.10: Throughput: fairness portion and flow portion

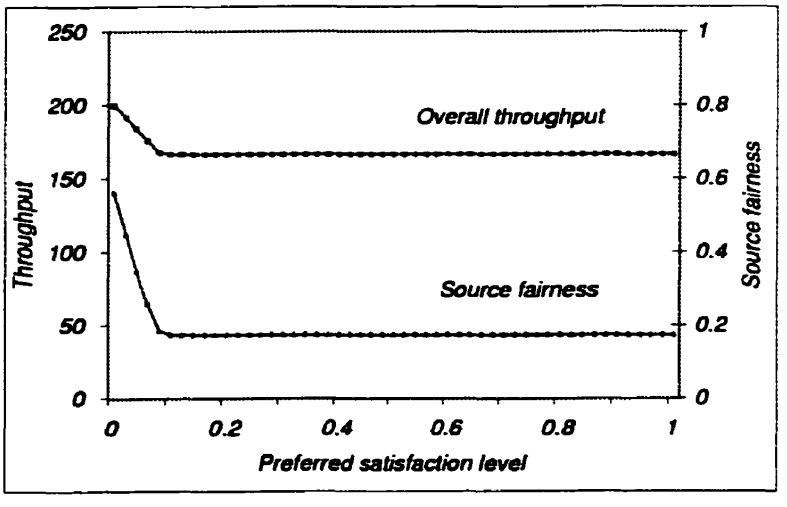

Figure 4.11: Combined scheme: throughput vs. source fairness

source fairness to deteriorate.

It is also interesting to compare our scheme to the max-min based scheme. We argue that our scheme is more flexible than max-min scheme. Assume all sources request the same amount of bandwidth from the network and the global preferred source satisfaction level is 1, our scheme would compute an allocation that is the same as the max-min fair allocation, which is $(S_A, S_1, S_2, S_3, S_4) = (16.7, 16.7, 16.7, 58.3, 58.3)$. However, our scheme offers a flexibility to balance between the throughput and source fairness by setting the value of preferred δ. For example, setting $\widetilde{\delta} = 0.05$ results in an optimum rate allocation of $(S_A, S_1, S_2, S_3, S_4) = (30, 10, 10, 65, 65)$ which delivers an overall throughput of 180, as opposed to 167 for max-min allocation.

4.5 Summary

We have proposed a distributed scheme that assigns source sending rates based on a global optimization of source fairness and overall throughput. By increasing or decreasing the global preferred δ, the scheme can assign source rates that achieve better source fairness or higher throughput. Although the experiments assumed static source request and link capacity, the proposed scheme also works in dynamic environments, where source requests and link capacities may change. The scheme is also extensible to the dynamic group membership case, where sources may join or leave the multipoint-to-point group.

Chapter 5

Conclusion

5.1 Concluding Remarks

In this work, we have modelled the flow control of multipoint-to-point communication as a multiple-objective optimization problem to reflect the multi-objective nature of this task. We have presented a theoretical model to validate this idea and have proposed a distributed scheme to implement the framework in practical environments. Our results suggest that this framework is feasible, powerful, and flexible since it allows flow control to meet a wide spectrum of decision preferences in a controlled manner.

We have proposed a new metric - satisfaction level (δ) - to measure the level of satisfaction for individual sources or group of sources. This measurement normalizes source-sending rates to a real number between 0 and 1. Based on the satisfaction level, we have defined new fairness measurement criteria to be the least square distances of

δ between any two sources or groups. This measurement actually introduces weights into the system. Sources or groups with higher weight (more request bandwidth) are likely to get more bandwidth than those requesting less. However there is no linear relationship between requested bandwidth and allocated bandwidth due to the limitation of network resources.

Based on the definition of fairness, we have defined theoretical models to optimize bandwidth allocation based on three different objectives: network throughput, source fairness, and group fairness. The model formulated is a quadratic programming model with linear constraints. The three objectives are unified into one using weighted-averages. Weighting (tuning) factors can be changed to adjust the weights of different objectives in the unified objective function. This is a unique and important feature of our model allowing it to reflect various decision preferences from different decision makers. Experiments have been conducted against a typical configuration and demonstrated that basing rate allocation on the optimization of multiple objectives is theoretically feasible and offers rate assignment flexibility not available to earlier existing schemes.

We have also developed practical distributed schemes that provide similar result as the proposed theoretical model but in a distributed environment. We have successfully extended the "link pricing/source utility function" framework (proposed by Kelly [8, 19] and Low [21, 22]) to solve the network throughput maximization objective. A source utility function has been designed for this purpose. Experiments have shown that the designed utility function can compute allocations that maximizes network throughput in any network configuration.

In order to deal with the objective of source fairness, an alternative scheme with built-in approximation has been proposed. This scheme "max-min" fairly [1] allocates bandwidth based on sources' satisfaction levels (rather than the actual bandwidth). The normalized allocated bandwidth is subject to a satisfaction level cap, which is the global preferred satisfaction level. This scheme has been combined with the throughput maximization scheme to become a unified scheme that optimizes bandwidth allocation on both objectives. The global preferred satisfaction level becomes the tuning factor used to control the weights of the fairness measures in the unified objective function. Experiments have been conducted and shown a similar relationship between source fairness and network throughput.

We argue that our scheme is better than max-min scheme [3] because our scheme offers more flexibility. Indeed, max-min fair allocation is actually a special case of our scheme. When the global preferred satisfaction level is set to 1 and all sources request the same amount of bandwidth from the network, our scheme generates the same results as the max-min algorithm.

5.2 Future Work

It is extremely difficult to find distributed algorithms that optimize a global objective. We have identified the following extensions to our works:

- We still need a distributed algorithm that can optimize bandwidth allocation solely based on the group fairness objective function. This algorithm should

[1] We use the term "max-min" in the sense that all bottlenecked sources get equal rate in proportion to their request

then be integrated within our proposed framework. In the current algorithm, knowledge and decisions are distributed among sources and links. In order to make a decision for a connection or group, it might be desirable to have the decision made at a network element that has the knowledge of the entire group. The destination node is a logical choice. It may be useful to define a destination utility function to measure the utility gained at the destination for the entire group. Then the source rate may be determined by the group's destination.

- We view that a natural placement of our proposed framework would be within the Internet Integrated Service Architecture (ISA) [28]. It is then important to investigate the implementation of our proposed scheme using RSVP (Resource Reservation Protocol) [29, 30]. Here RSVP is used not only to reserve bandwidth resources to deploy the assigned rate for sources but also to serve as a mechanism for flow control to pass various parameters between different network elements. RSVP can be extended for this purpose since it has many characteristics that can be used for multipoint-to-point communication. RSVP makes reservations for unidirectional data flows from the senders to the destination. RSVP shares a number of attributes as our proposed multipoint-to-point rate allocation framework. First, control and data messages follow the same path. Second, RSVP assumes that routing has been already set by other protocol.

  In addition, RSVP is a "soft-state" protocol. With the periodical messages sent from sources and destinations, flow control parameters can be exchanged between different network elements. This makes RSVP very suitable for multipoint-to-point sources to periodically send control packets to the routers and the destination.

New RSVP object classes are needed to convey data from sources to destination and back to sources. Path messages and Resv messages should be modified to include the new defined objects. These new objects may be modified at routers and forwarded to next hop routers.

Bibliography

[1] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal, and Mukul Goyal, "QoS and multipoint support for multimedia applications over the ATM ABR service," *IEEE Communications Magazine*, vol.37, no.1, January 1999, pp.53-57.

[2] J. Wroclawski, "Specification of the controlled-load network element service," *Internet RFC 2211*, September 1997, http://www.ietf.org/rfc/rfc2211.txt.

[3] J. M. Jaffe, "Bottleneck flow control," *IEEE Transactions on Communications*, vol.29, no.7, July 1981, pp.954-962.

[4] Dimitri P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[5] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal and B. Vandalore, "On determining the fair bandwidth share for ABR connections in ATM networks," *Proceedings of IEEE International Conference on Communications (ICC)*, vol.3, June 1998, pp.1485-1491.

[6] The ATM Forum, *The ATM forum traffic management specification version 4.1*, March 1999, ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0121.000.pdf,

[7] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, and M. Goyal, "A definition of general weighted fairness and its support in explicit rate switch algorithms," *Proceedings of Sixth International Conference on Network Protocols 1998 (ICNP'98)*, October 1998, pp.22-30.

[8] F. Kelly, "Charging and rate control for elastic traffic," *Europ. Trans. Telecom*, vol.8, 1997, pp.33-37.

[9] L. Massoulié and J. Roberts, "Fairness and quality of service for elastic traffic," *CNET-France Télécom*, February 1998.

[10] P. Hurley, J. L. Boudec, and P. Thiran, "A note on the fairness of additive increase and multiplicative decrease," *ITC 16*, 1999.

[11] S. Fahmy, R. Jain, R. Goyal, and B. Vandalore, "ATM ABR multipoint-to-point connections and fairness issues," Department of Computer and Information Science, The Ohio State University, June 1999, http://www.cis.ohio-state.edu/ jain/papers/mptfair.htm.

[12] Sonia Fahmy, Raj Jain, Rohit Goyal, and Bobby Vandalore, "Fairness for ABR multipoint-to-point connections," *Proceedings of SPIE Symposium on Voice, Video and Data Communications, vol.3530, Conference on Performance and Control of Network Systems II*, November 1998, pp.131-142

[13] S. Fahmy and R. Jain, "ABR flow control for multipoint connections," *IEEE Network Magazine, ATM Forum Perspectives column*, September-October 1998, vol.12, Issue 5, pp.6-7.

[14] W. Moh and Y. Chen, "Design and evaluation of multipoint-to-point multicast flow control," *Performance and Control of Network Systems II, Proc. of SPIE Int. Symposium on Voice, Video, and Data Communications*, November 1998, pp.143-154.

[15] R. Jain, "Congestion control and traffic management in ATM networks: Recent advances and a survey," *Computer Networks and ISDN Systems*, vol.28, no.13, October 1996, pp.1723-1738.

[16] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," *ATM Forum-TM 94-0692*, July 1994.

[17] S. Habrah, H. Hassanein, and H. AboElfotoh, "Congestion control in multipoint-to-point LAN interconnection over ATM networks." *Proceedings of the International Conference on Computer and Information*, November 2000.

[18] H.S. Hassanein, H.M.F. AboElfotoh, S.K. Habra, "Dynamic resource-allocation for congestion-control in high-speed LAN interconnection," *Computer Communications* 22(1999), 1999, pp.1423-1439.

[19] F. Kelly, A.K. Maulloo, and DKH Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society 49 (1998)*, pp.237-252.

[20] R.J. Gibbens, F.P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, 35 (1999), pp.1969–1985.

[21] S.H. Low and D.E. Lapsley, "Optimization flow control I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol.7, no.6, December 1999, pp.861-874.

[22] S. Low, "Optimization flow control with on-Line measurement or multiple paths," *In Proceedings 16th International Teletraffic Congress*, vol.39, 1999, pp.237-249.

[23] David Lapsley and Steven Low, "An optimiation approach to ABR control," *In IEEE ICC'98*, June 1998.

[24] J. H. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on networking*, vol.8, no.5, October. 2000, pp.556-567.

[25] L. Massoulie and J. Roberts, "Bandwidth sharing: Objectives and algorithms," *Proceedings of the Infocom 99*, 1999, pp.1395-1403.

[26] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, *et al*, "The ERICA switch algorithm for ABR traffic management in ATM Networks," *IEEE/ACM Transactions on Networking*, vol.8, no.1, February 1997, pp.87-98.

[27] R. Jain *et al*, "ERICA+: Extensions to the ERICA switch algorithm," *ATM Forum/95-1346*, October 1995.

[28] R. Braden, D. Clark, S. Shenker, "Integrated services in the Internet architecture: an overview," *Internet RFC 1633*, June 1994, http://www.ietf.org/rfc/rfc1633.txt.

[29] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) – version 1 functional specification," *Internet RFC 2205*, September 1997, http://www.ietf.org/rfc/rfc2205.txt.

[30] Wroclawski, J, "The use of RSVP with IETF integrated services," *Internet RFC 2210*, September 1997, http://www.ietf.org/rfc/rfc2210.txt.

Appendix A

ATM ABR Service Model

ABR Flow Control Model

ABR point-to-point flow control occurs between a sending end-system (source) and a receiving end-system (destination). The two end-systems are connected via bi-directional connections. The standard ABR congestion control scheme is a rate-based, closed-loop mechanism that utilizes the feedback information from the network to control the rate of transmitting cells at the source. The source adapts its rate to the changing network conditions. Information about the state of the network like bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special control cells called Resource Management Cells (RM-cells).

Each ABR source generates RM cells in proportional to its current data cell rate. The source indicates its current rate to the network in a special field in the RM cell called the current cell rate (CCR). RM cells travelling from the source to the destination are called forward RM (FRM) cells. When the destination receives these

cells it turns them around and send them back to the source as backward RM (BRM) cells, see Figure A.1.

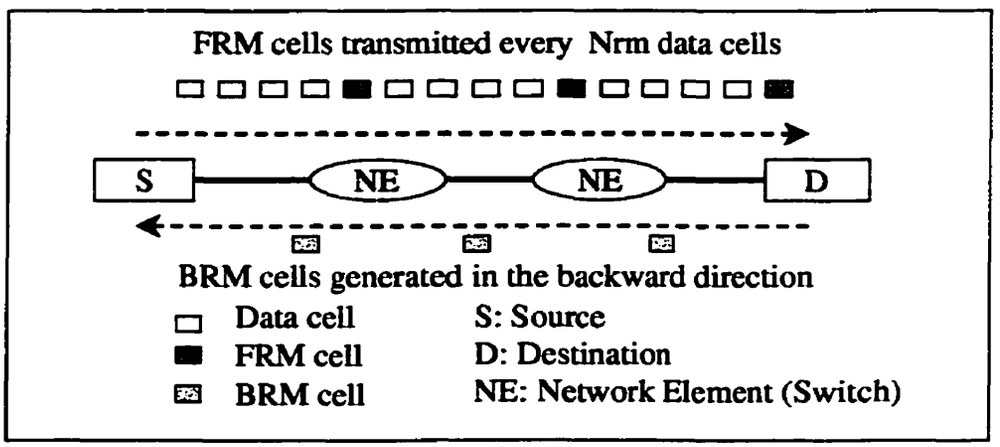

Figure A.1: ABR flow control model

The RM cells are examined by the network elements (ATM switches) and possibly modified in both directions to carry the feedback information of the state of congestion and the fairness. Three fields in RM cells are used to provide direct feedback and indirect information about the network. The fields are: - CI: The CI (congestion indication) bit allows a network element to indicate that there is congestion in the network. When a source receives a BRM-cell with CI=1 it decreases its ACR. - NI: The NI (no increase) bit is used to prevent a source from increasing its ACR. In contrast to CI=1, NI=1 does not require any decrease. A network element might set NI to 1 to indicate impending congestion. - ER: The ER (Explicit Rate) field is used to limit the source ACR to a specific value. For each RM-cell ER is set by the source to a requested rate (such as PCR). It may be subsequently reduced by any network element (include the destination) in the path to a value that the element can sustain.

Basic ABR Service Parameters

At the time of connection setup, ABR sources negotiate several parameters with the network. The most important parameters are the MCR (minimum cell rate, the rate at which the source is always allowed to send) and PCR (peak cell rate, the cell rate that the source may never exceed) of the connection. The call admission control would only accept the connection if its MCR requirement is less than the free bandwidth along the path to the destination. Once the connection is accepted the source starts sending data at ICR (initial cell rate, the rate at which a source should send initially and after an idle period) which is MCR. The first cell transmitted is always a RM cell and a new RM cell is scheduled every Nrm data cells. The source will continue to send data at ICR until it receives the first BRM and adjusts its rate accordingly. The rate at which the source is allowed to send data is called ACR (allowed cell rate, the current rate at which a source is allowed to send). This rate is dynamically changed between MCR and PCR in accordance with the BRM fields. To govern the rate by which the source changes its rate (either increase or decrease), two additional parameters are negotiated. They are the RIF (rate increase factor, that controls the amount by which the cell transmission rate may increase upon receipt of an RM-cell) and the RDF (rate decrease factor, which controls the decrease in the cell transmission rate.

The Source Behavior and Destination Behavior

Source Behavior When a FRM cell is to be scheduled - Set the CCR field in the RM cell to ACR - Set the ER field to PCR - Set the MCR field to MCR When a BRM cell is received - If CI=1 - ACR = max(min(ER, ACR-ACR*RDF),

MCR) "Multiplicative decrease" - else - if NI=1 - ACR=min(ER,ACR) - else - ACR=min(ER,ACR+RIF*PCR, PCR) "additive increase"

Destination Behavior - If destination is congested - Apply congestion control scheme - turn around BRM cel

Appendix B

MPL Modelling System

The optimization software packages we use in this research are MPL and CPLEX.

MPL (Mathematical Programming Language) is an advanced modelling system that allows you to set up complicated models, involving thousands of constraints, in a clear, concise, and efficient way and is extremely user-friendly and powerful. MPL offers a feature rich model development environment that takes full advantage of the graphical user interface in MS Windows, making MPL a valuable tool for developing LP models. MPL has features that allow you import data directly from a database and then after solving the problem export the solution back into the database. This along with the ability to be called directly from other Windows applications, such as databases and spreadsheets, make MPL ideal for creating end-user applications.

Models developed in MPL can be used with nearly all LP-solvers on the market today as MPL supports a number of industrial strength solvers that have recently been ported to Windows as DLL libraries as well as traditional DOS solvers.

For more information about MPL, see their web-site at http://www.maximal-usa.com/mpl/.

CPLEX is a world-class linear programming and mixed integer programming solver. The version we use is CPLEX for MPL, which gives MPL users access to CPLEX solver from within the Windows environment of MPL. For more information about CPLEX for MPL, please see http://www.maximal-usa.com/mpl/mplcplex.html.

Appendix C

Low's Optimization Framework

This appendix briefly describes Steven Low's optimization framework. For detailed description please refer to [21]. This framework is an optimization approach to flow control where the objective is to maximize the aggregate source utilities over their transmission rates. This framework converts the centralized optimization problem to several distributed less expensive optimization problems, each is solved by one source assuming limited knowledge accessible to the source. Thus this framework can compute a globally optimal solution in an entirely distributed manner.

Consider a network that consists of a set L of unidirectional links of capacities c_l, $l \in L$. The network is shared by a set S of sources, where source s is characterized by a utility function $U_s(x_s)$ that is concave increasing in its transmission rate x_s. The goal is to calculate source rates that maximize the sum of the utilities $\sum_{s \in S} U_s(x_s)$ over x_s subject to capacity constraints. Solving this problem centrally would require not only the knowledge of all utility functions, but the complex coordination among potentially

all source due to coupling of sources through shared links. Instead, a decentralized scheme is proposed to eliminate this requirement and adapt naturally to changing network conditions. The key is to consider the dual problem whose structure suggests treating the network links and the sources as processors of a distributed computation system to solve an equivalent dual problem using the gradient projection method. Each processor executes a local algorithm, communicates its computation result to others, and the cycle repeats.

The algorithm takes the familiar form of reactive flow control. Based on the local aggregate source rate each link $l \in L$ calculates a 'price' p_l for a unit of bandwidth at link l. A source s is fed back the scalar price $p^s = \sum p_l$, where the sum is taken over all links that s uses, and it chooses a transmission rate x_s that maximizes its own benefit $U_s(x_s) - p^s x_s$, utility minus the bandwidth cost. These individually optimal rates $(x_s(p^s),\ s \in S)$ may not be globally optimal for a general price vector $(p_l,\ l \in L)$, i.e., they may not maximize the aggregate utility. The algorithm iteratively approaches a price vector $(p_l^*,\ l \in L)$ that aligns individual and global optimality such that $(x_s(p^{*s}), s \in S)$ indeed maximizes the aggregate utility.

In equilibrium, sources that share the same links do not necessarily equally share the available capacity. Rather their shares reflect how they value the resources as expressed by their utility function and how their usage inflect cost on other users.

Primal problem

Our objective is to choose source rates $x = (x_s, s \in S)$ so as to:

$$P: \max_{x_s \in I_s} \sum_s U_s(x_s) \tag{C.1}$$

$$subject\ to \sum_{s \in S(l)} x_s \leq c_l,\ l = 1, ..., L. \tag{C.2}$$

The constraint says that the aggregate source rate at any link l does not exceed the link capacity. A unique optimal solution exists since the objective function is strictly concave and the feasible solution set is compact (close and convex).

The key to solve problem P is to look at its dual problem.

Dual problem

Define the Lagrangian

$$L(x,p) = \sum_s U_s(x_s) - \sum_l p_l(\sum_{s \in S(l)} x_s - c_l) \tag{C.3}$$

$$= \sum_s (U_s(x_s) - x_s \sum_{l \in L(s)} p_l) + \sum_l p_l c_l \tag{C.4}$$

Notice the first term are separable in x_s, and hence

$$\max_{x_s} \sum_s (U_s(x_s) - x_s \sum_{l \in L(s)} p_l) = \sum_s \max_{x_s} (U_s(x_s) - x_s \sum_{l \in L(s)} p_l)$$

The objective of the dual problem is thus,

$$D(p) = \max_{x_s \in I_s} L(x,p) = \sum_s B_s(p^s) + \sum_l p_l c_l$$

where

$$B_s(p^s) = \max_{x_s \in I_s} (U_s(x_s) - x_s p^s) \tag{C.5}$$

$$p^s = \sum_{l \in L(s)} p_l \tag{C.6}$$

and the dual problem is:

$$D: \quad \min_{p \geq 0} D(p). \tag{C.7}$$

The first term of the dual objective function is decomposed into S separable subproblems C.5-C.6. We can interpret the dual problem as follows. Let p_l be the price per unit bandwidth at link l and p^s the total price per unit bandwidth for all links in the path of s. Hence, $x_s p^s$ represents the bandwidth cost to source s when it transmits at rate x_s. If we view utility $U_s(x_s)$ as the benefit of source s at rate x_s, then $B_s(p^s)$ represents the maximum net benefit s can achieve at the given price p^s.

According to the Duality theorem, the optimum solution of the dual problem ($p^* \geq 0$) corresponds to the optimal solution (x^*) of the primal problem.

The striking characteristics of the dual problem is that, given a price p, individual source s can solve C.5 separately without the need to coordinate with other sources. C.5 can be easily solved by:

$$x_s(p) = \min(\max(U_s'^{-1}(p), m_s), M_s) \tag{C.8}$$

where $U_s'^{-1}$ is the inverse of U_s'.