國立臺灣大學資訊管理研究所

博士論文

指導教授： 林永松 博士

多媒體網路群播演算法

Multicasting Algorithms in Multimedia Networks

研究生： 鄭旭成 撰

中華民國九十四年七月

Multicasting Algorithms in Multimedia Networks

by Hsu-Chen Cheng

A dissertation submitted to
the Graduate School of Information Management
of National Taiwan University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

July 2005

This dissertation is dedicated to my beloved Ariel.

論文摘要

多媒體網路群播演算法

鄭旭成
中華民國九十四年七月
指導教授：林永松　博士
國立台灣大學資訊管理研究所

隨著近年來傳輸與編碼技術的發展，有愈來愈多的多媒體應用被推出並廣泛使用，例如視訊會議與隨選視訊。大部份這些應用都需要大量的頻寬來同步傳輸多媒體資訊給許多的使用者，其中可能最有效率的方式就是透過群播網路來達到這個目標。在多媒體傳輸的環境中，由於使用者可能使用不同的傳輸技術與網際網路進行連接(例如撥接數據機、Cable modem 或者 xDSL…等)，並且不同使用者對於品質的需求亦不相同。因為網路頻寬與使用者需求之差異，如何透過良好的設計架構與傳輸機制，有效率的利用頻寬並達到服務的彈性是個重要的研究議題。

由於高階視訊編碼技術與視訊閘道器的推出，當不同的使用者向傳輸端要求不同的品質視訊時，傳輸端只要將滿足最高頻寬需求的視訊，透過單一群播樹傳輸進行傳輸即可，這種群播技術稱為階層編碼群播技術或者多重速率群播技術。單一速率最小成本群播樹的問題就是大家熟知支史坦那樹問題，這是一個 NP-Complete 的問題，而多重速率群播樹問題將是比史坦那樹更複雜的問題。

在此論文中，我們將針對單一速率與多重速率群播樹之路由問題進行探討，使用數學模型來描述此類網路規劃與網路運作問題，並使用拉格蘭日鬆弛法作為基礎，以最佳化的方式提出適合的演算法。本論文的研究內涵與成果簡述如下：

- 最小成本多重速率群播路由問題：同時考慮路由決策與多重群播技術下每個鏈結上所應傳輸之資料量，以求得最小成本傳輸群播樹。我們成功的將此問題以數學模型進行描述，並提出以最佳化為基礎的演算法。根據實驗的結果顯示，我們所提出的演算法較之前相關研究所提出之演算法優越。此外，我們亦針對在網路鏈結有容量限制下的多群組群播問題進行研究，並提出相關

的演算法。

- 單一速率與多重速率群播允入控制研究：傳統的演算法在計算是否允入群播群組時是以整個群組為單位，在此我們提出群組部分允入的作法來求得最大收入之群播路由指定與資源預留機制。所謂的部份允入是指在無法允入全部群播群組內之使用者時，系統會以最大收入為目標嘗試對於群組內的部份使用者提供服務，以充分使用網路資源並最大化系統營收。此外，我們也從長期收益最大化的觀點，提出及時性的群播允數控制機制，並針對允入控制決策時間與系統負載流量之間的關係進行研究。我們所提出的演算法分別在單一速率群播與多重速率群播上可達到 186%與 905%的改善。

- 考慮使用行為之最小成本群播樹研究：在既定的群播群組成員中，由於使用者並不一定全時的在接收傳輸端的資訊，其接收的行為可以以機率來表示。若考慮使用者是否正在接收的行為，傳統的最小成本樹演算法並無法有效率的被應用。因此我們將此問題以數學模式來表示，並提出最佳化的演算法。根據實驗的結果，我們的演算法較傳統的最小成本數演算法可改善達到 38%。

論文的最後，我們提出五個未來重要的延續研究議題，並根據本論文的研究成果明確地提出這些問題之數學模式供後續學者進行研究。這些議題包括：最小成本多速率多群播樹問題、最大使用者滿意度之多速率多群播樹問題、最大利潤多速率群播樹問題、考慮重新路由之多速率群播樹問題與考慮次群組行為之群播問題。

關鍵詞：群播網路、多重速率群播、多階層編碼技術、史坦那樹、允入控制、網路規劃、拉格蘭日鬆弛法、數學規劃、網路最佳化

Dissertation Abstract

Multicasting Algorithms in Multimedia Networks

Hsu-Chen Cheng, Ph.D.

July 2005

Advisor: Prof. Frank Yeong-Sung Lin

Graduate School of Information Management

National Taiwan University

Based on recent developments in transmission and computing technologies, multimedia applications, such as the teleconferencing and video on demand, have already become achievable and are comprehensively and widely used. Nevertheless, most of these applications require a large amount of bandwidth to deliver multimedia information to multiple destinations simultaneously. One possible way to meet this requirement is via multicasting. Multimedia application environments are characterized by large bandwidth variations due to the heterogeneous access technologies of networks (e.g. analog modem, cable modem, xDSL, and wireless access etc.) and different receivers' quality requirements. In video multicasting, the heterogeneity of the networks and destinations makes it difficult to achieve bandwidth efficiency and service flexibility. There are many challenging issues that need to be addressed in designing architectures and mechanisms for multicast data transmission.

Taking advantage of recent advances in video encoding and transmission technologies, either by a progress coder or video gateway, different destinations can request a different bandwidth requirement from the source. The source then only needs to transmit signals that are sufficient for the highest bandwidth destination into a single multicast tree.

In this dissertation, we study several multicast routing problems, which belong to both single-rate and multi-rate categories. Mathematical formulations are used to model the planning and operational problems, and Lagrangean relaxation techniques, based on the proposed mathematical formulations, are adopted to solve the network planning and operational problems. The scope and contributions of this dissertation are highlighted by the following.

For the min-cost multirate multicasting routing problem, we propose some heuristics to jointly determine the following decision variables: (1) the routing assignment; and (2) the maximum allowable traffic rate of each multicast user group through each link. We successfully model the traffic flow on the links for multi-rate multicasting, and the proposed optimization-based heuristic outperform than the heuristic proposed in the earlier researches. We also deal with the multi-group multicasting planning problem with a capacity constraint.

We also consider the call admission control issues for the single-rate and multi-rate multicasting. We consider the problem of maximum-revenue routing with a partial admission control mechanism. The mechanism means that the admission policy of a multicast group is not based on a traditional “all or none” strategy. Instead it considers accepting partial destinations for the requested multicast group. For a given network topology, a given link capacity, destinations of a multicast group, and the bandwidth requirement of each destination, we attempt to find a feasible routing solution to execute call admission control and apply resource reservation to maximize the revenue of the multicast trees. In addition, we propose a real-time model to deal with long term revenue analysis. The improvement is up to 186% better than the simple algorithm in single-rate transmission, and 905% in multi-rate transmission.

Furthermore, we address the problem of constructing a minimum cost multicast tree by considering dynamic user membership. The motivation of this is to create a mechanism for finding and evaluating the cost-efficiency of a multicast tree with a given network and a fixed set of group members. Unlike other minimum cost multicast tree algorithms, this problem consists of one multicast group of fixed members, where each destination member is dynamic and has a probability of being active, which is observed over some period of time. The improvement of our proposed algorithm is up to 38%.

Finally, we point out five challenging issues to be tackled in the future. We also proposed some feasible mathematical models to formulate these problems. These models are based on the research results of the dissertation.

Keywords: Multicast Network, Multi-rate Multicasting, Layered encoding, Steiner Tree, Call Admission Control, Network Planning, Lagrangean Relaxation, Mathematical Modeling, Optimization.

Table of Contents

CHAPTER 1 INTRODUCTION 1

1.1 OVERVIEW 1

1.2 RESEARCH SCOPE 3

CHAPTER 2 RESEARCH BACKGROUND 8

2.1 QOS ROUTING 8

2.2 QOS MULTICASTING 13

2.3 MULTIRATE MULTICASTING 18

2.5 LAGRANGEAN RELAXATION METHOD 25

2.6 NETWORK TOPOLOGIES FOR EXPERIMENTS AND SIMULATIONS 29

CHAPTER 3 MINIMUM-COST MULTIRATE MULTICASTING ROUTING PROBLEM 33

3.1 OVERVIEW 33

3.2 HEURISTICS OF MULTIRATE MULTIMEDIA MULTICASTING 36

3.2.1 Some Scenarios of the Modified T-M Heuristic 37

3.2.2 Enhanced Modified T-M Heuristic 39

3.3 THE MODEL 40

3.3.1 Problem Description 40

3.3.2 Mathematical Formulation 41

3.4 SOLUTION APPROACH 43

3.4.1 Lagrangean Relaxation 43

3.4.2 Getting Primal Feasible Solutions 46

3.5 COMPUTATIONAL EXPERIMENTS 47

3.6 CONCLUDING REMARKS 50

CHAPTER 4 CAPACITATED MINIMUM-COST MULTIRATE MULTICAST ROUTING PROBLEM 51

4.1 OVERVIEW 51

4.2 A SIMPLE HEURISTIC FOR LINK CONSTRAINED MULTIRATE MULTIMEDIA MULTICASTING 52

4.3 PROBLEM FORMULATION 53

4.3.1 Problem Description 53

4.3.2 Mathematical Formulation 55

4.4 SOLUTION APPROACH 56

4.4.1 Lagrangean Relaxation 56

4.4.2 Getting Primal Feasible Solutions 60

4.5 EXPERIMENTAL RESULTS 61

4.6 CONCLUDING REMARKS 64

CHAPTER 5 THE PARTIAL ADMISSION CONTROL PROBLEM OF SINGLE-RATE MULTICASTING....... 65

5.1 OVERVIEW....... 65

5.2 PROBLEM FORMULATION....... 66

5.3 SOLUTION PROCEDURE....... 70

5.3.1 Lagrangean relaxation....... 70

5.3.2 Getting primal feasible solutions....... 72

5.4 COMPUTATIONAL EXPERIMENTS....... 74

5.5 REAL-TIME ADMISSION CONTROL....... 77

5.5.1 A Real-time LR-based CAC algorithm....... 79

5.5.2 Performance metrics....... 81

5.5.3 Simulation Results....... 82

5.6 CONCLUDING REMARKS....... 95

CHAPTER 6 THE PARTIAL ADMISSION CONTROL PROBLEM OF MULTIRATE MULTICASTING....... 96

6.1 INTRODUCTION....... 96

6.2 PROBLEM FORMULATION....... 97

6.3 SOLUTION PROCEDURE....... 100

6.3.1 Lagrangean relaxation....... 100

6.3.2 Getting primal feasible solutions....... 103

6.4 COMPUTATIONAL EXPERIMENTS....... 105

6.5 REAL-TIME PARTIAL ADMISSION CONTROL FOR MULTIRATE MULTICASTING....... 108

6.5.1 A Real-time LR-based CAC algorithm....... 108

6.5.2 Performance metrics....... 109

6.5.3 Simulation Results....... 110

6.6 CONCLUDING REMARKS....... 122

CHAPTER 7 MINIMUM-COST MULTICAST ROUTING PROBLEM WITH THE CONSIDERATION OF DYNAMIC USER MEMBERSHIP....... 124

7.1 OVERVIEW....... 124

7.2 PROBLEM FORMULATION....... 126

7.2.1 Problem Description....... 126

7.2.2 Mathematical Formulation....... 127

7.3 SOLUTION APPROACH....... 128

7.3.1 Lagrangean Relaxation....... 128

7.3.2 Getting Primal Feasible Solutions....... 131

7.4 COMPUTATIONAL EXPERIMENTS....... 132

CHAPTER 8 CONCLUSION AND FUTURE WORK....... 136

8.1 SUMMARY 136
8.2 FUTURE WORK 137
8.2.1 Min-Cost Multi-tree Multirate Multicast Problem *137*
8.2.2 Max-Revenue Multi-tree Multirate Multicast Problem *140*
8.2.3 Max-Profit Multirate Multicast Problem *143*
8.2.4 Routing Algorithms Considering Multirate Multicast Service and Traffic Rerouting147
8.2.5 Considering Subgroup Behavior *151*

REFERENCES 152

List of Tables

TABLE 1.1: SCOPE AND PROBLEM DEFINITION OF THIS DISSERTATION 7
TABLE 2.1: PROPERTIES OF MULTICAST PROTOCOLS 17
TABLE 3.1: DESCRIPTION OF NOTATIONS (MCMR) 40
TABLE 3.2: PARAMETERS FOR LAGRANGEAN RELAXATION (MCMR) 48
TABLE 3.3: SELECTED RESULTS OF COMPUTATIONAL EXPERIMENTS (MCMR) 49
TABLE 4.1: DESCRIPTION OF NOTATIONS (C-MCMR) 54
TABLE 4.2: PARAMETERS FOR LAGRANGEAN RELAXATION (C-MCMR) 62
TABLE 4.3: SELECTED RESULTS OF COMPUTATIONAL EXPERIMENTS (C-MCMR) 62
TABLE 5.1: DESCRIPTION OF NOTATIONS (PCAC-S) 67
TABLE 5.2: SELECTED RESULTS OF COMPUTATIONAL EXPERIMENTS (PCAC-S) 75
TABLE 5.3: AN EXAMPLE OF CALL NUMBER CALCULATION WITH Λ=30, T=90, AND H=3 78
TABLE 5.4: AVERAGE ERROR GAP FOR SOLVING PCAC-S PROBLEM 93
TABLE 5.5: COMPARISON OF LONG-TERM SYSTEM ACCUMULATED REVENUE (PCAC-S) 94
TABLE 6.1: DESCRIPTION OF NOTATIONS (PCAC-M) 97
TABLE 6.2: SELECTED RESULTS OF COMPUTATIONAL EXPERIMENTS (PCAC-M) 107
TABLE 6.3: AVERAGE ERROR GAP FOR SOLVING THE PCAC-S PROBLEM 120
TABLE 6.4: COMPARISON OF LONG-TERM SYSTEM ACCUMULATED REVENUE (PCAC-S) 122
TABLE 7.1: TOTAL COST OF THE EXAMPLE NETWORK 125
TABLE 7.2: DESCRIPTION OF NOTATIONS (MCRD) 126
TABLE 7.3: PARAMETERS FOR LAGRANGEAN RELAXATION (MCRD) 133
TABLE 7.4: SELECTED RESULTS OF COMPUTATIONAL EXPERIMENTS (MCRD) 134
TABLE 8.1: DESCRIPTION OF NOTATIONS (FW1) 138
TABLE 8.2: EXAMPLE OF UTILITY FUNCTION 141
TABLE 8.3: DESCRIPTION OF NOTATIONS (FW2) 141
TABLE 8.4: DESCRIPTION OF NOTATIONS (FW3) 144
TABLE 8.5: DESCRIPTION OF NOTATIONS (FW4) 148

List of Figures

FIGURE 1.1: THE CLASSIFICATION OF MULTICAST RESEARCH 4
FIGURE 1.2: THE RESEARCH PROBLEMS ADDRESSED IN THIS DISSERTATION 5
FIGURE 2.1: THE CATEGORIES OF MULTICAST ROUTING PROBLEMS 12
FIGURE 2.2: EXAMPLE OF THE T-M HEURISTIC FOR STEINER TREE 21
FIGURE 2.3: EXAMPLE OF THE T-M HEURISTIC 21
FIGURE 2.4: EXAMPLE OF THE T-M HEURISTIC 22
FIGURE 2.5: THE RELATIONSHIP BETWEEN BASIC COMPONENTS OF ADMISSION CONTROL SCHEMES 25
FIGURE 2.6: THE CONCEPT OF THE DUAL PROBLEM 28
FIGURE 2.7: THE OVERALL PROCEDURE OF THE LR APPROACH 28
FIGURE 2.8: DEGREE DISTRIBUTION OF EXPONENTIAL NETWORKS AND POWER-LAW NETWORKS 31
FIGURE 2.9: EXAMPLE NETWORKS WITH POWER-LAW DISTRIBUTION 31
FIGURE 3.2: AN EXAMPLE OF THE M-T-M HEURISTIC FOR MULTI-LAYERED VIDEO DISTRIBUTION WITH CONSTANT LINK COSTS. 37
FIGURE 3.3: AN EXAMPLE OF THE M-T-M HEURISTIC FOR MULTI-LAYERED VIDEO DISTRIBUTION WITH ARBITRARY LINK COSTS. 38
FIGURE 3.4: REGULAR NETWORKS (MCMR) 47
FIGURE 4.1: REGULAR NETWORKS (C-MCMR) 62
FIGURE 5.1: REGULAR NETWORKS (PCAC-S) 74
FIGURE 5.2: THE TIME DIAGRAM OF REAL-TIME ADMISSION CONTROL 78
FIGURE 5.3: THE PROCEDURE OF THE LR-BASED REAL-TIME CAC ALGORITHM 79
FIGURE 5.4: EFFECT OF THE TIME BUDGET ON THE SR OF A GRID NETWORK (PCAC-S) 83
FIGURE 5.5: EFFECT OF THE TIME BUDGET ON THE SR OF A CELLULAR NETWORK (PCAC-S) 84
FIGURE 5.6: EFFECT OF THE TIME BUDGET ON THE SR OF A RANDOM NETWORK (PCAC-S) 84
FIGURE 5.7: EFFECT OF THE TIME BUDGET ON THE SR OF A SCALE-FREE NETWORK (PCAC-S) 84
FIGURE 5.8: EFFECT OF THE TIME BUDGET ON THE BR OF A GRID NETWORK (PCAC-S) 85
FIGURE 5.9: EFFECT OF THE TIME BUDGET ON THE BR OF A CELLULAR NETWORK (PCAC-S) 85
FIGURE 5.10: EFFECT OF THE TIME BUDGET ON THE BR OF A RANDOM NETWORK (PCAC-S) 86
FIGURE 5.11: EFFECT OF THE TIME BUDGET ON THE BR OF A SCALE-FREE NETWORK (PCAC-S) 86
FIGURE 5.12: EFFECT OF THE TIME BUDGET ON THE AD OF A GRID NETWORK (PCAC-S) 86
FIGURE 5.13: EFFECT OF THE TIME BUDGET ON THE AD OF A CELLULAR NETWORK (PCAC-S) 87
FIGURE 5.14: EFFECT OF THE TIME BUDGET ON THE AD OF A RANDOM NETWORK (PCAC-S) 87
FIGURE 5.15: EFFECT OF THE TIME BUDGET ON THE AD OF A SCALE-FREE NETWORK (PCAC-S) 87
FIGURE 5.16: EFFECT OF THE TIME BUDGET ON THE NI OF A GRID NETWORK (PCAC-S) 88
FIGURE 5.17: EFFECT OF THE TIME BUDGET ON THE NI OF A CELLULAR NETWORK (PCAC-S) 88
FIGURE 5.18: EFFECT OF THE TIME BUDGET ON THE NI OF A RANDOM NETWORK (PCAC-S) 89
FIGURE 5.19: EFFECT OF THE TIME BUDGET ON THE NI OF A SCALE-FREE NETWORK (PCAC-S) 89

FIGURE 5.20: EFFECT OF THE TIME BUDGET ON THE MIR OF A GRID NETWORK (PCAC-S) 90
FIGURE 5.21: EFFECT OF THE TIME BUDGET ON THE AIR OF A GRID NETWORK (PCAC-S) 90
FIGURE 5.22: EFFECT OF THE TIME BUDGET ON THE MIR OF A CELLULAR NETWORK (PCAC-S) 90
FIGURE 5.23: EFFECT OF THE TIME BUDGET ON THE AIR OF A CELLULAR NETWORK (PCAC-S) 91
FIGURE 5.24: EFFECT OF THE TIME BUDGET ON THE MIR OF A RANDOM NETWORK (PCAC-S) 91
FIGURE 5.25: EFFECT OF THE TIME BUDGET ON THE AIR OF A RANDOM NETWORK (PCAC-S) 91
FIGURE 5.26: EFFECT OF THE TIME BUDGET ON THE MIR OF A SCALE-FREE NETWORK (PCAC-S) 92
FIGURE 5.27: EFFECT OF THE TIME BUDGET ON THE AIR OF A SCALE-FREE NETWORK (PCAC-S) 92
FIGURE 5.28: LOWER BOUND CALCULATED FOR REAL TIME SIMULATION 94
FIGURE 6.1: REGULAR NETWORKS (PCAC-M) 105
FIGURE 6.2: EFFECT OF THE TIME BUDGET ON THE SR OF A GRID NETWORK (PCAC-M) 111
FIGURE 6.3: EFFECT OF THE TIME BUDGET ON THE SR OF A CELLULAR NETWORK (PCAC-M) 111
FIGURE 6.4: EFFECT OF THE TIME BUDGET ON THE SR OF A RANDOM NETWORK (PCAC-M) 112
FIGURE 6.5: EFFECT OF THE TIME BUDGET ON THE SR OF A SCALE-FREE NETWORK (PCAC-M) 112
FIGURE 6.6: EFFECT OF THE TIME BUDGET ON THE BR OF A GRID NETWORK (PCAC-M) 113
FIGURE 6.7: EFFECT OF THE TIME BUDGET ON THE BR OF A CELLULAR NETWORK (PCAC-M) 113
FIGURE 6.8: EFFECT OF THE TIME BUDGET ON THE BR OF A RANDOM NETWORK (PCAC-M) 113
FIGURE 6.9: EFFECT OF THE TIME BUDGET ON THE BR OF A SCALE-FREE NETWORK (PCAC-M) 114
FIGURE 6.10: EFFECT OF THE TIME BUDGET ON THE AD OF A GRID NETWORK (PCAC-M) 114
FIGURE 6.11: EFFECT OF THE TIME BUDGET ON THE AD OF A CELLULAR NETWORK (PCAC-M) 114
FIGURE 6.12: EFFECT OF THE TIME BUDGET ON THE AD OF A RANDOM NETWORK (PCAC-M) 115
FIGURE 6.13: EFFECT OF THE TIME BUDGET ON THE AD OF A SCALE-FREE NETWORK (PCAC-M) 115
FIGURE 6.14: EFFECT OF THE TIME BUDGET ON THE NI OF A GRID NETWORK (PCAC-M) 116
FIGURE 6.15: EFFECT OF THE TIME BUDGET ON THE NI OF A CELLULAR NETWORK (PCAC-M) 116
FIGURE 6.16: EFFECT OF THE TIME BUDGET ON THE NI OF A RANDOM NETWORK (PCAC-M) 116
FIGURE 6.17: EFFECT OF THE TIME BUDGET ON THE NI OF A SCALE-FREE NETWORK (PCAC-M) 117
FIGURE 6.18: EFFECT OF THE TIME BUDGET ON THE MIR OF A GRID NETWORK (PCAC-M) 117
FIGURE 6.19: EFFECT OF THE TIME BUDGET ON THE AIR OF A GIRD NETWORK (PCAC-M) 118
FIGURE 6.20: EFFECT OF THE TIME BUDGET ON THE MIR OF A CELLULAR NETWORK (PCAC-M) 118
FIGURE 6.21: EFFECT OF THE TIME BUDGET ON THE AIR OF A CELLULAR NETWORK (PCAC-M) 118
FIGURE 6.22: EFFECT OF THE TIME BUDGET ON THE MIR OF A RANDOM NETWORK (PCAC-M) 119
FIGURE 6.23: EFFECT OF THE TIME BUDGET ON THE AIR OF A RANDOM NETWORK (PCAC-M) 119
FIGURE 6.24: EFFECT OF THE TIME BUDGET ON THE MIR OF A SCALE-FREE NETWORK (PCAC-M) 119
FIGURE 6.25: EFFECT OF THE TIME BUDGET ON THE AIR OF A SCALE-FREE NETWORK (PCAC-M) 120
FIGURE 7.1: EXAMPLE NETWORK (MCRD) 125
FIGURE 7.2: REGULAR NETWORKS (MCRD) 133

CHAPTER 1 INTRODUCTION

1.1 Overview

With the popularity of the Internet, applications based on network services are growing rapidly. The power of the Internet lies in its ability to interconnect computers worldwide, so long as they follow the same protocols. After more than a decade of continuous commercial development, this global network has revolutionized the way people communicate and the way business is conducted.

In order to meet the bandwidth requirements for such applications, network operators are spending more and more resources on enlarging their network capacity, including setting up new physical links and upgrading their existing links to higher transmission rates. In terms of enlarging network capacity, there is another way to achieve the goal of providing better service quality, i.e., through network planning or so-called traffic engineering. Traffic engineering is the process of controlling how traffic flows through a network in order to optimize resource utilization and network performance. It can also provide Quality-of-Service (QoS) assurances. The ability to provide reliable QoS may well become a crucial factor in influencing customers' willingness to pay for networks.

The current Internet operates in a best-effort manner, which is considered inefficient for applications that demand QoS. These applications, such as voice over IP (VoIP), video on demand (VoD), multimedia on demand (MoD), video conferencing, and Tele-Health require QoS or some other form of prioritization guarantees to make successful connections. To achieve this, admission control is essential.

Based on recent developments in transmission and computing technologies, multimedia applications, such as the teleconferencing and video on demand, have already become achievable and are comprehensively and widely used. Nevertheless, most of these applications require a large amount of bandwidth to deliver multimedia information to multiple destinations simultaneously. One possible way to meet this requirement is via multicasting.

Multicast means the transmission of data from one node (source node) to a selected multicast group of nodes (member nodes or destination nodes) in a communication network. Multicast routing takes advantage of trees, called multicast routing trees, in the network topology for transmissions, which minimizes resource usage, such as cost and bandwidth, by sharing links when transmitting data from one node to many destination nodes. The routing algorithm will only replicate at appropriate locations in order to reach all its destination nodes. A minimum cost multicast tree is also referred to as a Steiner tree. In other words, a Steiner tree constructs a minimum cost tree for a subset of the nodes in a network with fixed costs on the corresponding network links. The problem of determining a Steiner tree is known to be NP-complete [1].

IP Multicast traffic for a particular (source, destination group) pair is transmitted from the source to the receivers via a spanning tree that connects all the hosts in the group. Although different IP Multicast routing protocols use different techniques to construct these multicast spanning trees, once a tree is constructed, all multicast traffic is distributed over it. IP Multicast routing protocols generally follow one of two basic approaches, depending on the expected distribution of multicast group members throughout the network. The first approach is based on the assumption that multicast group members are densely distributed throughout the network (i.e., many of the subnets contain at least one group member) and that bandwidth is large. So-called “dense-mode” multicast routing protocols rely on a technique called flooding to propagate information to all network routers. Dense-mode routing protocols include the Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), and Protocol-Independent Multicast - Dense Mode (PIM-DM). The multicasting backbone (MBone), which uses DVMRP for multicast routing, is one of the applications that have been developed rapidly on the Internet using IP multicasting technology.

The second approach to multicast routing, called sparse mode, basically assumes that multicast group members are sparsely distributed throughout the network and that bandwidth is not necessarily widely available; for example, across many regions of the Internet, or if users are connected via ISDN lines. Sparse-mode does not imply that the group has only a few members, just that they are widely dispersed. In this case, flooding would unnecessarily waste network bandwidth and could cause serious performance problems. Hence, “sparse-mode” multicast routing protocols must rely on more selective techniques to set up and maintain multicast trees. Sparse-mode routing protocols include Core-Based Trees (CBT) and Protocol-Independent Multicast - Sparse Mode (PIM-SM) [2].

Furthermore, current real-time applications, such as teleconferencing, remote collaboration, and distance education, involve the transmission of multimedia information. Therefore, it is essential to satisfy quality-of-service constraints (such as bounded end-to-end delay, bounded delay-variation, and bandwidth requirements). At the routing level, these three requirements translate into the problem of determining a multicast tree, usually rooted at the source node and spanning the set of receiver nodes. These quality-of-service constraints typically impose a restriction on acceptable multicast trees.

1.2 Research Scope

Many researchers have focused on multicast routing problems and proposed various solutions. Such problems can be divided into different categories according to different dimensions. For example, from the view point of tree type, multicast routing algorithms can be divided into two categories: source-based tree and shared tree. Meanwhile, from the viewpoint of tree construction, multicast routing algorithms can be divided into two categories: centralized algorithms and distributed algorithms. A detailed survey will be presented in Chapter 2.

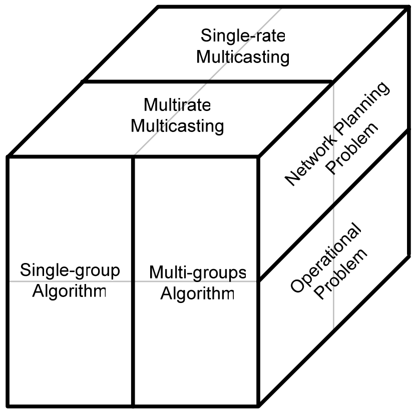


Figure 1.1: The classification of multicast research

In this dissertation, we discuss the multicast problem from three different dimensions. First, a multicast problem can be categorized by its purpose. The goal of multicast network planning is to design a network with minimum installation and operating costs, subject to traffic requirements and other performance constraints. In the planning problem, we know all the related parameters from measurement or forecasting in advance. The time budget for executing an algorithm is not a constraint in the planning problem. On the other hand, the algorithm used for operational problems is constrained by the processing time.

The second dimension relates to the object that the algorithm is concerned with. Most multicast routing problems only consider a single multicast session. However, in the real world, several multicast sessions are broadcast simultaneously, and therefore contend for the limited resources (such as bandwidth) of networks. This creates the multiple-multicast routing problem. If we use a single-group multicast algorithm to deal with a multi-group multicast problem, we may not obtain a feasible solution, even though one exists.

The third dimension is about whether or not an algorithm can deal with the heterogeneity of the users. The heterogeneity of networks and destinations makes it difficult to achieve bandwidth efficiency and service flexibility. Single-rate multicasting means that the bandwidth requirements of the users within a multicast group are the same, whereas multirate multicasting means that users can request a different quality of video from the sender. The sender then encodes the video into several different layered streams and transmits it through single or multiple multicast tree(s).

In this dissertation, we study several multicast routing problems (see Table 1.1), which belong to both single and multiple categories. Mathematical formulations are used to model the planning and operational problems, and Lagrangean relaxation techniques, based on the proposed mathematical formulations, are adopted to solve the network planning and operational problems.

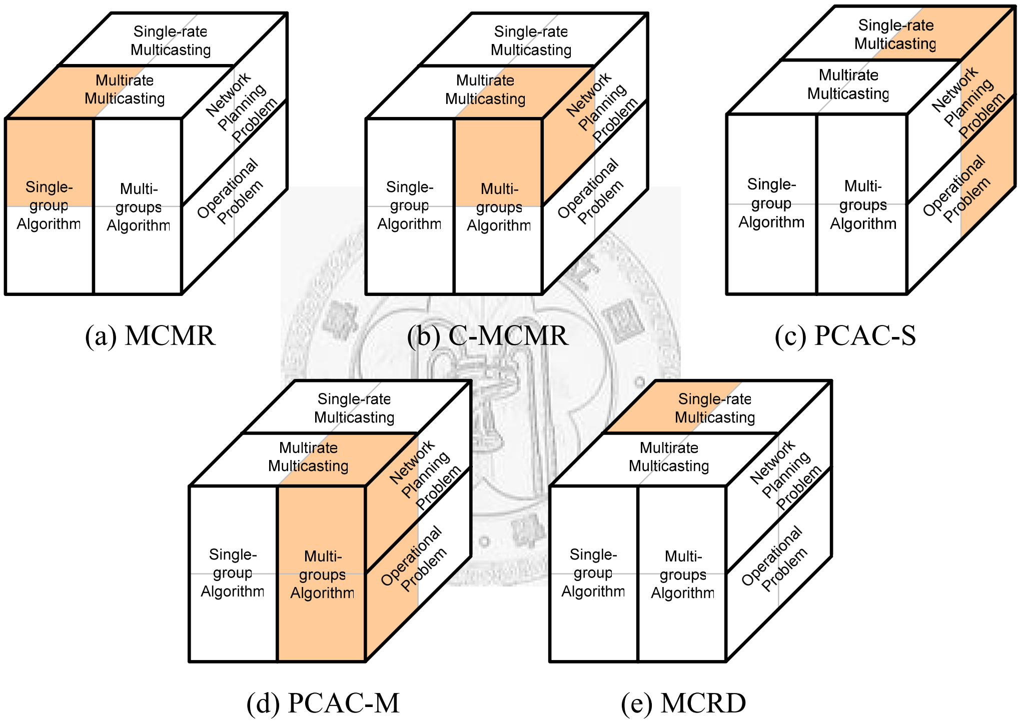


Figure 1.2: The research problems addressed in this dissertation

In Chapter 3, we discuss the min-cost multirate multicasting routing problem (MCMR). A user group is an application requesting transmission on a network that has one source and one or more destinations. Given the network topology and bandwidth requirement of every destination of a user group, we want to jointly determine the following decision variables: (1) the routing assignment; and (2) the maximum allowable traffic rate of each multicast user group through each link. This problem is a single-group multirate multicasting planning problem.

In Chapter 4, continuing from Chapter 3, we deal with the multicasting planning problem with a capacity constraint (C-MCMR). This problem is a multi-group multirate multicasting planning problem.

In Chapter 5, we consider the problem of maximum-revenue routing with a partial admission control mechanism for single-rate multicasting (PCAC-S). The mechanism means that the admission policy of a multicast group is not based on a traditional “all or none” strategy. Instead it considers accepting partial destinations for the requested multicast group. More specifically, for a given network topology, a given link capacity, destinations of a multicast group, and the bandwidth requirement of each multicast group, we attempt to find a feasible admission decision and routing solution to maximize the revenue of the multicast trees. In this chapter, we also perform a simulation of real-time CAC.

In Chapter 6, we consider the problem of maximum-revenue routing with a partial admission control mechanism for multirate multicasting (PCAC-M). Multirate multicasting is different from single-rate multicasting. Specifically, for a given network topology, a given link capacity, destinations of a multicast group, and the bandwidth requirement of each destination, we attempt to find a feasible routing solution to execute call admission control and apply resource reservation to maximize the revenue of the multicast trees.

In Chapter 7, we address the problem of constructing a minimum cost multicast tree by considering dynamic user membership (MCRD). Unlike other minimum cost multicast tree algorithms, this problem consists of one multicast group of fixed members, where each destination member is dynamic and has a probability of being active, which is observed over some period of time. Because of omission of node join/leave handling, this model is more suitable for prediction and planning purposes than for online maintenance of multicast trees.

A summary of our research problem is presented in Table 1.1.

Table 1.1: Scope and problem definition of this dissertation

Problem Name	Input Parameter	Constraints	Output
Min-cost Multirate Multicast Routing (MCMR)	▪ Network topology ▪ Link transmission Costs ▪ Group members ▪ Traffic requirements	▪ Tree topology constraint ▪ Multirate multicasting constraint	▪ Routing assignment ▪ Link traffic
Capacitated Min-cost Multirate Multicast Routing (C-MCMR)	▪ Network topology ▪ Link capacity ▪ Link transmission Costs ▪ Group members ▪ Traffic requirements	▪ Tree topology constraint ▪ Link capacity constraint ▪ Multirate multicasting constraint ▪ Multi-commodity flow constraint	▪ Routing assignment ▪ Link traffic
Partial Admission Control Problem for Single-rate Multicasting (PCAC-S)	▪ Network topology ▪ Link Capacity ▪ Group members ▪ Traffic requirements ▪ Revenue information	▪ Tree topology constraint ▪ Link capacity constraint ▪ Multi-commodity flow constraint	▪ Routing assignment ▪ Resource reservation ▪ Call admission result
Partial Admission Control Problem for Single-rate Multicasting (PCAC-M)	▪ Network topology ▪ Link Capacity ▪ Group members ▪ Traffic requirements ▪ Revenue information	▪ Tree topology constraint ▪ Link capacity constraint ▪ Multirate multicasting constraint ▪ Multi-commodity flow constraint	▪ Routing assignment ▪ Resource reservation ▪ Call Admission result
Min-cost Multicast Routing Problem with the Consideration of Dynamic User Membership (MCRD)	▪ Network Topology ▪ Group members ▪ Group behavior ▪ Traffic requirements ▪ Link transmission cost ▪ Link installation cost	▪ Tree topology constraint ▪ Multicasting constraint	▪ Routing assignment

CHAPTER 2 RESEARCH BACKGROUND

2.1 QoS Routing

The Internet which is now a vital communications channel was originally used in the 80s and the early 90s by research and education communities for computer data transmission, such as electronic mail, network news, and file transfers. The most demanding application from the service quality point of view was a network remote logon as an interactive application. Also, the bandwidth requirementwas small and occasional delay variations in the order of several seconds could be tolerated.

The routing deployed in today's Internet focuses on connectivity and typically supports only one type of datagram service called "best effort". In other words, the Internet will try its best to forward user traffic, but it can not provide any guarantees regarding loss rate, bandwidth, delay, delay jitter, etc. For example, packets can be dropped indiscriminately in the event of congestion. This kind of service works fine for some traditional applications (such as FTP and email), but recently, many interactive or real-time services have been introduced and, at the same time, the economic importance of the Internet has grown. Transmitting interactive real-time media is the greatest challenge in packet-based networks, such as IP networks. The end-to-end delay, the delay variations (jitter), and the packet loss must not exceed certain limits; otherwise, the usability of the service will be badly degraded. This is intolerable for emerging real-time multimedia applications, which require high bandwidth, low delay, and low delay jitter. In other words, these new applications require better transmission services than "best-effort". Thus, the issue of Quality-of-Service (QoS) has become a major research area.

Current Internet routing protocols [3], e.g., OSPF and RIP, use "shortest path routing", which is optimized for a single arbitrary metric, administrative weight, or hop count. Alternative paths with acceptable costs, but non-optimal costs, can not be used to route traffic. QoS-based routing must extend the current routing paradigm in three basic ways. First, to support traffic using integrated-service class of services, multiple paths between node pairs must be calculated. Such calculations require the distribution of routing metrics, such as delay and available bandwidth. If the metrics change often, routing updates become more frequent and consume more network bandwidth and router CPU cycles.

Second, today's opportunistic routing shifts traffic to a "better" path as soon as it is found, even if the service requirement is satisfied. However, such rerouting can introduce routing oscillations as traffic shifts back and forth between alternate paths. Furthermore, delay variation and jitter experienced by end users increase.

Third, as mentioned earlier, today's optimal path routing algorithms do not support alternative routing. If the best existing path cannot admit a new flow, the associated traffic cannot be forwarded, even if an adequate alternate path exists.

QoS routing is a critical network function for the transmission and distribution of digitalized audio or video content throughout communication networks. It has two objectives: (1) to find routes that satisfy the QoS requirements, and (2) to make efficient use of network resources. A great deal of research has been conducted on QoS routing issues in recent years. Overall, based on the way information is maintained, existing QoS routing algorithms can be divided into three broad classes: (1) source routing algorithms, (2) distributed routing algorithms, and (3) hierarchical routing algorithms. In [4], S. Chen and K. Nahrstedt conduct a thorough survey of these QoS routing algorithms. However, they focus on network models in virtual circuit mode, which is connection oriented. In [5], J. Kleinberg address an NP-complete problem that combines selected paths for routing and allocating bandwidth fairly among connections in the max-min sense. But, as in [5], their approach is still more connection-oriented with a single source. In [6], Ghosh, Sarangan, and Acharya propose a new distributed routing algorithm for QoS flows. That contains a new packet forwarding mechanism based on the QoS requirements of the connection. The two-level forwarding mechanism has a low overhead compared to flooding-based call setup. However, the algorithm only considers bandwidth requirements, but other QoS requirements such as loss, delay, and jitter are also important and must be considered. Sufficient bandwidth alone cannot provide smooth video-on-demand service. The algorithm should control the delay and jitter under certain requirements. In addition, it only focuses on the unicast flows, without considering multicast flows. The following are some traffic handling mechanisms:

802.1p: 802.1p is a traffic-handling mechanism that supports QoS in IEEE 802technology LANs. 802.1p defines a field in the layer-2 header of 802 packets that can carry one of eight priority values. Typically, hosts or routers sending traffic into a LAN mark each transmitted packet with the appropriate priority value. LAN devices, such as switches, bridges and hubs, are expected to treat the packets accordingly (by making use of underlying queuing mechanisms). The scope of the 802.1p priority mark is limited to the LAN. Once packets leave the LAN, through a layer-3 device, the 802.1p priority is removed.

Differentiated Services (Diffserv): Diffserv is a layer-3 QoS mechanism that defines a field in the layer-3 header of IP packets, called the diffserv codepoint (DSCP). Typically, hosts or routers sending traffic into a diffserv network mark each transmitted packet with the appropriate DSCP, which is a six-bit field, spanning the fields formerly known as the type-of-service (TOS) fields and the IP precedence fields. Routers within the diffserv network use the DSCP to classify packets and apply specific queuing or scheduling behavior (known as a per-hop behavior or PHB) based on the results of the classification.

Integrated Services (Intserv): Intserv is a service framework comprised of two services: guaranteed service and controlled load service. The former promises to carry a certain traffic volume with a quantifiable bounded latency. The latter agrees to carry a certain traffic volume with the “appearance of a lightly loaded network”. These are quantifiable services in the sense that they are designed to provide quantifiable QoS for a specified quantity of traffic.

QoS routing is an important element for supporting multimedia applications. The goal of QoS routing is to select network routes with sufficient resources for the requested QoS parameters and satisfy the QoS requirements for every admitted connection. It must also achieve efficiency in resource utilization. Many QoS routing algorithms with a variety of constraints have been proposed in recent years.

Wang and Crowcroft [7] consider a number of issues in QoS routing. They try to evaluate the basic component of QoS routing, namely, finding a path that satisfies multiple constraints and its implications for routing metric selection. Moreover, they propose three path computation algorithms for source routing and hop-by-hop routing. However, as QoS routing is an integral part of a resource management system, it should be jointly considered with other components, such as admission control, in resource management architectures.

Ergun, Sinha, and Zhang [8] examine a network model in which each link is associated with a set of delays and costs. The aim is to choose a path for each O-D pair and determine a set of “per link” delay guarantees along this path to satisfy the requested constraint, while minimizing the total cost. In the case where the O-D path is known, the authors try to optimally partition the end-to-end delay constraint into link constraints along the path. To this end, they present approximation algorithms for both problems. For the first problem, polynomial-time ε-approximations are presented. However, the authors use heuristics to solve PARTION problems, and do not consider more complicated structures, such as multicast trees.

Fang and Ellen [9] specifically focus on topology aggregation, which can reduce overhead by orders of magnitude. They also investigate the interaction of topology aggregation with other important factors that contribute to performance, such as routing algorithms and network configurations. They consider five common route selection methods and propose two methods for aggregating routing information. As a result, for multimedia applications, we can adopt this scalable concept to adjust the above route selection methods and different network configurations to satisfy our efficient and flexible principles.

Most QoS routing algorithms consider the optimization of resource utilization based on an abstract metric, such as cost. Apostolopoilos et al., [10] study complexity and frequent computation costs and propose solutions, such as a higher level of admission control in a heavily loaded environment, which achieve good performance with reduced costs. This is called "trunk reservation". However, from a network operator's point of view, it would be beneficial to develop a generic algorithm in advance, instead of implementing the approaches at execution time.

Chen and Nahrstrdt [4] discuss the QoS requirements of a connection. QoS can be represented as a set of constraints, which can be link constraints, path constraints, or tree constraints. The basic function of QoS routing is to find a feasible path (tree). In their research, the authors also provide a complete survey of recent developments in QoS routing, which is presented in Figure 2.1. In the next section, we discuss the QoS routing issue in multicasting networks.

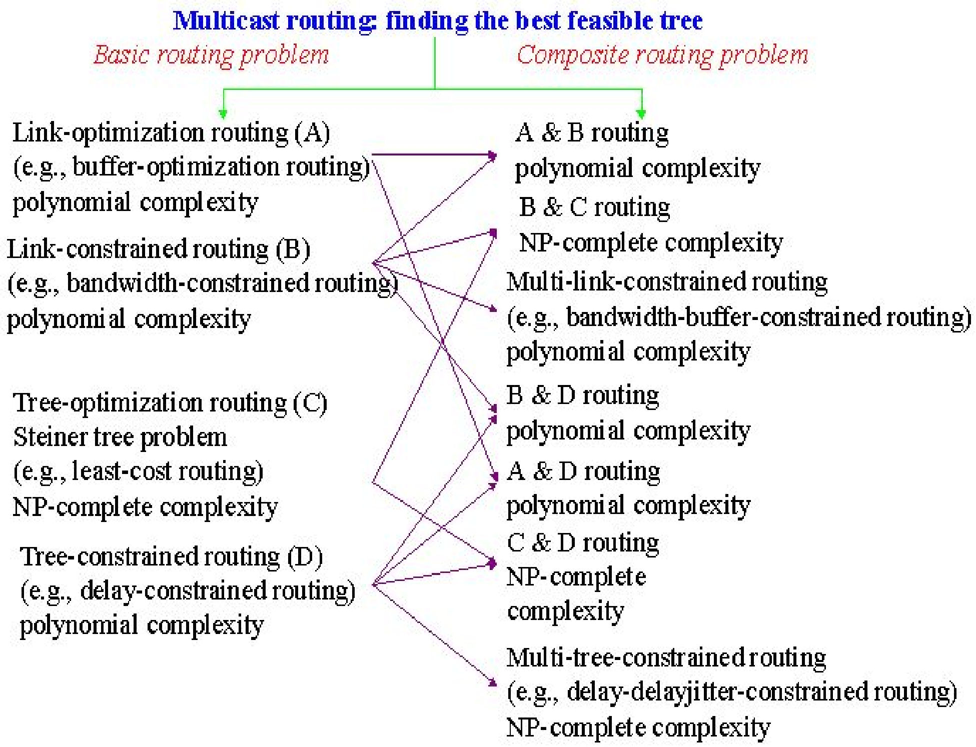


Figure 2.1: The categories of multicast routing problems

2.2 QoS Multicasting

Multicasting is widely used by many multimedia applications because of the benefits of sharing link utilization. Some fundamental issues in IP multicasting, such as dynamic group management, routing efficiency, time-sensitive delivery of multicast traffic, and scalability, have all been investigated and are still the focus of intensive research [20]. With the rapid development of network technologies, there is a growing demand for quality of service (QoS) support in multicasting. This support can be formulated as some parameters and constraints that consider the design of a multicast tree as well as traffic flow control.

In [10], the author divides the constraints used in multicast tree construction into two categories:

1. Link constraints: the restrictions on the use of links for route selection, such as the bandwidth or buffer on one link.
2. Tree constraints: the restrictions on the whole multicast tree. For example, the delay along the tree-path from the tree's root to the group destinations, the difference in the delay to each member, or the routing cost of the whole multicast tree.

Obviously, the tree constraints are obtained from the link constraints or link metrics. Thus, the tree constraints can also be classified into three basic types; let $m(P)$ be a performance metric for a path, P:

1. <u>Additive tree constraints</u>: for any path $P_T(u,v)=(u,i,j,\ldots,k,v)$ of a multicast tree, T, its constraint is additive if

$$m(u,v) = m(u,i) + m(i,j) + \ldots + m(k,v).$$

2. <u>Multiplicative tree constraint</u>s: for any path $P_T(u,v)=(u,i,j,\ldots,k,v)$ of a multicast tree T, its constraint is multiplicative if

$$m(u,v) = m(u,v) \text{、} m(i,j) \text{、} \ldots \text{、} m(k,v).$$

The probability, $1- P_L(u,v)$, for a packet to reach v from u along $P_T(u,v)$ is multiplicative, where $P_L(u,v)$ is the loss rate of a packet from u to v, and

$$P_L(u,v) = 1\text{-}[\,(1\text{-}P_L(u,i))\ \text{、}(1\text{-}P_L(i,j))\ \text{、}\ldots\ \text{、}(1\text{-}P_L(k,v))].$$

3. <u>Concave tree constraint</u>s: for any path $P_T(u,v)=(u,i,j,\ldots,k,v)$ of a multicast tree, T, its constraint is concave if

$$m(u,v) = min\{m(u,i), m(i,j), \ldots, m(k,v)\}.$$

The bandwidth available on the path $P_T(u,v)$ is concave.

Therefore, each tree constraint, such as the packet delivery rate and delay variation, can be derived from the three types described above.

However, Wang and Crowcroft [7] proved that finding a path with multiple additive constraints, multiple multiplicative constraints, or multiple additive and multiplicative constraints is an NP-complete problem.

For multicasting algorithms, there are two classification criteria: 1) the multicast tree type, and 2) the method of tree construction. The shared tree, the source-based tree, and the Steiner tree are the three tree types; and the distributed algorithm and the centralized algorithm are the two methods for tree construction.

Data packets addressed to a multicast group may be routed on a tree that is specific to a particular sender and group, or a tree that is shared by all senders to the group. The routing tree used in the first approach is a source-based tree, and the other is a shared tree. In the source-based tree approach, all senders build separate trees to all group members, whereas in the shared tree approach, all group members share the same multicast tree. A source-based tree is normally a shortest-path tree (SPT), while the Steiner tree (ST) tries to span all group members at minimal cost. Constructing a Steiner tree is an NP-complete problem, and many heuristic algorithms have been proposed to solve it [26][27][28][29][30][31][32].

Reverse path forwarding (RPF), a type of source-based tree, has been widely used in IP multicasting. Although it is optimized for a dense mode, it does not consider group membership. Thus, an improved RPF method that uses a “flood and prune” approach has been proposed to overcome this weakness. However, if the number of sources and groups grows too large, the routers’ memories could become saturated. Traditional multicast protocols, such as DVMRP [12] and MOSPF [13], are also types of source-based tree. Such a tree is denoted by (*S*, *G*), where *S* is the sender and also the tree root; and *G* is the group ID, usually the IP class D address. Although the source tree approach is much easier to implement than the shared tree approach, it is difficult to maintain source trees if there are several senders in one group. We assume there are *n* members in a group. If *x* new members join the group, then *x* new trees are built, and *n* trees updated; if y members leave the group, then *y* trees are deleted and *n*-*y* trees updated.

Besides, communications between a pair of group members, say *x* and *y*, may be routed along different paths, because the tree path from *x* to *y* is different to the path from *y* to *x*. This may result in the wrong order of interaction between users. However, a source tree can be built with suitable paths to choose from, since it does not require that all group members use the same path.

As shared trees are the most recent routing approach and their protocols are still experimental. Core Based Trees (CBT) [14][15] and Protocol Independent Multicast-Sparse Mode (PIM-SM) [16] are famous examples of such trees. The shared-tree method builds a tree that spans all users, and each user sends data along the tree to the tree root, which then forwards the data to other members. Such a root is usually called the core, center, or rendezvous point (RP). The main advantage of a shared-tree is that all members use the same tree, so it is not necessary to maintain multiple trees as some members join and leave. But there is a serious problem with traffic concentration. Because all members use the same tree, the tree links near the core become bottlenecks when too many members transfer data at the same time. Besides, how to choose the optimal core in a shared tree is also an important issue, since the group members can not be known a priori [17][18]. In 1999, some researchers proposed a non-core- based shared tree architecture for IP multicasting to solve the problems of core selection and traffic congestion [19]. In this approach, the concept of multicast nodes was introduced to replace the core node. Multicast nodes are nodes through which group members can join a multicast tree. Some members will select the closest on-tree multicast node to join the tree. In a core-based architecture, the existence of core nodes is broadcast in the subnet by a boot-strap router. Thus, all nodes in the network should maintain the current tree information, since members may join or leave at any time.

Algorithms for constructing a tree with QoS constraints can be classified as centralized and distributed algorithms. The multicast tree for the "best effort" service should not be computed before being built completely. There are many algorithms for building such trees [20]

For centralized algorithms, the tree constructor, usually the tree root or tree source is assumed to have all the information necessary to build the tree. The needed information is usually the QoS metrics of the links between any two network nodes, such as the link delay and the link bandwidth. Although, the information can be collected and updated using a topology-broadcast algorithm, the computation load on the tree constructor becomes much heavier as the scale of the Internet gets larger. Many centralized multicast algorithms have been proposed, for example, Zhu[21], Rouskas[22].

On the other hand, distributed multicast algorithms assume that each network node knows its local information, and executes the same algorithm. Therefore, the complexity of the distributed algorithm is much lower than the centralized one. But part of the cost of tree construction is incurred by transferring control messages among nodes so that each node can maintain tree states and decide how to proceed with the algorithm. If a distributed algorithm needs to satisfy many QoS constraints, the message complexity will be extremely high. Several distributed algorithms have been proposed, for example Jia [23] and Cehn [24]. Also, Wang [11] has made a complete summary of these algorithms.

In accordance with above protocols, the following parameters must be considered [25].

1. Connection: Generally speaking, multicast protocols are based either on a single tree shared by all the members, or on several trees. That is, we distinguish between shared trees and source-based trees.

2. Aggregation: The aggregation is said to be greedy if a joining node connects to the closest node already in the group. This aggregation is said to be RPF if a joining node is connected to the group by an optimal path to the source.

3. Quality of Service: Some protocols try to optimize parameters, such as bandwidth delay etc. Although one cannot formally consider all QoS parameters,

some are more concerned with certain aspects of QoS than others.

4. Construction: Some trees are based on an underlying unicast protocol; some use a Breadth-First Search technique; others use a pruned spanning tree of the network; while others explore multiple paths and keep the best path.

5. Loop: Some protocols can theoretically avoid a loop. For the others, either there are situations in which loops occur, or the existence of loops has not been proven.

Due to the nature of the constantly changing network environment, a multicast routing protocol must cope with the dynamic nature of computer networks [36]. Network dynamics are the result of a new router being installed or a link failure; changes of a link's status in the network, such as the change of the link's residual bandwidth or delay characteristics; or group membership changes due to members joining or leaving. Among these dynamic factors, the link's status and the group membership change frequently in the Internet environment. Coping with dynamic group membership changes, which is one of the most important issues in multicasting, has attracted the attention of several researchers [34][35][36][37][38].

Table 2.1: Properties of multicast protocols

	Connection	**Aggregation**	**QoS**	**Construction**	**Loop**
DVMRP	Source-based	RPF	No	Broadcast/Pruning	
MOSPF	Source-based	RPF	No	OSPF + Group	
CBT	Shared	RPF	No	Unicast	
PIM-SM	Shared & S-B	RPF	No	Unicast/Pruning	
YAM	Shared	Greedy	Yes	Multiple paths	
BGMP	Shared	RPF	No	BGP	
SM	Shared	RPF	No	Unicast	No
QoSMIC	Shared & S-B	Greedy	Yes	Multiple paths	No

Approaches to solving the dynamic multicast routing problem can be classified into two categories: static and dynamic [39]. The first is tree reconstruction or re-computation oriented, and normally belongs to thecentralized approach. This approach is more static in the sense that the major goal is to completely rebuild an optimal delivery tree for all members when triggered by pre-defined events or by a periodic re-computation signal. The second category is tree maintenance oriented, which distributed computes attachment path segment in general. This approach is more dynamic in the sense that it is on-demand-based and processes one request at time to incrementally attach newly arrived members to an existing tree without globally re-computing the whole multicast tree.

2.3 Multirate Multicasting

In the current environment, receivers are typically computers with a wide range of processing capabilities, possibly augmented by special purpose video processing hardware. As a result, some receivers can implement more complex decompression algorithms at a higher frame rate or resolution than others. In addition, different receivers have different connection rates to the network. Data is sent from the source node and arrives at the receiver nodes at different rates depending on each receiver's bandwidth requirement. Connections to the Internet range from voice band modems of a few tens of kilobits per second for homes, to gigabits per second for large computer centers. In a pay-per-view system, pricing can also be used to encourage receivers to limit the demands that they place on the network. At present, most video broadcasts over the Mbone deliver the same signal to all receivers and operate conservatively so that all intended receivers can receive and decode the signal. In effect, everyone gets the grade of service of the least capable receivers.

Today's Internet lacks QoS support, which makes the transmission of real-time traffic challenging. Besides, the heterogeneity of the Internet makes the QoS control difficult. Consequently, we must adapt video traffic over networks to match various receivers' requirements and network conditions. Within this framework of bandwidth adaptation, we can envision the following three approaches to multicasting digital video [39].

1. The adaptive single stream approach

    The source uses feedback information to adapt its data rate. However, this may cause feedback implosion if there are a large number of receivers attempting to send feedback to the source. Although the single stream approach is the most straightforward, it is can not deal with the heterogeneity problem appropriately.

2. The adaptive replicated bit-stream approach

The source sends multiple bit-streams with the same video content, but with different quality levels and bit rates. Each bit-stream is multicast to a different multicast address, and receivers can join the group according to their own capabilities. Because a receiver's capabilities can change over time, the adaptive scheme must allow receivers to move among the different bit-streams. In addition, it has the problem of requiring the network to carry redundant information, because the video streams replicate each other.

3. The adaptive layered-video streams approach

This scheme relies on the ability of layered video compression schemes to divide their output bit-stream into layers: a base layer and one or more enhancement layers. The multicasting server can then send each layer to a different multicast group. A receiver joins one or more groups to adapt its capacity, thereby receiving different-quality video content. This approach provides the most efficient way to deal with the heterogeneity problem. But how to provide protocol support and deal with the increased complexity are still major problems.

In order to provide every receiver with only the bandwidth that it requests, we have to reduce the bandwidth of the signal as it passes through the network. M. Ghanbari [41] and F. Kishino et al.[42] used a two-layered coding scheme to extract critical video data. Ghanbari [41] proposed a method that divides the bit stream generated by a conditional-replenishment inter-frame coding technique into two parts. The first part contains the contents of the so-called 'guaranteed packets' and the second part holds the contents of the 'enhancement packets'. Guaranteed packets are transmitted on the guaranteed channel, whereas enhancement packets are transmitted without any guarantee. Kishino [42], proposed a DCT layered coding technique, which separates the DCT coefficients into MSP's (most significant parts) and LSP's (least significant parts), where MSP packets take priority over the LSP packets.

Therefore, this method can be implemented by using a progressive coder or by converting between encoding formats. An example of a progressive coder is a Fourier transform coder in which the high resolution components and low resolution components are placed in different packets. The low resolution signal can be transmitted to all receivers, while the high resolution components are only transmitted to those that request them. Similarly, progressive intra-frame coders can be designed to deliver 30, 15, or 5 frames per second, by marking the frames and not forwarding all of them along all of the branches. Consequently, we only need to consider the maximum requested bandwidth of each group that passes through the link, and aggregate those requests to determine how much should be paid to the network service provider for the link lease.

The MPEG-4 Fine-Granularity Scalable (FGS) [43][44][45][46] coding standard is an example of a scheme that encodes a video into a multiple bit-stream with multiple bit-rates. An FGS encoder encodes video data into more than one video stream, including one *base layer stream* and several *enhancement layer* streams. The base layer contains the most important portions of the video stream needed to achieve the minimum quality level. The enhancement layers contain the other portions of the video stream for refining the quality of the base layer stream.

Maxemchuk [47] discusses the issue of video distribution on multicast networks. This type of application requires more network bandwidth than e-mail or most information retrieval functions on the WWW. Maxemchuk's goal is to construct a minimum cost tree from the source to every destination, whereby destinations can request different bandwidth signals from the source. The source then transmits only one signal that is sufficient for the highest bandwidth destination. In this research, the author proposes an algorithm named M-T-M heuristic (Modified Takahashi-Matsuyama heuristic), which is a modification of the T-M heuristic (Takahashi-Matsuyama heuristic). However, the author's solution is heuristic-based. Obviously, it could be further optimized in his work. Charikar, Naor, and Schieber [48] extend this concept to present heuristics with provable performance guarantees for the Steiner tree problem in the rate model and the priority model. However, no simulation results are reported to justify the proposed approaches.

In Chapter 3, we discuss the same the problem as that in the Maxemchuk's research and further improve the results of the M-T-M heuristic. We now describe the T-M heuristic and M-T-M heuristic in detail.

The T-M heuristic operates in a similar manner to Solin's MST (Min-cost Spanning Tree) algorithm. At each step a receiver is added to the tree. The added receiver has the shortest path between itself and the current tree, just as the node that is added in Solin's algorithm has the shortest path. The difference between the two procedures is that the path in Solin's algorithm is a single link, allowing a straightforward search, while the path in the T-M heuristic may contain several links.

The T–M heuristic can be implemented as a combination of the MDT and the MST algorithms. The nodes that are permanently connected to the tree are assigned a depth of zero, as in the implementation of Solin's algorithm. Initially, only the source is permanently connected. At each step, the minimum depth algorithm is applied, and nodes are temporarily connected to the tree, until a receiver is temporarily connected. When this occurs, the links and nodes between the tree and the new receiver are made permanent and each permanent node is assigned a depth of zero. The other links and nodes that were temporarily connected are removed from the tree. This allows us to search for a shorter path from the nodes that were temporarily connected to the old tree and the new permanent tree. The operation of the algorithm with the source at node 1 and the receivers at nodes 2 and 4 is shown in Figure 2.2.

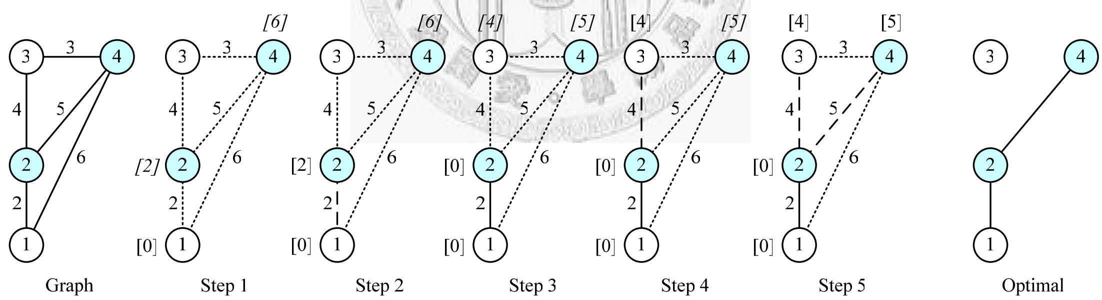


Figure 2.2: Example of the T-M heuristic for Steiner tree

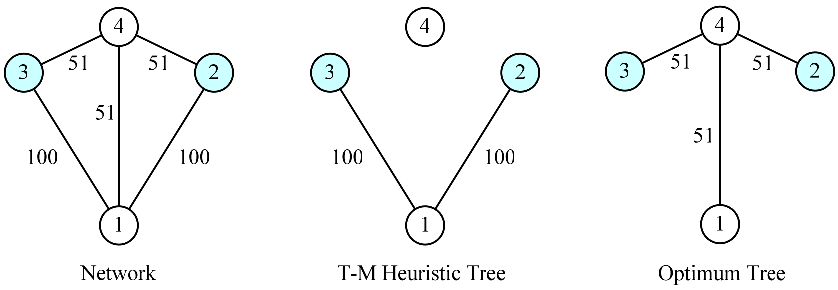


Figure 2.3: Example of the T-M heuristic

In the example in Figure 2.2, the T–M heuristic produces the optimal tree. However, this is not always the case, especially in multirate multicasting. Consider the network in Figure 2.3 with the source at node 1 and the receivers at nodes 2 and 3. The T–M heuristic results in the tree is 200. The optimum solution, which is also shown, is 153. The T–M heuristic is known to have a solution that is within a factor of two of the optimum; however, in most networks, the performance is much better.

Now, consider the network in Figure 2.4, which is a case of multirate multicasting with the source at node 1 and the receivers at nodes 2 and 3. The cost of a link is the basic cost of the link times the highest rate of the receiver that uses the link. The cost of the tree generated by the T–M heuristic is 28, and the minimum cost is 26. The steps of the M-T-M heuristic are:

1. Separate the receivers into subsets according to their rates.
2. Run the T–M heuristic on the subset with the highest requirements.
3. Once the tree with the subset of receivers with the highest requirement has been constructed, repeat the heuristic using this tree as the starting tree for the subset of receivers with the next highest set of requirements.
4. Repeat the procedure until all subsets of the receivers have been connected to the tree.

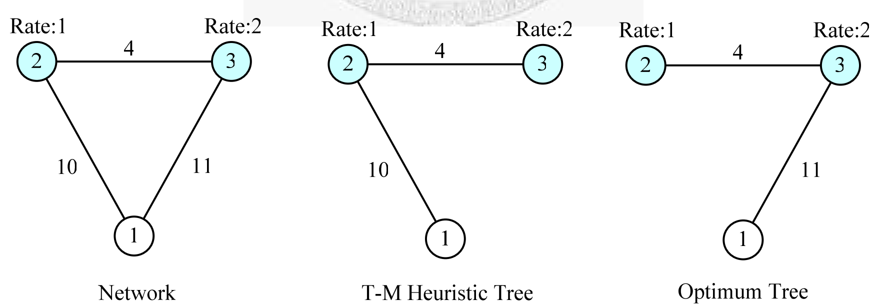


Figure 2.4: Example of the T-M heuristic

In a network with high and low bandwidth requirements, the receivers with the highest requirements are used to create a backbone of high bandwidth circuits. After the backbone has been created, the receivers in the low requirement subset that are not attached as part of the backbone are added as thinner branches of low bandwidth circuits. By following this procedure, the links that are added to the tree do not increase the requirements of the links that were previously assigned.

2.4 Admission Control

The objective of admission control is to ensure uninterrupted service provision to existing connections and, at the same time, accommodate new connection requests in an optimal way. Closely related to reserving resources is the technique of admission control in advance, usually at call setup time. Admission control handles the question of whether or not a network can accept a new connection [55]. The decision is based on (1) Would the new connection affect the QoS of the connections currently being carried by the network? (2) Can the network provide the QoS requested by the new connection? Once a request has been accepted, the required resources must be guaranteed. Admission control is used for Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services as a preventative scheme in congestion control [56].

Admission control is often considered a by-product of QoS routing and resource conservation. If such conservation is successful along the route(s) selected by the routing algorithm, the connection request is accepted; otherwise, it is rejected.

One approach to calculating the bandwidth to be allocated to a connection is statistical allocation, which takes advantage of statistical gains when multiplexing a number of bursty sources on a single link. A variety of algorithms proposed in the literature are based on different approximations or types of bandwidth allocation schemes that do not require complicated queuing solutions. The effective bandwidth algorithm is one such scheme.

Cetinkaya and Knightly [50] propose a method for performing admission control based on passive measurement, where routers monitor the passing traffic. When the routers receive a set-up request, they decide whether or not to provide the service based on the collected estimates about current resource usage. This technique is less precise than the active measurement approach in the estimation of available resources. It also requires that each router can perform admission control. Meanwhile, Lai and Baker [51] adopt the active measurement technique, which is used to estimate the capacity of the bottleneck link along a path. However, none of the above solutions have mechanisms to deal with multicast communications.

For effective resource management, one needs to find the key relationship between the traffic descriptor of users and the resources necessary to support the desired QoS. Effective bandwidth is the minimum bandwidth required by the connection to accommodate its desired QoS requirement. The notion of effective bandwidth provides a practical framework for admission control and capacity planning in high-speed communication networks [57][58][59].

Firoiu and Towsley [52] decompose the problem of admission control into the following subproblems: the division of end-to-end QoS requirements into local QoS requirements, the mapping of local QoS requirements into resource requirements, and the reclaiming of the resources allocated in excess of requirements. The authors solve the independent subproblems by a set of mechanisms and policies that provide admission control and resource conservation for multicast connection establishment. However, since route establishment is an important part of the connection process, the solution would be better if it considered the routing and admission control problems jointly.

Jia, Zhang, Pissinou, and Makki [53] propose a real-time multicast connection setup mechanism that integrates multicast routing with real-time admission control. It performs real-time admission experiments on a cost optimal tree (COT) and a shortest path tree (SPT) in parallel to optimize the network cost of the routing tree under real-time constraints. This approach has the following important features: (1) it is fully distributed; (2) it achieves a sub-optimal network cost for routing trees; and (3) it takes less time and exchanges fewer messages for a connection setup. However, the link costs of the network are fixed, whereas in our model, the link costs are dependent on the set of destinations that share the link.

Pagani and Rossi [54] propose a call admission multicast protocol (CAMP) that provides bandwidth guarantees to multicast applications with dynamic changes of the destination group membership. The authors prove that the protocol terminates, and thereby avoids a destination making an incorrect decision. Simulation results show that the mechanism effectively performs admission control; however, the authors do not consider the properties of heterogeneous destinations.

Tang, Tsui, and Wang [58] describe three basic components of admission control schemes (see Figure 2.5): traffic descriptors, admission criteria, and measurement processes. Whether a request will be accepted or not depends on these three factors. However, most of their research focuses on “measurement processes”.

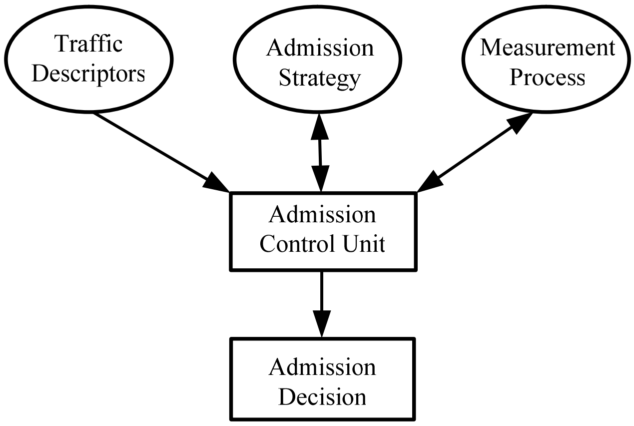


Figure 2.5: The relationship between basic components of admission control schemes

From our review of the above works, we believe that QoS for the broadband Internet should consider, three closely-related mechanisms, namely, admission control, routing, and resource reservation jointly. Furthermore, a novel admission control mechanism such as “partial admission control” that collocates other components would enhance QoS enormously.

2.5 Lagrangean Relaxation Method

Optimization plays an important role in application fields. In engineering, for instance, design tasks are routinely cast as optimization problems and algorithms are applied to search for parameters. Actually, optimization techniques could be widely used to address a number of problems found in computer networks, such as traffic routing challenges that have recently emerged with the arrival of connection-oriented architectures. In this dissertation, network planning and operation problems are modeled as mathematical problems that are computationally hard and for which no polynomial-time algorithm is known. The major approach applied to solving the optimization problem is a Lagrangean relaxation technique, which is expected to yield near-optimal solutions in a reasonable time.

Lagrangean methods were originally used in both scheduling and general integer programming problems. However, it has become one of the best tools for solving optimization problems like integer programming, linear programming combinatorial optimization, and non-linear programming. Adopting Lagrangean relaxation as our approach has the following advantages:

1. The approach is very flexible, since it is often possible to divide and conquer models in several ways and properly apply Lagrangean relaxation to each subproblem.

2. In decomposing problems, Lagrangean relaxation solves primal problems as individual components. Consequently, the solution approach permits us to exploit any known methodology or algorithm to solve the problem.

3. We can use Lagrangean relaxation methods to devise effective heuristic solutions to solve complex combinatorial optimization problems and integer problems.

Lagrangean relaxation also permits us to remove constraints from the original problem and place them in the objective function with associated Lagrangean multipliers instead. The optimal value of the relaxed problem is always a lower bound (for minimization problems) on the objective function value of the problem. By adjusting the multiplier of Lagrangean relaxation, we can obtain the upper and lower bounds of the problem. Although the Lagrangean multiplier problem can be solved in a variety of ways, the subgradient optimization technique is probably the most popular approach.

We now present an example of an optimization problem (P). By relaxing constraint $Ax=b$, the original primal problem (P) is transformed into an LR problem, where $Z_D(v) \leq Z$. In other words, the solution of (LR) is a lower bound of the primal problem (P).

$$Z = \min \ cx \tag{P}$$

subject to:

$$Ax = b$$

$$Dx \leq e$$

$$x \geq 0$$

x in *integral.*

$$Z_D(v) = \min cx + v(Ax-b) \qquad \text{(LR)}$$

subject to:

$$Dx \leq e$$

$$x \geq 0$$

x in *integral.*

With respect to the optimization problem (LR), we denote $V = (v^1, v^2, \ldots) \geq 0$ as the vector of Lagrangean multipliers with respect to relaxed constraints. According to the weak Lagrangean duality theorem, for any $V \geq 0$, the objective value of $Z_D(V)$ is a lower bound (LB) of Z_p. Thus, the dual problem (D) is constructed to calculate the tightest LB by adjusting multipliers, subject to $V \geq 0$. Then, the sub-gradient method is used to solve the dual problem. Let the vector S be a sub-gradient of $Z_D(V)$ at $V \geq 0$. In iteration k of the sub-gradient optimization procedure, the multiplier vector is updated by $\omega^{k+1} = \omega^k + t^k s^k$. The step size, t^k, is determined by $t^k = \delta(Z^*_{IP} - Z_D(\omega^k)/\|s^k\|^2)$, where Z^*_{IP} is an upper bound (UB) of the primal objective function value after iteration k; and δ is a constant, where $0 < \delta \leq 2$. To calculate the UB of (P), an algorithm to find primal feasible solutions must be developed. The maximum number of iterations and the improvement counter for the problem are decided on a case-by-case basis. We present our experiment settings in each chapter. The parameter δ adopted in the sub-gradient method is initialized to be 2, which is halved when the dual objective function value does not improve for improvement counter iterations.

$$Z_D = \max Z_D(V) \qquad (D)$$

To better describe how the dual problem is solved, the detailed concept adapted from [67] is illustrated in Figure 2.6.

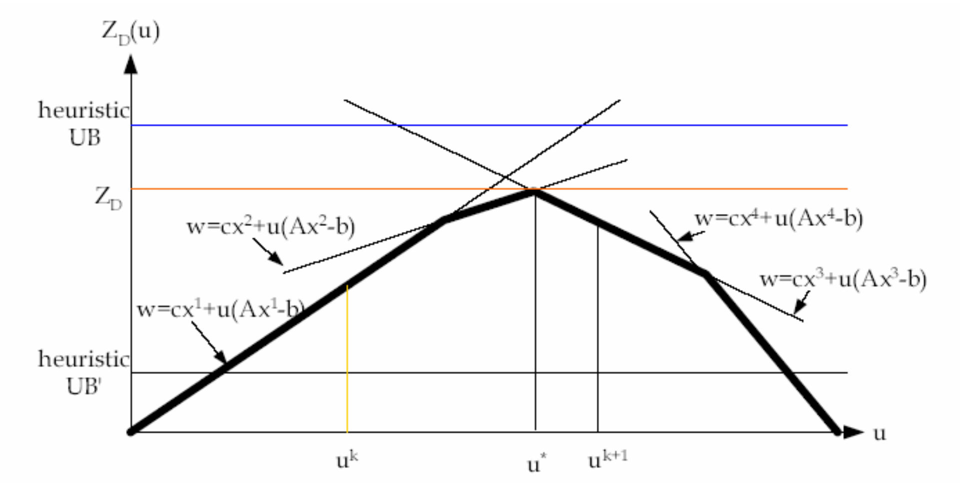


Figure 2.6: The concept of the dual problem

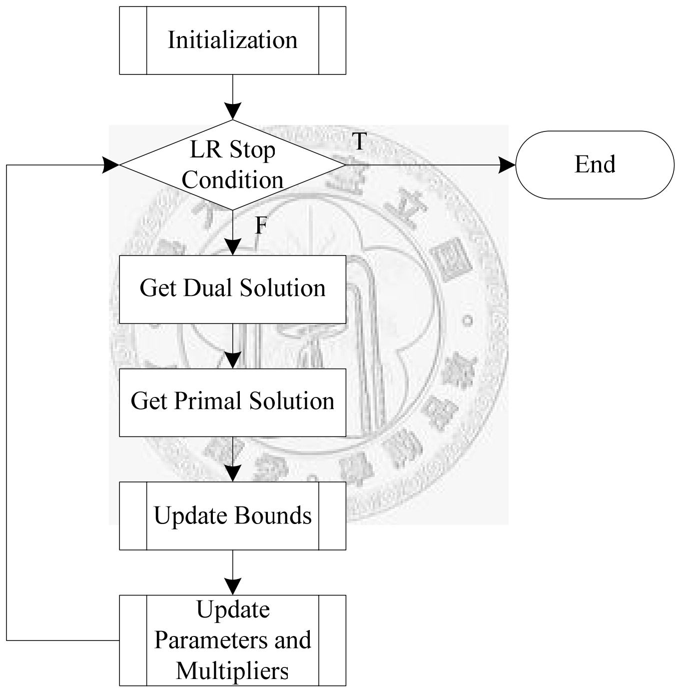


Figure 2.7: The overall procedure of the LR approach

The overall procedure of the LR approach is shown in Figure 2.7, but the algorithms to find primal feasible solutions must still be developed. After optimally solving the dual problem (D), we get a set of decision variables. However, this solution is not feasible for the primal problem, since some of constraints are not satisfied. Thus, minor modifications of the decision variables must be made to get a primal feasible solution for problem (P). Generally speaking, the UB of problem (P) is the better primal solution, while the solution of problem (D) guarantees the LB of problem (P). Iteratively, by solving the Lagrangean dual problem and getting a primal feasible solution, we get the LB and UB, respectively. So, the error gap between UB and LB, computed by (UB-LB)/LB*100%, illustrates the optimality of the solution. The smaller the gap computed, the better the optimality achieved. The algorithms proposed in this dissertation are coded in C++ and run on a PC with an INTEL P4-2.0Ghz CPU and a 1G MB RAM.

2.6 Network Topologies for Experiments and Simulations

In [74], the authors quantify the structure properties of networks by their characteristic path length *L* and clustering coefficient *C*. *L* measures the typical separation between two vertices in the network (a global property), whereas *C* measures the cliquishness of a typical neighborhood (a local property).

Characteristic path length *L* can be calculated by summing over the shortest path between any two vertices, averaged over all pairs of vertices.

$$L = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i>J} d_{ij} \qquad (L)$$

Clustering coefficient *C* is the mean probability that two vertices that are network neighbors of the same other vertex will themselves is neighbors.

$$C_i = \frac{\text{number of triangles connected to vertex } i}{\text{number of triples centered on vertex } i} \qquad (C_i)$$

$$C = \frac{1}{n} \sum_i C_i \qquad (C)$$

The networks of interest to us have many vertices with sparse connections, but not so sparse that the graph is in danger of becoming disconnected. In most kinds of networks, there are at least a few different types of vertices, and the probability of connection between vertices often depends on types. In order to test our proposed heuristic optimality, we test the heuristics on different categories of network topology, such as regular networks, random networks, and scale-free networks. The criteria of categorizing networks are based on characteristic path length *L* and clustering coefficient *C*.

For regular networks, such as grid networks, every vertex connects with a well-defined set of closest neighbors. Regular networks are characterized by high clustering coefficient and high characteristic path length. For random networks, every vertex has the same probability of being connected to any other vertex. Random networks are characterized by low clustering coefficient and low characteristic path length. The scale-free networks, which are power-law networks, are characterized by high clustering coefficient and low characteristic path length.

The classic models of random networks were defined to study properties of typical graphs among those with a given number of vertices. The model most commonly used for this purpose was introduced by Gilbert [75]. In Gilbert's model, $G(n,p)$, the $n(n-1)/2$ potential edges of a simple undirected graph with n vertices are included independently with probability $0 < p < 1$. This edge probability is usually chosen dependent on the number of vertices, ie. $p = p(n)$. The number of edges of a graph created according to the model $G(n,p)$ is not known in advance. The closely related model $G(n,m)$, in which all simple undirected graph with n vertices and exactly $0 \le m \le n(n-1)/2$ edges are equiprobable, was introduced by Erdős and Rényi.

In our experiments and simulations, we used Gilbert's $G(n,p)$ model to generated random networks. In [75], the authors proposed an efficient algorithm to generate random networks. The algorithm is showed as follow. We set p=2%.

```
[Random Networks]
Input: number of vertices n
       Edge probability 0<p<1
Output: G = ({0,…, n-1}, E)

E ← Ø
V ← 1; w ← -1
while v < n do
    draw r∈ [0,1) uniformly at random
    w ← w + 1 + ⌊log(1 − r) / log(1 − p)⌋
    while w ≥ v and v < n do
        w ← w - v; v ← v + 1
    if v < n then E ← E∪ {v,w}
```

Figure 2.8 shows the degree distribution of random networks (exponential networks) and scale-free networks (power-law networks). Poisson distribution is the distribution with bell-shape and exponential-tail, however, the power-law distribution is the one with heavy-tail. In Figure 2.8, we can see that the distribution decays exponentially for Poisson, binomial and normal distribution and the distribution decays more slowly for power-law distribution.

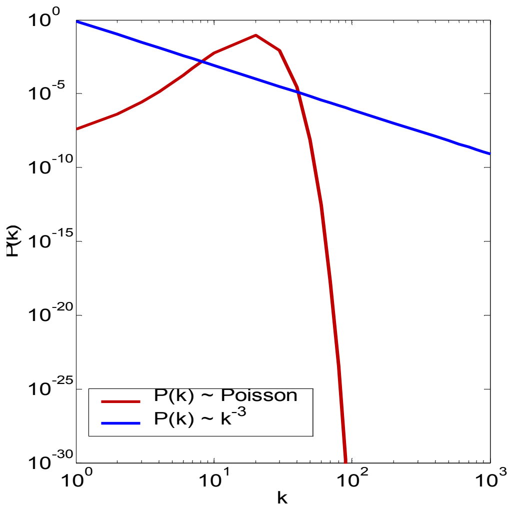


Figure 2.8: Degree distribution of exponential networks and power-law networks

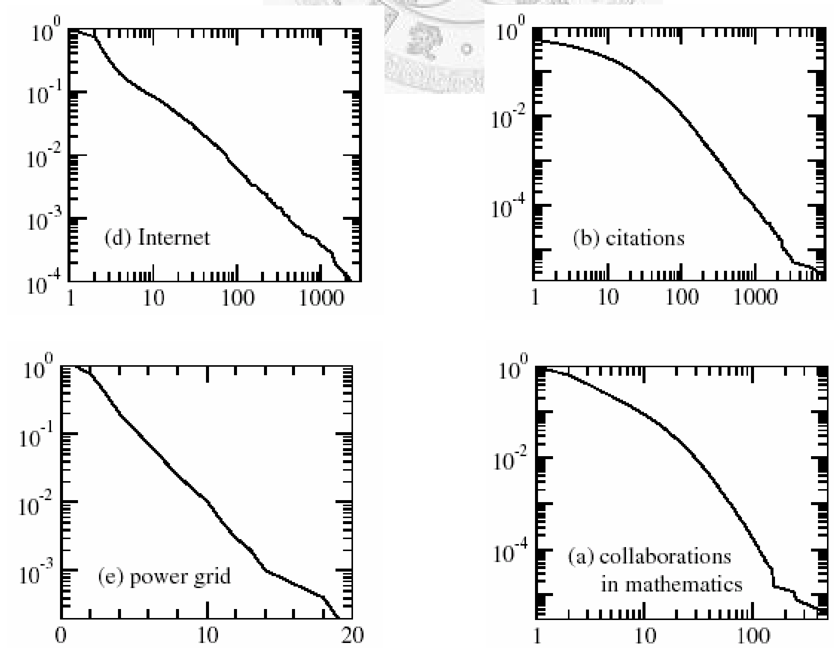


Figure 2.9: Example networks with power-law distribution

Reference [72] shows that the topology of the Internet is characterized by power laws distribution. The power laws describe concisely skewed distributions of graph properties, such as the node degree. Figure 2.9 also shows some other example networks with power-law degree distribution.

Scale-free networks also have two important characteristics: growth and preferential attachment. Growth means that the number of vertices N is not fixed. Networks continuously expand by the addition of new vertices. Preferential attachment means that the attachment is not uniform. A vertex is linked with higher probability to a vertex that already has a large number of edges.

In [73], the authors propose a method to construct a scale-free network. Firstly, to incorporate the growing character of the network, starting with a small number (m_0) of vertices, at every time step we add a new vertex with m ($\leq m_0$) edges that link the new vertex to m different vertices already present in the system. To incorporate preferential attachment, each edge connects with a vertex in the netgwork according to a probability Π_i proportional to the connectivity k_i of the vertex, where $\Pi(k_i) = k_i / \Sigma_j k_j$. After t time steps, the model leads to a random network with $t+m_0$ vertices and mt edges. The result is a network with degree distribution $P(k) \sim k^{-r}$. In [75], the authors proposed an efficient algorithm to implement preferential attachment. The algorithm is showed as follow. We set (m_0, m) to (2, 2) in our experiments.

```
[Preferential Attachment]
Input: number of vertices n
       minimum degree d≥1
Output: scale-free multigragh
        G = ({0,…, n-1}, E)

M: array of length 2nd
for v = 0 to n-1 do
    for i = 0 to d-1 do
        M[2(vd+1)] ← v
        draw r∈{0,…, 2(vd+1)} uniformly at random
        M[2(vd+1)+1] ← M[r]

E ← Ø
for i = 0,…,nd-1 do
    E ← E∪{M[2i], M[2i+1]}
```

CHAPTER 3 MINIMUM-COST MULTIRATE MULTICASTING ROUTING PROBLEM

3.1 Overview

Multimedia application environments are characterized by large bandwidth variations due to the heterogeneous access technologies of networks (e.g. analog modem, cable modem, xDSL, and wireless access etc.) and different receivers' quality requirements. In video multicasting, the heterogeneity of the networks and destinations makes it difficult to achieve bandwidth efficiency and service flexibility. There are many challenging issues that need to be addressed in designing architectures and mechanisms for multicast data transmission [60].

Unicast and multicast delivery of video are important building blocks of Internet multimedia applications. Unicast means that the video stream goes independently to each user through point-to-point connection from the source to each destination, and all destinations get their own stream. Multicast means that many destinations share the same stream through point-to-multipoint connections from the source to every destination, thus reducing the bandwidth requirements and network traffic. Consider the network in Figure 3.1(a), where node *S* is the source and nodes D_1, D_2, D_3, and D_4 are the receivers. Node D_1 requests 2 Mbps video stream and nodes D_2, D_3, and D_4 each request 0.5 Mbps video stream. Figure 3.1(b) shows the result of using four separate point-to-point connections, each for a different destination. Figure 3.1(c) shows the result of using two point-to-multipoint multicast connections, each for a category of traffic requirement. The efficiency of multicasting is achieved at the cost of losing the service flexibility of unicast, because in unicast each destination can individually negotiate a service contract with the source.

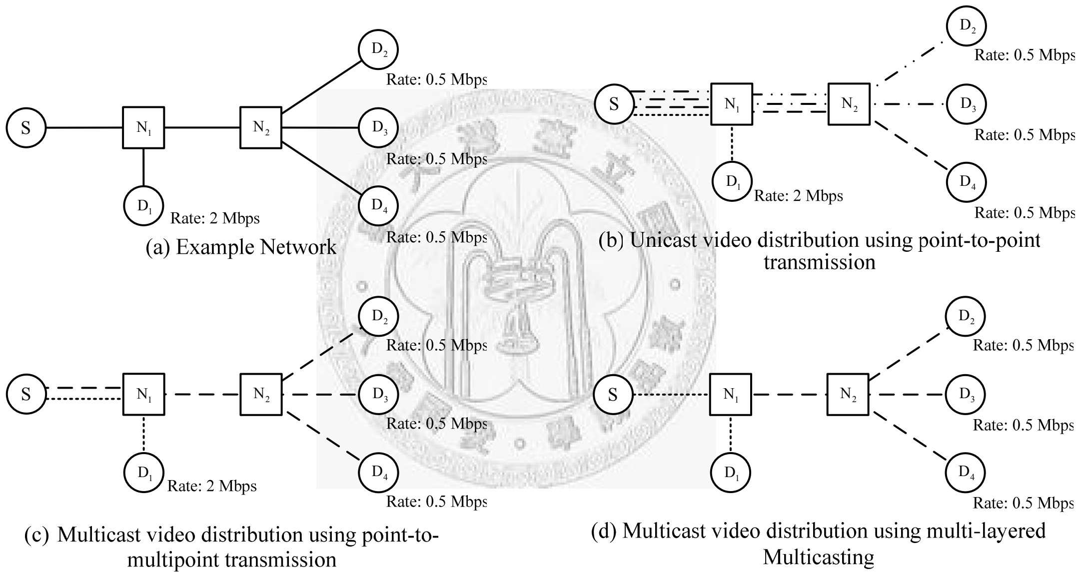


Figure 3.1: Video distribution

Taking advantage of recent advances in video encoding and transmission technologies, either by a progress coder [61] or video gateway [62][63], different destinations can request a different bandwidth requirement from the source. The source then only needs to transmit signals that are sufficient for the highest bandwidth destination into a single multicast tree. This concept is called single-application multiple-stream (SAMS). A multi-layered encoder encodes video data into more than one video stream, including one *base layer* stream and several *enhancement layer* streams. The base layer contains the most important portions of the video stream for achieving the minimum quality level. The enhancement layers contain the other portions of the video stream for refining the quality of the base layer stream. For instance, in the example in Figure 1(d), the base layer contains the video stream encoded at 0.5 Mbps for all destinations, and the enhancement layer contains the video stream encoded at 1.5 Mbps for destination D_1. When N_1, which is an advanced intermediate device like a multi-layered capable video gateway, receives both video steams from the source, *S*, it transfers the base layer and the enhancement layer to D_1, but only transfers the base layer to the other destinations, according to the pre-established routing decision. This mechanism is similar to destination-initiated reservations and packet filtering used in the RSVP protocol [64].

The minimum cost multicast tree problem, which is the Steiner tree problem, is known to be NP-complete. The Steiner tree problem is different to the minimum spanning tree problem in that it permits us to construct, or select, intermediate connection points to reduce the cost of the tree. References [65] and [66] survey the heuristics of Steiner tree algorithms.

For the conventional Steiner tree problem, the link costs in the network are fixed. However, for the minimum cost multi-layered video multicast tree, the link costs are dependent on the set of receivers sharing the link. This is a variant of the Steiner tree problem. The heterogeneity of the networks and destinations makes it difficult to design an efficient and flexible mechanism for servicing all multicast group users.

Reference [47] discusses the issue of multi-layered video distribution in multicast networks and proposes a heuristic to solve the problem, namely: the modified T-M heuristic (M-T-M Heuristic). Its goal is to construct a minimum cost tree from the source to every destination. However, the reference [47] provides only experimental evidence for its performance. Reference [48] extends this concept and presents heuristics with provable performance guarantees for the Steiner tree problem and proof that the problem is NP-hard, even in the special case of broadcasting. From the results, the cost of the multicast tree generated by M-T-M heuristics is no more than 4.214 times the cost of an optimal multicast tree. However, no simulation results are reported to justify the approaches in [48]. The solution approaches described above are heuristic-based and could be further optimized. Consequently, for multimedia distribution on multicast networks, we intend to find multicast trees that have a minimal total incurred cost for multi-layered video distribution.

In this Chapter, we extend the idea of [47] to minimize the cost of a multi-layered multimedia multicast tree and propose two more precise procedures (the tie-breaking procedure and the drop-and-add procedure) to improve the solution quality of the M-T-M heuristic. Furthermore, we formally model this problem as an optimization problem. In the structure of mathematics, the models undoubtedly have the properties of linear programming problems. We apply the Lagrangean relaxation method and the subgradient method to solve the problems [67][68]. Properly integrating the M-T-M heuristics and the results of Lagrangean dual problems may be useful for improving the solution quality. In addition, the Lagrangean relaxation method not only obtains a good feasible solution, it also provides the lower bound of the problem solution, which helps verify the solution quality. We call this method Lagrangean Based M-T-M Heuristics.

The remainder of this Chapter is organized as follows. In Section 3.2, we describe the M-T-M heuristic in detail and present evidence that the M-T-M heuristic does not perform well under some often seen scenarios. We then propose two procedures to improve the solution quality. In Section 3.3, we formally define the problem being studied, and propose a mathematical formulation of min-cost optimization is proposed. Section 3.4 applies Lagrangean relaxation as a solution approach to the problem, and Section 3.5, illustrates the computational experiments.

3.2 Heuristics of Multirate Multimedia Multicasting

Reference [69] proposes an approximate algorithm, called the T-M heuristic, to deal with the Steiner tree problem, which is a min-cost multicast tree problem. The T-M heuritic uses the concept of the minimum depth tree (MDT) algorithm to construct the tree. Initially, the source node is added to the tree permanently, and then, each iteration of MDT, a node is temporarily added to the tree until the added node is a receiver of the multicast group. Once the iterated tree reaches one of the receivers of the multicast group, it removes all unnecessary temporary links and nodes added earlier and marks the remaining nodes as permanently connected to the tree. The depth of the permanently connected nodes is then set to zero and the iterations continue until all receivers are permanently added to the tree. In [47], the author gives examples of the performance of the T-M heuristic and shows that, in some cases, it does not produce an optimum tree.

Reference [47] modified the T-M heuristic to deal with the min-cost multicast tree problem in multi-layered video distribution. For multi-layered video distribution, which is different from the conventional Steiner tree problem, each receiver can request a different quality of video. This means that each link's flow on the multicast tree is different and is dependent on the maximum rate of the receiver sharing the link. The author proposes a modified version of the T-M heuristic (i.e., the M-T-M heuristic) to approximate the minimum cost multicast tree problem for multi-layered video distribution.

The M-T-M heuristic separates the receivers into subsets according to the receiving rate. First, the M-T-M heuristic constructs the multicast tree for the subset with the highest rate by using the T-M heuristic. Using this initial tree, the T-M heuristic is then applied to the subsets according to the order of the receiving rate from high to low. For further details of the M-T-M heuristic, please refer to reference [47].

3.2.1 Some Scenarios of the Modified T-M Heuristic

In most networks, the performance of the Modified T-M heuristic is better than the T-M heuristic in multi-layered video multicasting. But, in some scenarios, we have found that the M-T-M does not perform well.

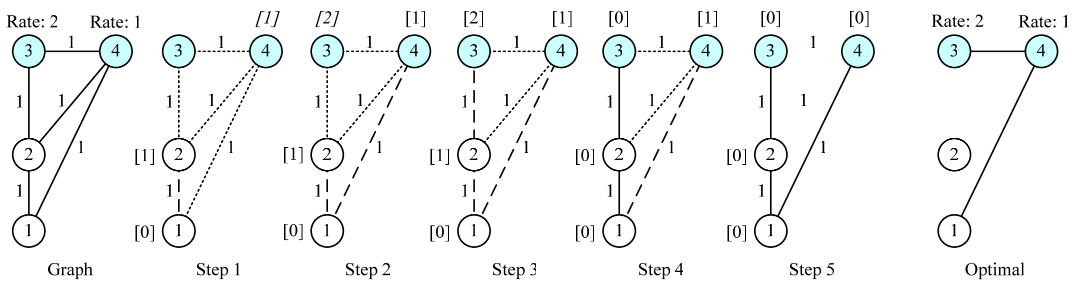


Figure 3.2: An example of the M-T-M heuristic for multi-layered video distribution with constant link costs.

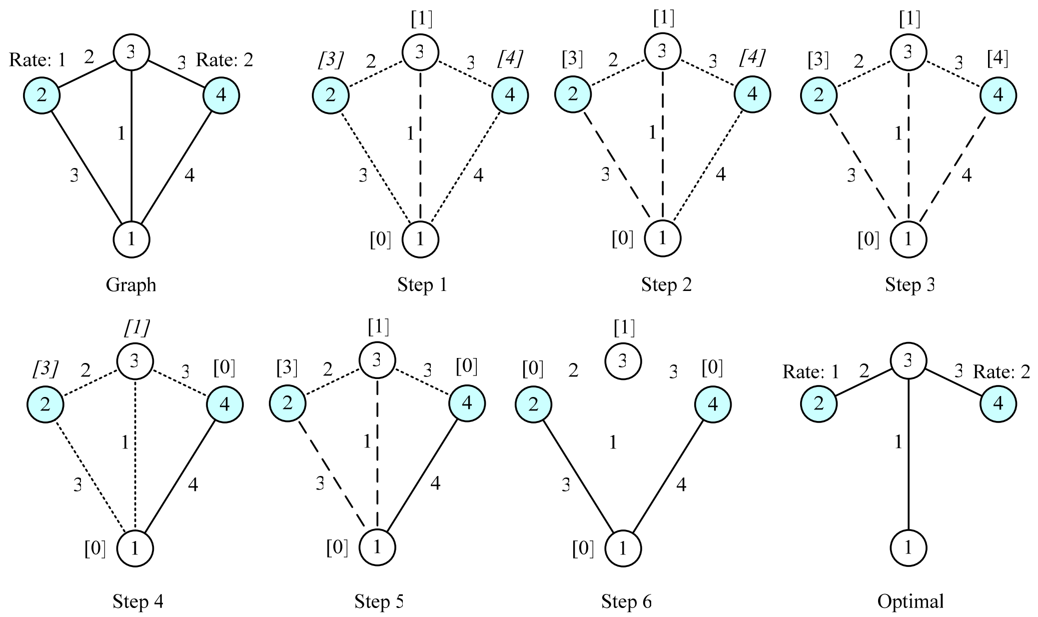


Figure 3.3: An example of the M-T-M heuristic for multi-layered video distribution with arbitrary link costs.

Consider the network in Figure 3.2, where with node 1 is the source and nodes 3 and 4 are the destinations requiring rates 2 and 1, respectively. Assume the base cost of all links is the same, which is 1. First, the M-T-M heuristic separates the receivers into two subsets, one for rate 1 and the other for rate 2. It then runs an MDT algorithm, such as Dijkstra algorithm, to construct the tree with the highest rate. At Step 4, the T-M heuristic reaches the destination with the highest rate and removes all unnecessary intermediate links. After setting the depth of the permanently connected nodes to zero, it continues the search process for the other destinations. At Step 5, the M-T-M heuristic tree is found and the sum of the link costs is 5; however, the sum of the link costs for the optimum tree shown is 4.

Consider the network in Figure 3.3, where node 1 is the source and nodes 2 and 4 are the destinations requiring rates 1 and 2, respectively. The link costs are indicated by the side of the links. At Step 4, the T-M heuristic reaches the destination with the highest rate and removes all unnecessary intermediate links. After setting the depth of the permanently connected nodes to zero, it continues the search process for the other destinations. At Step 6, the M-T-M heuristic tree is found and the sum of the link costs is 11; however, the sum of the link costs for the optimum tree shown is 10.

3.2.2 Enhanced Modified T-M Heuristic

With reference to the above scenarios, we propose two adjustment procedures to improve the solution performance. The first one is *the tie breaking procedure*, which is used to handle node selection when searching for the nearest node within the M-T-M heuristic. The second is *the drop-and-add procedure*, which is used to adjust the multicast tree resulting from the M-T-M heuristic in order to achieve a lower cost.

Tie Breaking Procedure. For the MDT algorithm, ties for the nearest distinct node may be broken arbitrarily, but the algorithm must still yield an optimal solution. Such ties are a sign that there may be multiple optimal solutions. All such optimal solutions can be identified by pursuing all ways of breaking ties to their conclusion. However, when executing the MDT algorithm within the M-T-M heuristic, we found that the tie breaking solution influences the cost of the multicast tree. For example in Figure 3.2, the depth of nodes 2 and 4 is the same and is minimal at Step 1. The tie may therefore be broken by randomly selecting one of them to be the next node to update the depth of all the vertices. In general, we choose the node with the minimal node number within the node set of the same minimal depth for simplicity of implementation. Although we choose node 1 as the next node to relax, node 2 is the optimal solution.

We therefore propose the following tie breaking procedure to deal with this situation. When there is a tie, the node with the largest requirement should be selected as the next node to join the tree. The performance evaluation will be presented in section 3.5.

Drop-and-Add Procedure. The drop-and-add procedure we propose is an adjustment procedure that adjusts the initial multicast tree constructed by M-T-M heuristic. Nevertheless, redundantly checking actions may cause a serious degradation of performance, even if the total cost is reduced. Therefore, we consider the most useful occurrence to reduce the total cost and control the used resources in an acceptable range. The steps of the procedure are:

1. Compute the number of hops from the source to the destinations.
2. Sort the nodes in descending order according to {incoming traffic / its own traffic demand}.

3. In accordance with the order, drop the node and re-add it to the tree. Consider the following possible adding measures and set the best one to be the final tree. Either add the dropped node to the source node, or to other nodes with the same hop count; otherwise, add the nodes with a hop count larger or smaller by one.

3.3 The Model

3.3.1 Problem Description

The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group is an application requesting transmission in the network, which has one source and one or more destinations. Given the network topology and bandwidth requirement of every destination of a user group, we want to jointly determine the following decision variables: (1) the routing assignment (a tree for multicasting or a path for unicasting) of each user group; and (2) the maximum allowable traffic rate of each multicast user group through each link.

By formulating the problem as a mathematical programming problem, we solve the issue optimally by obtaining a network that will enable us to achieve our goal, i.e., one that ensures the network operator will spend the minimum cost on constructing the multicast tree. The notations used to model the problem are listed in Table 3.1.

Table 3.1: Description of notations (MCMR)

Given Parameters	
Notation	**Description**
a_l	Transmission cost associated with link l
α_{gd}	Traffic requirement of destination d of multicast group g
G	The set of all multicast groups
V	The set of nodes in the network
L	The set of links in the network
D_g	The set of destinations of multicast group g
h_g	The minimum number of hops to the farthest destination node in multicast group g

I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of paths that destination d of multicast group g may use
δ_{pl}	The indicator function, which is 1 if link l is on path p and 0 otherwise
Decision Variables	
Notation	**Description**
x_{gpd}	1 if path p is selected for group g destined for destination d, and 0 otherwise
y_{gl}	1 if link l is on the sub-tree adopted by multicast group g, and 0 therwise
m_{gl}	The maximum traffic requirement of the destinations in multicast group g that are connected to the source through link l

3.3.2 Mathematical Formulation

According to the problem description in the pervious section, the min-cost problem is formulated as a combinatorial optimization problem in which the objective function is to minimize the link cost of the multicast tree. Of course, a number of constraints must be satisfied.

Objective function:

$$Z_{IP3} = \min \sum_{g\in G}\sum_{l\in L} a_l\ m_{gl} \qquad \text{(IP 3)}$$

subject to:

$$\sum_{p\in P_{gd}} x_{gpd}\alpha_{gd}\delta_{pl} \le m_{gl} \qquad \forall g\in G, d\in D_g, l\in L \qquad (3.1)$$

$$m_{gl} \in [0, \max_{d\in D_g} \alpha_{gd}] \qquad \forall l\in L, g\in G \qquad (3.2)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l\in L, g\in G \qquad (3.3)$$

$$\sum_{l\in L} y_{gl} \ge \max\{h_g, |D_g|\} \qquad \forall g\in G \qquad (3.4)$$

$$\sum_{d \in D_g} \sum_{p \in P_{gd}} x_{gpd} \delta_{pl} \le |D_g| y_{gl} \qquad \forall g \in G, l \in L \qquad (3.5)$$

$$\sum_{l \in I_v} y_{gl} \le 1 \qquad \forall g \in G, v \in V - \{r_g\} \qquad (3.6)$$

$$\sum_{l \in I_{r_g}} y_{gl} = 0 \qquad \forall g \in G \qquad (3.7)$$

$$\sum_{p \in P_{gd}} x_{gpd} = 1 \qquad \forall d \in D_g, g \in G \qquad (3.8)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall d \in D_g, g \in G, p \in P_{gd} . \qquad (3.9)$$

The objective function of (IP 3) is to minimize the total transmission cost of servicing the maximum bandwidth requirement destination through a specific link for all multicast groups *G*, where *G* is the set of user groups requesting connection. The maximum bandwidth requirement on a link in a specific group m_{gl} can be viewed so that the source would be required to transmit in a way that matches the most constrained destination.

Constraint (3.1) is referred to as the capacity constraint, where the variable m_{gl} can be interpreted as the "estimate" of the aggregate flow. Since the objective function is strictly an increasing function with m_{gl} and (IP 3) is a minimization problem, each m_{gl} will be equal to the aggregate flow in an optimal solution. Constraint (3.2) is a redundant constraint that provides upper and lower bounds on the maximum traffic requirement for multicast group *g* on link *l*. Constraints (3.3) and (3.4) require that the number of links on the multicast tree adopted by multicast group *g* be at least the maximum of h_g and the cardinality of D_g. The h_g and the cardinality of D_g are the legitimate lower bounds of the number of links on the multicast tree adopted by the multicast group *g*. Constraint (3.5) is called the tree constraint, which requires that the union of the selected paths for the destinations of user group *g* forms a tree. Constraints (3.6) and (3.7) are both redundant constraints. Constraint (3.6) requires that the number of selected incoming links, y_{gl}, to a node is 1 or 0, while constraint (3.7) requires that there are no selected incoming links, y_{gl}, to the node that is the root of multicast group *g*. As a result, the links we select can form a tree. Finally, constraints (3.8) and (3.9) require that only one path is selected for each multicast source/destination pair.

3.4 Solution Approach

3.4.1 Lagrangean Relaxation

By using the Lagrangean Relaxation method, we can transform the primal problem (IP 3) into the following Lagrangean Relaxation problem (LR 3) where Constraints (3.1) and (3.5) are relaxed. For a vector of non-negative Lagrangean multipliers, a Lagrangean Relaxation problem of (IP 3) is given by

Optimization problem (LR):

$$Z_{D3}(\beta,\theta)=\min \sum_{g\in G}\sum_{l\in L}a_l m_{gl}+\sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L}\sum_{p\in P_{gd}}\beta_{gdl}x_{gpd}\alpha_{gd}\delta_{pl}-\sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L}\beta_{gdl}m_{gl}$$
$$+\sum_{g\in G}\sum_{l\in L}\sum_{d\in D_g}\sum_{p\in P_{gd}}\theta_{gl}x_{gpd}\delta_{pl}-\sum_{g\in G}\sum_{l\in L}\theta_{gl}\left|D_g\right|y_{gl} \quad \text{(LR 3)}$$

subject to: (3.2) (3.3) (3.4) (3.6) (3.7) (3.8) (3.9).

Where β_{gdl},θ_{gl} are Lagrangean multipliers and $\beta_{gdl},\theta_{gl}\geq 0$. To solve (LR 3), we decompose (LR 3) into the following three independent and easily solvable optimization subproblems.

Subproblem 1: for decision variable x_{gpd}

$$Z_{Sub3.1}(\beta,\theta)=\min \sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}[\sum_{l\in L}\delta_{pl}(\beta_{gdl}\alpha_{gd}+\theta_{gl})]x_{gpd} \quad \text{(SUB 3.1)}$$

subject to:

$$\sum_{p\in P_{gd}}x_{gpd}=1 \qquad \forall d\in D_g, g\in G \quad (3.8)$$

$$x_{gpd}=0 \text{ or } 1 \qquad \forall d\in D_g, g\in G, p\in P_{gd}. \quad (3.9)$$

Subproblem (SUB 3.1) can be further decomposed into $|G||D_g|$ independent shortest path problems with nonnegative arc weights. Each shortest path problem can be easily solved by Dijkstra's algorithm.

Subproblem 2: (related to decision variable y_{gl})

$$Z_{Sub3.2}(\theta) = \min \sum_{g \in G} \sum_{l \in L} (-\theta_{gl} |D_g|) y_{gl} \tag{SUB 3.2}$$

subject to:

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l \in L, g \in G \tag{3.3}$$

$$\sum_{l \in L} y_{gl} \geq \max\{h_g, |D_g|\} \qquad \forall g \in G \tag{3.4}$$

$$\sum_{l \in I_v} y_{gl} \leq 1 \qquad \forall g \in G, v \in V - \{r_g\} \tag{3.6}$$

$$\sum_{l \in I_{r_g}} y_{gl} = 0 \qquad \forall g \in G. \tag{3.7}$$

Algorithm 3.1 that optimally solves Subproblem (SUB 3.2) is:

[Algorithm 3.1]

Step 1 Compute $\max\{h_g, |D_g|\}$ for multicast group g.

Step 2 Compute the number of negative coefficients, $(-\theta_{gl} |D_g|)$, for all links in the multicast group g.

Step 3 If the number of negative coefficients is greater than $\max\{h_g, |D_g|\}$ for multicast group g, then assign the corresponding negative coefficient of y_{gl} to 1, and 0 otherwise.

Step 4 If the number of negative coefficients is no greater than $\max\{h_g, |D_g|\}$ for multicast group g, assign the corresponding negative coefficient of y_{gl} to 1. Then, assign [$\max\{h_g, |D_g|\} -$ the number of positive coefficients of y_{gl}] numbers of the smallest positive coefficient of y_{gl} to 1, and 0 otherwise.

Subproblem 3: (related to decision variable m_{gl})

$$Z_{Sub3.3}(\beta) = \min \sum_{g\in G}\sum_{l\in L}(a_l - \sum_{d\in D_g} \beta_{gdl})m_{gl} \qquad \text{(SUB 3.3)}$$

subject to:

$$m_{gl} \in [0, \max_{d\in D_g} \alpha_{gd}] \qquad \forall l \in L, g \in G. \qquad (3.2)$$

We decompose Subproblem (SUB3.3) into $|L|$ independent problems. For each link $l \in L$:

$$Z_{Sub3.3.1}(\beta) = \min \sum_{g\in G}(a_l - \sum_{d\in D_g} \beta_{gdl})m_{gl} \qquad \text{(SUB 3.3.1)}$$

subject to: (3.2).

The algorithm to solve Subproblem (SUB 3.3.1) is:

Step 1 Compute $a_l - \sum_{d\in D_g} \beta_{gdl}$ for link *l* of multicast group *g*.

Step 2 If $a_l - \sum_{d\in D_g} \beta_{gdl}$ is negative, assign the corresponding m_{gl} to the maximum traffic requirement in the multicast group, otherwise assign the corresponding m_{gl} to 0.

According to the weak Lagrangean duality theorem [70], for any $\beta_{gdl}, \theta_{gl} \ge 0$, $Z_D(\beta_{gdl}, \theta_{gl})$ is a lower bound of Z_{IP}. The following dual problem (D) is then constructed to calculate the tightest lower bound.

Dual Problem:

$$Z_{D3} = \max Z_{D3}(\beta_{gdl}, \theta_{gl}) \qquad \text{(D3)}$$

subject to:

$$\beta_{gdl}, \theta_{gl} \ge 0.$$

Several methods can be used to solve the dual problem (D3), the most popular of which is the subgradient method [71], employed here. Let a vector, *s*, be a subgradient of $Z_D(\beta_{gdl}, \theta_{gl})$. Then, in iteration *k* of the subgradient optimization procedure, the multiplier vector is updated by $\omega^{k+1} = \omega^k + t^k s^k$. The step size, t^k, is determined by $t^k = \delta(Z^h{}_{IP} - Z_D(\omega^k) / \|s^k\|^2)$, $Z^h{}_{IP}$ is the primal objective function value for a heuristic solution (an upper bound on Z_{IP}), δ is a constant, and $0 < \delta \le 2$.

3.4.2 Getting Primal Feasible Solutions

After optimally solving the Lagrangean dual problem, we have a set of decision variables. However, this solution would not be feasible for the primal problem, since some constraints are not satisfied. Thus, minor modification of the decision variables, or the hints of the multipliers, must be considered in order to obtain the primal feasible solution of the problem (IP). Generally speaking, an upper bound (UB) of the problem (IP) is the better primal feasible solution, while the Lagrangean dual problem solution guarantees the lower bound (LB) of the problem (IP). Iteratively, by solving the Lagrangean dual problem and getting the primal feasible solution, we get the LB and UB, respectively. So, the gap between the UB and the LB, computed by (UB-LB)/LB*100%, illustrates the optimality of the problem solution. The smaller gap computed, the better the optimality.

To calculate the primal feasible solution of the minimum cost tree, we consider the solutions to the Lagrangean Relaxation problems. The set of $\{x_{gpd}\}$ obtained by solving Subproblem (SUB 1) may not be a valid solution for problem (IP 3), because the capacity constraint is relaxed. However, the capacity constraint may be a valid solution for some links. Also, the set of $\{y_{gl}\}$ obtained by solving Subproblem (SUB 2) may not be a valid solution, because of the link capacity constraint and the union of $\{y_{gl}\}$ may not be a tree.

Here, we propose a comprehensive, two-part method to obtain a primal feasible solution. It utilizes a Lagrangean-based modified T-M heuristic, followed by adjustment procedures. While solving the Lagrangean relaxation dual problem, we may obtain some multipliers related to each OD pair and links, which could make our routing more efficient. We describe the Lagrangean based modified T-M heuristic below.

[Lagrangean Multiplier-based modified T-M heuristic]

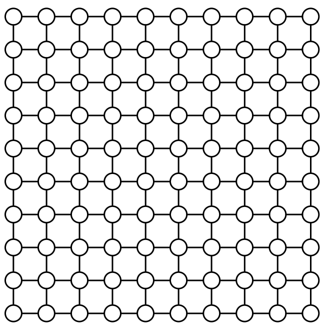
Step 1 Use $a_l - \sum_{d \in D_g} \beta_{dgl}$ as link *l*'s arc weight and run the M-T-M heuristic.

Step 2 After getting a feasible solution, apply *the drop-and-add procedure* described earlier to adjust the result.

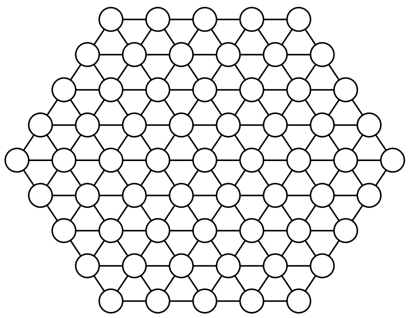
Initially, we set all the multipliers to 0, so we will the same routing decision as the M-T-M heuristics followed by the drop-and-add procedure at the first iteration.

3.5 Computational Experiments

In this section, computational experiments on the Lagrangean relaxation based heuristic and other primal heuristics are reported. The heuristics are tested on three kinds of networks: regular networks, random networks, and scale-free networks. Two regular networks, shown in Figure 3.4, are tested in our experiment. The first is a grid network containing 100 nodes and 180 links, and the second is a cellular network containing 61 nodes and 156 links. Random networks tested in this experiment are generated randomly. Each network has 500 nodes. The candidate links between all node pairs are given a probability following the uniform distribution. In the experiments, we link the node pair with a probability smaller than 2%. If the generated network is not a connected network, we generate a new network.



(a) Grid Network



(b) Cellular Network

Figure 3.4: Regular networks (MCMR)

Reference [73] shows that scale-free networks can be developed from a simple dynamic model that combines incremental growth with a preference for new nodes to connect to existing ones that are already well connected. In our experiments, we applied this preferential attachment method to generate the scale-free networks. The corresponding preferential variable (m_0, m) is (2, 2), and the number of nodes in each test network is 500.

Table 3.2: Parameters for Lagrangean relaxation (MCMR)

Number of Iterations	2,000
Initial Multipliers	0
Improvement Counter	25
Delta Factor	2
Optimal Condition	Gap < 0.001

For each test network, several distinct cases, which have different pre-determined parameters, such as the number of nodes, are considered. The traffic demands for each destination are drawn from a random variable that is uniformly distributed in pre-specified categories {1, 2, 5, 10, 15, 20}. The link costs are randomly generated between 1 and 5. The parameters used for all cases are listed in Table 3.2. The cost of the multicast tree is decided by multiplying the link cost and the maximum bandwidth requirement on a link. We conducted 2,000 experiments for each kind of network. For each experiment, the result was determined by the group destinations and link costs generated randomly. Table 3.3 summaries selected results of the computational experiments.

In general, the results of LR are all better than the M-T-M heuristic (MTM), the M-T-M heuristic with the tie breaking procedure (TB), and the M-T-M heuristic followed by the drop-and-add procedure (DA). The reason is that we get the same solution as the M-T-M heuristic in the first iteration of LR. For each test network, the maximum improvement ratio between the M-T-M heuristic and the Lagrangean-based modified T-M heuristic is 16.18 %, 23.23%, 10.41 %, and 11.02%, respectively. To claim optimality, we also depict the percentile of the gap in Table 3. The results show that 60% of the regular and scale free networks have a gap of less than 10%, but the results of random networks show a larger gap. However, we also found that the

M-T-M heuristic perform well in many cases, such as case D of the grid network and case D of the random network.

According to the experiment results, the tie breaking procedure we propose is not uniformly better than random selection. For example, in case H of the cellular network, the performance of M-T-M (1,517) is better than TB (1,572). Consequently, we suggest that, in practice, we can try both tie breaking methods (random selection or the method we propose), and take the better result. The experiment results also show that the drop-and-add procedure does reduce the cost of the multicast tree.

Table 3.3: Selected results of computational experiments (MCMR)

CASE	Dest. #	M-T-M	TB	DA	UB	LB	GAP	Imp.
Grid Network						Max Imp. Ratio: 16.18 %		
A	5	332	330	332	290	286.3714	1.27%	14.48%
B	5	506	506	506	506	503.6198	0.47%	0.00%
C	10	158	153	148	136	123.1262	10.46%	16.18%
D	10	547	547	547	547	541.8165	0.96%	0.00%
E	20	522	507	502	458	397.8351	15.12%	13.97%
F	20	1390	1405	1388	1318	1206.235	9.27%	5.46%
G	50	2164	2229	2154	1940	1668.448	16.28%	11.55%
H	50	759	700	759	693	588.3226	17.79%	9.52%
Cellular Network						Max Imp. Ratio: 23.23 %		
A	5	182	167	172	167	160.4703	4.07%	8.98%
B	5	119	119	119	109	105.9671	2.86%	9.17%
C	10	194	185	190	180	156.9178	14.71%	7.78%
D	10	174	174	170	150	138.0774	8.63%	16.00%
E	20	382	349	382	310	266.1146	16.49%	23.23%
F	20	815	800	811	756	689.6926	9.61%	7.80%
G	50	602	595	602	567	479.9626	18.13%	6.17%
H	50	1517	1572	1503	1357	1187.332	14.29%	11.79%
Random Networks						Max Imp. Ratio: 10.41 %		
A	5	107	107	107	107	94.70651	12.98%	0.00%
B	5	88	88	88	86	74.63349	15.23%	2.27%
C	10	170	170	170	170	134.6919	26.21%	0.00%
D	10	123	125	123	123	97.90988	25.63%	0.00%
E	20	317	317	317	284	221.2635	28.35%	10.41%
F	20	226	216	226	216	168.0432	28.54%	4.42%
G	50	850	860	850	806	558.5077	44.31%	5.18%
H	50	702	715	702	690	446.9637	54.37%	1.71%
Scale-Free Networks						Max Imp. Ratio: 11.02 %		
A	5	82	82	82	82	78.35047	4.66%	0.00%
B	5	79	75	75	75	73.70663	1.75%	5.33%
C	10	210	210	210	208	196.3969	5.91%	0.96%
D	10	528	528	528	506	505.4039	0.12%	4.35%
E	20	886	896	886	854	770.9776	10.77%	3.75%
F	20	1068	1050	1022	962	920.2371	4.54%	11.02%

G	50	1869	1871	1869	1754	1502.061	16.77%	6.56%
H	50	1911	1946	1911	1891	1598.817	18.27%	1.06%

M-T-M: The result of the modified T-M heuristic
TB: The result of the modified T-M heuristic with the tie breaking procedure
DA: The result of the modified T-M heuristic followed by the drop-and-add procedure
UB and LB: Upper and lower bounds of the Lagrangean-based modified T-M heuristic
GAP: The duality gap of Lagrangean relaxation
Imp.: The improvement ratio of the Lagrangean-based modified T-M heuristic {(M-T-M - UB)/ UB}

3.6 Concluding Remarks

In this chapter, we have attempted to solve the problem of min-cost multicast routing for multi-layered multimedia distribution. Our contribution can be expressed in terms of the mathematical formulation and experiment performance. For the formulation, we have proposed a precise mathematical expression, which models the problem efficiently. With regard to performance, the proposed Lagrangean relaxation and subgradient based algorithms outperform the primal heuristics (M-T-M heuristic) with acceptable computation time. According to the experiment results, the Lagrangean-based heuristic can achieve up to 23.23% improvement over the M-T-M heuristic. We have also proposed two adjustment procedures (the tie-breaking procedure and the drop-and-add procedure) to enhance the solution quality of the M-T-M heuristic.

Our model can be easily extended to deal with the constrained multicast routing problem for multi-layered multimedia distribution by adding capacity and delay constraints. Moreover, the min-cost model proposed in this chapter can be modified as a max-revenue model, with the objective of maximizing total system revenues by totally, or partially, admitting destinations into the system. These issues will be addressed in the following chapters.

CHAPTER 4 CAPACITATED MINIMUM-COST MULTIRATE MULTICAST ROUTING PROBLEM

4.1 Overview

The multicast routing problem discussed in the previous chapter only considers how to find a minimum cost routing solution for one or more multirate multicast groups. It does not consider capacity issues. Without considering the capacity constraints, a multiple multicast routing problem can be seen as several single multicasting routing problems, which can be solved by a single multicast routing algorithm, such as the algorithm proposed in the previous chapter. For constrained cases, because multimedia transmission is time sensitive, the bandwidth requirement must also be considered in multicast routing problems.

In general, most multiple-multicast routing problems consist of finding a set of routing trees that satisfy certain constraints and minimize the total tree cost. One of the constraints is the bandwidth constraint, which means the trees have to compute for limited bandwidth. Many existing multiple multicast routing algorithms use the following approach to obtain a feasible solution. Initially, for each multicast session is regard as a single multicast routing problem and solve it by single multicast routing algorithm. Then, a set of multicast trees will be obtained, but this solution may be infeasible. Therefore, some coordinative strategy on the solution trees must be adopted to meet the bandwidth constraint. The above approach to multiple multicast routing uses a single multicast routing algorithm as its underlying method. Therefore, solution approaches for multiple multicast routing problems are strongly related to single multicast routing methods.

In this chapter, we deal with the link-constrained multirate multicast optimization problem. We formally model this issue as an optimization problem, and apply the Lagrangean relaxation method and the subgradient method to solve the problem. Properly integrating the M-T-M heuristics and the results of the Lagrangean dual problems may be useful for improving the solution quality. In addition, the Lagrangean relaxation method not only finds a good feasible solution, but also provides the lower bound of the solution, which helps verify the solution quality. We call this method Lagrangean-multiplier-based Heuristics.

The remainder of this chapter is organized as follows. In Section 4.2, we describe our proposed simple heuristic in detail. It is composed of the M-T-M heuristic and the adjustment procedures to ensure that the link capacity constraint is not violated. In Section 4.3, we formally define the problem being studied, and propose a mathematical formulation of min-cost optimization. Section 4.4, we apply Lagrangean relaxation as a solution approach to the problem. Section 4.5, describes the computational experiments. Finally, in Section 4.6, we present our conclusions.

4.2 A Simple Heuristic for Link Constrained Multirate Multimedia Multicasting

Under the link capacity constraint, the routing decision generated by the M-T-M heuristic described in Chapter 3 may cause an overflow of the links. We propose an adjustment procedure (AP), which we use to adjust the multicast tree resulting from the M-T-M heuristic in order to find a feasible solution and comply with the link capacity constraint. Nevertheless, redundantly checking actions may cause a serious degradation in performance, even if the total cost is reduced. Therefore, we consider the most useful occurrence to reduce the total cost and control the used resources in an acceptable range. The details of the procedure are:

Adjustment Procedure (AP)

Step 1 Compute the aggregate flow of each link.

Step 2 Sort the links in descending order based on the difference between the aggregate flow of each link and the link's capacity.

Step 3 Choose the first link. If the difference value of the link is positive, go to Step 4; otherwise go to Step 6.

Step 4 Choose the maximal loaded group on that link to drop and re-add it to the tree. Consider the following possible adding measures and set the best one to be the final tree. Either add the dropped node to the source node, or to other nodes with the same hop count, or to the nodes with a hop count larger or smaller by one.

Step 5 If a feasible solution is found, go to Step2; otherwise go to Step 6.

Step 6 Stop.

The performance evaluation of the simple heuristic will be discussed in Section 4.5.

4.3 Problem Formulation

4.3.1 Problem Description

The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group is an application requesting transmission over the network, which has one source and one or more destinations. Given the network topology, the capacity of the links and the bandwidth requirement of every destination of a user group, we want to jointly determine the following decision variables: (1) the routing assignment (a tree for multicasting or a path for unicasting) of each user group; and (2) the maximum allowable traffic rate of each multicast user group through each link.

By formulating the problem as a mathematical programming problem, we solve it optimally by obtaining a network that enables us to achieve our goal, i.e., one that ensures the network operator incurs the minimum cost in constructing the multicast tree. The notations used to model the problem are listed in Table 4.1.

Table 4.1: Description of notations (C-MCMR)

Given Parameters	
Notation	**Description**
a_l	Transmission cost associated with link l
α_{gd}	Traffic requirement of destination d of multicast group g
G	The set of all multicast groups
V	The set of nodes in the network
L	The set of links in the network
D_g	The set of destinations of multicast group g
h_g	The minimum number of hops to the farthest destination node in multicast group g
C_l	The capacity of link l
I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of paths destination d of multicast group g may use
δ_{pl}	The indicator function, which is 1 if link l is on path p and 0 otherwise
Decision Variables	
Notation	**Description**
x_{gpd}	1 if path p is selected for group g destined for destination d, and 0 otherwise
y_{gl}	1 if link l is on the sub-tree adopted by multicast group g, and 0 otherwise
m_{gl}	The maximum traffic requirement of destinations in multicast group g that are connected to the source through link l

4.3.2 Mathematical Formulation

According to the description in previous section, the min-cost problem is formulated as a combinatorial optimization problem in which the objective function is to minimize the link cost of the multicast tree. Of course, a number of constraints must be satisfied.

Objective function:

$$Z_{IP4} = \min \sum_{g\in G}\sum_{l\in L} a_l \; m_{gl} \qquad \text{(IP 4)}$$

subject to:

$$\sum_{p\in P_{gd}} x_{gpd}\alpha_{gd}\delta_{pl} \le m_{gl} \qquad \forall g\in G, d\in D_g, l\in L \qquad (4.1)$$

$$\sum_{g\in G} m_{gl} < C_l \qquad \forall l\in L \qquad (4.2)$$

$$m_{gl} \in [0, \max_{d\in D_g} \alpha_{gd}] \qquad \forall l\in L, g\in G \qquad (4.3)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l\in L, g\in G \qquad (4.4)$$

$$\sum_{l\in L} y_{gl} \ge \max\{h_g, |D_g|\} \qquad \forall g\in G \qquad (4.5)$$

$$\sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} \le |D_g| y_{gl} \qquad \forall g\in G, l\in L \qquad (4.6)$$

$$\sum_{l\in I_v} y_{gl} \le 1 \qquad \forall g\in G, v\in V-\{r_g\} \qquad (4.7)$$

$$\sum_{l\in I_{r_g}} y_{gl} = 0 \qquad \forall g\in G \qquad (4.8)$$

$$\sum_{p\in P_{gd}} x_{gpd} = 1 \qquad \forall d\in D_g, g\in G \qquad (4.9)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall d\in D_g, g\in G, p\in P_{gd}. \qquad (4.10)$$

The objective function of (IP 4) is to minimize the total transmission cost of servicing the maximum bandwidth requirement destination through a specific link for all multicast groups G, where G is the set of user groups requesting connection. The maximum bandwidth requirement on a link in the specific group, m_{gl}, can be viewed so that the source is required to transmit in a way that matches the most constrained destination.

Constraints (4.1) and (4.2) are referred to as the capacity constraints, which require that the aggregate flow on each link, l, does not exceed its link capacity, C_l. In Constraint (4.1), a variable, m_{gl}, is introduced, where m_{gl} can be interpreted as the "estimate" of the aggregate flow. Since the objective function is strictly an increasing function with m_{gl} and (IP 4) is a minimization problem, each m_{gl} will equal the aggregate flow in an optimal solution. Constraint (4.3) is a redundant constraint, which provides upper and lower bounds on the maximum traffic requirement for multicast group g on link l. Constraints (4.4) and (4.5) require that the number of links on the multicast tree adopted by the multicast group g be at least the maximum of h_g and the cardinality of D_g. The h_g and the cardinality of D_g are the legitimate lower bounds of the number of links on the multicast tree adopted by the multicast group g. Constraint (4.6) is referred to as the tree constraint, which requires that the union of the selected paths for the destinations of user group g forms a tree. Constraints (4.7) and (4.8) are both redundant constraints. Constraint (4.7) requires that the number of selected incoming links, y_{gl}, to a node is 1 or 0, while Constraint (4.8) requires that there are no selected incoming links, y_{gl}, to the node that is the root of multicast group g. As a result, the links we select can form a tree. Finally, Constraints (4.9) and (4.10) require that only one path is selected for each multicast source/destination pair.

4.4 Solution Approach

4.4.1 Lagrangean Relaxation

By using the Lagrangean Relaxation method, we can transform the primal problem (IP 4) into the following Lagrangean Relaxation problem (LR) where Constraints (4.1) and (4.6) are relaxed. For a vector of non-negative Lagrangean multipliers, a Lagrangean Relaxation problem of (IP) is given by

Optimization problem (LR):

$$Z_{D4}(\beta,\theta)=\min \sum_{g\in G}\sum_{l\in L}a_l m_{gl}+\sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L}\sum_{p\in P_{gd}}\beta_{gdl}x_{gpd}\alpha_{gd}\delta_{pl}-\sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L}\beta_{gdl}m_{gl} + \sum_{g\in G}\sum_{l\in L}\sum_{d\in D_g}\sum_{p\in P_{gd}}\theta_{gl}x_{gpd}\delta_{pl}-\sum_{g\in G}\sum_{l\in L}\theta_{gl}\left|D_g\right|y_{gl} \quad \text{(LR 4)}$$

subject to: (4.2) (4.3) (4.4) (4.5) (4.7) (4.8) (4.9) (4.10).

Where β_{gdl}, θ_{gl} are Lagrangean multipliers and β_{gdl}, $\theta_{gl}\geq 0$. To solve (LR), we decompose it into the following three independent and easily solvable optimization subproblems.

Subproblem 1: (related to decision variable x_{gpd})

$$Z_{Sub4.1}(\beta,\theta)=\min \sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}[\sum_{l\in L}\delta_{pl}(\beta_{gdl}\alpha_{gd}+\theta_{gl})]x_{gpd} \quad \text{(SUB 4.1)}$$

subject to:

$$\sum_{p\in P_{gd}}x_{gpd}=1 \qquad \forall d\in D_g, g\in G \quad (4.9)$$

$$x_{gpd}=0 \text{ or } 1 \qquad \forall d\in D_g, g\in G, p\in P_{gd}. \quad (4.10)$$

Subproblem (SUB 4.1) can be further decomposed into |G||D_g| independent shortest path problems with nonnegative arc weights. Each shortest path problem can be easily solved by Dijkstra algorithm.

Subproblem 2: (related to decision variable y_{gl})

$$Z_{Sub4.2}(\theta)=\min \sum_{g\in G}\sum_{l\in L}(-\theta_{gl}\left|D_g\right|)y_{gl} \quad \text{(SUB 4.2)}$$

subject to:

$$y_{gl}=0 \text{ or } 1 \qquad \forall l\in L, g\in G \quad (4.4)$$

$$\sum_{l\in L}y_{gl}\geq \max\{h_g,\left|D_g\right|\} \qquad \forall g\in G \quad (4.5)$$

$$\sum_{l\in I_v}y_{gl}\leq 1 \qquad \forall g\in G, v\in V-\{r_g\}. \quad (4.7)$$

$$\sum_{l \in I_{r_g}} y_{gl} = 0 \qquad \forall g \in G \quad (4.8)$$

The algorithm for solving Subproblem (SUB 4.2) is:

[Algorithm 4.1]

Step 1 Compute max{h_g, $|D_g|$} for multicast group g.

Step 2 Compute the number of positive coefficients $\theta_{gl}|D_g|$ for all links in the multicast group g.

Step 3 If the number of positive coefficients is greater than max{h_g, $|D_g|$} for multicast group g, then assign the corresponding positive coefficients of y_{gl} to 1, and 0 otherwise.

Step 4 If the number of positive coefficients is no greater than max{h_g, $|D_g|$} for multicast group g, assign the corresponding positive coefficients of y_{gl} to 1. Then, assign [max{h_g, $|D_g|$} – the number of positive coefficients of y_{gl}] number of the smallest negative coefficient of y_{gl} to 1, and 0 otherwise.

Subproblem 3: (related to decision variable m_{gl})

$$Z_{Sub4.3}(\beta) = \min \sum_{g \in G} \sum_{l \in L} (a_l - \sum_{d \in D_g} \beta_{gdl}) m_{gl} \qquad \text{(SUB 4.3)}$$

subject to:

$$\sum_{g \in G} m_{gl} < C_l \qquad \forall l \in L \quad (4.2)$$

$$m_{gl} \in [0, \max_{d \in D_g} \alpha_{gd}] \qquad \forall l \in L, g \in G \quad (4.3)$$

We decompose Subproblem (SUB 4.3) into |L| independent problems. For each link $l \in L$:

$$Z_{Sub4.3.1}(\beta) = \min \sum_{g \in G} (a_l - \sum_{d \in D_g} \beta_{gdl}) m_{gl} \qquad \text{(SUB 4.3.1)}$$

subject to: (2) (3).

The algorithm for solving Subproblem 3.1 is:

Step 1 Compute $a_l - \sum_{d \in D_g} \beta_{gdl}$ for link l of multicast group g.

Step 2 Sort the negative coefficients $a_l - \sum_{d \in D_g} \beta_{gdl}$ from the smallest value to the largest value

Step 3 According the sorted sequence. <i> assign the corresponding m_{gl} to the maximum traffic requirement in the multicast group and add to the sum value until the total amount of the maximum traffic requirement on link l is less than the capacity of link l. <ii> assign the boundary negative coefficient of m_{gl} to the difference between the capacity on link l and the sum value of m_{gl}, <iii> assign the other coefficients of m_{gl} to 0.

According to the weak Lagrangean duality theorem, for any β_{gdl}, $\theta_{gl} \geq 0$, $Z_D(\beta_{gdl}, \theta_{gl})$ is a lower bound of Z_{IP}. The following dual problem (D) is then constructed to calculate the tightest lower bound.

Dual Problem (D):

$$Z_{D4} = \max Z_{D4}(\beta_{gdl}, \theta_{gl}) \tag{D4}$$

subject to:

$$\beta_{gdl}, \theta_{gl} \geq 0 .$$

Several methods can be used to solve the dual problem (D). The most popular is the subgradient method, which is employed here. Let a vector, s, be a subgradient of $Z_D(\beta_{gdl}, \theta_{gl})$. Then, in iteration k of the subgradient optimization procedure, the multiplier vector is updated by $\omega^{k+1}=\omega^k+t^k s^k$. The step size, t^k, is determined by $t^k=\delta(Z^h_{IP} - Z_D(\omega^k))/||s^k||^2$, Z^h_{IP} is the primal objective function value for a heuristic solution (an upper bound on Z_{IP}), δ is a constant, and $0 < \delta \leq 2$.

4.4.2 Getting Primal Feasible Solutions

By optimally solving the Lagrangean dual problem, we have a set of decision variables. However, this solution would not be a feasible one for the primal problem, since some constraints are not satisfied. Thus, minor modification of the decision variables, or the hints of multipliers, must be considered in oder to obtain the primal feasible solution of the problem (IP). Generally speaking, an upper bound (UB) of the problem (IP) is a better primal feasible solution, while the Lagrangean dual problem solution guarantees the lower bound (LB) of the problem. Iteratively, by solving the Lagrangean dual problem and getting the primal feasible solution, we get the LB and UB, respectively. So, the gap between the UB and the LB, computed by (UB-LB)/LB*100%, illustrates the optimality of problem solution. The smaller the gap computed, the better the optimality.

To calculate the primal feasible solution of the minimum cost tree, we consider the solutions to the Lagrangean Relaxation problems. The set of $\{x_{gpd}\}$ obtained by solving (Subproblem 1) may not be a valid solution for problem (IP), because the capacity constraint is relaxed. However, the capacity constraint may be a valid solution for some links. Also, the set of $\{y_{gl}\}$ obtained by solving (Subproblem 2) may not be a valid solution, because of the link capacity constraint and the union of $\{y_{gl}\}$ may not be a tree.

Here, we propose a comprehensive, two-part method for obtaining a primal feasible solution. It utilizes a Lagrangean based modified T-M heuristic, followed by adjustment procedures. While solving the Lagrangean relaxation dual problem, we may find some multipliers related to each OD pair and links, which could make our routing more efficient. We describe the Lagrangean based modified T-M heuristic below.

[Lagrangean-multiplier-based Modified T-M Heuristic]

1) Use $a_l - \sum_{d \in D_g} \beta_{gdl}$ as link *l*'s arc weight and run the M-T-M heuristic.

2) After getting a feasible solution, we apply *the Lagrangean-multiplier-based adjustment procedure* (LAP) to adjust the result.

[Lagrangean-Multiplier-based Adjustment Procedure (LAP)]

1) Compute the aggregate flow of each link.

2) Sort the links in descending order based on the difference between aggregate flow of each link and the link's capacity.

3) Choose the first link. If the difference value of the link is positive, go to Step 4; otherwise go to Step 6.

4) Choose the group that has the minimal sensitivity value $a_l - \sum_{d \in D_g} \beta_{gdl}$ on that link, drop and use $a_l - \sum_{d \in D_g} \beta_{gdl}$ as link l's arc weight, and run the M-T-M heuristic to re-add it to the tree. Consider the following possible adding measures and set the best one as the final tree. Either add the dropped node to the source node, or to other nodes with the same hop count, or to the nodes with a hop count larger or smaller by one.

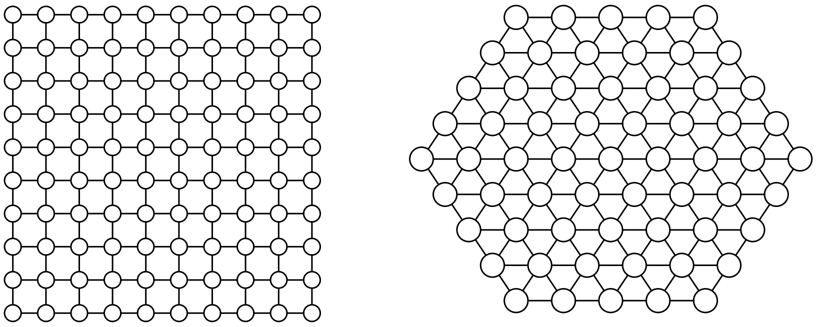
5) If a feasible solution is found, go to Step2; otherwise go to tep 6.

6) Stop.

4.5 Experimental Results

In this section, computational experiments on the Lagrangean based heuristic and the simple heuristics are reported. The heuristics are also tested on three kinds of networks- regular networks, random networks, and scale-free networks.

Two regular networks shown in Figure 4.1 are tested in our experiment. The first is a grid network containing 100 nodes and 180 links, and the second is a cellular network containing 61 nodes and 156 links. Random networks tested in this experiment are generated randomly, each having 500 nodes. The candidate links between all node pairs are given a probability following the uniform distribution. In the experiments, we link the node pair with a probability smaller than 2%. If the generated network is not a connected network, we generate a new network.



(a) Grid Network (b) Cellular Network

Figure 4.1: Regular networks (C-MCMR)

In our experiments, we apply this preferential attachment method to generate scale-free networks. The corresponding preferential variable (m_0, m) is (2, 2), and the number of nodes in the test networks is 500.

Table 4.2: Parameters for Lagrangean relaxation (C-MCMR)

Number of Iterations	2,000
Initial Multipliers	0
Improvement Counter	25
Delta Factor	2
Optimal Condition	Gap < 0.001

For each test network, several distinct cases, which have different pre-determined parameters, are considered. The traffic demand for each destination is drawn from a random variable that is uniformly distributed in pre-specified categories {1, 2, 5, 10, 15, 20}. The link costs are randomly generated between 1 and 5, and the group number of each test case is 20. The cost of the multicast tree is decided by multiplying the link cost and the maximum bandwidth requirement on a link. We conducted 500 experiments for each kind of network. For each experiment, the result is determined by the group source, and destinations and link costs are generated randomly. Table 4.3 summaries the selected results of the computational experiments.

Table 4.3: Selected results of computational experiments (C-MCMR)

C#	N #	SA	UB	LB	GAP	Imp.
Grid Network						
A	5	9,045	8,825	8,685.48	1.61%	2.49%

B	5	10,507	9,639	9,425.01	2.27%	9.01%
C	10	16,476	14,691	13,906.21	5.64%	12.15%
D	10	16,805	15,318	15,147.38	1.13%	9.71%
E	20	23,978	21,133	20,791.90	1.64%	13.46%
F	20	N/A	22,910	19,884.47	15.22%	∞
G	50	40,167	36,241	32,476.30	11.59%	10.83%
H	50	N/A	34,708	30,964.02	12.09%	∞
Cellular Network						
A	5	5,248	4,965	4,890.18	1.53%	5.70%
B	5	4,628	4,281	4,070.81	5.16%	8.11%
C	10	8,928	8,238	7,936.96	3.79%	8.38%
D	10	9,874	9,253	8,904.63	3.91%	6.71%
E	20	15,375	14,750	13,067.21	12.88%	4.24%
F	20	N/A	13,912	12,271.44	13.37%	∞
G	50	N/A	25,160	20,557.85	22.39%	∞
H	50	N/A	25,973	21,261.94	22.16%	∞
Random Networks						
A	5	3,984	3,763	3,487.12	7.91%	5.87%
B	5	3,952	3,465	3,421.11	1.28%	14.05%
C	10	6,765	5,862	5,474.57	7.08%	15.40%
D	10	8,790	8,360	7,300.52	14.51%	5.14%
E	20	14,465	12,782	11,558.87	10.58%	13.17%
F	20	13,266	11,811	9,364.91	26.12%	12.32%
G	50	28,690	24,555	21,540.62	13.99%	16.84%
H	50	28,833	25,774	21,864.95	17.88%	11.87%
Scale-Free Networks						
A	5	5,503	5,176	4,853.17	6.65%	6.32%
B	5	3,939	3,801	3,603.81	5.47%	3.63%
C	10	9,109	8,485	8,051.22	5.39%	7.35%
D	10	9,649	8,847	8,580.09	3.11%	9.07%
E	20	16,361	15,143	14,533.04	4.20%	8.04%
F	20	14,831	13,459	13,107.20	2.68%	10.19%
G	50	30,676	27,737	25,813.31	7.45%	10.60%
H	50	N/A	28,239	25,068.99	12.65%	∞

C#: Case Number
N#: Number of destinations in a group
SA: The result of the simple heuristic
UB and LB: Upper and lower bounds of the Lagrangean-based modified T-M heuristic
GAP: Bound difference {(UB-LB)/LB}
Imp.: The improvement ratio of the Lagrangean-based modified T-M heuristic {(SA - UB)/UB}

For each test network, the maximum improvement ratio between the simple heuristic and the Lagrangean-based heuristic is 13.46 %, 8.83%, 15.40 %, and 10.60%, respectively. In general, the Lagrangean-based heuristic performs well compared to the simple heuristic, even when the simple algorithm can not find a feasible solution, such as case F and H of the grid network and case F, G, and H of the cellular network.

There are two reasons why the Lagrangean-based heuristic works better than the simple algorithm. First, the simple algorithm routes the group in accordance with a fixed link cost and residual capacity only, whereas the Lagrangean based heuristic makes use of the related Lagrangean multipliers. The multipliers include the potential cost for routing on each link of the topology. Second, the Lagrangean-based heuristic is iteration-based and is guaranteed to improve the solution quality iteration by iteration. Therefore, in a more complicated testing environment, the improvement ratio is higher.

To claim optimality, in Table 4.3, we also depict the percentile of gap. The results show that 72% of the regular and scale free networks have a gap of less than 10%, but the results of the random networks show a larger gap. We also find that the simple heuristic performs well in many cases, such as case A of grid network and case B of the scale-free network.

4.6 Concluding Remarks

In this chapter, we have attempted to solve the problem of capacitated min-cost multicast routing for multirate multimedia distribution. Our contribution can be expressed in terms of the mathematical formulation and experiment performance. For the formulation, we have proposed a precise mathematical expression to model the problem efficiently. With regard to performance, the proposed Lagrangean-based heuristic outperforms the simple heuristics.

Our model can also be extended to deal with the QoS multicast routing problem for multirate multimedia distribution by adding QoS constraints. Moreover, the min-cost model proposed in this chapter can be modified as a max-revenue model with the objective of maximizing total system revenues by totally, or partially, admitting destinations into the system.

CHAPTER 5 THE PARTIAL ADMISSION CONTROL PROBLEM OF SINGLE-RATE MULTICASTING

5.1 Overview

Multimedia application environments are characterized by large bandwidth variations due to the heterogeneous access technologies of networks and different receivers' quality requirements, which make it difficult to achieve bandwidth efficiency and service flexibility. There are many challenging issues that need to be addressed when designing architectures and mechanisms for multicast data transmission. Traffic engineering is the process of controlling how traffic flows through a network in order to optimize resource utilization and network performance, while simultaneously providing QoS. The goal of QoS routing is to select network routes with sufficient resources for the requested QoS parameters, satisfy the QoS requirements for every admitted connection, and to achieve global efficiency in resource utilization. Admission control is often considered a by-product of QoS routing and resource reservation. If the latter is successfully performed along the route(s) selected by the routing algorithm, the connection request is accepted; otherwise, it is rejected. It is clear from the above introduction that when considering the QoS assurance issue, the three closely-related mechanisms of admission control, routing and resource reservation should be treated jointly.

In this chapter, we consider the above three mechanisms jointly and attempt to solve the problem of maximum-revenue multicast routing with a partial admission control mechanism. The partial admission control mechanism means that the admission policy of a multicast group is not based on a traditional "all or none" strategy. Instead, it considers accepting portions destinations for the requested multicast group. More specifically given a network topology, a link capacity, the destinations of a multicast group, and the bandwidth requirement of each multicast group, we attempt to find a feasible admission decision and routing solution to maximize the revenue of the multicast trees. We begin by modeling this problem as a linear optimization problem, and then propose a simple heuristic algorithm and an optimization-based heuristic to solve the problem. The methodology used to solve the problem is Lagrangean relaxation. We perform computational experiments on regular networks, random networks, and scale-free networks.

The remainder of this chapter is organized as follows. In Section 5.2, we formally define the problem, and propose a mathematical formulation of max-revenue optimization. Section 5.3 applies Lagrangean relaxation as a solution approach to the problem. Section 5.4 describes our computational experiments. In Section 5.5, we simulate our algorithm in a real-time scenario. Finally, in Section 5.6 we present our conclusions and indicate the direction of future research.

5.2 Problem Formulation

The network is modeled as a graph where, the switches are depicted as nodes and the links are depicted as arcs. A user group, which has one source and one or more destinations, is an application requesting transmission over the network. Given the network topology, the capacity of the links and the bandwidth requirement of user groups, we want to determine the following decision variables: (1) the routing assignment (a tree for multicasting, or a path for unicasting) of each admitted destination; and (2) the admitted number of destinations of each partially admitted multicast group. We assume that the multicasting is single-rate.

By formulating the problem as a mathematical programming problem, we intend to solve it optimally and thereby obtain a network that fits our goal, i.e., it ensures the network operator can earn the maximum revenue by servicing partially admitted destinations.

This model is based on the following viable assumptions.

- The revenue from each partially admitted group can be fully characterized by two parameters: the entire admitted revenue of the group and the number of admitted destinations.
- The revenue from each partially admitted group is a monotonically increasing function with respect to the number of admitted destinations.
- The revenue function from each partially admitted group is a convex function with respect to the entire admitted revenue of the group and the number of admitted destinations. However, taken together, the total amount of admitted revenue and the total number of admitted destinations may not be a concave function.
- The revenue from each partially admitted group is independent.

The notations used to model the problem are listed in Table 5.1.

Table 5.1: Description of notations (PCAC-S)

Given Parameters	
Notation	**Description**
F_g	Revenue generated by admitting partial users of multicast group g, which is a function of f_g and a_g
a_g	Revenue generated by admitting multicast group g
α_g	Traffic requirement of multicast group g
G	The set of all multicast groups
V	The set of nodes in the network
L	The set of links in the network
D_g	The set of destinations of multicast group g
C_l	The capacity of link l
I_v	The incoming links to node v

r_g	The multicast root of multicast group *g*
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of paths user *d* of multicast group *g* may use
δ_{pl}	The indicator function, which is 1 if link *l* is on path *p*, and 0 otherwise
Decision Variables	
Notation	**Description**
x_{gpd}	1 if path *p* is selected for group *g* destined for destination *d* , and 0 otherwise
y_{gl}	1 if link *l* is on the sub-tree adopted by multicast group *g*, and 0 otherwise
f_g	The number of admitted destinations in multicast group *g*

Based on the description in the previous section, the max-revenue problem is formulated as a combinatorial optimization problem in which the objective function is to maximize revenue from servicing partially admitted destinations. Of course a number of constraints must be satisfied.

Optimization Problem:

Objective function:

$$\min \; -\sum_{g\in G} F_g(a_g, f_g) \qquad \text{(IP 5)}$$

subject to:

$$\sum_{g\in G} \alpha_g y_{gl} \le C_l \qquad \forall l\in L \qquad (5.1)$$

$$\sum_{l\in L} y_{gl} \ge \sum_{d\in D_g} \sum_{p\in P_{gd}} x_{gpd} \qquad \forall g\in G \qquad (5.2)$$

$$\sum_{d\in D_g} \sum_{p\in P_{gd}} x_{gpd}\delta_{pl} \le |D_g| y_{gl} \qquad \forall g\in G, l\in L \qquad (5.3)$$

$$\sum_{l\in I_v} y_{gl} \le 1 \qquad \forall g\in G, v\in V-\{r_g\} \qquad (5.4)$$

$$\sum_{l\in I_{r_g}} y_{gl} = 0 \qquad \forall g\in G \qquad (5.5)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l\in L, g\in G \qquad (5.6)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall g \in G, p \in P_{gd}, d \in D_g \qquad (5.7)$$

$$\sum_{p \in P_{gd}} x_{gpd} \le 1 \qquad \forall d \in D_g, g \in G \qquad (5.8)$$

$$\sum_{d \in D_g} \sum_{p \in p_{gd}} x_{gpd} = f_g \qquad \forall g \in G \qquad (5.9)$$

$$f_g \in \{0, 1, 2, ..., |D_g|\} \qquad \forall g \in G. \qquad (5.10)$$

The objective function of (IP 5) is to maximize the total revenue from servicing the partially admitted destinations in multicast group g, where $g \in G$ and G is the set of user groups requesting transmission. F_g reflects the priority of partial users belonging to group g, while different choices of F_g may provide different physical interpretations of the objective function. For example, if F_g is chosen to be the mean traffic requirement of partial users belonging to group g, then the objective function is to maximize the total system throughput. On the other hand, if F_g is chosen to be the earnings of servicing partial users belonging to group g, then the objective function is to maximize the total system revenue. In general, if user group g is given a higher priority, then the corresponding F_g may be assigned a larger value.

Constraint (5.1) is the capacity constraint. It requires that the aggregate flow on each link, l, does not exceed its physical capacity, C_l. Constraint (5.2) requires that if one path is selected for group g destined for destination d, it must also be on the sub-tree adopted by multicast group g. Constraint (5.3) is the tree constraint, which requires that the union of the selected paths for the destinations of user group g should form a tree. Constraints (5.4) and (5.6) require that the number of selected incoming links y_{gl}, is 1 or 0 and each node, except the root, has only one incoming link. Constraint (5.5) requires that there is no selected incoming link y_{gl} that is the root of multicast group g. As a result, the links we select can form a tree. Constraints (5.7) and (5.8) require that at most one path is selected for each admitted multicast source-destination pair, while Constraint (5.9) relates the routing decision variables x_{gpd} to the auxiliary variables f_g. The introduction of the auxiliary variables f_g may facilitate decomposition in the Lagrangean relaxation problem, which we discuss later. Constraint (5.10) requires that the number of admitted destinations in multicast group g is an integer within the predefined set.

5.3 Solution Procedure

5.3.1 Lagrangean Relaxation

By using the Lagrangean Relaxation method, we can transform the primal problem (IP 5) into the following Lagrangean Relaxation problem (LR), where constraints (5.1), (5.2), (5.3), and (5.9) are relaxed.

For a vector of Lagrangean multipliers, the Lagrangean Relaxation problem of (IP 5) is given by

Optimization problem (LR):

$$\begin{aligned} & Z_{D5}(\beta,\lambda,\theta,\varepsilon)= \\ & \min\ -\sum_{g\in G} F_g(a_g,f_g)+\sum_{g\in G}\sum_{l\in L}\beta_l\alpha_g y_{gl}-\sum_{l\in L}\beta_l C_l+\sum_{g\in G}\sum_{l\in L}\lambda_g y_{gl} \\ & \quad -\sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}\lambda_g x_{gpd}+\sum_{g\in G}\sum_{l\in L}\sum_{d\in D_g}\sum_{p\in P_{gd}}\theta_{gl}x_{gpd}\delta_{pl}-\sum_{g\in G}\sum_{l\in L}\theta_{gl}\left|D_g\right|y_{gl} \\ & \quad +\sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}\varepsilon_g x_{gpd}-\sum_{g\in G}\varepsilon_g f_g \end{aligned} \qquad \text{(LR 5)}$$

subject to: (5.4)(5.5)(5.6)(5.7)(5.8)(5.10).

Where β_l, λ_g, θ_{gl}, and ε_g are Lagrangean multipliers and $\beta_l,\theta_{gl}\geq 0$. To solve (LR), we decompose (LR) into the following five independent and easily solvable optimization subproblems.

Subproblem 1: (related to decision variable x_{gpd})

$$Z_{Sub5.1}(\lambda,\theta,\varepsilon)=\min\ \sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}(\varepsilon_g-\lambda_g+\sum_{l\in L}\theta_{gl}\delta_{pl})x_{gpd} \qquad \text{(SUB 5.1)}$$

subject to:

$$x_{gpd}=0 \text{ or } 1 \qquad \forall g\in G, p\in P_{gd}, d\in D_g \qquad (5.7)$$

$$\sum_{p\in P_{gd}} x_{gpd}\leq 1 \qquad \forall d\in D_g, g\in G. \qquad (5.8)$$

The Subproblem (SUB 5.1) is to determine x_{gpd} and can be further decomposed into $|G||D_g|$ independent shortest path problems with nonnegative arc weights, θ_{gl}. Each shortest path problem can be easily solved by Dijkstra's algorithm.

Subproblem 2: (related to decision variable y_{gl})

$$Z_{Sub5.2}(\beta,\lambda,\theta)=\min \sum_{g\in G}\sum_{l\in L}(\beta_l\alpha_g+\lambda_g-\theta_{gl}\left|D_g\right|)y_{gl} \quad , \qquad \text{(SUB 5.2)}$$

subject to:

$$\sum_{l\in I_v} y_{gl}\le 1 \qquad \forall g\in G, v\in V-\{r_g\} \qquad (5.4)$$

$$\sum_{l\in I_{r_g}} y_{gl} = 0 \qquad \forall g\in G \qquad (5.5)$$

$$y_{gl}=0 \text{ or } 1 \qquad \forall l\in L, g\in G . \qquad (5.6)$$

The Subproblem (SUB 5.2) can be decomposed into |G| independent problems. For each multicast group $g\in G$:

$$Z_{Sub5.2.1}(\beta,\lambda,\theta)=\min \sum_{l\in L}(\beta_l\alpha_g+\lambda_g-\theta_{gl}\left|D_g\right|)y_{gl} \qquad \textit{(SUB 5.2.1)}$$

subject to: (5.4)(5.5)(5.6).

The algorithm for solving Subproblem (SUB 5.2.1) is stated as follows:

1. Compute the coefficient $\beta_l\alpha_g+\lambda_g-\theta_{gl}|D_g|$ for all links in multicast group g.
2. Sort the links in ascending order according to the coefficient.
3. According to the order and complying with constraints (5.4) and (5.5), if the coefficient is less than zero, assign the corresponding negative coefficient of y_{gl} to 1; otherwise, 0.

Subproblem 3: (related to decision variable f_g)

$$Z_{Sub5.3}(\varepsilon)=\min \ -\sum_{g\in G}(F_g(a_g,f_g)+\varepsilon_g f_g) \qquad \text{(SUB 5.3)}$$

subject to:

$$f_g \in \{0,1,2,......,|D_g|\} \quad \forall g \in G. \quad (5.10)$$

We can easily solve Subproblem (SUB 5.3) optimally by exhaustively searching from the known set of f_g.

According to the weak Lagrangean duality theorem, for any $\beta_l, \theta_{gl} \geq 0$, $Z_D(\beta_l, \lambda_g, \theta_{gl}, \varepsilon_g)$ is a lower bound on Z_{IP5}. The following dual problem (D) is then constructed to calculate the tightest lower bound.

Dual Problem (D):

$$Z_D = \max Z_D(\beta_l, \lambda_g, \theta_{gl}, \varepsilon_g)$$

subject to: $\beta_l, \theta_{gl} \geq 0$.

There are several methods for solving the dual problem (D). The most popular is the subgradient method [5], which is employed here. Let a vector, s, be a subgradient of $Z_D(\beta_l, \lambda_g, \theta_{gl}, \varepsilon_g)$. Then, in iteration k of the subgradient optimization procedure, the multiplier vector is updated by $\omega^{k+1} = \omega^k + t^k s^k$. The step size, t^k, is determined by $t^k = \delta(Z^h_{IP} - Z_D(\omega^k))/\|s^k\|^2$, Z^h_{IP} is the primal objective function value for a heuristic solution (an upper bound on Z_{IP}), and δ is a constant and $0 < \delta \leq 2$.

5.3.2 Getting Primal Feasible Solutions

After optimally solving the Lagrangean dual problem, we have a set of decision variables. However, this solution is not feasible for the primal problem, since some of constraints are not satisfied. Thus, minor modification of the decision variables, or the hints of multipliers, must be considered in order to obtain the primal feasible solution of problem (IP 5). Generally speaking, the best primal feasible solution is an upper bound (UB) of the problem (IP 5), while the Lagrangean dual problem solution guarantees the lower bound (LB) of problem (IP 5). Iteratively, by solving the Lagrangean dual problem and getting the primal feasible solution, we get the LB and UB, respectively. So, the gap between UB and LB, computed by (UB-LB)/LB*100%, illustrates the optimality of problem solution. The smaller the gap computed, the better the optimality.

To calculate the primal feasible solution of the maximum revenue tree, we considerthe solutions to the Lagrangean relaxation problems. The set of $\{x_{gpd}\}$ obtained by solving Subproblem 1 may not be a valid solution to problem (IP 5), because the capacity constraint is relaxed. However, the capacity constraint may be a valid solution for some links. Also, the set of $\{y_{gl}\}$ obtained by solving Subproblem 2 may not be a valid solution because of the link capacity constraint and the union of $\{y_{gl}\}$ may not form a tree. Furthermore, Constraint (9) is relaxed, the set of $\{f_g\}$ obtained by solving Subproblem 3 may not be a valid solution.

Here, we propose a comprehensive, two-part method to obtain a primal feasible solution. It utilizes a Lagrangean multiplier-based heuristic, followed by adjustment procedures. While solving the Lagrangean relaxation dual problem, we may find some multipliers related to each OD pair and links, which could make our routing more efficient. We describe the Lagrangean based heuristic below.

[Lagrangean Multipliers based heuristic]

Step 1 Use $\beta_1\alpha_g+\lambda_g-\theta_{g1}|D_g|$ as link l's arc weight and run the T-M heuristic to get a spanning tree for each multicast group.

Step 2 **Drop procedures**:

2.1 Check the capacity constraint of each link. If a link violates the capacity constraint, go to Step 2.2; otherwise goto Step 3.

2.2 Sort the links in descending order according to {C_l – the aggregate flow on the link}. Choose the maximal overflow link and drop the group with the maximal subgradient $(-F_g(a_g, f_g)-\varepsilon_g f_g)$. Go to Step 2.1.

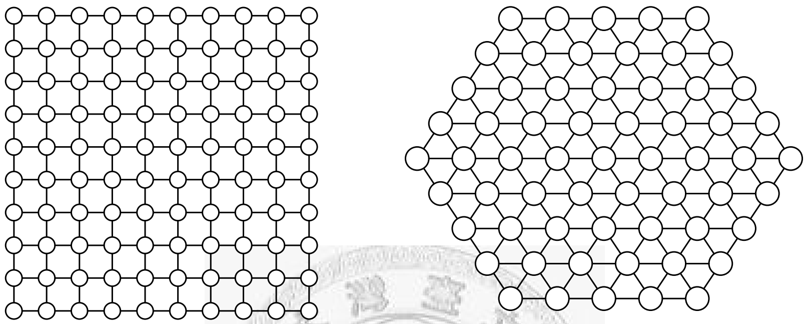
Step 3 **Add procedures:**

3.1 Sort the dropped group in ascending order according to the subgradient $(-F_g(a_g, f_g)-\varepsilon_g f_g)$.

3.2 In accordance with the order, re-add the groups to the network. Use $\beta_1\alpha_g+\lambda_g-\theta_{g1}|D_g|$ as link l's arc weight, remove the overflow links from the graph and run the T-M heuristic. If it can not find a route for the destinations, drop the destinations.

5.4 Computational Experiments

In this section, computational experiments on the Lagrangean relaxation based heuristic and other primal heuristics are reported. The heuristics are tested on three kinds of network: regular networks, random networks, and scale-free networks. Two regular networks shown in Figure 5.1, are tested in our experiment. The first is a grid network containing 100 nodes and 180 links, and the second is a cellular network containing 61 nodes and 156 links.



(a) Grid Network (b) Cellular Network

Figure 5.1: Regular networks (PCAC-S)

The random networks tested in the experiments are generated randomly, each having 100 nodes. The candidate links between all node pairs are given a probability following the uniform distribution. In the experiments, we link the node pair with a probability smaller than 2%. If the generated network is not a connected network, we generate a new network.

Reference [73] shows that scale-free networks can be developed from a simple dynamic model that combines incremental growth with a preference for new nodes to connect to existing ones that are already well connected. In our experiments, we apply this *preferential attachment method* to generate scale-free networks. The corresponding preferential variable (m_0, m) is (2, 2). The number of nodes in the test networks is 100.

In order to prove that our heuristics are superior, we also implement a simple algorithm to compare with our heuristic.

[Simple Algorithm]

```
Step 1 Set link l's arc weight to 1 and run the T-M heuristic to
       get a spanning tree for each multicast group.
```

Step 2 **Drop procedures**:

2.1 Check the capacity constraint of each link. If a link violates the capacity constraint, go to Step 2.2, otherwise goto Step 3.

2.2 Sort the links in ascending order according to {C_l - the aggregate flow on the link}. Choose the maximal overflow link and drop the group with the minimal revenue. Go to Step 2.1.

Step 3 **Add procedures:**

3.1 Sort the dropped group in descending order according to the unit revenue {Group revenue/number of destinations of the group}.

3.2 In accordance with the order, re-add the groups to the network. Remove the overflow links from the graph, set each link's arc weight to the aggregate flow of the link and run the T-M heuristic. If it can not find a feasible route for the destinations, drop the destinations.

For each test network, several distinct cases with different pre-determined parameters, such as the link capacity, the number of multicast groups and the number of nodes in a group, are considered. The traffic demand for each multicast group is drawn from a random variable that is uniformly distributed in pre-specified categories {1, 2, 5, 10, 15, 20}. We conducted 120 experiments for each kind of network. For each experiment, the result was determined by the group source and destinations generated randomly. Table 5.2 summaries selected results of the computational experiments.

Table 5.2: Selected results of computational experiments (PCAC-S)

CASE	Cap.	G #	N #	SA	UB	LB	GAP	Imp.
Grid Networks								**Max Imp. Ratio: 186.46 %**
A	20	20	20	-1777.01	-1998.12	-2400	16.75%	12.44%
B	20	20	50	-2010.87	-3536.48	-5274.67	32.95%	75.87%
C	20	50	20	-3052.47	-3731.75	-5918.61	36.95%	22.25%
D	20	50	50	-1998.79	-5725.72	-8123.47	29.52%	186.46%
E	20	100	20	-3744.71	-5859.17	-9232.08	36.53%	56.47%
F	20	100	50	-5844.29	-9574.34	-14114.3	32.17%	63.82%
G	40	20	20	-2170.69	-2208.75	-2322.6	4.90%	1.75%

H	40	20	50	-3854.31	-4105.33	-5450	24.67%	6.51%
I	40	50	20	-3613.02	-3636.07	-5086.36	28.51%	0.64%
J	40	50	50	-5382.09	-6862.85	-10767.9	36.27%	27.51%
K	40	100	20	-6118.06	-6506.32	-11033.3	41.03%	6.35%
L	40	100	50	-10594.3	-14500.4	-20074.9	27.77%	36.87%
Cellular Networks								**Max Imp. Ratio: 93.37 %**
A	20	20	20	-1531.19	-1748.98	-2340	25.26%	14.22%
B	20	20	50	-4686.17	-5394.88	-5600.02	3.66%	15.12%
C	20	50	20	-4212.02	-4407.76	-5813.74	24.18%	4.65%
D	20	50	50	-4262.02	-8241.3	-9765.03	15.60%	93.37%
E	20	100	20	-4620.5	-6083.93	-8604.21	29.29%	31.67%
F	20	100	50	-7117.66	-12337.8	-14587.2	15.42%	73.34%
G	40	20	20	-2031.8	-2040	-2044.53	0.22%	0.40%
H	40	20	50	-4329.65	-4529.65	-4900	7.56%	4.62%
I	40	50	20	-5244.04	-5352.35	-5840	8.35%	2.07%
J	40	50	50	-8684.42	-9577.18	-11413.8	16.09%	10.28%
K	40	100	20	-7301.44	-7538.01	-11184.8	32.61%	3.24%
L	40	100	50	-13701.3	-17705.2	-20706.6	14.49%	29.22%
Random Networks								**Max Imp. Ratio: 137.08 %**
A	20	20	20	-1799.28	-1945.48	-2060	5.56%	8.13%
B	20	20	50	-4161.85	-4609.93	-4750	2.95%	10.77%
C	20	50	20	-4204.04	-4541.83	-5460	16.82%	8.03%
D	20	50	50	-5168.37	-7950.94	-11279.4	29.51%	53.84%
E	20	100	20	-4323.71	-4979.78	-9704.55	48.69%	15.17%
F	20	100	50	-5050.87	-11974.8	-18540.9	35.41%	137.08%
G	40	20	20	-2033.63	-2044.82	-2123.08	3.69%	0.55%
H	40	20	50	-5153.08	-5239.75	-5450	3.86%	1.68%
I	40	50	20	-6155.9	-6160	-6175.29	0.25%	0.07%
J	40	50	50	-12539.6	-12676.2	-16000	20.77%	1.09%
K	40	100	20	-5811.08	-5962.94	-10734.4	44.45%	2.61%
L	40	100	50	-11940.5	-15569	-23335.7	33.28%	30.39%
Scale-free Networks								**Max Imp. Ratio: 139.17 %**
A	20	20	20	-1969.75	-2117.51	-2580	17.93%	7.50%
B	20	20	50	-2997.46	-3343.65	-4892.02	31.65%	11.55%
C	20	50	20	-2933.91	-3426.09	-5429.09	36.89%	16.78%
D	20	50	50	-4588.44	-7384.51	-10542.8	29.96%	60.94%
E	20	100	20	-2908.92	-4809.64	-8109.17	40.69%	65.34%
F	20	100	50	-4146.94	-9918.12	-14771.3	32.86%	139.17%
G	40	20	20	-2184.54	-2216.01	-2237.26	0.95%	1.44%
H	40	20	50	-3980.47	-4096.46	-4857.93	15.67%	2.91%
I	40	50	20	-4062.27	-4171.08	-5440	23.33%	2.68%
J	40	50	50	-7237.48	-9053.87	-12152.2	25.50%	25.10%
K	40	100	20	-5421.27	-6266.76	-10723.7	41.56%	15.60%
L	40	100	50	-9482.01	-14139.5	-19914.6	29.00%	49.12%

Cap.: The capacity of each link
G#: The number of multicast groups
N#: The number of destinations in each multicast group
SA: The result of the simple algorithm
UB: Upper bounds of the Lagrangean-based heuristic
LB: Lower bounds of the Lagrangean-based heuristic
GAP: The error gap of Lagrangean relaxation
Imp.: The improvement ratio of the Lagrangean-based heuristic

For each test network, the maximum improvement ratio between the simple heuristic and the Lagrangean based heuristic is 186.46 %, 93.37%, 137.08 %, and 139.17%, respectively. In general, the Lagrangean-based heuristic performs well compared to the simple heuristic. We also find that in more congested networks, either with more destinations or with less link capacity, the Lagrangean based heuristic outperforms the simple heuristic, such as in the case D of the grid network and case F of the scale-free network.

The Lagrangean-based heuristic works better than the simple algorithm for two reasons. First, it makes use of the related Lagrangean multipliers, which include the potential cost for routing on each link in the topology. Second, it is iteration-based and is guaranteed to improve the solution quality iteration by iteration. Therefore, in a more complicated test environment, the improvement ratio is higher.

To claim optimality, in Table 3, we also depict the percentile of gap. The results show that most cases have a gap of less than 40%. We also find that the simple heuristic performs well in many cases, such as case I of the grid network and case G of the random network.

5.5 Real-Time Admission Control

In a good call admission control (CAC) mechanism, after receiving a request from an end user, the CAC mechanism should make a decision to admit or reject the user request within a reasonable time, i.e., 10 to 20 seconds. However, in the previous session, we treated the max-revenue call admission control problem as a planning problem and did not consider the time spent on decision-making. In this session, we consider a real-time admission control mechanism that must make all call admission decisions in seconds.

In the overall system, new call requests are Poisson distributed with mean rate λ. The call holding time is assumed to be exponentially distributed with mean τ. For a specific time slot, η, time Γ-1 and Γ are, respectively, the start and stop points of the time slot, as shown in Figure 5.2. At time Γ, λ_Γ, α_Γ, γ_Γ, and ε_Γ are the number of arriving calls, the number of admitted calls, the number of remaining calls, and the total existence respectively. Accordingly, the call admission control mechanism is a function of λ_Γ and ε_Γ,in which $\varepsilon_\Gamma = (\alpha_\Gamma + \gamma_\Gamma)$ is the sum of admitted and remaining calls.

$$\alpha_\Gamma = CAC(\lambda_{\Gamma-1}, \varepsilon_{\Gamma-1}) \tag{5.11}$$

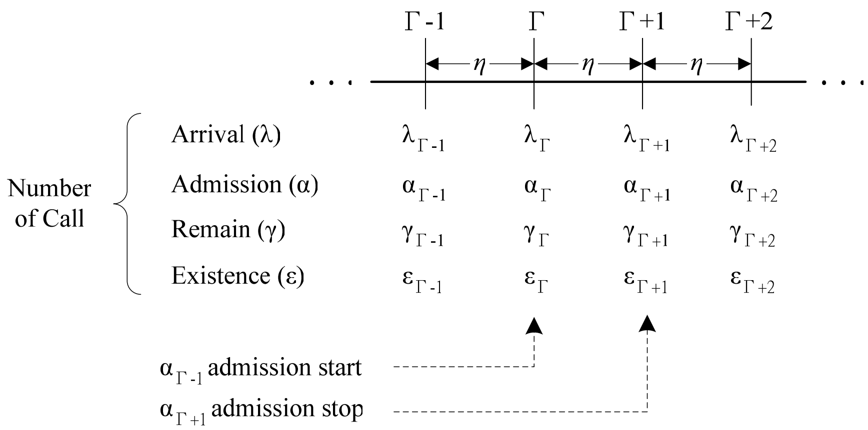


Figure 5.2: The time diagram of real-time admission control

Since the call holding time is assumed to be exponentially distributed with mean τ, $\gamma_\Gamma = \lfloor \varepsilon_{\Gamma-1} \cdot e^{-\eta/\tau} \rfloor$ represents the number of remaining calls of $\varepsilon_{\Gamma-1}$ after time slot η, where $\lfloor\ \rfloor$ is a floor function. The initial values are α_0=0, γ_0=0, and ε_0=0. To clarify the real-time CAC mechanism, Table 5.3 illustrates an example of call number calculation with λ=30, τ=90, and η=3. At the end of time slot 3, for example, 39 of 45 arriving calls at the end of time slot 2 have been admitted. Also, 52 of 54 calls at the end of time slot 2 remain after time slot η. Thus, the total existence at the end of time slot 3 is 91.

Table 5.3: An example of call number calculation with λ=30, τ=90, and η=3

Time slot (t)	**0**	**1**	**2**	**3**	**4**	**5**	**…**	**…**
Arrival (λ_Γ)	26	34	45	28	22	31	…	…
Admission (α_Γ)	0	25	30	39	24	19	…	…
Remain (γ_Γ)	0	0	24	52	88	108	…	…
Existence (ε_Γ)	0	25	54	91	112	127	…	…

5.5.1 A Real-time LR-based CAC Algorithm

Based on the LR approach described in Section 5.3, a pre-defined time budget, η, e.g., 5 seconds, is given to solve the Lagrangean dual problem and get primal feasible solutions iteratively. Actually the time budget is equivalent to the time slot. In a specific time slot, real-time call admission control is completed at the end of the time slot. The number of call requests admitted depends on the time budget.

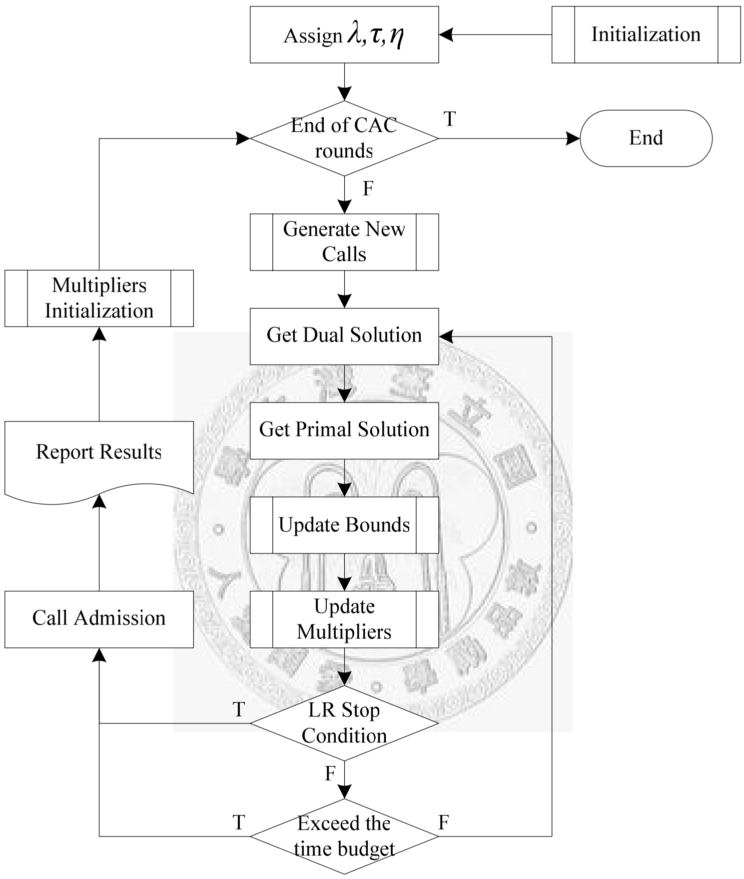


Figure 5.3: The procedure of the LR-based real-time CAC algorithm

The primal feasible solution algorithm described in Section 5.3 uses the multipliers as the arc weight. In a real-time scenario, the number of arriving and departing calls is relatively small, which means the network link state in the last time slot can be used in the next time slot. Thus, some combination of multipliers used in the last time slot could be assigned as the initial multipliers in the current time slot. If we appropriately assign initial values, the algorithm will probably speed up to converge, instead of requiring more iteration.

The overall procedure of the LR-based RTCAC algorithm is illustrated in Figure 5.x. The associated input parameters are: λ (new call arrival rate), τ (average time of call holding), and η (time budget for CAC). The detailed algorithms for each process in the procedure are as follow.

```
[Initialization]
    a)  Generate nodes and links to construct the topology;
    b)  Set UB*=0, LB*=-∞;
    c)  Set initial Lagrangean Multipliers π^0=1, where π is a
        multiplier vector;
    d)  Set iteration counter k=0, improvement counter m=0;
    e)  Set the number of CAC rounds(T);
    f)  Set the number of nodes in a group (GN);
    g)  Set Update_Counter_Limit=5;

[Generate New Calls]
    a)  Generate the number of new arrival groups (λ_t,t≤T);
    b)  Set NewCallCount=0;
    c)  do
        {
          Randomly select a node as the source and GN-1 nodes as
          destinations;
          Randomly generate the traffic requirement of the group;
          NewCallCount += NewCallCount;
        }until NewCallCount=λ_t;

[Get Dual Solution]
    a)  k=k+1, m=m+1;
    b)  Get dual Decision Variables (DVS) to calculate LB^k on Z_IP;

[Get Primal Solution]
    a)  Get Primal feasible solutions to calculate UB^k on Z_IP subject
        to constraints;

[Update Bounds]
    a)  Check LB
        If LB^k>LB*, then  LB*=LB^k;
    b)  Check UB
        If UB^k<UB*, then  UB*=UB^k;

[Update Parameters and Multipliers]
    a)  If (m = Update_Counter_Limit) { Scalar of step size = Scalar
        of step size / 2  };
    b)  Update the multipliers;

[Call Admission]
    a)  According to the DVs of UB*, admit the groups.
    b)  Calculate the number of terminated calls (TerminatedCalls);
    c)  do
        {
            Randomly select a group;
```

```
        If (isRemainCallFlag=1) {Set isRemainCallFlag=0}
        TerminatedCount+=1;
    } until TerminatedCount = TerminatedCalls;
```

[Multipliers Initialization]

a) Set initial Lagrangean Multipliers of the K+1 round $\pi^{k+1}(\lambda_g, \theta_{gl}, \varepsilon_g)$ = 1;
b) Continue using $\pi^k(\beta_1)$ as the initial multipliers in K+1 rounds.

5.5.2 Performance Metrics

Simulations are conducted to compare the LR-based algorithm with the simple algorithm (SA). For a fair comparison, we control the arrival process so that the simple algorithm uses the same arrival rate as the LR-based algorithm. Real-time CAC is based on a series of events. All arriving calls are aggregated.

To effectively analyze real-time CAC, we consider six performance metrics, namely, system revenue, blocking probability, number of admitted destinations (Resource utilization), average number of iterations within a time slot, average and maximum improvement ratio within a time slot, and long-term system accumulated revenue. A detailed description of these measures follows.

1) **System revenue (SR)**: The primary goal of the real-time CAC is to maximize system revenue in a near-realistic scenario. We will show the system revenue in different network topologies under different arrival rates, user mean holding times, and time budgets.

2) **Number of admitted destinations (AD)** and **blocking ratio (BR)**: In our model, the objective function is to maximize the system revenue. We do not consider the number of admitted destinations. The performance metric, AD, shows the number of admitted destinations after a period of time under different sets of simulation parameters. The blocking ratio is calculated by $(\lambda_{T-1} - \alpha_T)/\lambda_{T-1}$, which is a simplified expression of call blocking analysis. It is the most important measure in our evaluation of the CAC mechanism.

3) **Average number of iterations within a time slot (NI)**: By increasing the time budget, the LR-based algorithm can do more iteration within a time slot to improve the solution quality. However, the problem size and problem complexity also increase

when the time budget is increased. The performance metric, NI, is a simple indicator for viewing the complexity of the problem.

4) M**aximum and average improvement ratio within a time slot (MIR and AIR)**: In our proposed LR-based algorithm, we use the result of the simple algorithm (SA) as the initial upper bound. The performance metrics MIR and AIR show the maximum and average improvement ratio, respectively, between the LR-based algorithm and the simple algorithm within a given time budget.

5) **Average Error Gap (AEG):** The performance metric AEG is the average gap between UB and LB, computed by ∑[(UB-LB)/LB*100%]/number of rounds. It illustrates the optimality of the problem solution. The smaller the gap computed, the better the optimality.

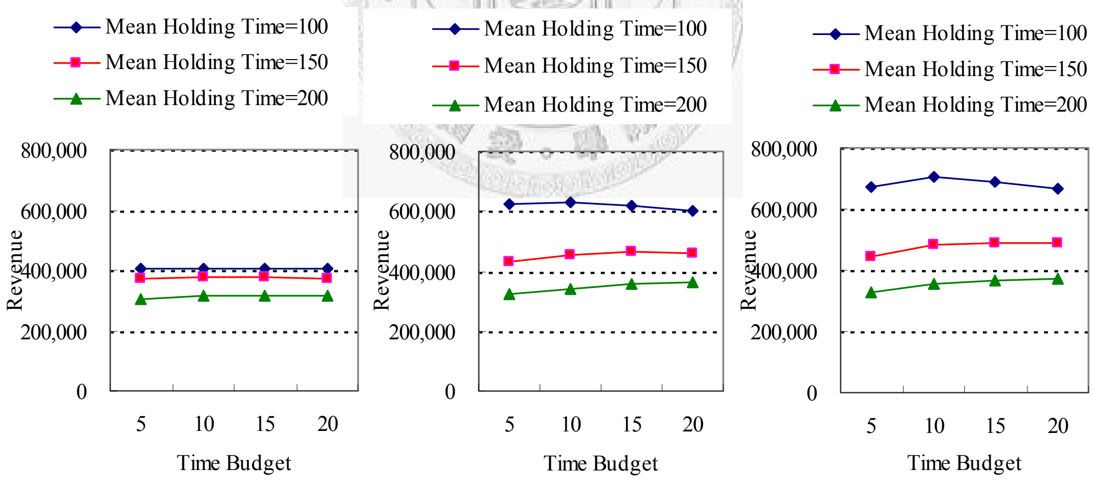
6) **Long-term system accumulated revenue (SAR)**: The performance metric SAR compares the long-term simulation results of the LR-based algorithm and the SA algorithm. Our proposed LR-based algorithm admits as many destinations as possible to maximize the revenue of one round, and does not consider the system capacity for the next round. The SAR metric compares the accumulated revenue of the LR-based algorithm (RLR) with that obtained by the SA algorithm (RSA) from a long-term viewpoint. The SAR is computed by [(RLR-RSA)/RLR].

5.5.3 Simulation Results

Four types of topologies are tested in the simulation, namely grid networks, cellular networks, random networks, and scale-free networks. The tested topologies are the same as these in Section 5.4. For the purposes of statistical analysis, the simulation duration is 15,000 sec. After 1,000 second, the system is expected to reach a steady state, and the final analysis report is based on the result after it reaches steady state. We examine the effect of the following three factors on the performance measures: (1) The time budget: real-time CAC is fulfilled subject to the time budget η, where 5, 10, 15, and 20 seconds are selected. (2) The average group holding time is another key factor that directly affects the number of remaining destinations, for which we choose 100, 150, and 200 seconds. (3) The effect of the number of group arrivals (λ) on performance analysis is considered. Assuming that the number of admitted users is proportional to the call arrivals, if more users arrive, the system will become busier. Arrivals not only provide a parameter, but also act as an indicator to evaluate the stability of the proposed CAC mechanism. From the overall system viewpoint, three cases of λ=5, 10, and 15 are examined to see how arrivals affect admission performance. In our simulations, the number of members in a group is 10, and the group revenue is drawn from a random variable.

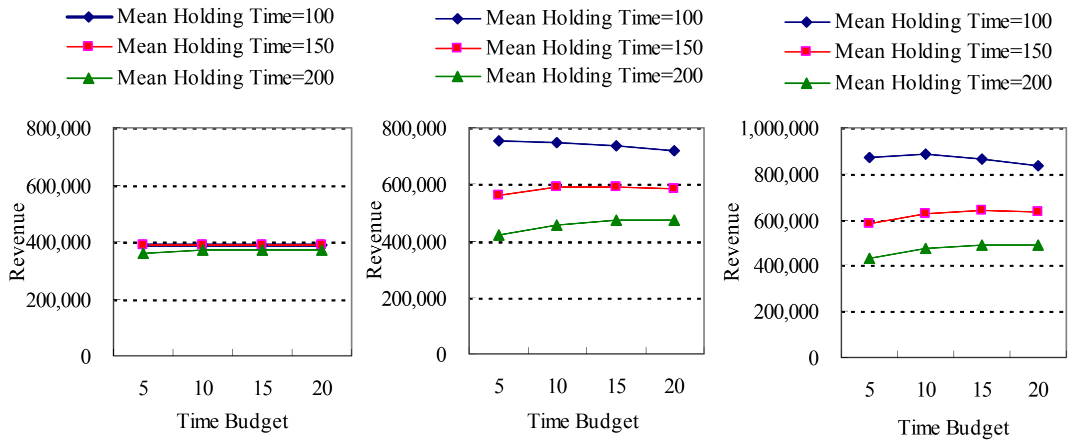
1) **System revenue (SR)**

Theoretically, the larger the time budget given, the greater the revenue received. However, Figures 5.4 - 5.7 show that the influence of the time budget on system revenue is not significant. For lightly loaded cases, such as Figure 5.5(a) and Figure 5.7(a), almost all destinations are admitted. The extended time budget does not contribute to the system's revenue, but reduce user satisfaction instead. This is because users have to wait a long time before they are admitted or rejected. For heavily loaded cases, for example in Figure 5.4(c) with λ=15, τ=100, extending the time budget from 10 to 15 seconds does indeed improve the system's revenue. However, when the time budget is stretched to 15 or 20 seconds, system revenue declines.



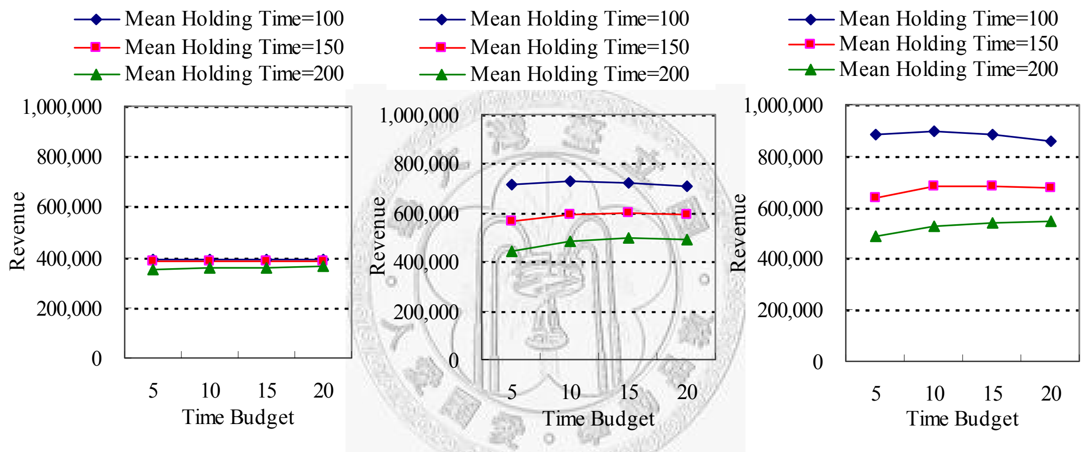
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.4: Effect of the time budget on the SR of a Grid Network (PCAC-S)



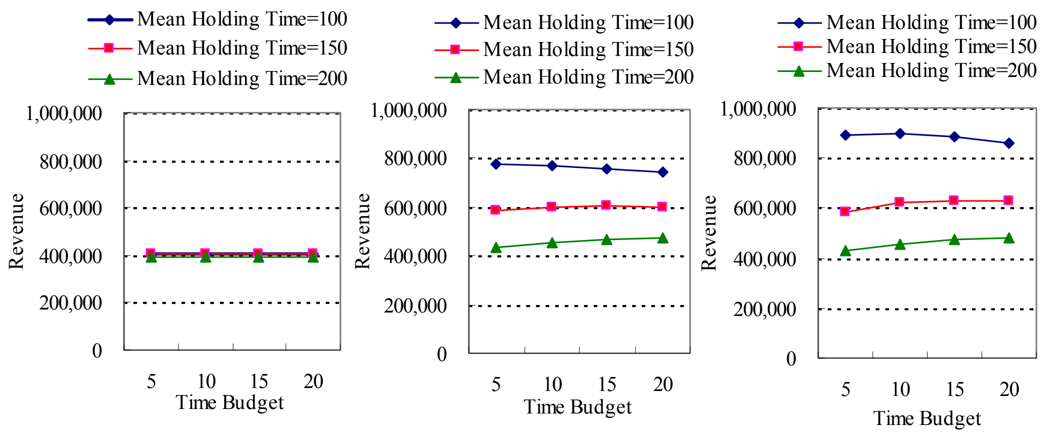
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.5: Effect of the time budget on the SR of a Cellular Network (PCAC-S)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.6: Effect of the time budget on the SR of a Random Network (PCAC-S)

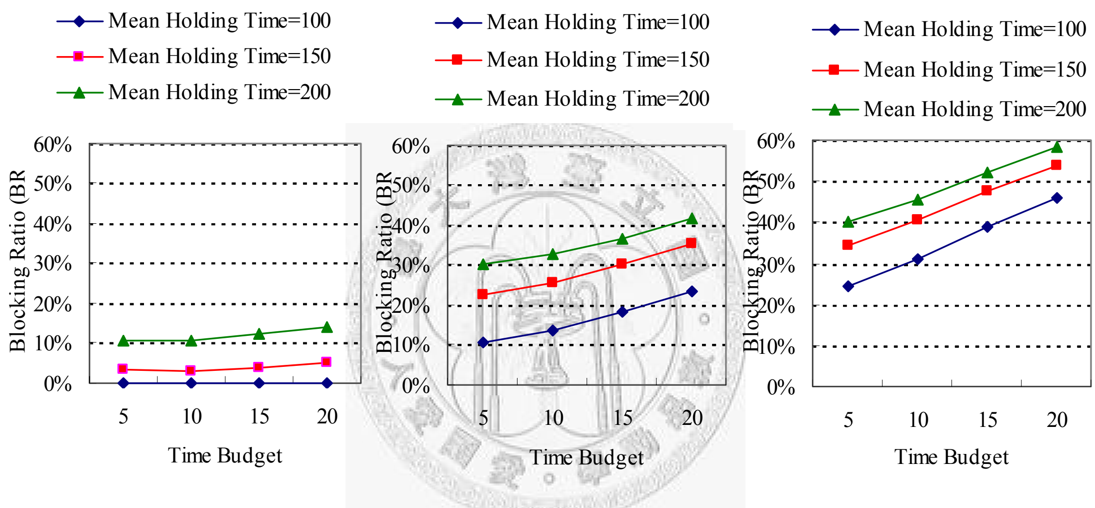


(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.7: Effect of the time budget on the SR of a Scale-free Network (PCAC-S)

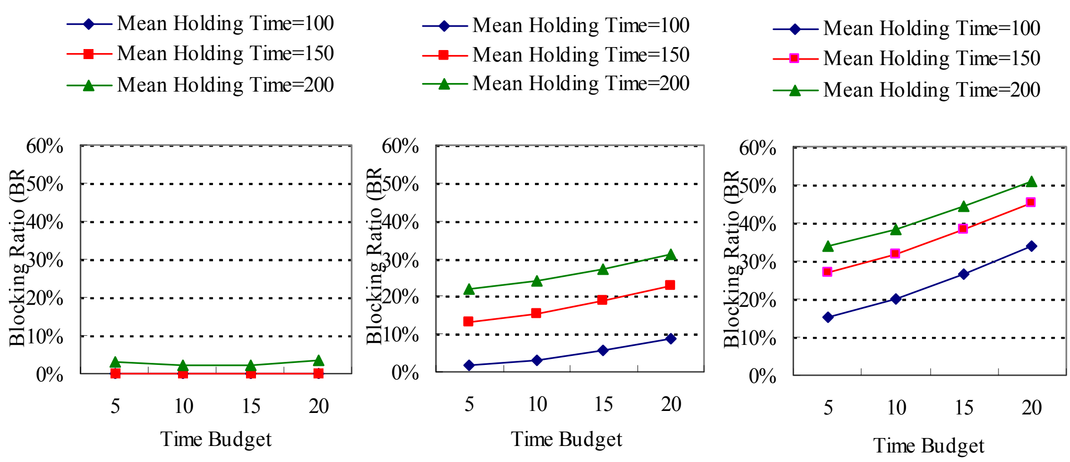
2) Number of admitted destinations (AD) and blocking ratio (BR):

Figures 5.8 - 5.11 show that the BR is an increasing function of time budget (η), and Figures 5.12 - 5.15 show that the AD is a decreasing function of time budget (η) in all cases of four different topologies. Unavoidably, the larger λ and τ given, the larger the BR calculated. This is because the objective function is to maximize system revenue. It does not try to maximize the number of admitted destinations. When the LR-based algorithm has a larger time budget, it tries to admit more valuable destinations, which may decrease the number of admitted destinations. By jointly considering the SR, the increase in the time budget does not increase the system's revenue, but it does increase the blocking ratio.



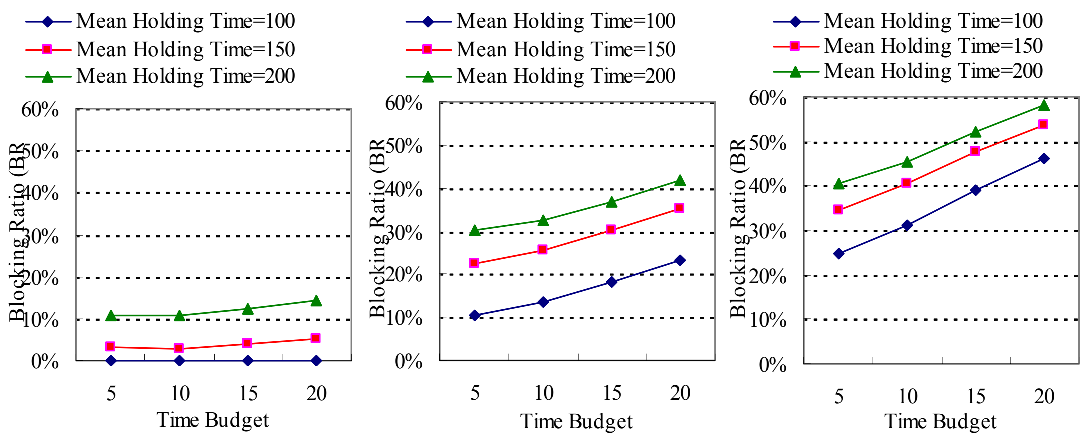
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.8: Effect of the time budget on the BR of a Grid Network (PCAC-S)



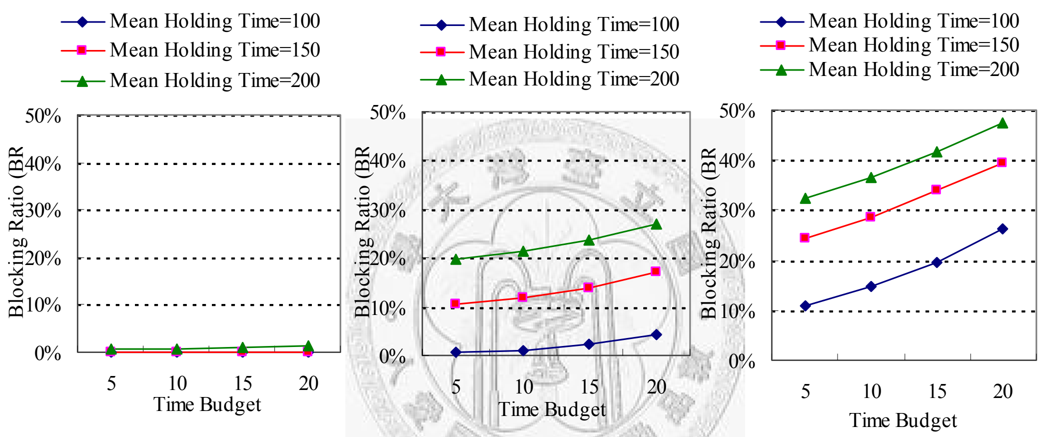
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.9: Effect of the time budget on the BR of a Cellular Network (PCAC-S)



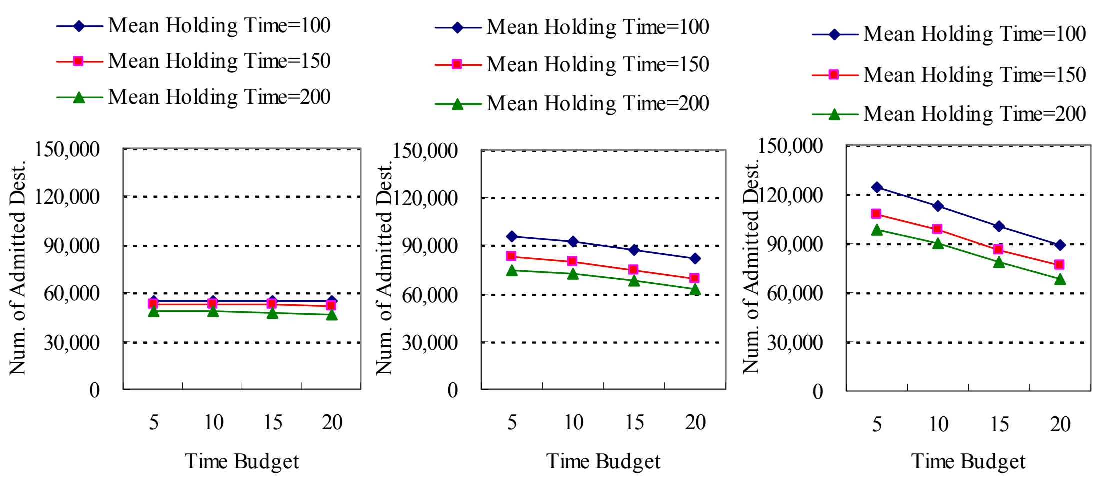
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.10: Effect of the time budget on the BR of a Random Network (PCAC-S)



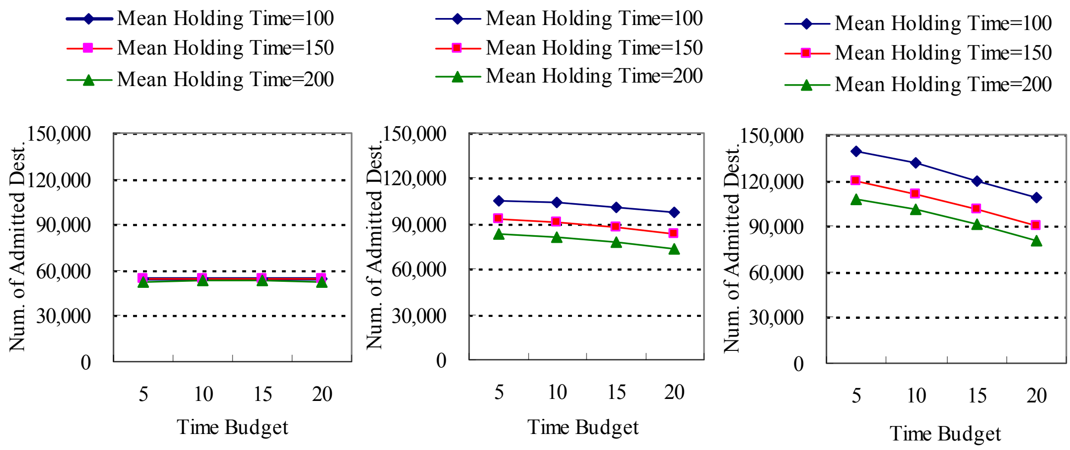
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.11: Effect of the time budget on the BR of a Scale-free Network (PCAC-S)



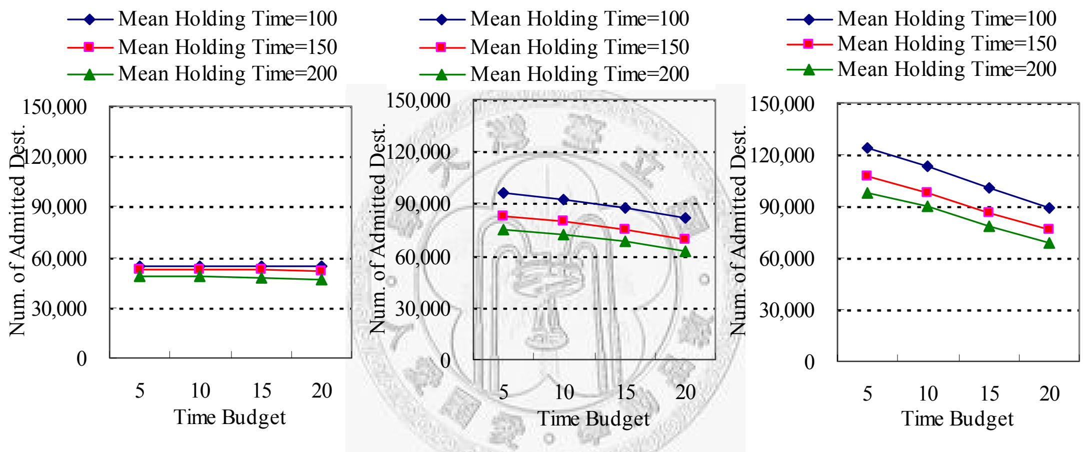
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.12: Effect of the time budget on the AD of a Grid Network (PCAC-S)



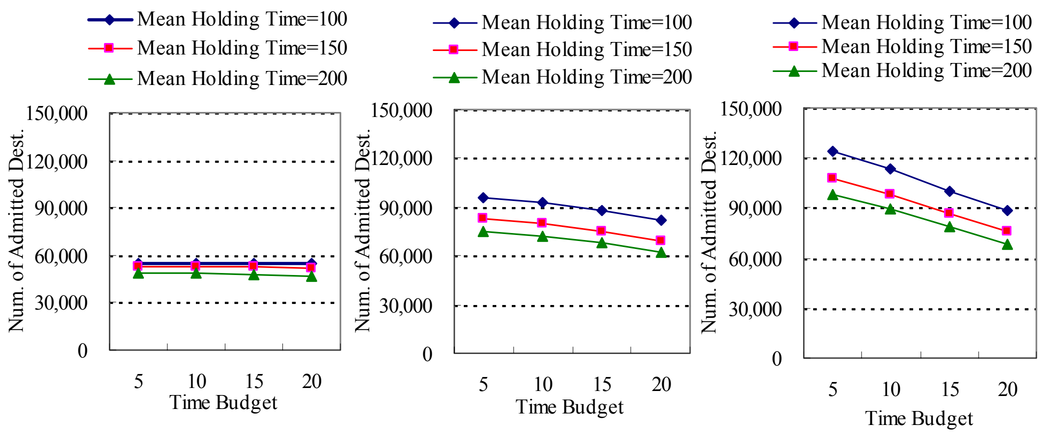
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.13: Effect of the time budget on the AD of a Cellular Network (PCAC-S)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.14: Effect of the time budget on the AD of a Random Network (PCAC-S)

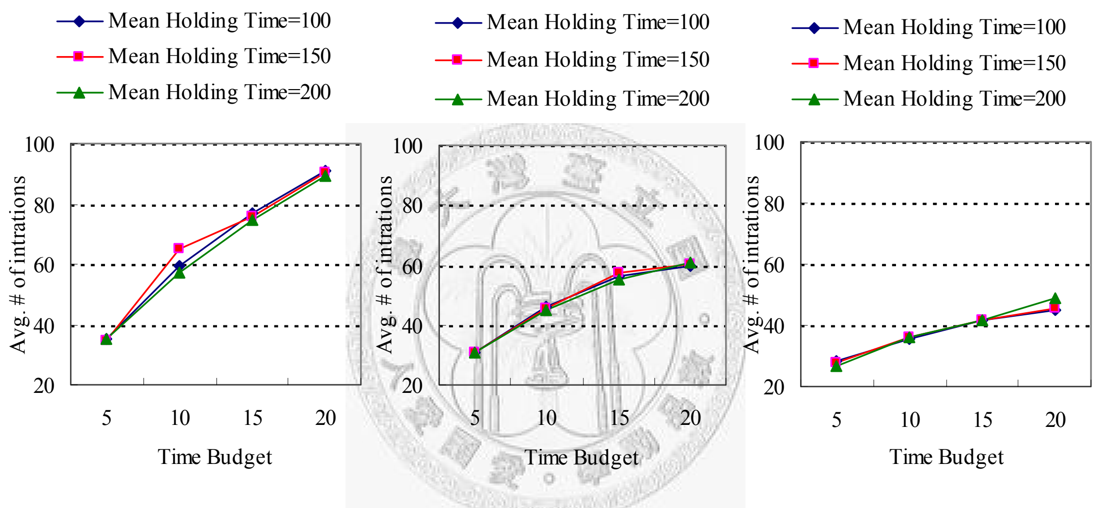


(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.15: Effect of the time budget on the AD of a Scale-free Network (PCAC-S)

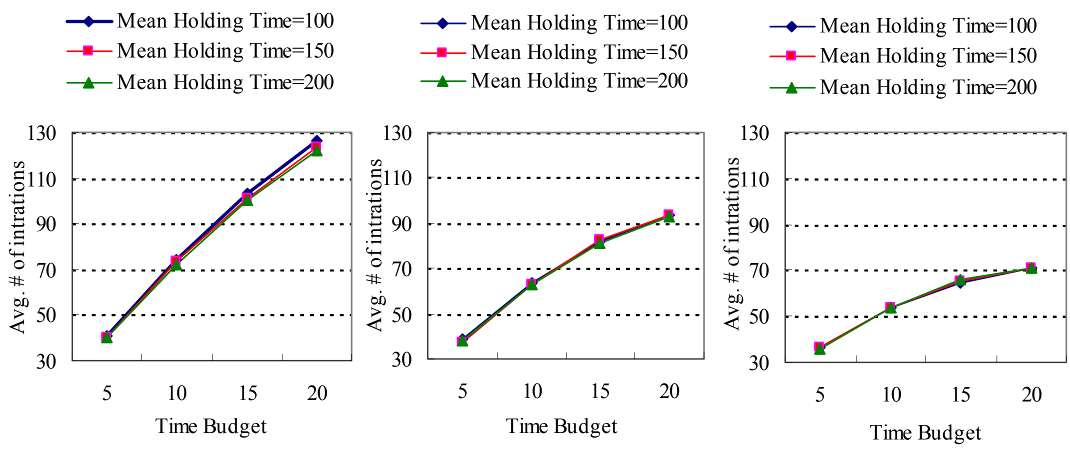
3) Average number of iterations within a time slot (NI)

With the increase in the time budget, the LR-based algorithm can perform more iterations within a time slot to improve the solution quality. However, the problem size and problem complexity also increase with a larger time budget. For the lightly loaded cases in Figures 5.16(a), 5.17(a), 5.18(a), and 5.19(a), NI is an increasing function with respect to the time budget. However, for the heavily loaded cases in Figures 5.16(c), 5.17(c), 5.18(c), and Figure 5.19(c), the increasing tendency of NI becomes lower. Furthermore, NI is not a monotonically increasing function of the time budget. In Figure 5.18(b), the number of iterations is 52 with λ=10, τ=200, and η=10. When η=25, the number of iterations is 50.



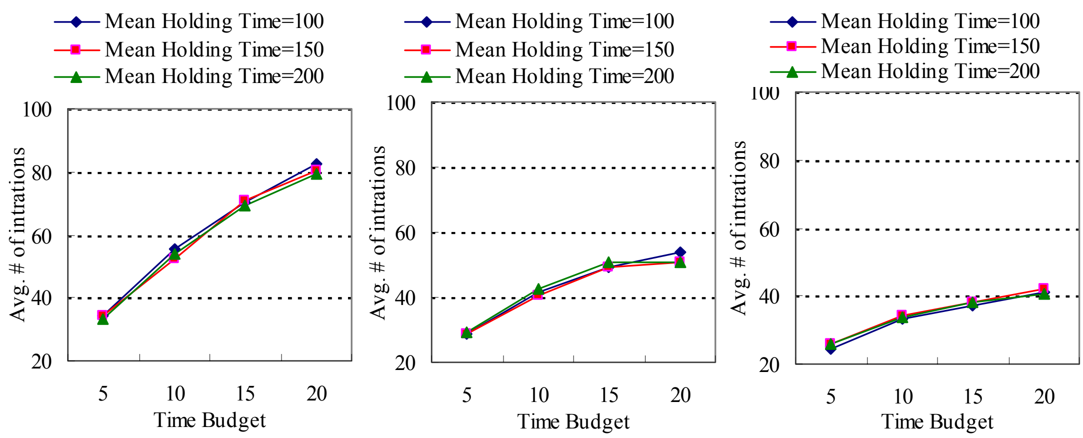
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.16: Effect of the time budget on the NI of a Grid Network (PCAC-S)



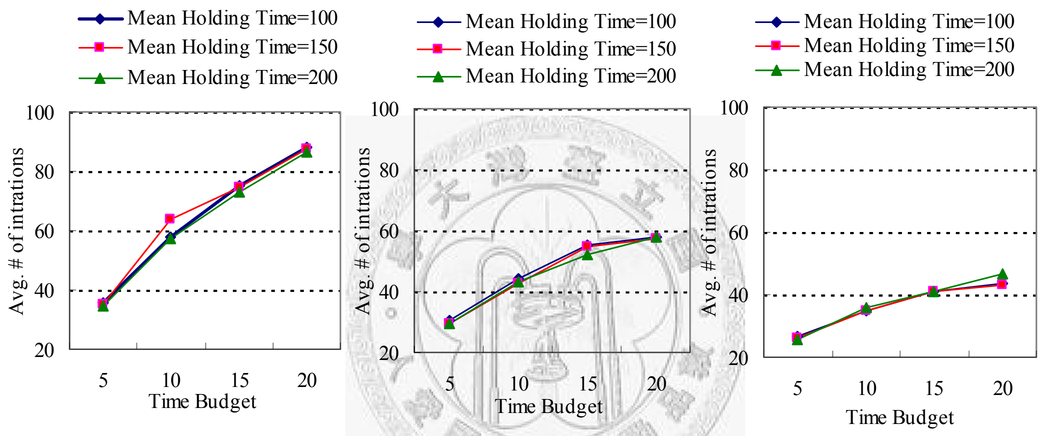
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.17: Effect of the time budget on the NI of a Cellular Network (PCAC-S)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.18: Effect of the time budget on the NI of a Random Network (PCAC-S)

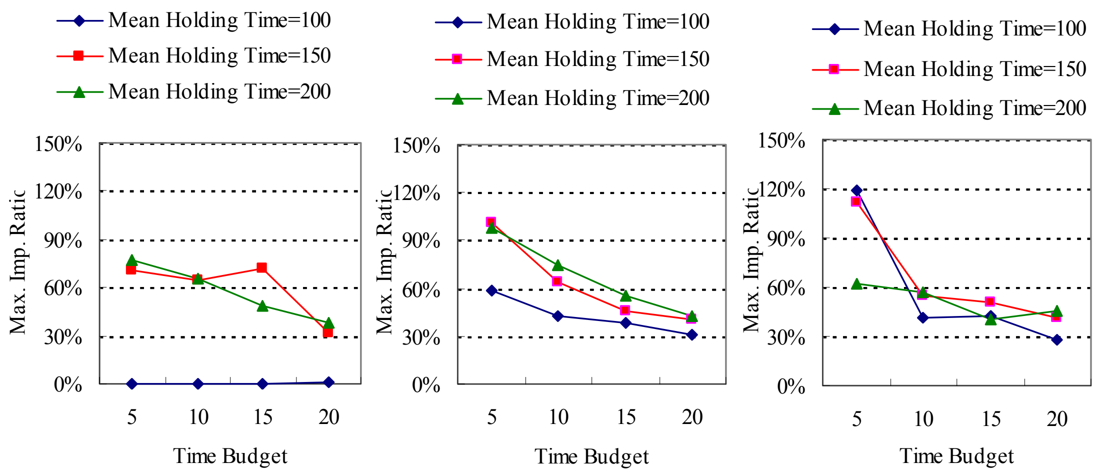


(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.19: Effect of the time budget on the NI of a Scale-free Network (PCAC-S)

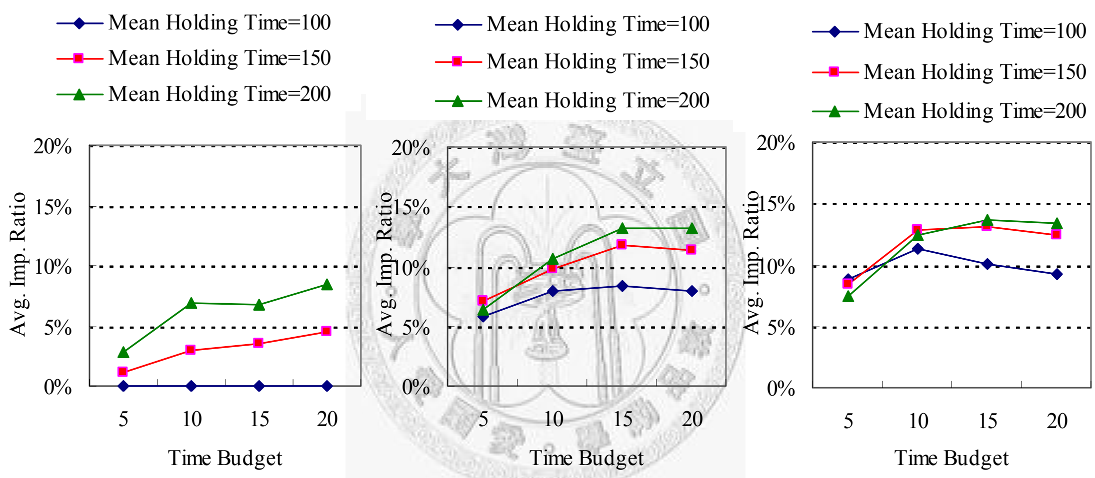
4) **Average and maximum improvement ratio within a time slot (AIR and MIR)**:

For each test network, within a time slot, the maximum improvement of the Lagrangean-based heuristic compared to the simple heuristic is 119.28 %, 157.34%, 100.68 %, and 70.85%, respectively. The average improvement ratio between the simple heuristic and the Lagrangean based heuristic within a time slot for different network topologies is shown in Figures 5.21, 5.23, 5.25, and 5.27, respectively. Although the time budget is only 5 to 20 seconds, the LR-based algorithm clearly outperforms the simple algorithm.



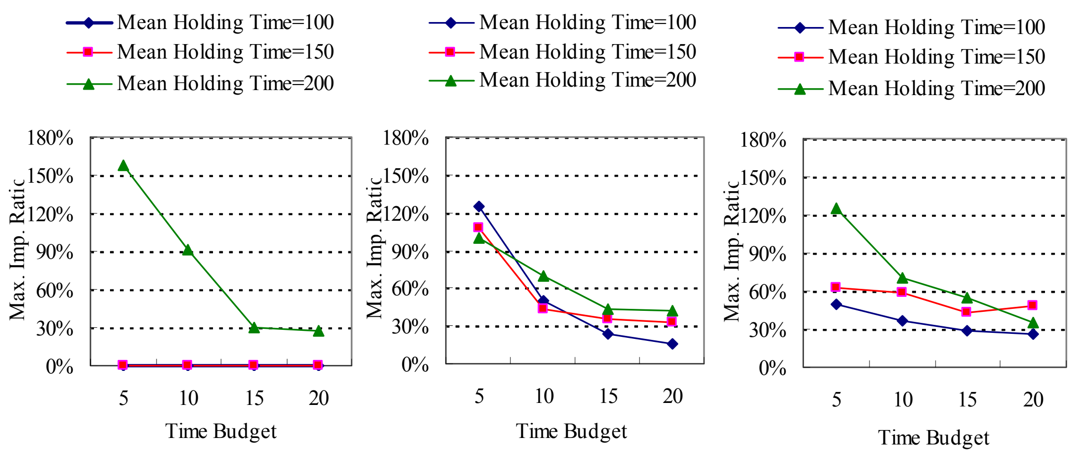
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.20: Effect of the time budget on the MIR of a Grid Network (PCAC-S)



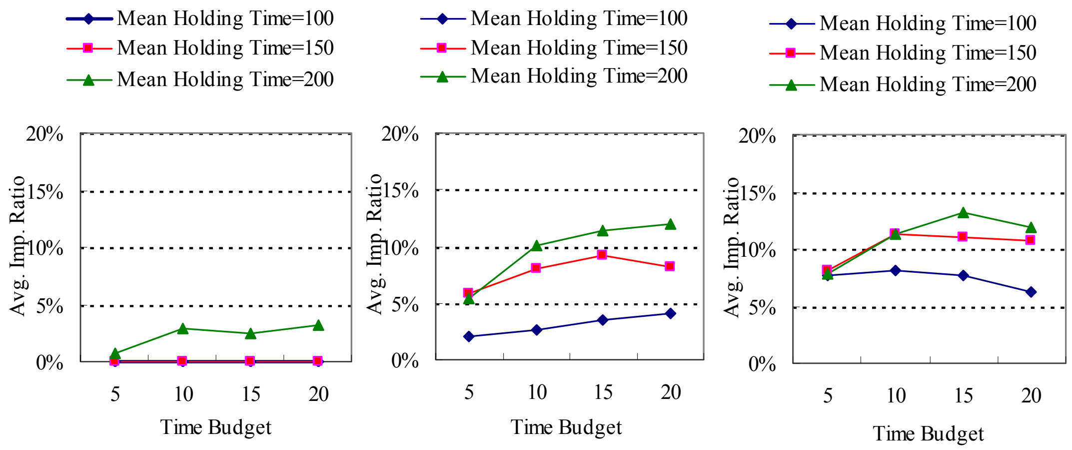
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.21: Effect of the time budget on the AIR of a Grid Network (PCAC-S)



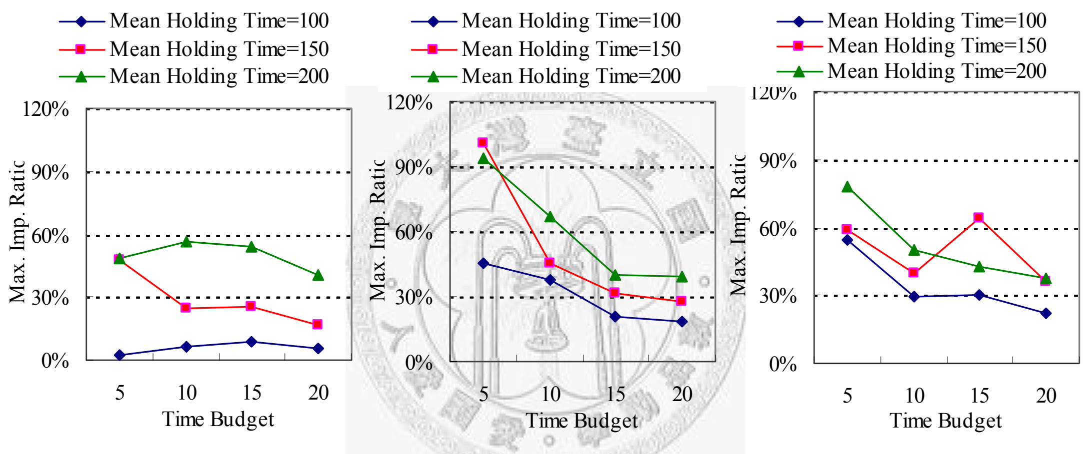
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.22: Effect of the time budget on the MIR of a Cellular Network (PCAC-S)



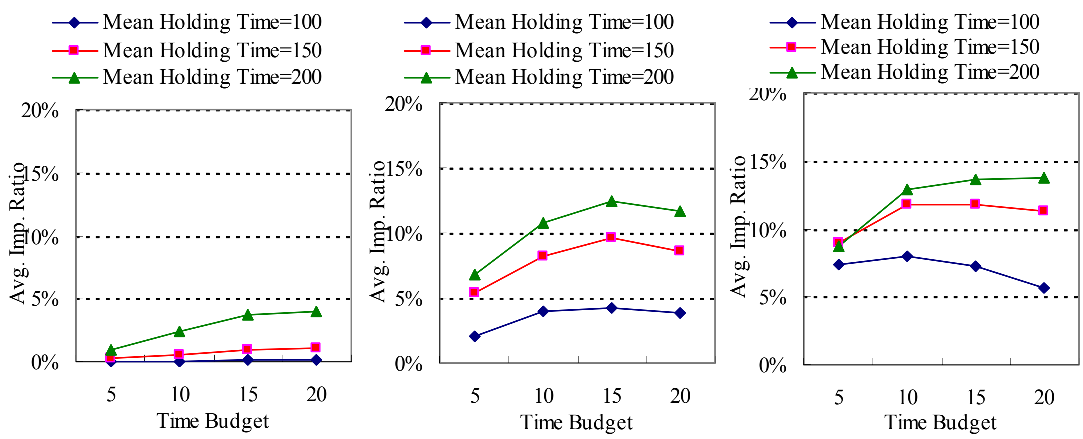
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.23: Effect of the time budget on the AIR of a Cellular Network (PCAC-S)



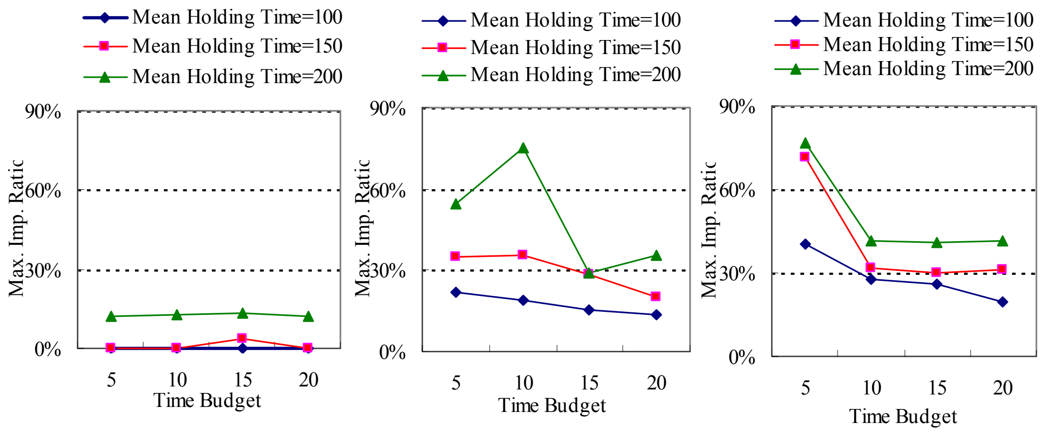
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.24: Effect of the time budget on the MIR of a Random Network (PCAC-S)



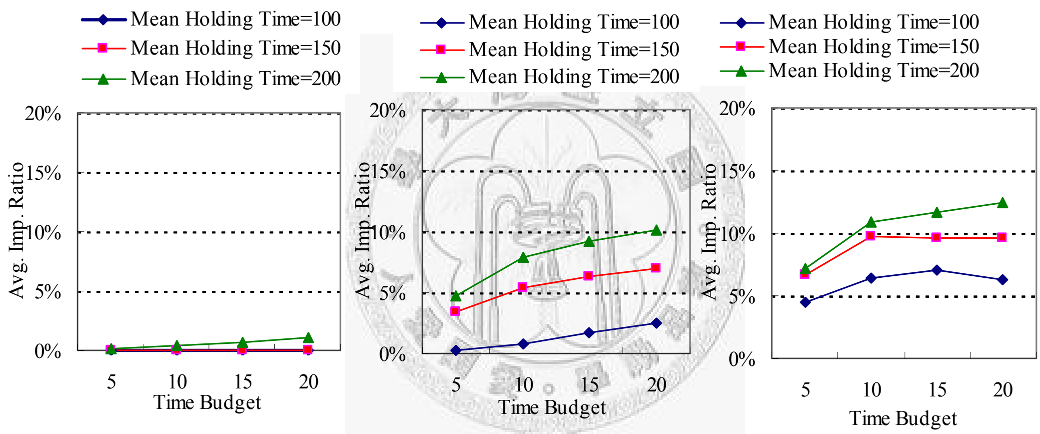
(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.25: Effect of the time budget on the AIR of a Random Network (PCAC-S)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.26: Effect of the time budget on the MIR of a Scale-free Network (PCAC-S)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 5.27: Effect of the time budget on the AIR of a Scale-free Network (PCAC-S)

5) **Average Error Gap (AEG):**

Table 5.4 illustrates the optimality of the problem solution obtained by the LR-based algorithm. The gap is relatively large compared with the gap shown in Table 5.2. The average gap is between 50.95% and 95.66%, because the execution time of the algorithm is very short; thus, subgradient method does not converge quickly. For lightly loaded cases in a gird network with λ=5, because the LR-based algorithm can perform more iterations, the gap is smaller than in heavily loaded cases.

Table 5.4: Average error gap for solving PCAC-S problem

Grid Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	95.14%	89.48%	82.66%	74.57%	82.97%	87.42%	84.98%	90.29%	93.02%
10	82.92%	61.04%	69.27%	72.78%	79.82%	84.82%	85.19%	90.47%	93.33%
15	58.33%	60.38%	68.34%	72.07%	77.71%	83.31%	87.92%	92.33%	94.60%
20	50.61%	61.55%	67.11%	74.00%	79.27%	84.29%	91.32%	94.29%	95.54%
Cellular Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	94.16%	93.54%	91.73%	82.84%	78.20%	83.21%	79.49%	86.55%	90.30%
10	90.58%	91.43%	71.18%	70.11%	75.58%	80.33%	80.63%	85.81%	89.84%
15	82.10%	73.78%	65.18%	71.18%	75.54%	79.66%	83.80%	87.35%	90.79%
20	71.78%	65.22%	65.48%	73.40%	77.78%	80.83%	87.24%	89.93%	92.19%
Random Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	96.21%	96.06%	95.99%	94.55%	94.35%	94.71%	93.21%	93.71%	94.76%
10	93.66%	94.83%	90.61%	83.72%	84.60%	86.54%	86.68%	89.45%	91.85%
15	88.12%	83.26%	79.93%	79.37%	81.93%	84.24%	89.48%	91.45%	93.21%
20	79.03%	74.67%	73.42%	81.75%	84.67%	86.78%	92.11%	90.62%	92.02%
Scale-free Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	95.86%	95.51%	94.71%	91.35%	89.02%	90.84%	87.83%	91.78%	94.01%
10	91.72%	87.15%	82.66%	77.55%	82.50%	86.60%	86.40%	91.35%	93.63%
15	80.80%	76.46%	73.70%	75.64%	80.51%	85.61%	87.66%	92.09%	94.37%
20	73.94%	71.38%	71.16%	77.24%	81.37%	85.66%	90.43%	93.65%	94.92%

A large gap does not necessarily mean that the solution quality is not good. However, we propose another method to calculate a more precise bound that does not change the solution accuracy. As shown in Figure 5.28, we insert time slot Ω between every two original time slots. Within Ω, we do not execute the getting primal feasible solution algorithm, but continue solving a series of Lagrangean relaxation problems and use the subgradient method to update the multipliers. So that we can continue improving the lower bound, we do not change the primal solution of the simulation result. In our experiment, we assign Ω=100 seconds. The modified average error gap of the grid networks, cellular networks, random networks, and scale-free networks are 39.71%, 41.65%, 42.79%, 39.13% respectively.

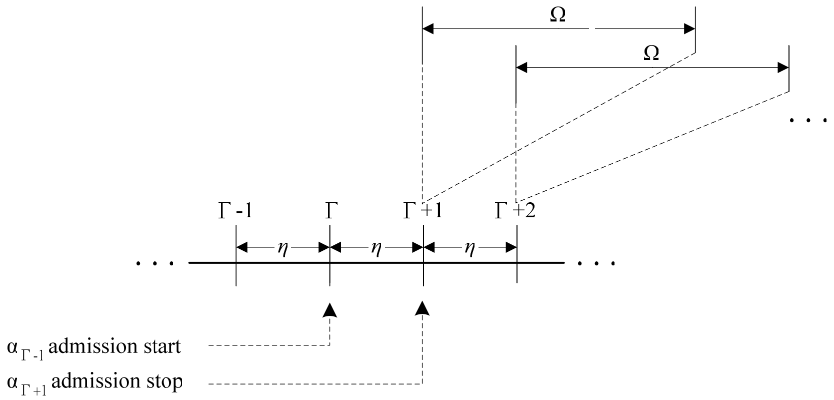


Figure 5.28: Lower bound calculated for real time simulation

6) **Long-term system accumulated revenue (SAR)**:

The SAR metric compares the accumulated revenue of the LR-based algorithm with that obtained by the SA algorithm from a long-term viewpoint. As shown in Table 5.5, the improvement ratio of the LR-based algorithm over the SA for the lightly loaded cases is very small. Even in some cases, the SA may better than the result of LR-based algorithm. However, the difference between them is very small and can not find out the tendency. For heavily loaded cases, such as the scale-free network with τ=200 and λ=15, the accumulated revenue of the SA are all better than that of LR-based algorithm. According the simulated results, the improvement ratio increases with the traffic load in the network.

Table 5.5: Comparison of long-term system accumulated revenue (PCAC-S)

Grid Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	0.00%	0.10%	0.27%	0.97%	2.13%	2.05%	2.88%	3.52%	3.32%
10	0.00%	0.97%	0.99%	2.31%	4.10%	4.71%	5.53%	7.34%	6.30%
15	0.00%	0.41%	2.02%	2.36%	5.55%	6.96%	5.41%	8.40%	9.34%

20	0.01%	0.67%	3.15%	3.19%	6.59%	8.42%	5.12%	8.35%	9.38%
Cellular Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	0.00%	-0.02%	-0.31%	0.43%	0.24%	0.01%	1.81%	2.55%	2.35%
10	0.00%	0.00%	0.45%	0.10%	3.42%	3.65%	3.36%	5.80%	6.66%
15	0.00%	0.00%	0.02%	-0.49%	4.30%	6.83%	3.27%	6.51%	7.92%
20	0.00%	0.00%	0.10%	-0.35%	3.87%	5.87%	2.46%	6.05%	8.45%
Random Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	-0.01%	-0.44%	0.02%	-0.93%	3.29%	4.39%	3.94%	6.28%	6.59%
10	-0.01%	0.81%	-0.74%	-1.18%	5.28%	8.02%	4.32%	8.84%	10.97%
15	-0.07%	0.91%	0.88%	-1.31%	5.51%	10.10%	3.60%	8.28%	10.10%
20	0.02%	-0.80%	-1.06%	-1.67%	4.18%	7.35%	1.18%	7.47%	9.91%
Scale-free Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	0.00%	0.00%	-0.13%	0.11%	2.79%	3.59%	4.54%	7.53%	7.61%
10	0.00%	0.03%	0.24%	0.39%	4.92%	8.36%	5.04%	8.92%	10.05%
15	0.00%	0.02%	0.36%	0.06%	5.56%	8.89%	4.76%	8.97%	10.46%
20	0.00%	0.01%	0.05%	0.21%	5.60%	8.74%	4.92%	7.89%	10.27%

5.6 Concluding Remarks

We have attempted to solve the problem of capacitated max-revenue multicast routing and partial admission control for multimedia distribution. Our contribution in this paper can be expressed in terms of the mathematical formulation and experiment performance. In terms of the formulation, we have proposed a precise mathematical expression to model this problem efficiently. In terms of performance, the proposed Lagrangean-based heuristic outperforms the simple heuristic. By modifying some constraints, our model can easily be extended to deal with the constrained multicast routing and admission control problem for multi-layered multimedia distribution. These issues will be addressed in the next chapter.

CHAPTER 6 THE PARTIAL ADMISSION CONTROL PROBLEM OF MULTIRATE MULTICASTING

6.1 Introduction

In Chapters 3 and 4, we discussed the problem of min-cost routing problem for multirate multicasting. In Chapter 5, we proposed the concept of partial admission control. We now consider the routing and call admission control mechanisms jointly and intend to solve the problem of maximum-revenue multicast routing with a partial admission control mechanism for multirate multimedia distribution. For multirate video distribution, which is different from the conventional Steiner tree problem, each receiver can request a different quality of video. This means that each link's flow on a multicast tree is different and dependent on the maximum rate of the receiver sharing the link. The partial admission control mechanism means that the admission policy of the multicast group is not based on a traditional "all or none" strategy. Instead, it considers accepting portions of destinations for the requested multicast group. More specifically, for a given network topology, a given link capacity, the destinations of the multicast group, and the bandwidth requirement of each multicast node, we attempt to find a feasible admission decision and routing solution to maximize the revenue of the multicast trees.

The rest of this paper is organized as follows. In Section 6.2, we formally define the problem being studied, and also propose a mathematical formulation of max-revenue optimization. Section 6.3 applies Lagrangean relaxation as a solution approach to the problem. Section 6.4, describes the computational experiments. In Section 6.5, we simulate our algorithm in a real-time scenario. Finally, in Section 6.6, we present our conclusions and indicate the direction of future research.

6.2 Problem Formulation

The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group, which has one source and one or more destinations, is an application requesting transmission over the network. Given the network topology, the capacity of links and the bandwidth requirement of every destination of a user group, we want to jointly determine the following decision variables: (1) the routing assignment (a tree for multicasting, or a path for unicasting) of each admitted destination; and (2) the admitted number of destinations of each partially admitted multicast group. We assume that the multicasting is multirate.

Table 6.1: Description of notations (PCAC-M)

Given Parameters	
Notation	**Descriptions**
F_{gq}	Revenue generated by admitting partial users of multicast group g with propriety q, which is a function of f_{gq} and a_{gq}
a_{gq}	Revenue generated by admitting multicast group g with priority q
α_{gd}	Traffic requirement of destination d in multicast group g
G	The set of all multicast groups
V	The set of nodes in the network
L	The set of links in the network
Q	The set of priorities in the network
D_g	The set of destinations of multicast group g

T_{gq}	The set of destinations of priority q in multicast group g
C_l	The capacity of link l
I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of paths user d of multicast group g may use
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise
σ_{gd}	The indicator function which is 1 if priority q is selected for destination d and 0 otherwise

Decision Variables

Notation	Descriptions
x_{gpd}	1 if path p is selected for group g destined for destination d and 0 otherwise.
y_{gl}	1 if link l is on the sub-tree adopted by multicast group g and 0 otherwise.
m_{gl}	The maximum traffic requirement of the destination in multicast group g that are connected to the source through link l.
f_{gq}	The number of admitted destinations of priority q in multicast group g.

By formulating the problem as a mathematical programming problem, we solve it optimally to obtain a network that fits our goal, i.e., it ensures the network operator can earn maximum revenue by servicing partially admitted destinations.

This model is based on the following viable assumptions.

- The revenue from each partially admitted group can be fully characterized by two parameters: the total amount of admitted revenue of the group associated with a specific priority, and the number of admitted destinations of the specific priority.
- The revenue from each partially admitted group associated with a specific priority is a monotonically increasing function with respect to the number of admitted destinations of the specific priority.
- The revenue function from each partially admitted group associated with a specific priority is a convex function with respect to the entire admitted revenue of the group associated with the specific priority and the number of admitted destinations of the specific priority. However, the entire admitted revenue and the number of admitted destinations jointly may not be a concave function.

- The revenue from each partially admitted group associated with a specific priority is independent.

The notations used to model the problem are listed in Table 6.1.

Optimization Problem:

Objective function:

$$\min \ -\sum_{g\in G}\sum_{q\in Q} F_{gq}(a_{gq}, f_{gq}) \qquad \text{(IP 6)}$$

subject to:

$$\sum_{p\in P_{gd}} \alpha_{gd} x_{gpd} \delta_{pl} \le m_{gl} \qquad \forall g\in G, d\in D_g, l\in L \qquad (6.1)$$

$$\sum_{g\in G} m_{gl} \le C_l \qquad \forall l\in L \qquad (6.2)$$

$$m_{gl} \in [0, \max_{d\in D} \alpha_{gd}] \qquad \forall g\in G, l\in L \qquad (6.3)$$

$$\sum_{l\in L} y_{gl} \ge \sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd} \qquad \forall g\in G \qquad (6.4)$$

$$\sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} \le |D_g| y_{gl} \qquad \forall g\in G, l\in L \qquad (6.5)$$

$$\sum_{l\in I_v} y_{gl} \le 1 \qquad \forall g\in G, v\in V-\{r_g\} \qquad (6.6)$$

$$\sum_{l\in I_{r_g}} y_{gl} = 0 \qquad \forall g\in G \qquad (6.7)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall g\in G, p\in P_{gd}, d\in D_g \qquad (6.8)$$

$$\sum_{d\in D_g}\sum_{p\in p_{gd}} x_{gpd}\sigma_{qd} = f_{gq} \qquad \forall g\in G, q\in Q \qquad (6.9)$$

$$f_{gq} \in \{0,1,2,......,|T_{gq}|\} \qquad \forall g\in G, q\in Q \qquad (6.10)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l\in L, g\in G \qquad (6.11)$$

$$\sum_{p\in P_{gd}} x_{gpd} \le 1 \qquad \forall g\in G, d\in D_g. \qquad (6.12)$$

The objective function of (IP 6) is to maximize the total revenue, F_{qg}, by servicing the partially admitted destinations in multicast group g associated with a specific priority, where $g\in G$, $q\in Q$, and G is the set of user groups requesting transmission. F_{gq} reflects the priority of partial users belonging to group g, while different choices of F_{gq} may provide different physical meanings of the objective function. For example, if F_{gq} is chosen as the mean traffic requirement of partial users belonging to group g associated with priority q, then the objective function is to maximize the total system throughput. In general, if user group g with priority q is to be given a higher priority, then the corresponding F_{gq} may be assigned a larger value.

Constraints (6.1) and (6.2) are the capacity constraints. In this model, the variable m_{gl} can be viewed as the estimate of the aggregate flows. Since the objective function is strictly decreasing with m_{gl}, and (IP 6) is a maximization problem, each m_{gl} will be exactly equal to the aggregate flow in an optimal solution. Constraint (6.3) is a redundant constraint, which provides the upper and lower bounds of the maximum traffic requirement for multicast group g on link l. Constraint (6.4) requires that if one path is selected for group g destined for destination d, it must also be on the sub-tree adopted by multicast group g. Constraint (6.5) is the tree constraint, which requires that the union of the selected paths for the destinations of user group g forms a tree. Constraints (6.4) and (6.6) require that the number of selected incoming links, y_{gl}, is 1 or 0 and each node, except the root, has only one incoming link. Constraint (6.6) requires that the number of selected incoming links, y_{gl}, to node is 1 or 0. Constraint (6.7) requires that no selected incoming link, y_{gl}, is the root of multicast group g. As a result, the links we select form a tree. Constraints (6.8) and (6.12) require that at most one path is selected for each admitted multicast source-destination pair, while Constraint (6.9) relates the routing decision variables x_{gpd} to the auxiliary variables f_{gq}. Constraint (6.10) requires that the number of admitted destinations in multicast group g with priority q is a set of integers.

6.3 Solution Procedure

6.3.1 Lagrangean Relaxation

By using the Lagrangean Relaxation method, we can transform the primal problem (IP) into the following Lagrangean Relaxation problem (LR 6), where Constraints (6.1) (6.4) (6.5), and (6.9) are relaxed. For a vector of Lagrangean multipliers, the Lagrangean Relaxation problem of (IP6) is given by

Optimization problem (LR):

$$Z_{D6}(\beta,\lambda,\theta,\varepsilon)=\min\ -\sum_{g\in G}\sum_{q\in Q}F_{gq}(a_{gq},f_{gq})+\sum_{g\in G}\sum_{l\in L}\sum_{d\in D_g}\sum_{p\in P_{gd}}\beta_{gdl}\alpha_{gd}x_{gpd}\delta_{pl}$$
$$-\sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L}\beta_{gdl}m_{gl}+\sum_{g\in G}\sum_{l\in L}\lambda_g y_{gl}-\sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}\lambda_g x_{gpd}$$
$$+\sum_{g\in G}\sum_{l\in L}\sum_{d\in D_g}\sum_{p\in P_{gd}}\theta_{gl}x_{gpd}\delta_{pl}-\sum_{g\in G}\sum_{l\in L}\theta_{gl}\left|D_g\right|y_{gl}$$
$$+\sum_{g\in G}\sum_{q\in Q}\sum_{d\in D_g}\sum_{p\in P_{gd}}\varepsilon_{gp}x_{gpd}\sigma_{qd}-\sum_{g\in G}\sum_{q\in Q}\varepsilon_{gq}f_{gq}$$

(LR 6)

subject to: (2)(3)(6)(7)(8)(10)(11)(12),

where β_{gdl}, λ_g, θ_{gl}, and ε_{gq} are Lagrangean multipliers and $\beta_{gd}, \theta_{gl} \geq 0$. To solve (LR 6), we can decompose it into the following four independent and easily solvable optimization subproblems.

Subproblem 1: (related to decision variable x_{gpd})

$$Z_{Sub6.1}(\beta,\lambda,\theta,\varepsilon)=\min\sum_{g\in G}\sum_{d\in D_g}\sum_{p\in P_{gd}}\Big(\sum_{l\in L}\delta_{pl}(\beta_{gdl}\alpha_{gd}+\theta_{gl})+\sum_{q\in Q}\varepsilon_g\sigma_{gd}-\lambda_g\Big)x_{gpd}$$

(SUB 6.1)

subject to:

$$x_{gpd}=0 \text{ or } 1 \qquad \forall g\in G, p\in P_{gd}, d\in D_g \qquad (6.8)$$

$$\sum_{p\in P_{gd}}x_{gpd}\leq 1 \qquad \forall g\in G, d\in D_g. \qquad (6.12)$$

The Subproblem (SUB 6.1) is to determine x_{gpd}. It can be further decomposed into $|G||D_g|$ independent shortest path problems with nonnegative arc weights $\beta_{gdl}\alpha_{gd}+\theta_{gl}$. If the shortest cost plus coefficient $\sum_{q\in Q}\varepsilon_g\sigma_{gd}-\lambda_g$ is no more than 0, than we assign the corresponding x_{gpd} to 1, and 0 otherwise.

Subproblem 2: (related to decision variable y_{gl})

$$Z_{Sub6.2}(\lambda,\theta)=\min\sum_{g\in G}\sum_{l\in L}(\lambda_g-\theta_{gl}\left|D_g\right|)y_{gl} \qquad \textbf{(SUB 6.2)}$$

subject to:

$$\sum_{l \in I_v} y_{gl} \le 1 \qquad \forall g \in G, v \in V - \{r_g\} \quad (6.6)$$

$$\sum_{l \in I_{r_g}} y_{gl} = 0 \qquad \forall g \in G \quad (6.7)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l \in L, g \in G. \quad (6.11)$$

The Subproblem (SUB 6.2) can be decomposed into |G| independent problems. The algorithm to solve the Subproblem (SUB 6.2) is:

1. Compute the coefficient $\lambda_g - \theta_{gl} / |D_g|$ for all links in multicast group g.
2. Sort the links in ascending order according to the coefficient.
3. According to the order and complying with constraints (6.6) and (6.7), if the coefficient is less than zero, assign the corresponding negative coefficient of y_{gl} to 1; otherwise 0.

Subproblem 3: (related to decision variable f_{gq})

$$Z_{Sub6.3}(\varepsilon) = \min \ -\sum_{g \in G} \sum_{q \in Q} (F_{gq}(a_{gq}, f_{gq}) + \varepsilon_{gq} f_{gq}) \qquad \text{(SUB 6.3)}$$

subject to:

$$f_{gq} \in \{0,1,2,......,|T_{gq}|\} \qquad \forall g \in G, q \in Q. \quad (6.10)$$

We can easily solve Subproblem (SUB 6.3) optimally by exhaustively searching from the known set of f_{gq}.

Subproblem 4: (related to decision variable m_{gl})

$$Z_{Sub6.4}(\beta) = \min \ \sum_{l \in L} (\sum_{g \in G} (-\sum_{d \in D_g} \beta_{gdl}) m_{gl}) \qquad \text{(SUB 6.4)}$$

subject to:

$$\sum_{g \in G} m_{gl} \le C_l \qquad \forall l \in L \qquad (6.2)$$

$$m_{gl} \in [0, \max_{d \in D} \alpha_{gd}] \qquad \forall g \in G, l \in L. \qquad (6.3)$$

We can decompose and solve Subproblem (SUB6.4) into |L| independent problems using the following algorithm:

Step 1 Compute $-\sum_{d \in D_g} \beta_{gdl}$ for link l of multicast group g.

Step 2 Sort the negative coefficient $-\sum_{d \in D_g} \beta_{gdl}$ from the smallest to the largest value

Step 3 According the sorted sequence: <i> assign the corresponding m_{gl} to the maximum traffic requirement in the multicast group and add to the sum value until the total amount of the maximum traffic requirement on link l is less than the capacity of link l. <ii> assign the boundary negative coefficient of m_{gl} to the difference between the capacity on link l and the sum value of m_{gl}, <iii> assign the other coefficients of m_{gl} to 0.

According to the weak Lagrangean duality theorem, for any β_{gdl}, $\theta_{gl} \ge 0$, $Z_D(\beta_{gdl}, \lambda_g, \theta_{gl}, \varepsilon_{gq})$ is a lower bound on Z_{IP6}. The following dual problem (D) is then constructed to calculate the tightest lower bound.

Dual Problem (D):

$$Z_{D6} = \max Z_{D6}(\beta_{gdl}, \lambda_g, \theta_{gl}, \varepsilon_{gq}),$$

subject to: $\beta_{gdl}, \theta_{gl} \ge 0$.

There are several methods for solving the dual problem (D). The most popular is the subgradient method [8], which we employ here.

6.3.2 Getting Primal Feasible Solutions

After optimally solving the Lagrangean relaxation problem, we have a set of decision variables. However, this solution is not feasible for the primal problem, since some constraints are not satisfied. Thus, minor modification of the decision variables, or the hints of multipliers, must be considered in order to obtain the primal feasible solution of a problem (IP). Generally speaking, the best primal feasible solution is an upper bound (UB) of the problem (IP), while the Lagrangean dual problem solution guarantees the lower bound (LB) of problem (IP). Iteratively, by solving the Lagrangean dual problem and getting the primal feasible solution, we get the LB and UB, respectively. So, the gap between the UB and LB, computed by (UB-LB)/LB*100%, illustrates the optimality of problem solution. The smaller gap computed, the better the optimality.

Here we propose a comprehensive, two-part method to obtain a primal feasible solution. It utilizes a Lagrangean multiplier-based heuristic, followed by adjustment procedures. While solving the Lagrangean relaxation dual problem, we may find some multipliers related to each OD pair and links, which could make our routing more efficient. We now describe the Lagrangean-based heuristic.

[Lagrangean Multipliers based heuristic]

Step 1 Use λ_g-$\theta_{gl}/D_g/$ as link l's arc weight and run the M-T-M heuristic [10] to obtain a spanning tree for each multicast group.

Step 2 **Drop procedures**:

2.1 Check the capacity constraint of each link. If there is a link violation of the capacity constraint, go to Step 2.2; otherwise go to Step 3.

2.2 Sort the links in descending order according to {C_l – the aggregate flow on the link}. Choose the maximal overflow link and drop the group with the maximal subgradient ($-F_{gq}(a_{gq}, f_{gq})-\varepsilon_{gq}f_{gq}$). Go to Step 2.1.

Step 3 **Add procedures:**

3.1 Sort the dropped group in ascending order according to the subgradient ($-F_{gq}(a_{gq}, f_{gq})-\varepsilon_{gq}f_{gq}$).

3.2 In accordance with the order, re-add the groups to the network. Use λ_g-$\theta_{gl}/D_g/$ as link l's arc weight, remove the overflow links from the graph and run the M-T-M heuristic.

If it can not find a route for the destinations, drop the destinations.

6.4 Computational Experiments

In this section, computational experiments on the Lagrangean relaxation based heuristic and other primal heuristics are reported. The heuristics are tested on three kinds of networks: regular networks, random networks, and scale-free networks.

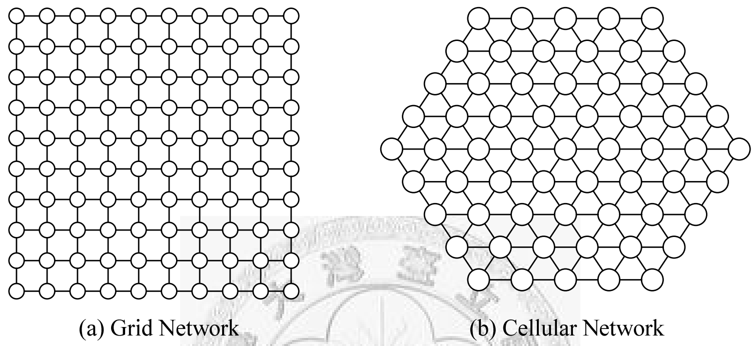


Figure 6.1: Regular networks (PCAC-M)

We test regular networks, shown in Figure 6.1, in our experiments. The first one is a grid network contains 100 nodes and 180 links, and the second is a cellular network containing 61 nodes and 156 links. Random networks tested in the experiments are generated randomly, each having 100 nodes. The candidate links between all node pairs are given a probability that follows the uniform distribution. In the experiments, we link the node pair with a probability smaller than 2%. Reference [9] shows that the scale-free networks can be developed from a simple dynamic model that combines incremental growth with a preference for new nodes to connect to existing ones that are already well connected. In our experiments, we apply this *preferential attachment method* to generate the scale-free networks. The number of nodes in the test networks is 100.

In order to prove that our heuristics are good enough, we also implement a simple algorithm for comparison with our heuristic.

[Simple Algorithm]

Step 1 Set link *l*'s arc weight to 1 and run the M-T-M heuristic to get a spanning tree for each multicast group.

Step 2 **Drop procedures**:

2.1 Check the capacity constraint of each link. If there is a link violation of the capacity constraint, go to Step 2.2, otherwise go to Step 3.

2.2 Sort the links in descending order according to {C_l - the aggregate flow on the link}. Choose the maximal overflow link and drop the nodes of the group that has the maximum flow on that link. Go to Step 2.1.

Step 3 **Add procedures:**

3.1 Sort the dropped group in ascending order according to the group ID and node ID.

3.2 In accordance with the order, re-add the groups to the network. Remove the overflow links from the graph, set each link's arc weight to the aggregate flow of the link and run the M-T-M heuristic. If it can not find a feasible route for the destinations, drop the destinations.

For each test network, several distinct cases with different pre-determined parameters, such as the link capacity, the number of multicast groups, and the number of nodes in each group, are considered. The traffic demands for each multicast group are drawn from a random variable uniformly distributed in pre-specified categories {1, 2, 5, 10, 15, 20}. We conducted 120 experiments for each kind of network. For each experiment, the result was determined by the group source and destinations generated randomly.

Table 6.2 summaries the selected results of the computational experiments. For each test network, the maximum improvement ratio between the simple heuristic and the Lagrangean based heuristic is 14.42 %, 23.73%, 22.70 %, and 25.22%, respectively. In general, the Lagrangean-based heuristic performs better than the simple heuristic. We also find that in less congested networks with fewer groups or destinations, the Lagrangean-based heuristic outperforms the simple heuristic, as in case A of the cellular network and case B of the scale-free network.

Table 6.2: Selected results of computational experiments (PCAC-M)

CASE	Cap.	G #	N #	SA	UB	LB	GAP	Imp.
Grid Networks						**Max Imp. Ratio: 14.42 %**		
A	20	20	20	-4049.86	-4615.56	-6279.5	26.50%	13.97%
B	20	20	50	-6892.2	-7413.97	-12741.7	41.81%	7.57%
C	20	50	20	-6995.97	-7861.03	-12495.7	37.09%	12.37%
D	20	50	50	-13198.5	-14230.2	-25305.7	43.77%	7.82%
E	20	100	20	-11141.4	-12747.9	-20387.3	37.47%	14.42%
F	20	100	50	-20751	-22121.2	-39519.2	44.02%	6.60%
Cellular Networks						**Max Imp. Ratio: 23.73 %**		
A	20	20	20	-2942.85	-3641.11	-4943.33	26.34%	23.73%
B	20	20	50	-6351.08	-7452.4	-11600.9	35.76%	17.34%
C	20	50	20	-7810.42	-8790	-13495.9	34.87%	12.54%
D	20	50	50	-14639.3	-15328.1	-27179.5	43.60%	4.71%
E	20	100	20	-11266.4	-12719.4	-20488	37.92%	12.90%
F	20	100	50	-23083.9	-23471.7	-42124.5	44.28%	1.68%
Random Networks						**Max Imp. Ratio: 22.70 %**		
A	20	20	20	-4192.43	-4927.5	-6104.6	19.28%	17.53%
B	20	20	50	-9366.11	-11492.2	-13222	13.08%	22.70%
C	20	50	20	-9111.25	-11073.7	-14217.8	22.11%	21.54%
D	20	50	50	-17207.1	-20381.1	-30306.7	32.75%	18.45%
E	20	100	20	-17614	-20959.7	-27758	24.49%	18.99%
F	20	100	50	-39313.6	-45728.5	-64584.5	29.20%	16.32%
Scale-free Networks						**Max Imp. Ratio: 25.22 %**		
A	20	20	20	-3380.06	-4120.19	-5075.61	18.82%	21.90%
B	20	20	50	-6662.23	-8342.77	-12229.2	31.78%	25.22%
C	20	50	20	-6714.48	-8176.26	-11380.1	28.15%	21.77%
D	20	50	50	-12933.3	-15112.9	-24120.7	37.34%	16.85%
E	20	100	20	-12276.8	-14648.1	-20221.3	27.56%	19.32%
F	20	100	50	-21033.6	-25898.6	-37790.4	31.47%	23.13%

Cap.: The capacity of each link
G#: The number of multicast groups
N#: The number of destinations in each multicast group
SA: The result of the simple algorithm
UB: Upper bounds of the Lagrangean-based heuristic
LB: Lower bounds of the Lagrangean-based heuristic
GAP: The error gap of Lagrangean relaxation
Imp.: The improvement ratio of the Lagrangean-based heuristic

The Lagrangean based heuristic outperforms than the simple algorithm for two reasons. First, it makes use of the related Lagrangean multipliers, including the potential cost for routing on each link in the topology. Second, the heuristic is iteration-based and is guaranteed to improve the solution quality iteration by iteration. Therefore, in a more complicated testing environment, the improvement ratio is higher. To claim optimality, the results show that most of the cases have a gap of less than 40%. We also find that the simple heuristic performs well in many cases, such as case B of the grid network and case D of the cellular network.

6.5 Real-time Partial Admission Control for Multirate Multicasting

In this section, we analyze the performance of our proposed LR-based algorithm in a real-time call admission scenario. The simulation scenario is the same as the one described in Section 5.5.

6.5.1 A Real-time LR-based CAC Algorithm

Based on the LR approach described in Section 6.3, a pre-defined time budget, η, e.g., 5 seconds, is given to solve the Lagrangean dual problem and get primal feasible solutions iteratively. Actually the time budget is equivalent to the time slot. In a specific time slot, real-time call admission control is fulfilled at the end of the time slot. The number of call requests admitted depends on the time budget.

The primal feasible solution algorithm described in Section 6.3 uses the multipliers as the arc weights. In a real-time scenario, the number of arriving and departing calls is relatively small, so the network link state in the last time slot can be used in the next time slot. Thus, some combination of multipliers used in the last time slot could be assigned as the initial multiplier in the new time slot. If we appropriately assign initial values, the algorithm will probably speed up and converge, instead of requiring more iterations.

The overall procedure of the LR based RTCAC algorithm is the same as shown in Figure 5.3. Associated input parameters are: λ (new call arrival rate), τ (average time of call holding), and η (time budget for CAC). The detailed algorithms for each process in the procedure are as follows:

```
[Initialization]
    a)  Generate nodes and links to construct the topology;
    b)  Set UB*=0, LB*=-∞;
    c)  Set initial Lagrangean Multipliers π0=1, where π is a
        multiplier vector;
    d)  Set the iteration counter k=0, and the improvement counter
        m=0;
    e)  Set the number of CAC rounds(T);
    f)  Set the number of nodes in a group (GN);
    g)  Set Update_Counter_Limit=5;
```

```
[Generate New Calls]

    a)  Generate the number of new arrival groups (λt,t≤T);
    b)  Set NewCallCount=0;
    c)  do
        {
          Randomly select a node as the source and GN-1 nodes as
          the destinations;
          Randomly generate the traffic requirement of the group;
          NewCallCount += NewCallCount;
        }until NewCallCount=λt;

[Get Dual Solution]

    a)  k=k+1, m=m+1;
    b)  Get dual Decision Variables (DVS) to calculate LB^k on Z_IP;

[Get Primal Solution]

    a)  Get Primal feasible solutions to calculate UB^k on Z_IP subject
        to constraints;

[Update Bounds]

    a)  Check LB
        If LB^k>LB^*, then  LB^*=LB^k;
    b)  Check UB
        If UB^k<UB^*, then  UB^*=UB^k;

[Update Parameters and Multipliers]

    a)  If (m = Update_Counter_Limit) { Scalar of step size = Scalar
        of step size / 2  };
    b)  Update the multipliers;

[Call Admission]

    a)  According the DVs of UB*, admit the groups.
    b)  Calculate the number of terminated calls (TerminatedCalls);
    c)  do
        {
            Randomly select a group;
            If (isRemainCallFlag=1) {Set isRemainCallFlag=0}
            TerminatedCount+=1;
        } until TerminatedCount = TerminatedCalls;

[Multipliers Initialization]

    a)  Set initial Lagrangean Multipliers of the K+1 round π^{k+1}(β_gdl,
        λ_g, ε_gq) = 1;
    b)  Set (∑_g θ_gl /Number of groups) as the initial multipliers θ_gl
        in K+1 rounds.
```

6.5.2 Performance Metrics

We conducted simulations to compare the LR-based algorithm with the simple algorithm (SA). For a fair comparison, we controlled the arrival process so that the simple algorithm used the same arrival rate as the LR-based algorithm. Real-time CAC is based on a series of events, and all arriving calls are aggregated

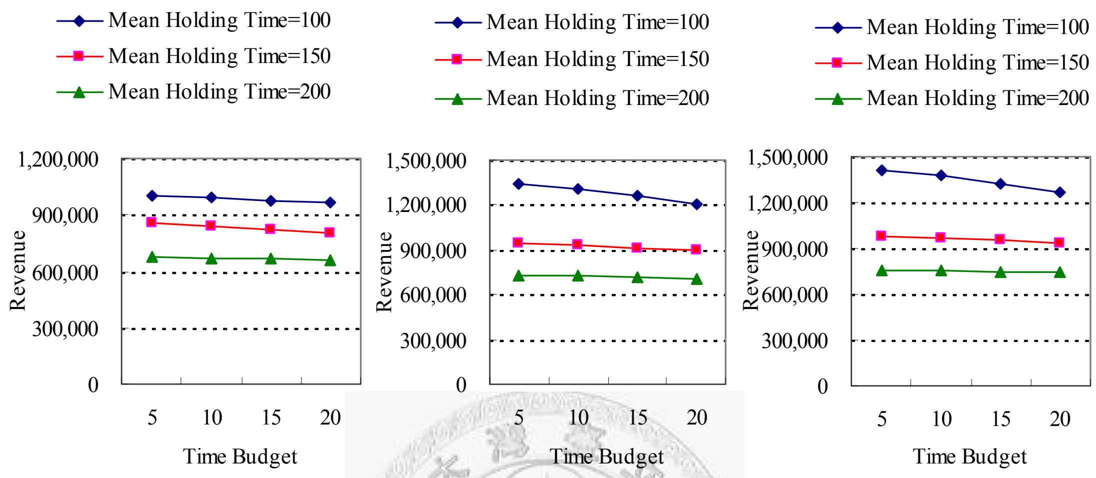
To effectively analyze real-time CAC, we consider six performance metrics: system revenue, blocking probability, number of admitted destinations (Resource utilization), average number of iterations within a time slot, average and maximum improvement ratios within a time slot, and long-term system accumulated revenue. Detailed descriptions of these measures is given in Section 5.5.2.

6.5.3 Simulation Results

Four types of topologies are tested in the simulation: grid networks, cellular networks, random networks and scale-free networks. The tested topologies are the same as shown in Section 6.4. For the purposes of statistical analysis, the simulation time is 15,000 sec. After 1,000 seconds, the system is expected to reach a steady state, and the final analysis report is based on the result after reaching the steady state. We examine the effect of the following three factors on the performance measures: (1) The time budget: real-time CAC is fulfilled subject to the time budget η, where 5, 10, 15, and 20 seconds are selected. (2) The average group holding time is another key factor that directly affects the number of remaining destinations, for which we choose 100, 150, and 200 seconds. (3) The effect of the number of group arrivals (λ) on performance analysis is considered. Assuming that the number of admitted users is proportional to the call arrivals, if more users arrive, the system will become busier. Arrivals not only provide a parameter, but also act as an indicator to evaluate the stability of the proposed CAC mechanism. From the overall system viewpoint, three cases of λ=5, 10, and 15 are examined to see how arrivals affect admission performance. In our simulations, the number of members in a group is 10, and the group revenue is drawn from a random variable.

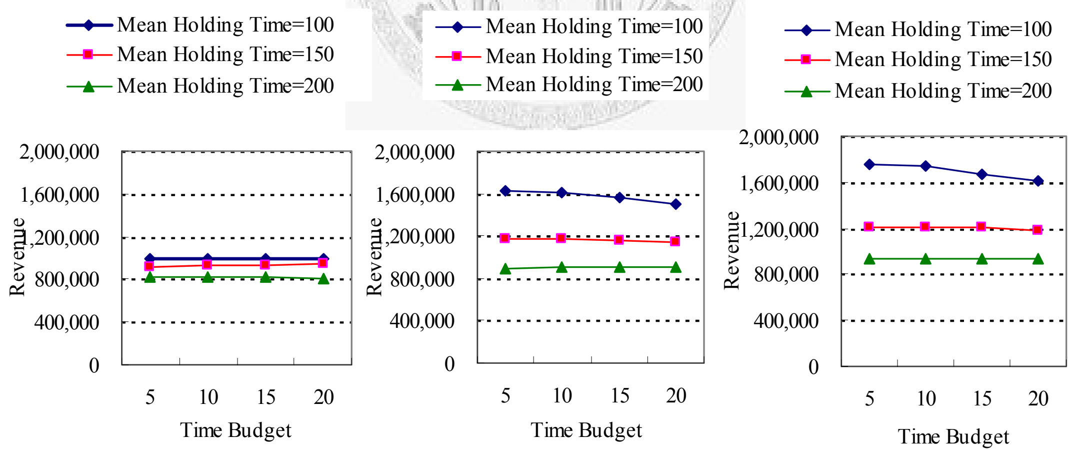
1) **System revenue (SR)**

Theoretically, the larger the time budget given, the greater the revenue received. However, Figures 6.2 - 6.5 show that the influence of the time budget on system revenue is not significant. For the lightly loaded cases, such as Figure 6.5(a), almost all destinations are admitted. The extended time budget does not contribute to the system's revenue, but reduces user satisfaction instead, because users have to wait a long time before they are admitted or rejected. For heavily loaded cases, for example in Figure 6.3(c) with λ=15, τ=100, extending the time budget does not improve the system's revenue. In almost all cases, SR is a decreasing function of the time budget (η).



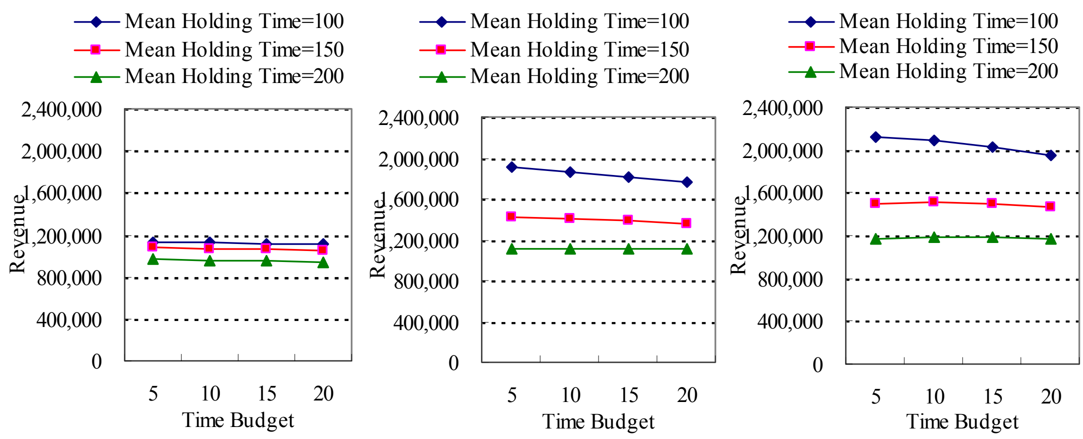
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.2: Effect of the time budget on the SR of a Grid Network (PCAC-M)



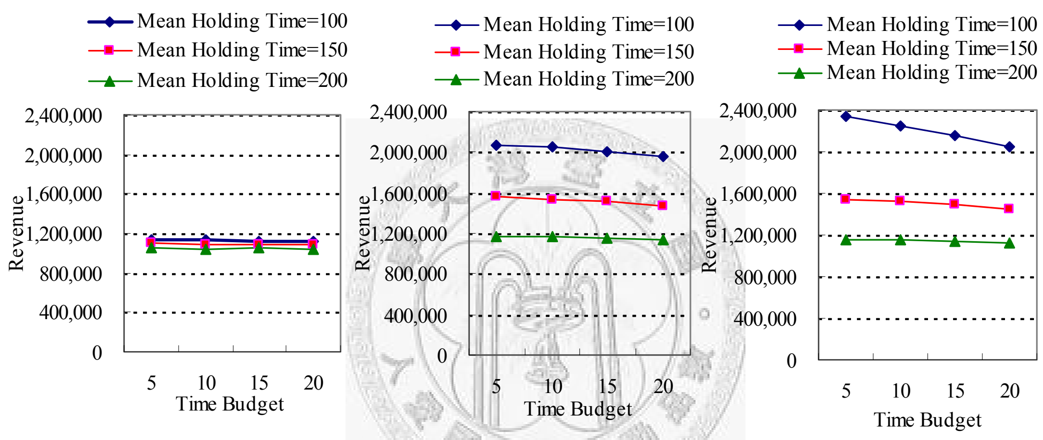
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.3: Effect of the time budget on the SR of a Cellular Network (PCAC-M)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.4: Effect of the time budget on the SR of a Random Network (PCAC-M)

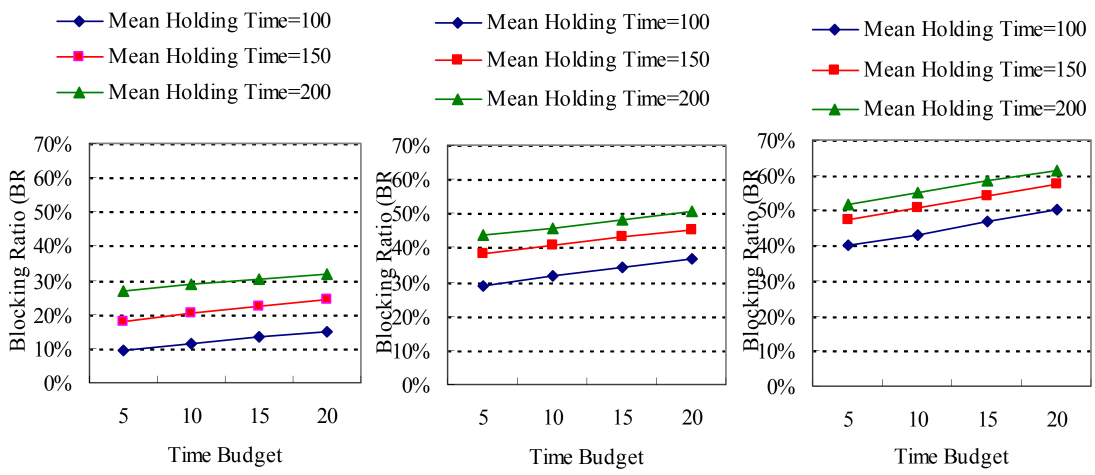


(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.5: Effect of the time budget on the SR of a Scale-free Network (PCAC-M)

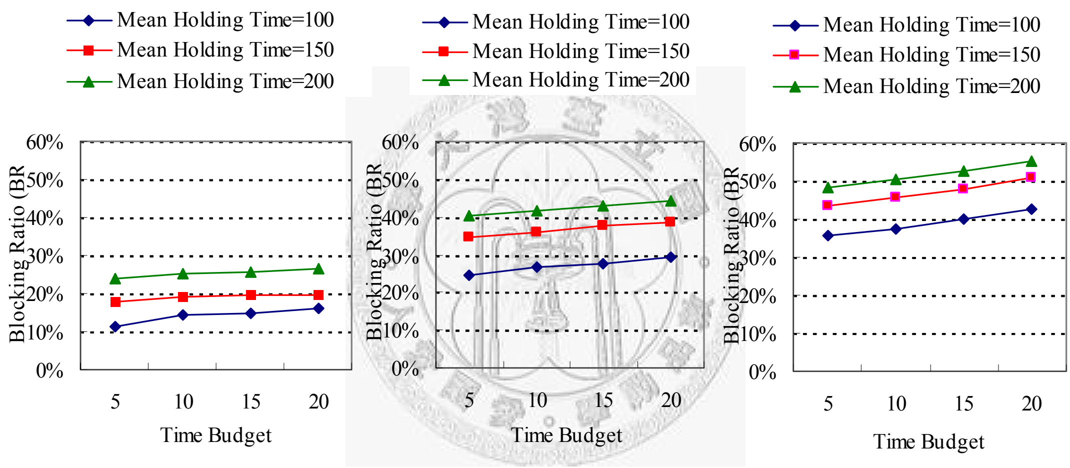
2) Number of admitted destinations (AD) and blocking ratio (BR):

Figures 6.6 - 6.9 show that BR is an increasing function of time budget (η), and Figures 6.10 - 6.13 show that AD is a decreasing function of time budget (η) in all cases for four different topologies. Unavoidably, the larger λ and τ given, the larger the BR calculated. This is because the objective function is to maximize system revenue. It does not try to maximize the number of admitted destinations. When the LR-based algorithm has a larger time budget, it tries to admit more valuable destinations, which may decrease the number of admitted destinations. By jointly considering the SR, the increase in time budget does not increase the system's revenue, but it does increase the blocking ratio.



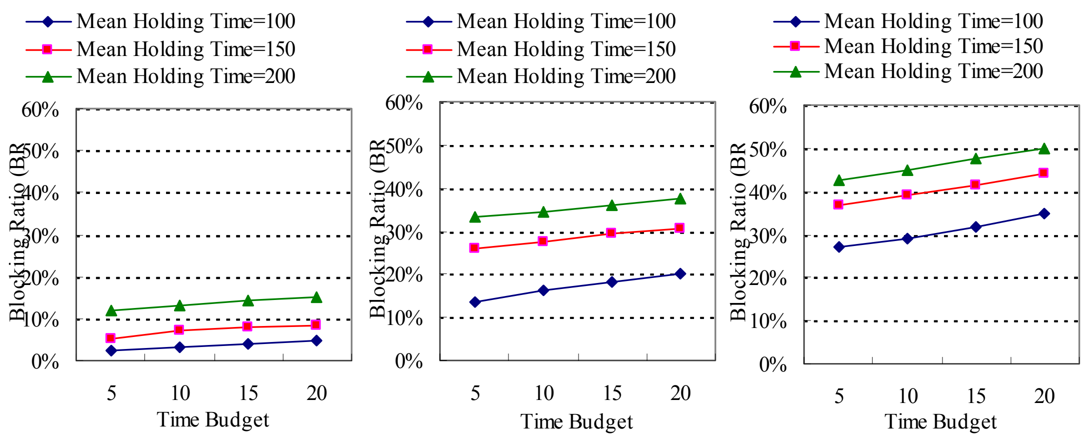
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.6: Effect of the time budget on the BR of a Grid Network (PCAC-M)



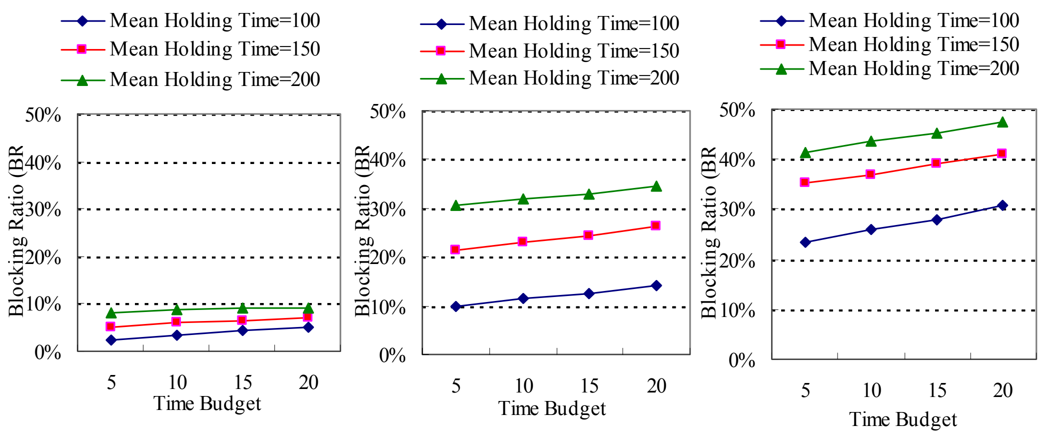
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.7: Effect of the time budget on the BR of a Cellular Network (PCAC-M)



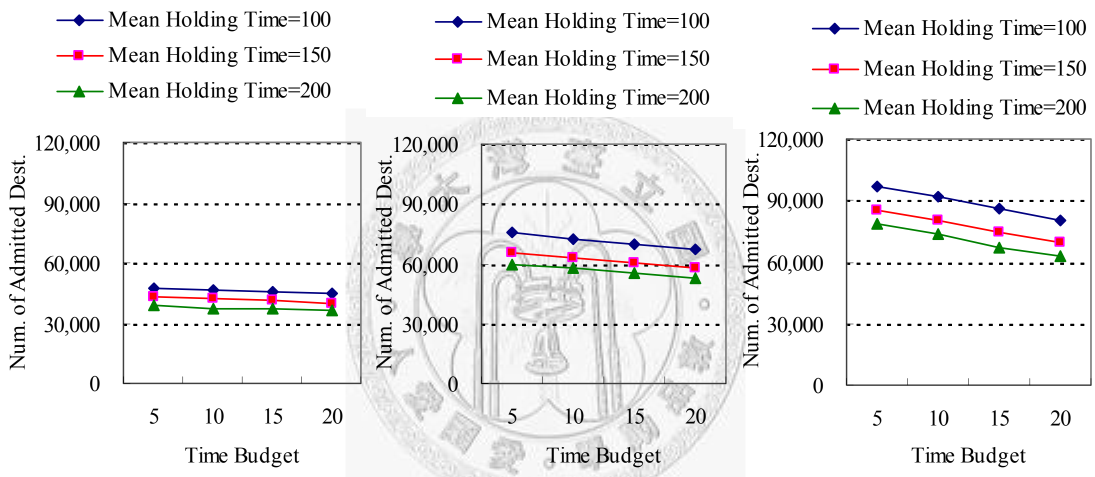
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.8: Effect of the time budget on the BR of a Random Network (PCAC-M)



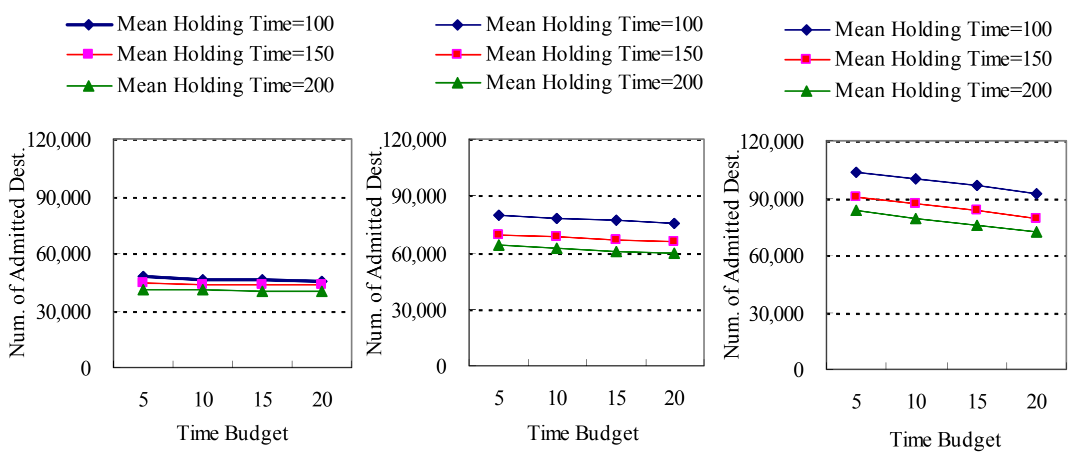
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.9: Effect of the time budget on the BR of a Scale-free Network (PCAC-M)



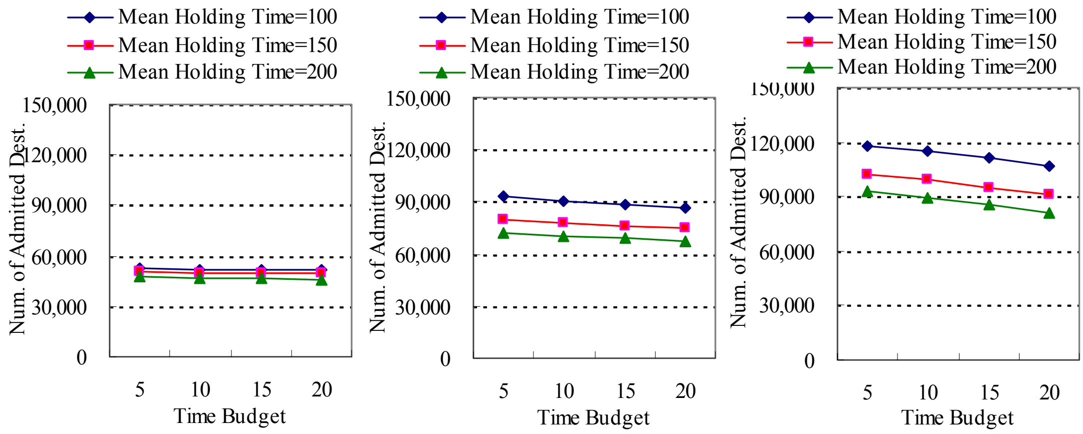
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.10: Effect of the time budget on the AD of a Grid Network (PCAC-M)



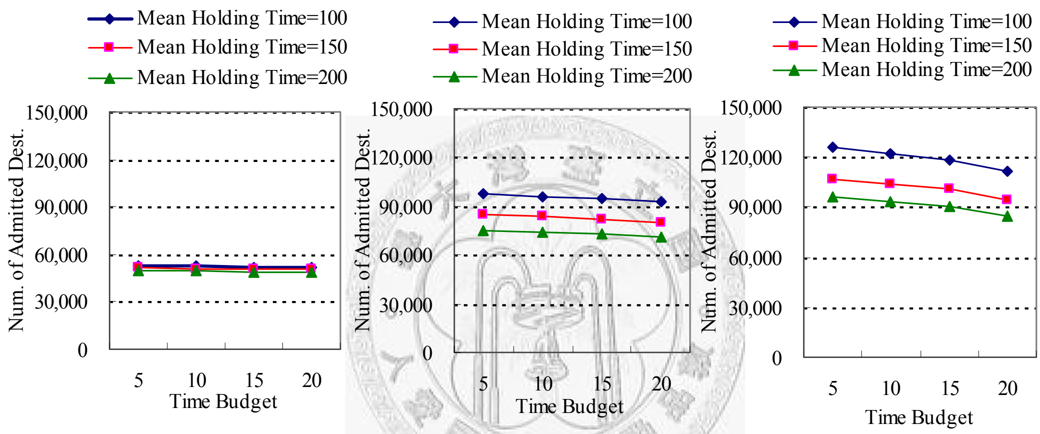
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.11: Effect of the time budget on the AD of a Cellular Network (PCAC-M)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.12: Effect of the time budget on the AD of a Random Network (PCAC-M)

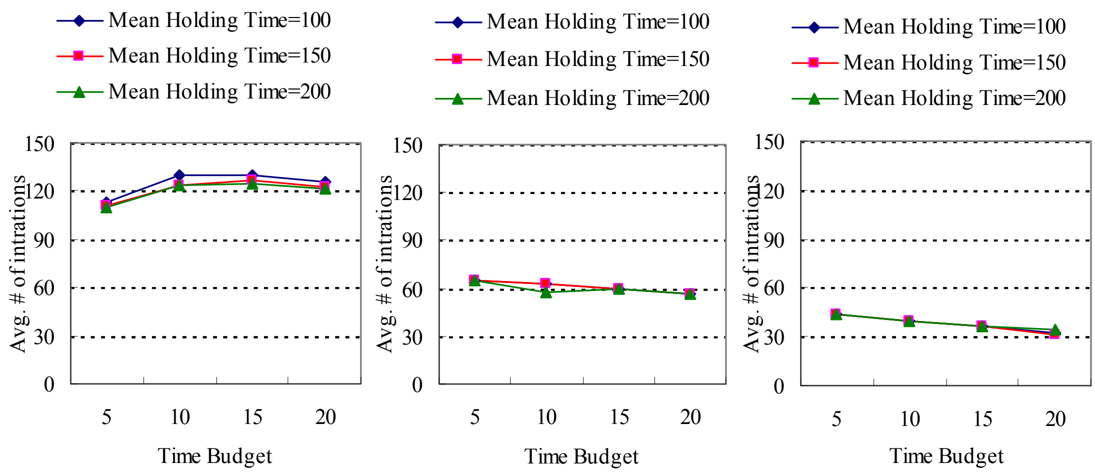


(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.13: Effect of the time budget on the AD of a Scale-free Network (PCAC-M)

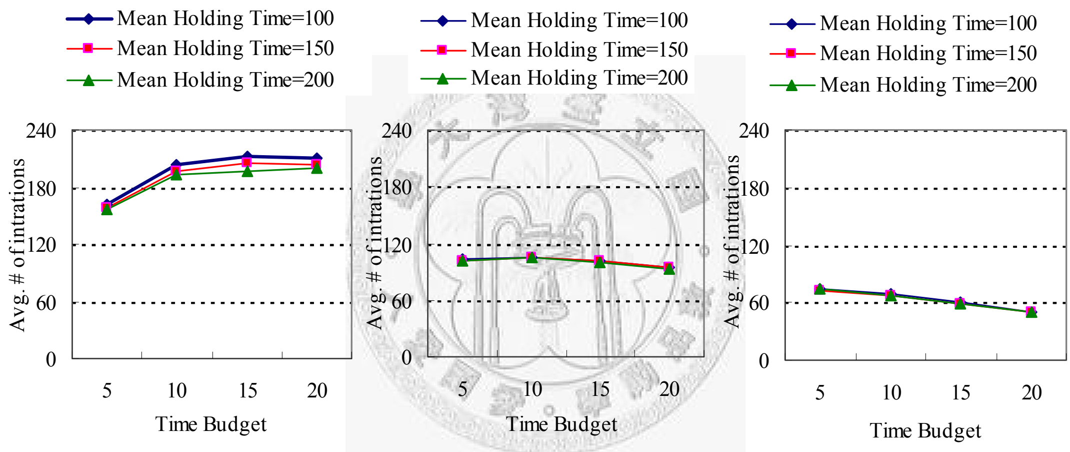
3) Average number of iterations within a time slot (NI)

With the increase in the time budget, the LR-based algorithm can perform more iteration within a time slot to improve the solution quality. However, the problem size and problem complexity also increase with the larger time budget. For the lightly loaded cases in Figures 6.14(a), 6.15(a), and 6.17(a), NI is an increasing function with respect to the time budget. However, for the heavily loaded cases in Figures 6.14(c), 6.15(c), 6.16(c) and 6.17(c), NI is a monotonically decreasing function with respect to the time budget. In Figure 6.16(c), the number of iterations is 37 with η=5, but when time budget increases fourfold (η=20), the number of iterations becomes 25.



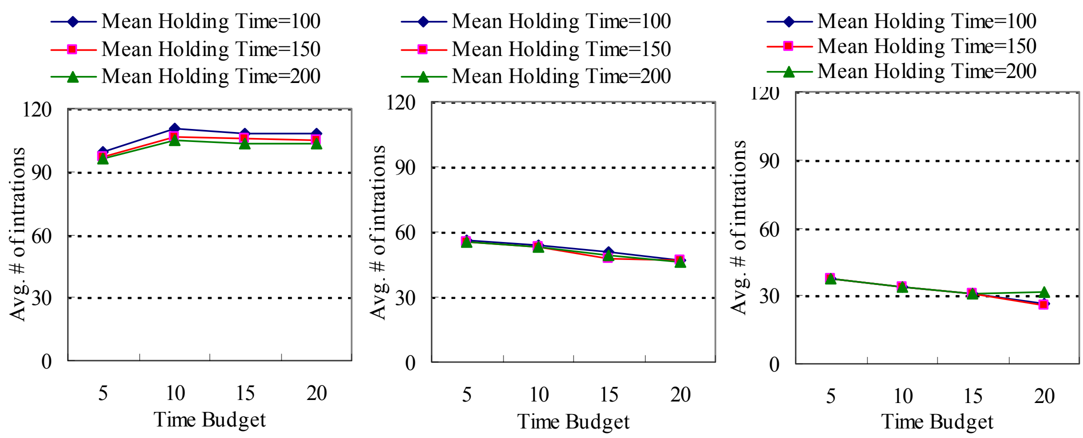
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.14: Effect of the time budget on the NI of a Grid Network (PCAC-M)



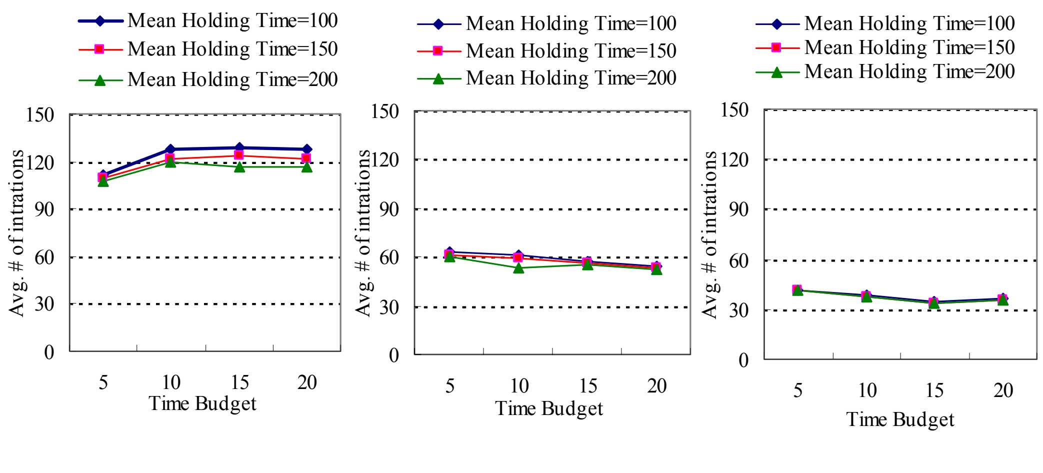
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.15: Effect of the time budget on the NI of a Cellular Network (PCAC-M)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.16: Effect of the time budget on the NI of a Random Network (PCAC-M)

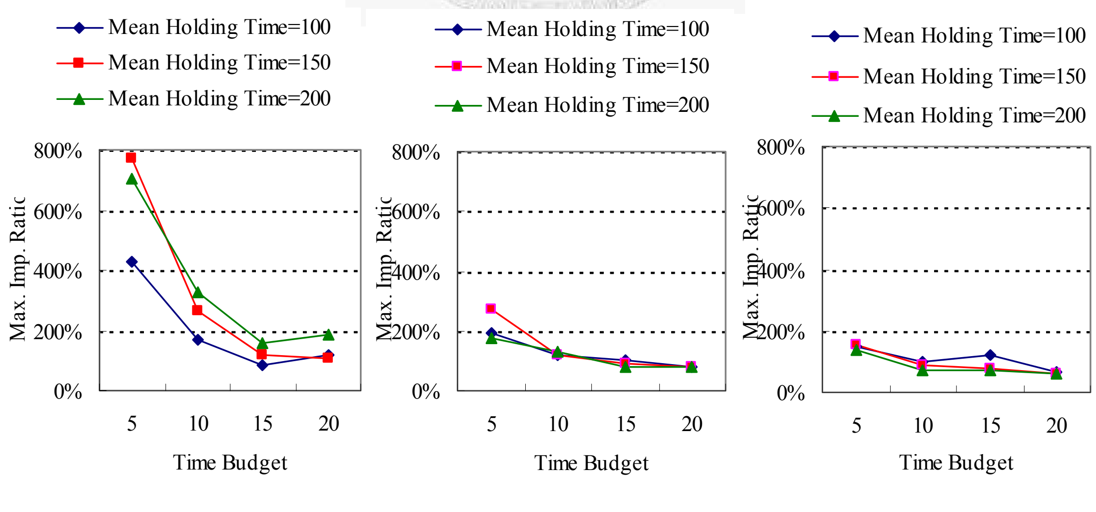


(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.17: Effect of the time budget on the NI of a Scale-free Network (PCAC-M)

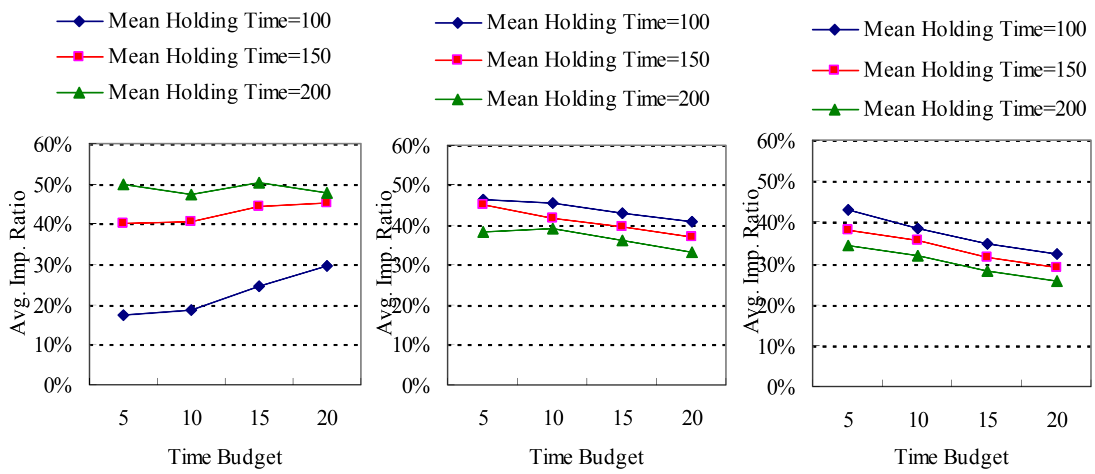
4) **Average and maximum improvement ratio within a time slot (AIR and MIR)**:

For each test network, within a time slot, the maximum improvement of the Lagrangean based heuristic compared to the simple heuristic is 770.83 %, 666.67%, 905.87%, and 625.34%, respectively. The average improvement ratio between the simple heuristic and the Lagrangean based heuristic within a time slot for different network topologies is shown in Figures 6.19, 6.21, 6.23, and 6.24, respectively. Although the time budget is only 5 to 20 seconds, the LR-based algorithm outperforms the simple algorithm.



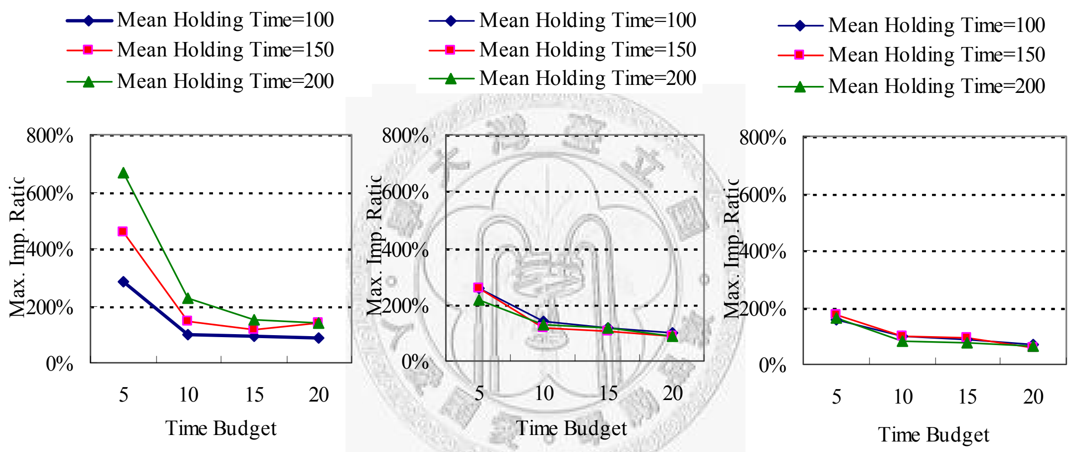
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.18: Effect of the time budget on the MIR of a Grid Network (PCAC-M)



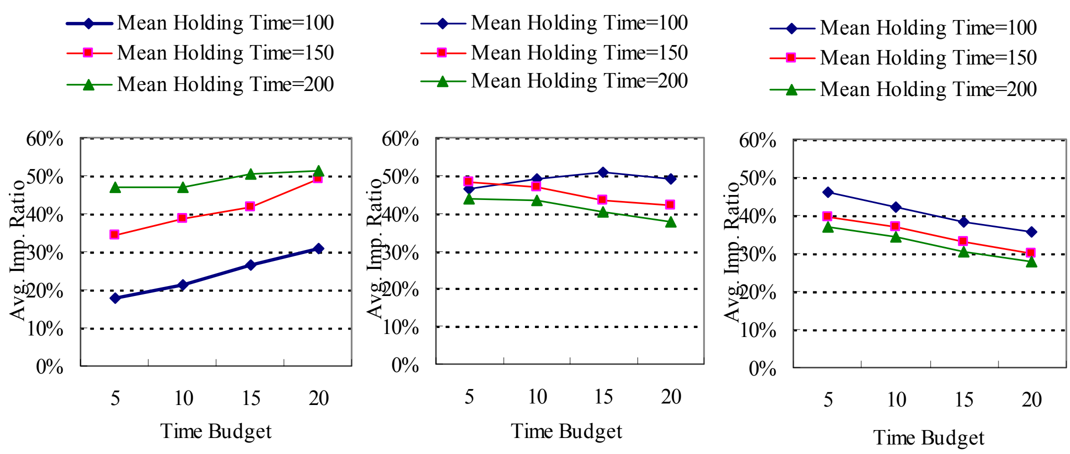
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.19: Effect of the time budget on the AIR of a Gird Network (PCAC-M)



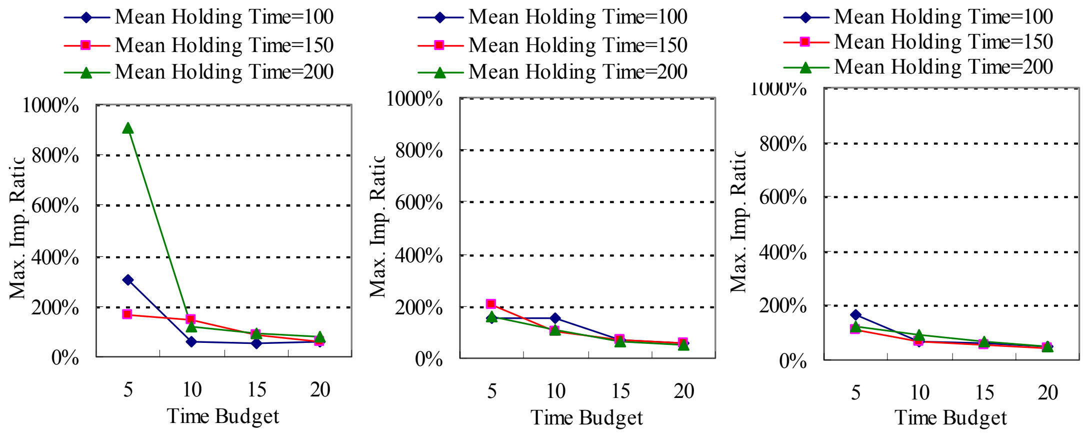
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.20: Effect of the time budget on the MIR of a Cellular Network (PCAC-M)



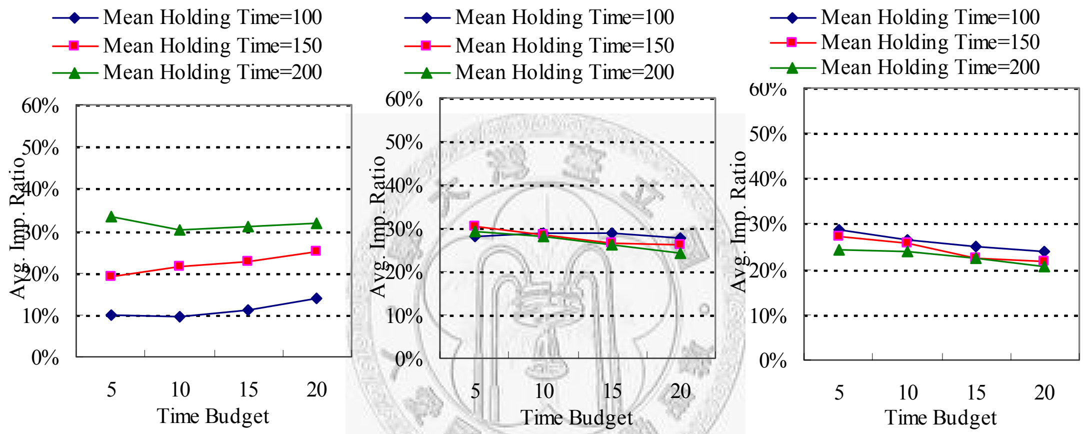
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.21: Effect of the time budget on the AIR of a Cellular Network (PCAC-M)



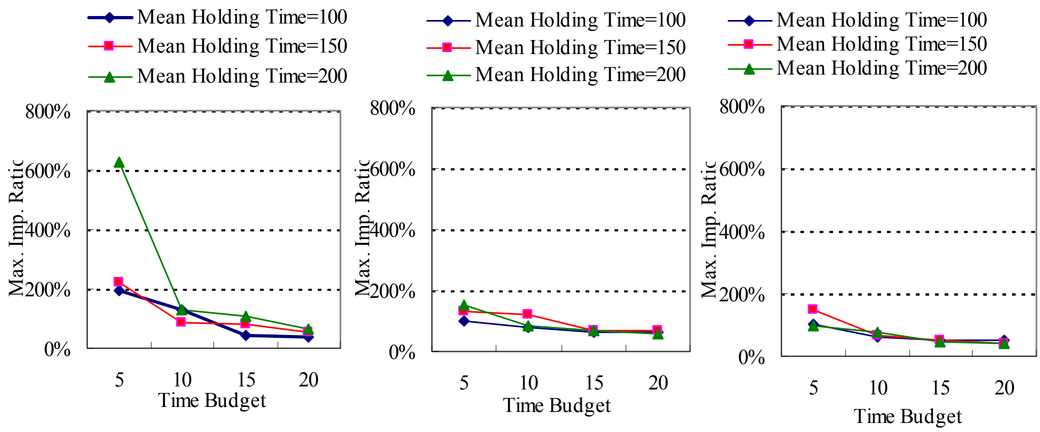
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.22: Effect of the time budget on the MIR of a Random Network (PCAC-M)



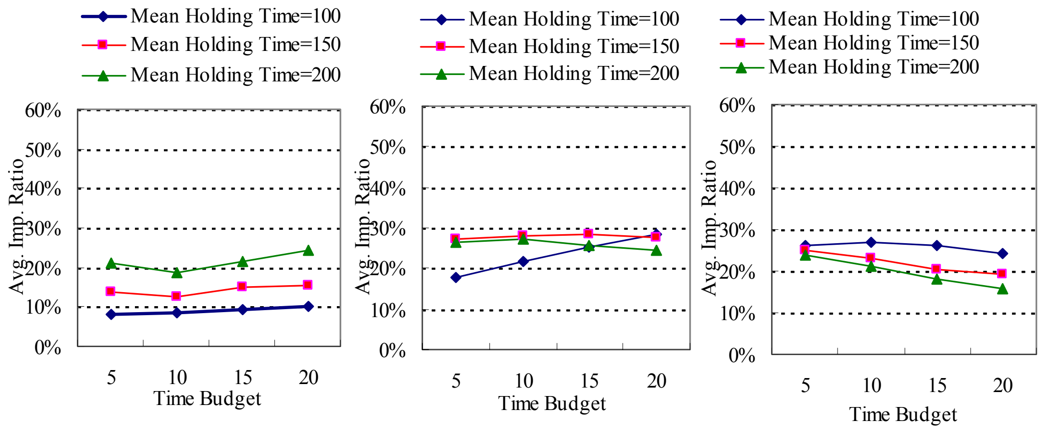
(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.23: Effect of the time budget on the AIR of a Random Network (PCAC-M)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.24: Effect of the time budget on the MIR of a Scale-free Network (PCAC-M)



(a) λ=5 (b) λ=10 (c) λ=15

Figure 6.25: Effect of the time budget on the AIR of a Scale-free Network (PCAC-M)

5) **Average Error Gap (AEG):**

Table 6.3 illustrates the optimality of the problem solution obtained by LR-based algorithm. The gap is relatively large compared to the gap shown in Table 6.2. The average gap is between 51.93% and 98.28%, because the execution time of the algorithm is very short, and subgradient method does not converge quickly. For the lightly loaded cases in the gird network with λ=5, because the LR-based algorithm can perform more iterations, the gap is smaller than heavily loaded cases.

Table 6.3: Average error gap for solving the PCAC-S problem

Grid Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	56.65%	67.46%	75.11%	81.16%	86.63%	89.35%	90.41%	92.88%	94.24%
10	59.67%	68.87%	75.33%	84.40%	88.36%	91.23%	94.14%	95.88%	96.82%
15	62.88%	70.36%	75.99%	87.16%	90.61%	92.53%	95.90%	97.19%	97.85%
20	64.89%	72.42%	77.15%	88.88%	92.20%	93.99%	97.30%	98.13%	98.28%
Cellular Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	58.41%	66.54%	72.90%	77.13%	83.31%	86.85%	85.97%	89.62%	91.53%
10	59.97%	66.73%	72.09%	78.35%	83.36%	86.38%	88.28%	91.27%	92.89%
15	60.82%	66.69%	71.57%	79.95%	88.47%	87.14%	91.19%	93.81%	95.15%

20	61.97%	66.83%	71.37%	81.88%	85.99%	88.70%	93.79%	95.72%	96.58%
Random Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	53.73%	60.72%	68.95%	78.11%	84.24%	87.39%	89.01%	91.89%	93.45%
10	55.85%	64.12%	70.60%	82.50%	86.78%	89.25%	92.67%	94.74%	95.87%
15	58.34%	66.87%	72.22%	85.35%	92.28%	91.46%	94.77%	96.32%	97.07%
20	60.74%	68.49%	73.92%	87.74%	91.15%	93.05%	96.49%	97.50%	97.33%
Scale-free Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	51.93%	57.64%	63.72%	73.59%	81.72%	86.20%	87.37%	91.26%	93.08%
10	53.21%	59.63%	65.64%	78.95%	84.75%	88.89%	91.60%	94.47%	95.89%
15	55.32%	61.89%	67.84%	82.15%	87.14%	90.17%	93.99%	96.17%	97.14%
20	57.04%	63.77%	69.26%	84.51%	89.06%	91.87%	94.65%	96.46%	97.36%

A large gap does not necessarily mean that the solution quality is not good. We use the method shown in Figure 5.28 to obtain the modified error gap. In our experiment, we assign Ω=100 seconds. The modified average error gaps of grid networks, cellular networks, random networks, and scale-free networks are 46.17%, 41.86%, 34.95%, 40.64% respectively.

6) **Long-term system accumulated revenue (SAR)**:

The SAR metric compares the accumulated revenue of the LR-based algorithm with that obtained by the SA algorithm from a long-term viewpoint. For lightly and medium loaded cases, such as the cellular network with τ=150 and λ=5, the accumulated revenue of the SA is much better than that of LR-based algorithm. As shown in Table 6.4, the improvement ratio of the LR-based algorithm over the SA can reach 33.16%. Even in some cases, the SA may better than the result of LR-based algorithm. However, the difference between them is very small and can not find out the tendency. According the simulated results, the improvement ratio decreases with the traffic load in the network.

Table 6.4: Comparison of long-term system accumulated revenue (PCAC-S)

Grid Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	18.72%	21.28%	14.69%	15.18%	10.25%	8.31%	11.42%	8.22%	7.66%
10	21.67%	22.38%	15.64%	17.85%	12.44%	10.95%	14.30%	10.80%	8.25%
15	23.78%	23.43%	16.37%	18.82%	14.25%	11.98%	15.72%	12.28%	9.53%
20	26.40%	24.45%	17.70%	19.41%	15.49%	12.96%	16.29%	13.36%	11.94%
Cellular Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	18.80%	22.41%	19.37%	24.31%	9.20%	4.37%	10.65%	4.96%	3.17%
10	21.87%	26.88%	23.03%	27.24%	12.87%	8.21%	15.28%	8.69%	7.60%
15	25.01%	30.98%	26.05%	28.45%	14.83%	11.45%	16.36%	11.67%	8.88%
20	28.21%	33.16%	26.64%	28.23%	17.38%	12.71%	16.90%	12.22%	10.92%
Random Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	10.18%	14.74%	12.84%	12.96%	2.10%	0.64%	3.81%	2.51%	1.64%
10	12.61%	16.55%	13.64%	14.48%	5.49%	4.60%	8.26%	6.94%	5.27%
15	14.26%	17.59%	15.79%	15.43%	7.93%	7.90%	10.59%	9.05%	7.72%
20	15.96%	19.61%	16.53%	15.69%	9.97%	9.37%	11.62%	10.36%	9.35%
Scale-free Networks									
	λ=5			λ=10			λ=15		
η	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200	τ=100	τ=150	τ=200
5	7.23%	11.07%	12.71%	13.64%	1.86%	2.46%	2.16%	2.93%	-1.85%
10	9.04%	13.05%	14.87%	15.76%	4.78%	0.10%	4.85%	-0.48%	2.32%
15	10.31%	14.44%	16.49%	16.55%	6.11%	2.49%	6.16%	1.00%	-1.00%
20	11.79%	16.14%	17.94%	16.76%	6.93%	3.62%	6.88%	2.56%	0.27%

6.6 Concluding Remarks

In this chapter, we have attempted to solve the problem of capacitated max-revenue multicast routing and partial admission control for multirate multimedia distribution. Our contribution in this paper is shown by the mathematical formulation and the experiment's performance. From the formulation, we have proposed a precise mathematical expression to model the problem efficiently. With regard to performance, the proposed Lagrangean-based heuristic outperforms the simple heuristic. By adding delay constraints, our model can be extended to deal with the QoS constrained multicast routing and admission control problems. These issues will be addressed in future works.

CHAPTER 7 MINIMUM-COST MULTICAST ROUTING PROBLEM WITH THE CONSIDERATION OF DYNAMIC USER MEMBERSHIP

7.1 Overview

From the multicast protocols surveyed, we observe that greatest complexity of these protocols comes from dealing with group membership changes, that is, nodes joining and leaving. The motivation of this paper is to create a mechanism for finding and evaluating the cost-efficiency of a multicast tree with a given network and a fixed set of group members. Also, the behavior of group members is dynamic in that individual members might shut-off for a while, and turn on again later. The probability of this could be determined by observing user behavior over a certain period of time.

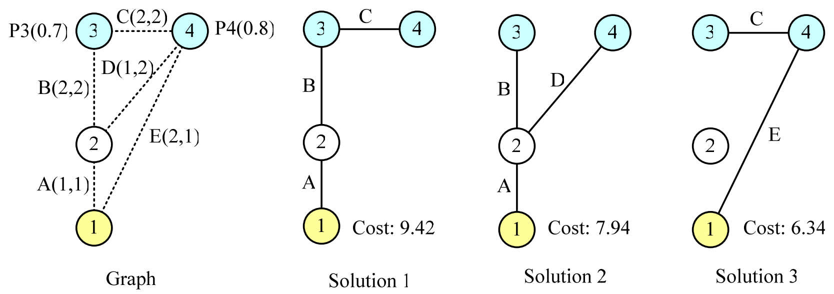


Figure 7.1: Example network (MCRD)

Consider the network in Figure 7.1, where node 1 is the source and nodes 3 and 4 are the destinations with active probabilities of 0.7 and 0.8 respectively. The connection setup costs and transmission costs of the links are shown in parentheses beside each link. Figure 7.1 shows three possible solutions for constructing a multicast tree. In Solution 1, because nodes 3 and 4 have active probabilities of 0.7 and 0.8 respectively, the probability that links A and B have no traffic is 0.06 and the probability that link C has traffic is 0.8. Consequently, the total cost of solution 1 is 9.42, and the cost of Solutions 2 and 3 is 7.94 and 6.34 respectively. Details of the results are shown in Table 7.1.

Table 7.1: Total cost of the example network

	Link A	Link B	Link C	Link D	Link E	Total
S 1	1+1×(1-(0.3×0.2))	2+2×(1-(0.3×0.2))	2+2×(1-0.2)	x	x	9.42
S 2	1+1×(1-(0.3×0.2))	2+2×(1-0.3)	x	1+2×(1-0.2)	x	7.94
S 3	x	x	2+2×(1-0.3)	x	2+1×(1-(0.3×0.2))	6.34

In this chapter, however, we do not deal with the complexity of nodes joining and leaving in our heuristic. Instead, we summarize the activity of a node as a probability. Therefore, the model proposed here is intended for analytical and planning purposes. Even so, as the problem of multicasting has a strong connection with the Steiner tree problem, which is NP-complete, Lagrangean relaxation is applied to achieve an accurate approximation with significantly reduced computation time.

The rest of this chapter is organized as follows. In Section 7.2, we formally define the problem being studied, and also propose a mathematical formulation of min-cost optimization. Section 7.3 applies Lagrangean relaxation as a solution approach to the problem. Section 7.4, describes the computational experiments. Finally, in Section 7.5, we present our conclusions and indicate the direction of future research.

7.2 Problem Formulation

7.2.1 Problem Description

For a network service provider, we consider the problem of constructing a multicast spanning tree that sends traffic to receivers (destinations), while minimizing the total cost of the tree at the same time. The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group is an application requesting transmission over the network, which has one source and one or more destinations. Given the network topology and bandwidth requirement of every destination, we want to determine the routing assignment (a tree for multicasting or a path for unicasting) of a user group.

By formulating the problem as a mathematical programming problem, we solve it optimally by obtaining a network that enables us to achieve our goal, i.e. one that ensures the network operator incurs the minimum cost when constructing and servicing a multicast tree. The notations used to model the problem are listed in Table 7.2.

Table 7.2: Description of notations (MCRD)

Given Parameters	
Notation	**Description**
D	The set of all destinations of a multicast group
r	The source of a multicast group
N	The set of all nodes in the network
L	The set of all links in the network
I_i	The set of all incoming links to node i
q_d	The probability that the destination d is active
a_l	The transmission cost associated with link l

b_l	The connection maintenance cost associated with link l
P_d	The set of all elementary paths from r to $d \in D$
δ_{pl}	The indicator function, which is 1 if link ℓ is on path p
Decision Variables	
Notation	**Description**
y_l	1 if link l is included in the multicast tree, and 0 otherwise
x_p	1 if path p is included in the multicast tree, and 0 otherwise
g_l	The fraction of time that the link l is active on the multicast tree
f_{dl}	1 if link l is used by destination $d \in D$ and 0 otherwise

Each destination $d \in D$ has a given probability, Q_d, whcih indicates the fraction of time that the destination is active, and thus the traffic is to be routed to that node. The probability may be determined by observing user behavior over a period of time. The costs associated with a link are: 1) the fixed cost of connection setup, and 2) the transmission cost proportional to link utilization. At the determination of the multicast tree, utilizations for all links may be computed and used to estimate the total cost.

7.2.2 Mathematical Formulation

According to the problem description in the previous section, the min-cost problem is formulated as a combinatorial optimization problem in which the objective is to minimize the total cost associated with the multicast tree, including the accumulated transmission costs (pay per time unit) and setup cost (pay per connection) on each link used.

Objective function (IP):

$$Z_{IP} = \min \sum_{l \in L} (b_l y_l + a_l g_l) \qquad \text{(LP 7)}$$

subject to:

$$g_l \geq 1 - \prod_{d \in D} (1 - q_d f_{dl}) \qquad \forall l \in L \qquad (7.1)$$

$$\sum_{l \in I_i} y_l \leq 1 \qquad \forall i \in N - \{r\} \qquad (7.2)$$

$$\sum_{l \in I_r} y_l = 0 \qquad (7.3)$$

$$\sum_{p \in p_d} \delta_{pl} x_p \le f_{dl} \qquad \forall l \in L, \forall d \in D \qquad (7.4)$$

$$\sum_{p \in P_d} x_p = 1 \qquad \forall d \in D \qquad (7.5)$$

$$f_{dl} \le y_l \qquad \forall l \in L, \forall d \in D \qquad (7.6)$$

$$f_{dl} = 0 \text{ or } 1 \qquad \forall l \in L, \forall d \in D \qquad (7.7)$$

$$y_l = 0 \text{ or } 1 \qquad \forall l \in L \qquad (7.8)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_d, \forall d \in D \qquad (7.9)$$

$$0 \le g_l \le 1 - \prod_{d \in D} (1 - q_d) \qquad \forall l \in L. \qquad (7.10)$$

The objective function of (IP 7) is to minimize the construction cost and total transmission cost of servicing the maximum bandwidth requirement destination through a specific link for the multicast group.

Constraint (7.1) is referred to as the utilization constraint, which defines the link utilization as a function of q_d and f_{dl}. Since the objective function is strictly an increasing function with g_l, and (LP 7) is a minimization problem, each g_l will equal the aggregate flow in an optimal solution. Constraints (7.2) and (7.3) are both tree constraints. Constraint (7.2) requires that the number of selected incoming links, y_l, to a node is less than 1, while constraint (7.3) requires that there are no selected incoming links, y_l, to the node that is the root of a multicast group. Constraints (7.4) and (7.5) require that only one path is selected for each multicast source-destination pair. Constraint (7.6) requires that if link l is not included in the multicast tree, then it will not be used by any destination.

7.3 Solution Approach

7.3.1 Lagrangean Relaxation

Initially, Lagrangean relaxation was used in both scheduling and general integer programming problems. However, it has become one of the best tools for dealing with optimization problems, such as integer programming, linear programming combinatorial optimization, and non-linear programming.

The Lagrangean relaxation method permits us to remove constraints and place them in the objective function with associated Lagrangean multipliers instead. The optimal value of the relaxed problem is always a lower bound (for minimization problems) on the objective function value of the problem. By adjusting the multiplier of Lagrangean relaxation, we can obtain the upper and lower bounds of this problem. Although the Lagrangean multiplier problem can be solved in a variety of ways, the subgradient optimization technique is probably the most popular approach.

By using the Lagrangean Relaxation method, we can transform the primal problem (IP 7) into the following Lagrangean Relaxation problem (LR 7), where Constraints (7.1), (7.4) and (7.6) are relaxed.

Optimization problem (LR):

$$Z_{D7}(\alpha,\beta,\theta)=\min\sum_{l\in L}(b_l y_l + a_l g_l)+\sum_{l\in L}\alpha_l\Big(\sum_{d\in D}\log(1-q_d\cdot f_{dl})-\log(1-g_l)\Big)$$
$$+\sum_{l\in L}\sum_{d\in D}\beta_{dl}\Big(\sum_{p\in P_d}\delta_{pl}\cdot x_p - f_{dl}\Big)+\sum_{l\in L}\sum_{d\in D}\theta_{dl}(f_{dl}-y_l) \qquad \text{(LR 7)}$$

subject to: (2) (3) (5) (7) (8) (9) and (10).

Where α_l, β_{dl}, and θ_{dl} are Lagrangean multipliers and β_{dl}, $\theta_{dl} \geq 0$. To solve (LR 7), we can decompose it into the following four independent and easily solvable optimization subproblems.

Subproblem 1: (related to decision variable x_p)

$$Z_{Sub7.1}(\beta)=\min\sum_{d\in D}\sum_{p\in P}\Big(\sum_{\ell\in L}\beta_{d\ell}\cdot\delta_{p\ell}\Big)\cdot x_p \qquad \text{(SUB 7.1)}$$

subject to:

$$\sum_{p\in P_d} x_p = 1 \qquad \forall d\in D \qquad (7.5)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_d, \forall d\in D. \qquad (7.9)$$

Subproblem (SUB 7.1) can be further decomposed into |D| independent shortest path problems with nonnegative arc weights β_{dl}. Each shortest path problem can be easily solved by Dijkstra's algorithm.

Subproblem 2: (related to decision variable y_l)

$$Z_{Sub7.2}(\theta) = \min \sum_{l \in L} (b_l - \sum_{d \in D} \theta_{dl}) \cdot y_l \qquad \text{(SUB 7.2)}$$

subject to: (2) (3) (8).

$$\sum_{l \in I_i} y_l \le 1 \qquad \forall i \in N - \{r\} \qquad (7.2)$$

$$\sum_{l \in I_r} y_l = 0 \qquad (7.3)$$

$$y_l = 0 \text{ or } 1 \qquad \forall l \in L. \qquad (7.8)$$

The algorithm to solve Subproblem (SUB 7.2) is:

Step 1 Compute the number of negative coefficients $(b_l - \sum_{d \in D} \theta_{dl})$ for all links.

Step 2 Sort the links in ascending order according to the coefficients.

Step 3 According to the order and complying with constraints (7.1) and (7.2), if the coefficient is less than zero, assign the corresponding negative coefficient of y_l to 1, and 0 otherwise.

Subproblem 3: (related to decision variable g_l)

$$Z_{Sub7.3}(\alpha) = \min \sum_{l \in L} (a_l g_l - \alpha_l \cdot \log(1 - g_l)) \qquad \text{(SUB 7.3)}$$

subject to:

$$0 \le g_l \le 1 - \prod_{d \in D} (1 - q_d) \qquad \forall l \in L. \qquad (7.10)$$

This minimization subproblem can be solved by substitution with its lower and upper bound, because the minimum of this function appears at the endpoints.

Subproblem 4: (related to decision variable f_{dl})

$$Z_{Sub7.4}(\alpha) = \min \sum_{l \in L} \sum_{d \in D} (\alpha_l \log(1 - q_d \cdot f_{dl}) + (\theta_{dl} - \beta_{dl}) f_{dl}) \qquad \text{(SUB 7.4)}$$

subject to:

$$f_{dl} = 0 \text{ or } 1 \qquad \forall l \in L, \forall d \in D. \qquad (7.7)$$

This minimization subproblem can be solved by simply substituting f_{dl} with 0 and 1 and keeping the value that yields the minimum.

According to the weak Lagrangean duality theorem, for any β_{dl}, $\theta_{dl} \geq 0$, $Z_D(\alpha_l, \beta_{dl}, \theta_{dl})$ is a lower bound on Z_{IP}. The following dual problem (D) is then constructed to calculate the tightest lower bound.

Dual Problem (D):

$$Z_{D7} = \max Z_{\mathrm{D}}(\alpha_l, \beta_{dl}, \theta_{dl}) \qquad (D7)$$

subject to:

$$\beta_{dl}, \theta_{dl} \geq 0.$$

There are several methods for solving the dual problem (D7). The most popular is the subgradient method, which is employed here. Let a vector, s, be a subgradient of $Z_D(\beta_{gdl}, \theta_{gl})$. Then, in iteration k of the subgradient optimization procedure, the multiplier vector is updated by $\omega^{k+1}=\omega^k+t^k s^k$. The step size, t^k, is determined by $t^k=\delta(Z^h_{IP} - Z_D(\omega^k))/||s^k||^2$. Z^h_{IP} is the primal objective function value for a heuristic solution (an upper bound on Z_{IP}), δ is a constant, and $0 < \delta \leq 2$.

7.3.2 Getting Primal Feasible Solutions

After optimally solving the Lagrangean dual problem, we get a set of decision variables. However, this solution would not be a feasible one for the primal problem since some constraints are not satisfied. Thus, minor modification of the decision variables, or the hints of the multipliers, must be considered in order to obtain the primal feasible solution of problem (IP). Generally speaking, the better primal feasible solution is an upper bound (UB) of the problem (IP), while the Lagrangean dual problem solution guarantees the lower bound (LB) of problem (IP). Iteratively, by solving the Lagrangean dual problem and getting the primal feasible solution, we get the LB and the UB, respectively. So, the gap between the UB and the LB, computed by (UB-LB)/LB*100%, illustrates the optimality of the problem solution. The smaller the gap computed, the better the optimality.

To solve the dual problem, a simple algorithm is needed to provide an adequate initial upper bound of the primal problem Z_{IP}. Dijkstra's algorithm is used to generate a minimum cost spanning tree over the given network, using the connection setup cost b_l as the arc weight of link l. The result yielded thereby is feasible and is expected to provide a better quality solution than a random guess. We also compare the result of this simple heuristic with the Lagrangean relaxation-based result in Section 4 to demonstrate the improvement made by our approach.

To calculate the primal feasible solution of the minimum cost tree, the solutions to the Lagrangean Relaxation problems are considered. By solving the dual problem optimally we get a set of decision variables that may be appropriate as inputs to obtain primal heuristics. However, as that solution might not be feasible, it requires some more modifications. The set of g_l obtained by solving (SUB 7.3) may not be a valid solution to problem (IP), because the utilization constraint is relaxed. However, the utilization constraint may be a valid solution for some links. Also, the set of f_{dl} obtained by solving (SUB 7.4) may not be a valid solution, because the path and link constraints are relaxed and the union of y_l may not be a tree.

Here we propose a heuristic to obtain a primal feasible solution. While solving the Lagrangean relaxation dual problem, we may find some multipliers related to each link, which could make our routing more efficient. We describe the Lagrangean based heuristic below.

[Lagrangean multiplier based heuristic]

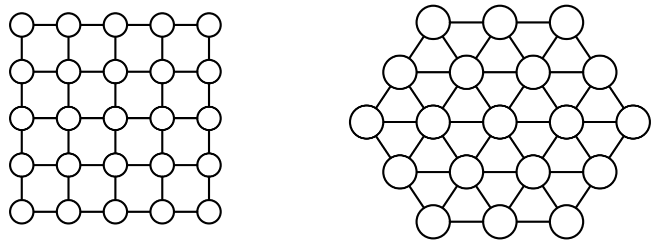
Step 1 Calculate $\sum_{d \in D} \beta_{dl}$ as link l's arc weight.

Step 2 Use the arc weight obtained in Step 1 and run the Dijkstra algorithm.

7.4 Computational Experiments

In this section, computational experiments on the Lagrangean relaxation-based heuristic and simple primal heuristics are reported. The heuristics are tested on three kinds of network: regular networks, random networks, and scale-free networks.

We test two regular networks, shown in Figure 7.2, in our experiment. The first one is a grid network that contains 25 nodes and 40 links, and the second is a cellular network containing 19 nodes and 42 links. Random networks tested in this experiment are generated randomly, each having 25 nodes. The candidate links between all node pairs are given a probability that follows the uniform distribution. In the experiments, we link the node pair with a probability smaller than 2%. If the generated network is not a connected network, we generate a new network.



(a) Grid Network (b) Cellular Network

Figure 7.2: Regular networks (MCRD)

Table 7.3: Parameters for Lagrangean relaxation (MCRD)

Number of Iterations	1,000
Initial Multipliers	0
Improvement Counter	15
Delta Factor	2
Optimal Condition	Gap < 0.001

For each test network, several distinct cases with different pre-determined parameters, such as the number of nodes, are considered. The traffic demand for a multicast group is drawn from a random variable. The link's connection, maintenance, and transmission costs are randomly generated between 1 and 5 and the active probability of each destination is randomly generated between 0.1 and 1. The parameters used for all cases are listed in Table 7.3. The cost of the multicast tree is decided by multiplying the link transmission cost and the bandwidth requirement of the multicast group, plus the link's maintenance costs. We conducted 200 experiments for each kind of network. For each experiment, the result was determined by the group destinations and link costs generated randomly. Table 7.4 summaries selected results of the computational experiments.

For each test network, the maximum improvement ratio of the Lagrangean based heuristic over the simple heuristic is 20.17%, 20.77 %, 37.69%, and 26.85%, respectively. In general, the Lagrangean-based heuristic performs better than the simple heuristic. There are two reasons for this. First, it makes use of related Lagrangean multipliers, including the potential cost for routing on each link in the topology. Second, it is iteration-based and is guaranteed to improve the solution quality iteration by iteration. Therefore, in a more complicated test environment, the improvement ratio is higher. To summarize, by relaxing constraints in the primal problem and optimally solving the dual problem, the set of LR multipliers revealed iteration by iteration becomes a unique source for improving our solutions.

To claim optimality, we also depict the percentile of gap in Table 7.4. The results show that most cases have a gap of less than 20%. We also find that the simple heuristic performs well in many cases, such as case A of the Cellular network and case A of the Random network.

Table 7.4: Selected results of computational experiments (MCRD)

CASE	Dest. #	SA	UB	LB	GAP	Imp.
Grid Network					Max Imp. Ratio: 20.17 %	
A	5	27.34	26.05	25.99	0.22%	4.73%
B	5	40.10	32.01	31.54	1.49%	20.17%
C	10	66.40	54.78	53.83	1.76%	17.51%
D	10	72.24	66.75	63.83	4.57%	7.60%
E	15	53.42	48.01	47.04	2.06%	10.13%
F	15	103.34	98.02	92.56	5.90%	5.15%
G	20	164.34	145.43	144.61	0.57%	11.50%
H	20	156.61	132.82	113.68	16.84%	15.19%
Cellular Network					Max Imp. Ratio: 20.77 %	
A	5	16.62	16.62	16.62	0.00%	0.00%
B	5	32.16	25.48	23.34	9.17%	20.77%
C	10	56.37	46.98	45.30	3.71%	16.66%
D	10	48.88	40.87	38.33	6.63%	16.39%
E	15	58.12	47.31	38.09	24.20%	18.60%
F	15	85.76	82.30	74.45	10.54%	4.03%
G	18	124.35	118.83	111.34	6.73%	4.44%
H	18	143.33	128.98	124.43	3.66%	10.01%
Random Networks					Max Imp. Ratio: 37.69 %	
A	5	7.75	7.75	7.75	0.00%	0.00%
B	5	14.32	12.94	12.48	3.69%	9.64%
C	10	53.83	42.47	39.89	6.47%	21.10%
D	10	59.16	36.86	33.04	11.56%	37.69%
E	15	60.38	57.82	53.37	8.34%	4.24%
F	15	76.42	68.88	62.34	10.49%	9.87%
G	20	93.46	74.83	73.08	2.39%	19.93%

H	20	103.46	92.64	83.78	10.58%	10.46%
Random Networks					Max Imp. Ratio: 26.85 %	
A	5	19.24	16.81	16.79	0.12%	14.48%
B	5	28.86	23.48	22.45	4.56%	22.93%
C	10	53.67	48.08	46.34	3.75%	11.63%
D	10	61.09	48.16	45.07	6.86%	26.85%
E	15	57.31	51.05	46.17	10.57%	12.26%
F	15	88.51	83.07	76.45	8.65%	6.55%
G	18	127.38	113.03	109.68	3.06%	12.70%
H	18	134.47	118.15	107.30	10.11%	13.81%

SA: The result of the simple heuristic
UB and LB: Upper and lower bounds of the Lagrangean-based modified heuristic
GAP: The error gap of Lagrangean relaxation
Imp.: The improvement ratio of the Lagrangean-based heuristic

The contribution of this research is quite academic. With the innovative idea of constructing a multicast tree that can adapt to the activity of end users in a minimization problem, the model itself can be aware of the phenomenon of dynamic user joining and leaving without all the fuss of dealing with it in our heuristic. For this reason, our model is ideal for network planning purposes. However, the computational results do show that the structure of the problem is suitable for the methodology of Lagrangean relaxation. Although the model is still in a simple form, interested researchers may develop several extensions to it with ease.

CHAPTER 8 CONCLUSION AND FUTURE WORK

8.1 Summary

In this dissertation, we have proposed several optimization-based heuristics to deal with different categories of the multicasting problem, including network planning and operational problems. First, we have proposed an optimization-based algorithm to solve the minimum cost single-group multicast routing problem. Then, by considering the link capacity constraint, we extended the algorithm to solve the capacitated minimum cost multicast routing problem for a multi-group environment. The experiment results show that our proposed heuristic outperforms algorithms proposed in earlier. An extension of this model could consider the delay constraint for each multicast group, which could be classified as a delay-constrained multicast routing problem.

We have also considered the call admission control mechanism and resource reservation mechanisms jointly, and attempted to solve the problem of maximum-revenue multicast routing with a partial admission control mechanism for both single-rate and multirate multicasting. We proposed the concept of partial admission control for the multicast call admission control. Partial admission control means that the admission policy of a multicast group considers accepting portions destinations for the requested multicast group. Apart from conducting on multicast network planning, we also run real-time simulations to show the performance superiority of LR-based algorithm. An extension of this model should consider the tendency of link usage, which would not only maximize short-term revenue, but also improve long-term system accumulated revenue and user satisfaction.

Finally, we have attempted to solve the problem of min-cost multicast routing by considering dynamic user membership. We have proposed a mechanism for finding and evaluating the cost-efficiency of a multicast tree with a given network and a fixed set of group members by considering the behavior pattern of users. The behavior of group members is dynamic in that individual members might shut-off for a while, and turn on again later. The probability of this could be determined by observing user behavior over a certain period of time. The proposed Lagrangean relaxation and subgradient based algorithms outperform the primal heuristics.

8.2 Future Work

Even though we have dealt with a series of routing problems for multimedia networks in an integrated and comprehensive manner, there are still many open issues to be further investigated. We point out five challenging issues to be tackled in the future. We also proposed some feasible mathematical models to formulate these problems. These models are based on the research results of the dissertation.

8.2.1 Min-Cost Multi-tree Multirate Multicast Problem

In Chapter 3 and 4, we discuss the minimum cost routing problem for multirate multimedia multicasting. We assume that all the layers go on one tree. There is another method, named multi-tree method, to deliver multirate multimedia streams over the network. For multi-tree multirate multicasting, an encoder encodes video data into more than one video stream, including one base layer stream and several enhancement layer streams. Each layer is transmitted by an independent multicast tree. When a user request the video, it have to receive the base layer contains the most important portions of the video stream needed to achieve the minimum quality level and several enhancement layers contain the other portions of the video stream for refining the quality of the base layer stream from multiple multicast trees.

The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group is an application requesting transmission in the network, which has one source and one or more destinations. Given the network topology and bandwidth requirement of every destination of a user group, we want to determine the routing assignment for multicasting. The notations used to model the problem are listed in Table 8.1.

Table 8.1: Description of notations (FW1)

Given Parameters	
Notation	**Description**
G	The set of all multicast groups
V	The set of nodes in the network
L	The set of links in the network
α_{gd}	Layers requirement of destination d of multicast group g
E_g	The set of layers of multicast group g
D_g	The set of destinations of multicast group g
a_l	Transmission cost associated with link l
m_{ge}	Traffic requirement of layer e of multicast group g
h_g	The minimum number of hops to the farthest destination node in multicast group g
C_l	The capacity of link l
I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of paths destination d of multicast group g may use
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise
Decision Variables	
Notation	**Descriptions**
x_{gepd}	1 if path p is selected for group g destined for destination d with requirement of layer e and 0 otherwise
y_{gel}	1 if link l is on the sub-tree adopted by layer e of multicast

	group g and 0 otherwise

According to the above problem description, the min-cost problem is formulated as a combinatorial optimization problem in which the objective function is to minimize the total link cost of the multicast trees.

Objective function:

$$Z_{IP8-1} = \min \sum_{g \in G} \sum_{e \in E_g} \sum_{l \in L} a_l y_{gel} m_{ge} \qquad \text{(IP 8-1)}$$

subject to:

$$\sum_{g \in G} \sum_{e \in E_g} \sum_{p \in P_{gd}} \sum_{d \in D_g} x_{gepd} \delta_{pl} m_{ge} \le C_l \qquad l \in L \qquad (8.1)$$

$$\sum_{p \in P_{gd}} x_{gepd} \le 1 \qquad \forall g \in G, d \in D_g, e \in E_g$$

$$\sum_{1 \le e \le \alpha_{gd}} \sum_{p \in P_{gd}} x_{gepd} = \alpha_{gd} \qquad \forall g \in G, d \in D_g \qquad (8.3)$$

$$\sum_{e > \alpha_{gd}} \sum_{p \in P_{gd}} x_{gepd} = 0 \qquad \forall g \in G, d \in D_g \qquad (8.4)$$

$$x_{gepd} = 0 \text{ or } 1 \qquad \forall g \in G, e \in E_g, p \in P_{gd}, d \in D_g \qquad (8.5)$$

$$y_{gel} = 0 \text{ or } 1 \qquad \forall l \in L, g \in G, e \in E_g \qquad (8.6)$$

$$\sum_{l \in L} y_{gel} \ge \max\{h_g, |D_g|\} \qquad \forall g \in G, e \in E_g \qquad (8.7)$$

$$\sum_{d \in D_g} \sum_{p \in P_{gd}} x_{gepd} \delta_{pl} \le |D_g| y_{gel} \qquad \forall g \in G, l \in L, e \in E_g \qquad (8.8)$$

$$\sum_{l \in I_v} y_{gel} \le 1 \qquad \forall g \in G, v \in V - \{r_g\}, e \in E_g \qquad (8.9)$$

$$\sum_{l \in I_{r_g}} y_{gel} = 0 \qquad \forall g \in G, e \in E_g. \qquad (8.10)$$

The objective function of (IP 8-1) is to minimize the total transmission cost of servicing all multicast groups G, where G is the set of user groups requesting connection. Each group has m_{ge} multicast tree. The bandwidth requirement on a link for a specific group is calculated by summing over the bandwidth requirement for all layers used the link.

Constraint (8.1) is referred to as the capacity constraint, where the total flow on the link can not over the link capacity. Constraints (8.3) and (8.4) require that the destinations should receive the streams they needed and there is at least one path selected for each layer. Constraints (8.5) and (8.6) are integral constraint for decision variables. Constraints (8.2) and (8.5) require that only one path be selected for each multicast layer. Constraints (8.7) and (8.8) require that the number of links on the multicast tree adopted by layer e of multicast group g be at least the maximum of h_g and the cardinality of D_g. The h_g and the cardinality of D_g are the legitimate lower bounds of the number of links on the multicast tree adopted by the multicast group g. Constraint (8.7) is called the tree constraint, which requires that the union of the selected paths for the destinations of user group g forms a tree. Constraints (8.9) and (8.10) are both redundant constraints. Constraint (8.9) requires that the number of selected incoming links, y_{gel}, to a node be 1 or 0, while constraint (8.10) requires that there are no selected incoming links, y_{gel}, to the node that is the root of multicast group g. As a result, the links we select can form a tree.

8.2.2 Max-Revenue Multi-tree Multirate Multicast Problem

In multi-tree model, each destination receives sufficient video data from several layers to reach the quality requirement, including the base layer and the enhancement layers. We define the user satisfaction level as a satisfaction function U_{gd} which describe the inclination and possibilities to join in the tree and receive the data. Consider the utility function in Figure 8.2, the highest video stream contains 3 layers including one base layer (layer 1) and two enhancement layers (layer 2 and 3). If the destination can receive all the layers, the satisfaction will be 100%. The satisfaction of a destination will be decreased by the service quality. Because the multirate multimedia is encoded in an incremental method, the destination can not decode the multimedia without the lower layers' information. For example, the satisfaction by receiving layer 1 and layer 3 is the same as by receiving layer 1. this is because the destination can not encode the video without layer 2.

Table 8.2: Example of satisfaction function

Layer received			Satisfaction
Layer 1	Layer 2	Layer 3	%
Y	Y	Y	100
Y	Y	N	80
Y	N	Y	50
Y	N	N	50
N	Y	Y	0
N	Y	N	0
N	N	Y	0
N	N	N	0

The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group is an application requesting transmission in the network, which has one source and one or more destinations. Given the network topology, bandwidth requirement of every destination of a user group and user satisfaction function, we want to jointly determine the routing assignment for multicasting and admission control. The notations used to model the problem are listed in Table 8.3.

Table 8.3: Description of notations (FW2)

Given Parameters	
Notation	**Description**
G	The set of all multicast groups
V	The set of nodes in the network
L	The set of links in the network
E_g	The set of layers of multicast group g
D_g	The set of destinations of multicast group g
α_{gd}	Layers requirement of destination d of multicast group g
f_{gd}	Revenue generated by admitting destination d of group g and servicing all layers it required
U_{gd}	Satisfaction of destination d of group g which is a function of $\overline{K}_{gd}$
m_{ge}	Traffic requirement of layer r of multicast group g
h_g	The minimum number of hops to the farthest destination node in multicast group g
C_l	The capacity of link l
I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g

P_{gd}	The set of paths destination d of multicast group g may use
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise
Decision Variables	
Notation	**Descriptions**
x_{gepd}	1 if path p is selected for group g destined for destination d with requirement of layer e and 0 otherwise
y_{gel}	1 if link l is on the sub-tree adopted by layer e of multicast group g and 0 otherwise
$\overline{K}_{gd}$	A vector $\{k_1, k_2, ..., k_{\alpha_{gd}}\}$ where k_e is 1 if layer e is received by destination d of group g and 0 otherwise

According to the above problem description, the max-revenue problem is formulated as a combinatorial optimization problem in which the objective function is to maximize the total system revenue.

Objective function:

$$Z_{IP8-2} = \max \sum_{g \in G} \sum_{e \in E_g} f_{gd} U_{gd}(\overline{K}_{gd}) \qquad \text{(IP 8-2)}$$

subject to:

$$\sum_{g \in G} \sum_{e \in E_g} \sum_{p \in P_{gd}} \sum_{d \in D_g} x_{gepd} \delta_{pl} m_{ge} \le C_l \qquad l \in L \quad (8.11)$$

$$k_e = \sum_{p \in P_{gd}} x_{gepd} \qquad \forall g \in G, d \in D_g, e \in E_g \quad (8.12)$$

$$\sum_{p \in P_{gd}} x_{gepd} \le 1 \qquad \forall g \in G, d \in D_g, e \in E_g \quad (8.13)$$

$$x_{gepd} = 0 \text{ or } 1 \qquad \forall g \in G, e \in E_g, p \in P_{gd}, d \in D_g \quad (8.14)$$

$$y_{gel} = 0 \text{ or } 1 \qquad \forall l \in L, g \in G, e \in E_g \quad (8.15)$$

$$\sum_{l \in L} y_{gel} \ge \max\{h_g, |D_g|\} \qquad \forall g \in G, e \in E_g \quad (8.16)$$

$$\sum_{d \in D_g} \sum_{p \in P_{gd}} x_{gepd} \delta_{pl} \le |D_g| y_{gel} \qquad \forall g \in G, l \in L, e \in E_g \quad (8.17)$$

$$\sum_{l \in I_v} y_{gel} \le 1 \qquad \forall g \in G, v \in V - \{r_g\}, e \in E_g \quad (8.18)$$

$$\sum_{l \in I_{r_g}} y_{gel} = 0 \qquad \forall g \in G, e \in E_g. \quad (8.19)$$

The objective function of (IP 8-2) is to maximize the total revenue from servicing the admitted destinations in multicast groups G, where G is the set of user groups requesting connection. Each group has m_{ge} multicast trees. The revenue from each admitted destination can be fully characterized by two parameters: the satisfaction of the received video stream and the complete revenue of the destination.

Constraint (8.11) is referred to as the capacity constraint, where the total flow on the link can not over the link capacity. Constraint (8.12) defines the vector in the satisfaction function. Constraints (8.13) and (8.14) require that mo more than one path is selected for each multicast layer. Constraints (8.14) and (8.15) are integral constraint for decision variables. Constraints (8.16) and (8.17) require that the number of links on the multicast tree adopted by layer e of multicast group g be at least the maximum of h_g and the cardinality of D_g. The h_g and the cardinality of D_g are the legitimate lower bounds of the number of links on the multicast tree adopted by the multicast group g. Constraint (8.17) is called the tree constraint, which requires that the union of the selected paths for the destinations of user group g forms a tree. Constraints (8.18) and (8.19) are both redundant constraints. Constraint (8.18) requires that the number of selected incoming links, y_{gel}, to a node be 1 or 0, while Constraint (8.19) requires that there are no selected incoming links, y_{gel}, to the node that is the root of multicast group g. As a result, the links we select can form a tree.

8.2.3 Max-Profit Multirate Multicast Problem

In Chapter 3 and 4, we discuss the min-cost routing problem for multirate multicasting. In Chapter 6, we discuss the max-revenue admission problem for multirate multicasting. From the viewpoint of service provider, we can easily joint these two models to deal with the maximum profit problem.

The network is modeled as a graph, where the switches are depicted as nodes and the links are depicted as arcs. A user group, which has one source and one or more destinations, is an application requesting transmission over the network. Given the network topology, the cost and capacity of links and the bandwidth requirement of every destination of a user group, we want to jointly determine the following decision variables: (1) the routing assignment (a tree for multicasting, or a path for unicasting) of each admitted destination; and (2) the admitted number of destinations of each partially admitted multicast group.

This model is based on the following viable assumptions.

- The revenue from each partially admitted group can be fully characterized by two parameters: the total amount of admitted revenue of the group associated with a specific priority, and the number of admitted destinations of the specific priority.
- The revenue from each partially admitted group associated with a specific priority is a monotonically increasing function with respect to the number of admitted destinations of the specific priority.
- The revenue function from each partially admitted group associated with a specific priority is a convex function with respect to the entire admitted revenue of the group associated with the specific priority and the number of admitted destinations of the specific priority. However, the entire admitted revenue and the number of admitted destinations jointly may not be a concave function.
- The revenue from each partially admitted group associated with a specific priority is independent.

Table 8.4: Description of notations (FW3)

Given Parameters	
Notation	**Descriptions**
F_{gq}	Revenue generated from admitting partial users of multicast group g with propriety q, which is a function of f_{gq} and a_{gq}
a_{gq}	Revenue generated from admitting multicast group *g* with propriety *q*
a_l	Transmission cost associated with link *l*
α_{gd}	Traffic requirement of destination d multicast group *g*
G	The set of multicast groups
V	The set of nodes in the network
L	The set of links in the network
Q	The set of priorities in the network
D_g	The set of destinations of multicast group *g*

T_{gq}	The set of destinations of priority q in multicast group g
C_l	Capacity of link l
I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of elementary paths user d of multicast group g may use
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise
σ_{gd}	The indicator function which is 1 if priority q is selected for destination d and 0 otherwise

Decision Variables

Notation	Descriptions
x_{gpd}	1 if path p is selected for group g destined for destination d and 0 otherwise.
y_{gl}	1 if link l is on the sub-tree adopted by multicast group g and 0 otherwise.
m_{gl}	The maximum traffic requirement of the destination in multicast group g that are connected to the source through link l.
f_{gq}	The number of admitted destinations of priority q in multicast group g.

Optimization Problem:

Objective function:

$$Z_{IP8-3} = \min \quad -\sum_{g\in G}\sum_{q\in Q} F_{gq}(a_{gq}, f_{gq}) + \sum_{g\in G}\sum_{l\in L} a_l m_{gl} \qquad \text{(IP 8-3)}$$

subject to:

$$\sum_{p\in P_{gd}} \alpha_{gd} x_{gpd} \delta_{pl} \le m_{gl} \qquad \forall g\in G, d\in D_g, l\in L \quad (8.20)$$

$$\sum_{g\in G} m_{gl} \le C_l \qquad \forall l\in L \quad (8.21)$$

$$m_{gl} \in [0, \max_{d\in D} \alpha_{gd}] \qquad \forall g\in G, l\in L \quad (8.22)$$

$$\sum_{l\in L} y_{gl} \ge \sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd} \qquad \forall g\in G \quad (8.23)$$

$$\sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} \le |D_g| y_{gl} \qquad \forall g\in G, l\in L \quad (8.24)$$

$$\sum_{l \in I_v} y_{gl} \le 1 \qquad \forall g \in G, v \in V - \{r_g\} \quad (8.25)$$

$$\sum_{l \in I_{r_g}} y_{gl} = 0 \qquad \forall g \in G \quad (8.26)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall g \in G, p \in P_{gd}, d \in D_g \quad (8.27)$$

$$\sum_{d \in D_g} \sum_{p \in p_{gd}} x_{gpd} \sigma_{qd} = f_{gq} \qquad \forall g \in G, q \in Q \quad (8.28)$$

$$f_{gq} \in \{0, 1, 2, \ldots\ldots, |T_{gq}|\} \qquad \forall g \in G, q \in Q \quad (8.29)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l \in L, g \in G \quad (8.30)$$

$$\sum_{p \in P_{gd}} x_{gpd} \le 1 \qquad \forall g \in G, d \in D_g . \quad (8.31)$$

The objective function of (IP 8-3) is to maximize the total profit. The profit is calculated by subtract the cost from the revenue by servicing the partially admitted destinations in multicast group g associated with a specific priority, where $g \in G$, $q \in Q$, and G is the set of user groups requesting transmission.

Constraints (8.20) and (8.21) are the capacity constraints. In this model, the variable m_{gl} can be viewed as the estimate of the aggregate flows. Constraint (8.22) is a redundant constraint, which provides the upper and lower bounds of the maximum traffic requirement for multicast group g on link l. Constraint (8.23) requires that if one path is selected for group g destined for destination d, it must also be on the sub-tree adopted by multicast group g. Constraint (8.24) is the tree constraint, which requires that the union of the selected paths for the destinations of user group g forms a tree. Constraints (8.25) and (8.27) require that the number of selected incoming links, y_{gl}, is 1 or 0 and each node, except the root, has only one incoming link. Constraint (8.25) requires that the number of selected incoming links, y_{gl}, to node is 1 or 0. Constraint (8.26) requires that no selected incoming link, y_{gl}, is the root of multicast group g. As a result, the links we select form a tree. Constraints (8.27) and (8.31) require that at most one path is selected for each admitted multicast source-destination pair, while Constraint (8.28) relates the routing decision variables x_{gpd} to the auxiliary variables f_{gq}. Constraint (8.29) requires that the number of admitted destinations in multicast group g with priority q is a set of integers.

8.2.4 Routing Algorithms Considering Multirate Multicast Service and Traffic Rerouting

In Chapter 5 and 6, the admission control algorithms only consider the residual resources of the networks based on the existing routing topology to decide whether to admit new traffic flows or not. Because the residual resources of links may be un-continual, we may not find a feasible solution to fulfill the new incoming users. However, it does not mean that the residual resource is not sufficient to admit new traffic flows. If the network operators reject the new coming traffic flow, the network utilization is not optimized, and the revenue is not maximized. If we try to reroute some traffic from one path to another, we may have a new continuous path to meet the requirements of new traffic flows. Therefore, we can admit more incoming users and the revenue is maximized.

The rerouted traffic flow from one path to another can also introduce interference between traffic flows, which in turns impacts other traffic flows in the network. Such interference could be packet loss increasing of transmission delay. Another issue to be mentioned is the cost of re-routing process.

When a path is rerouted, network operators should temporarily stop to transmit all traffic flows on the path. After a period of time, the source restarts to transmit data along the new routing topology. The stopping time period should be sufficient for draining out all traffic flow already input in the network through the old routing path. This is to ensure the packets are not out-of-order duo to path rerouting. From the sender's point of view, the delay is just equal to the maximum end-to-end delay of the originally path within the tree. On the other hand, from the receiver's point of view, the delay is equal to the end-to-end delay of the new path. With the considering the of rerouting cost discussed above, the longer the end-to-end delay of the tree is, the less likely the path is rerouted.

In this section, we formulate the routing and partial admission control problem for multirate multicasting with considering traffic rerouting. New traffic flows are admitted as many as possible, and after admitting new traffics, try to find a rerouting policy that has a less cost. The objective function is to maximize the total revenue minus the cost of rerouting existing traffic flows.

Table 8.5: Description of notations (FW4)

Notation	Descriptions
Given Parameters	
Notation	**Descriptions**
F_{gq}	Revenue generated from admitting partial users of multicast group g with propriety q, which is a function of f_{gq} and a_{gq}
a_{gq}	Revenue generated from admitting multicast group g with propriety q
α_{gd}	Traffic requirement of destination d multicast group g
G'	The set of existing multicast groups
G''	The set of new incoming groups
G	$\{G' \cup G''\}$
V	The set of nodes in the network
L	The set of links in the network
Q	The set of priorities in the network
D_g	The set of destinations of multicast group g
T_{gq}	The set of destinations of priority q in multicast group g
C_l	Capacity of link l
I_v	The incoming links to node v
r_g	The multicast root of multicast group g
I_{r_g}	The incoming links to node r_g
P_{gd}	The set of elementary paths user d of multicast group g may use
y'_{gl}	1 if link l is used to transmit the traffic for multicast group g on the original routing topology, and 0 otherwise
h_{gd}	The number of hop from existing destination d of group g to the source
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise
σ_{gd}	The indicator function which is 1 if priority q is selected for destination d and 0 otherwise
S_g	Cost of rerouting users of existing user group g
Decision Variables	
Notation	**Descriptions**
x_{gpd}	1 if path p is selected for group g destined for destination d and 0 otherwise.
y_{gl}	1 if link l is on the sub-tree adopted by multicast group g and 0 otherwise.
m_{gl}	The maximum traffic requirement of the destination in multicast group g that are connected to the source through link l.
f_{gq}	The number of admitted destinations of priority q in multicast group g.

z_g	1 if the original routing topology of group $g \in G'$ is changed and *0* otherwise.

The given rerouting cost, S_g, of the group g, can be any value chosen by system operator. $\max_{d \in D_g} \{t_d + 3\sigma_d\}$ is a reasonable upper bound for end-to-end delay, where t_d is the mean end-to-end delay and σ_d is the standard deviation of the delay.

Optimization Problem:

Objective function:

$$Z_{IP8-4} = \min - \sum_{g \in G''} \sum_{q \in Q} F_{gq}(a_{gq}, f_{gq}) + \sum_{g \in G'} z_g S_g \qquad \text{(IP 8-4)}$$

subject to:

$$\sum_{p \in P_{gd}} \alpha_{gd} x_{gpd} \delta_{pl} \le m_{gl} \qquad \forall g \in G, d \in D_g, l \in L \quad (8.32)$$

$$\sum_{g \in G} m_{gl} \le C_l \qquad \forall l \in L \quad (8.33)$$

$$m_{gl} \in [0, \max_{d \in D} \alpha_{gd}] \qquad \forall g \in G, l \in L \quad (8.34)$$

$$\sum_{l \in L} y_{gl} \ge \sum_{d \in D_g} \sum_{p \in P_{gd}} x_{gpd} \qquad \forall g \in G \quad (8.35)$$

$$\sum_{d \in D_g} \sum_{p \in P_{gd}} x_{gpd} \delta_{pl} \le |D_g| y_{gl} \qquad \forall g \in G, l \in L \quad (8.36)$$

$$\sum_{l \in I_v} y_{gl} \le 1 \qquad \forall g \in G, v \in V - \{r_g\} \quad (8.37)$$

$$\sum_{l \in I_{r_g}} y_{gl} = 0 \qquad \forall g \in G \quad (8.38)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall g \in G, p \in P_{gd}, d \in D_g \quad (8.39)$$

$$\sum_{d \in D_g} \sum_{p \in p_{gd}} x_{gpd} \sigma_{qd} = f_{gq} \qquad \forall g \in G'', q \in Q \quad (8.40)$$

$$f_{gq} \in \{0, 1, 2, ..., |T_{gq}|\} \qquad \forall g \in G, q \in Q \quad (8.41)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall l \in L, g \in G \quad (8.42)$$

$$\sum_{p \in P_{gd}} x_{gpd} \le 1 \qquad \forall g \in G'', d \in D_g \quad (8.43)$$

$$\sum_{p \in P_{gd}} x_{gpd} = 1 \qquad \forall g \in G', d \in D_g \quad (8.44)$$

$$\sum_{l \in L} (y_{gl} - y'_{gl})^2 \le |L| z_g \qquad \forall g \in G' \quad (8.45)$$

$$z_g \le \sum_{l \in L} (y_{gl} - y'_{gl})^2 \qquad \forall g \in G' \quad (8.46)$$

$$z_g = 0 \text{ or } 1 \qquad \forall g \in G'. \quad (8.47)$$

The objective function of (IP 8-4) is to maximize the total revenue, F_{qg}, by servicing the partially admitted destinations in multicast group g associated with a specific priority, where $g \in G''$, $q \in Q$, and G'' is the new incoming user groups requesting transmission. In general, if user group g with priority q is to be given a higher priority, then the corresponding F_{gq} may be assigned a larger value.

Constraints (8.32) and (8.33) are the capacity constraints. In this model, the variable m_{gl} can be viewed as the estimate of the aggregate flows. Constraint (8.34) is a redundant constraint, which provides the upper and lower bounds of the maximum traffic requirement for multicast group g on link l. Constraint (8.35) requires that if one path is selected for group g destined for destination d, it must also be on the sub-tree adopted by multicast group g. Constraint (8.36) is the tree constraint, which requires that the union of the selected paths for the destinations of user group g forms a tree. Constraints (8.35) and (8.37) require that the number of selected incoming links, y_{gl}, is 1 or 0 and each node, except the root, has only one incoming link. Constraint (8.37) requires that the number of selected incoming links, y_{gl}, to node is 1 or 0. Constraint (8.38) requires that no selected incoming link, y_{gl}, is the root of multicast group g. As a result, the links we select form a tree. Constraints (8.39) and (8.43) require that at most one path are selected for each admitted multicast source-destination pair, while Constraint (8.40) relates the routing decision variables x_{gpd} to the auxiliary variables f_{gq}. Constraint (8.41) requires that the number of admitted destinations in multicast group g with priority q be a set of integers.

Constraint (8.43) requires that each destination of new incoming groups can only choice one path at most. Constraint (8.44) requires that no originally admitted destinations should be dropped after admitting new traffic. Constraints (8.45) – (8.47) are reroute indication constraints. If there exists a link of group $g \in G'$ rerouted, z_g will be 1 and 0 otherwise.

8.2.5 Considering Subgroup Behavior

Possible directions for future research of the MCRD problem might be: 1) extending the mode to deal with multi-group problems. 2) Multiple trees may be constructed over the network at the same time, with different data-rate demands. 3) Quality-of-service constraints, such as link capacity, hop count, and delay constraints, may be added. 4) The dependency among destinations could be made to this problem. The dependency among destinations, e.g., the members of a group can be further divided into subgroups such that the group members within each subgroup behave identically. The link utilization can be modeled as follows:

$$g_l = 1 - \prod_{m \in G} (1 - q_m (1 - \prod_{i \in M_m} (1 - f_{il}))) \quad \text{Where } G \text{ is the set of subgroups} \qquad (8.48)$$

Note that the structure of this formula resembles the constraint for link utilization of constraint (7.1), but f_{dl} replaced with $(1 - \prod_{i \in M_m} (1 - f_{il}))$.

REFERENCES

[1] M. R. Garey and D.S. Johnson, Computers and Intractability—a Guide to the Theory of NP-Completeness, Freeman: New York, 1979.

[2] F. Kenji, O. Masataka and I. Katsuo, ”Integration of Multicast Routing and QoS Routing,” *Proc. The 10th Annual Internet Society Conference*, pp. 18-21, July 2000.

[3] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, "A Framework for QoS-based Routing in the Internet", *IETF RFC2386*, Aug. 1998.

[4] S. Chen and K. Nahrstedt, “An Overview of Quality-of-Service Routing for the Next Generation High-Speed Network: Problems and Solutions,” *IEEE Network, Special Issue on Transmission and Distribution of Digital Video*, November/December 1998.

[5] J. Kleinberg, Y. Rabani and E. Tardos, “Fairness in Routing and Load Balancing,” *Proc. the 40th Annual Symposium on Foundations of Computer Science*, pp. 568-578, 1999.

[6] D. Ghosh, V. Sarangan and R. Acharya, “Quality-of-Service Routing in IP Networks,” *IEEE Transactions on Multimedia*, vol. 3, Issue 2, June 2001.

[7] Z. Wang and J. Crowcroft, “Quality-of-service routing for supporting multimedia applications”, *IEEE JSAC*, vol. 14, pp. 1228-1234, Sept. 1996.

[8] F. Ergun, R. Sinha and L. Zhang, “QoS Routing with Performance-Dependent Costs”, *Proc. IEEE INFOCOM 2000* , March, 2000.

[9] H. Fang, E. W. Zegura, “On Scalable QoS Routing: Performance Evaluation of Topology Aggregation,” *Proc. IEEE INFOCOM 2000*, May 2000.

[10] G. Apostolopoulos, R. Guérin, S. Kamat and S. K. Tripathi, “Quality of service based routing: a performance perspective,” *ACM SIGCOMM Computer Communication Review*, v.28 n.4, p.17-28, Oct. 1998.

[11] B. Wang and J. C. Hou, "Multicast routing and its QoS extension: problems, algorithms, and protocols," *IEEE Network*, vol. 14, no. 1, pp. 22-36, Jan./Fab. 2000.

[12] D. Waltzman, S. Deering, and C. Partridge, "Distance vector multicast routing protocol," *IETF RFC 1075*, Nov. 1988.

[13] J. Moy, "Multicast extensions to OSPF," *IETF RFC 1584*, Mar. 1994.

[14] A. Ballardie, "Core based trees multicast routing architecture," *IETF RFC 2201*, Sep. 1997.

[15] A. Ballardie, "Core based trees (CBT version 2) multicast routing protocol specification," *IETF RFC 2189*, Sep. 1997.

[16] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol independent multicast-sparse mode (PIM-SM): protocol specification," *IETF RFC 2362*, Jun. 1998.

[17] K. L. Calvert, E. W. Zegura and M. J. Donahoo, "Core selection methods for multicast routing," *Proc. Int. Conf. Computer Communications and Networks*, pp. 638-642, Jul. 1995.

[18] D. G. Thaler and C. V. Ravishankar, "Distributed center-location algorithms," *IEEE JSAC*, vol. 15, no. 3, pp. 291-303, Apr. 1997.

[19] S. J. Koh, M. K. Shin, J. H. Yi, J. H. Hahm and C. H. Park, "Non-core based shared tree architecture for IP multicasting," *Electronics Letters*, vol. 35, no. 11, 27th May 1999.

[20] D. Kosiur, IP multicasting: the complete guide to interactive corporate networks, John Wiley & Sons, 1998.

[21] Q. Zhu, M. Parsa and J. J. Garcia-Luna-Aceves, "A source-bases algorithm for delay-constrained minimum-cost multicasting," *Proc. IEEE INFOCOM '95*, pp. 377-385, 1995.

[22] G. N. Rouskas, and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," *IEEE JSAC*, vol. 15, no. 3, Apr. 1997.

[23] X. Jia, "A distributed algorithm of delay-bounded multicast routing for multimedia

applications in wide area networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 828-837, Dec. 1998.
[24] S. Chen, and K. Nahrstedt, "Distributed quality-of-service routing in high-speed networks based on selective probing," *Tech. Report*, Dep. of Computer Science, UIUC, 1998.
[25] J. Ignacio and P. Fraigniaud, “A survey of multicast protocols for Internet,” *Proc. AlgoTel 2000*, 2000.
[26] S. Ramanathan, "Multicast Tree Generation in Networks with Asymmetric Links," *IEEE/ACM Trans. on Networking*, vol.4, No.4, pp.558-568, Aug. 1996.
[27] F. Bauer and A. Varma, "Distributed Algorithms for Multicast Path Setup in Data Networks," *IEEE/ACM Trans. on Networking*, vol.4, No.2, pp.181-191, April 1996.
[28] L. Wei and D. Estrin, "The Trade-Offs of Multicast Trees and Algorithms," *Proc. Third Int. Conference on Computer Communications and Networking*, pp.17-24, 1994.
[29] Y. Tanaka and P.C. Huang, "Multiple Destination Routing Algorithms," *IEICE Trans. on Communication*, vol. E76-B, no.5, pp.544-552, May 1993.
[30] X. Jiang, "Distributed Path Finding Algorithm for Stream Multicast," *Computer Communication*, vol.16, no.12, pp.767-775, Dec. 1993.
[31] X. Jiang, "Routing Broadband Multicast Streams," *Computer Communication*, vol.15, no.1, pp.45-51, Jan./Feb., 1992.
[32] K. Bharathi-Kumar and J.M. Jaffe, "Routing to Multiple Destinations in Computer Networks," *IEEE Trans. on Communication*, vol.COM-31, no.3, pp.343-351, March, 1983.
[33] J. Lin and R.S. Chang, "A Comparison of the Internet Multicast Routing Protocols," *Computer Communications*, vol. 22, pp.144-155, 1999.
[34] T. Asaka, T. Miyoshi and Y. Tanaka, "Dynamic Multicast Routing Algorithm Using Predetermined Path Search," *IEICE Trans. Communication*, vol. E83-B, no.5, pp. 1128-1135, May, 2000.
[35] J. Cho and J. Breen, "Analysis of the Performance of Dynamic Multicast Routing Algorithms," *Computer Communications*, vol. 22, pp.667-674, 1999.
[36] S. Raghavan, G. Manimaran and C. S. R. Murthy, "A Rearrangeable Algorithm for the Construction of Delay-Constrained Dynamic Multicast Trees," *IEEE/ACM Trans. on Networking*, vol.7, no.4, Aug., 1999.
[37] W. Effelsberg and E. Muller-Menrad, "Dynamic Join and Leave for Real-Time Multicast," *TR-93-056*, International Computer Science Institute, Berkeley, October (1993).
[38] M. Imase and B. Waxman, "Dynamic Steiner Tree Problem," *SIAM J. Discrete Math.*, vol. 4, no. 3, pp.369-384, August 1991.
[39] F. Aiguo and M. Gerla, "Receiver-Initiated Multicasting with Multiple QoS Constraints," *Proc. IEEE INFOCOM 2000*, vol. 1, pp.62-70, 2000.
[40] X. Li, M. H. Ammar and S. Paul, “Video multicast over the Internet,” *IEEE Network*, March/April 1999.
[41] M. Ghanbari, “Two-Layered Coding of Video Signals for VBR Networks,” *IEEE JSAC*, vol. 7, no. 5, pp. 771-781, June 1989.
[42] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, “Variable Bit-Rate Coding of Video Signals for ATM Networks,” *IEEE JSAC*, vol.7, pp.801-806, June 1989.
[43] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 video standard,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301-317, Mar. 2001.
[44] W. Li, “Bit-Plane Coding of DCT Coefficients for Fine Granularity Scalability,” *ISO/IEC JTC1/SC29/WG11, MPEG98/M3989*, Oct. 1998.
[45] J. Liang, J. Yu, Y. Wang, M. Srinath and M. Zhou, “Fine Granularity Scalability Video Coding using Combination of MPEG4 Video Objects and Still Texture Objects,” *ISO/IEC JTC1/SC29/WG11, MPEG98/M4025*, Oct. 1998.

[46] W. Li, "Fine Granularity Scalability Using Bit-Plane Coding of DCT Coefficients," *ISO/IEC JTC1/SC29/WG11, MPEG98/M4204*, Dec. 1998.

[47] N. F. Maxemchuk, "Video Distribution on Multicast Networks", *IEEE JSAC*, vol. 15, no. 3, pp. 357-372, April 1997.

[48] M. Charikar, J. Naor, B. Schieber, "Resource Optimization in QoS Multicast Routing of Real-Time Multimedia," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 340-348, April 2004.

[49] H.G. Perros and K.M. Elsayed, "Call admission control schemes: A review," *IEEE Communications Magazine*, pp. 82-91, Nov. 1996.

[50] C. Cetinkaya and E. Knightly "Egress Admission Control", *Proc. INFOCOM 2000*, Mar. 2000.

[51] K. Lai and M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," *Proc. SIGCOMM 2000*, pp. 283-294, 2000.

[52] V. Fioriu and D. Towsley, "Call admission and resource reservation for multicast sessions," *Technical Report 95-17*, University of Massachusetts, 1995.

[53] X. Jia, Y. Zhang, N. Pissinoun and S. Makkis "An Integrated Routing andAdmission Control Mechanism for Real-time Multicast Connection Establishment," *International Journal of Communication Systems*, vol. 14, no. 3, pp 287-303, April 2001.

[54] E. Pagani and G. P. Rossi, "Distributed Bandwidth Broker for QoS Multicast Traffic", *Proc. 22nd IEEE ICDCS*, Vienna, pp. 319-326, Jul. 2-5 2002.

[55] N. Tang, S. Tsui and L. Wang, "A survey of admission control algorithms," *Technical Report*, December 1998.

[56] S. Jamin, P. B. Danzig, S. J. Shenker and L. Zhang, "A measurement-based admission control algorithm for integrated service packet networks," *IEEE/ACM Transactions on Networking*, vol. 5, No. 1, Feb. 1997, pp. 56-70.

[57] Berger, A. W. and Whitt, W., "Effective bandwidths with priorities," *IEEE Transactions on Networking*, vol. 6, no. 4, pp. 447, Aug. 1998.

[58] C. S. Chang and J. A. Thomas "Effective bandwidth in high-speed digital networks," *IEEE JSAC*, vol. 13, no. 6, Aug. 1995.

[59] R. Guerin, A. Ahmadi, and M. Nagshineh, "Equivalent capacity and its applications to bandwidth allocation in high-speed networks," *IEEE JSAC*, vol. 9, no. 7, Sep. 1991.

[60] D. Wu, Y. T. Hou, W. Zhu, T. Q. Zhang and J. Peha, "Streaming video over the Internet: approaches and directions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 282-300, March 2001.

[61] G. Karlsson and M. Vetterli, "Packet video and its integration into the network architecture", *IEEE JSAC*, vol. 7, pp. 739–751, 1989.

[62] E. Amir, S. McCanne and H. Zhang, "An application level video gateway," *Proc. ACM Multimedia*, pp. 255-265, Nov. 1995.

[63] T. Turletti and J. C. Bolot, "Issues with multicast video distribution in heterogeneous packet networks", *Proc. the 6th International Workshop on Packet Video*, pp. F3.1-F3.4, 1994.

[64] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A new resource reservation protocol", *IEEE Network*, vol. 7, pp. 8–18, 1993.

[65] P. Winter, "Steiner problem in networks: A survey," *Networks*, pp. 129-167, 1987.

[66] F. K. Hwang, Steiner Tree Problems, *Networks*, pp. 55-89, 1992.

[67] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems", *Management Science*, vol. 27, pp. 1-18, 1981.

[68] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, ISBN0-13-027765-7, 1993.

[69] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," *Math Japonica*, vol. 6, pp. 573-577, 1980.

[70] A. M. Geoffrion, "Lagrangean relaxation and its uses in integer programming," *Math. Programming Study*, vol. 2, pp. 82-114, 1974.

[71] M. Held and R. M. Karp, "The Traveling salesman problem and minimum spanning trees: Part I," *Operations Research*, vol. 18, pp. 1138-62, 1970.
[72] C. Faloutsos, P. Faloutsos, and M. Faloutsos, "On power-law relationships of the Internet topology," *Proc. of the ACM SIGCOMM*, Sept. 1999.
[73] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286. pp. 509–512, Oct. 1999.
[74] D. J. Watt and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440-442, June 1998.
[75] V. Batagelj and U. Brandes, "Efficient generation of large random networks," *Physical Review*, 2005.

Received Award:

Best Student Paper Award 7th IFIP/IEEE International Conference on Management of Multimedia and Network Services 2004

Conference Publications:

[1] Hsu-Chen Cheng and Frank Yeong-Sung Lin, "Minimum-Cost Multicast Routing for Multi-Layered Multimedia Distribution," *Proc. 7th IFIP/IEEE International Conference on Management of Multimedia and Network Services (MMNS '04),* pp.102-114, October 3- 6, 2004, San Diego, USA.

[2] Hsu-Chen Cheng, Chih-Chun Kuo and Frank Yeong-Sung Lin, A Multicast Tree Aggregation Algorithm in Wavelength-routed WDM Networks, *Proc. 2004 SPIE Asia-Pacific Optical Communications Conference (APOC '04),* November 7–11, 2004, Beijing, China.

[3] Hsu-Chen Cheng and Frank Yeong-Sung Lin, "A Capacitated Minimum-Cost Multicast Routing Algorithm for Multirate Multimedia Distribution," *Proc. 2004 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '04),* November 18~19, 2004. Seoul, Korea.

[4] Hsu-Chen Cheng and Frank Yeong-Sung Lin, "Maximum-Revenue Multicast Routing and Partial Admission Control for Multimedia Distribution," *Proc. International Computer Symposium (ICS'04),* Dec. 15-17, 2004, Taipei, Taiwan.

[5] Frank Yeong-Sung Lin, Hsu-Chen Cheng and Jung-Yao Yeh, "A Minimum Cost Multicast Routing Algorithm with the Consideration of Dynamic User Membership," *Proc. 2005 The International Conference on Information Networking (ICOIN 2005),* Jan. 31- Feb 2, Jeju, Korea.

[6] Hsu-Chen Cheng and Frank Yeong-Sung Lin, "Maximum-Revenue Multicast Routing and Partial Admission Control for Multirate Multimedia Distribution," *Proc. The 19th IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA2005)*, pp.21-26, Taiwan.

[7] Chih-Wei Shiou, Frank Yeong-Sung Lin, Hsu-Chen Cheng and Yean-Fu. Wen, "Optimal Energy-Efficient Routing for Wireless Sensor Networks," *Proc. The 19th IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA2005)*, pp. 325-330, Taiwan.

Book Chapters:

- Minimum-Cost Multicast Routing for Multi-Layered Multimedia Distribution, *Lecture Notes in Computer Science* (LNCS), No. 3271, October 2004.
- A Minimum Cost Multicast Routing Algorithm with the Consideration of Dynamic User Membership, *Lecture Notes in Computer Science* (LNCS), No. 3395, 2005.