Mechanism Design and Cost Sharing in Multicast Transmission

By
Deepak Saraswat

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario

September 16, 2002

© Copyright
2002, Deepak Saraswat

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-79758-9

Canada

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis,

Mechanism Design and Cost Sharing in Multicast Transmission

submitted by

Deepak Saraswat

Dr. Frank Dehne
(Director, School of Computer Science)

Dr. Anil Maheshwari
(Thesis Supervisor)

Carleton University
September 16, 2002

Abstract

This thesis focuses on the issue of cost sharing in a multicast transmission where each link of the multicast tree is shared by a number of users. Therefore, multicast transmission cannot be priced according to unicast policy, and solutions to the problem of how to distribute the multicast cost among the set of receivers is required. It is assumed that multicast pricing is dependent on users' willingness to pay for a service, but the presence of selfish users complicates pricing. Selfish users are reluctant to report their true willingness to pay and tends to report the values that maximize their own welfare.

The thesis chooses a novel approach of mechanism design to solve the multicast pricing problem. Strategyproof cost sharing mechanisms can be used to avoid the deception of selfish receivers. The main objective in this thesis is to design distributed and communication efficient pricing algorithms that can implement the marginal cost and Shapley value strategyproof cost sharing mechanisms.

The first part of the thesis considers the pricing problem in the source-rooted multicast transmission, focusing on the pricing of both single and multirate multicast transmissions. The aim is to develop a pricing algorithm that can choose the transmission rate according to fixed selection criteria and user preference. In single rate multicast pricing, three criteria are considered for source rate selection: maximizing social welfare, maximizing the size of the final set of receivers, and maximizing the individual utility of the maximum number of users. The marginal cost solution for the single rate transmission model is shown to be communication efficient while the Shapley value mechanism proves to be difficult to implement. In multirate multicast pricing, the problem of maximizing social welfare is considered and the marginal cost

solution for the multirate multicast is shown to be communication efficient.

The second part of the thesis examines the pricing problem in shared tree multicast, focusing on the pricing of multiparty communication and single source multicast. Distributed and scalable pricing schemes are developed, and these pricing schemes are shown to be communication efficient, and requiring only a constant number of messages traversing each link of the shared multicast tree.

Acknowledgments

I would like to thank Dr Anil Maheshwari, my supervisor, for his able guidance, constructive criticism, useful discussions, and valuable support during the course of this thesis. This thesis has been made considerably more intelligible through his painstaking reviews.

I would like to thank the members of my committee Professors Stefan Dobrev, John Oommen, and Michiel Smid. I would also like to thank my friends David, Peter, and Rama for their loyal friendship and moral support.

Most of all, I would like to thank my family, especially my parents, my brother and sister-in-law, Sanjay and Kanika, my sisters and brothers-in-law, Sangeeta and Dinesh, Babita and Pankaj, Preeti and Dharmendra for their great encouragement throughout my graduate studies at Carleton. Without their support, this thesis would not have been completed.

Contents

1	Introduction	1
1.1	Basic Pricing Schemes	3
1.2	Multicast Pricing	5
1.3	Mechanism Design	6
1.3.1	Game Theory	6
1.3.2	Multicast Service as a Game	7
1.4	Summary of the Thesis	8
1.4.1	Pricing Single Rate Multicast Transmission	9
1.4.2	Pricing Multirate Multicast Transmission	10
1.4.3	Pricing Multiparty Communication	11
1.4.4	Pricing Single Source Multicast	12
2	An Introduction to Mechanism Design	13
2.1	Individual Decision Making	14
2.1.1	Utility Function	15
2.2	Social Choice Theory	15
2.3	Quasi-Linear Environment	18
2.4	Mechanism Design	19
2.4.1	Mechanism Design Problem	20
2.4.2	Mechanism	22
2.4.3	Revelation Principle and Dominant Strategy Implementation .	25
2.5	Vickrey-Groves-Clarkes Mechanism	27

3	Multicast Transmission	29
3.1	Introduction	29
3.2	Routing Scenarios in Multicast	30
3.3	Routing in Multicast Transmission	31
3.3.1	Source Based Routing	32
3.3.2	Shared Tree Routing	33
3.3.2.1	Steiner Tree	33
3.3.2.2	Core Based Tree	33
3.4	Internet Routing Protocols	35
3.5	Cost Sharing in Multicast Transmission	35
3.5.1	Basic Definitions	37
3.5.2	Marginal Cost Pricing Mechanism	39
3.5.3	Budget Balance (Shapley Value Mechanism)	40
3.6	Related Work	42
I	Multicast Pricing of Source Rooted Tree	45
4	Single Rate Multicast	46
4.1	Introduction	46
4.2	The Model	47
4.3	Marginal Cost Mechanism for Multicast Transmission	48
4.3.1	Maximizing Social Welfare	51
4.3.1.1	An Algorithm	52
4.3.2	Maximizing the Size of the Final Set of Receivers	61
4.3.2.1	An Algorithm	61
4.3.3	Maximizing Individual Utility of the Maximum Number of Users	66
4.3.3.1	An Algorithm	66
4.4	Shapley Value Mechanism	70
4.4.1	Algorithm	71
4.5	Summary	74

5	Multirate Multicast	76
5.1	Introduction	76
5.2	Multirate Multicast Transmission	78
5.2.1	Network Model	79
5.2.2	Problem Description	80
5.3	Marginal Cost Solution	82
5.4	Summary	87
II	Multicast Pricing of Shared Tree Multicast	89
6	Multiparty Communication	90
6.1	Introduction	90
6.2	Distribution Model	91
6.2.1	Unicast Receive and Multicast Send	92
6.3	Network Model	93
6.4	Pricing Multiparty Communication	94
6.4.1	Charging Senders	94
6.4.2	Marginal Cost Mechanism for Charging Receivers	96
6.5	Summary	99
7	Single Source Multicast Using Shared Trees	100
7.1	Unicast Receive and Multicast Send	100
7.2	Core Based Multicast	102
7.2.1	Network Model	102
7.2.2	Pricing in Core Based Multicast	102
7.2.2.1	Charging Sender/Receivers	104
7.3	Summary	106
8	Conclusions	108
8.1	Future Work	110
	Bibliography	112

List of Tables

2.1	Preferences of the three players.	16
2.2	Preferences of the three players and preferences due to social welfare considerations.	16
4.1	Preferences of children and user located at node n	53
4.2	An example of welfare computation at node n	54
4.3	Cost computation of $User_1$	57
4.4	Welfare and number of users vector from all the children of node n for each available rate.	64
4.5	An example for the computation of number of users vector $Nuser(n)$	64
4.6	Welfare and $Ninusers$ vector from all the child of nodes n for each available rate.	68
4.7	An example for $Ninusers(n)$ computation at node n	68

List of Figures

1.1	Unicast vs. multicast transmission.	2
1.2	Multicast network model and pricing problems solved in this thesis. .	10
3.1	A source-based multicast routing.	31
3.2	A Steiner tree-based multicast routing.	32
3.3	A core-based multicast routing.	34
4.1	An example of a source-rooted multicast tree in the Internet.	49
4.2	Computing welfare during the bottom-up traversal of the tree.	54
4.3	Computing cost share and allocation during the top-down traversal of the tree.	58
4.4	An example of marginal cost service allocation at the rate j_s	60
4.5	A Shapley cost sharing algorithm.	74
5.1	A single rate vs. multirate multicast routing.	77
5.2	An example of a source-based multirate multicast tree.	81
6.1	An example of the unicast receive and the multicast send scheme. . .	92
7.1	An example of a unicast receive and multicast send scheme in single- source multicast.	101
7.2	An example of core-based routing in single-source multicast.	103
7.3	A core-based multicast tree in single-source multicast.	104

Chapter 1

Introduction

In recent years, the Internet has seen an explosive growth in multimedia traffic, and it is very likely that in the future this will constitute the largest percentage of data transmitted over the Internet. This data is usually intended for a group of receivers. *Broadcast*, *multicast*, or *unicast* can all be used to deliver data in the Internet, but they are not equally suitable for transmitting data to groups of receivers.

In general, broadcast is not suitable for sending the data to a selective set of users; rather it is intended for delivering the data to all the receivers. Multicast or unicast are the preferred mechanisms if we want to send the same data to multiple users. In unicast each packet that is transmitted from the source is intended for a single receiver. If unicast is used to send similar data to a set of receivers, the source needs to send a separate copy of the data to all receivers. This results in many identical copies of the data traversing over shared links¹ of the network. Therefore, unicast is not suitable for group communication as it results in sub-optimal utilization of bandwidth.

In contrast, multicast provides an efficient way of transmitting the data from a sender to a group of receivers. It is basically designed to minimize the amount of identical traffic when the same information is transmitted to multiple receivers. Many

¹A link shared by multiple users for receiving data from a source node.

Figure 1.1: Unicast vs. multicast transmission.

real time Internet applications, such as teleconferencing, software update distribution and collaborative interaction, require communication in groups, and multicast is the only suitable mode of delivery for these applications.

In multicast, a single packet is sent on each link. Multicast creates a tree connecting the source to all of the receivers. When a packet reaches a branch point in the tree, a router duplicates the packet and sends a copy over each of the downstream links [14]. In this way a data packet from the source can reach multiple receivers without duplicate copies of the packet having to be sent over any link. Figure 1.2 shows the advantages of multicast over unicast transmission. In unicast, the number of messages sent to any link depends on the number of receivers located downstream. In multicast a link contains only a single message; there is no direct relation between

the number of messages and the users located downstream, thus providing a lot of bandwidth savings over unicast.

A lot of work in network routing [12, 74] has been done on multicast. A number of multicast architectures have been proposed for the scalability of multicast trees [13, 15, 73]. In this thesis, we explore the issue of *cost allocation in multicast transmission*, that is, how receivers can be charged for receiving a multicast transmission in a network such as the Internet. Hence the next section presents an overview of the basic pricing schemes that are used to price Internet usage today.

1.1 Basic Pricing Schemes

From the perspective of users, the Internet can be viewed as a pathway providing a number of communication services. From the point of view of pricing, these communication services can be seen as traditional economic goods, and therefore well established ideas of economic theory can be applied to their pricing. However, the basic idea used for pricing Internet communication services is to price the scarce resources, i.e., bandwidth, and not the information as goods. Three types of pricing schemes are primarily used for pricing Internet services [20].

1. **Flat-rate pricing:** In flat rate or linear pricing a fixed fee is charged irrespective of the actual bandwidth usage or service quality. The fee does not vary with bandwidth utilization or with the level of congestion. This pricing scheme is simple and popular because of its low accounting and billing overheads. However, it ignores the issue of non-uniformity in Internet traffic and is not adequate for differentiated services. For example, flat-rate charging for real time multimedia applications is similar to that of non-real time application such as email.
2. **Capacity-based pricing:** Currently, most Internet users are not billed for each bit sent; they pay a fee to connect and are charged accordingly for different types

of connections. For example, a user may pay for a *T1* capacity link regardless of how many bits they receive or send. The decision as to the speed of the connection is based upon the expected usage, not on the actual usage of a network link.

While most users consider paying a flat rate for their Internet connections, most actually pay a flat rate for a particular type of connection based on the maximum bandwidth of the connection. This is capacity-based pricing. At a number of places the term “flat-rate” pricing can be seen as describing what is, in fact, capacity-based pricing.

3. **Usage-sensitive pricing:** In a usage-based pricing scheme, users pay a portion of their Internet bill for the connection and possibly varying portions for each bit sent and/or received. Therefore, the fee is charged to the actual users of the service, whereas in flat-rate or capacity-based pricing everyone has to pay irrespective of their usage.

This scheme is not good for general usage, primarily due to high accounting and billing costs. Moreover, it is assumed that the marginal cost of sending or receiving the data in absence of congestion² is zero. Therefore, it is a good idea to use usage-based pricing for applications that require a large bandwidth for a small time period, for example, video-conferencing, live concerts, etc.

Although flat-rate and capacity-based pricing protect the actual users from large bills, they fail to optimize the charges. Therefore, usage-based pricing has attracted a number of researchers. Most of the research work in this area has been focused on the following two usage-sensitive pricing schemes:

- **The smart market method:** Mackie and Varian [47] initially proposed a *smart market* method [32] for Internet pricing. In this scheme, packets passing through the network are dynamically priced. Each packet sent through the Internet includes a “*willingness-to-pay*” value. When congestion occurs, a Vickrey auction is held to determine the market price of

²Congestion occurs in the network when bandwidth demand is greater than supply. In other words, bandwidth becomes a scarce resource.

each packet. Users with high willingness to pay for their packets get a high priority. However, there is a major drawback with this method in that it allows the users to report false values about their willingness-to-pay. Such deceptions can be avoided by offering incentives to users who bid true values.

- **The rate of transfer method:** This scheme [32] targets both demand and supply of the Internet market. It uses effective bandwidth as a measure of how highly a source utilizes a link. If the resource usage is above a certain level, then the overloaded packets are marked and sent back to the users. These packets act as a signal to the users that the price of the scarce resource is going up and they can adjust their demand or the price accordingly. Therefore, on the supply side, prices compensate suppliers, and on the demand side, prices allocate a resource efficiently.

There are a few other pricing schemes that have been used for the Internet, for example, *paris metro pricing* [55], *zone based pricing* [7], *edge pricing* [71], and *priority pricing* [9], but these schemes will not be discussed here.

1.2 Multicast Pricing

Multicast is a point-to-multipoint communication in which a multicast tree is formed and the data is replicated only at the branching node. Each link of the multicast tree is shared by the downstream users. From the perspective of basic economic theory, if a resource (bandwidth) is shared by multiple receivers (users), then its cost should also be shared appropriately among those receivers. Hence, from a pricing perspective, multicast transmission needs to be treated differently.

Now, the question is how to distribute the cost among the set of receivers. We consider that multicast pricing is based on users' *willingness to pay* for the service. If each receiver i shows their willingness to pay u_i , also called receiver's utility, by bidding for a service, let $x_i > 0$ and q_i denote the amount charged from receiver i and

whether user i gets the service or not, respectively ($q_i = 1$ if the receiver receives the service and $q_i = 0$ otherwise) then the *individual welfare* achieved by any user after receiving the service is given by $w_i = u_i q_i - x_i$.

Since the network does not know *a priori* the receivers' individual utility, it has to depend on the receivers to report these values. We consider that Internet users, either as individuals or members of different organizations can act selfishly to maximize their own satisfaction or welfare. Therefore, we cannot simply believe that each receiver on the Internet will faithfully follow the protocol and report the *truth value* of their u_i . Thus, it is important to design a truthful mechanism for achieving a socially desirable allocation of multicast service on the Internet. We have used a novel scheme based on *mechanism design* to accomplish this task.

1.3 Mechanism Design

Mechanism design is a sub-field of game theory and microeconomics that is used to solve microeconomics problems and other problems that can be classified as having the theoretical properties of a good game. Papadimitriou [57] describe mechanism design in terms of game theory as follows: *Given desired goals, design a game in such a clever way that individual players, motivated solely by self-interest, end up by achieving the designer's goal.*

1.3.1 Game Theory

Game theory was founded by John Von Neumann and Oskar Morgenstern. It is used as a mathematical framework for the analysis of the behavior of rational agents. An introduction to this field can be found in Fudenberg [23] and Osborne [56]

On the basis of agent behavior, games are usually classified into two types: *co-operative* and *non-cooperative*. In a cooperative game, agents communicate information about their strategies or actions. A non-cooperative game, however, requires

agents to work individually without information from others. In either type of game, the goal of each agent is to optimize their utility. Non-cooperative games have the advantage of requiring relatively low player-to-player communications overhead.

Game theory has been applied by a number of researchers to the study of a variety of computer science problems including multicast transmission. Hsiao and Lazar [36] and Korilis and Lazar [42] have used game theory to study flow control, whereas Feigenbaum et al. [19], Economides et al. [17] and Lazar et al. [44] have used it to study pricing, routing and bandwidth allocation in the network respectively. In all these applications both cooperative and non-cooperative agents have been considered. For example, the games studied by Feigenbaum et al. [19] and Lazar et al. [44] are examples of non-cooperative games. Feigenbaum et al. [19] in particular have applied non-cooperative game theory to the pricing of multicast transmission.

1.3.2 Multicast Service as a Game

In this section, we discuss game theoretic properties of the multicast transmission or, more generally, of a network like the Internet. We also describe why multicast transmission service could be modelled as a game.

On the Internet, users are distributed throughout the network. Therefore, it is very difficult to obtain any knowledge about other players of the game (i.e., other service users). Players (users) do not know who the other players are, or even how many they are. They are not aware of the other players' actions and how their own payoff depends on the actions of other players. Moreover, the players have no or very little knowledge of the underlying network topology and how it affects the payoff of other players. The only information available to the players is their own actions and the resulting payoffs. This lack of information is the main reason why multicast service could be modelled as a non-cooperative game³.

As mentioned earlier, mechanism design is good for solving problems with game

³See [21] for games arising on the Internet.

theoretic properties. Therefore, it can be effectively applied to the issue of cost distribution in a multicast transmission. In this work, we refer to such a mechanism as a *cost sharing mechanism*. We use a strategyproof cost sharing mechanism to avoid the deception of selfish users. There are two strategyproof cost sharing mechanisms: *Marginal cost (MC)* and *Shapley value* (see [19]). However, implementing these mechanisms in multicast transmission is challenging, as the users are distributed throughout the network.

1.4 Summary of the Thesis

In this thesis, the main focus is to develop *distributed* and *communication efficient* cost sharing algorithms for implementing the strategyproof cost sharing mechanism in multicast transmission. The need for distributed algorithms arises due to the distributed nature of user allocation (users are placed throughout the network). However, the practability of any pricing algorithm depends upon the communication efficiency. If the algorithm requires too many messages then the approach has no advantage over the other approaches, e.g., centralized.

The thesis is divided into two parts. The first part is dedicated to pricing problems related to source rooted multicast transmission. In the second part, pricing schemes for shared tree multicast transmission are presented.

Part 1: Multicast Pricing of Source Rooted Tree

In source rooted multicast transmission, a multicast tree rooted at the source node is created. This multicast tree connects all the receivers to the source node. The source uses this tree to multicast the data. The following pricing problems in the source rooted multicast transmission are considered.

1.4.1 Pricing Single Rate Multicast Transmission

In Fiegenbaum et al. [19], the source multicasts the data to all the receivers at the same rate. This rate selection is governed by the source. However, it is not a good strategy to transmit the data arbitrarily at any rate. For example, if most of the users are willing to receive the muticast at rate y_1 , then transmitting the data at rate $y_2 > y_1$ is not a good solution because of high packet loss, overflow etc. Therefore, proper selection of the source rate is important for the successful multicast transmission.

In the single rate transmission model, the source is considered to have a finite set of transmission rates available and is required to select the transmission rate on the basis of user preference and charge the users for receiving the multicast transmission. The single rate pricing problem can be stated as follows:

Problem 1 : *Design a distributed and communication efficient pricing algorithm that can charge the users for receiving the multicast transmission and select the source rate from a finite set of possible rates on the basis of user preference and three distinct criteria:*

- (a) *maximizing social welfare,*
- (b) *maximizing the size of the final set of receivers, and*
- (c) *maximizing individual utility of the maximum number of users.*

In maximizing social welfare, the source rate with maximum social welfare⁴ is selected. However, in maximizing the size of the final receiver set, the source rate is selected such that the source can facilitate the multicast transmission to the maximum number of users. In maximizing individual utility of the maximum number of users, the source transmits the data at the rate preferred by most of the users⁵.

Chapter 4 shows that the marginal cost solution to the aforementioned pricing problems is communication efficient and requires only two messages per link of the

⁴Social welfare is defined as the difference of the sum of utilities of the receivers minus the total cost of transmission.

⁵Preferences are decided by the corresponding utility value.

Figure 1.2: Multicast network model and pricing problems solved in this thesis.

multicast tree. We also show that it is difficult to implement the Shapley value pricing mechanism for the single rate transmission model.

1.4.2 Pricing Multirate Multicast Transmission

Due to the heterogeneity of the Internet, the pricing issue cannot be limited to single rate transmission, but must include pricing in multirate multicast transmission. A layered multirate multicast transmission is considered, in which all the data is encoded in a group of layers $\{L_1, \dots, L_m\}$. Any user who wants to receive the data at an aggregate rate of L_j can join the layers from L_1 to L_j . It should be noted that in multirate multicast, receivers are free to receive the transmission at different rates⁶. The multirate multicast pricing problem can be specified as follows:

⁶The fact that users have different and discrete utilities for receiving the transmission at different rates is considered.

Problem 2 : *Design a distributed and communication efficient pricing algorithm that can charge the users for receiving the multicast transmission and results in maximum social welfare.*

Chapter 5 presents the marginal cost solution to the multirate multicast pricing problem. The marginal cost pricing algorithms are communication efficient and requires only two messages per link. The Shapley value mechanism is not considered due to its low communication efficiency.

Part 2: Pricing of Shared Tree Multicast

In shared tree multicast, a multicast tree rooted at the fixed node, called a core, is formed. This shared tree connects all the users of the group and can be used by any source to multicast the data to the set of users in the group.

1.4.3 Pricing Multiparty Communication

In this scenario, the problem of charging for multiparty communication, for example, video-conferencing, distributed games etc, is studied. In multiparty communication (video-conferencing), a user can be either a pure sender, pure receiver or both sender and receiver. A fixed node (core) acts as a conferencing server, mixing the data received from different senders in a predefined policy and multicasting it to a set of receivers. The links of the shared multicast tree are considered to be of infinite capacity. Therefore, senders and core nodes are free to transmit the data at the maximum rate. The multiparty pricing problem is defined as follows:

Problem 3 : *Design a distributed and communication efficient pricing scheme for charging the different members (pure sender, pure receiver or both) of the multiparty communication.*

Chapter 6 presents the distributed pricing scheme that implements the marginal cost sharing mechanism and also calculates the status of each user⁷ according to their

⁷That is, the group member is acting as a pure sender, pure receiver or both.

willingness to pay for receiving the transmission.

1.4.4 Pricing Single Source Multicast

In single source multicast, the pricing of unicast send and multicast receive, and core based routing are considered. A multicast tree rooted at the core node is built and each source multicasts the data on the shared tree. The single source multicast pricing problem is defined as follows:

Problem 4 *Design a distributed and communication efficient pricing scheme that can charge all the members (source and receivers) of the single source multicast.*

Chapter 7 describes a distributed and communication efficient pricing scheme for single source multicast. This pricing scheme implements the marginal cost mechanism and requires two messages per link of the shared multicast tree. Moreover, both the source and receivers have to pay for using the resources.

Chapter 2

An Introduction to Mechanism Design

The previous chapter shows that multicast transmission possesses the theoretical properties of a game and that mechanism design can be used to develop a cost sharing mechanism for multicast transmission. This chapter further explores the field of mechanism design.

The structure of this chapter is as follows. Section 2.1 studies the theory of *individual decision making* and describes the concept of *preference relation* and *utility function*. Section 2.2 reviews *social choice theory* and introduces the concept of *social welfare functional*. *Arrow's impossibility theorem* illustrates the difficulties of accomplishing social preferences.

Sections 2.4 and 2.5 are devoted to a detailed study of the field of mechanism design. A few application areas of mechanism design are mentioned and the concepts of *mechanism implementation*, *equilibrium outcome*, *direct revelation*, *dominant strategies* and *truthful implementation* are described. Finally, the celebrated *Vickrey-Grooves and Clarkes* mechanism is discussed.

2.1 Individual Decision Making

Each of us faces a variety of decision making problems all the time. For example, one may have to decide how many more hours he will work today. The basic issue addressed here is, How does an individual make decisions in general?

Theoretically, the starting point of any individual's decision making could be a set of possible mutually exclusive alternatives from which s/he must choose. If the set of all alternatives¹ are denoted by X , an individual's (agent's) choice behavior can be modelled using a *preference relation*. A preference relation can be defined as follows:

Definition 1 (Preference Relation) [46]: *Suppose there are n agents and a set of possible choices $X = \{x_1, x_2, \dots, x_m\}$. The preference relation $\succsim_i$ over X is defined as the ordering of set X according to the preference of the agent i .*

Note that the preference relation $\succsim$ is a binary relation on the set of alternatives X . The expression $x \succsim y$ means that x is at least as good as y . Most individual preference relations are assumed to be *rational*. Rationality of the preference relation can be described as follows:

Definition 2 (Rational Relation) [46]: *A preference relation is rational if it possesses the following two properties:*

1. Completeness: *For all $x, y \in X$, $x \succsim y$ or $y \succsim x$.*
2. Transitivity: *For all $x, y, z \in X$, if $x \succsim y$ and $y \succsim z$, means $x \succsim z$.*

Completeness means that an individual has a well defined preference between any two alternatives; transitivity means that one can possibly find a preference between x and z if one knows the preference between x and y , and y and z .

¹For example, in the work example, X is the set of all the hours that are possibly available.

2.1.1 Utility Function

In economics, preference relations are described by means of *utility functions*. A utility function $u(x)$ assigns a numerical value to each element of the set of alternatives in X . It basically assigns rank to each element of X in accordance with the individual's preference. This can be more precisely described by the following definition.

Definition 3 [46]: *Given a preference relation $\succsim_i$ on a set X , utility function $u_i(\cdot)$ of an agent i is any function $u_i : X \rightarrow R$, such that for all $x, y \in X$,*

$$x \succsim y, \text{ if and only if } u(x) \geq u(y).$$

Intuitively we know there can be more than one utility function representing the same preference relation. For example, any strictly increasing function $f : R \rightarrow R$ gives a new utility function, $v(x) = f(u(x))$. A preference relation $\succsim$ can be represented by a utility function only if it is rational, i.e., if there is a utility function that represents preference relation $\succsim$, then $\succsim$ must be complete and transitive.

2.2 Social Choice Theory

Social choice theory deals with the aggregation of individual preferences into social preferences, as in voting. This social preference order is used to make a social decision. The aggregate social preference order is computed by a *social welfare functional (SWF)* that is defined as follows:

Definition 4 (Social Welfare Functional): *A social welfare functional is a function $f(\succsim_1, \dots, \succsim_n)$ that takes individual preferences (utility) as input and outputs a global preference.*

Here, an example is given to get a clearer idea of the social welfare functional. A society consisting of three players, $Player_1$, $Player_2$, $Player_3$ is considered. Table

<i>Preference</i>	<i>Player₁</i>	<i>Player₂</i>	<i>Player₃</i>
<i>First</i>	x_1	x_1	x_3
<i>Second</i>	x_3	x_2	x_2
<i>Third</i>	x_2	x_3	x_1

Table 2.1: Preferences of the three players.

2.1 shows the preferences of all players on the set of alternatives x_1, x_2, x_3 . In the table, for example, *Player₁* prefers x_1 , followed by x_3 , followed by x_2 . A social welfare functional takes these preferences and creates an ordering for the entire society.

One possible ordering, shown in Table 2.2, is that society prefers x_1 to x_2 and x_3 to x_2 , since two out of three agents have these preferences. Therefore, the social preference for the society is given by $x_1 \succsim x_3 \succsim x_2$, while the preferences for each agent stay the same.

<i>Preference</i>	<i>Player₁</i>	<i>Player₂</i>	<i>Player₃</i>	<i>Society</i>
<i>First</i>	x_1	x_1	x_3	x_1
<i>Second</i>	x_3	x_2	x_2	x_3
<i>Third</i>	x_2	x_3	x_1	x_2

Table 2.2: Preferences of the three players and preferences due to social welfare considerations.

Dictatorship, *democracy*, *Borda's rule*, etc., are a few examples [46] of social welfare functionals. Social welfare functions can be defined infinitely for any set of n -agent preference patterns. However, the kind of SWF can be restricted by listing a few properties that the SWF should satisfy. A few properties that are mentioned by Sandholm [66] and appear reasonable for any definition of SWF are:

1. **Complete:** The social preference ordering should be defined for all vectors of inputs.
2. **Transitive:** The social preference ordering that results from the social welfare functional should be transitive.
3. **Pareto Optimal:** Various criteria can be used to check whether the social

welfare functional is producing an optimal outcome or not. One such criterion is the property of *pareto optimal* (or *pareto efficient*). Pareto optimality states that it is impossible to make the allocation of an agent “better” without making the allocation for someone else “worse.” This can be formally described as follows:

Definition 5 (Pareto Optimal): A feasible allocation vector $Y = \{y_1, \dots, y_I\}$ is said to be *pareto optimal* (or *pareto efficient*) if there is no other feasible allocation $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_I\}$ such that

$$u(\hat{y}_i) \geq u(y_i) \quad \text{for all or some } i = \{1, \dots, I\}.$$

4. **Independence of Irrelevant Alternatives:** This property of SWF says that the socially preferred allocation is invariant under a change in individual *utility functions* that leave individual preferences unaffected.
5. **Non-Dictatorial:** A SWF f is said to be non-dictatorial if there exists no individual i with utility function u_i such that

$$u_i(x_i) \geq u_i(y_i) \Leftrightarrow f(x) \geq f(y) \quad \forall x, y.$$

Here x and y are the social choice vectors over the set of choice X . In other words, a function in which one agent alone determines the outcomes that are considered desirable is called *dictatorial*.

Now, we turn our attention to the *Arrow Impossibility theorem*. It is one of the most important results of social choice theory.

Theorem 1 (Arrow Impossibility Theorem) [46]: For $|X| \geq 3$, there does not exist any social welfare functional that satisfies all of the following properties:

1. It is independent from irrelevant alternatives.
2. It respects unanimity.
3. It is non-dictatorial.

2.3 Quasi-Linear Environment

The most commonly studied class of environments in which agents have quasi-linear preferences is called a quasi-linear environment. In a quasi-linear environment, it is assumed that there is a way for the mechanism to transfer money among the agents in order to induce them to provide accurate information. The quasi-linear environment can be formally defined as follows:

Definition 6 [46]: *In a quasi-linear environment the following are assumed:*

1. *There is a finite set K called the project choice.*
2. *X is the set of outcomes $x = \{(k, t_1 \dots t_n) : k \in K, t_i \in R, \text{ and } \sum_i t_i \leq 0\}$ where t_i is a transfer of money to agent i .*
3. *Each agent's utility function takes a quasi-linear form, i.e., for each i*

$$u_i(x, t_i) = v_i(k, t_i) + t_i.$$

where $v_i(k, t_i)$ is a real valued function called the agent i 's valuation of project choice k .

2.4 Mechanism Design

In the previous sections the theory of individual decision making and social choice is discussed. This section looks at mechanism design and describes situations where it can be effectively used.

The field of mechanism design is broad and has many important applications throughout the fields of economics and computer science. A significant amount of research in computer science is based on designing protocols and algorithms for networks such as the Internet. The designers of such algorithms or protocols make an implicit assumption that the users will act as instructed. However, this assumption does not remain valid for long due to the presence of selfish users who have different and conflicting goals².

Therefore, an algorithm or protocol that is resistant to the selfish behavior of the users (computers) is needed. If the development of such an algorithm is viewed as a game, then the algorithm designer (game designer) has to consider that he/she will not know individual preferences with certainty because this information is typically private, i.e., it is known only by the individuals themselves. For example, in the case of multicast transmission, only the receivers of the multicast transmission know the private information about their willingness to pay for receiving the service. Similarly in a real life situation, only taxpayers possess the private information about their entire earnings. Therefore, the economic allocations that are desirable or that can be obtained are dependent on the realization of the types (utility) of agents. Also it may not be in the self-interest of the agents to reveal their true types without first having proper motivation or incentives. In such situations, mechanism design is an approach that can be used effectively to allocate resources. The following are a few examples of situations that arise in the Internet environment where mechanism design can be applied to cope with the selfish behavior of users.

1. **Multicast Pricing:** In multicast pricing, the users are charged on the basis

²By selfish we mean users who act to maximize their own satisfaction and do not care whether any of the other players satisfy their goals or not.

of their willingness to pay for receiving the multicast transmission. However, users belonging to different companies or organizations can also act selfishly. Therefore, an algorithm that can provide some motivation or incentive to the users to report the true value of their willingness to pay is needed.

2. **Routing:** In the Internet, routers route the information voluntarily³ with best effort service. However, best effort service does not provide guaranteed quality of service and is not good for real time applications. Most of the real time applications prefer to reserve resources (bandwidth) in advance and require guaranteed quality of service. Since unselfish behavior of the routers may not be taken for granted, one must design an algorithm or routing protocol that specifically takes the routers' self-interest into account.
3. **Resource Allocation:** In distributed computing environments available over the Internet, resources belong to different persons or organizations who may not allow other parties to freely use them. Hence, the algorithm required to solve this problem should provide some motivation for these resource owners to allow use of their computing resources by other users.

Other examples of application areas that fall into this category are electronic trade and distributed games.

2.4.1 Mechanism Design Problem

This section describes the *mechanism design problem*, limiting the discussion to the quasi-linear environment (see Section 2.3). Any mechanism design problem consists of two components: the algorithmic output specification and the agents' preferences (formally given as agent utility over the set of possible outputs).

Definition 7 (Mechanism Design Problem) [53]: *A Mechanism design problem is described by the following:*

³Primarily due to low marginal cost of forwarding the data.

1. A set of $I = \{1, \dots, n\}$ agents that must make a collective (social) choice over some finite set O of possible outcomes.
2. Preferences of each agent i are given by a real valued function $v^i(o \in O)$ called its valuation or type t^i . It is privately known to agent i , and determines its benefit from each possible output o in terms of some common currency. V^i is the set of possible valuation functions.
3. If the mechanism hands on p^i units of currency to the agent i , then its utility u^i equals $v^i(o) + p^i$. This is what the agent aims to optimize.
4. The goal of the mechanism is to select an output $o \in O$ that maximizes the total welfare $g(v, o) = \sum_i v^i(o)$.

Here, it should be noted that the utility functions $u(\cdot)$ are assumed to be common knowledge among the agents, but the specific value of its type or valuation is known only to the individual agent. The goal of any agent is to optimize their utility for the given type. For further clarification a simple example within the framework of mechanism design is formulated.

Example (A Simple Routing Problem): Suppose there are two computers and one wants to send a message to the other. These two computers are connected through $l > 1$ parallel links. Each of these l links is owned by a different agent. These agents charge c_e amount of money for sending a message along the link. The cost c_e is privately known to its owner.

Within the framework of mechanism design, the above stated problem can be described as follows.

1. The set of feasible output contains all the links from one computer to the other.
2. The valuation of each selected link for the output o is equal to $-c_e$, otherwise zero. This valuation is privately known only to the owner of the link.

3. The utility of each agent is equal to the sum of the payment received from the chosen output and its valuation. Payment of any unselected link is equal to zero.
4. The goal of the mechanism is to select a link with the minimal cost so can total welfare can be minimized.

Naive Solution: Consider a traditional algorithmic approach in which the algorithm first requests each agent to report their cost. It then compares the cost and chooses the link with the minimal declared cost and routes the message along that link. This approach sounds very good from an algorithmic perspective, but is likely to fail with the rational agent. Rational agents will try to report their costs as higher than their actual costs [63]. $\square$

2.4.2 Mechanism

The question is now how to solve the above stated problem in the presence of selfish agents. Intuitively we know that the solution requires new protocols that solve the given problem even in the presence of selfish agents. These protocols are referred to as the *mechanism*.

The mechanism basically assures that the required outcome occurs when the agents choose their *strategies* so as to maximize their own welfare. In non-cooperative game theory, *strategies* are the primitive moves from which a player can choose. In other words, a player's strategy is the action or plan of action that a player chooses out of a given set of strategies. At an auction, for example, the strategy of a player describes the way this player bids. The mechanism can be formally described as follows:

Definition 8 (Mechanism) [53]: A mechanism $m = (k, p)$ is described as follows:

1. The mechanism defines for each agent i , a family of strategies S^i . An Agent can choose to perform any $s^i \in S^i$.

2. The first thing a mechanism must provide is an output function $k : s^1 \times s^2 \times \dots \times s^n \rightarrow O$. Each agent i chooses their own strategy from the strategy set S^i . The mechanism then sets the outcome to $k(s^1, \dots, s^n) \in O$.
3. The second thing a mechanism must provide is a payment function $p(S) = (p^1(S), \dots, p^n(S))$. Here $p^i(S)$ corresponds to the payment handed by the mechanism to the i^{th} agent.

Mechanism for Solving a Simple Routing Problem: A mechanism for solving a simple routing problem requires that each link e declares its cost d_e for sending the message from one computer to the other. Note that these declared values may be falsified. The output function k takes the vector of declared valuation and chooses the link with the minimal declared cost. The payment function p pays the owner of this link the minimum among the declared costs of the other agents, i.e., $p_e = \min_{e \neq e} d_e$. This mechanism is said to be a second-priced sealed bid auction mechanism. $\square$

Subsection 2.4.3 shows that the second-priced sealed bid mechanism is truthful; i.e., for each agent the best strategy is to reveal their actual cost.

Notation: The tuple $(s^1, \dots, s^{i-1}, s^{i+1}, \dots, s^n)$ is denoted by (s^{-i}) , and (s^i, s^{-i}) is used to denote the tuple $(s^1, \dots, s^n)$.

A strategy of agent i depends on the information available to agent i about the other agents. With regard to the information that agents possess about each other, i.e., *complete information* or *incomplete information*, two types of environment are possible.

In the complete information environment, each agent knows his own type as well as the types of all the other agents. However, in the case of incomplete information, individual agents have no information about the other agents. A strategy for agent i in the game of incomplete information is a mapping $s^i : t^i \rightarrow S^i$; while in the complete information case, a strategy for agent i is a mapping $s^i : t^1 \times \dots \times t^n \rightarrow S^i$. This thesis is restricted to the game of incomplete information.

A mechanism designer's aim is to compute the *socially desirable outcome* for all

possible preference profiles. This is stated more formally in the next definition.

Definition 9 [46]: *A mechanism $m = (k, p)$ implements a social choice function $f(\cdot)$ if there exists an equilibrium strategy profile S^* of the game induced by m such that $k(S^*(T)) = f(T)$ for all $T = (t^1, \dots, t^n)$.*

The outcome for the equilibrium strategy profile is called an *equilibrium outcome*. However, there is no single equilibrium concept that is universally applicable as the solution concept. The two main equilibrium concepts that are applicable as a solution are the *dominant strategy equilibrium* and the *Bayesian Nash equilibrium*.

Definition 10 (Dominant Strategy Equilibrium) [46]: *A strategy profile $S^* = \{s^{1*}, \dots, s^{n*}\}$ is a dominant strategy equilibrium of a mechanism m , if for all i and all $t^i \in T^i$,*

$$u_i(k(s^{i*}(t^i), s^{-i}), t^i) \geq u_i(k(\hat{s}^i, s^{-i}), t^i)$$

for all $\hat{s}^i \in S^i$ and all $s^{-i} \in S^{-i}$.

In other words, a strategy profile is in a dominant strategy equilibrium if no matter what the other agents do, no agent can perform any other action that can increase their utility.

Definition 11 (Bayesian Nash Equilibrium) [46]: *The Nash equilibrium of a mechanism m is a strategy profile $S^*(\cdot)$ such that for all agent i and all $t^i \in T^i$,*

$$u_i(k(s^{i*}(t^i), s^{-i*}(t^{-i})), t^i) \geq u_i(k(\hat{s}^i, s^{-i*}(t^{-i})), t^i).$$

for all $\hat{s}^i \in S^i$.

In other words, a strategy profile is a Nash equilibrium if no agent can unilaterally deviate from the prescribed action and increase its utility.

Definition 12 [46]: *An equilibrium outcome produced by any mechanism m , for some type of profile T , is the outcome produced by an equilibrium strategy profile S^* of m .*

In this thesis, only the implementation of **dominant strategy equilibrium** is considered. The mechanism is said to be polynomial time computable if the output and payment functions are computable in polynomial time [53].

2.4.3 Revelation Principle and Dominant Strategy Implementation

The Revelation principle is one of the main concepts used in game theory and economics. It is one of the simplest types of mechanisms in which agents are asked to simply report their types. A revelation mechanism computes the output according to the type declaration of the agents. It can be formally described as follows:

Definition 13 (Revelation Mechanism): *A mechanism is said to be a direct revelation mechanism if for all agents $i \in I$ the agents strategy space (S^i) is the same as its type space T^i*

$$S^i = T^i.$$

In a direct revelation mechanism the agents' strategy is to reveal their type. More specifically, the focus is on the mechanisms in which *truth telling is a dominant strategy for each agent*. This fact motivates the notion of *truthful implementation* in dominant strategy equilibrium.

Definition 14 (Truthful Mechanism): *A direct revelation mechanism is said to be truthful if truth telling is a dominant strategy for every participating agent $i \in I$, i.e.*

$$u_i(s^{i*}, s^{-i}) \geq u_i(s^i, s^{-i}). \quad (2.1)$$

In direct revelation $s^{i*} = t^i$ for all $t^i \in T^i$ and for all $i \in I$. Therefore, Equation 2.1 can be rewritten as

$$u_i(t^{i*}, t^{-i}) \geq u_i(t^i, t^{-i}). \quad (2.2)$$

According to truthful implementation, truth telling is a dominant strategy for each agent. However, the question is whether a truthful mechanism is found as the solution to each problem or not. This will be discussed under the next proposition.

Proposition 1 (The Revelation Principle for Dominant Strategies): If there exists a mechanism m that implements the given problem with dominant strategies, then there exists a truthful mechanism as well.

Proof: Let $m = (k, p)$ be the mechanism that implements a social choice function $f(\cdot)$ in dominant strategies, then by definition 9 there exists a strategy profile $s^* = (s_1^*, s_2^*, \dots, s_n^*)$ such that output $k(s^*(t)) = f(t)$ for all $t \in T$ and all $t_i \in T_i$, i.e.,

$$u_i(k(s^{i*}(t^i), s^{-i}), t^i) \geq u_i(k(\hat{s}^i, s^{-i}), t^i) \quad (2.3)$$

for all $\hat{s}^i \in S^i$ and $\forall s^{-i} \in S^{-i}$.

Equation 2.3 reveals the fact that strategy profile s^* is a dominant strategy equilibrium of the mechanism m . However, if the agent reports a different type from the true one under the dominant strategy, i.e., for all i and all $t^i \in T^i$

$$u_i(k(s^{i*}(t^i), s^{-i*}(t^{-i})), t^i) \geq u_i(k(s^{i*}(\hat{t}^i), s^{-i*}(t^{-i})), t^i) \quad (2.4)$$

for all $\hat{t}^i \in T^i$.

But it is known from definition 9 that $k(S^*(t)) = f(t)$ for all t . This implies Equation 2.4 can be rewritten as follows:

$$u_i(f(t^i, t^{-i}), t^i) \geq u_i(f(\hat{t}^i, t^{-i}), t^i) \quad (2.5)$$

for all $\hat{t}^i \in T^i$ and $t^i \in T^i$.

Hence, from the above proof, it is clear that a truthful mechanism is always found under the dominant strategy implementation. $\square$

Theorem 2 *The second-priced sealed bid auction mechanism discussed for a simple routing problem in Section 2.4.2 is truthful.*

Proof: In a second-price sealed bid auction, each link e is required to declare its cost d_e for sending the message from one computer to the other. If link e wins the auction, it will get an amount equal to the second lowest valuation, i.e., $p_e = \min_{\hat{e} \neq e} d_{\hat{e}}$. If t_e and d_e are, respectively, the actual and declared valuation of link e , then consider the incentive of link e for telling the truth. If link e declares its valuation to be $d_e \geq t_e$, link e can receive a utility of $p_e - t_e \geq 0$ by truthfully declaring its valuation as t_e . For any other declaration, link e 's resulting utility is either the same or zero. However, link e can receive only a negative utility by making a false claim that gets it the transmission. That is, for $d_e \leq t_e$ the link e can receive a utility of $p_e - t_e \leq 0$. Therefore, it can be said that truth telling is a dominant strategy for a second-price sealed bid auction. $\square$

2.5 Vickrey-Groves-Clarkes Mechanism

This subsection focuses on the *Vickrey-Groves-Clarkes (VGC)* mechanism. The VGC mechanism is said to be incentive compatible, meaning that the rational agents maximize their utility by truthfully revealing their type. In the VGC mechanism, each agent receives a payment for their participation, and this payment is proportional to the added value the agent brings to the system.

Definition 15 (VGC Mechanism) [53]: *A direct revelation mechanism $m = (k, p)$ belongs to the VGC family if:*

1. $k(T)$ maximizes the total welfare according to T .

2. The payment is calculated according to the VGC formula: $p^i(T) = \sum_{j \neq i} v^j(k(T)) + h^i(t^{-i})$.

Theorem 3 [53] *A VGC mechanism is Truthful.*

Proof: This theorem will be proved by contradiction. Assume that there is a VGC mechanism which is not truthful. This means that there exists an agent i of type t^i who uses another strategy that maximizes the agent's utility rather than reporting its true type t^i . This strategy should be dominant, otherwise it contradicts the assumption of the rationality of the agents. If the declared type of the agent under the dominant strategy is d^i , then

$$\begin{aligned} & v^i(k(t^i, d^{-i})) + \sum_{j \neq i} d^j(k(t^i, d^{-i})) + h^i(d^{-i}) \\ & \leq v^i(k(d^i, t^{-i})) + \sum_{j \neq i} d^j(k(d^i, t^{-i})) + h^i(d^{-i}). \end{aligned} \quad (2.6)$$

this is same as writing

$$\sum_{i=1}^n v^i(k(t^i, t^{-i})) \leq \sum_{i=1}^n v^i(k(d^i, t^{-i})). \quad (2.7)$$

This contradicts the fact that a VGC mechanism maximizes the total welfare and therefore completes the proof. $\square$

It is worth noting that VGC is the only generally known method for construction of a truthful mechanism. A weighted version of VGC is also possible [53].

Chapter 3

Multicast Transmission

3.1 Introduction

Multicast transmission plays a prominent role in the Internet. It is a requirement, not an option, for most of the real time applications that require communication in groups such as teleconferencing and audio/video broadcasting. Multicast creates a directed tree connecting the source to all the receivers, and sends only a single packet on each link. When a packet reaches a branching point in the tree, a router duplicates the packet and sends a copy of the packet to each of the downstream links [14]. In this fashion a packet from the source can reach multiple receivers without sending duplicate copies of the packet over any link. Though this sharing of the upstream link provides effective bandwidth usage, it complicates the issue of pricing.

In this chapter, multicast transmission is examined from a theoretical perspective. Section 3.2 presents a few *routing scenarios* for the multicast transmission. In Sections 3.3 and 3.4 the *construction of the multicast tree* and *multicast protocols*¹ are discussed. Section 3.5 looks at multicast pricing, i.e., *cost sharing in a multicast transmission*, and introduces a few basic requirements such as *no positive transfer*, *consumer sovereignty* and *voluntary participation* under subsection 3.5.1. Two classes

¹See [16] for a complete survey of multipoint communication.

of mechanism, *budget balanced* and *efficient*, are presented in subsections 3.5.3 and 3.5.2 respectively. Finally, Section 3.6 provides a brief overview of other related work on multicast pricing.

3.2 Routing Scenarios in Multicast

Routing scenario in multicast depends on applications. These scenarios are classified on the basis of the number of receivers and sources in a multicast application. Usually, all multicast applications fall into three general categories:

1. **One-to-Many:** One-to-Many applications consist of a single sender and multiple simultaneous receivers. These receivers are distributed randomly throughout the network. A few examples of One-to-Many applications are:
 - **Scheduled Audio/Video Distribution:** Applications in this category include movie, presentations, meetings, or any other type of scheduled event whose transmission could benefit a large number of viewers.
 - **File Distribution and Caching:** Applications in this category include web pages, executable binaries, and other file-based updates sent to distributed end-users or replication/caching sites.
2. **Many-to-Many:** In Many-to-Many applications, more than one receiver acts as a sender. These applications are characterized by two-way multicast communications. Each host running a Many-to-Many application may receive data from multiple senders and also sends data to all of them. Many-to-Many applications often present complex coordination and management challenges. Many-to-Many multicast capabilities enable the most unique and powerful applications such as multimedia conferencing that handle multiple data-streams from a number of senders in the group. A few other similar examples of such applications are distance learning, chat groups and multi-player games.

Figure 3.1: A source-based multicast routing.

3. **Many-to-One:** Unlike One-to-Many and Many-to-Many application categories, Many-to-One applications have multiple senders and one (or only a few) receiver(s). A few examples of Many-to-One application are resource discovery, data collection and data polling. In general, these applications have scaling issues such that too many simultaneous senders can potentially overwhelm receiver(s).

3.3 Routing in Multicast Transmission

In this section, the routing scenario in a multicast transmission is described. The aim is to provide an understanding of multicast tree construction by examining two major routing schemes, namely, *source base* and *shared tree* based routing [16].

Figure 3.2: A Steiner tree-based multicast routing.

3.3.1 Source Based Routing

In source based routing, a shortest path tree (SPT) rooted at the source node is constructed. This tree consists of the union of shortest paths between the source and each of the receivers in the multicast group. The SPT does not minimize the total cost of distribution; rather it optimizes the size of the multicast tree for dense receiver distribution and can easily be computed and implemented in a distributed environment with local recovery [16, 75].

Although SPT does not offer a minimal cost tree, it is widely accepted and used due to the fact that multicast offers a substantial savings over multiple unicast [12, 74]. However, source-based routing is not good for large scale applications such as multipoint-to-multipoint communication.

3.3.2 Shared Tree Routing

Shared tree multicast constructs a single tree that is shared by all the senders for multicasting the data. The *Steiner tree* and *Core based tree* are the two main algorithms used for constructing a shared tree.

3.3.2.1 Steiner Tree

The Steiner tree is a monolithic algorithm [16]. It designs a multicast tree that spans a group of members with minimal cost. The Steiner tree used for all sources within a multicast group is the same irrespective of the role of a sender and a receiver. Therefore, it requires only one state entry per group and can scale well. However, the problem of constructing the Steiner tree is an NP-complete problem. This means that finding a Steiner tree in a graph has an exponential running time for a result that is optimal. As it is difficult to obtain a Steiner tree with minimal cost in a large graph, it is adequate to accept a near optimal solution instead of a minimal Steiner tree. There are some approximation algorithms such as those defined in Winter [76] that can produce a near optimal solution. However, the Steiner tree algorithm has never been implemented in practical multicast protocols, but only simulated [16, 75].

The Steiner tree has a few other drawbacks that are noted by Diot et al. [16] and Wei and Estrin [75]. The two main drawbacks are: (1) A rerun is required each time there is a change in group membership or in network topology. (2) As the tree designed is undirected, it requires a network with symmetric links for group communication.

3.3.2.2 Core Based Tree

Core based routing is the most recent approach developed for multiple senders and multiple receivers, as opposed to single sender and fixed recipient scenarios. It is suitable for sparsely distributed receivers but incurs an extra delay/cost over source-rooted trees.

Figure 3.3: A core-based multicast routing.

In core based routing a fixed node is chosen as a core and a shortest path tree is constructed from the core to all the receivers. The shortest path tree is the same for all sources. However, the path from the source to the core may differ from source to source². If the source node is on the shortest path from the core to a receiver, then this part is empty [3]. Figure 3.3 shows an example of core based multicasting with two sources S_1 and S_2 .

²It is that portion of the shortest path from the source to the core that is distinct from the core's shortest path tree.

Core selection is a main problem in core based multicast because the whole performance of the distribution tree depends on the shared tree rooted at the core node³.

3.4 Internet Routing Protocols

This section briefly mentions some of the multicast protocols used in the Internet. The very first multicast protocol that was used within the Internet was based on the Deering model [13]. Later, Distance-Vector multicast routing protocol (DVMRP) [74] was developed. DVMRP builds a source-based multicast tree for delivering the data. This multicast tree is built by exchanging the reverse path distance. A revision of DVMRP called Hierarchical DVMRP [73] was developed later to introduce the concept of hierarchy.

The Multicast open shortest path first (MOSPF) allows routers to build an efficient source-based tree by using the link state database [51]. Since the aforementioned source-based protocols do not scale well, they are not suitable for global Internet use [16]. Therefore, the Internet Engineering Task Force (IETF) has defined core-based and protocol-independent multicast [15] that can scale well in sparse or dense operating modes.

3.5 Cost Sharing in Multicast Transmission

In this section, *cost sharing mechanisms* for multicast transmission are discussed, and some of the computational and economic constraints that are desirable for a *strategyproof* pricing mechanism are introduced. A mechanism is said to be strategyproof if the dominant strategy of each user (receiver) is to reveal the truth of their utility. A mechanism is said to be *group strategyproof* if truth telling is a dominant strategy for a coalition as well.

³There are different heuristics available for core selection. Refer to [4] for further details.

As discussed in Chapter 1, multicast is an inherent mode of delivery for applications such as video concerts, movies and video conferencing. However, delivering a multicast transmission to a large number of receivers requires a lot of network resources such as bandwidth and CPU time. Therefore, it cannot be transmitted with best effort service.

As network resources are scarce and cannot be used free of cost, multicast transmissions also incur a significant cost. Unlike the unicast case, resources used by multicast transmission are not directly attributable to a single receiver; they must be shared among all the receivers. Therefore, one must find a way to distribute the cost among all receivers. A mechanism to distribute the cost among the receivers is called a *cost-sharing* mechanism. The cost sharing mechanism determines which user receives the service and at what price.

For example, consider the multicast transmission of a high quality video concert. Suppose there are n receivers that want to receive the service and a user i shows their willingness to pay by their utility u_i after seeing the video concert. Assume that $x_i \geq 0$ is the amount that user i needs to pay for receiving the concert. Then the individual welfare of the user i is given by the formula $w_i = u_i q_i - x_i$, where q_i denotes whether user i receives the concert or not, $q_i = 1$ if the user receives the transmission, and $q_i = 0$ otherwise. User individual benefit or the welfare w_i decides whether the user wants to receive the transmission or not.

In the above example, each receiver is free to report any value. Since a network does not know *a priori* the valuation u_i of each user i , it has to rely on the values reported by the users. Receivers can report false values to maximize their own satisfaction or welfare. Therefore, it cannot be assumed that receivers truthfully report their utilities. Instead, it is assumed that the receivers are selfish and report the value that maximizes their own welfare w_i . Such deception can be avoided by developing a *strategyproof* cost sharing mechanism in which each user maximizes their welfare by revealing the true value of u_i , i.e., $w_i(u) > w_i(\hat{u})$.

This output and the payoff are the functions of the utility vector u . Therefore,

a strategyproof cost sharing mechanism is defined by the functions $x(u)$ and $q(u)$ ⁴. These functions calculate the individual payoff x_i and the output q_i for each receiver i .

3.5.1 Basic Definitions

This section describes the terminology and basic requirements that a strategyproof mechanism should possess. Let $S = \{1, \dots, n\}$ denote the set of users willing to receive a multicast transmission and $c(s)$ is the cost of serving any subset s of the set of users S . A cost function is a *submodular* function of the set of receivers, if

1. $c(\emptyset) = 0$ and,
2. for all $R_1, R_2 \subset S$, $c(R_1) + c(R_2) \geq c(R_1 \cup R_2) + c(R_1 \cap R_2)$.

Submodularity of the cost function guarantees that the union of the two efficient sets is also an efficient set [19]. A cost function is said to be *supermodular* if the inequality of the second condition is reversed. *Nondecreasing* means the cost of serving the users will not decrease with the increase in the number of users, *i.e.*, $c(R_1 + i) \geq c(R_1)$ for all $i \notin R_1$.

A cost sharing mechanism accepts the receivers' utility vector $u = \langle u_1 \dots u_n \rangle$ (where u_i corresponds to the utility of the i^{th} user, which is privately known to user i) as an input and calculates the output vector $q = \langle q_1 \dots q_n \rangle$ with the corresponding payoff $x = \langle x_1 \dots x_n \rangle$. The output vector q gives the information regarding the final set of receivers. The receiver i with $q_i = 1$ is selected for receiving the transmission.

The basic requirements that a reasonable strategyproof cost sharing mechanism should possess are:

⁴As discussed in Sections 2.4.1 and 2.4.2, a mechanism is merely a pair of output and payoff functions.

1. **No Positive Transfer (NPT):** For all u , each receiver $i \in R_1$, pays an amount $x_i \geq 0$ for receiving a message; i.e., no receiver will get paid for receiving the transmission.
2. **Voluntary Participation (VP):** Receivers are always free not to receive the transmission and not to be charged for a utility profile u that results in an individual welfare of zero; i.e., receivers will never be asked to pay more than their reported utility, i.e., $w_i(u) \geq 0$.
3. **Consumer Sovereignty (CS):** For all u , a cost sharing mechanism cannot exclude a user i with a sufficiently high utility (willingness to pay) value; i.e., by reporting a high enough willingness to pay, an agent is guaranteed to receive the service.

Many strategyproof cost-sharing mechanisms can satisfy the above requirements [19]. However, not all of them are of practical importance. Therefore, the requirements of *budget balance* and *efficiency* are included to stay away from mechanisms that are of no practical importance.

1. **Budget Balance (BB):** Budget balance requires that the payment collected from the final set of receivers exactly covers the total cost of multicasting the message; i.e., $\sum_{i \in R_m} x_i = c(T(R_m))$ where $c(T(R_m)) = \sum_{l \in E_m} c_l(r)$.
2. **Efficiency:** Efficiency requires that the final receiver set maximizes the overall welfare ($\max(\sum_{i \in R_m^*} u_i - c(T(R_m^*))) \quad \forall R_m^* \subset R_m$) of the network for a given utility profile u . A set with the maximum welfare is called an *efficient set*.

Ideally, one wants to design a mechanism that satisfies all the above-mentioned requirements. However, a classical result in game theory [27, 62] shows that there is no such strategyproof mechanism that is both budget balanced and efficient. They are basically mutually exclusive properties.

Strategyproofness for cost-sharing problems gives two classes of mechanisms [19] corresponding to the choice between efficiency and budget balance. In the first choice,

if the cost-function is non-decreasing and submodular then there is only one way to maximize the efficiency [50]. Such a mechanism is referred to as the *marginal cost* mechanism. The second choice, the *cross-monotonic cost sharing method* (see Section 3.5.3 for the definition), gives rise to the *Shapley value* mechanism. These mechanisms are discussed individually in Sections 3.5.2 and 3.5.3 respectively.

As such, a cost-sharing mechanism is merely a pair of formulas of $x(u)$ and $q(u)$. However, computing these values in the multicast transmission is not an easy task because receivers are distributed throughout the network and it is necessary to consider the cost of each link in the multicast tree for the final computation of vectors x and q . Moreover, each receiver i must be informed about their cost share $x_i(u)$ and $q_i(u)$. Therefore, the challenge is to find an algorithm that can implement the cost-sharing mechanism in a *distributed* manner. Moreover, a cost-sharing algorithm has to deal with a large user population in multicast; i.e., a scalable algorithm is required. An algorithm will not scale well if it requires a lot of messages for the computation of $x(u)$ and $q(u)$. Therefore, the ultimate goal is to design a scalable and distributed cost-sharing algorithm, i.e., one that incorporates the two most important factors by which the practical importance of an algorithm can be judged.

3.5.2 Marginal Cost Pricing Mechanism

The marginal cost mechanism is a special case of the Vickrey, Groves and Clarke's mechanism [28, 8]. It is an efficient and strategyproof mechanism under the quasi-linear utilities. The submodularity⁵ of the cost function guarantees that the union of the two efficient sets is also an efficient set. That is, we can always find a largest efficient set $R_m^*(u)$ of the initial set of receivers $R_m(u)$ for the given utility vector u . The allocation $q_i = 1$ for all $i \in R_m^*$ and $q_i = 0$ otherwise. Hence, netwelfare over the set of users in R_m^* is given by $NW(R_m^*) = \sum_{i \in R_m^*} u_i - c(T(R_m^*))$.

The cost share of any user $i \in R_m^*$ is computed as follows:

⁵See Section 3.5.1 for the definition of submodularity.

$$x_i = q_i u_i - (NW(R_m^*, u) - NW(R_m^* - i, u)) \quad (3.1)$$

The marginal cost pricing mechanism gives each user a welfare $w_i = (NW(R_m^*, u) - NW(R_m^* - i, u))$, which is the marginal contribution to the overall welfare provided by user i having non-zero utility for the transmission. The choice of a marginal cost pricing mechanism out of the whole VCG class depends on the fact that it is the only available VCG mechanism that meets the requirement of *NPT*, *VP* and *CS*. All VCG mechanisms possess *CS*.

Theorem 4 [50]: *The marginal cost mechanism is strategyproof, efficient, NPT, VP and CS.*

Although the marginal cost mechanism is strategyproof, it is not group strategyproof⁶. The main drawback associated with marginal cost mechanism is that it never runs a budget surplus but can run a budget deficit. In many cases, it generates no revenue at all [50].

3.5.3 Budget Balance (Shapley Value Mechanism)

The Shapley value mechanism is one of the most natural of all the strategyproof and budget-balanced mechanisms. Budget balanced mechanisms are derived from a *cost-sharing method*, which is a formula ξ associating to each cost function c and to each coalition R_m of agents. The cost-sharing method assigns a non-negative cost share ξ_i to agent $i \in R_m$. These cost shares must satisfy the constraints of budget-balance and cross-monotonicity. A cost-sharing method is said to be *cross-monotonic* if the cost share of the agent i does not increase with the increase in the size of receiver set. It can be written as follows:

⁶Truth telling is a dominant strategy for a coalition.

$$\sum_{i \in R_m} \xi_i(R_m) = c(T(R_m)) \quad (3.2)$$

$$\xi_i(R_m + j) \leq \xi_i(R_m) \quad \forall i, j \in R_m. \quad (3.3)$$

One can use the function ξ to define $q(u)$ and $x(u)$ in an iterative fashion. In the first step of the algorithm, all agents are offered a service; i.e., set $q_i^{(1)} = 1$. At each step k , $R_m^{(k)} = \{i | q_i^{(k-1)} = 1\}$; i.e., only those in step $(k-1)$ are allowed to be in the k^{th} step and $x_i^{(k)} = \xi_i^{(k)}$. If an agent survives step k , this means that $q_i^{(k)} = 1$ if $u_i \geq x_i^{(k)}$ or else $q_i^{(k)} = 0$ for $u_i \leq x_i^{(k)}$ [19].

The algorithm defines a monotonic sequence in set $R_m^{(k)}$, i.e. $R^{(k)} \subseteq R^{(k-1)}$, and will converge to equilibrium output $\bar{R}_m$, after a finite number of iterations. The value of x_i gives a cost share and $\bar{R}_m$ is the receiver set, if ξ^* is used as the Shapley value cost-sharing method. The cost share for the network sharing problem is defined as follows [19]:

$$\xi_i^*(R_m) = \sum_{\bar{R}_m \subseteq R_m - i} \frac{|\bar{R}_m|! (|R_m| - |\bar{R}_m| - 1)!}{|R_m|!} [c(T(\bar{R}_m \cup i)) - c(T(\bar{R}_m))]. \quad (3.4)$$

The formula seems quite complex, but it has an easy explanation. The cost of each link l is shared equally among all the users located downstream of the link. The mechanism $M(\xi^*)$ is referred to as the Shapley value mechanism.

Moulin and Shenker [50] showed that among all budget balanced mechanisms $M(\xi)$ derived from the cross monotonic cost-sharing methods, SH has the lowest worst-case loss of efficiency over all the utility profiles. If $\gamma(\xi)$ is the maximal welfare loss of mechanism over all profiles, the equation can be written as

$$\gamma(\xi) = \text{Max}_u [NW(R^*) - NW(R)] \quad (3.5)$$

$$\gamma(\xi^*) \leq \gamma(\xi) \quad \text{for all } \xi \neq \xi^*. \quad (3.6)$$

3.6 Related Work

A number of researchers have applied game theoretical properties to the Internet [18, 30, 42, 43]. However, most of this research is based on the concept of *Nash equilibrium*. The Nash equilibrium approach is based on the fact that simultaneous user selfishness leads the system to a self-consistent socially desirable outcome. For example, the Nash implementation approach [48, 10] for resource allocation mechanisms yields the socially desirable outcomes such as efficient, budget-balanced, etc. However, Friedman et al. [21] show that Nash equilibria are not necessarily achieved in a distributed setting such as the Internet and therefore different solution concepts are needed.

Strategyproofness is the desired solution concept for networks such as the Internet. Strategyproofness guarantees that no matter how the other agents behave, truth telling is the dominant strategy for each agent. There are many instances in the recent literature [1, 19, 38, 39] where algorithmic thinking is combined with game-theoretical properties, or, more generally, economic concepts are used to address problems arising in the context of pricing multicast transmission on the Internet. While the existence of selfish users on the Internet is quite natural, Internet architecture/services must be made safe from these users. Selfish users neither cooperate with nor are they harmful to other users. In fact, their only goal is to optimize their own welfare. Although, one cannot assume that the selfish users will cooperate, one can assume that they will respond to incentives. Thus, Internet architecture/services must be designed with incentives in mind. For example, McCanne [49] considers incentive issues in the packet dropping algorithms for receiver-driven layered multicast. Networks with multiple quality of services have also promoted the use of incentives in their design [22]. Mechanism design use incentives to cope with user selfishness. Therefore, it has been widely applied as an approach to various network designs to cope with selfish users [18, 65, 69, 70].

Questions of incentives are well studied in game theory, and there is a vast literature available on *mechanism design* or *implementation theory* [46]. In fact, game-theory emphasises incentive and pays little attention to computational or communication complexity, while theoretical computer science leans more towards complexity [19]. Noam Nisan and Amir Ronen [52, 53] are two of the first researchers to blend these fields together. Feigenbaum, Papadimitriou, and Scott Shenker [19] are the first to look into multicast transmission as a combination of game theory and theoretical computer science. They designed the marginal cost and Shapley value algorithms for the cost distribution of multicast transmissions. Recently Alder and Dan [1] also presented a marginal cost pricing mechanism for multicast transmission, each looking at a different type of multicast. Feigenbaum et al. consider single rate transmission while Alder and Dan consider the multirate transmission model.

This work is very much in the same spirit. While Feigenbaum et al. [19] did not consider the limits on link capacity and assumed that a source is free to transmit data at any rate, this research consider that there are different rates available at which sources can transmit data. In a practical situation, it is extremely important that a source should make a best choice globally about the rate of multicast transmission. This rate selection is not arbitrary but based on the preferences of users that are distributed throughout the network.

For multirate transmission, the focus here is on designing a mechanism that allocates the service to the receivers at different rates and also charges the corresponding cost share for receiving the service. The pricing of multicast applications using shared trees is also looked into. A communication efficient pricing scheme is proposed for multiparty communication (application involving multiple senders and multiple receivers) and single source multicast (application involving single sender and multiple receivers) using shared trees.

The rest of the thesis is divided into two parts based on the type of multicast tree. In the first part, the focus is to develop a distributed cost-sharing algorithm for the source rooted trees. In the second part, the cost-sharing algorithm for the shared trees is discussed. This work is mainly based on the marginal cost mechanism

because it is communication efficient as compared to the Shapley value mechanism [19].

Part I

Multicast Pricing of Source Rooted Tree

Chapter 4

Single Rate Multicast

4.1 Introduction

In this chapter, the problem of cost sharing in single rate source-rooted multicast transmission is considered. As discussed in Subsection 1.4.1, single rate transmission requires all the users of a multicast group to receive the transmission at the same rate.

The chapter proceeds as follows. It begins with the description of a single rate transmission model in Section 4.2 and describes the notions of *networth* or *social welfare*. In Section 4.3, the problem of *maximizing social welfare*, *maximizing the size of the final receiver set*, and *maximizing individual utility of the maximum number of users* is discussed. The marginal cost solutions to these problems are presented in Subsections 4.3.1, 4.3.2 and 4.3.3 respectively.

The Shapley value mechanism is considered in Section 4.4 and shown to be communication inefficient for cost sharing in single rate multicast transmission. Moreover, it is found that any choice about the source rate is difficult to make using the Shapley value pricing mechanism.

4.2 The Model

The communication model in the context of multicast transmission is described as follows. Consider a network $G = \{V, E\}$, where V is the set of network nodes and E is the set of bidirectional network links. A user population P resides¹ at the set of network nodes V , and is connected through a set of bidirectional link E .

The complete network is shared by a set of M multicast groups. The infrastructure of any multicast group $m \in M$ is specified by a tuple $\{S_m, R_m, T_m\}$, where $S_m \in V$ is the source node, $R_m \subseteq P$ is the set of receivers in group m and T_m is the source rooted multicast tree of group m . All the nodes of the multicast tree T_m are multicast enabled, i.e., multicasting is possible at every node of the multicast tree T_m and with no cost.

There is a fixed path $T_m(i)$ from the source to each user $i \in R_m$, and the multicast tree $T_m(R_m)$ is the union of these paths, i.e., $T_m(R_m) = \cup_{i \in R_m} T_m(i)$. This approach is quite simple for implementation and makes the multicast tree quite stable as the receivers join and leave the multicast group [19]. However, there is a drawback: the subtree connecting a certain subset R_m of receivers may be arbitrarily more costly than the cheapest tree (see Section 3.3).

In this model, it is assumed that $Y = \{y_1, \dots, y_n\}$ is the set of possible transmission rates, where y_1 and y_n correspond to the minimum and maximum possible rates of transmission respectively. The source can transmit the data at any rate $y_j \in Y$. As mentioned in Section 1.4.1, single rate multicast transmission requires that all receivers of the group m receive the service at the same rate $y_j \in Y$. Suppose $E_m \in E$ is the set of links in the multicast tree T_m and the source transmits the message at the rate y_j . If link $l \in E_m$ charges the price $c_l(y_j)$ for transmitting a single message at rate y_j , then the cost of multicasting a message at the rate y_j becomes equal to the sum of the price charged by each link.

¹In the current context, nodes are visualized to act like routers. Users residing at any node means the users connected to the network through this router or node.

$$c(T(R_m)) = \sum_{l \in E_m} c_l(y_j). \quad (4.1)$$

Furthermore, the total utility of the users after receiving the service at rate y_j becomes equal to the sum of the users' individual utility and can be written as

$$U_{R_m} = \sum_{i \in R_m} u_i(y_j). \quad (4.2)$$

Thus, the *overall social welfare* or *networth* for receiving the transmission can be given by:

$$NW(R_m) = U_{R_m} - c(T(R_m)).$$

Note that networth or overall social welfare measures the total benefit of providing the multicast transmission. It is not the sum of individual welfares and does not depend on the cost shares [19]. Due to the fact that $T_m(R_m) = \cup_{i \in R_m} T_m(i)$, the cost function $c(T(R_m))$ is a nondecreasing and submodular function of the set of receivers R_m .

It is considered that each node n knows the cost c_l of each link incident on it. Where c_l is a k -component vector such that $[c_l]_j$ denotes the price of the link l at the rate j , each receiver i expresses their willingness to pay via a bid, denoted by b_i , a k -component vector such that $[b_i]_j$ indicates the price that i is willing to pay to receive a message at rate j .

4.3 Marginal Cost Mechanism for Multicast Transmission

This section presents an efficient distributed algorithm for realizing the marginal cost mechanism in the source-rooted tree for single rate multicast transmission. It is considered that source S_m can choose² a rate from all the available rates. However,

²As explained previously in Section 4.2.

Figure 4.1: An example of a source-rooted multicast tree in the Internet.

this rate selection is not arbitrary but is based on certain information about the multicast group m .

Three variations of the source rate selection are considered. These are *maximizing social welfare*, *maximizing the size of the final set of receivers*, and *maximizing individual utility of the maximum number of users*. However, it is better to understand the concept of welfare at any node before describing the marginal cost algorithm for source rate selection.

Let $n \in V(R_m)$ be any node of the multicast group m and $res(n)$ denote the set of receivers located at the node n . Let $Ch(n)$ denote the children of node n and $W(n)$ denote the welfare at the node n , which is a k -component vector such that $W_j(n)$ gives the welfare corresponding to the rate j at node n . The welfare at any node n for the rate j depends on the sum of the bids received by the users located at node n or downstream minus the cost of all the links located downstream, including the one which connects node n to its parent $p(n)$. If c_p represents the cost of the parent link, the following formula³ can be used to compute the welfare $W_j(n)$.

$$W_j(n) = \max \left(0, \sum_{i \in res(n)} [b_i]_j + \sum_{n_i \in ch(n)} W_j(n_i) - ([c_p]_j - 1) \right) \quad \forall j \in Y \quad (4.3)$$

The fact that the marginal cost for a binary service⁴ is strategy-proof does not imply that the marginal cost for multiple goods is also a strategy proof mechanism. The following theorem states this result.

Theorem 5 [1]: *The marginal cost mechanism for multiple goods is strategy-proof, efficient, NPT, VP, and CS.*

This thesis focuses on the communication cost rather than the local computation cost. Moreover, it is assumed that the message size is reasonable due to distributed

³Note that all the operations are component-wise vector operations.

⁴Binary service means that there is only a single rate available for the transmission and the source has no other rate available to choose from.

computation through out the network. It would not be satisfactory, for example, if each node, after receiving a message from all its children, concatenates all of the received messages, utilities of the resident users, and cost of the parent link and send it to its parent and so on until the message reaches to the source node⁵. However, this is not the case with our cost sharing algorithms, and therefore, message size is not a major issue in this thesis.

4.3.1 Maximizing Social Welfare

In this section, a distributed and communication efficient marginal cost sharing algorithm is described for a single-rate transmission model. Source rate selection of the single-rate transmission model depends on the social welfare⁶ achieved by each available rate. It is necessary to choose a rate with maximum social welfare, and, compute the final set of receivers with their payoffs, *i.e.*, who is going to receive the transmission and at what price? The following statement describes the problem formally:

Problem Statement: *Design a distributed and communication efficient marginal cost algorithm that can choose a source rate with maximum social welfare, subject to the link constraints, as well as the maximum/minimum rate constraints. The problem can be written as:*

$$P : \max_{\forall y \in Y} \left(\sum_{i \in R_m} U_i(y) - c(T(R_m(y))) \right)$$

subject to:

$$y = y(i) = y(i') \quad \forall i, i' \in R_m \text{ and } i \neq i' \quad (4.4)$$

$$y \in Y \quad (4.5)$$

User preferences for the amount of bandwidth y are characterized by a concave utility function $U(y)$. This optimization problem has a solution [41] for the above

⁵This approach achieves the goal of $O(1)$ messages per link with no bound on the message size.

⁶Social welfare is equal to the sum of utilities minus the total cost of transmission.

mentioned constraints. Equation 4.4 requires that all receivers of the multicast group receive the transmission at the same rate⁷. According to Equation 4.5, source rate y should be an element of the possible rate set Y .

4.3.1.1 An Algorithm

An Algorithm for maximizing social welfare is a distributed algorithm that chooses a rate $y = j_s$ from all the available rates such that the source maximizes the network welfare if it transmits the message at the rate j_s . There are k different rates at which a source can transmit the multicast message.

This computation is divided into two phases. In the first phase, the welfare at each node is computed by bottom-up traversal of the tree. In the second phase, the final set of receivers and their corresponding cost share is computed by a top down traversal of the tree as explained below:

1. Bottom-Up Traversal

Each user $i \in R_m$ submits a bid b_i for receiving the service, which is a k -component vector, where $[b_i]_j$ represents the price that i is willing to pay for receiving the service at rate j . The welfare value at any node $n \in V(R_m)$ is computed by a *bottom-up traversal* of the multicast tree T_{R_m} . First, each leaf node⁸ sends the computed welfare to its parent node $p(n)$ then a parent node gathers all such information (from the child nodes) and calculates its own welfare⁹, and sends the welfare vector to its parent node, and so on, until the welfare vector reaches the source node.

The welfare vector $W(S_m)$ at the source node gives the information regarding the social welfare achieved by the source for each available rate. In other words, the source node has full knowledge about the welfare achieved for each available rate and it can choose the rate j_s with the maximum social welfare.

⁷This is the condition for single rate transmission.

⁸For leaf node n , $\sum_{n_i \in ch(n)} W_j(n_i) = 0$.

⁹All the additions are component-wise vector addition.

Algorithm 1 : Bottom-Up traversal for computing welfare values at each node of T_m .

- 1: **for** each node $n \in V(R_m)$ **do**
 - 2: After receiving the welfare vector $W(Ch(n))$ from all of its children, node n computes,
 - 3: **for** each rate j **do**
 - 4: **for** each child $n_i \in Ch(n)$ **do**
 - 5: **if** ($W_j(n_i) \geq 1$) **then**
 - 6: $W_j(n_i) = W_j(n_i) - 1$
 - 7: **else**
 - 8: $W_j(n_i) = 0$
 - 9: **end if**
 - 10: **end for**
 - 11: Node n computes the welfare $W_j(n)$ by using the formula

$$W_j(n) = \max \left(0, \sum_{i \in res(n)} [b_i]_j + \sum_{n_i \in Ch(n)} W_j(n_i) - ([c_p]_j - 1) \right)$$
 - 12: **end for**
 - 13: Each node n sends the welfare vector $W(n)$ to its parent $p(n)$.
 - 14: **end for**
 - 15: Source node S_m selects the rate j , for which $W_j(S_m)$ is maximum.
-

As an example, Figure 4.2 shows a node n with three children, $node_1$, $node_2$ and $node_3$, and a user, $User_1$, on the node n . $Rate_1$, $Rate_2$ and $Rate_3$ are the possible sources rate and node n needs to choose a rate which maximizes the social welfare.

In Table 4.1, columns $W(1)$, $W(2)$ and $W(3)$ give the corresponding welfare achieved by $node_1$, $node_2$ and $node_3$ with respect to available rates $Rate_1$, $Rate_2$ and $Rate_3$. Column $User_1$ presents the utility of $User_1$ for the different rates. The cost of the parent link for the different rates is given under the column c_p .

<i>Preference</i>	$W(1)$	$W(2)$	$W(3)$	$User_1$	c_p
$Rate_1$	10	5	5	5	35
$Rate_2$	8	15	12	10	38
$Rate_3$	20	0	25	8	48

Table 4.1: Preferences of children and user located at node n .

$$W_j(n) = \max \left(0, \sum_{i \in res(n)} [b_i]_j + \sum_{n_i \in Ch(n)} W_j(n_i) - ([c_p]_j - 1) \right) \quad \forall j \in Y$$

Figure 4.2: Computing welfare during the bottom-up traversal of the tree.

Preference	$W_j(n)$
$Rate_1$	$\max(0, 9 + 4 + 4 + 5 - 35 + 1) = 0$
$Rate_2$	$\max(0, 7 + 14 + 11 + 10 - 38 + 1) = 5$
$Rate_3$	$\max(0, 19 + 0 + 24 + 8 - 48 + 1) = 4$

Table 4.2: An example of welfare computation at node n .

Now, the goal is to determine how to calculate the social preference at node n . The social preference at node n is basically given by the welfare vector at node n . Equation 4.3 is used to compute it.

From Table 4.2 it is clear that rate $Rate_2$ is the socially desirable source rate at node n , because n achieves the maximum welfare at $Rate_2$.

Lemma 1 *Algorithm 1 returns the source rate with the maximum possible network welfare under the single rate multicast transmission.*

Proof: The proof that Algorithm 1 returns the source rate with the maximum

possible network welfare is based on induction on height h_n of node n . The height of the multicast tree is denoted by h .

For the base case, consider a leaf node n of the multicast tree T_m . The welfare obtained at the leaf node depends on the user's bid and the cost of the parent link. All users bid and the cost of the parent link is available at a leaf node. Therefore, leaf nodes can easily compute the welfare value and pass it to the parent node.

For the inductive step, it is assumed that node n , at height $h_n < h$, receives the welfare from all its children and shown that welfare $W(n)$ is passed to the parent $p(n)$. The correctness of the computed welfare follows from the fact that node n has information about b_i for all $i \in res(n)$ and the cost of the parent link. Since it is assumed that node n knows the values of welfare $W(n')$ for all $n' \in Ch(n)$, it can easily compute the welfare vector and send it to its parent $p(n)$. That is, each node (including the source) can compute welfare for each available rate and the source node can select the rate with the maximum possible welfare. $\square$

2. Top-Down Traversal

This section shows how to use the welfare values to compute the marginal cost share of the final set of receivers $R_m^*(u)$ ¹⁰ at the socially desirable rate j_s . First, how to compute the final set of receivers is discussed, and then the cost allocation of the final set of receivers.

A top-down traversal of the multicast tree is required to compute the final set of receivers. Starting from the source node, the welfare achieved by all the nodes $n \in V(R_m)$ at rate j_s is checked. If the welfare achieved at any node n is greater than or equal to one, the service at rate j_s will be awarded to all the users located at node n . However, if the welfare achieved at j_s is less than one, then all the users located on the subtree rooted at node n will be dropped.

¹⁰Here $R_m^*(u)$ is the final set of receivers for the current utility vector u .

Algorithm 2 Top-down traversal of the tree T_{R_m} .

```

1: Source node  $S_m$  sends  $A = W_{j_s}(S_m) - 1$  and rate  $j_s$  to each of its children.
2: for each node  $n \in \{V(R_m) - S_m\}$  do
3:   After receiving a message  $A, j_s$  from parent  $p(n)$ 
4:   if ( $W_{j_s}(n) < 1$  or  $A < 0$ ) then
5:     for all users  $i \in res(n)$  do
6:       set  $x_i \leftarrow 0$  and  $q_i \leftarrow 0$ 
7:       send  $-1$  to  $\beta$  for all  $\beta \in ch(n)$ 
8:     end for
9:   else
10:     $W_{j_s}(n) = W_{j_s}(n) - 1$ 
11:     $A \leftarrow \min(A, W_{j_s}(n))$ 
12:    for each  $i \in res(n)$  do
13:      if ( $b_{j_s} \leq A$ ) then
14:         $x_i \leftarrow 0$ 
15:      else
16:         $x_i \leftarrow b_{j_s} - A$ 
17:      end if
18:    end for
19:    for each  $\alpha \in Ch(n)$  do
20:      Send  $A, j_s$  to  $\alpha$ 
21:    end for
22:  end if
23: end for

```

The marginal cost of receiver i depends upon the maximum welfare achieved by the source without admitting receiver i . This maximum welfare can be easily computed by a bottom-up traversal of the tree without admitting receiver i . However, this approach is not message efficient as it requires one bottom-up traversal per receiver.

How to compute the marginal cost share using a single top-down traversal of the tree is now described. The idea [19] is quite simple: for each node $n \in V(R_m)$, it is necessary to keep track of the smallest value of the welfare at any node from root to the current node n , because this minimum welfare value might occur at the current node n .

Let A be the smallest welfare¹¹ at any node α in the path from n to the root S_m . There are two possible cases:

- (a) If utility u_i of user i is greater than A for the rate j_s (i.e., $u_i \geq A$), then dropping the user i results in the elimination of a subtree of total welfare A . Therefore, the marginal contribution provided by user i to the overall welfare is equal to A . Hence, the cost share of user i becomes equal to:

$$\begin{aligned} x_i &= u_i - (W(R_m^*, u) - W(R_m^* - i, u)) \\ &= u_i - A. \end{aligned} \quad (4.6)$$

- (b) If the utility of an agent i is less than that of the minimum welfare A , then dropping user i does not affect the efficient set. That is, $R^*(u)$ is the same as $R^*(u_i = 0, u)$. The welfare difference $W(R_m^*, u) - W(R_m^* - i, u)$ is equal to u_i . Hence, the cost share of user i becomes equal to

$$\begin{aligned} x_i &= u_i - (W(R_m^*, u) - W(R_m^* - i, u)) \\ &= u_i - u_i \\ &= 0. \end{aligned} \quad (4.7)$$

For example, in Table 4.3 the cost share of the $User_1$ for the two different values of A is computed. This idea is described by Algorithm 2.

Preference	$W(n)$	$A_{p(n)}$	A	$User_1$	x_{User_1}
Case ₁	7	10	6	10	4
Case ₂	7	5	5	1	0

Table 4.3: Cost computation of $User_1$.

Lemma 2 *Algorithm 2 computes the marginal cost of the user $i \in R_m^*$ under the single rate transmission model as discussed in Section 4.2.*

¹¹For the current utility vector u .

$$x_i = \max \{0, b_i - A \}$$

Figure 4.3: Computing cost share and allocation during the top-down traversal of the tree.

Proof: As discussed earlier in Section 3.5.2, the marginal cost mechanism gives each user a welfare equal to their marginal contribution to the overall welfare. Therefore, to prove that Algorithm 2 computes the correct value of marginal cost share x_i for all $i \in R_m^*$, we need to show that the computation of marginal welfare is correct.

The computation of marginal welfare is based on the fact that dropping user i decreases the welfare of all the nodes (from the current node up to the source node) by u_i . Therefore, if utility $u_i > A$, then at some lowest ancestor $\hat{n}$, dropping the user i makes the welfare negative and it is necessary to drop all the users located on the subtree rooted at node $\hat{n}$. Dropping users of the subtree rooted at node $\hat{n}$ might result in other nodes with negative welfare. This process continues until the removal of a minimum-welfare subtree A . Thus, the marginal welfare provided by user $i \in R_m^*$ is equal to A . However, if utility $u_i < A$ then the tree structure of $R_m^*(u - i)$ is the same as $R_m^*(u)$. That is, the marginal welfare due to user i is equal to its utility u_i . $\square$

Theorem 6 *Algorithms 1 and 2 are optimal with respect to the number of messages sent for source rate selection and for the computation of the cost share of the final set of receivers¹².*

Proof: Algorithm 1 computes the welfare at each node $n \in V(R_m)$. This welfare computation requires a single bottom-up traversal of the tree, that is, one message per link of the multicast tree. While Algorithm 2 computes the final set of users with their corresponding cost share for receiving the transmission. However, we should observe that Algorithm 2 combines the computation of service and cost allocation and needs only one top-down traversal of the tree, that is, one message per link of the multicast tree. Therefore, the implementation of marginal cost mechanism needs only two messages per link (one bottom-up and one top-down) in total and is optimal with respect to the number of messages sent for source rate selection and for the computation of the cost share of the final set of receivers . $\square$

Figure 4.4 presents an example of the marginal cost pricing mechanism in multicast transmission. This example shows that if the social welfare at any node n is equal to zero, then the receivers in the subtree rooted at that node n will not receive the service. For example, nodes C and D have zero welfare and therefore the final efficient set will not contain the receiver from the subtree rooted at nodes C and D .

¹²Final set is the same as efficient set for the rate j_s .

_____ Shows the part in the final multicast tree
 - - - - - Shows the part not in the final multicast tree

S : Source node

$N = \{A, B, C, D, E, F, G\}$

$R_m = \{a, b, c, d, e, f, g\}$

$R_m^* = \{a, b, e, f, g\}$

a, b, c, d, e, f, g : Represents the set of users at the node
 A, B, C, D, E, F, G respectively

Figure 4.4: An example of marginal cost service allocation at the rate j_s .

4.3.2 Maximizing the Size of the Final Set of Receivers

In this section, a variation of the social welfare maximizing problem¹³ is considered. The main difference between the social welfare maximizing problem and the current one is that of source rate selection criteria. In this problem, attention is focussed on facilitating the maximum number of users to receive multicast transmission. This problem can be formally described as follows:

Problem Statement: *Design a distributed and communication efficient marginal cost algorithm that can choose a source rate with the maximum number of users in the final set of receivers.*

It is known that the users are distributed through the network and that the source has no knowledge about the number of users located at each node. Moreover, even if the source knows the number of users located downstream, it is hard to guess how many users are going to be served at each rate because the service location at each rate depends on the corresponding user utilities and the cost of transmission at that rate.

4.3.2.1 An Algorithm

Selecting the source rate with the largest final set of receivers is not a trivial task because of the distributed environment in which all the receivers are distributed throughout the whole network. Therefore, the main goal is to design an algorithm that can provide the information regarding the number of users in the final receiver set corresponding to each available rate. The aim is not restricted just to the design of a distributed algorithm; the algorithm should also be communication efficient.

First, the idea behind solving the above mentioned problem is described. The terminology used here is that described in Section 4.3.1. The approach used is very similar to that of the social welfare maximizing problem.

¹³As discussed in Section 4.3.1.

1. Bottom-Up Traversal

Each user $i \in R_m$ bids b_i for receiving the service. A node n gathers the bid b_i from all the users $i \in res(n)$. When node n receives a welfare from all the child nodes, it can easily calculate the welfare by using the formula:

$$W_j(n) = \max \left(0, \sum_{i \in res(n)} [b_i]_j + \sum_{n_i \in ch(n)} W_j(n_i) - ([c_p]_j - 1) \right) \quad (4.8)$$

However, the question is how to calculate the number of receivers in the final set corresponding to each available rate. The idea for calculating this is quite simple. The number of users at each node n that are going to be part of the final receiver set is tracked by building a *number of users* vector $Nuser(n)$, that is a k -component vector where $Nuser_j(n)$ represents the number of users that are going to be the part of the final receiver set at the rate j and are located in the subtree rooted at the node n . If $Nres(n)$ represents the number of users located at node n , then, following strategy for the computation of $Nuser_j(n)$ can be applied:

$$Nuser_j(n) = \begin{cases} 0 & \text{if } W_j(n) < 1, \\ \sum_{\hat{n} \in Ch(n)} Nuser_j(\hat{n}) + Nres(n) & \text{if } W_j(n) \geq 1. \end{cases}$$

The computation of $Nuser_j(n)$ depends on the welfare achieved at the rate j . The reason behind using the above formula is that the subtree with negative welfare is not going to be served. In other words, these users (with negative welfare) are not part of the final receiver set.

This is how Algorithm 3 works. First, the welfare vector $W(n)$ and the vector $Nuser(n)$ are computed at the leaf node n during the bottom-up traversal of the tree. A leaf node sends the welfare and number of users vector to its parent. A parent node gathers the information from the child nodes and calculates its own welfare $W(n)$ and $Nuser(n)$. This process repeats until welfare and the number of users vector reaches the source node.

Algorithm 3 Bottom-Up traversal for computing $W(S_m)$ and $Nuser(n)$ at each Node of T_m .

```

1: for each node  $n \in V(R_m)$  do
2:    $Nuser(n) = 0$ 
3:   After receiving the welfare vector  $W(Ch(n))$  and the users vector
      $Nuser(Ch(n))$  from all the child nodes, node  $n$  computes
4:   for each rate  $j$  do
5:     for each children  $n_i \in Ch(n)$  do
6:       if ( $W_j(n_i) \geq 1$ ) then
7:          $W_j(n_i) = W_j(n_i) - 1$ 
8:       else
9:          $W_j(n_i) = 0$ 
10:      end if
11:    end for
12:    Node  $n$  computes the welfare  $W_j(n)$ 
        
$$W_j(n) = \max \left( 0, \sum_{i \in res(n)} [b_i]_j + \sum_{n_i \in ch(n)} W_j(n_i) - ([c_p]_j - 1) \right)$$

13:    if ( $W_j(n) \geq 1$ ) then
14:       $Nuser_j(n) = \sum_{n_i \in Ch(n)} Nuser_j(n_i) + Nres(n)$ 
15:    else
16:       $Nuser_j(n) \leftarrow 0$ 
17:    end if
18:  end for
19:  Each node  $n$ , sends the welfare and user vector to its parent  $p(n)$ .
20: end for
21: Source node  $S_m$  select the rate  $j$ , for which  $Nuser_j(S_m)$  is maximum.

```

The number of users vector $Nuser(S_m)$ at the source node gives the information regarding the number of users in the final receiver set for each available rate. The source node can choose the rate $y = j_{maxuser}$ with the maximum number of users. If the rates have the same number of users, the source will select the rate with maximum social welfare.

The example presented in Section 4.3.1.1 could be used here. However, this time each node α also reports the $Nuser(\alpha)$ vector. Table 4.4 presents the example value of the welfare and the number of users at nodes $node_1$, $node_2$ and $node_3$ with respect to available rates $Rate_1$, $Rate_2$ and $Rate_3$.

Preference	$(W(1), Nuser(1))$	$(W(2), Nuser(2))$	$(W(3), Nuser(3))$	$User_1$
$Rate_1$	(10, 3)	(5, 2)	(5, 2)	5
$Rate_2$	(8, 1)	(15, 3)	(12, 3)	10
$Rate_3$	(20, 4)	(0, 0)	(25, 5)	8

Table 4.4: Welfare and number of users vector from all the children of node n for each available rate.

Preference	c_p	$W_j(n)$	$Nuser(n)$
$Rate_1$	35	$\max(0, 9 + 4 + 4 + 5 - 35 + 1) = 0$	0
$Rate_2$	38	$\max(0, 7 + 14 + 11 + 10 - 38 + 1) = 5$	8
$Rate_3$	48	$\max(0, 19 + 0 + 24 + 8 - 48 + 1) = 4$	10

Table 4.5: An example for the computation of number of users vector $Nuser(n)$.

Table 4.5 shows the computation of the welfare vector and number of users vector according to Algorithm 3. It is clear from the table that rate $Rate_3$ is the socially desirable source rate at node n , because rate $Rate_3$ can serve the maximum number of users rooted at node n .

Lemma 3 *Algorithm 3 returns the source rate with the maximum number of receivers in the final set.*

Proof: The proof that Algorithm 3 returns the source rate with the maximum number of users in the final set is based on induction on the height $h_n \leq h$ of node n , where h is the height of the multicast tree.

For the base case, consider a leaf node n of the multicast tree T_m . The computation of number of users vector $Nuser(n)$ depends upon the welfare achieved at the leaf node. Since each node knows the welfare $W(n)$ (Lemma 1), it can easily compute $Nuser(n)$ and send it to the parent $p(n)$.

For the inductive step, it is assumed that node n , at height $h_n < h$, receives the values of number of user vector from all its children, and shown that node n sends the correct value of $Nuser(n)$ to its parent. The correctness of the

$Nuser(n)$ vector follows from the fact that node n can compute the welfare at node n and also knows the number of users located at node n . Therefore, it can compute the $Nuser_j(n) = \sum_{\hat{n} \in Ch(n)} Nuser_j(\hat{n}) + Nres(n)$ for each available rate and sends it to its parent. Hence, each node computes the correct value of number of users for each available rate, and therefore, source can select the rate with the maximum number of users in the final set of receivers. $\square$

2. Top-Down Traversal

The cost share and service allocation is computed by the top-down traversal of the tree. The approach is similar to that described in Subsection 4.3.1.1 for the social welfare maximizing problem, i.e., the minimum welfare A at any node is tracked from the source to the current node. If the utility of user i is greater than that of A , the user is charged $x_i = b_i - A$ otherwise x_i is zero¹⁴. Algorithm 2 is used for the computation of x_i and q_i of each user.

Theorem 7 *Algorithms 2 and 3 are optimal with respect to the number of messages sent for source rate selection and for the computation of the cost share of the final set of receivers.*

Proof: Algorithms 3 computes the welfare and number of users vector at each node n of the multicast tree. This computation requires only a single bottom-up traversal of the tree, that is, one message per link of the multicast tree. Algorithm 2 computes the final set of receivers with their cost shares and needs only a single top-down traversal of the tree, that is, one message per link. Therefore, one can say that both Algorithms 2 and 3 are communication efficient and a total of two messages per link are required for source rate selection and for the computation of the cost shares of the final set of receivers. $\square$

¹⁴Refer to Section 4.3.1.1 for further information on top down traversal in a multicast tree.

4.3.3 Maximizing Individual Utility of the Maximum Number of Users

In this section, another criteria for the source rate selection is considered. In this problem, individual preferences are taken as a deciding factor. It is assumed that receivers have discrete utility for different rates of transmission.

Problem Statement: *Design a distributed and communication efficient marginal cost sharing algorithm that can choose a source rate maximizing the individual utility of the maximum number of users.*

4.3.3.1 An Algorithm

The problem is to find a distributed and scalable solution for the above mentioned problem. The approach is quite similar to what was explained in the previous problem. However, this time for each rate, only those users having the highest preference for a particular rate are tracked. This preference is given by the users' utility. For example, if the user has the maximum utility for rate j , then s/he has the highest preference for receiving the transmission at rate j . The vector $Ninusers(n)$ is used to keep track of the number of users and their highest preference. It is a k -component vector such that $Ninusers_j(n)$ represents the number of users located on the subtree rooted at n and having the highest preference for rate j .

The approach is quite similar to that used for maximizing the size of the final set of receivers in Section 4.3.2. One bottom-up and one top down traversal of the multicast tree are needed for the computation of the source rate, cost share and service allocation to the set of users as explained below.

1. Bottom-Up Traversal

In the bottom-up traversal, the welfare vector $W(n)$ is computed at each node n . The welfare vector $W(n)$ gives the information about the users located on

Algorithm 4 :Bottom-Up traversal for computing welfare values and the number of users corresponding to their highest preference at each node of T_m .

```

1: for each node  $n \in V(R_m)$  do
2:    $Ninres(n) = 0$ 
3:    $Ninusers(n) = 0$ 
4:   After receiving the welfare vector  $W(Ch(n))$  and the users vector
      $Ninusers(Ch(n))$  from all the child nodes, node  $n$  computes
5:   for each rate  $j$  do
6:     for each children  $n_i \in ch(n)$  do
7:       if ( $W_j(n_i) \geq 1$ ) then
8:          $W_j(n_i) = W_j(n_i) - 1$ 
9:       else
10:         $W_j(n_i) = 0$ 
11:      end if
12:    end for
13:    Node  $n$  computes the welfare  $W_j(n)$ 
        
$$W_j(n) = \max \left( 0, \sum_{i \in res(n)} [b_i]_j + \sum_{n_i \in ch(n)} W_j(n_i) - ([c_p]_j - 1) \right)$$

14:    if ( $W_j(n) \geq 1$ ) then
15:      For each users  $i \in res(n)$ 
16:        if ( $[b_i]_j == \max(b_i)$ ) then
17:           $Ninres_j = Ninres_j + 1$ 
18:        end if
19:         $Ninusers_j(n) = \sum_{n_i \in Ch(n)} Ninusers_j(n_i) + Ninres_j(n)$ 
20:      else
21:         $Ninusers_j(n) \leftarrow 0$ 
22:      end if
23:      Each node  $n$ , sends the welfare and user vector to its parent  $p(n)$ .
24:    end for
25:  end for
26: Source node  $S_m$  select the rate  $j$ , for which  $Ninusers_j(S_m)$  is maximum.

```

Preference	$(W(1), Ninuser(1))$	$(W(2), Ninuser(2))$	$(W(3), Ninuser(3))$	$User_1$
$Rate_1$	(10, 3)	(5, 2)	(5, 2)	5
$Rate_2$	(8, 1)	(15, 3)	(12, 3)	10
$Rate_3$	(20, 4)	(0, 0)	(25, 5)	8

Table 4.6: Welfare and $Ninusers$ vector from all the child of nodes n for each available rate.

Preference	c_p	$W_j(n)$	$Ninuser(n)$
$Rate_1$	35	$\max(0, 9 + 4 + 4 + 5 - 35 + 1) = 0$	0
$Rate_2$	38	$\max(0, 7 + 14 + 11 + 10 - 38 + 1) = 5$	8
$Rate_3$	48	$\max(0, 19 + 0 + 24 + 8 - 48 + 1) = 4$	9

Table 4.7: An example for $Ninusers(n)$ computation at node n .

the subtree rooted at node n . If the welfare achieved by node n at rate j is greater than or equal to one, it is necessary to check the utility vector $[b_i]$ of each receiver $i \in res(n)$ to find out the highest preference and the corresponding rate. For example, if $[b_i]_j = \max[b_i]$ then the user has a maximum utility at rate j and $Ninres_j(n) = Ninres_j(n) + 1$, otherwise $Ninres_j(n)$ remains unchanged. Where $Ninres_j(n)$ represent the number of user located at node n and having a highest preference for rate j . After computing the vector $Ninres(n)$, the $Ninusers(n)$ can be computed by using the following formula:

$$Ninusers_j(n) = \begin{cases} 0 & \text{if } W_j(n) < 1, \\ \sum_{\hat{n} \in Ch(n)} Ninusers_j(\hat{n}) + Ninres_j(n) & \text{if } W_j(n) \geq 1. \end{cases}$$

However, if the welfare achieved is less than one, it is necessary to set $Ninusers_j(n) = 0$ because any user located in the subtree rooted at node n as expressed by Algorithm 4 is not going to be served.

Lemma 4 *Algorithm 4 returns the source rate with the maximum number of users maximizing the individual utility of the maximum number of receivers in the final set.*

Proof: The proof that Algorithm 4 returns the source rate with the maximum number of users maximizing the individual utility of the maximum number of users is based on induction on the height of node n .

For the base case, consider a leaf node n of the multicast tree T_m . The computation of vector $Ninusers(n)$ depends upon the welfare achieved by the leaf node. Since it is known that the welfare computed at each node of the multicast tree is correct (see Lemma 1), it can be said that leaf node n sends the correct value of $Ninusers(n)$ to its parent $p(n)$.

For the inductive step, it is assumed that node n at height $h_n < h$ knows the values of $Ninusers(n')$ for all $n' \in Ch(n)$, and shown that node n sends the correct value of $Ninusers(n)$ to its parent. The correctness of the $Ninusers(n)$ values follows from the fact that node n knows the welfare at node n and the highest preference of all the receivers located at node n . Therefore, node n can easily compute $Ninusers_j(n) = \sum_{\hat{n} \in Ch(n)} Ninusers_j(\hat{n}) + Ninres_j(n)$ for all the available rates. Since each node computes the correct value of $Ninusers$ for each available rate, the source can select the rate maximizing the individual utility of the maximum number of users. $\square$

2. Top-Down Traversal

The cost share and service location is a part of top-down traversal. A marginal cost mechanism is used for the cost allocation. The algorithm and the idea behind the marginal cost computation is similar to what was explained in the social welfare maximizing problem. Refer to Algorithm 2 for further details on service and cost allocation in a marginal cost pricing scheme.

Theorem 8 *Algorithm 2 and 4 are optimal with respect to the number of messages sent for source rate selection and for the computation of cost shares of the final set of receivers.*

Proof: Algorithm 4 computes the welfare and $Ninusers(n)$ vector at each

node n of the multicast tree. This computation requires only a single bottom-up traversal of the tree, that is, one message per link of the multicast tree, while Algorithm 2 computes the final set of receivers with their corresponding cost shares. It should be noted that Algorithm 2 needs only a single top-down traversal of the tree to compute the cost share of the final set of receivers, that is, one message per link. Therefore, one can say that Algorithms 3 and 2 are communication efficient and require only two messages per link in total. $\square$

In the next section, another class of mechanism is used to solve the social welfare maximization problem as stated in Section 4.3.1.

4.4 Shapley Value Mechanism

In this section, the focus is to price single rate multicast transmission using the Shapley value mechanism. As discussed in Section 3.5.3, Shapley value is a budget balanced mechanism, i.e., the payment received by the users exactly covers the cost of transmission. In multicast transmission, the Shapley value mechanism divides the cost of a link l equally among all receivers located downstream. Figenbaum et al. [19] present a natural, iterative brute force algorithm that computes the cost shares by using $\Omega(N.P)$ number of message exchange, where N is the number of nodes and P is the user population.

Here a single rate transmission model as described in Section 4.2 is being used. According to this model, there is a set of possible source rate whose selection is based on certain information about the group. It is established that *it is hard to make any choice regarding the source rate selection with the Shapley value pricing mechanism.*

The social welfare maximizing problem is used to make this claim¹⁵. Social welfare at any node is computed by the Equation 4.3. The problem can be precisely described as follows:

¹⁵See Sections 4.3.2 and 4.3.3 for the other possible criteria for source rate selection.

Algorithm 5 :Bottom-Up traversal for computing number of users at each node of $n \in V(R_m)$.

- 1: **for** each node $n \in V(R_m)$ **do**
 - 2: After receiving the number of users located downstream from all of the children's, node n compute the total number of users located in the subtree rooted at node n by using

$$Nusers(n) = \sum_{n' \in ch(n)} Nusers(n') + Nres(n)$$
 - 3: Each node n sends the number of users in the subtree rooted at node n to its parent.
 - 4: **end for**
-

Problem Statement: *Develop a distributed and budget balanced cost sharing algorithm that can choose a source rate with maximum social welfare.*

4.4.1 Algorithm

The algorithm to solve the above problem is divided into two phases. In the first phase, the final set of receivers is found for each available source rate such that the payment received from the users exactly covers the cost of transmission and in the second phase the source rate with maximum social welfare¹⁶ is selected.

1. In the first phase of the algorithm, the budget balance algorithm [19] can be applied for all source rates. This algorithm specifically find the final set of receivers and the corresponding cost share for each available source rates. This cost and service allocation is done by the sequence of bottom-up and top-down traversals of the tree for each available rate as explained below:
 - The bottom-up traversal of the tree for the j^{th} source rate, starting from the leaf node and finally reaching up to the source node, determines the number of users $Nuser(n)$ located on the subtree rooted at each node $n \in V(R_m)$ of the multicast tree for the j^{th} source rate. Then, we do a top down traversal of the tree for finding a cost share at the j^{th} source rate.

¹⁶This is the rate for which the sum the users utility minus the cost of transmission is maximum.

- The source node initiates the top-down traversal of the tree by sending a message $A = 0$ to all its children. After receiving the message A , each node calculates the cost share of the user $i \in \text{res}(n)$ by using the formula $A = x_i = \frac{c_j(l)}{N_{\text{users}}(n)} + A$, where $c_j(l)$ is the cost of the parent link¹⁷ at the j^{th} rate. A gets its new value from x_i and each node forward this value to all its children. Each node allocate a cost share x_i to all its users. However, there are two possible cases:

- (a) If $[b_i]_j > x_i$, then the user set remains the same.
- (b) If $[b_i]_j < x_i$, then possibly user i does not want to receive the service and must be dropped from the initial set of users. However, dropping users decreases the number of users located on the subtree rooted at node n and the algorithm no longer remains budget balanced. Therefore, it is necessary to apply the Algorithm 5 and 6 iteratively until there are no more users to drop.

2. The second phase of the algorithm is similar to what was explained in the bottom-up traversal of the tree under Section 4.3.1. The aim of the second phase is to find out the rate with maximum social welfare for the current utility vector u .

Feigenbaum et al. [19] makes the following claim about the Shapley value mechanism.

Theorem 9 [19]: *There is an infinite family of multicast computations, with N nodes and $O(N)$ users, such that any linear distributed algorithm that implements the Shapley value mechanism requires $\Omega(N^2)$ messages exchange in the worst case.*

In Algorithms 5 and 6, we see that each user ends up paying a fraction of the cost of each link in its path from the source. This fraction depends on the number of users sharing this link. Moreover, in the Shapley value pricing mechanism all the users of node n pay a similar price for receiving the service.

¹⁷A network link between node n and its parent.

Algorithm 6 :Top down traversal of the tree T_{R_m} .

```

1: Source node  $S_m$  sends  $A = 0$  and  $j$  to each of the children.
2: for Each node  $n \in (V(R_m) - S_m)$  do
3:   After receiving the message  $A$ ,  $j$  node  $n$  computes
      $A = \left\lceil \frac{c_j(l)}{N_{users}(n)} \right\rceil + A$ 
4:   for each  $i \in res(n)$  do
5:     if  $[b_i]_j < A$  then
6:        $x_i \leftarrow 0$  and  $q_i \leftarrow 0$ 
7:        $res(n) = res(n) - 1$ 
8:     end if
9:   end for
10:  for each  $\beta \in Ch(n)$  do
11:    Send  $A$  to  $\beta$ 
12:  end for
13: end for

```

Theorem 10 *It is not possible to communication efficiently select a source rate with maximum social welfare using Shapley value pricing mechanism.*

Proof: As discussed in Subsection 4.4.1, we need to apply Shapley value pricing algorithm for each available rate and then compute the welfare for each source rate by doing bottom-up traversal of tree. However, Theorem 9 states that the Shapley value mechanism is communication inefficient, and therefore, it is not possible to communication efficiently select a source rate with maximum social welfare using Shapley value pricing mechanism. $\square$

There are a few other interesting results related to the Shapley value mechanism. These results show that the brute force algorithm gives an optimal result and the quadratic dependence is quite natural. The following lemma of the distributed computing confirms that the Shapley value mechanism is communication inefficient.

Lemma 5 : [19] *Any linear distributed algorithm by two nodes for checking the inequalities $Ax + By > a$ (assumed to have a feasible solution), where x is an n -vector known to node I, y is an n -vector known to node II, A, B are $n \times n$ nonsingular matrices known to both, and a is an n -vector known to both, requires n message exchanges.*

Figure 4.5: A Shapley cost sharing algorithm.

Due to the lesser practical importance of the Shapley value mechanism, the remainder of this thesis is restricted to the marginal cost pricing mechanism.

4.5 Summary

In this chapter, the cost sharing problem of a single rate multicast transmission is considered. It is assumed that there are different rates available at which the source can transmit the data and the source need to make a best choice globally. However, this source rate selection is not arbitrary but based on the preferences of users for

different available rates. Three criteria for source rate selection are considered. These are maximizing social welfare, maximizing the size of the final set of receivers, and maximizing individual utility of the maximum number of users.

The focus is on two cost sharing mechanisms: marginal cost and Shapley value. It is shown that it is easy to implement a marginal cost sharing mechanism for charging the receivers under the above source rate selection criteria. For the Shapley value pricing mechanism, it is hard to make any source rate selection.

Chapter 5

Multirate Multicast

5.1 Introduction

In conventional multicast, all users of the same multicast group receive the service at the same rate. However, using a single-rate transmission to satisfy all users interested in a given transmission is inadequate due to the different characteristics of the receivers. For example, users with a low available bandwidth may prefer not to receive the transmission at a higher rate due to the possibility of high packet loss. Similarly, users with high bandwidth might complain about the poor quality due to the under-utilization of their system resources. Hence, a single rate transmission per multicast group is not a scalable solution. It may overwhelm the slow receivers and starve the fast ones.

One way to overcome this limitation is to use *multirate multicast* transmission. Multirate transmission permits different receivers of the same multicast group to receive the data at different rates according to their requirements, capabilities and the capacity of the path leading to them from the source node. In other words, rate adaptation in multirate multicast transmission moves from the source to the receivers. This is accomplished by distributing the data at different levels of quality using multiple layers and by giving each of the receivers the chance to adapt their

(a) Single Rate Transmission.

(b) Multirate Transmission.

Figure 5.1: A single rate vs. multirate multicast routing.

reception rate by adjusting the number of layers they can subscribe to, i.e., simply by joining and leaving layers.

Figure 5.1 shows a comparison between single rate and multirate multicast transmissions in the heterogeneous network. Three users u_1, u_2 , and u_3 are connected through 128 *Mbps*, 45 *Mbps* and 128 *Kbps* capacity lines. In single rate transmission, multicasting the data at the rate of 1 *Mbps* starves the users u_1 and u_2 and overwhelms receiver u_3 . However, in multirate transmission every user gets the data at the rate suitable to its capacity; i.e., each receiver receives the transmission according to available bandwidth. Carle et al. [5] show that the use of multirate multicast over ATM networks is good for both audio and video transmissions over the Internet and is more fair in comparison to single rate transmission [64].

This chapter focuses on pricing the multirate multicast transmission. The structure of this chapter is as follows. Section 5.2 provides a description of multirate multicast transmission and explains the network model. Section 5.2.2 formally introduces the problem of pricing the multirate multicast, Section 5.3 presents a marginal cost solution for cost sharing and service allocation in a layered paradigm. The Shapley value mechanism for multirate multicast transmission is not considered as it is impossible to implement due to its communication inefficiency.

5.2 Multirate Multicast Transmission

Multirate multicast transmission is accomplished by the *hierarchical encoding* of the data [24, 25]. Hierarchical data encoding is an efficient technique for fulfilling the requirements of heterogeneous networks. Using this technique, the source encodes the audio or video stream into various partitions or in a set of layers. Each of the layers is transmitted as a separate multicast group. Bandwidth of any layer L_i is usually predetermined and cannot be changed according to the needs of the network. Therefore, a receiver either receives a layer fully or does not receive the layer at all [67].

It should be noted that the layers $L_1, L_2, \dots, L_m$ in the multirate multicast are ordered (typically a base layer and a number of enhancement layers) such that any receiver who joins the group containing layer L_j must also join all the layers L_i for all $1 \leq i < j$. Therefore, a receiver joined up to layer L_i receives data from the source at an aggregate rate equal to the sum of the bandwidth of the layers L_1 to L_i . In fact, a receiver can increase the transmission rate by joining layers, while leaving layers decreases the aggregate rate. In layered multicast, the transmission rate at any link needs to be equal to the maximum rate among those of all receivers downstream from that link.

5.2.1 Network Model

In the context of multirate multicast routing, consider a communication network $G = \{V, E\}$, where V is the set of network nodes and E is the set of bidirectional network links. A user population P resides at a set of network nodes V and is connected by a bidirectional link E . For simplicity, all the receivers of the multicast session are assumed to be on the leaf node. In general, there is no harm in assuming this since a receiver at non-leaf node can be replaced by creating a new leaf node and placing the receiver in it, and connecting the new leaf node to the original non-leaf node with a link of infinite capacity [75].

A multicast session is considered to emanate from a single source S with R being the set of receivers. Each user $i \in R$ has a fixed path $T(i)$ to the source. This path is determined by the multicast routing infrastructure [19]. A source rooted multicast tree T is built for multicasting the data to a set of receivers $R \subseteq P$. This tree is the union of the shortest paths from the source to all the receivers, i.e., $T(R) = \cup_{i \in R} T(i)$.

The cost of any link $l \in T(R)$ is given by c_l , a k -component vector such that $[c_l]_j$ is the cost of the link l for providing up to the j^{th} group of layers. The cost of the multicast tree $c(T(R))$ is a nondecreasing and a submodular function of the set of receivers R . Moreover, all the nodes are assumed to be multicast enabled; i.e., nodes are capable of forwarding the data to multiple outgoing edges without any charge.

It is assumed that each receiver $i \in R$ expresses its willingness to pay through a bid denoted by b_i , a k -component vector, such that $[b_i]_j$ represents the price that i is willing to pay for receiving up to the j^{th} group of layers. The input data is distributed as follows: each node knows the cost c_l of each link l incident to node n and the willingness to pay b_i of user i located at node n . Let $Y = \{y_1, \dots, y_n\}$ be the set of possible aggregate rates and the rate at any link l of the multicast tree be given by the rate variable $y(l)$.

Consider Figure 5.2 as an example of multirate multicast where S is the source node and $R = \{r_1, \dots, r_{10}\}$ is the set of receiver nodes located in the subtree T_1 , rooted at the node S . As assumed, all receivers of the multicast session are placed on leaf

nodes. The remaining nodes of the multicast tree are classified into junction nodes and non-junction nodes. Junction node means that the multicast tree branches off at that node; for example $\{n_1, n_2, n_3, n_4\}$ are the junction nodes while $\{n_5, n_6\}$ are the non-junction nodes. The term internal node is used to represent both the junction and the non-junction nodes. A *Parent* $p(n)$ of any receiver or internal node n refers to the closest internal node in the upstream path towards the source, while *Child* $Ch(n)$ of any internal node n means the node whose parent is n .

These receivers want to receive the transmission at different aggregate rates $Rate_1$ $Rate_2$ $Rate_3$ ¹. Different types of arrows are used to represent different rates and user preferences. Each node n forwards its request to the parent node $p(n)$, which gathers all such requests from its child nodes and chooses the maximum rate from all of the requested rates, and requests that rate from its parent. These requests go up the multicast tree through the intermediate nodes till they reach the source node. In this way the source knows at what rate it has to send the data to its children. The source sends the data to its children at the requested rates; these nodes then send the traffic to their child nodes and the process repeats. In this way, the data from the source node reaches all the receivers at their requested rates.

5.2.2 Problem Description

This section states the social welfare optimization problem in the framework of multirate multicast transmission. The objective is to deliver a multicast such that maximum social welfare is achieved. The problem can be expressed as follows:

$$P \quad : \quad \max \left(\sum_{i \in R} U_i - c(T(R)) \right) \quad (5.1)$$

subject to:

$$y(l) \leq C_l \quad \forall l \in T(R) \quad (5.2)$$

$$y(l) \leq y(p(l)) \quad \forall l \in T(R) \text{ and } p(l) \neq \phi \quad (5.3)$$

¹The aggregate rate is achieved by joining the different groups of layers.

Figure 5.2: An example of a source-based multirate multicast tree.

$$y(l) \in Y \quad (5.4)$$

where $p(l)$ is the parent link of link l and $y(l)$ is the aggregate rate at any link l .

Equation 5.2 requires that the aggregate rate at any link l does not exceed the available capacity C_l . According to Equation 5.3, the aggregate rate of the parent link should be greater than or equal to the rates of the child links. Equation 5.4 restricts the aggregate rate at any link to one chosen from the set of possible rates.

5.3 Marginal Cost Solution

This section presents a distributed algorithm for realizing a marginal cost mechanism in the layered (multirate) multicast transmission. The complete mechanism is divided into two phases. In the first phase, an algorithm is provided for computing the maximum social welfare at each node, while in the second phase an algorithm for computing the marginal cost is provided.

1. In the first phase of the mechanism, a bottom-up traversal of the tree is done for computing the maximum social welfare achieved by all the nodes. The welfare achieved by a node n is given by a vector $W(n)$, a k -component vector such that $W_j(n)$ represents the maximum welfare achieved by a node n for the j^{th} group of layers. The welfare at any node n depends on the sum of users' willingness to pay minus the cost of the subtree located at node n (including the one which connects the node n to its parent), where all the users are located at node n or downstream. Therefore, for the multirate multicast transmission, the welfare achieved by the leaf nodes (receivers) can be computed by using the following formula:

$$W_j(n) = \max \left(([b_n]_j - [c_{p(n),n}]_j), W_{j-1}(n) \right).$$

It should be noted that the welfare achieved at no service is equal to zero, i.e., $W_0(n) = 0$.

Algorithm 7 :Bottom-up traversal of the tree for computing welfare values at each node of T .

- 1: Each leaf node n of the multicast tree computes the welfare $W(n)$, for each layer j , $0 \leq j \leq k$, by using the formula:

$$W_j(n) = \max \left(([b_n]_j - [c_{p(n),n}]_j), W_{j-1}(n) \right)$$

and sends it to its parent node $p(n)$.

- 2: After receiving the welfare vector from all its child nodes. Node n calculates its own welfare for each available group of layers by using the formula:

$$W_j(n) = \max \left(\left(\sum_{n' \in Ch(n)} W_j(n') - [c_{p(n),n}]_j \right), W_{j-1}(n) \right)$$

and forward the welfare vector to its parent $p(n)$.

The welfare at the internal node of the multicast tree depends on the total welfare achieved by the child nodes minus the cost of the parent link, i.e.,

$$W_j(n) = \max \left(\left(\sum_{n_i \in Ch(n)} W_j(n_i) - [c_{p(n),n}]_j \right), W_{j-1}(n) \right).$$

The idea behind using the above-mentioned welfare formulas is based on the fact that node n can achieve the maximum welfare from its parent for all the rates greater than its actual maximum welfare rate.

Each node computes the welfare vector and forwards it to the parent node until the source node is reached. The welfare vector at the source node gives the information regarding the maximum welfare achieved for each group of layers.

2. In the second phase, the aim is to find a final set of receivers with their cost shares and rate allocations. As discussed earlier in Section 3.5.2, a marginal cost mechanism gives each user a welfare equal to its marginal contribution to the overall welfare. Thus, to compute the cost share of the receiver i , it is necessary to find out the maximum network welfare when receiver i is not admitted. This information can easily be attained by the bottom-up traversal of the tree without

admitting receiver i . However, this approach is not communication efficient as it requires a bottom-up traversal of the tree for each receiver i . Therefore, a message efficient way is required for computing the cost shares of all the users. An algorithm is described for computing the cost share and service allocation by using a single downward traversal of the tree.

As discussed, the cost share of a receiver $i \in R$ depends on the information regarding the maximum welfare achieved with or without receiver i . However, the question is how efficiently this information can be provided to the receiver i . Maximum welfare computed during the bottom-up traversal of the tree can be easily passed back down the tree to inform each receiver about the maximum welfare achieved for the initial set of users. Moreover, the information regarding the configuration of each receiver can also be combined to achieve the maximum welfare. That is, the root node informs its children of which configuration they are in; this information combined with the stored information is sufficient for them to determine the configuration of their children. This process repeats until every node knows which configuration it must be in to achieve the maximum welfare. In other words, a top-down traversal of the tree is needed to inform each receiver about the maximum welfare and rate allocation. The most difficult part, however is to compute the maximum welfare without admitting user i .

A vector $A(n)$ is used to denote the maximum welfare without node n . It is a k -component vector such that $A_j(n)$ is the maximum possible welfare achieved by the source for the j^{th} group of layers without admitting node n . To compute this in the layered paradigm, it is assumed that every node n stores the welfare vector $W(n_i)$ from all its children $n_i \in Ch(n)$ learned during the bottom-up traversal of the tree.

In the downward traversal of the tree, each node n computes the welfare vector $A(n_i) \forall n_i \in Ch(n)$, subject to the following conditions:

- If node n_i is an internal node of the multicast tree, then $A_j(n_i)$ is the maximum network welfare without considering any profit from the subtree rooted at node n_i .

Algorithm 8 :Top down traversal of the tree T_{R_m} .

- 1: Source node S sends $A_j(n_i)$ to all its children n_i , for $0 \leq j \leq k$, computed by using the formula:

$$A_j(n_i) = \left(\sum_{n'_i \in Ch(s); n_i \neq n'_i} W_k(n'_i) - [c_{s,n_i}]_j \right)$$

- 2: Every node n , on receiving the vector $A(n)$ from its parent, computes $A(n_i)$ for each of its children $n_i \in Ch(n), n_i \notin R$, $0 \leq j \leq k$ and sends its value to n_i by using the following formula:

$$A_j(n_i) = \max_{t=j}^k \left(\left(A_t(n) + \sum_{n'_i \in Ch(n); n_i \neq n'_i} W_t(n'_i) \right) - [c_{n,n_i}]_j \right)$$

- 3: Every node n , on receiving the vector $A(n)$ from its parent, computes $A(n_i)$ for each of its children $n_i \in Ch(n), n_i \in R$, $0 \leq j \leq k$ and sends its value to n_i by using the following formula:

$$A_j(n_i) = \max_{t=j}^k \left(A_t(n) + \sum_{n'_i \in Ch(n); n_i \neq n'_i} W_t(n'_i) \right)$$

- If node n_i is a leaf node, no profit from the receiver n_i is included (that is, without admitting user n_i) in the network welfare.

This idea is expressed in Algorithm 8. Once the receiver n_i knows the maximum welfare without admitting itself, it can easily compute the marginal welfare and the cost share for receiving the multirate multicast transmission.

Theorem 11 *The welfare computed by the Algorithm 8 is the maximum welfare achieved by the source without admitting receiver n_i .*

Proof: The induction is used on the multirate multicast tree height h_n to prove that $A_j(n)$ is the maximum welfare without admitting user n . For the base case, node n with height one is considered, i.e., the child of the source node. According to

the assumption, the maximum welfare obtained by providing the multicast up to the internal node n is the sum of the maximum welfare obtainable in the subtrees rooted at the other children of the source minus the cost of sending the required layers from the source to node n . Therefore, Algorithm 8 computes the correct value of $A_j(n)$ for any n that is a child of the source node.

For the inductive step, it is assumed that node n at height h_n receives the values of the vector $A(n)$ from its parent $p(n)$ and shows that this implies that for any $n_i \in Ch(n)$, n computes the correct values of $A(n_i)$. There are two possible cases. In the first case, n_i is considered to be the leaf node, i.e., $n_i \in R$, while in the second case n_i is considered as an internal node.

In the case where n_i is the leaf node, the maximum welfare vector without admitting n_i depends on the maximum welfare obtained by the other users of the multicast tree. It is assumed that parent $p(n)$ informs the correct value of $A(n)$ to node n . That is, node n has knowledge about the welfare obtained for different layers. Moreover, node n has already learned² the values of maximum welfare achieved by the subtree rooted at other children of node n . Therefore, the formula $A_j(n_i) = \max_{t=j}^k \left(A_t(n) + \sum_{n'_i \in Ch(n); n_i \neq n'_i} W_t(n'_i) \right)$ correctly computes the maximum welfare without providing any layer to the receiver (leaf node) n_i .

In the second case, n_i is considered to be the internal node. The proof is similar; however, for the internal node, it is necessary to subtract the cost of sending the required layers from n to n_i . Therefore, $A(n_i)$ is correct by induction. $\square$

Lemma 6 *The maximum welfare computed with receiver n_i is always greater than or equal to the welfare achieved without admitting user n_i .*

Proof: This lemma can be proved by contradiction. It is assumed that the welfare achieved without admitting receiver n_i is greater than that with receiver n_i , i.e., $\sum_{i \in R} u_i - c(T(R)) \leq \sum_{i \in (R - n_i)} u_i - c(T(R - n_i))$. The welfare with receiver n_i can be rewritten as follows:

²During the bottom-up traversal of the tree

$$\begin{aligned}
NW &= \sum_{i \in (R)} (u_i - c(T(R))) \\
&= \sum_{i \in (R-n_i)} (u_i - c(T(R - n_i))) + (u_{n_i} - c(T(n_i))) \\
&< \sum_{i \in (R-n_i)} (u_{n_i} - c(T(R - n_i))).
\end{aligned}$$

The above statement holds only if $u_r - c(T(r))$ is negative. That is, r is not part of the efficient set. Thus, the welfare computed without user i is always less than or equal to that with receiver r . $\square$

Theorem 12 *Algorithms 7 and 8 are optimal with respect to the number of messages sent for the computation of cost shares and service allocations of the final set of receivers.*

Proof: Implementation of the marginal cost pricing mechanism in multirate multicast transmission is based on the maximum welfare achieved by each node $n \in V(R_m)$ of the multicast group. This maximum welfare can be easily computed by a single bottom-up traversal of the tree, i.e., one message per link. However, it should be noted that the cost and service allocations of the final set of receivers requires only a single top-down traversal of the tree. Therefore, the mechanism implementation takes only two messages per link, i.e., one bottom-up and one top-down traversal of the tree, and is message efficient. $\square$

5.4 Summary

In this chapter, the problem of pricing multirate multicast transmission under a layered paradigm is considered. The goal is to achieve the maximum social welfare from the multirate multicast transmission. The focuss is on the marginal cost pricing mechanism and it is shown that the obtained solution to marginal cost pricing mechanism

is communication efficient and requires only two messages per link for the cost and service allocations to the set of users.

Part II

Multicast Pricing of Shared Tree Multicast

Chapter 6

Multiparty Communication

6.1 Introduction

This chapter considers the problem of pricing *multiparty communication or conferencing*. Multi-party communication can involve multiple senders and a large group of receivers who together form a multicast group. Audio/video-conferencing and discussion groups are a few basic examples of multiparty communication. They are also the primary applications to benefit from multicast.

Agarwal and Paul [2] and Grossglauser and Ramakrishna [29] show the requirement of *shared tree multicast* for realizing multiparty communication. In shared tree multicast, a single tree is built and shared by all the senders for multicasting the data. Shared trees have a significant advantage over source-based trees [77]. Shared trees require only a single routing table for an entire group, instead of one per group as required by source based trees. Steiner trees and core based trees are the two good examples of shared trees.

Grossglauser and Ramakrishna [29] proposed a scalable multipoint-to-multipoint architecture for ATM, called SEAM. They used a shared tree, which is very close to the *core based trees*, while the work of Agarwal and Paul [2] is based on *Steiner trees*.

Steiner trees, however, are not a good solution for pricing the multicast transmission¹.

In this chapter, a core based tree is used for constructing the shared multicast tree. A node C is selected as a core and a shortest path tree rooted at this node is constructed. From the pricing perspective of multiparty communication, it can be assumed that all the senders are virtually shifted to one node, i.e., core node. The core node multicasts the data to the set of receivers. In multiparty conferencing, group members can act as senders only, receivers only or both senders and receivers.

The remainder of this chapter is structured as follows. Section 6.2 introduces the distribution model used in multiparty communication, Section 6.3 discusses the network model and Section 6.4 describes pricing in multiparty communication.

6.2 Distribution Model

This section presents *distribution models* used for multiparty communication. Singh et al. [72] defines a large number of distribution models that are based on the topology and the type of application. These models are *centralized*, *full mesh*, *multicast*, *unicast receive and multicast send*, and *endpoint mixing*. In this chapter, the unicast send and multicast receive distribution model is chosen due to its scalability over other models.

In unicast send and multicast receive, the core node acts like a conferencing server for the multiparty communication. In multiparty communication, the conferencing server is basically responsible for mixing the data under a pre-defined policy and distributing it to different sets of receivers. However, the aim here is to study the distribution and charging of multiparty conference; therefore there will not be any focus on how conferencing server mixes media streams.

¹Feigenbaum et al. [19] applied the marginal cost mechanism in Steiner trees and found that there is no unique largest efficient set that maximizes the network. Moreover, they found that there is no feasible algorithm that can implement the marginal cost mechanism. They showed that computing welfare $W()$ is equivalent to the net-worth maximization version of the *prize collecting Steiner tree* problem and is NP-Hard to approximate within any constant factor, even if the graph $G = (V, E)$ is a bounded degree graph.

Figure 6.1: An example of the unicast receive and the multicast send scheme.

6.2.1 Unicast Receive and Multicast Send

In this scheme, senders send their media streams using unicast to the core node. The server mixes them and sends them on the shared multicast tree. In this way, all the members of the multicast group receive the mixed stream. Senders also receive their own stream until they filter it out.

Figure 6.1 presents an example of multiparty communication, a multiparty application with two senders S_1 and S_2 . Each sender uses unicast to send the data to the core node C , which mixes it and multicasts it on the shared tree. This scheme

is good for a large number of receivers in comparison to a small number of senders. This scheme can also be used for realizing a single-source multicast.

6.3 Network Model

In the context of multiparty communication, consider a network $G = \{V, E\}$, where V is the set of nodes and E is the set of bidirectional links. A user population P resides on the set of network nodes.

Consider a multiparty conferencing where $S = \{S_1 \dots, S_N\}$ is the set of senders and R is the set of receivers. A shared² multicast tree is built connecting all the members to the core of the multicast tree. This multicast tree is defined as the union of the paths from all members (pure sender, pure receiver, or both) to the core of the group. Hence, the cost function is submodular and nonnegative function of the set of receivers.

As mentioned previously, core C of the shared multicast tree acts as a conferencing server. However, there is a price c_ε involved for enabling the core node as a conferencing server³. The cost of sending a message at any link $l \in E$ is denoted by c_l . Each node n of the shared multicast tree knows the price c_l of each link l incident on it. It is assumed that the links are of infinite capacity and the sender/conferencing server can transmit the data at the maximum rate.

Each receiver $i \in R$ shows their willingness to pay through a bid, denoted by b_i , which is the amount that receiver i wants to pay for receiving the transmission. Let T be the shared multicast tree connecting all the receivers. The set of receivers located at any node n is given by $res(n)$ while $Ch(n) \in V$ denotes the children of node n . In order to realize the pricing mechanism, one must understand the concept of welfare $W(n)$ at any node n . If c_p denotes the cost of the parent link of node n , then the social welfare at node n is given by the following formula:

²The presence of multiple senders requires the use of a single shared tree instead of multiple source rooted trees.

³For enabling the conferencing functionality of the core node.

$$W(n) = \max \left(\sum_{i \in \text{res}(n)} b_i + \sum_{n' \in \text{Ch}(n)} W(n') - (c_p - 1), 0 \right) \quad (6.1)$$

In the next section, this welfare function is used for the cost and the service allocation to the set of receivers.

6.4 Pricing Multiparty Communication

In this section, an efficient distributed algorithm for cost allocation in multiparty communication is discussed. The communication model is based on unicast receive and multicast send. Therefore, the charging scheme has two phases. In the first phase, all the senders of the multi-party communication are charged; in the second phase, the marginal-cost mechanism is applied to select the set of users and allocate the cost for receiving the multi-party communication. This cost-allocation is based on the willingness of users to pay.

6.4.1 Charging Senders

In the first phase, all the senders are charged for unicasting the data to the core node and for enabling the core node to act as a conferencing server.

The pricing policy for charging senders is quite straight forward. Each sender S_i is charged the fixed cost $c_u(i)$ for transmitting the data to the core node. This cost is equal to the sum of the costs of the links in the path from the sender to the core, and is fixed⁴ and different for all senders. It is given by

$$c_u(i) = \sum_{l \in T(i)} c_l .$$

⁴All the resources are used by a sender to transmit the data to core node. Therefore, the sender has to bear the total cost of transmission and it cannot be distributed among other users of the network.

Where $T(i)$ is the shortest path from the sender i to the core node.

The cost of enabling the core node has to be equally distributed among all the senders. If c_ε is the cost of enabling the core node and N is the number of senders, then the cost share of any sender i is given by:

$$c_\varepsilon(i) = \frac{c_\varepsilon}{N} .$$

The question of how to charge different users of the multicast group is now addressed. A group member of any multiparty communication can act as a pure sender, a pure receiver, or both sender and receiver. If x_i denotes the amount paid by any member i , then the following strategy can be adopted for charging the members of the multiparty communication:

1. **Pure Sender:** The cost of any member i participating as a pure sender is equal to the sum of the cost of unicast transmission and the cost share for enabling the core node, i.e.,

$$x_i = c_u(i) + c_\varepsilon(i).$$

2. **Both Sender and Receiver:** If members act as both senders as well as a receivers, they must pay the cost equal to the sum of unicast cost, the enabling cost share, and the cost of receiving the multicast transmission (see Section 6.4.2 for computing the multicast cost $c_m(i)$). Hence, the cost x_i is given by:

$$x_i = c_u(i) + c_\varepsilon(i) + c_m(i).$$

3. **Pure Receiver:** As a pure receiver, a member needs to pay only the multicast cost, i.e.,

$$x_i = c_m(i).$$

The next section looks at the way to find the multicast cost that a receiver must pay for receiving the transmission.

Algorithm 9 :Bottom-Up traversal for computing welfare values at each node n of T .

```

1: for each node  $n \in V(R_m)$  do
2:   After receiving the welfare vector  $W(Ch(n))$  from all of its children,
3:   for each child  $n_i \in Ch(n)$  do
4:     if ( $W(n_i) \geq 1$ ) then
5:        $W(n_i) = W(n_i) - 1$ 
6:     else
7:        $W(n_i) = 0$ 
8:     end if
9:   end for
10:  Node  $n$  computes the welfare  $W(n)$  by using the formula
      
$$W(n) = \max \left( 0, \sum_{i \in res(n)} [b_i] + \sum_{n_i \in Ch(n)} W(n_i) - ([c_p] - 1) \right)$$

11:  Node  $n$  sends the welfare vector  $W(n)$  to its parent  $p(n)$ .
12: end for
13: Core node  $C$  returns the value  $W(C)$ .
```

6.4.2 Marginal Cost Mechanism for Charging Receivers

In this section, a marginal cost algorithm for cost distribution in multicast transmission is presented. The mechanism requires one bottom-up and one top down traversal of the shared multicast tree for computing the cost share. It is assumed that there is no bound on the link capacity, i.e., the sender/server can transmit the data at the maximum rate.

Each receiver $i \in R$ (including senders of the multicast group) bids b_i for receiving the multiparty communication. The welfare $W(n)$ is calculated at each node n during the bottom up traversal of the shared multicast tree. A node gathers b_i from each $i \in res(n)$, $W(n')$ from each $n' \in Ch(n)$ and computes the welfare by using Equation 6.1. A node forwards the resultant welfare to its parent and so on. Finally, the total welfare reaches to the core node.

In the top-down traversal of the tree, the final set of receivers and the corresponding price that a receiver needs to pay for receiving the multicast transmission is calculated. For calculating the service and cost allocation, the ideas mentioned in

Algorithm 10 :Top down traversal of the tree T_{R_m} .

```

1: Core node  $C$  sends  $A = W(C) - 1$  to each of its children.
2: for each node  $n \in \{V(R_m) - C\}$  do
3:   After receiving a message  $A$  from parent  $p(n)$ 
4:   if ( $W(n) < 1$  or  $A < 0$ ) then
5:     for all users  $i \in res(n)$  do
6:       set  $x_i \leftarrow 0$  and  $q_i \leftarrow 0$ 
7:       send  $-1$  to  $\beta$  for all  $\beta \in Ch(n)$ 
8:     end for
9:   else
10:     $W(n) = W(n) - 1$ 
11:     $A \leftarrow \min(A, W(n))$ 
12:    for each  $i \in res(n)$  do
13:      if ( $b_i \leq A$ ) then
14:         $x_i \leftarrow 0$ 
15:      else
16:         $x_i \leftarrow b_i - A$ 
17:      end if
18:    end for
19:    for each  $\alpha \in Ch(n)$  do
20:      Send  $A$  to  $\alpha$ 
21:    end for
22:  end if
23: end for

```

the single rate transmission (see Section 4.3.1) are used.

Theorem 13 *The multiparty pricing scheme correctly computes the amount charged from all the members of the multiparty communication. Furthermore, it is communication efficient.*

Proof: First, it is shown that the multiparty pricing scheme, discussed in Section 6.4, correctly computes the amount charged from each member, and then shown that it is communication efficient.

In the first phase, all the senders are charged for unicasting the data to core node and for enabling the core node to act as a conferencing server. This cost can be easily

computed by using two messages per link on the shortest path from each sender to the core node, that is, starting from the sender and reaching up to the core node to compute the unicast cost of each sender. It is assumed that the core node knows the number of senders in the multiparty conferencing, and therefore, it can easily compute the cost share of each sender for enabling the core node. This cost can be informed to each sender by using another message on the links of the shortest path from each sender to the core node.

In the second phase of the pricing scheme, the marginal cost of each receiver for receiving the multiparty communication is computed. This can be done (see Lemma 2) using only two messages per link (see Theorem 6) on the shared multicast tree rooted at the core node, that is, one bottom-up traversal for the computation of welfare at each node, and one top-down traversal for computing the final set of receivers and the corresponding cost shares.

Therefore, it can be inferred that the multiparty pricing scheme is communication efficient and requires at most four messages per link of the shared multicast tree. $\square$

Remark: If the welfare at any node (from the sender up to the core node) becomes equal to zero, then the sender acts only as a receiver. This is because multicast transmission is not provided to the set of members with zero welfare (during the top-down traversal of the tree). Therefore, if any node wants to take the role of both sender and receiver, it must have welfare greater than zero on all nodes up to the core node of the shared multicast tree.

All the variations of the source rate selection methods presented in Chapter 4 can be applied to the core-based multiparty communication. However, the rate of transmission of all senders should be at least equal to the multicast rate of the core node. The budget balanced Shapley value mechanism can also be applied for pricing of multiparty communication, but it may not be effective due to its low communication efficiency.

6.5 Summary

In this chapter, the problem of pricing multiparty communication is considered. Multiparty communication involves multiple senders and receivers, and therefore, the use of shared multicast trees is essential for realizing the multiparty communication. The main focus of this chapter is to show how a communication efficient method can be developed to charge the different members of the multicast group in the multiparty communication.

A core based pricing scheme is developed due to the presence of multiple senders. In this core based pricing scheme, all members of the multicast group are charged with respect to the core node. It is shown that core based pricing is communication efficient and, in the worst case, requires only four messages per link to charge all members of the multiparty communication.

Chapter 7

Single Source Multicast Using Shared Trees

This chapter considers the problem of pricing single source multicast using shared trees. Two routing schemes, in Sections 7.1 and 7.2, are discussed. The main focus in this chapter is to develop a pricing method for core based routing.

7.1 Unicast Receive and Multicast Send

This section examines the pricing of single source multicast under unicast receive and multicast send schemes. As discussed in Section 6.2.1, the source unicasts the data to the core node, which is responsible for multicasting it to the set of receivers. This scenario is quite similar to that of source rooted trees, except that in this case multicasting takes place on the shortest path tree rooted at the core node instead of the source node. In other words, the source is virtually shifted to the core node and the core multicasts the data to the set of receivers.

The pricing scenario in single source multicast using shared trees is quite similar to that of the source rooted tree, except that in a shared tree multicast, the source has to pay an amount equal to its unicast cost up to the core node. Hence, the marginal

Figure 7.1: An example of a unicast receive and multicast send scheme in single-source multicast.

cost algorithm can be used for cost and service allocation for the set of receivers R .

Figure 7.1 shows a scenario for single source multicast with two sources S_1 and S_2 . These sources unicast the data to the core node and then the core node forwards it to the set of receivers; that is, the core node acts as a forwarding node instead of a conferencing server for the multiparty communication.

In the next section, a pricing mechanism for core based multicast routing is discussed.

7.2 Core Based Multicast

This section first provides a description of a network model and core based multicast routing in the network, then proposes a charging mechanism for core based multicast.

7.2.1 Network Model

Consider a graph $G = (V, E)$, where V is the set of nodes and E is the set of bidirectional edges. A multicast group m is characterized by the tuple $\{C_m, T_m, S_m, R_m\}$ where C_m is the core node of the group; T_m is the shared multicast tree rooted at the core node; S_m is the sender of the group; and R_m is the set of receivers in the multicast group m . Tree T_m is the union of the paths $T(i)$, where $T(i)$ is a fixed path from user i to the core node C_m . Each node knows the cost of edges incident on it. Moreover, it is assumed that the links are of infinite capacity and each sender can transmit the data at the maximum rate.

Figure 7.2 presents an example of core based routing. When the sender S (that is, a group member) wants to send the data to different group members, the packet first reaches to the sender's local router¹, which forwards the packet to all its edges that are in the multicast tree. Similarly, each node in the multicast tree also forwards the packet to all its edges except the one from which the packet has come. Therefore, packets in the core based tree flow up the tree towards the core node and down the tree away from the core node, depending on the location of the source node [31].

7.2.2 Pricing in Core Based Multicast

Pricing in core based multicast is not as simple as that in the unicast send and multicast receive scheme. In the core based tree, the sender has to route the data (packets) on the shared multicast tree rooted at the core node. Path traversal for reaching a different set of receivers from the source node is highly dependent on the

¹A node in which a sender resides.

Figure 7.2: An example of core-based routing in single-source multicast.

receiver's location. For example, in Figure 7.2, the source S_m can send the data packets directly to the set of receivers $\{R_1, R_2, R_3\}$ without routing through a core node. However, this is not the case with the receiver set $\{R_4, R_5\}$. In other words, all the receivers located in the subtree rooted at the core node get their data through the core node.

A core based shared multicast transmission can be exhibited by Figure 7.3. Here a shared multicast tree T_m is rooted at the core node C_m with a sender S_m . N_1 and N_2 are the intermediate nodes, i.e., the nodes which lie on the shortest path from the core node to a sender. The sender uses these intermediate nodes to route the data up to the core node. Hence, the presence of all these intermediate nodes is necessary for realizing core based routing, and a pricing scheme is useful only if it can guarantee the presence of all the intermediate nodes.

Figure 7.3: A core-based multicast tree in single-source multicast.

In the proposed pricing scheme described in the next subsection, the core node acts as a charging point. Both the sender and receivers of the single source multicast are charged.

7.2.2.1 Charging Sender/Receivers

In core based multicast routing, the sender is basically responsible for sending the data up to the core node. This transmission requires certain resources and is likely to incur a significant charge. Therefore, it is fair to bill the sender for using the resources. However, the question is How much should a sender be charged for sending the data up to the core node? One possibility is to charge an amount equal to the unicast cost, but this is not an adequate solution since all the intermediate links are simultaneously used by other receivers to receive the transmission (see Section 7.2). Moreover, it is

necessary to charge sender in a such way that the presence of all the intermediate nodes is guaranteed.

The required features can be achieved by applying the marginal cost mechanism (see Section 6.4.2) on the shared tree rooted at the core node. Each member (sender or receiver) of the multicast group shows their willingness to pay by bidding for resources. However, senders' willingness to pay should never be less than the unicast cost of the transmission to guarantee that the welfare achieved at all the intermediate nodes is always greater than zero. That is, if the welfare at any of the intermediate nodes becomes zero, the set of users (including the intermediate nodes) must be rooted out from the final set of receivers, core based multicasting cannot be realized. However, by bidding an amount equal to the unicast cost, the sender can assure the presence of all the intermediates nodes.

Hence, in contrast to the source based pricing mechanism, the core based pricing mechanism charges the senders for multicasting the data to the set of receivers.

Lemma 7 *The core based multicast pricing scheme correctly computes the marginal cost share of all the members of the multicast group.*

Proof: The proof that the core based multicast pricing scheme computes the marginal cost of all the members of the multicast group is based on the computation of network welfare achieved at each node. Where h denotes the height of the shared multicast tree from the core node, induction is used on height $h_n \leq h$ of node n of the shared multicast tree to show that it is feasible to compute the network welfare at each node n of the multicast tree.

For the base case, consider a leaf node n of the multicast tree T_m . The welfare obtained at the leaf node depends on the user's bid and the cost of the parent link. All the user bids and the cost of the parent link are available at the leaf node. Therefore, it can easily compute the welfare value and pass it to the parent node.

For the inductive step, it is assumed that node n , at height $h_n < h$, receives the welfare from all its children and shown that welfare $W(n)$ is passed to the parent

$p(n)$. The correctness of the computed welfare follows from the fact that node n has information about b_i for all $i \in res(n)$ and the cost of the parent link. Since it is assumed that node n knows the values of welfare $W(n')$, for all $n' \in Ch(n)$, it can easily compute the welfare vector and send it to its parent $p(n)$. That is, each node can compute the total welfare achieved by all the users located on the subtree rooted at node n .

The correctness of the marginal cost computation is implied from the Lemma 2. $\square$

Theorem 14 *Core based multicast pricing is optimal with respect to the number of messages sent for the computation of the final set of receivers and the cost share of all the members of the multicast group.*

Proof: Core based multicast pricing requires the welfare achieved by each node $n \in V(R_m)$ to compute the cost share of all the members (both the source and receivers) of the multicast group. This welfare computation requires a single bottom-up traversal of the tree, that is, one message per link of the multicast tree. The computation of the final set of users with their corresponding cost share can be combined and computed using only one top-down traversal of the tree, that is, one message per link of the shared multicast tree. Therefore, core based multicast pricing needs only two messages per link (one bottom-up and one top-down) in total and is optimal with respect to the number of messages sent for the computation of the cost share of all the group members.

7.3 Summary

In this chapter, the problem of pricing single source multicast using shared trees is considered. Two distribution schemes are discussed: unicast receive and multicast send, and core based routing.

The main focus of this chapter is pricing in core based routing. A marginal cost pricing algorithm is proposed for core based routing in which the source also needs

to bid for utilizing the resources. However, this bid should be at least equal to the unicast cost from the source to the core node to guarantee the presence of all the intermediate nodes. In single source core based pricing, all receivers and the source of the multicast group are charged with respect to the core node. It is also shown that core based multicast pricing is optimal with respect to the number of messages sent for the computation of the final set of receivers and the cost share of all the members of the multicast group.

Chapter 8

Conclusions

The focus of this thesis is to solve the problem of multicast pricing using mechanism design. In multicast, each link of the multicast tree is shared by a number of downstream users. Hence, multicast cannot be priced according to unicast policy, and innovative solutions are required to distribute the multicast cost among the set of participating receivers.

We consider that the cost sharing of multicast transmission is dependent on users' willingness to pay. This complicates the issue of pricing further as selfish users do not want to report their true willingness to pay, but instead want to report a value that maximizes their own welfare. A “strategyproof” cost sharing mechanism is used to cope with this problem. Two strategyproof cost sharing mechanisms are presented, namely “marginal cost” and “Shapley value”. However, the thesis is mainly restricted to the marginal cost mechanism due to its low communication complexity.

This thesis is divided into two parts. In the first part, the problem of pricing in a source rooted multicast transmission is studied; in the second part, the problem of pricing in shared tree multicast is studied. The main focus in both parts is been on determining whether or not these mechanisms can be feasibly implemented on different network models presented in this thesis. The communication complexity of each mechanism is used as a major criteria to determine their feasibility.

In the first part, source rooted multicast cost sharing mechanisms for *single rate* and *multirate* transmission are discussed. Feigenbaum et al. [19] were the first to study the problem of pricing in multicast transmission. However, they assume that the links are of infinite capacity and the source is free to multicast the data at the maximum rate. This work considers a more realistic situation and develops a distributed and scalable marginal cost algorithms for pricing single rate and multirate multicast transmissions.

In single rate transmission, it is assumed that the source has a set of different transmission rates and needs to make source rate selections as per the information provided by the set of receivers. This is not a trivial task as the receivers are distributed throughout the network and have different preferences over different available rates. Three criteria of the source rate selection in the single rate transmission model are considered:

- to maximize social welfare;
- to maximize the size of efficient set; and
- to maximize the individual utility of the maximum number of users.

It is shown that the marginal cost algorithms for pricing the final set of receivers under the single rate transmission model are communication efficient and can be easily implemented with the communication complexity of only two messages per link, that is, one bottom-up and one top down traversal of the multicast tree. It is also shown that it is difficult to make any choice regarding the source rate with the Shapley value pricing mechanism. Moreover, Shapley value mechanism is communication inefficient.

In multirate multicast transmission, the problem of maximizing social welfare is considered. In multirate multicast transmission, different receivers can receive the service at different rates instead of the same rate. For solving this problem, we present a marginal cost algorithm that is communication efficient and requires only two messages per link for the computation of cost and service allocation to the set of receivers.

In the second part, pricing in shared tree multicast is considered. The scenario of multiparty communication and single source multicast for pricing in shared tree multicast is considered, and a communication efficient pricing scheme for both scenarios is proposed. The effectiveness of this pricing scheme is that senders also need to pay for multicasting the data to the set of receivers. This amount is determined by the distribution model and the resources used by the senders. To the best of our knowledge nobody has previously applied mechanism design to solve the problem of pricing in shared multicast tree.

8.1 Future Work

This study leaves open issues that could be potential subjects for future work. These are described in the following subsections .

1. We consider that multicast transmission incurs network cost per link. However, today's service providers do not charge per link for the transmission. Therefore, it would be of interest to determine how the results of the mechanism would change with the change in the network model.
2. In single rate transmission, it would be interesting to determine whether the source can make any sensible choice or not, if it has the information regarding the welfare achieved, the number of users in the final set and the number of users with maximum utility for each possible rate. That is, What should the possible preferences of the source over the above-mentioned possible criteria be?
3. In single rate transmission, we consider only maximizing social welfare, maximizing the size of the final set of receivers, maximizing individual utility of the maximum number of users, and budget balance as the goal of a multicast cost sharing mechanism. However, profit maximization is also an important goal. Therefore, the current work could be extended to include this new goal and test whether a "profit maximization mechanism" would be strategyproof or not.

4. Is there any budget balanced mechanism for source rate selection in the single rate transmission model?
5. The effect of relaxation in the basic requirements could be analyzed; for instances, can an efficient and strategyproof mechanism be found that is more easily implementable than marginal cost?
6. How would the communication complexity of the discussed mechanism change with the bounds in message size?
7. In multiparty communication, we consider only a pricing mechanism for single rate transmission. However, it would be interesting to know whether a multirate marginal cost algorithm for maximizing social welfare in multiparty communication can be realized. If so, then what assumptions regarding different senders rate would have to be considered?. In general, how are receiver rates connected to senders rates.

Bibliography

- [1] M. Adler and D. Rubenstein. Pricing multicasting in more practical networks. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium On Discrete Algorithms*, pages 981-990, 2002.
- [2] S. Agarwal and S. Paul. A flexible protocol architecture for multi-party conferencing. In *Proceedings of IEEE ICCCN*, pages 81-91, 1996.
- [3] T. Ballardie, P. Francis, and J. Crowcroft. An architecture for scalable inter-domain multicast routing. In *Proceedings of ACM SIGCOMM*, pages 85–95, 1993.
- [4] K. L. Calvert, E. W. Zegura, and M. J. Donahoo. Core selection methods for multicast routing. In *Proceedings of IEEE ICCCN*, pages 638-642, 1995.
- [5] G. Carle, G. Fankhauser, and B. Stiller. Charging for ATM-based IP multicast services. In *Proceedings of the 5th Annual Network+Interop Engineers Conference*, pages 149-154, 1998.
- [6] R. C. Chalmers and K. C. Almeroth. Developing a multicast metric. In *Proceedings of IEEE Global Internet (GLOBECOM'00)*, 2000.
- [7] D. D. Clark. Combining sender and receiver payments in the Internet. In *Proceedings of the 24th Telecommunications Policy Research Conference*, 1996.
- [8] E. H. Clarke. Multipart pricing of public goods. In *Public Choice*, pages 17-33, 1971.

- [9] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. A study of priority pricing in multiple service class networks. In *SIGCOMM*, pages 123-130, 1991.
- [10] P. Dasgupta, P. Hammond, and E. Maskin. The implementation of social choice rules. In *Review of Economics Studies*, vol. 46, 185-216, 1979.
- [11] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. In *Artificial Intelligence*, vol. 20, number 1, pages 63-109, 1983.
- [12] S. Deering. Host extensions for IP Multicasting. *RFC 1112*, vol. 17, 1989.
- [13] S. Deering. Multicast routing in a datagram internetwork. Ph.D. dissertation, Stanford University, 1991.
- [14] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs. In *ACM Transactions on Computer Systems*, vol. 8, no 2, pages 85-110, 1990.
- [15] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. G. Liu, and L. Wei. Protocol independent multicast (PIM): Sparse mode protocol specification. *Internet Draft*, 1994.
- [16] C. Diot, W. Dabbous, and J. Crowcroft. Multipoint communication: A survey of protocols, functions, and mechanisms. In *IEEE Journal on Selected Areas in Communication*, vol 15(3), pages 277-290, 1997.
- [17] A. A. Economides and J. A. Silvester. Multi-objective routing in integrated services networks: A game theory approach. In *Proceedings of INFOCOM'91*, pages 1220-1225, 1991.
- [18] D. F. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks. In *Proceedings of IEEE INFOCOM'89*, pages 110-118, 1989.
- [19] J. Fiegenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmission (preliminary version). In *Thirty-Second Annual ACM Symposium on Theory of Computing (STOC00)*, pages 218-227, 2000.

- [20] S. Fotiou and C. Siriopoulos. Modelling network economics: A survey. www.erudit.de/erudit/events/esit99/12619_p.pdf.
- [21] E. Friedman. Learnability in non-atomic externality games, with applications to computer networks. *Mimeo.*, Rutgers University, 1997.
- [22] E. J. Friedman and S. Shenker. Learning and implementation on the Internet. *Mimeo.*, Rutgers University, 1997.
- [23] D. Fudenberg and J. Tirole. *Game theory*. MIT press, 1991.
- [24] M. W. Garret and M. Vetterli. Joint source/channel coding of statictically multiplexed real time services on packet network. In *IEEE/ACM transactions on Networking*, vol. 1, number 1, pages 71-80, 1993.
- [25] M. Ghanbari. Two layering coding of video signals for VBR networks. In *IEEE Journal on Selected Areas in Communication*, vol. 7, number 5, pages 771-781, 1989.
- [26] A. V. Goldenberg and J. D. Hartline. Competitive auctions for multiple digital goods. In *Ninth European Symposium on Algorithms*, pages 416-427, 2001.
- [27] J. Green, E. Kihlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. In *Journal of Public Economics*, vol 6, pages 375-394, 1976.
- [28] T. Grooves. Incentives in teams. In *Econometrica*, 41, pages 617-613, 1973.
- [29] M. Grossglauser and K. K. Ramakrishnan. SEAM: Scalable and efficient ATM multicast. In *IEEE INFOCOM '97*, 1997.
- [30] A. Gupta, D. Stahl, and A. Whinston. Managing the Internet as an economic System. Technical Report, Department of Economics, University of Austin at Texas, 1994.
- [31] M. Handley and J. Crowcroft. Internet multicast today. In *The Internet Protocol Journal*, vol. 2, number 4, 1999.

- [32] T. Henderson. Internet economics - A literature review. Technical report, University College London, 2000.
- [33] T. Henderson and S. N. Bhatti. Protocol independent multicast pricing. In *Proceedings of NOSSDAV 2000*, pages 11-17, 2000.
- [34] S. Herzog, S. Shenker, and D. Estrin. Sharing the “cost” of multicast trees: An axiomatic analysis. In *IEEE/ACM Transactions on Networking*, vol. 5, number 6, pages 847-860, 1997.
- [35] H. W. Holbrook and D. R. Cheriton. IP Multicast Channels: EXPRESS support for large-scale single-source applications. In *SIGCOMM symposium on Communications Architectures and Protocols*, 1999.
- [36] M. T. Hsiao and A. A. Lazar. Optimal decentralized flow control of Markovian queueing networks with multiple controllers. In *Performance Evaluation*, vol. 13, number 3, pages 181-204, 1991.
- [37] L. Hurwicz. The design of mechanisms for resource allocation. In *American Economic Review*, vol 63, page 1-30, August 1973.
- [38] K. Jain and V. V. Vazirani. Applications of approximation algorithms to cooperative games. To appear in *Proc. 33rd ACM Symposium on the Theory of Computing*, ACM press, pages 364-372, 2001.
- [39] K. Jain and V. V. Vazirani. Group strategyproofness and no subsidy via LP-duality. <http://www.cs.huji.ac.il/~noam/econcs/jv.ps>, 1999.
- [40] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multirate multicast sessions. Technical Report TR 2000-21, Institute of Systems Research and University of Maryland, 2000.
- [41] P. Key, D. McAuley, P. Barham, and K. Laevens. Congestion pricing for congestion avoidance. Technical Report, Microsoft Research MSR-TR-99-15, 1999.

- [42] Y. A. Korilis and A. A. Lazar. On the existence of equilibria in noncooperative optimal flow control. In *Journal of the ACM*, vol 42(3), pages 584-613, May 1995.
- [43] Y. A. Korilis, A. A. Lazar, and A. Orda. The role of the manager in a non-cooperative network. In *Proceedings of IEEE INFOCOM'96*, pages 1285-1293, 1996.
- [44] A. A. Lazar, A. Orda, and D. E. Pendarakis. Virtual path bandwidth allocation in multi-user networks. In *IEEE/ACM Transactions on Networking*, vol. 5, number 6, 861-871, 1997.
- [45] N. Lineal. Game theoretic aspects of Computing. In *Handbook of Game Theory*, vol. 2, pages 1339-1395, Elsevier Science Publishers B.V, 1994.
- [46] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [47] J. K. Mackie-Mason and H. R. Varian. Some economics of the Internet. In *Networks, Infrastructure and the New Task for Regulation*, University of Michigan Press, pages 107-136, 1996.
- [48] E. Maskin. The theory of implementation in Nash Equilibrium, . In *Social goals and organization: Essays in Memory of Elish Pazner*, Cambridge University Press, pages 173-204, 1985.
- [49] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, volume 26, 4, pages 117-130, 1996.
- [50] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. To appear in *Economic Theory*. Available in preprint form at <http://www.aciri.org/shenker/cost.ps>.
- [51] J. Moy. OSPF version 2. *Internet RFC 1247*, 1991.

- [52] N. Nisan. Algorithms for selfish agents. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 1-15, March 1999.
- [53] N. Nisan and A. Ronen. Algorithm mechanism design. In *Proceedings of 31st ACM Symp. on Theory of Computing*, pages 129-140, 1999.
- [54] A. M. Odlyzko. The economics of the Internet: Utility, utilization, pricing, and quality of service. Technocal report, AT & T Labs-Research, 1998.
- [55] A. M. Odlyzko. Paris metro pricing for the Internet. In *Proceedings of the 2nd International Conference on Information and Computer Economics(ICE-99)* , 1999.
- [56] M. J. Osborne and A. Rubinstein. A course in game theory. *MIT press*, 1994.
- [57] C. H. Papadimitriou. Algorithms, games, and the Internet: A survey of algorithmic problems related to game theory and the Internet. In *STOC 2001 (extended abstract in ICALP 2001)*, 2001.
- [58] C. H. Papadimitriou. Computational aspects of organization theory. In *Proceedings of the 1996 European Symposium on Algorithms*, Springer LNCS, 1996.
- [59] C. H. Papadimitriou and M. Yannakakis. On the value of information in distributed decision-making (extended abstract). In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, pages 61-62, 1991.
- [60] C. H. Papadimitriou and M. Yannakakis. Linear programming without the matrix (extended abstract). In *Proceedings of the Twenty-fifth Annual ACM Symposium on the Theory of Computing.*, pages 121-129, San Diego, California, May 1993.
- [61] K. Ravindran and T. Gong. Cost analysis in multicast transport architectures in multiservice networks. In *IEEE/ACM Transactions on Networking*, vol. 6, number 1, pages 94-109, 1998.

- [62] K. Roberts. The characterization of implementable choice Rules. *Aggregation and Revelation of Preferences*, pages 321-348, 1979.
- [63] A. Ronen. Algorithms for rational agents. In *Proceedings of the 27th Annual Conference on Current Trends in Theory and Practice of Informatics*, pages 56-70, 2000.
- [64] D. Rubenstein, J. Kurose, and D. Towsley. The impact of multicast layering on network fairness. In *ACM SIGCOMM*, pages 27-38, 1999.
- [65] B. Sanders. An incentive compatible flow control algorithm for rate allocation in computer networks. In *IEEE Transactions on Computers*, volume 37, 1067-1072, number 9, 1998.
- [66] T. Sandholm. Distributed rational decision making. In the textbook *Multiagent systems: Modern introduction to distributed artificial intelligence*. MIT Press, pages 201-258, 1999.
- [67] S. Sarkar and L. Tassiulas. Layered bandwidth allocation for multicasting of hierarchically encoded sources. citeseer.nj.nec.com/297420.html.
- [68] N. Shacham. Multipoint communication by hierarchically encoded data. In *Proceedings of IEEE INFOCOM*, pages 2107-2114, 1992.
- [69] S. Shenker. Efficient network allocations with selfish users. In *Proceedings of Performance*, pages 279-285, 1990.
- [70] S. Shenker. Making greed work in networks: A game-theoretic analysis of switch service disciplines. In *IEEE TNWKG: IEEE/ACM Transactions on Networking* *IEEE Communications Society, IEEE Computer Society and the ACM with its Special Interest Group on Data Communication (SIGCOMM)*, *ACM Press*, vol. 3, 819-831, 1995.
- [71] S. Shenker, D. Clark, D. Estrin, and S. Herzog. Pricing in computer networks: Reshaping the research agenda. In *ACM Computer Communication Review*, vol. 26, number 2, pages 19-23, April 1996.

- [72] K. Singh, G. Nair, and H. Schulzrinne. Centralized conferencing using SIP. Columbia University, <http://citeseer.nj.nec.com/471602.html>.
- [73] A. Thyagarajan and S. Deering. Hierarchical distance-vector multicast routing for the MBone. In *Proceedings ACM SIGCOMM'95*, pages 60-66, 1995.
- [74] D. Waitzman, C. Patridge, and S. Deering. Distance vector multicast routing protocol. *RFC 1075, IETF*, 1988.
- [75] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *Proceedings of ICCCN '94*, 1994.
- [76] P. Winter. Steiner problems in networks: A survey. In *Networks*, vol. 17, number 2, pages 129-167, 1987.
- [77] D. Zappala and A. Fabbri. An evaluation of shared multicast trees with multiple active cores. In *IEEE International Conference on Networking, ICN'01*, 2001.