國立臺灣大學資訊管理研究所

碩士論文

Department of Information Management
National Taiwan University

Master Thesis

移動式隨意網路下多群組群播之
低延遲與能耗排程演算法

A Low-latency and Energy-efficient Scheduling Algorithm for Multi-group Multicasting in Mobile Ad Hoc Networks

李培維

Pei-Wei Li

指導教授：林永松 博士

Advisor: Frank Yeong-Sung Lin, Ph.D.

中華民國九十八年七月

July, 2009

移動式隨意網路下多群組群播之低延遲與能耗排程演算法

A Low-latency and Energy-efficient Scheduling Algorithm for Multi-group Multicasting in Mobile Ad Hoc Networks

本論文係提交國立台灣大學
資訊管理學研究所作為完成碩士
學位所需條件之一部份

研究生： 李培維 撰

中華民國九十八年七月

謝詞

本篇碩士論文的正式完成，代表了兩年來的研究成果，也意味著學生生涯的結束，這一路上有好多幫助我的人，在此向他們一一致謝。首先，要感謝我的指導教授 林永松老師，總是很有耐心的和我討論，遇到困難時指引我正確的方向，不斷的鼓勵我，在多次討論至深夜的時刻，也十分關心我的安危，點點滴滴學生都銘記在心，老師您辛苦了！ 此外，感謝呂俊賢教授、林盈達教授、鐘嘉德教授、傅新彬教授在口試時給予我許多寶貴的建議，使本篇論文能更臻完善。

實驗室的夥伴們是我們最強大的後盾，感謝柏皓學長對我的鼓勵、Sherry 學姐對我的關心、政達學長幫我化解了許多研究上的疑問、明宗學長和俊甫學長給予我論文上的建議與修正。還有最棒的四位碩一學弟世昌、耀元、怡緯、永斌，真的非常感謝你們在口試的時候協助處理許多事務，讓我們沒有後顧之憂專心準備口試。

再來陪我走過兩年研究生活的好朋友 FuFu、YY、Wei、Ray、阿保、猷順，非常感謝這些時間以來的照顧，難過的時候幫我加油打氣，這段一起努力的日子，只會讓我們的友情更加堅定，真的很高興能認識你們，也希望在未來的人生中，我們都能擁有屬於自己的一片天空。也要特別感謝子雋，在口試期間和論文相關事務上的大力協助。

最感謝的便是我的家人，我的爸媽是世界上最好的父母，從小到大對我的栽培，尤其在我論文最艱辛的時刻，無怨無悔的付出，照顧我生活上的大小事，讓我可以全力專心做研究，沒有他們就沒有今天的我。還有謝謝我的弟弟在熬夜的日子裡幫我買宵夜，默默的陪伴我至深夜。最後要感謝一直陪在我身邊的耀寬，總是對我很有耐心，遇到困難心情沮喪時給我最正面的想法與鼓勵，讓我能夠有勇氣面對接下來的挑戰，希望未來的日子裡還能繼續互相扶持。

李培維 謹 識

于台大資訊管理研究所

民國九十八年八月

論文摘要

論文題目：移動式隨意網路下多群組群播之低延遲與能耗排程演算法

作者：李培維

指導教授：林永松 博士

移動式隨意網路由許多自主移動的節點所組成，節點之間以無線媒介通訊。此網路架構不需仰賴已存的基礎建設，而網路拓樸會隨著節點的移動而持續變化。另一項重要的特性即為節點的電力資源有限，能量耗損會影響網路壽命長短，因此成為一項重要的效能衡量指標。

群播為移動式隨意網路中許多應用的基本運作機制，這些應用大部份都強調資訊傳輸的延遲限制，因此需要一個低延遲的群播演算法來滿足其限制，然而在設計的過程中同時考慮上述議題包括低能耗和節點移動性，會使問題變得十分複雜。

本論文主要研究在移動式隨意網路中多群組進行群播時，如何進行路由以及節點傳輸時間的排程問題。我們將此問題設計成一個數學模型，目標為最小化群播的延遲時間，同時我們也將群播產生之能耗限制在某個合理的範圍並且避免傳輸時會產生的資料碰撞，此碰撞情形會造成能量消耗。最後我們提出以拉格蘭日鬆弛法為基礎的演算法來解決此問題。我們設計一系列的實驗以測試演算法的表現，實驗結果顯示此演算法在多種網路情境下均能提供低延遲以及節能的傳輸排程。

關鍵字：移動式隨意網路、排程、群播、高效率節能、低延遲、移動性、拉格蘭日鬆弛法

THESIS ABSTRACT

DEPARTMENT OF INFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

NAME: PEI-WEI LI

ADVISOR: FRANK YEONG-SUNG LIN

A Low-latency and Energy-efficient Scheduling Algorithm for Multi-group Multicasting in Mobile Ad Hoc Networks

A Mobile Ad Hoc Network (MANET) consists of a set of mobile nodes which communicate over the wireless medium. The network topology, which does not relay on any pre-existing infrastructure, changes rapidly due to the mobility of nodes. Another property of such networks is that the battery capability of nodes is limited. As a result, energy-efficiency becomes an important performance measure since it directly affects the network lifetime.

In MANET, multicasting is a fundamental operation to a wide range of applications which impose end-to-end latency constraints of transmissions. Designing a low-latency multicast protocol which satisfies these constraints is crucial important. However, it becomes a challenging task while addressing the critical issues of energy-efficiency and mobility at the same time.

In this thesis, we focus on the problem of routing and transmission scheduling for multi-group multicasting in MANET. We formulate the problem as a linear integer programming problem, in which the objective is to minimize the latency of multicasting. In addition, the formulation ensures the energy consumption within a reasonable range and avoids possible collisions of transmission which consume a large amount of valuable energy resources. A set of heuristic algorithms based on Lagrangean relaxation method is proposed to solve this problem. We conduct a series of experiment designed from the perspective of design and operation both. Experimental studies indicate that our algorithm has good performance and high practicability under various network conditions.

Keywords: MANET, Multicast, Scheduling, Energy-Efficient, Low-latency, Mobility, Lagrangean Relaxation Method

TABLE OF CONTENTS

謝詞 I

論文摘要 III

THESIS ABSTRACT V

TABLE OF CONTENT VII

LIST OF TABLES IX

LIST OF FIGURES XI

CHAPTER 1 INTRODUCTION 1

1.1 BACKGROUND 1

1.2 MOTIVATION 4

1.3 LITERATURE SURVEY 6

1.3.1 Multicasting in Ad Hoc Networks *6*

1.3.2 Scheduling of Broadcasting *11*

1.3.3 Mobility Model *14*

1.4 PROPOSED APPROACH 18

1.5 THESIS ORGANIZATION 19

CHAPTER 2 PROBLEM FORMULATION 21

2.1 PROBLEM DESCRIPTION 21

2.2 PROBLEM NOTATION 31

CHAPTER 3 SOLUTION APPROACH 43

3.1 INTRODUCTION TO LAGRANGEAN RELAXATION METHOD 43

3.2 LAGRANGEAN RELAXATION 46

3.2.1 Subproblem 1 (related to decision variable x_p) *49*

3.2.2 Subproblem 2 (related to decision variable y_{ijs}) *49*

3.2.3 Subproblem 3 (related to decision variable n_{is}) *51*

3.2.4 Subproblem 4 (related to decision variable ϕ_{ij}) *52*

3.2.5 Subproblem 5 (related to decision variable h_{ijt}) *53*

3.2.6 Subproblem 6 (related to decision variable r_i, $\Phi_{ij}(r_i)$ and $p(r_i)$) *54*

3.2.7 Subproblem 7 (related to decision variable b_{its}) *54*

3.2.8 Subproblem 8 (related to decision variable z_t) *57*
3.3 THE DUAL PROBLEM AND THE SUBGRADIENT METHOD 58
CHAPTER 4 GETTING PRIMAL FEASIBLE SOLUTION **59**
4.1 HEURISTIC FOR ROUTING POLICY 60
4.2 HEURISTIC FOR CALCULATING MAXIMAL TRANSMISSION TIME 61
4.3 HEURISTIC FOR SCHEDULING POLICY 64
4.4 HEURISTIC FOR SAVING ENERGY CONSUMPTION 66
4.5 LAGRANGEAN RELAXATION BASED ALGORITHM 67
CHAPTER 5 COMPUTATIONAL EXPERIMENTS **69**
5.1 EXPERIMENT ENVIRONMENT 69
5.2 SIMPLE ALGORITHM FOR COMPARISON AND PERFORMANCE METRICS 70
5.3 EXPERIMENT SCENARIO 71
5.3.1 Static Network with Different Number of Nodes *71*
5.3.2 Static Network with Different Number of Source Nodes *74*
5.3.3 Dynamic Network with Different Density of Destination Nodes *75*
5.3.4 Static Network with Different Transmission Radius *78*
5.3.5 Dynamic Network with Different Transmission Radius *81*
5.3.6 Summary *83*
5.4 IMPLEMENTATION IN LONG-TERM OPERATION 84
5.4.1 A Scenario of Constant Speed during a Phase *87*
5.4.2 A Scenario of Variable Speed during a Phase *98*
CHAPTER 6 CONCLUSION **113**
6.1 SUMMARY 113
6.2 FUTURE WORK 114
REFERENCES **116**

LIST OF TABLES

TABLE 2-1 PROBLEM DESCRIPTION 28

TABLE 2-2 NOTATION DESCRIPTIONS OF GIVEN PARAMETERS 31

TABLE 2-3 NOTATION DESCRIPTIONS OF DECISION VARIABLES 33

TABLE 2-4 EXPLANATION OF CONSTRAINT (2.10) AND (2.11) 38

TABLE 2-5 EXPLANATION OF CONSTRAINT (2.14) AND (2.15) 39

TABLE 2-6 EXPLANATION OF CONSTRAINT (2.32) AND (2.33) 41

TABLE 4-1 ROUTING HEURISTIC 61

TABLE 4-2 CALCULATING MAXIMAL TRANSMISSION TIME 63

TABLE 4-3 SCHEDULING HEURISTIC 65

TABLE 4-4 HEURISTIC OF SAVING ENERGY CONSUMPTION 66

TABLE 4-5 LAGRANGEAN RELAXATION BASED ALGORITHM 67

TABLE 5-1 EXPERIMENT ENVIRONMENT AND PARAMETERS 69

TABLE 5-2 PARAMETERS OF STATIC NETWORK WITH DIFFERENT NUMBER OF NODES 71

TABLE 5-3 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT NUMBER OF NODES 72

TABLE 5-4 PARAMETERS OF STATIC NETWORK WITH DIFFERENT NUMBER OF NODES 73

TABLE 5-5 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT NUMBER OF NODES 73

TABLE 5-6 PARAMETERS OF STATIC NETWORK WITH DIFFERENT NUMBER OF SOURCE NODES 74

TABLE 5-7 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT NUMBER OF SOURCE NODES 74

TABLE 5-8 PARAMETERS OF DYNAMIC NETWORK WITH DIFFERENT DENSITY OF DESTINATION NODES 76

TABLE 5-9 EXPERIMENT RESULT OF DYNAMIC NETWORK WITH DIFFERENT DENSITY OF DESTINATION NODES 76

TABLE 5-10 ENERGY CONSUMPTION OF DYNAMIC NETWORK WITH DIFFERENT DENSITY OF DESTINATION NODES 76

TABLE 5-11 PARAMETERS OF STATIC NETWORK WITH DIFFERENT TRANSMISSION

RADIUS 79

TABLE 5-12 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT TRANSMISSION RADIUS 79

TABLE 5-13 ENERGY CONSUMPTION OF STATIC NETWORK WITH DIFFERENT TRANSMISSION RADIUS 80

TABLE 5-14 PARAMETERS OF DYNAMIC NETWORK WITH DIFFERENT TRANSMISSION RADIUS 81

TABLE 5-15 EXPERIMENT RESULT OF DYNAMIC NETWORK WITH DIFFERENT TRANSMISSION RADIUS 82

TABLE 5-16 PARAMETERS OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT SPEED 88

TABLE 5-17 EXPERIMENT RESULT OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT SPEED 89

TABLE 5-18 PARAMETERS OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT NUMBER OF TIME SLOTS IN EACH PHASE 93

TABLE 5-19 EXPERIMENT RESULT OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT NUMBER OF TIME SLOTS IN EACH PHASE 94

TABLE 5-20 PARAMETERS OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT SPEED 99

TABLE 5-21 EXPERIMENT RESULT OF VARIABLE SPEED DURING A PHASE WITH SPEED = 0.1~1.1 100

TABLE 5-22 EXPERIMENT RESULT OF VARIABLE SPEED DURING A PHASE WITH SPEED = 0.9~1.9 101

TABLE 5-23 PARAMETERS OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT SPEED CHANGE FREQUENCY 105

TABLE 5-24 EXPERIMENT RESULT OF VARIABLE SPEED DURING A PHASE WITH SPEED CHANGE FREQUENCY = 2, 3, 4, 6 106

TABLE 5-25 EXPERIMENT RESULT OF VARIABLE SPEED DURING A PHASE WITH SPEED CHANGE FREQUENCY =10, 15, 30 107

LIST OF FIGURES

FIG 1-1 WIRELESS BROADCAST ADVANTAGE AND MULTI-HOP TRANSMISSION 2

FIG 1-2 AN EXAMPLE OF TREE-BASED MULTICAST 7

FIG 1-3 AN EXAMPLE OF MESH-BASED MULTICAST 8

FIG 1-4 WIRELESS MULTICAST ADVANTAGE 10

FIG 1-5 BREADTH FIRST SEARCH TREE 13

FIG 1-6 BROADCAST TREE AND TRANSMISSION SCHEDULE 13

FIG 1-7 TRAVELING PATTERN OF MOBILE NODES USING RANDOM WALK MOBILITY MODEL.. 15

FIG 1-8 TRAVELING PATTERN OF MOBILE NODES USING RANDOM WAYPOINT MOBILITY MODEL [12] 16

FIG 1-9 TRAVELING PATTERN OF MOBILE NODES USING RANDOM DIRECTION MOBILITY MODEL 17

FIG 1-10 TRAVELING PATTERN OF MOBILE NODES USING GAUSS-MARKOV MOBILITY MODEL.. 18

FIG 2-1 THE SCENARIO OF MULTIPLE MULTICAST TREE 21

FIG 2-2 THE SCHEDULE OF MULTI-GROUP MULTICASTING 23

FIG 2-3 TIME FRAME DIVISION 24

FIG 2-4 ILLUSTRATION OF LINK HOLDING TIME t 25

FIG 3-1 ILLUSTRATION OF LAGRANGEAN RELAXATION METHOD 44

FIG 3-2 PROCEDURES OF LAGRANGEAN RELAXATION METHOD 45

FIG 5-1 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT NUMBER OF NODES 72

FIG 5-2 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT NUMBER OF NODES 74

FIG 5-3 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT NUMBER OF SOURCE NODES 75

FIG 5-4 EXPERIMENT RESULT OF DYNAMIC NETWORK WITH DIFFERENT DENSITY OF DESTINATION NODES 76

FIG 5-5 ENERGY CONSUMPTION OF DYNAMIC NETWORK WITH DIFFERENT DENSITY OF DESTINATION NODES 77

FIG 5-6 EXPERIMENT RESULT OF STATIC NETWORK WITH DIFFERENT TRANSMISSION

RADIUS 79

FIG 5-7 ENERGY CONSUMPTION OF STATIC NETWORK WITH DIFFERENT TRANSMISSION RADIUS 80

FIG 5-8 EXPERIMENT RESULT OF DYNAMIC NETWORK WITH DIFFERENT TRANSMISSION RADIUS 82

FIG 5-9 THE TIME DIVISION OF IMPLEMENTATION IN LONG-TERM OPERATION........ 86

FIG 5-10 SCENARIO OF CONSTANT SPEED DURING A PHASE 87

FIG 5-11 EXPERIMENT RESULT OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT SPEED 90

FIG 5-12 ENERGY CONSUMPTION OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT SPEED 90

FIG 5-13 EXPERIMENT RESULT OF EACH PHASE IN THE SCENARIO OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT SPEED 91

FIG 5-14 EXPERIMENT RESULT OF EACH PHASE IN THE SCENARIO OF CONSTANT SPEED DURING A PHASE WITH SPEED=0.9~1.3 92

FIG 5-15 EXPERIMENT RESULT OF EACH PHASE IN THE SCENARIO OF CONSTANT SPEED DURING A PHASE WITH SPEED = 0.7~1.1 92

FIG 5-16 EXPERIMENT RESULT OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT NUMBER OF TIME SLOTS IN EACH PHASE 95

FIG 5-17 EXPERIMENT RESULT OF EACH PHASE IN THE SCENARIO OF CONSTANT SPEED DURING A PHASE WITH DIFFERENT NUMBER OF TIME SLOTS IN EACH PHASE 96

FIG 5-18 EXPERIMENT RESULT OF EACH PHASE IN THE SCENARIO OF CONSTANT SPEED DURING A PHASE WITH NUMBER OF TIME SLOTS = 60 IN EACH PHASE 97

FIG 5-19 SCENARIO OF VARIABLE SPEED DURING A PHASE........ 98

FIG 5-20 AVERAGE DELIVERY RATIO 1 OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT SPEED 102

FIG 5-21 AVERAGE DELIVERY RATIO 2 OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT SPEED 102

FIG 5-22 DELIVERY RATIO 1 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT SPEED........ 103

FIG 5-23 DELIVERY RATIO 1 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH SPEED = 0.7~1.1 104

FIG 5-24 DELIVERY RATIO 2 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED

DURING A PHASE WITH DIFFERENT SPEED.......... 104

FIG 5-25 DELIVERY RATIO 2 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH SPEED = 0.5~0.9 105

FIG 5-26 AVERAGE DELIVERY RATIO 1 OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT CHANGE FREQUENCY OF SPEED AND DIRECTION 108

FIG 5-27 AVERAGE DELIVERY RATIO 2 OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT CHANGE FREQUENCY OF SPEED AND DIRECTION 108

FIG 5-28 DELIVERY RATIO 1 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH DIFFERENT CHANGE FREQUENCY OF SPEED AND DIRECTION 109

FIG 5-29 DELIVERY RATIO 1 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH CHANGE FREQUENCY = 4 110

FIG 5-30 DELIVERY RATIO 2 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH CHANGE FREQUENCY = 2, 3, 4 110

FIG 5-31 DELIVERY RATIO 2 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH CHANGE FREQUENCY = 30, 15, 10 111

FIG 5-32 DELIVERY RATIO 2 OF EACH PHASE IN THE SCENARIO OF VARIABLE SPEED DURING A PHASE WITH CHANGE FREQUENCY = 15 111

Chapter 1 Introduction

1.1 Background

Mobile ad hoc networks (MANET) are composed of a collection of mobile nodes which are connected by wireless channels and communicate without the assistance of any base station. MANET does not rely on any underlying fixed infrastructure since nodes move arbitrarily causing the rapidly changing of network topology [1].

Due to the nature of wireless medium, assuming the omni-direction antennas are used, the transmission of a node can be received by all nodes which are in its transmission range. The property is called “Wireless Broadcast Advantage” (WBA) [2], which can save additional cost for a single transmission of a node to reach all neighbor nodes. Therefore, two nodes can communicate with each other directly if they are within each other’s transmission range. However, two nodes which want to communicate with each other are usually out of one another’s transmission range. In this case, they must have the support of some intermediate nodes relaying the messages. There is a simple example below. Node A can transmit the packet to node B directly. However, if node A wants to sends a packet to node C, it has to forward the packet to node B first, then node B relays the packet to node C.

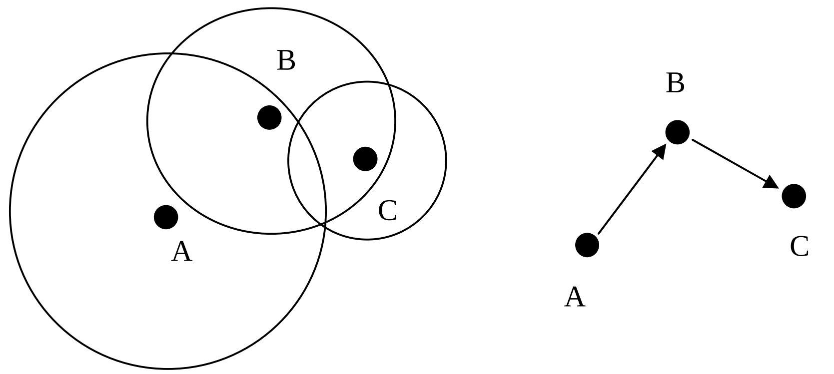


Fig 1-1 Wireless Broadcast Advantage and Multi-Hop transmission

Multicasting is the transmission of packets to multiple receivers called multicast group members which form a multicast group. A node can join or leave the group arbitrarily and also can be a member of more than one group at a time [3].

The major applications in MANET include military communications, emergency relief, rescue sites, video conferencing, and video/audio broadcasting. Multicasting is the fundamental operation to these applications which require close collaboration of the member nodes. Also, these applications require time-critical information. It is important to design an efficient multicast algorithm in which the end-to-end latency of multicasting, which is the time taken for all multicast members receiving the packet, should be as minimal as possible. However, routing and scheduling are not easy task under such dynamic environment in which nodes are mobile and a path may break at any time. Therefore, the reliable paths to all multicast members should be chosen since it can improve the packet delivery ratio.

Another important issue is that power and bandwidth are valuable resources due to the limited battery capability of nodes. If the routing and scheduling of multicast packets is not carefully designed, it will result in transmission collision and network congestion that consume a large amount of energy. Hence, energy-efficiency becomes an important performance measure since it directly affects the network lifetime.

In conclusion, multicasting in MANET becomes a challenging problem considering those important issues concurrently.

1.2 Motivation

There is an interesting trade-off among energy consumption, end-to-end latency of multicast, and link holding time between nodes. The effect is more significant when the transmission radius (power) of each node is adjustable. There are two extreme cases. In the first case, when all nodes transmit with their maximum possible radius, they can reach more nodes and deliver the packets to the multicast members more quickly. Also, the link between two neighbor nodes will sustain for longer time before breaking because the node has to move for longer distance to be out of the transmission range of the other node. Consequently, it can minimize the latency of multicasting and maximize the link holding time. However, it will lead to tremendous energy consumption while transmitting. In the second case, when all nodes narrow their transmission radius as small as possible, the energy consumption can be minimized; but it will result in high latency and short link holding time. It is important to develop the optimal solution of balancing the latency and energy consumption by adjusting the transmission radius of nodes. This is exactly the core of our work.

To make our work more realistic, we consider the scenario that there are multiple multicast groups in the network rather than single multicast group. The scheduling of multi-group multicasting is more complicated because the collision of transmission will occur more frequently. A collision-free transmission scheduling algorithm can improve the delivery ratio in the wireless environment as the transmission error rate is high.

In order to consider the property of mobility, we use the well-known mobility model which can simulate the realistic movement pattern to predict the behavior of nodes. Therefore, the link holding time also can be predicted easily. We try to choose reliable route to forward the packet and finish the transmission before the link breaks.

Many researchers have engaged in the field of wireless communication especially in MANET for many years. They developed many efficient protocols to improve the performance of a variety of applications in MANET. That is the motivation of my work to propose an efficient scheduling algorithm for multi-group multicasting while addressing all these important issues together.

1.3 Literature Survey

1.3.1 Multicasting in Ad Hoc Networks

1) Multicast protocols for MANET can be categorized into two types according to the multicast structure [4].

   - Tree-based (Figure 1-2): The structure is a tree rooted at multicast source and can reach to all multicast group members, in which every node on the tree has only one parent. It can be further divided into two methods. The first method is per-source tree, that is, every multicast source maintains a tree to reach all its group members. The multicast packet can be forwarded along the most efficient path from source to each member. However, this method may incur a large amount of overhead due to many multicast tree have to be maintained. The second method is called shared tree, in which multicast packets from all multicast sources are unicast to a core node at first. A shared tree rooted and maintained at the core node must reach all members of all multicast groups. Then all multicast packets are distributed along the shared tree until they reach all multicast group members. This method has lower control overhead, but the path from a source to a destination is not necessarily optimal (shortest).

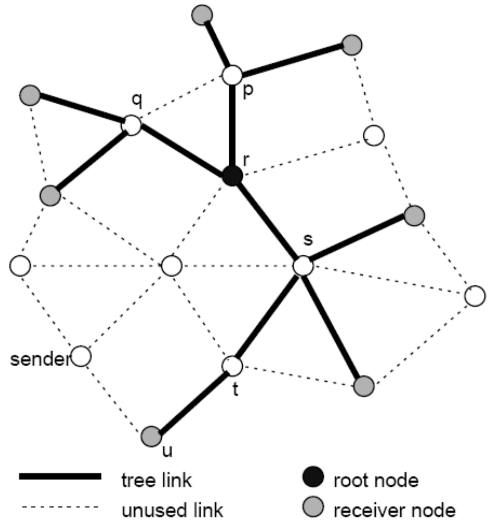


Fig 1-2 An Example of Tree-based Multicast [4]

- Mesh-based (Figure 1-3): The mesh structure is different from the tree structure since every node on the mesh can have multiple parents. This approach constructs a mesh structure with redundant links between nodes. There are still other alternative paths available if primary path is broken due to the mobility of nodes. Therefore, it can improve packet delivery ratio under such environment. However, this method may perform worse in terms of energy efficiency since multiple redundant paths must be maintained.

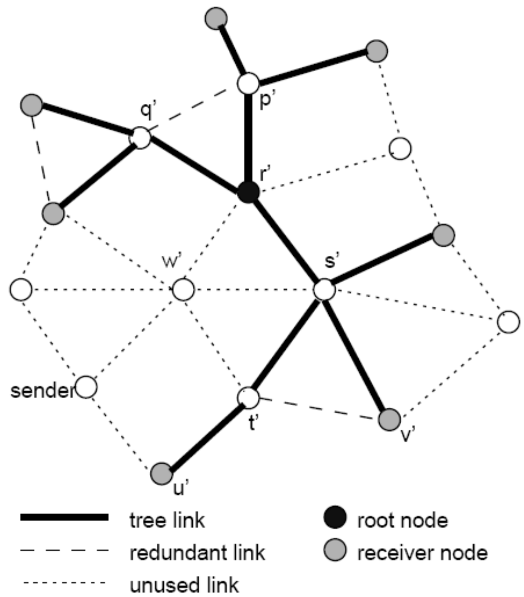


Fig 1-3 An Example of Mesh-based Multicast [4]

In conclusion, mesh-based structure is more robust to mobility, but tree-based structure may be preferable when energy is a primary concern. Hence, in order to achieve energy efficiency, we adopt the tree-based method in our formulation.

2) An energy-efficient multicast protocol is used to determine a minimum-power multicast tree rooted at the multicast source. Packets are forwarded along the tree to all of the multicast members. There are many algorithms proposed in recent years, e.g., [5] used centralized approach and [6] used distributed approach. Both of them do not address the mobility issue.

A. Energy model

The following energy model [5] is used in proposed algorithms. Assume that each node can adjust its power level within a given range. The received signal power is assumed to be equal to $pr^{-\alpha}$, where p is the transmission power, r is the distance and α is a parameter called path loss which depends on the characteristics of the communication medium and is usually between 2 (in unobstructed environment) and 4 (in urban environment). A simplified interference model is used in which the interference level is independent of network traffic and the same at all nodes. Hence, the transmission power required to support a link between two nodes with distance r is proportional to r^{α} since the received signal power must exceed some threshold. Without loss of generality, we set this threshold equal to 1. As a result,

$p_{ij} =$ power required for the link between node i and node j $= r_{ij}^{\alpha}$,

where r_{ij} is the distance of node i and node j.

B. Wireless multicast advantage

Assuming the omni-direction antenna is used, a transmission of a node can be received if nodes are in its communication range. Consider the example below, in which node i transmits to its neighbors, node j and node k. The power required to reach node j is P_{ij} and the power required to reach node k is P_{ik}. A single transmission with power $P_{i(j,k)} = \max\{P_{ij}, P_{ik}\}$ is sufficient to reach both node j

and node k. This property of wireless communication is called "wireless multicast advantage" [5] that makes multicasting an excellent setting to study the potential benefits of energy-efficient protocols.

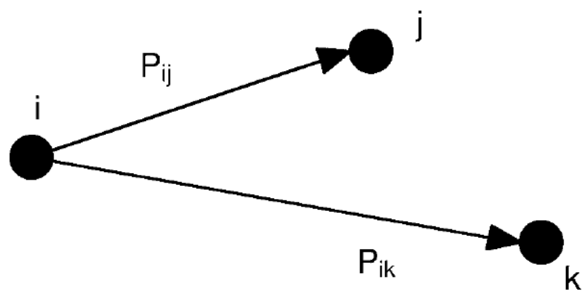


Fig 1-4 Wireless Multicast Advantage [5]

Therefore, the total power required for a node to reach a set of its neighbors is simply the maximum required to reach any of them individually, and the total power associated with the multicast tree is the sum of the powers of all transmitting nodes.

- Broadcast Incremental Power (BIP) and Multicast Incremental Power (MIP) [5]

  In the beginning, BIP determine the node that the source node can reach with minimal power consumption, that is, the nearest neighbor. Then BIP choose a node that can be reached with minimal additional cost (power) from current formed tree and add this node to the tree. This procedure is repeated until there is no unconnected node left. A minimum-cost broadcast tree is formed. In MIP, in order to obtain the multicast tree, the broadcast tree is pruned by eliminating the transmission that is not necessary for reaching all members of the multicast group. The nodes without downstream destinations will not transmit, and some nodes will be able to reduce their transmission power if their farther downstream neighbors have been pruned from the tree.

- Broadcast Decremental Power (BDP) and Multicast Decremental Power (MDP) [6]

  This algorithm is a distributed protocol which is based on Bellman-Ford algorithm. At first, a spanning tree is constructed by applying Bellman-Ford algorithm. Then each node asks all its neighbor nodes about how much additional cost must pay to be its parent and chooses the node with the smallest additional cost as its parent. After the adjusting procedure, a broadcast tree with minimum cost is formed.

1.3.2 Scheduling of Broadcasting

Network wide broadcasting is a fundamental operation in ad hoc networks. The goal is to send data to all nodes in the network. A simple mechanism proposed first is flooding where every node in the network retransmits the flooding message to its neighbors after receiving it. It guarantees the flooding message will reach all nodes in the network only if the network is connected. Nevertheless, it generates a large amount of redundant traffic that causes the congestion, resource contention, and transmission collision. This is called the broadcast storm problem [7]. Many efficient algorithms were proposed to solve the problem [8]. However, some reduce redundant transmission but ignore the latency of broadcast and possible collision introduced. In order to make good use of the resource in an efficient way, one of the good solutions is to schedule the transmission order [9].

A set of algorithms were proposed to solve the transmission scheduling problem. Our work is based on the research of Gandhi et al. [10] that proposed a collision-free broadcast scheduling algorithm to minimize the latency and the number of transmissions in the broadcast. The algorithm consists of two cases.

1) Single broadcast source: there are two stages to compute a schedule of broadcasting.

- Broadcast tree construction:

  First, construct a breadth first search tree (BFS) that is rooted at source. Next, choose the appropriate parent for each node on BFS tree level by level. The final result is a broadcast tree rooted at source which can reach all nodes in the network.

- Schedule of transmission:

  According to the broadcast tree, each non-leaf node u will transmit at time t that satisfies the following constraints: 1) u has received the message before time t, 2) none of its neighbors receive the message at time t.

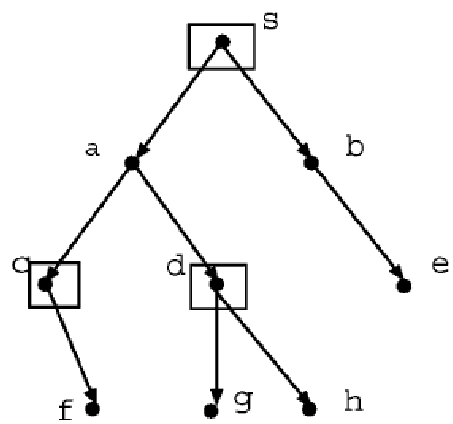


Fig 1-5 Breadth First Search Tree [10]

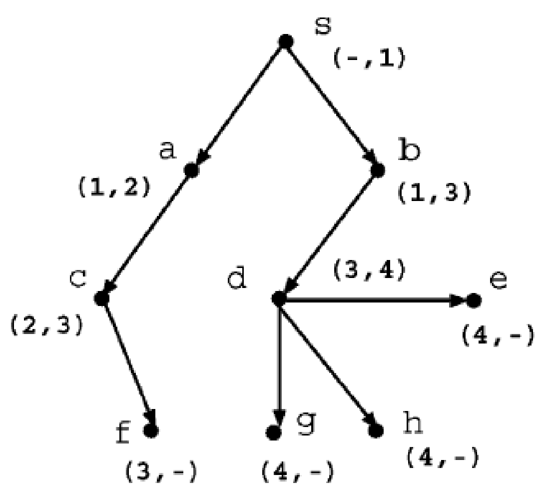


Fig 1-6 Broadcast Tree and Transmission Schedule [10]

2) Multiple broadcast sources: two algorithms are presented.

- Intermediate source broadcast (ISB):

  Messages from multiple sources get unicast to an arbitral intermediate source. A scheduling strategy ensures that the unicast is collision free. After the intermediate source receives all messages, it broadcast them using single source broadcast algorithm. It has lower computational complexity but higher broadcast latency.

- Multi-source broadcast (MSB):

  Each source uses single source broadcast algorithm to construct a broadcast tree and schedule the transmission. The transmission of different messages on different trees avoids collision by using the scheduling strategy. It has higher computational complexity but lower broadcast latency.

We adopt the idea of MSB in our work since the latency is the issue concerned. To make the problem more generic, we transform the broadcast problem addressed in the research of Gandhi et al. into a multicast problem since broadcast is a special case of multicast. In addition, we consider the scenario of multiple multicast groups in which the transmission range of a node is adjustable.

1.3.3 Mobility Model

In order to evaluate the performance of a protocol for ad hoc networks, it is necessary to use a mobility model that can best describe the behavior of mobile nodes. In realistic environment, a mobile node may change its speed and direction in reasonable time slots. The mobility model should accurately mimic the movement of mobile nodes including speed and direction during a reasonable time interval. Currently, there are two types of mobility models used to simulate. One of them is synthetic entity mobility model that attempts to realistically represent the behavior of mobile node. There are different synthetic entity mobility models for ad hoc networks [11] [12].

1) Random Walk Mobility Model: A mobile node moves from it current location to a new location with a speed and direction randomly chosen from predefined ranges, $[Minspeed, Maxspeed]$ and $[0, 2\pi]$ respectively. Upon reaching a location, it will turn to another location without taking pause. This model is a memoryless that the current speed and direction of a mobile node is independent of its past speed and direction. It can be used to mimic the environment that mobile nodes move in extremely unpredictable ways.

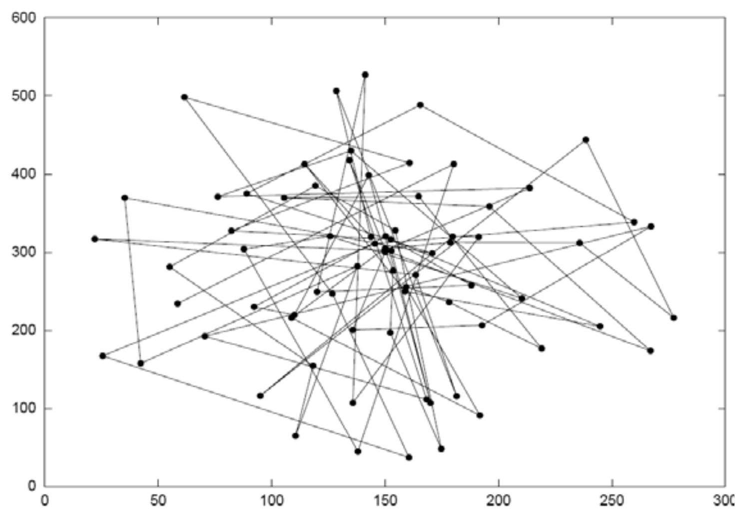


Fig 1-7 Traveling Pattern of Mobile Nodes Using Random Walk Mobility Model

[12]

2) Random Waypoint Mobility Model: It differs from Random Walk Mobility Model since it has pause time between changes in direction and/or speed. At beginning, a mobile node stays in one location for a certain period of time. After the time expires, it chooses a random destination and moves towards there with a speed between $[Minspeed, Maxspeed]$. Upon arrival, it stays for a specified time period before

starting the process again.

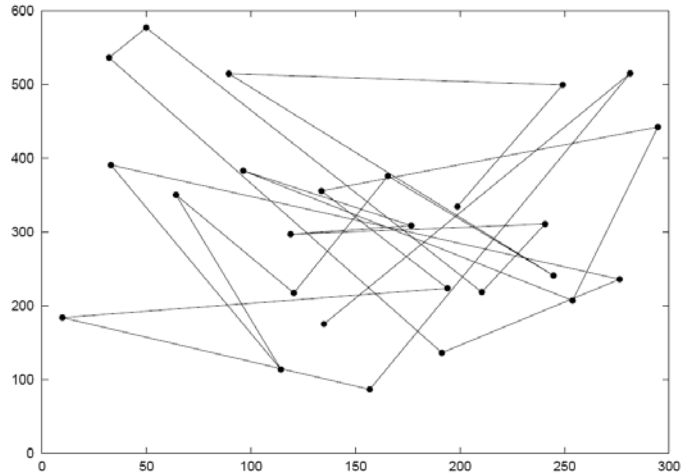


Fig 1-8 Traveling Pattern of Mobile Nodes Using Random Waypoint Mobility Model

[12]

3) Random Direction Mobility Model: In order to overcome the clustering of nodes in one part of simulation area in Random Waypoint mobility model, a mobile node chooses a random direction and moves to the border of the simulation area in this model. After reaching the border of the area, the mobile node pauses a specified time period and chooses a new direction to start this process again.

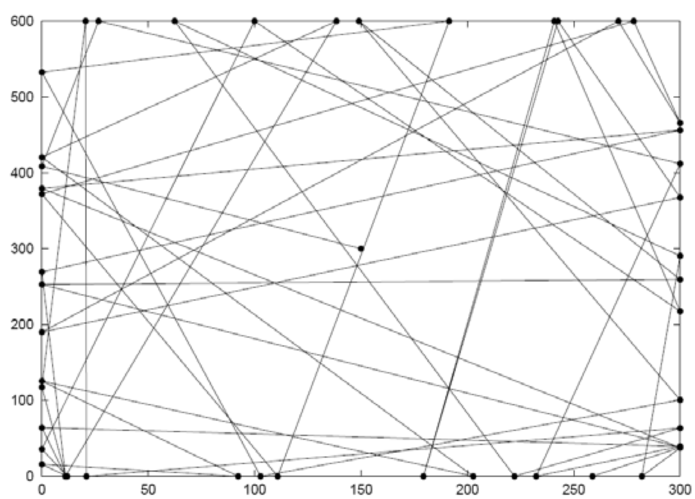


Fig 1-9 Traveling Pattern of Mobile Nodes Using Random Direction Mobility Model [12]

4) Gauss-Markov Mobility Model: This model uses a tuning parameter to vary the degree of randomness in mobility pattern. The value of speed and direction of a mobile node at n^{th} instance is calculated based upon the value of speed and direction at $(n-1)^{st}$ instance, a random variable, and the tuning parameter. The equation is as follows:

$$s_n = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}}$$
$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}}$$

where s_n and d_n are the speed and direction at time interval n; α is the tuning parameter which is between $[0,1]$; $\bar{s}$ and $\bar{d}$ are mean value of speed and direction as $n \to \infty$; $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are random variables from a Gaussian distribution. At time interval n, the position of a mobile node is calculated by the following equations:

$$x_n = x_{n-1} + s_{n-1}\cos d_{n-1}$$

$$y_n = y_{n-1} + s_{n-1}\sin d_{n-1}$$

where (x_n, y_n) and (x_{n-1}, y_{n-1}) are the x and y coordinates of the mobile node position at the n^{th} and $(n-1)^{st}$ time intervals, respectively.

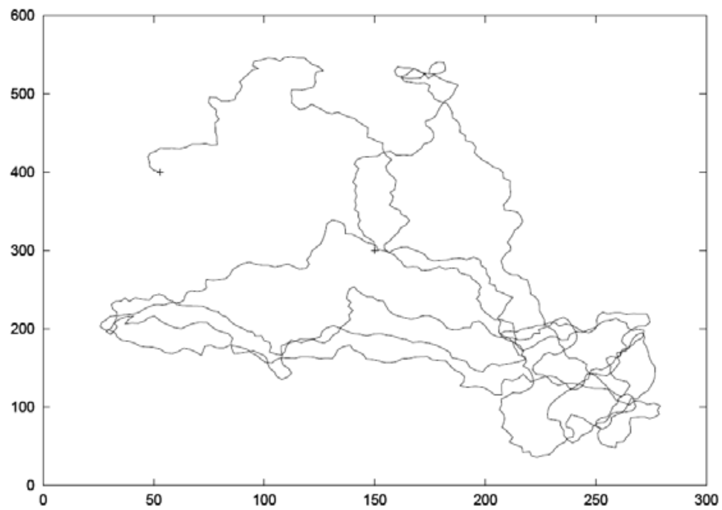


Fig 1-10 Traveling Pattern of Mobile Nodes Using Gauss-Markov Mobility Model [12]

We adopt Gauss-Markov mobility model in our work because it can eliminate the sudden stop and sharp turns encountered in Random Walk mobility model by correlating the current position and speed of a mobile node with its past position and speed. In addition, it can also represent a wide range of user mobility pattern by tuning the randomness parameter α.

1.4 Proposed Approach

In our work, we formulate the multicast scheduling problem as a linear integer programming problem. Next, we use the Lagrangean relaxation method in conjunction with a set of heuristic algorithms to solve the problem.

1.5 Thesis Organization

The thesis is organized as follows. In Chapter 2, the formulation of scheduling problem is presented. In Chapter 3, we introduce the solution approach adopted. In Chapter 4, the set of heuristic algorithms for solving the problem is proposed. The result of computational experiment is shown in Chapter 5. Finally, we present our conclusion as well as the future work in Chapter 6.

Chapter 2 Problem Formulation

2.1 Problem Description

When multiple multicast sources have message to send to their individual multicast members, we construct a multicast tree for each source to reach its multicast group members and schedule the transmission time of the nodes on these multicast trees to avoid collision and minimize the multicast latency, considering the mobility of nodes and the energy consumption of transmission.

Considering the general case below, there are two multicast group 1 and 2. We construct a multicast tree rooted at the multicast source for each multicast group. Some intermediate nodes may be on the multicast tree to help forwarding the message. Therefore, all multicast members will receive the message of the source after all non-leaf nodes on the multicast tree transmit the message.

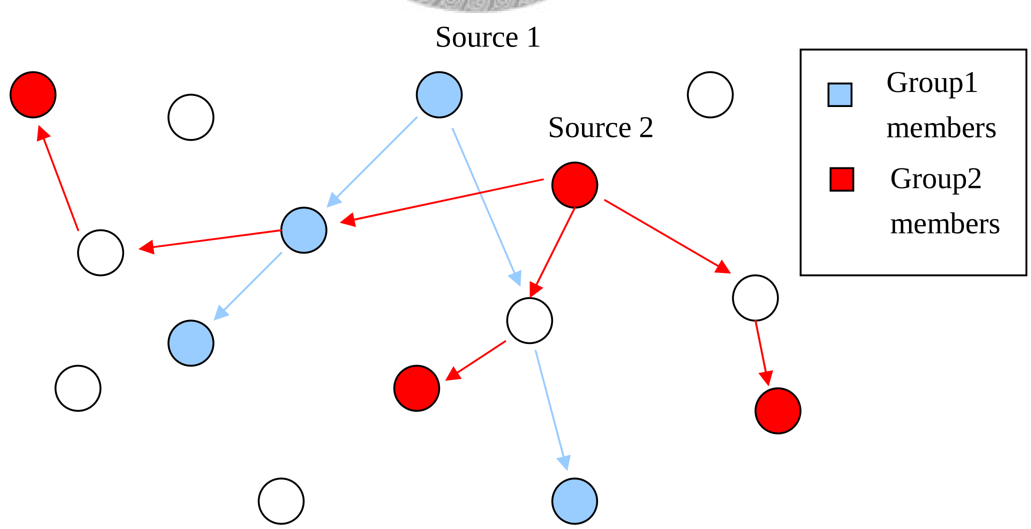


Fig 2-1 The Scenario of Multiple Multicast Tree

In this model, we assume that the transmission time can be divided into discrete time slots. Each non-leaf node on a multicast tree will be assigned to a time slot to transmit the message. Also, the propagation delay of packets is ignored. Therefore, if a node transmits in a time slot, all nodes in its transmission radius will receive the message in the same time slot.

The case below is the scheduling of the transmission time of the non-leaf nodes on two multicast trees. There are two multicast groups, consisting of group A {1, 2, 3} and group B {4, 5, 6, 7}. Every multicast group forms a tree rooted at its source. Every non-leaf node on the multicast tree will transmit the message in a particular time slot after receiving it from the parent node. For example, node 1 of group A transmits the message in time slot 1 first, and node 8 transmits to node 2 and 7 in time slot 2. Finally, node 7 transmits to node 3 in time slot 3, and all members of group A receive the message from source node 1.

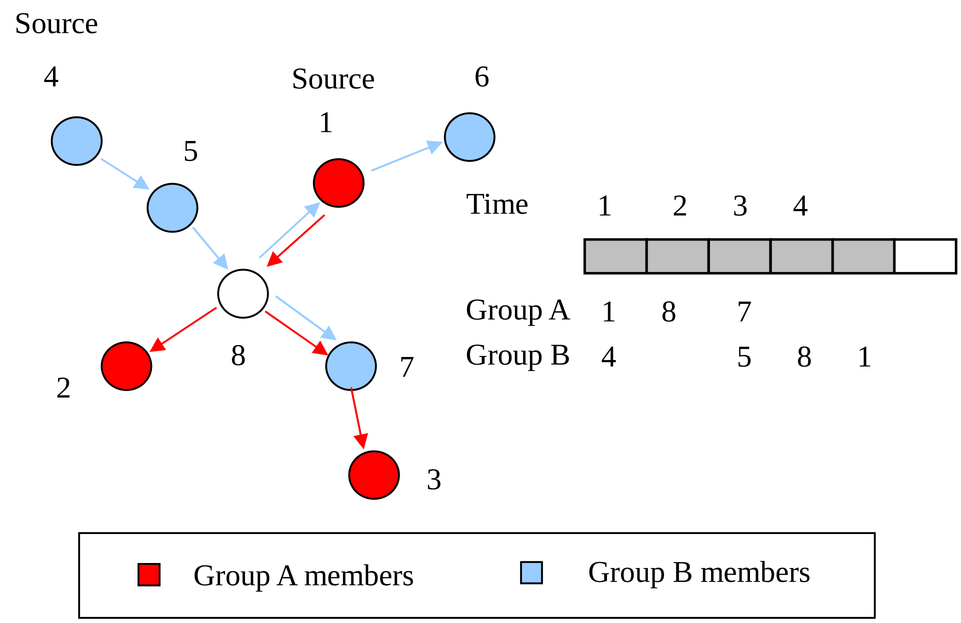


Fig 2-2 The Schedule of Multi-group Multicasting

When scheduling the transmission time, we have to consider the interference between transmissions of nodes. There are two typical scenarios [13]:

1) Exposed terminal problem: When two neighbor nodes transmit at the same time, an outgoing transmission of a node collides with an incoming transmission from it neighbor node.

2) Hidden terminal problem: When two nodes transmit to a common neighbor node at the same time, the two incoming transmission collides at the common neighbor node.

Therefore, we have to schedule the transmission time carefully to avoid the collision of messages in the two scenarios.

Time frame division

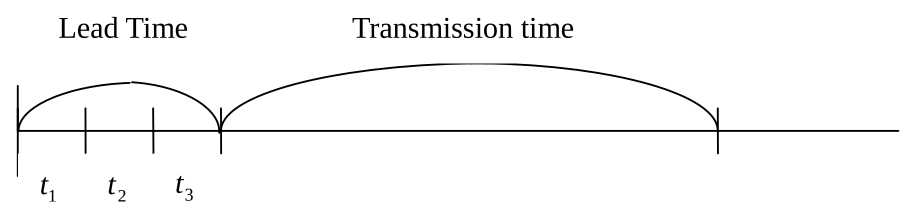


Fig 2-3 Time Frame Division

t_1: Global information collection time

t_2: Scheduling computation time

t_3: Scheduling decision dissemination time

In order to take the mobility of nodes into account, the time is divided into several time frames. We get the location and mobility information of nodes from GPS during t_1. Then using Gauss-Markov mobility model, we could probabilistically predict the oncoming position and velocities of nodes at time $(t_1 + t_2 + t_3)$. We compute the routing assignment for each destination of each source and scheduling transmission time of each node on multicast trees during t_2. The routing and scheduling decision we compute during t_2 is disseminated to all nodes in the network during t_3 and used in the period of transmission time.

Because of the mobility of nodes, the topology of the network will change when time passes. It means the link between two nodes may break at any time. In order to transmit the data of all multicast sources successfully, we want to finish the data transmission on a particular link before the link is disconnected. We use the following function to calculate how long will the link be connected.

Effective link holding time calculation

We adopt the simple method proposed by Lee et al. [14] to predict the duration of time that a link between two neighbor nodes stay connected.

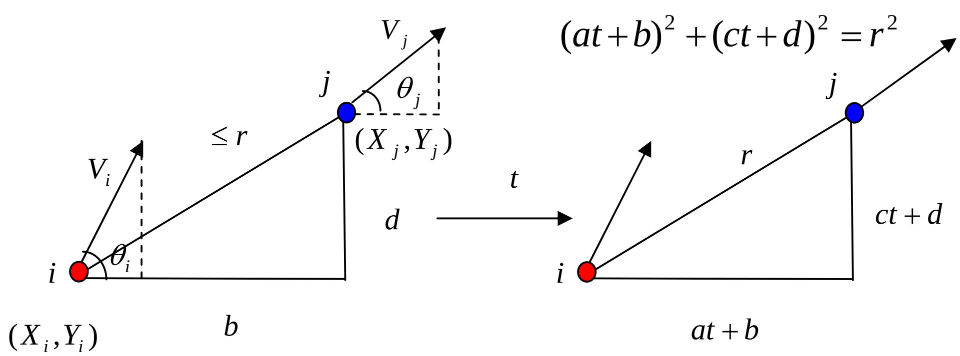


Fig 2-4 Illustration of Link Holding Time t

Assume node j is within the transmission range r of node i. The amount of time that the link (i, j) will stay connected, t, is predicted by

$$t = \frac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2};$$

Where

$a = V_i \cos\theta_i - V_j \cos\theta_j$, the difference of horizontal velocity of node i, j

$b = X_i - X_j$, the difference of horizontal position of node i, j

$c = V_i \sin\theta_i - V_j \sin\theta_j$, the difference of vertical velocity of node i, j

$d = Y_i - Y_j$, the difference of vertical position of node i, j

Note that when $V_i = V_j$ and $\theta_i = \theta_j$, t is set to ∞ without applying the equation.

Energy model

The energy consumed by a transceiver of a wireless node is composed of receiving/processing energy and transmission energy. We assume the transmission energy is much greater than other communication activities, therefore we consider only the transmission energy consumption. Similar to the energy model used in [5], we assume that each node can adjust its power level within a given range. The received signal power is assumed to be equal to $pr^{-\alpha}$, where p is the transmission power, r is the distance and α is a parameter called path loss which is usually between 2 (unobstructed environment) and 4 (urban environment), depending on the characteristics of the communication medium. A simplified interference model is used in which the interference level is independent of network traffic and the same at all nodes. Hence, the transmission power required to support a link between two nodes with distance r is proportional to r^{α} since the received signal power must exceed some threshold. Without loss of generality, we set this threshold equal to 1. As a result,

$p_{ij} =$ power required for the link between node i and node j $= r_{ij}^{\alpha}$,

where r_{ij} is the distance of node i and node j.

The value of α is assumed to be 2 in our work since we are interested in the comparison of different algorithm not the exact value of energy. In addition, based on the "wireless multicast advantage", the total power required for a node to reach a set of its neighbors is simply the maximum required to reach any of them individually.

Therefore, the transmission power of node i is R_i^2 (where R_i is the transmission radius) since the transmission radius of a node is decided by the neighbor with the longest distance. Finally, the total power associated with the multicast trees is the sum of the powers of all transmitting nodes.

Table 2-1 Problem Description

Assumption：

1. There is a centralized node for computing the transmission schedule.
2. We get the location and mobility information of all nodes from GPS.
3. Prediction of velocities and oncoming positions of nodes can be provided by Gauss-Markov mobility model.
4. All nodes in the network have their clocks synchronized.
5. Transmission time can be divided into discrete slots.
6. Propagation delay of packets can be ignored.
7. Each multicast source has single data to send.
8. All nodes have omni-directional antennas.
9. No energy expenditure is involved in signal reception/processing activities.

Given：

1. The set of all nodes in the network
2. The set of multicast sources
3. The set of destinations for each multicast source
4. The set of all candidate paths for each source to reach their destinations
5. The set of possible transmission radius that a node can adopt
6. Maximum transmission energy consumption requirement
7. The transmission power function of each node
8. The effective holding time function of each link
9. The mobility information of nodes in each time slot
10. The set of hop count requirements of each destination on each multicast tree

Objective：

To minimize the total number of time slots used for multiple multicast sources to transmit messages to their individual group members.

Subject to：

1. Routing constraint
   - Each destination should be assigned exactly one path to receive the packet.
   - If a path is selected, all links on this path should be on the multicast tree.
   - The hop count of each path reaching each destination should be smaller than the hop count requirement of the destination.
2. Tree constraint
   - The source node should not have any incoming link.
   - Every destination should have exactly one incoming link.
   - Every intermediate node should have at most one incoming link.
3. Scheduling constraint
   - Every node will not transmit until it receives the message from its parent on the multicast tree.
   - Each non-leaf node on a multicast tree will be assigned to a time slot to transmit the message.
   - The link can be used for transmission only if it is connected.
4. Collision-free constraint
   - If node B is covered by node A, node A should not transmit when node B is communicating with the other node.
   - If the link between node A and node B is on the tree, node A should not transmit when node B is hearing the message from the other node.
5. Power consumption constraint
   - The total transmission power of each node on multicast trees should be smaller than maximum transmission energy consumption requirement.

To determine：

1. Transmission radius of each node
2. Routing paths from each multicast source to its individual destinations
3. The set of multicast trees for each multicast source
4. The transmission time of each non-leaf node on each multicast tree

2.2 Problem Notation

Table 2-2 Notation Descriptions of Given Parameters

Given Parameters	
Notation	**Definition**
V	The set of all nodes in the network
S	The set of multicast sources
D_s	The set of destination nodes for each source s, where $s \in S$
P_{sd}	The set of candidate paths from source s to destination d, where $s \in S, d \in D_s$
$\delta_{p(ij)}$	1 if link (i, j) is on the path p, and 0 otherwise (where $p \in P_{sd}, s \in S, d \in D_s$)
R_i	The set of possible transmission radius of node i, where $i \in V$
P	The maximum transmission energy consumption requirement
H_d	Hop count constraint for each destination d, where $d \in D_s, s \in S$
h_s	The longest hop count distance among the shortest paths which the source s used to reach each destination d, where $s \in S, d \in D_s$
v_i	The velocity of node i, where $i \in V$
θ_i	The moving direction of node i, where $i \in V$
(x_i, y_i)	The position of node i, where $i \in V$
a	The difference of horizontal velocity of node i and j, which is $v_i \cos\theta_i - v_j \cos\theta_j$ (where $i, j \in V$)
b	The difference of horizontal position of node i and j, which is $x_i - x_j$ (where $i, j \in V$)

c	The difference of vertical velocity of node i and j, which is $v_i \sin\theta_i - v_j \sin\theta_j$ (where $i, j \in V$)
d	The difference of vertical position of node i and j, which is $y_i - y_j$ (where $i, j \in V$)
d_{ij}	The distance between node i and node j, where $i, j \in V$
M_1	A large number
M_2	A large number
M_3	A large number
M_4	A large number
M_5	A large number
$\overline{T}$	The index set of time slots, which is large enough for all members of all multicast groups to receive the messages

Table 2-3 Notation Descriptions of Decision Variables

Decision variables	
Notation	**Description**
x_p	1 if source node use path p to reach the destination, and 0 otherwise (where $p \in P_{sd}, s \in S, d \in D_s$)
y_{ijs}	1 if link (i, j) is on the multicast tree rooted at source s, and 0 otherwise (where $i, j \in V, s \in S$)
n_{is}	1 if node i is a non-leaf node on the multicast tree rooted at source s, and 0 otherwise (where $i \in V, s \in S$)
ϕ_{ij}	1 if node j is within the transmission range of node i, and 0 otherwise (where $i, j \in V$)
h_{ijt}	1 if node j hears the transmission of node i in time slot t, and 0 otherwise (where $i, j \in V, t \in \overline{T}$)
r_i	The transmission radius of node i, where $i \in V$
$\Phi_{ij}(r_i)$	The effective link holding time between node i and node j, which is a function of r_i (where $i, j \in V$)
$p(r_i)$	The transmission power of node i, which is a function of r_i (where $i \in V$)
b_{its}	1 if node i transmits the message from source s in time slot t, and 0 otherwise (where $i \in V, s \in S, t \in \overline{T}$)
z_t	1 if any node transmits in time slot t, and 0 otherwise (where $t \in \overline{T}$)

Objective function

$$\min \sum_{t \in T} Z_t \qquad \textbf{(IP)}$$

Subject to:

Routing Constraints

$$\sum_{p \in P_{sd}} x_p = 1 \qquad \forall d \in D_s, s \in S \qquad \textbf{(2.1)}$$

$$\sum_{p \in P_{sd}} x_p \delta_{p(ij)} \le y_{ijs} \qquad \forall i, j \in V, d \in D_s, s \in S \qquad \textbf{(2.2)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_{sd}, d \in D_s, s \in S \qquad \textbf{(2.3)}$$

$$\sum_{i \in V} \sum_{j \in V} \sum_{p \in P_{sd}} x_p \delta_{p(ij)} \le H_d \qquad \forall d \in D_s, s \in S \qquad \textbf{(2.4)}$$

Tree Constraints

$$\sum_{i \in V} y_{ijs} = 0 \qquad \forall j, s \in S, j = s \qquad \textbf{(2.5)}$$

$$\sum_{i \in V} y_{ijs} = 1 \qquad \forall j \in D_s, s \in S \qquad \textbf{(2.6)}$$

$$\sum_{i \in V} y_{ijs} \le 1 \qquad \forall j \in V, s \in S \qquad \textbf{(2.7)}$$

$$\sum_{i \in V} \sum_{j \in V} y_{ijs} \ge \max\{h_s, |D_s|\} \qquad \forall s \in S \qquad \textbf{(2.8)}$$

$$y_{ijs} = 0 \text{ or } 1 \qquad \forall i, j \in V, s \in S \qquad \textbf{(2.9)}$$

$$n_{is} M_1 \ge \sum_{j \in V} y_{ijs} \qquad \forall i \in V, s \in S \qquad \textbf{(2.10)}$$

$$n_{is} \le \sum_{j \in V} y_{ijs} \qquad \forall i \in V, s \in S \qquad \textbf{(2.11)}$$

$$n_{is} = 0 \text{ or } 1 \qquad \forall i \in V, s \in S \qquad \textbf{(2.12)}$$

$$n_{is} = 1 \qquad \forall i, s \in S, i = s \qquad \textbf{(2.13)}$$

$$\frac{r_i - d_{ij}}{M_2} \le \phi_{ij} \qquad \forall i, j \in V \qquad \textbf{(2.14)}$$

$$\phi_{ij} d_{ij} \le r_i \qquad \forall i, j \in V \qquad \textbf{(2.15)}$$

$$y_{ijs} \le \phi_{ij} \qquad \forall i, j \in V, s \in S \qquad \textbf{(2.16)}$$

$$r_s \ne 0 \qquad \forall s \in S \qquad \textbf{(2.17)}$$

$$r_i \in R_i \qquad \forall i \in V \qquad \textbf{(2.18)}$$

$$\sum_{i \in V} \phi_{ij} \ \ge \ 1 \qquad \forall j \in D_s, s \in S \qquad \textbf{(2.19)}$$

$$\phi_{ij} \ = \ 0 \ or \ 1 \qquad \forall i, j \in V \qquad \textbf{(2.20)}$$

Scheduling constraints

$$n_{is} = \sum_{t \in \overline{T}} b_{its} \qquad \forall i \in V, s \in S \qquad \textbf{(2.21)}$$

$$\sum_{t \in \overline{T}} b_{its} = 1 \qquad \forall i, s \in S, i = s \qquad \textbf{(2.22)}$$

$$\sum_{t \in \overline{T}} b_{its} \le 1 \qquad \forall i \in V, s \in S \qquad \textbf{(2.23)}$$

$$\sum_{s \in S} b_{its} \le 1 \qquad \forall i \in V, t \in \overline{T} \qquad \textbf{(2.24)}$$

$$b_{jts} + y_{ijs} \le \sum_{w=1}^{t} b_{iws} - b_{its} + 1 \qquad \forall i, j \in V, s \in S, t \in \overline{T} \qquad \textbf{(2.25)}$$

$$z_t M_3 \ge \sum_{i \in V} \sum_{s \in S} b_{its} \qquad \forall t \in \overline{T} \qquad \textbf{(2.26)}$$

$$\sum_{t \in \overline{T}} z_t \ge \max_{s \in S} \{h_s\} \qquad \textbf{(2.27)}$$

$$t - \Phi_{ij}(r_i) \le M_4 (2 - y_{ijs} - b_{its}) \qquad \forall i, j \in V, s \in S, t \in \overline{T} \qquad \textbf{(2.28)}$$

$$z_t \ = \ 0 \ or \ 1 \qquad \forall t \in \overline{T} \qquad \textbf{(2.29)}$$

$$b_{its} \ = \ 0 \ or \ 1 \qquad \forall i \in V, t \in \overline{T}, s \in S \qquad \textbf{(2.30)}$$

Collision-free constraints

$$b_{its} + \sum_{s' \in S} b_{jts'} \le 2 - y_{ijs} \qquad \forall i, j \in V, t \in \overline{T}, s \in S \qquad \textbf{(2.31)}$$

$$\phi_{ij} + \sum_{s \in S} b_{its} \le h_{ijt} + 1 \qquad \forall i, j \in V, t \in \overline{T} \qquad \textbf{(2.32)}$$

$$\phi_{ij} + \sum_{s \in S} b_{its} \ge 2h_{ijt} \qquad \forall i, j \in V, t \in \overline{T} \qquad \textbf{(2.33)}$$

$$\sum_{i'\in V} h_{i'jt} - 1 \le M_5\left(2 - y_{ijs} - b_{its}\right) \qquad \forall i, j \in V, t \in \overline{T}, s \in S \qquad \textbf{(2.34)}$$

$$h_{ijt} = 0 \, or \, 1 \qquad \forall i, j \in V, t \in \overline{T} \qquad \textbf{(2.35)}$$

Power consumption constraints

$$\sum_{i\in V} p(r_i) \le P \qquad \textbf{(2.36)}$$

Explanation of objective function:

The objective function (IP) is to minimize the total number of time slots used for multiple multicast sources to transmit messages to their individual group members. If the time slots used for transmission are not continuous, we can use a renumber procedure to reorganize the distribution of the time slots used.

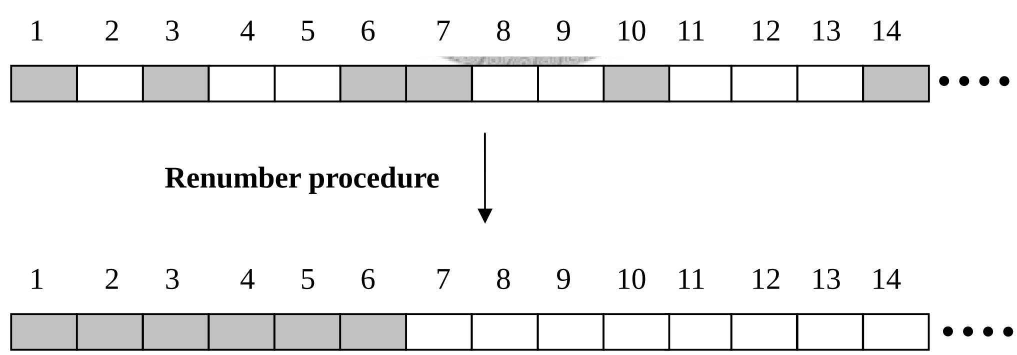


Fig 2-5 Renumber Procedure

By the renumber procedure of time slots, the objective function can imply that the latency of all multicasting will be minimized.

Explanation of constraints:

(1) Routing Constraints

- Constraints (2.1) and (2.3) indicate that for each multicast source, it should choose exactly one path to reach each of their individual destinations to ensure the message will be transmitted to their multicast group members.
- Constraint (2.2) enforces all the links on the path that is selected by the O-D pair should be on the multicast tree. Therefore, the decision variable y_{ijs} should be equal to 1.
- Constraint (2.4) indicates the hop count of the path selected for each destination d should be less than its hop count constraint H_d .

(2) Tree Constraints

- Constraint (2.5) enforces that the incoming link of all multicast sources should be equal to 0.
- Constraint (2.6) enforces that every destination should have exactly one incoming link.
- Constraint (2.7) enforces that all intermediate nodes should have at most one incoming link.
- Constraint (2.8) confines that the total number of links on the multicast tree rooted at source s is at least the maximal value chosen from h_s and the number of destinations D_s.

- Constraint (2.10) and (2.11) jointly model the relationship between n_{is} and $\sum_{j \in V} y_{ijs}$. n_{is} stands for whether the node i is the non-leaf node on the multicast tree rooted at s. If the node i on the multicast tree rooted at s has at least one outgoing link ($\sum_{j \in V} y_{ijs} \geq 1$), n_{is} should be equal to 1 and 0 otherwise. The relationship of them is shown in Table 2-4.

Table 2-4 Explanation of Constraint (2.10) and (2.11)

n_{is}	Constraint (2.10)	Constraint (2.11)
$\sum_{j \in V} y_{ijs} \geq 1$	$n_{is} = 1$	$n_{is} = 0,1$
$\sum_{j \in V} y_{ijs} = 0$	$n_{is} = 0,1$	$n_{is} = 0$

- Constraint (2.13) enforces that the each source s is a non-leaf node on its multicast tree.
- Constraint (2.14) and (2.15) jointly model the relationship between ϕ_{ij}, r_i, and d_{ij} . ϕ_{ij} stands for whether the node j is covered within the transmission radius of the node i. If the transmission radius of the node i is larger than the distance between node i and j ($r_i \geq d_{ij}$), ϕ_{ij} should be equal to 1 and 0 otherwise. The relationship of them is shown in Table 2-5.

Table 2-5 Explanation of Constraint (2.14) and (2.15)

ϕ_{ij}	Constraint (2.14)	Constraint (2.15)
$r_i \geq d_{ij}$	$\phi_{ij} = 1$	$\phi_{ij} = 0,1$
$r_i < d_{ij}$	$\phi_{ij} = 0,1$	$\phi_{ij} = 0$

- Constraint (2.16) indicates that if the node j is not covered within the transmission radius of the node i ($\phi_{ij} = 0$) then y_{ijs} must be 0. In other words, the link (i, j) can be on the multicast tree rooted at s ($y_{ijs} = 1$) only if the node j is covered within the transmission radius of the node i ($\phi_{ij} = 1$).
- Constraint (2.17) indicates that the transmission radius of each multicast source must not be 0.
- Constraint (2.18) indicates that the transmission radius of node i is chosen from its own discrete radius set R_i.
- Constraint (2.19) enforces that all destination should be within at least one node's transmission range.

(3) Scheduling constraints

- Constraint (2.21) indicates that if node i is a non-leaf node on the multicast tree rooted at s ($n_{is} = 1$) then it should transmit the message of source s in a particular time slot ($\sum_{t \in \overline{T}} b_{its} = 1$). Otherwise, $\sum_{t \in \overline{T}} b_{its}$ should be equal to 0.

- Constraint (2.22) enforces that each multicast source should transmit its own data in a particular time slot.
- Constraint (2.23) indicates that each node will transmit the data from a multicast source at most once.
- Constraint (2.24) enforces that every node should transmit at most one message of a particular source s in a time slot.
- Constraint (2.25) indicates that if link (i, j) is on the multicast tree rooted at s, the node j can not transmit the message of source s at time slot t until it receives the message from parent node i. That is, b_{jts} can be equal to 1 only if the node i transmits the message of source s during the interval of time slot 1 to time slot $t-1$ ($\sum_{w=1}^{t} b_{iws} - b_{its} = 1$).
- Constraint (2.26) indicates that if there is at least one node transmitting in time slot t ($\sum_{i \in V} \sum_{s \in S} b_{its} \geq 1$), z_t should be equal to 1 and 0 otherwise.
- Constraint (2.27) confines the total number of time slots used should be at least the longest hop count distance of the shortest path to each destination for all multicast trees.
- Constraint (2.28) indicates that if node i transmits the message of source s to node j in time slot t ($y_{ijs} = 1$ and $b_{its} = 1$), the duration of time slot 1 to t should be smaller than the effective link holding time of link (i, j). It ensures that the link can be used to transmit only if it is connected.

(4) Collision-free constraints

- Constraint (2.31) indicates that if node i is the parent of node j on the multicast tree rooted at s ($y_{ijs} = 1$), the node j can not transmit ($\sum_{s' \in S} b_{jts'} = 0$) to avoid message collision when the parent node i transmit the message of source s to node j in time slot t ($b_{its} = 1$). That is, only one of b_{its} and $\sum_{s' \in S} b_{jts'}$ can be equal to 1.

- Constraint (2.32) and Constraint (2.33) jointly model the relationship between ϕ_{ij}, b_{its}, and h_{ijt}. If node j is within the transmission range of node i ($\phi_{ij} = 1$) and node i transmits the data of any multicast source s in time slot t ($\sum_{s \in S} b_{its} = 1$), it means that node j hears the transmission of node i in this time slot. Therefore, h_{ijt} should be 1 and 0 otherwise. The relationship of them is shown in Table 2-6.

Table 2-6 Explanation of Constraint (2.32) and (2.33)

ϕ_{ij}	$\sum_{s \in S} b_{its}$	Constraint (2.14)	Constraint (2.15)
$\phi_{ij} = 1$	$\sum_{s \in S} b_{its} = 1$	$h_{ijt} = 1$	$h_{ijt} = 0,1$
$\phi_{ij} = 1$	$\sum_{s \in S} b_{its} = 0$	$h_{ijt} = 0,1$	$h_{ijt} = 0$
$\phi_{ij} = 0$	$\sum_{s \in S} b_{its} = 1$	$h_{ijt} = 0,1$	$h_{ijt} = 0$
$\phi_{ij} = 0$	$\sum_{s \in S} b_{its} = 0$	$h_{ijt} = 0,1$	$h_{ijt} = 0$

- Constraint (2.34) indicates that for each node j, when its parent node i on the multicast tree rooted at s ($y_{ijs} = 1$) transmits the data of source s, it should not hear any transmission from other node except node i in the same time slot to avoid collision of messages. Therefore, when the parent node i transmits the message of source s to node j in time slot t ($b_{its} = 1$), $\sum_{i' \in V} h_{i'jt}$ must smaller than 1.

(5) Power consumption constraints

- Constraint (2.36) enforce that the total transmission power of each node should be smaller than maximum transmission energy consumption requirement P.

(6) Boundary constraints

- Constraints (2.3), (2.9), (2.12), (2.20), (2.29), (2.30) and (2.35) are integer constraints of the decision variables.

Chapter 3 Solution Approach

3.1 Introduction to Lagrangean Relaxation Method

The idea of Lagrangean relaxation method has come into being since early 1970s, which was first applied in solving scheduling problems and general integer programming problems [15, 16]. Due to the flexibility and effectiveness, it has become one of the best techniques for solving optimization problems such as integer programming, liner programming with combinatorial objective function, and non-linear programming. In addition, this method has improved the algorithms for a number of important problems in the areas of routing and scheduling.

The fundamental principle of Lagrangean relaxation method is to relax complicated constraints of primal optimization problem and put them into objective function with the corresponding Lagrangean multipliers for each relaxed constraint. Therefore, the primal optimization problem is transformed into a Lagrangean relaxation problem. Next, we can divide the LR problem into several independent sub-problems that is easy to solve. For each sub-problem, we apply some heuristics or well-known algorithms to optimally solve it.

With relaxing procedures addressed above, we loosen the complexity and difficulty of original problem. However, the optimal value of the Lagrangean Relaxation problem is a lower bound (for minimizing problems) on the optimal value of original problem. Therefore, we design a set of heuristics according to the bound to get the primal feasible solution. In order to minimize the gap between the primal problem and the Lagrangean Relaxation problem, we improve the lower bound by using subgradient method to adjust the set of multipliers iteration by iteration.

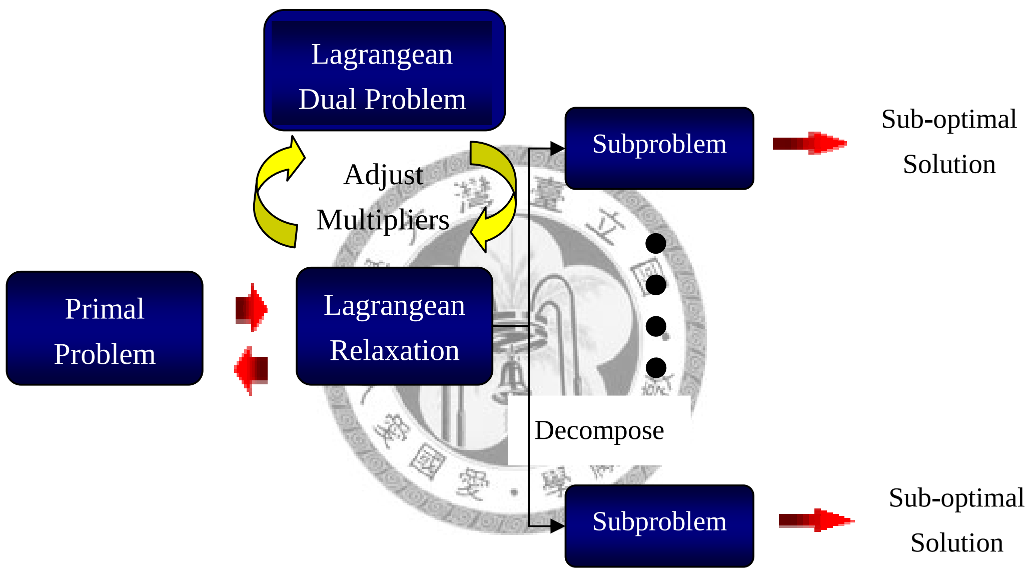


Fig 3-1 Illustration of Lagrangean Relaxation Method

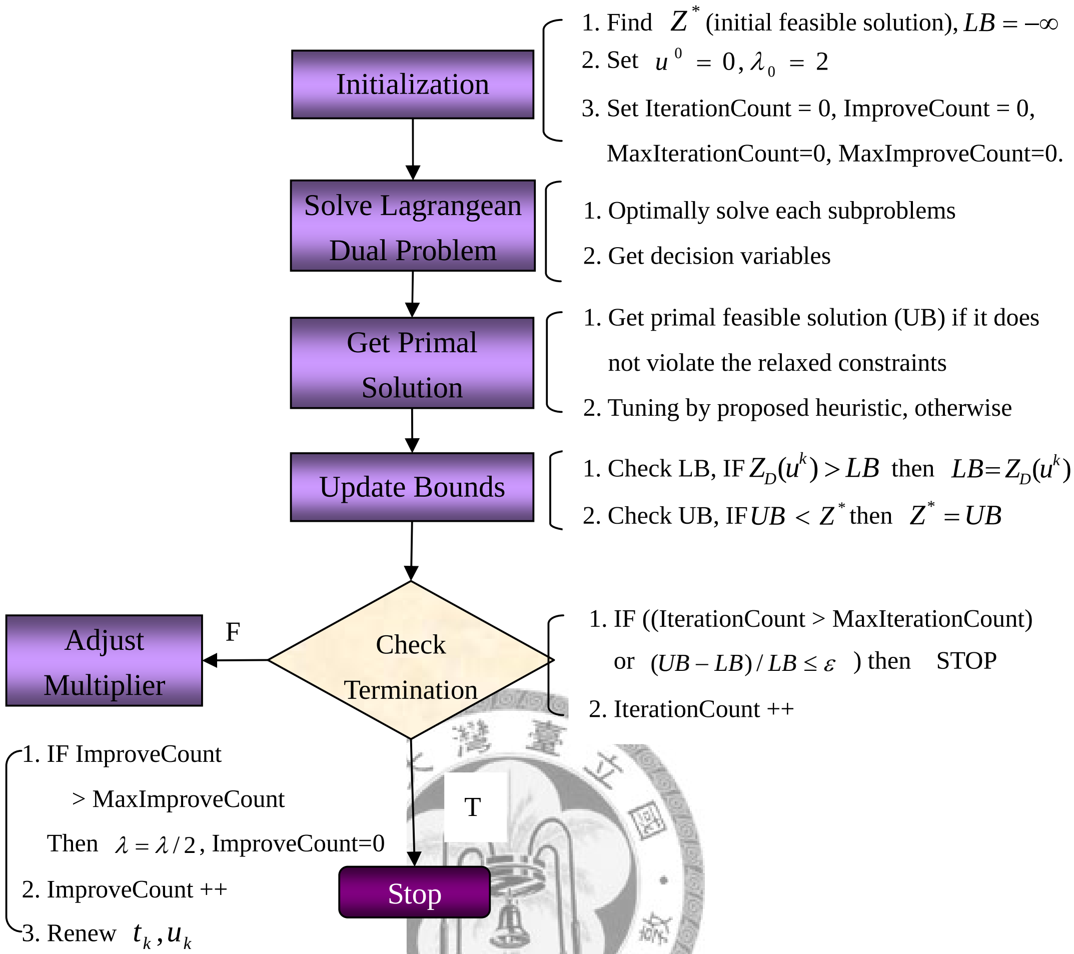


Fig 3-2 Procedures of Lagrangean Relaxation Method

3.2 Lagrangean Relaxation

By introducing Lagrangean multiplier vectors $u^1, \ldots\ldots, u^{16}$, we relax constraints (2.2), (2.10), (2.11), (2.14), (2.15), (2.16), (2.21), (2.24), (2.25), (2.26), (2.28), (2.31), (2.32), (2.33), (2.34), (2.36) in (IP) to obtain the following Lagrangean relaxation problem (LR).

$$
\begin{aligned}
& Z_{LR}(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8,u^9,u^{10},u^{11},u^{12},u^{13},u^{14},u^{15},u^{16}) = \\
& \min \sum_{t\in\overline{T}} z_t \\
& + \sum_{s\in S}\sum_{d\in D_s}\sum_{i\in V}\sum_{j\in V} u^1_{sdij}\Big(\sum_{p\in P_{sd}} x_p \delta_{p(ij)} - y_{ijs}\Big) \\
& + \sum_{i\in V}\sum_{s\in S} u^2_{is}\Big(\sum_{j\in V} y_{ijs} - n_{is} M_1\Big) \\
& + \sum_{i\in V}\sum_{s\in S} u^3_{is}\Big(n_{is} - \sum_{j\in V} y_{ijs}\Big) \\
& + \sum_{i\in V}\sum_{j\in V} u^4_{ij}(r_i - d_{ij} - \phi_{ij} M_2) \\
& + \sum_{i\in V}\sum_{j\in V} u^5_{ij}(\phi_{ij} d_{ij} - r_i) \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{s\in S} u^6_{ijs}(y_{ijs} - \phi_{ij}) \\
& + \sum_{i\in V}\sum_{s\in S} u^7_{is}\Big(n_{is} - \sum_{t\in\overline{T}} b_{its}\Big) \\
& + \sum_{i\in V}\sum_{t\in\overline{T}} u^8_{it}\Big(\sum_{s\in S} b_{its} - 1\Big) \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{s\in S}\sum_{t\in\overline{T}} u^9_{ijst}\Big[b_{jts} + y_{ijs} - \Big(\sum_{w=1}^{t} b_{iws} - b_{its} + 1\Big)\Big] \\
& + \sum_{t\in\overline{T}} u^{10}_t\Big(\sum_{i\in V}\sum_{s\in S} b_{its} - z_t M_3\Big) \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{s\in S}\sum_{t\in\overline{T}} u^{11}_{ijst}[t - \Phi_{ij}(r_i) - M_4(2 - y_{ijs} - b_{its})] \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{s\in S}\sum_{t\in\overline{T}} u^{12}_{ijst}\Big[b_{its} + \sum_{s'\in S} b_{jts'} - (2 - y_{ijs})\Big] \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{t\in\overline{T}} u^{13}_{ijt}\Big[\phi_{ij} + \sum_{s\in S} b_{its} - (h_{ijt} + 1)\Big] \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{t\in\overline{T}} u^{14}_{ijt}\Big[2h_{ijt} - \Big(\phi_{ij} + \sum_{s\in S} b_{its}\Big)\Big] \\
& + \sum_{i\in V}\sum_{j\in V}\sum_{s\in S}\sum_{t\in\overline{T}} u^{15}_{ijst}\Big[\sum_{i'\in V} h_{i'jt} - 1 - M_5(2 - y_{ijs} - b_{its})\Big] \\
& + u^{16}\Big(\sum_{i\in V} p(r_i) - P\Big)
\end{aligned}
$$

(LR)

Subject to:

$$\sum_{p \in P_{sd}} x_p = 1 \qquad \forall d \in D_s, s \in S \qquad \textbf{(3.1)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_{sd}, d \in D_s, s \in S \qquad \textbf{(3.2)}$$

$$\sum_{i \in V} \sum_{j \in V} \sum_{p \in P_{sd}} x_p \delta_{p(ij)} \le H_d \qquad \forall d \in D_s, s \in S \qquad \textbf{(3.3)}$$

$$\sum_{i \in V} y_{ijs} = 0 \qquad \forall j, s \in S, j = s \qquad \textbf{(3.4)}$$

$$\sum_{i \in V} y_{ijs} = 1 \qquad \forall j \in D_s, s \in S \qquad \textbf{(3.5)}$$

$$\sum_{i \in V} y_{ijs} \le 1 \qquad \forall j \in V, s \in S \qquad \textbf{(3.6)}$$

$$\sum_{i \in V} \sum_{j \in V} y_{ijs} \ge \max\{h_s, |D_s|\} \qquad \forall s \in S \qquad \textbf{(3.7)}$$

$$y_{ijs} = 0 \text{ or } 1 \qquad \forall i, j \in V, s \in S \qquad \textbf{(3.8)}$$

$$n_{is} = 1 \qquad \forall i, s \in S, i = s \qquad \textbf{(3.9)}$$

$$n_{is} = 0 \text{ or } 1 \qquad \forall i \in V, s \in S \qquad \textbf{(3.10)}$$

$$\sum_{i \in V} \phi_{ij} \ge 1 \qquad \forall j \in D_s, s \in S \qquad \textbf{(3.11)}$$

$$\phi_{ij} = 0 \text{ or } 1 \qquad \forall i, j \in V \qquad \textbf{(3.12)}$$

$$h_{ijt} = 0 \text{ or } 1 \qquad \forall i, j \in V, t \in \overline{T} \qquad \textbf{(3.13)}$$

$$r_s \ne 0 \qquad \forall s \in S \qquad \textbf{(3.14)}$$

$$r_i \in R_i \qquad \forall i \in V \qquad \textbf{(3.15)}$$

$$\sum_{t \in \overline{T}} b_{its} = 1 \qquad \forall i, s \in S, i = s \qquad \textbf{(3.16)}$$

$$\sum_{t \in \overline{T}} b_{its} \le 1 \qquad \forall i \in V, s \in S \qquad \textbf{(3.17)}$$

$$b_{its} = 0 \text{ or } 1 \qquad \forall i \in V, t \in \overline{T}, s \in S \qquad \textbf{(3.18)}$$

$$\sum_{t \in \overline{T}} z_t \ge \max_{s \in S}\{h_s\} \qquad \textbf{(3.19)}$$

$$z_t = 0 \text{ or } 1 \qquad \forall t \in \overline{T} \qquad \textbf{(3.20)}$$

We decompose (LR) into eight independent subproblems and solve them optimally.

3.2.1 Subproblem 1 (related to decision variable x_p)

$$\min \sum_{s\in S}\sum_{d\in D_s}\sum_{i\in V}\sum_{j\in V}\sum_{p\in P_{sd}} u^1_{sdij} x_p \delta_{p(ij)} \quad \textbf{(SUB 3.1)}$$

Subject to:

$$\sum_{p\in P_{sd}} x_p = 1 \qquad \forall d\in D_s, s\in S \qquad \textbf{(3.1)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_{sd}, d\in D_s, s\in S \qquad \textbf{(3.2)}$$

$$\sum_{i\in V}\sum_{j\in V}\sum_{p\in P_{sd}} x_p \delta_{p(ij)} \le H_d \qquad \forall d\in D_s, s\in S \qquad \textbf{(3.3)}$$

(SUB 3.1) is to compute the shortest path for all multicast source s to all its destinations D_s. It is a shortest path problem with nonnegative link costs u^1_{sdij} and hop count constraints H_d, and can be solved by Bellman-Ford algorithm. The computational complexity (SUB 3.1) is $O\left(|S|\times|V|^3\right)$.

3.2.2 Subproblem 2 (related to decision variable y_{ijs})

$$\min \sum_{i\in V}\sum_{j\in V}\sum_{s\in S}[u^2_{is} - u^3_{is} + u^6_{ijs} - \sum_{d\in D_s} u^1_{sdij} + \sum_{t\in \overline{T}}(u^9_{ijst} + u^{11}_{ijst}M_4 + u^{12}_{ijst} + u^{15}_{ijst}M_5)]y_{ijs}$$
$$+ \sum_{i\in V}\sum_{j\in V}\sum_{s\in S}\sum_{t\in \overline{T}}[-u^9_{ijst} + u^{11}_{ijst}(t-2M_4) - 2u^{12}_{ijst} - u^{15}_{ijst}(1+2M_5)] \quad \textbf{(SUB 3.2)}$$

Subject to:

$$\sum_{i\in V} y_{ijs} = 0 \qquad \forall j, s\in S, j=s \qquad \textbf{(3.4)}$$

$$\sum_{i\in V} y_{ijs} = 1 \qquad \forall j\in D_s, s\in S \qquad \textbf{(3.5)}$$

$$\sum_{i\in V} y_{ijs} \le 1 \qquad \forall j\in V, s\in S \qquad \textbf{(3.6)}$$

$$\sum_{i\in V}\sum_{j\in V} y_{ijs} \ge \max\{h_s, |D_s|\} \qquad \forall s\in S \qquad \textbf{(3.7)}$$

$$y_{ijs} = 0 \text{ or } 1 \qquad \forall i, j\in V, s\in S \qquad \textbf{(3.8)}$$

(SUB 3.2) can be further decomposed into $|S|$ subproblems. For each source s,

$$\min \sum_{i\in V}\sum_{j\in V}\{[u_{is}^2 - u_{is}^3 + u_{ijs}^6 - \sum_{d\in D_s} u_{sdij}^1 + \sum_{t\in T}(u_{ijst}^9 + u_{ijst}^{11} M_4 + u_{ijst}^{12} + u_{ijst}^{15} M_5)] y_{ijs}$$

$$+ \sum_{t\in T}[-u_{ijst}^9 + u_{ijst}^{11}(t - 2M_4) - 2u_{ijst}^{12} - u_{ijst}^{15}(1 + 2M_5)]\} \qquad \textbf{(SUB 3.2.1)}$$

Subject to:

$$\sum_{i\in V} y_{ijs} = 0 \qquad \forall j \in S, j = s$$

$$\sum_{i\in V} y_{ijs} = 1 \qquad \forall j \in D_s$$

$$\sum_{i\in V} y_{ijs} \le 1 \qquad \forall j \in V$$

$$\sum_{i\in V}\sum_{j\in V} y_{ijs} \ge \max\{h_s, |D_s|\}$$

$$y_{ijs} = 0 \ or \ 1 \qquad \forall i, j \in V$$

We propose an algorithm described below to solve each (SUB 3.2.1) optimally:

Step1: We compute the coefficient

$u_{is}^2 - u_{is}^3 + u_{ijs}^6 - \sum_{d\in D_s} u_{sdij}^1 + \sum_{t\in T}(u_{ijst}^9 + u_{ijst}^{11} M_4 + u_{ijst}^{12} + u_{ijst}^{15} M_5)$ for each y_{ijs}.

Step 2: For all node j on multicast tree rooted at source s, we find the link with the smallest coefficient among its all incoming links. In order to satisfy the constraint $\sum_{i\in V} y_{ijs} \le 1$, if the smallest coefficient is negative, set the corresponding y_{ijs} to be 1 and other incoming links y_{ijs} to be 0. Otherwise, set all incoming links y_{ijs} to be 0.

Step 3: In order to satisfy the constraint $\sum_{i\in V} y_{ijs} = 0$, for source s, we need to check whether there is an incoming link to s. If there is any incoming links, set the corresponding y_{ijs} to be 0.

Step 4: In order to satisfy the constraint $\sum_{i\in V} y_{ijs} = 1$, for all destinations D_s of source

s , we need to check whether there is any incoming links to it. If there is no incoming links, find the link with smallest coefficient among its all incoming links and set the corresponding y_{ijs} to be 1.

Step 5: If the total number of y_{ijs} with the value of 1(donate as N) is smaller than $\max\{h_s, |D_s|\}$, we identify the nodes whose incoming links are all 0. From those identified nodes, we choose $\max\{h_s, |D_s|\} - N$ nodes whose incoming link with the smallest coefficient must be the smallest ones among those nodes. Then set the incoming link with the smallest coefficient of each selected node to be 1.

The computational complexity (SUB 3.2) is $O(|S| \times |V|^2)$.

3.2.3 Subproblem 3 (related to decision variable n_{is})

$$\min \sum_{i \in V} \sum_{s \in S} (-u_{is}^2 M_1 + u_{is}^3 + u_{is}^7) n_{is} \quad \textbf{(SUB 3.3)}$$

Subject to:

$$n_{is} = 1 \qquad \forall i, s \in S, i = s \quad \textbf{(3.9)}$$

$$n_{is} = 0 \ or \ 1 \qquad \forall i \in V, s \in S \quad \textbf{(3.10)}$$

(SUB 3.3) can be further decomposed into $|V| \times |S|$ subproblems. For each n_{is},

$$\min(-u_{is}^2 M_1 + u_{is}^3 + u_{is}^7) n_{is} \quad \textbf{(SUB 3.3.1)}$$

Subject to:

$$n_{is} = 0 \ or \ 1$$

For each (SUB 3.3.1), calculate the coefficient $-u_{is}^2 M_1 + u_{is}^3 + u_{is}^7$ for each n_{is}. If the coefficient is negative, we set n_{is} to be 1. Otherwise, n_{is} is set to be 0. Then check whether the corresponding n_{is} of each source s is set to be 1. If not, set the n_{is} to be 1. The computational complexity (SUB 3.3) is $O(|V| \times |S|)$.

3.2.4 Subproblem 4 (related to decision variable ϕ_{ij})

$$\min \sum_{i \in V} \sum_{j \in V} [-u_{ij}^4 M_2 + u_{ij}^5 d_{ij} - \sum_{s \in S} u_{ijs}^6 + \sum_{t \in \bar{T}} (u_{ijt}^{13} - u_{ijt}^{14})] \phi_{ij} \qquad \textbf{(SUB 3.4)}$$

Subject to:

$$\sum_{i \in V} \phi_{ij} \geq 1 \qquad \forall j \in D_s, s \in S \qquad \textbf{(3.11)}$$

$$\phi_{ij} = 0 \; or \; 1 \qquad \forall i, j \in V \qquad \textbf{(3.12)}$$

(SUB 3.4) can be further decomposed into $|V| \times |V|$ subproblems. For each link (i, j),

$$\min[-u_{ij}^4 M_2 + u_{ij}^5 d_{ij} - \sum_{s \in S} u_{ijs}^6 + \sum_{t \in \bar{T}} (u_{ijt}^{13} - u_{ijt}^{14})] \phi_{ij} \qquad \textbf{(SUB 3.4.1)}$$

Subject to:

$$\phi_{ij} = 0 \; or \; 1$$

For each (SUB 3.4.1), we calculate the coefficient $-u_{ij}^4 M_2 + u_{ij}^5 d_{ij} - \sum_{s \in S} u_{ijs}^6 + \sum_{t \in \bar{T}} (u_{ijt}^{13} - u_{ijt}^{14})$ for each link (i, j). If the coefficient of link (i, j) is negative, we set ϕ_{ij} to be 1. Otherwise, ϕ_{ij} is set to be 0. In order to satisfy the constraint (3.11), check the corresponding ϕ_{ij} of all destinations j. If there is no node i which covers the destination j, we choose the one with the smallest coefficient and set the corresponding ϕ_{ij} to be 1.

The computational complexity of (SUB 3.4) is $O(|V|^2)$.

3.2.5 Subproblem 5 (related to decision variable h_{ijt})

$$\min \sum_{i\in V}\sum_{j\in V}\sum_{t\in \overline{T}}[(2u_{ijt}^{14}-u_{ijt}^{13})h_{ijt}+\sum_{s\in S}\sum_{i'\in V}u_{ijst}^{15}h_{i'jt}-u_{ijt}^{13}] \quad \textbf{(SUB 3.5)}$$

Subject to:

$$h_{ijt} = 0\, or\, 1 \qquad \forall i, j \in V, t \in \overline{T} \quad \textbf{(3.13)}$$

In order to solve (SUB 3.5) efficiently, we can rewrite the objective function of the subproblem to make the index notation of decision variable h_{ijt} consistent by transforming.

Transformation:

$$\sum_{i\in V}\sum_{j\in V}\sum_{t\in \overline{T}}[(2u_{ijt}^{14}-u_{ijt}^{13})h_{ijt}+\sum_{s\in S}\sum_{i'\in V}u_{ijst}^{15}h_{i'jt}-u_{ijt}^{13}]$$

$$=\sum_{i\in V}\sum_{j\in V}\sum_{t\in \overline{T}}[(2u_{ijt}^{14}-u_{ijt}^{13})h_{ijt}+\sum_{s\in S}\sum_{i'\in V}u_{i'jst}^{15}h_{ijt}-u_{ijt}^{13}]$$

$$=\sum_{i\in V}\sum_{j\in V}\sum_{t\in \overline{T}}(2u_{ijt}^{14}-u_{ijt}^{13}+\sum_{s\in S}\sum_{i'\in V}u_{i'jst}^{15})h_{ijt}-\sum_{i\in V}\sum_{j\in V}\sum_{t\in \overline{T}}u_{ijt}^{13}$$

After transforming, (SUB 3.5) can be decomposed into $|V|^2 \times |\overline{T}|$ independent subproblems. For each h_{ijt},

$$\min(2u_{ijt}^{14}-u_{ijt}^{13}+\sum_{s\in S}\sum_{i'\in V}u_{i'jst}^{15})h_{ijt} \quad \textbf{(SUB 3.5.1)}$$

Subject to:

$$h_{ijs} = 0\, or\, 1$$

For each (SUB 3.5.1), we calculate the coefficient $(2u_{ijt}^{14}-u_{ijt}^{13}+\sum_{s\in S}\sum_{i'\in V}u_{i'jst}^{15})$ for each h_{ijt}. If the coefficient is negative, we set h_{ijt} to be 1. Otherwise, h_{ijt} is set to be 0. The computational complexity of (SUB 3.5) is $O(|V|^2 \times |\overline{T}|)$.

3.2.6 Subproblem 6 (related to decision variable r_i, $\Phi_{ij}(r_i)$ and $p(r_i)$)

$$\min \sum_{i\in V}\{u^{16}p(r_i)+\sum_{j\in V}[(u_{ij}^4-u_{ij}^5)r_i-u_{ij}^4 d_{ij}-\sum_{s\in S}\sum_{t\in \overline{T}}u_{ijst}^{11}\Phi_{ij}(r_i)]\}-u^{16}P \quad \textbf{(SUB 3.6)}$$

Subject to:

$$r_s \neq 0 \qquad \forall s\in S \qquad \textbf{(3.14)}$$

$$r_i \in R_i \qquad \forall i\in V \qquad \textbf{(3.15)}$$

(SUB 3.6) can be further decomposed into $|V|$ subproblems. For each node i,

$$\min\{u^{16}p(r_i)+\sum_{j\in V}[(u_{ij}^4-u_{ij}^5)r_i-u_{ij}^4 d_{ij}-\sum_{s\in S}\sum_{t\in \overline{T}}u_{ijst}^{11}\Phi_{ij}(r_i)]\} \quad \textbf{(SUB 3.6.1)}$$

Subject to:

$$r_i \in R_i$$

For each (SUB 3.6.1), we calculate the objective function $\{u^{16}p(r_i)+\sum_{j\in V}[(u_{ij}^4-u_{ij}^5)r_i-u_{ij}^4 d_{ij}-\sum_{s\in S}\sum_{t\in \overline{T}}u_{ijst}^{11}\Phi_{ij}(r_i)]\}$ for each possible transmission radius of node i. Then we choose the one which can minimize the objective function to be r_i.

3.2.7 Subproblem 7 (related to decision variable b_{its})

$$\min \sum_{i\in V}\sum_{s\in S}\sum_{t\in \overline{T}}\{[-u_{is}^7+u_{it}^8+u_t^{10}+\sum_{j\in V}(u_{ijst}^9+u_{ijst}^{11}M_4+u_{ijst}^{12}+u_{ijt}^{13}-u_{ijt}^{14}+u_{ijst}^{15}M_5)]b_{its}$$
$$+\sum_{j\in V}u_{ijst}^9 b_{jts}+\sum_{j\in V}\sum_{s'\in S}u_{ijst}^{12}b_{jts'}-\sum_{j\in V}u_{ijst}^9\sum_{w=1}^{t}b_{iws}\} \quad \textbf{(SUB 3.7)}$$

Subject to:

$$\sum_{t\in \overline{T}}b_{its}=1 \qquad \forall i,s\in S, i=s \qquad \textbf{(3.16)}$$

$$\sum_{t\in \overline{T}}b_{its}\le 1 \qquad \forall i\in V, s\in S \qquad \textbf{(3.17)}$$

$$b_{its} = 0 \text{ or } 1 \qquad \forall i\in V, t\in \overline{T}, s\in S \qquad \textbf{(3.18)}$$

In order to solve (SUB 3.7) efficiently, we can rewrite the objective function of the subproblem to make the index notation of decision variable b_{its} consistent by transforming.

Transformation:

$$\sum_{i\in V}\sum_{s\in S}\sum_{t\in \overline{T}}\{[-u_{is}^{7}+u_{it}^{8}+u_{t}^{10}+\sum_{j\in V}(u_{ijst}^{9}+u_{ijst}^{11}M_{4}+u_{ijst}^{12}+u_{ijt}^{13}-u_{ijt}^{14}+u_{ijst}^{15}M_{5})]b_{its}$$

$$+\sum_{j\in V}u_{ijst}^{9}b_{jts}+\sum_{j\in V}\sum_{s'\in S}u_{ijst}^{12}b_{jts'}-\sum_{j\in V}u_{ijst}^{9}\sum_{w=1}^{t}b_{iws}\}$$

$$=\sum_{i\in V}\sum_{s\in S}\sum_{t\in \overline{T}}\{[-u_{is}^{7}+u_{it}^{8}+u_{t}^{10}+\sum_{j\in V}(u_{ijst}^{9}+u_{ijst}^{11}M_{4}+u_{ijst}^{12}+u_{ijt}^{13}-u_{ijt}^{14}+u_{ijst}^{15}M_{5})]b_{its}$$

$$+\sum_{j\in V}u_{jist}^{9}b_{its}+\sum_{j\in V}\sum_{s'\in S}u_{jis't}^{12}b_{its}-\sum_{j\in V}u_{ijst}^{9}\sum_{w=1}^{t}b_{iws}\}$$

$$=\sum_{i\in V}\sum_{s\in S}\sum_{t\in T}\{[-u_{is}^{7}+u_{it}^{8}+u_{t}^{10}+\sum_{j\in V}(u_{ijst}^{9}+u_{ijst}^{11}M_{4}+u_{ijst}^{12}+u_{ijt}^{13}-u_{ijt}^{14}+u_{ijst}^{15}M_{5}+u_{jist}^{9}+\sum_{s'\in S}u_{jis't}^{12}]b_{its}$$

$$-\sum_{j\in V}\sum_{w=1}^{t}u_{ijst}^{9}b_{iws}\}$$

After transforming, (SUB 3.7) can be decomposed into $|V|\times|S|$ independent subproblems. For each node i of each multicast tree s,

$$\min\sum_{t\in \overline{T}}\{[-u_{is}^{7}+u_{it}^{8}+u_{t}^{10}+\sum_{j\in V}(u_{ijst}^{9}+u_{ijst}^{11}M_{4}+u_{ijst}^{12}+u_{ijt}^{13}-u_{ijt}^{14}+u_{ijst}^{15}M_{5}+u_{jist}^{9}+\sum_{s'\in S}u_{jis't}^{12}]b_{its}$$

$$-\sum_{j\in V}\sum_{w=1}^{t}u_{ijst}^{9}b_{iws}\} \qquad \textbf{(SUB 3.7.1)}$$

Subject to:

$b_{its} = 0\ or\ 1 \qquad \forall t\in \overline{T}$

In order to solve (SUB 3.7.1) efficiently, we rewrite the objective function (SUB 3.7.1) into another form. First, let

$$\alpha_{t}=-u_{is}^{7}+u_{it}^{8}+u_{t}^{10}+\sum_{j\in V}(u_{ijst}^{9}+u_{ijst}^{11}M_{4}+u_{ijst}^{12}+u_{ijt}^{13}-u_{ijt}^{14}+u_{ijst}^{15}M_{5}+u_{jist}^{9}+\sum_{s'\in S}u_{jis't}^{12})$$

$$\beta_{jt}=u_{ijst}^{9}$$

The objective function (SUB 3.7.1) can be transformed into $\sum_{t\in\bar{T}}(\alpha_t b_t - \sum_{j\in V}\sum_{w=1}^{t}\beta_{jt}b_w)$.

$$
\begin{aligned}
&\sum_{t\in\bar{T}}(\alpha_t b_t - \sum_{j\in V}\sum_{w=1}^{t}\beta_{jt}b_w) \\
&= \alpha_1 b_1 - \sum_{j\in V}\beta_{j1}b_1 \\
&\quad + \alpha_2 b_2 - \sum_{j\in V}(\beta_{j2}b_1 + \beta_{j2}b_2) \\
&\quad + \alpha_3 b_3 - \sum_{j\in V}(\beta_{j3}b_1 + \beta_{j3}b_2 + \beta_{j3}b_3) \\
&\quad + \cdots + \alpha_{|\bar{T}|}b_{|\bar{T}|} - \sum_{j\in V}(\beta_{j|\bar{T}|}b_1 + \beta_{j|\bar{T}|}b_2 + \beta_{j|\bar{T}|}b_3 + \cdots + \beta_{j|\bar{T}|}b_{|\bar{T}|}) \\
&= [\alpha_1 - \sum_{j\in V}(\beta_{j1} + \beta_{j2} + \beta_{j3} + \cdots + \beta_{j|\bar{T}|})]b_1 \\
&\quad + [\alpha_2 - \sum_{j\in V}(\beta_{j2} + \beta_{j3} + \cdots + \beta_{j|\bar{T}|})]b_2 \\
&\quad + [\alpha_3 - \sum_{j\in V}(\beta_{j3} + \cdots + \beta_{j|\bar{T}|})]b_3 \\
&\quad + \cdots + [\alpha_{|\bar{T}|} - \sum_{j\in V}\beta_{j|\bar{T}|}]b_{|\bar{T}|} \\
&= \sum_{t\in\bar{T}}(\alpha_t - \sum_{j\in V}\sum_{w=t}^{|\bar{T}|}\beta_{jw})b_t
\end{aligned}
$$

For each node i of each multicast tree s, we calculate the coefficient $\alpha_t - \sum_{j\in V}\sum_{w=t}^{|\bar{T}|}\beta_{jw}$ of each time slot t. In order to satisfy the constraint $\sum_{t\in\bar{T}} b_{its} \le 1$, we check the time slot t with the smallest coefficient. If the value is negative, set the corresponding b_{its} to be 1 and other time slots b_{its} to be 0. Otherwise, all time slots b_{its} are set to be 0. In order to satisfy the constraint $\sum_{t\in\bar{T}} b_{its} = 1$ for all source s, set the corresponding b_{its} of the time slot t with the smallest coefficient to be 1 and other time slots b_{its} to be 0.

The computational complexity of (SUB 3.7) is $O(|V| \times |S| \times |\bar{T}|)$.

3.2.8 Subproblem 8 (related to decision variable z_t)

$$\min \sum_{t \in \overline{T}} [(1 - u_t^{10} M_3) z_t - \sum_{i \in V} u_{it}^8] \quad \textbf{(SUB 3.8)}$$

Subject to:

$$\sum_{t \in \overline{T}} z_t \geq \max_{s \in S}\{h_s\} \quad \textbf{(3.19)}$$

$$z_t = 0 \text{ or } 1 \qquad \forall t \in \overline{T} \quad \textbf{(3.20)}$$

(SUB 3.8) can be further decomposed into $|\overline{T}|$ independent subproblems. For each time slot t,

$$\min (1 - u_t^{10} M_3 + u_t^{17}) z_t - \sum_{i \in V} u_{it}^8$$

(SUB 3.8.1)

Subject to:

$$z_t = 0 \text{ or } 1$$

For each (SUB 3.8.1), we calculate the coefficient $1 - u_t^{10} M_3 + u_t^{17}$ for each time slot t. If the coefficient of time slot t is negative, we set z_t to be 1. Otherwise, z_t is set to be 0. In order to satisfy the constraint $\sum_{t \in \overline{T}} z_t \geq \max_{s \in S}\{h_s\}$, if the total number of z_t with the value of 1 (donate as Z) is smaller than $\max_{s \in S}\{h_s\}$, we select ($\max_{s \in S}\{h_s\} - Z$) time slots whose value is 0 and the coefficient is the smallest. Then set the corresponding z_t of the selected time slot to be 1.

The computational complexity of (SUB 3.8) is $O(|\overline{T}|)$.

3.3 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem [15], for any given set of nonnegative multipliers, the optimal value of the corresponding Lagrangean Relaxation problem is a lower bound (for minimizing problems) on the optimal value of primal problem.

Hence, if $u^1_{sdij}, u^2_{is}, u^3_{is}, u^4_{ij}, u^5_{ij}, u^6_{ijs}, u^8_{it}, u^9_{ijst}, u^{10}_t, u^{11}_{ijst}, u^{12}_{ijst}, u^{13}_{ijt}, u^{14}_{ijt}, u^{15}_{ijst}, u^{16} \geq 0$, $Z_{LR}(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8,u^9,u^{10},u^{11},u^{12},u^{13},u^{14},u^{15},u^{16})$ is a lower bound on (IP). We construct the following dual problem (DP) to get the tightest lower bound.

Dual Problem (DP)

$$Z_{DP} = \max Z_{LR}(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8,u^9,u^{10},u^{11},u^{12},u^{13},u^{14},u^{15},u^{16})$$

Subject to:

$$u^1_{sdij}, u^2_{is}, u^3_{is}, u^4_{ij}, u^5_{ij}, u^6_{ijs}, u^8_{it}, u^9_{ijst}, u^{10}_t, u^{11}_{ijst}, u^{12}_{ijst}, u^{13}_{ijt}, u^{14}_{ijt}, u^{15}_{ijst}, u^{16} \geq 0$$

Then, we solve (DP) by subgradient method [17]. In iteration k of the subgradient optimization procedure, the multiplier vector $\pi^k = (u^{1k}, u^{2k}, \ldots, u^{16k})$ is updated by $\pi^{k+1} = \pi^k + t^k g^k$, where g is a subgradient of $Z_{DP}(u^1, u^2, \ldots, u^{16})$. The step size t^k is determined by $\lambda \frac{Z^h_{IP} - Z_{DP}(\pi^k)}{\left\| g^k \right\|^2}$, where Z^h_{IP} is the best primal objective function value found during k iterations (an upper bound on Z_{IP}) and λ is a constant ($0 \leq \lambda \leq 2$).

Chapter 4 Getting Primal Feasible Solution

After optimally solving those subproblems mentioned in previous chapter, a set of multipliers and decision variable is derived. However, the solution provided by the value of these decision variables would not be feasible to the primal problem because many complicated constraints are relaxed. Hence, we must design a series of heuristics to tune the decision variable to ensure the solution will satisfy all of the primal constraints.

In this chapter we propose a four-stage heuristic which takes the multipliers derived in LR problem as a hint for getting good primal feasible solution. Due to the complexity of the primal problem, we divide the process of solving problem into several stages and decide the decision variables respectively.

At first, we decide the routing decision for each multicast group. In order to finishing the transmission before the breaking of the links, the maximal transmission time of each node on each multicast tree is calculated. Then the transmission time of these nodes are arranged according to their maximal transmission time derived previously. In other words, the transmission time of node should be smaller than its maximal transmission time to ensure the connectivity of the link. Finally, when a node transmits according to the transmission schedule, we can calculate the actual transmission radius the node use, which may be smaller than we calculate in routing stage due to the mobility of nodes, to save the energy consumption.

4.1 Heuristic for Routing Policy

In this heuristic, we have three major decision variables to be determined including x_p, y_{ijs}, r_i. Once $\{x_p\}$ are determined, $\{y_{ijs}\}$ and $\{r_i\}$ can be handily derived.

First, we construct a tree for each multicast group. When calculating the multicast tree for source s, set the arc weight to be $1 + coefficientY[i][j][s]$ and run Bellman-Ford algorithm to get the shortest paths to all destinations. The arc weight $coefficientY[i][j][s]$ is $u_{is}^2 - u_{is}^3 + u_{ijs}^6 - \sum_{d \in D_s} u_{sdij}^1 + \sum_{t \in T} (u_{ijst}^9 + u_{ijst}^{11} M_4 + u_{ijst}^{12} + u_{ijst}^{15} M_5)$, which is the coefficient of decision variable y_{ijs} in subproblem 2 and gives us a good reference of arc weight to construct a low-latency multicast tree according to our experiment result.

After constructing the trees, we must check whether the power constraint is violated. The transmission power $p(r_i)$ of node i is defined as r_i^2 and can be derived when $\{r_i\}$ are determined. We show the details of the procedure in Table 4-1.

Table 4-1 Routing Heuristic

For each multicast source s, do the following steps. Step 1：Set the arc weight of each link (i, j) to be $1 + coefficientY[i][j][s]$ and run Bellman-Ford algorithm to calculate the shortest path to each destination. Step 2：If a path from source s to destination d is selected as the shortest path, set the corresponding $\{x_p\}$ to be 1 and $\{y_{ijs}\}$ on this path must be 1. Step 3：The transmission radius of node is determined by the longest distance of the outgoing link whose corresponding $\{y_{ijs}\}$ is 1. Then set the value of $\{r_i\}$ to be the nearest value from $\{R_i\}$. Step 4：Use $\{y_{ijs}\}$ to construct the multicast trees. Step 5：Calculate the total transmission power. If the result does not satisfy the constraint $\sum_{i \in V} p(r_i) \le P$, give up this iteration.

4.2 Heuristic for Calculating Maximal Transmission Time

The structure of multicast tree is not permanent due to the mobility of nodes and the connection time of a link is limited. Hence, the node on the tree must transmit the data before any downstream link is broken, which prevents the children of the node from not receiving the data. The latest time the node must transmit to avoid unfinished transmission of data, which is defined as the maximal transmission time, is determined by the holding time of all links on its subtree and the distance from it to all nodes on its subtree.

We do the calculation for each multicast tree. The maximal transmission time of each node j on the tree rooted at source s (donated as $node[j].\max\_t[s]$) has initial value of $\Phi_{ij}(r_i)+1$ which is the holding time of link between node j and its parent i plus 1. It means that the parent must send out the data before the link holding time $\Phi_{ij}(r_i)$ is passed and the latest time its children receive and send it again is $\Phi_{ij}(r_i)+1$.

Next, we check if there is any node whose maximal transmission time minus the hop distance to source is smaller than 1. The value represents the latest time the source should transmit to guarantee the data reception of the node. If the value is smaller than 1, which means the timely transmission of source is impossible, we enlarge the transmission range of the parent to prolong the link holding time.

Finally, for each node i, when is comes to sending the data to a node j in its downstream before the link connected to node j is broken, it must consider the link holding time as well as the time slot used to transmit the data to node j. Therefore, the final value of the maximal transmission time of node i should be the smallest value of D (which is the link holding time minus the hop distance to node i) among all nodes on its subtree. We show the details of the procedure in Table 4-2.

Table 4-2 Calculating Maximal Transmission Time

For each multicast tree rooted at s, run Step 1－Step 4.

Step 1：If link (i, j) is on this tree, set initial $node[j].\max\_t[s]$ to be $\Phi_{ij}(r_i)+1$.

Step 2：Check the node i on the tree level by level to see whether

$node[i].\max\_t[s]-(node[i].level[s]-1)$ is smaller than 1.

Step 3：If there is any node i whose $node[i].\max\_t[s]-(node[i].level[s]-1)$ is smaller than 1, enlarge the transmission radius of its parent until its $node[i].\max\_t[s]-(node[i].level[s]-1)$ is larger than 1. Then update the decision variables related to r_i such as $p(r_i)$、$\Phi_{ij}(r_i)$ and ϕ_{ij}.

Step 4：For each non-leaf node i, calculate

$node[j].\max\_t[s]-(node[i].level[s]-node[j].level[s])$ of all nodes j in its subtree and choose the smallest one to be the updated value of $node[i].\max\_t[s]$.

Step 5：Calculate the total transmission power again. If the result does not satisfy the constraint $\sum_{i \in V} p(r_i) \le P$, give up this iteration.

4.3 Heuristic for Scheduling Policy

In this heuristic, we schedule the transmission time of all non-leaf nodes on all trees. We try to make the total time slot used for multicasting as short as possible. It means that the nodes tend to transmit in the same time slot if they did not interference each other. On the other hand, some basic rules have to be followed. One is that a node can not transmit the data until its parent sends the data to it. Therefore, the transmission time of a node is larger than that of its parent. Another one is that the transmission time of a node should be smaller than its maximal transmission time we calculate in previous stage to ensure the data can reach all destinations.

We implement this heuristic using the data structure of array. An array stores all non-leaf nodes of all trees, which is sorted by maximal transmission time of node first and by $coeffcientB[i][s]$ later. The $coeffcientB[i][s]$ is the coefficient of decision variable b_{its} in subproblem 7, which is the value of $\sum_{t \in \overline{T}} (\alpha_t - \sum_{j \in V} \sum_{w=t}^{|\overline{T}|} \beta_{jw})$. Then each node has a flag which represents the transmission status of the node. The initial value of flag is 0, which means it does not receive the data yet. The transmission of its parent can change the value of its flag from 0 to 1, which means it has received the data and can send it out. When the node finishes its transmission, it changes the flag from 1 to 2. For each time slot t, we tend to let the node with smaller value of maximal transmission time transmit as soon as possible. We proceed with this process until all non-leaf nodes have sent. We show the details of the procedure in Table 4-3.

Table 4-3 Scheduling Heuristic

Step 1：	Put all non-leaf nodes of all multicast trees into an array. Sort this array into increasing order by $node[i].\max\_t[s]$ of each non-leaf node.
Step 2：	If there are many nodes whose $node[i].\max\_t[s]$ are the same value, sort these node again into increasing order by $coeffcientB[i][s]$.
Step 3：	Each non-leaf node has a status flag. When the flag is 1, it means the node has received the message before and can send the message. If a node has transmitted before, its flag is 2. Otherwise, the flag is set to be 0. First, the flag of each source node has its initial value 1, and the flag of other non-leaf nodes is 0.
Step 4：	From time slot $t = 1$ to T, repeat Step 5 and Step 6 until all non-leaf nodes have sent their messages.
Step 5：	Choose the first node i whose flag is 1 in the array and check all nodes with flag 1 to see if they can transmit with node i simultaneously without any collision.
Step 6：	For node i and the set of nodes which do not interfere with node i, set their transmission time to be this time slot t and flag to be 2. Then change the flags of their children in this array from 0 to 1.
Step 7：	Set $\{b_{its}\}$ according to the transmission time of each node on each tree. Also,

set the corresponding $\{z_t\}$ to be 1 if there is any node transmitting in time slot t.

4.4 Heuristic for Saving Energy Consumption

The transmission radius derived in routing stage is a conservative estimate due to the mobility of nodes. When a node executes the transmission in a particular time slot according to the transmission schedule produced in scheduling stage, its children may be closer to it than that at the beginning. Therefore, the node can adapt smaller transmission range to reach all of its children. In this heuristic based upon this concept, each node would adjust its transmission radius by tracing the position of each child to save the energy consumption of transmission. We show the details of the procedure in Table 4-4.

Table 4-4 Heuristic of Saving Energy Consumption

For each time slot t, execute Step1－Step2.

Step 1：For all nodes transmitting in time slot t, calculate the position of all children in this time slot.

Step 2：Update $\{r_i\}$ to be the nearest value chosen from $\{R_i\}$, which can reach all of its children.

Step 3：Update the total transmission power $\sum_{i \in V} p(r_i)$.

4.5 Lagrangean Relaxation Based Algorithm

Solving the Lagrangean Relaxation problem optimally as described in Chapter 3 is to get a lower bound (LB) of the primal problem while the primal feasible solutions derived by the four-stage heuristic in previous section in this chapter is a upper bound (UB) of the primal problem. In each iteration, by solving both Lagrangean dual problem and the four-stage heuristic, we get the LB and UB respectively. The gap between UB and LB, computed by (UB-LB)/LB, provides the optimality of primal feasible solution. The smaller gap is computed, the better solution is derived. Next, we adopt a subgradient method to update Lagrangean multipliers which is used in next iteration. The complete Lagrangean Relaxation based algorithm is composed of the following steps.

Table 4-5 Lagrangean Relaxation Based Algorithm

```
Initialize the value of Lagrangean multipliers vectors (μ^1, μ^2,...., μ^16) to be 0.
UB := 0 , LB := - Infinity;
Max_Iteration_Number := 5000;
Max_improve_counter := 90;
improve_counter := 0;
step_size_coefficient := 10;
For iteration := 1 to Max_Iteration_Number do
Begin
    solve subproblem1 (Sub 1)
```

```
        solve subproblem2 (Sub 2)
        solve subproblem3 (Sub 3)
        solve subproblem4 (Sub 4)
        solve subproblem5 (Sub 5)
        solve subproblem6 (Sub 6)
        solve subproblem7 (Sub 7)
        solve subproblem8 (Sub 8)
        Calculate Zdu which is the value of dual problem.
        If Zdu > LB then
            LB := Zdu;
            improve_counter :=0;
        Run the Primal Heuristic to get the value of Zip.
        If Zip < UB then
            UB := Zip;
        If improve_counter  ≥  Max_improve_counter then
            step_size_coefficient := step_size_coefficient / 2;
            improve_counter :=0;
        improve_counter := improve_counter +1;
        iteration := iteration +1;
        Update the step size and adjust multipliers according to subgradient method.
    End;
```

Chapter 5 Computational Experiments

In this chapter, we conduct a series of computational experiment to test the solution quality of our heuristic for getting primal feasible solution. In the mean time, we implement three simple algorithms for comparison.

5.1 Experiment Environment

The computational experiments program is developed in C, and we use a Pentium M 1.86GHz, 1.5GB, Windows XP Service Pack 2 as our test platform. Table 5-1 shows the experiment parameters and test platform.

Table 5-1 Experiment Environment and Parameters

Parameter	Value
Network area	200*200
Number of nodes	10~80
Number of source nodes	1~8
Density of destination nodes	10%~100%
Transmission radius	30~120
Number of iteration	5000
Improvement counter	90
Initial value of multipliers	0
Test Platform	CPU : Intel® Pentium® M processor 1.86GHz RAM :1.5GB OS : Microsoft Windows XP with SP2
Development Tool	Code::Blocks 8.02

5.2 Simple Algorithm for Comparison and Performance Metrics

We implement three simple algorithms for comparison. The first one is Shortest Path Tree Algorithm, in which we construct a shortest path tree for each multicast group in routing stage and the same scheduling policy as ours. We donate this method as Simple Algorithm 1 (SA1). The second one is multiple source broadcast (MSB) proposed by Gandhi et al. [10] which is design for broadcasting in a static wireless ad hoc network. We compare it with our heuristic in such an extreme scenario and donate it as Simple Algorithm 2 (SA2). The last one is Bread First Search Tree Algorithm, in which we construct a breadth first search tree for each multicast group in routing stage and the same scheduling policy as ours. We donate this method as Simple Algorithm 3 (SA3).

In addition, we donate the best solution of our dual problem as LB, and the best solution of our Lagrangean based heuristic as LR. The solution derived by simple algorithm is donated as SA. We consider two performance metrics to evaluate our solution quality, including "Gap" and "Improvement Ratio" Gap is calculated by $\frac{LR-LB}{LB}\times 100\%$, and Improvement Ratio is calculated by $\frac{SA-LR}{LR}\times 100\%$ which is donated as $I.R.$ in the table of experiment result.

5.3 Experiment Scenario

In the following experiment scenarios, we place nodes at random within a 200*200 square, and choose the source and destination nodes also in a random manner. We divide the scenarios into static and dynamic network environments. In a static network, the position of nodes is fixed while in a dynamic network, nodes move with a randomly chosen speed in a random direction. We conduct our heuristic in both two scenarios. In addition, we test our heuristic both in multicasting and broadcasting since multicasting is a general case of broadcasting.

5.3.1 Static Network with Different Number of Nodes

At first, ten nodes are placed within the square, in which three sources has data to broadcast to all nodes. In each case, ten extra nodes are added in the network at a time. The following table shows the experiment results as the network size increases.

Table 5-2 Parameters of Static Network with Different Number of Nodes

Number of Source Nodes	Density of Destination Nodes	Transmission Radius
3	100%	50~90

Table 5-3 Experiment Result of Static Network with Different Number of Nodes

Number of Nodes	LB	LR	Gap (%)	SA1	I.R. to SA1 (%)	SA2	I.R. to SA2 (%)	SA3	I.R. to SA3 (%)
10	3	6	100	6	0	8	33.33333	7	16.66667
20	3	5	66.66666667	7	40	13	160	7	40
30	3	4	33.33333333	7	75	14	250	6	50
40	3	4	33.33333333	7	75	14	250	6	50
50	3	4	33.33333333	7	75	15	275	6	50
60	3	4	33.33333333	7	75	16	300	6	50
70	3	4	33.33333333	7	75	16	300	5	25

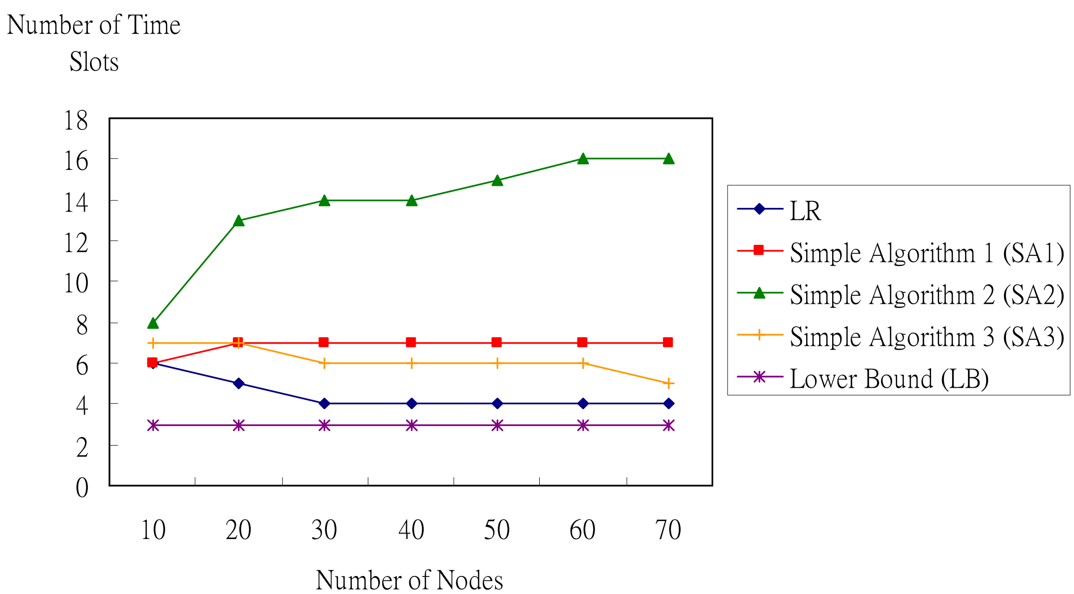


Fig 5-1 Experiment Result of Static Network with Different Number of Nodes

With the fixed multicast groups, it is more possible to reduce the latency as the number of nodes in the network increases. When we add ten nodes in the network, a good algorithm should find better multicast trees which can lead to less delay or the same trees as the ones before adding extra nodes. It should not result in larger latency. Otherwise, the algorithm has poor solution quality. According to the result above, SA1 and SA2 is the algorithm with poor solution quality we describe. The delay they compute is become longer as the nodes increases especially for SA2. SA3 and our algorithm can explore better solution as the network size increases, but the delay derived by our algorithm is shorter.

We run another case with bigger problem size in order to test the scalability of our algorithm. At first, twenty nodes are placed within the square, in which five sources has data to broadcast to all nodes. In each case, ten extra nodes are added in the network at a time. The following table shows the experiment results as the network size increases.

Table 5-4 Parameters of Static Network with Different Number of Nodes

Number of Source Nodes	Density of Destination Nodes	Transmission Radius
5	100%	50~90

Table 5-5 Experiment Result of Static Network with Different Number of Nodes

Number Of Nodes	LB	LR	Gap (%)	SA1	I.R. to SA1 (%)	SA2	I.R. to SA2 (%)	SA3	I.R. to SA3 (%)
20	5	9	80	16	77.77777778	21	133.3333333	18	100
30	5	8	60	17	112.5	24	200	15	87.5
40	5	8	60	16	100	22	175	15	87.5
50	5	6	20	16	166.6666667	23	283.3333333	15	150
60	5	8	60	11	37.5	25	212.5	15	87.5
70	5	7	40	12	71.42857143	26	271.4285714	15	114.2857
80	5	8	60	12	50	27	237.5	15	87.5

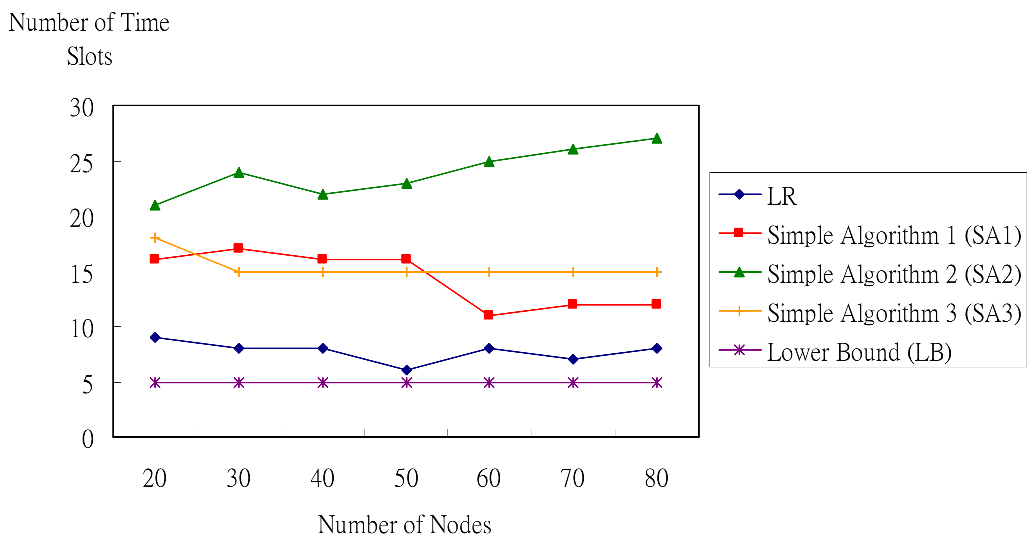


Fig 5-2 Experiment Result of Static Network with Different Number of Nodes

5.3.2 Static Network with Different Number of Source Nodes

Table 5-6 Parameters of Static Network with Different Number of Source Nodes

Number of Nodes	Density of Destination Nodes	Transmission Radius
60	100%	50~60

Table 5-7 Experiment Result of Static Network with Different Number of Source Nodes

Number of Source Nodes	LB	LR	Gap (%)	SA1	I.R. to SA1 (%)	SA2	I.R. to SA2 (%)	SA3	I.R. to SA3 (%)
1	7	7	0	8	14.28571429	9	28.57142857	7	0
2	7	7	0	12	71.42857143	12	71.42857143	7	0
3	7	10	42.85714	15	50	13	30	10	0
4	7	10	42.85714	16	60	18	80	16	60
5	7	10	42.85714	16	60	18	80	19	90
6	7	13	85.71429	19	46.15384615	18	38.46153846	21	61.53846154
7	7	13	85.71429	29	123.0769231	22	69.23076923	22	69.23076923
8	7	14	100	32	128.5714286	23	64.28571429	25	78.57142857

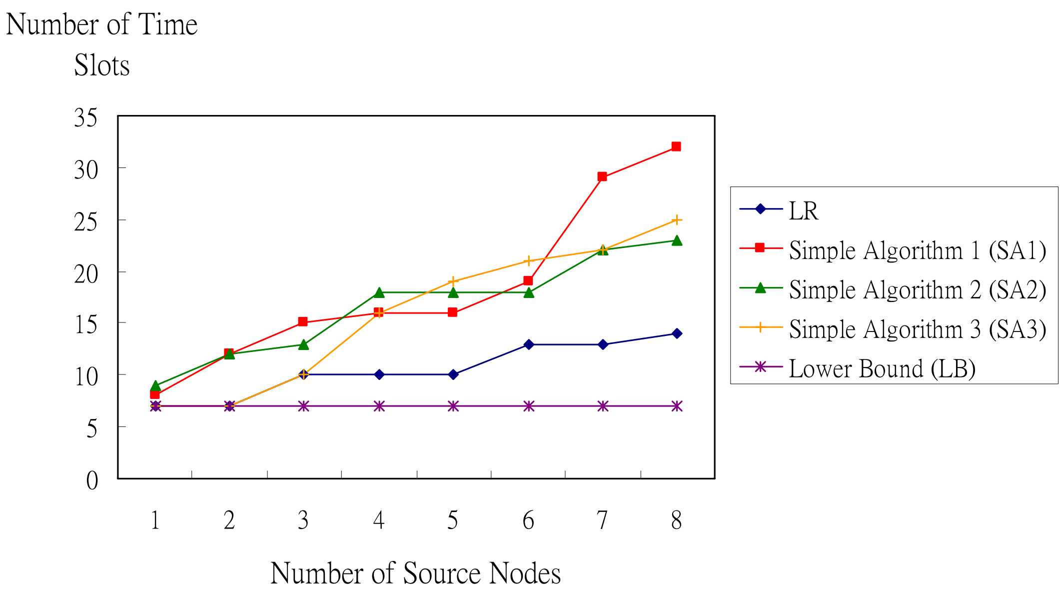


Fig 5-3 Experiment Result of Static Network with Different Number of Source Nodes

As the result shown above, total number of time slots used for transmission will increase as the number of multicast group increases. The result is straightforward since more multicast groups introduce more possible transmission collision which would lead to longer delay to complete entire transmission requests. Our heuristic has better performance than three simple algorithms and our improvement ratio to them is bigger when the number of multicast group is more than 6.

5.3.3 Dynamic Network with Different Density of Destination Nodes

We studied the effect of varying the density of destination nodes on the total number of time slot used. In this scenario, five multicast groups are randomly chosen, and we vary the density of destination nodes of each multicast tree from 10% to 90% of total number of nodes.

Table 5-8 Parameters of Dynamic Network with Different Density of Destination Nodes

Number of Nodes	Number of Source Nodes	Transmission Radius
40	5	50~70

Table 5-9 Experiment Result of Dynamic Network with Different Density of Destination Nodes

Density of destination nodes	Lower Bound (LB)	LR	Gap (%)	Simple Algorithm 1 (SA1)	Improvement Ratio to SA1 (%)
10%	5	6	20	8	33.33333333
20%	6	6	0	10	66.66666667
30%	7	7	0	9	28.57142857
40%	7	7	0	9	28.57142857
50%	7	7	0	9	28.57142857
60%	7	7	0	9	28.57142857
70%	7	7	0	9	28.57142857
80%	7	7	0	10	42.85714286
90%	7	7	0	10	42.85714286

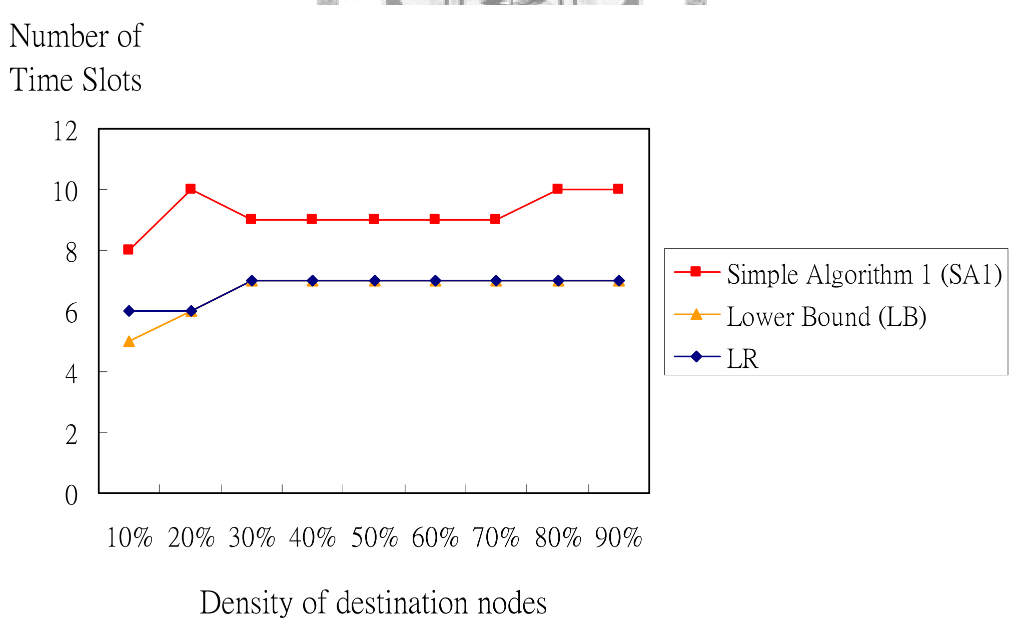


Fig 5-4 Experiment Result of Dynamic Network with Different Density of Destination Nodes

Table 5-10 Energy Consumption of Dynamic Network with Different Density of

Destination Nodes

Density of Destination Nodes	LR	Simple Algorithm 1 (SA1)
10%	46558	60085
20%	57006	69693
30%	70477	77947
40%	73373	81712
50%	79551	82776
60%	79707	84082
70%	80454	85487
80%	82571	87431
90%	82403	87431

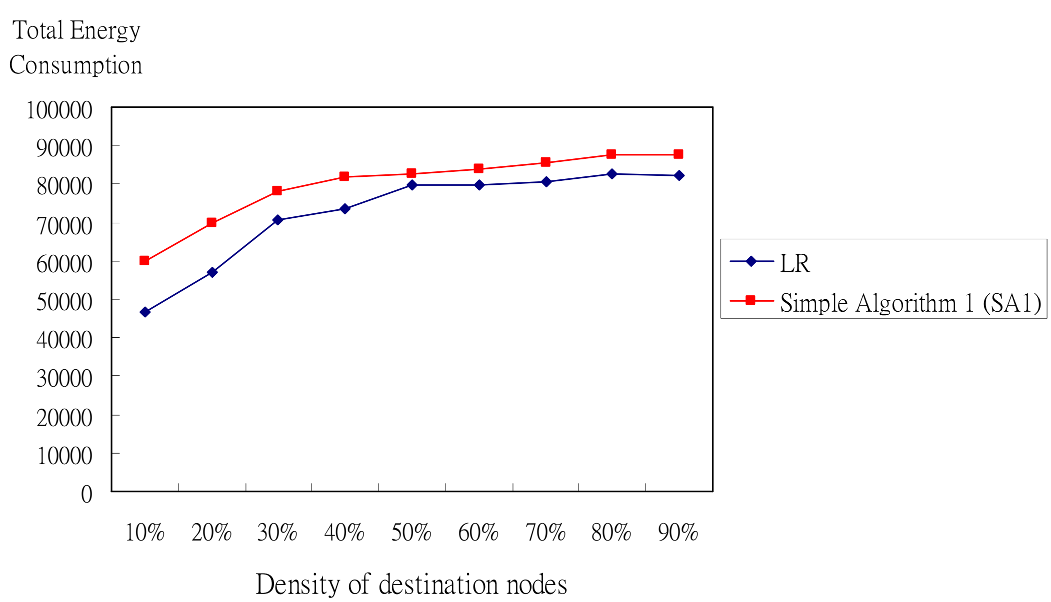


Fig 5-5 Energy Consumption of Dynamic Network with Different Density of Destination Nodes

The result indicates that the density of destination nodes does not make a great impact on latency. According to the plot of our algorithm, the latency remains the same as the density of destinations nodes is more than 30%. The reason is that the major factor which influences the latency mostly is the depth of the multicast tree. While the density of destination nodes become higher enough, the non-leaf nodes on the multicast trees constructed has more outgoing links which can reach more destinations, and the depth of multicast tree remains stable. Consequently, the value of the total number of time slots will tend to converge.

We also use another performance metric "Total Energy Consumption", which is the energy consumption to complete the transmission request of all multicast groups, to evaluate our algorithm. The result shows that to transmit data to more destinations will lead to the increase of power since some non-leaf nodes may use bigger transmission radius in order to reach more destinations which are far from it. Note that our algorithm not only use less time slots but also consume less energy to complete the transmission. That is because the multicast tree we construct has few non-leaf nodes, which will introduce less power for transmission.

5.3.4 Static Network with Different Transmission Radius

We assign the transmission radius of each node at random. The transmission radius we discussed here represents the maximal transmission radius a node can adopt. A node may choose not to turn on if it is not the intermediate nodes from a multicast source to destinations. Therefore, the transmission radius of this node is set to be 0. In this experiment, we increase the transmission radius 5 units at a time to observe the impact on delay.

Table 5-11 Parameters of Static Network with Different Transmission Radius

Number of Nodes	**Number of Source Nodes**	**Density of Destination Nodes**
60	3	100%

Table 5-12 Experiment Result of Static Network with Different Transmission Radius

Transmission Radius	**LB**	**LR**	**Gap (%)**	**SA1**	**I.R. to SA1 (%)**	**SA2**	**I.R. to SA2 (%)**	**SA3**	**I.R. to SA3 (%)**
50~60	7	10	42.85714	15	50	13	30	10	0
55~65	6	7	16.66667	16	128.5714286	15	114.2857143	10	42.85714
60~70	5	8	60	16	100	16	100	9	12.5
65~75	5	9	80	17	88.88888889	11	22.22222222	13	44.44444
70~80	5	7	40	18	157.1428571	15	114.2857143	14	100
75~85	5	6	20	15	150	16	166.6666667	12	100
80~90	4	6	50	15	150	15	150	12	100

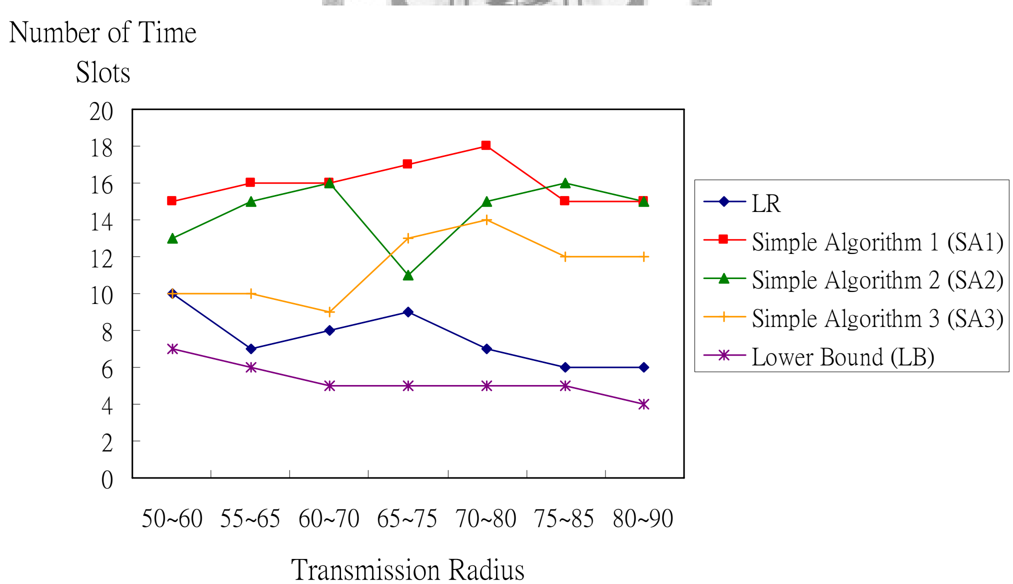


Fig 5-6 Experiment Result of Static Network with Different Transmission Radius

Table 5-13 Energy Consumption of Static Network with Different Transmission Radius

Transmission Radius	LR	Simple Algorithm 1	Simple Algorithm 3
50~60	85845	87605	89777
55~65	101137	112099	102563
60~70	110328	121061	110602
65~75	147010	133762	139259
70~80	129889	147557	129625
75~85	142716	163494	143268
80~90	154689	179425	138698

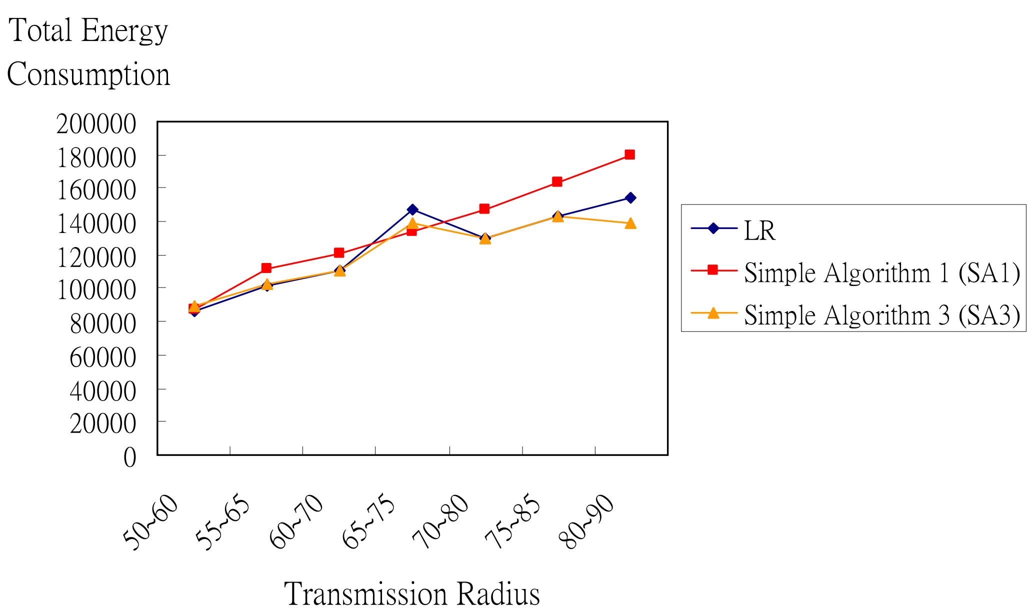


Fig 5-7 Energy Consumption of Static Network with Different Transmission Radius

In this experiment, we observe the effect of enlarging transmission radius to the latency. For a single multicast tree, a larger transmission range a node adopts can reach more destinations and shorten the depth of multicast trees, which can decrease the number of time slots. However, in the environment of multiple multicast groups, the bigger coverage of transmission can make two non-leaf nodes have common children, which can introduce more possible interference. In order to avoid such collision, the nodes can not transmit simultaneously, and consequently it takes longer time to finish the transmission.

The result shows the interaction of these two factors to the latency. Although the latency may increase slightly in our algorithm as we adjust the transmission radius a small unit at a time, the latency has the trend of dropping down as the transmission radius become larger. Compared to our algorithm, the plots of SA1, SA2 and SA3 display an unstable solution and higher latency.

Note the energy consumption will increase as the adoptable transmission range become larger, due to the non-leaf nodes try to reach all children by adopting larger radius which will consume more energy. The performance of our algorithm is better than SA1 in most cases since the total number of non-leaf nodes on all multicast trees we construct is less than the one constructed by SA1. More Non-leaf nodes which are responsible for transmission will potentially consume more energy.

5.3.5 Dynamic Network with Different Transmission Radius

Table 5-14 Parameters of Dynamic Network with Different Transmission Radius

Number of Nodes	Number of Source Nodes	Density of Destination Nodes
40	4	50%

Table 5-15 Experiment Result of Dynamic Network with Different Transmission Radius

Transmission Radius	Lower Bound (LB)	LR	Gap (%)	SA1	I.R. to SA1 (%)	SA3	I.R. to SA3 (%)
30~70	7	11	57.14286	13	18.18181818	14	27.27272727
40~80	6	8	33.33333	10	25	12	50
50~90	5	8	60	9	12.5	12	50
60~100	5	8	60	12	50	13	62.5
70~110	4	8	100	10	25	14	75
80~120	4	6	50	10	66.66666667	10	66.66666667

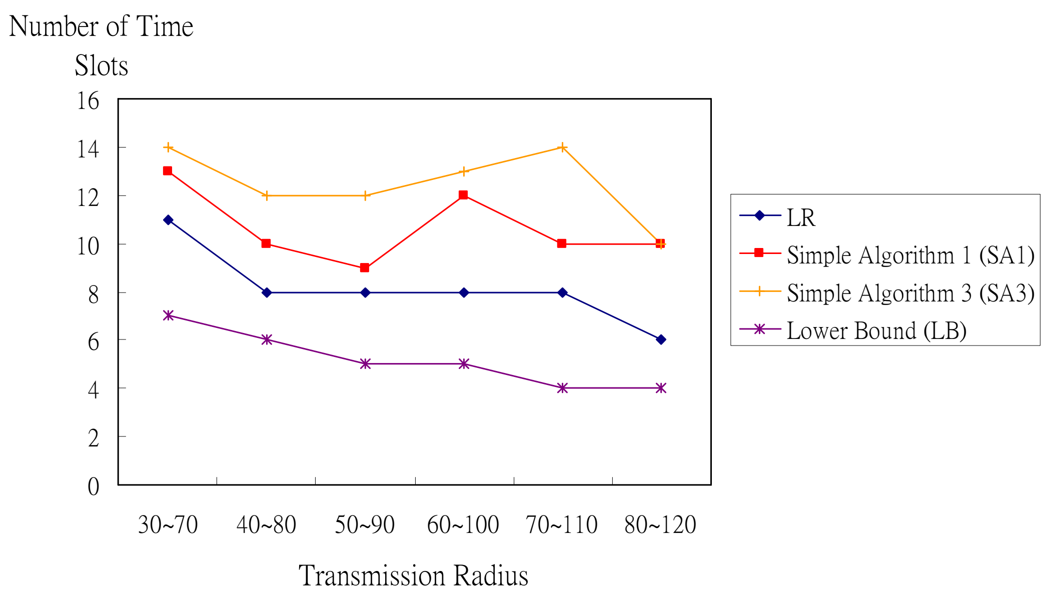


Fig 5-8 Experiment Result of Dynamic Network with Different Transmission Radius

As we increase the transmission radius 10 units at a time, the decreasing trend of number of time slots is more obvious than previous experiment. The performance of our algorithm is much better than SA1 and SA3 as the transmission radius increases.

5.3.6 Summary

The solution derived by our algorithm has less latency or even lower energy in some cases than the solutions of any other simple algorithms according to the series of experiment with different features. The only difference between these algorithms is the routing policy which leads to constructing different sets of multicast trees. The scheduling policy is the same for all algorithms. Therefore, as soon as the multicast trees are constructed, the transmission order is about to be decided by the structure of these multicast trees. Based on this observation, we can conclude the most important factor which influences the delay mostly is the method of routing. We also find the trees we construct has less depth and fewer non-leaf nodes compared to the one constructed by SA1. This structure potentially results in non-leaf nodes with more outgoing links and decrease the latency of transmission. However, it may produce large energy consumption and more potential interference which can delay the transmission schedule. As a result, the depth of trees constructed by SA3 is the least one but the delay computed by SA3 is not the shortest one. Our algorithm can find a good solution in these two effects with a tradeoff relationship. Another important factor is that the topology of network, which can decide the lower bound of delay. And a small change of topology may lead to totally different transmission schedule result.

5.4 Implementation in Long-Term Operation

The experiment we conduct so far is based on the design concept. We design a LR based algorithm and run many iterations to explore the best solution for the scheduling problem, which is proved to have good performance in both latency and energy consumption according to the experiment results in the previous section. However, in a real environment of MANET, nodes move with any speed and any direction in any time. As receiving a multicast request, we have to calculate a good enough decision before the topology become invalid. That means the computational time must be limited. Therefore, we have to run our algorithm in limited iteration and get a decision in time which must still have good quality. We conduct a variety of experiments to test the performance of our algorithm.

We set the scenario of experiment to be a long time interval which is composed of small phase. A set of multicast group requests to exchange short messages in each phase. The scenario can be applied to some practical military or rescue cases in which there are many teams exchanging strategical commands or rescue finding periodically. Our algorithm must calculate the transmission schedule in single phase which represents a short time interval, and the decision we compute in each phase is implemented in next phase. As we are in the process of computation for a phase, the multipliers we use in the first iteration is the one derived in last iteration of previous phase. From long-term operational perspective, we are convinced that the multipliers derived by many phases can optimize our solution as long as the environment has no rapid changes. This set of multipliers can provide a good reference to schedule the transmission order since it is accumulated by the experience of many different but related scenarios of each phase in a long time interval.

The more detail division of time interval is in the following graph. There are three stages for a multicast request before it is served. First, we receive the multicast requests in the duration equal to a phase time. Next, we use a phase time to calculate the scheduling decision, which means that we run our algorithm with limited iterations. In next phase time, the transmission schedule we just derived is implemented. The time interval composed of the three stages is called a cycle. In order to improve the efficiency of process, these cycles can be executed in a parallel manner. Therefore, three tasks proceed simultaneously in each phase. We receive the multicast requests while computing the decision which is requested in previous phase and used in next phase. And the transmission schedule calculated in previous phase is implemented in this phase.

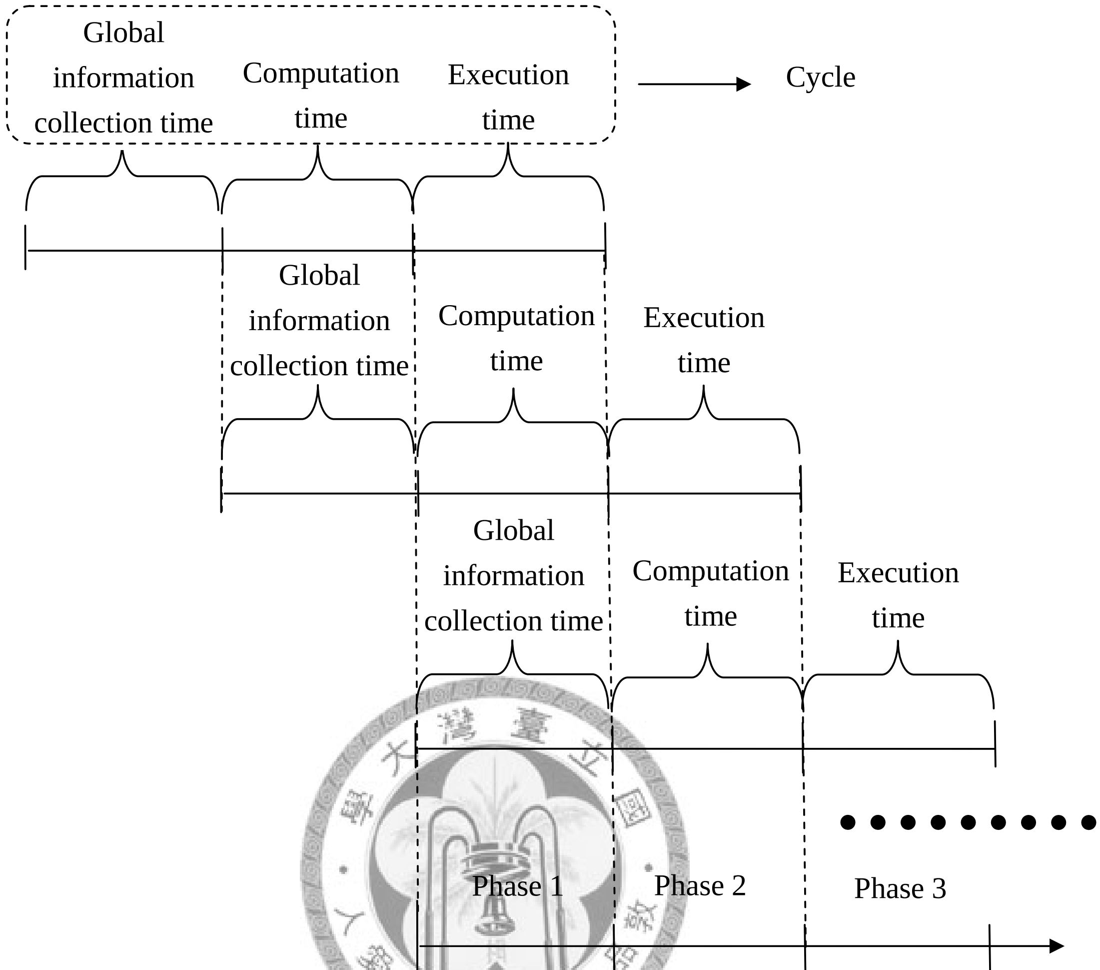


Fig 5-9 The Time Division of Implementation in Long-term Operation

5.4.1 A Scenario of Constant Speed during a Phase

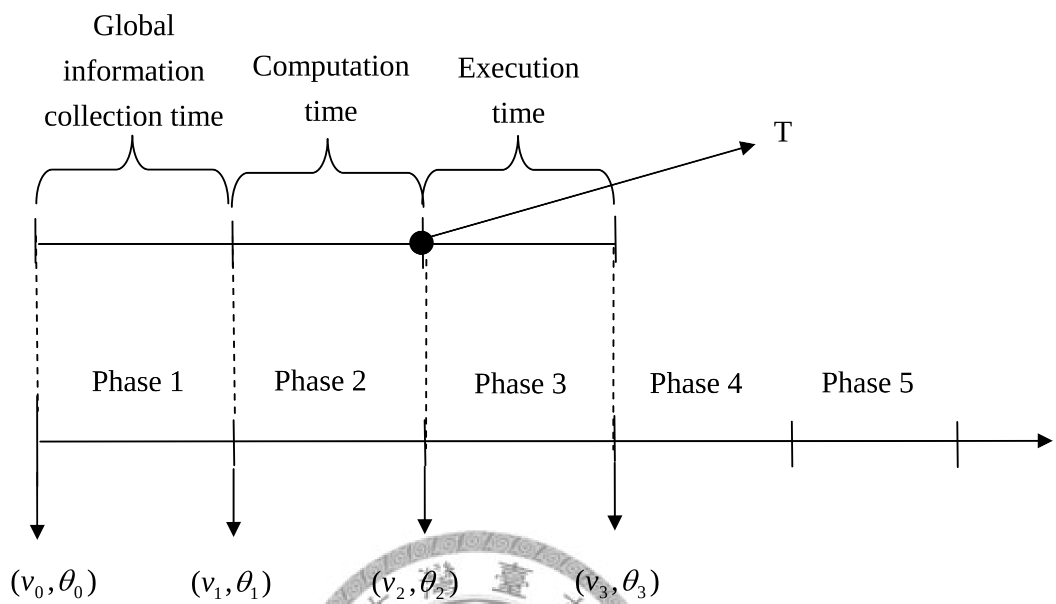


Fig 5-10 Scenario of Constant Speed during a Phase

In this experiment, the nodes change their speed and direction only between phases. During a phase time, they move with the same speed and direction as the initial value of speed and direction at the beginning of this phase. We assume that the mobility pattern can be described by Gauss-Markov mobility model which is introduced in chapter 1. In the graph above, after receiving the multicast requests and their mobility information in Phase 1, we predict the position, speed, and direction of nodes at time T according to the Gauss-Markov mobility model and compute the decision in Phase 2 based on the prediction. We assume that the prediction is accurate. Then the nodes execute the transmission schedule in Phase 3. The performance metric we consider is the delay, which is the total number of time slots to finish the data transmission. We vary two major factors including speed and the duration of a phase, which may have a great impact on delay. In the following experiment, we run our algorithm for 20 phases to solve the periodical multicast request of each phase.

- The Experiment Result of Different Speed

In each phase, our LR based algorithm is executed for limited iterations which are decided by the duration of a phase. In this experiment, we assume that a phase occupies 30 time slots and the number of iterations which our algorithm is permitted to run is about 100. We vary the speed from 0.1 to 1.9 and also test an extreme case in which the speed of each node is 0.

Table 5-16 Parameters of Constant Speed during a Phase with Different Speed

Number of Nodes	**Number of Source Nodes**
50	5
Density of Destination Nodes	**Transmission Range**
40%	50~90
Number of Time Slots / Phase	**Number of Iterations / Phase**
30	100

Table 5-17 Experiment Result of Constant Speed during a Phase with Different Speed

0		**0.1~0.5**		**0.3~0.7**		**0.5~0.9**		**0.7~1.1**		**0.9~1.3**		**1.1~1.5**		**1.3~1.7**		**1.5~1.9**	
LR	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**
8	14	13	18	15	8	13	18	13	18	13	18	15	8	13	18	13	18
9	14	11	21	10	20	9	16	7	18	10	20	8	13	11	20	12	19
9	14	10	19	13	23	11	15	9	18	11	18	11	25	10	12	8	16
8	14	10	15	12	13	11	24	10	14	8	18	8	17	7	16	11	18
10	14	8	19	9	16	11	11	11	16	11	17	9	20	11	23	10	16
8	14	11	15	10	17	11	15	11	16	9	14	10	12	11	22	12	16
7	14	11	20	9	17	12	17	12	26	12	19	7	12	10	27	8	26
9	14	12	14	10	19	10	14	12	22	12	17	13	21	16	14	11	16
10	14	11	22	9	15	11	17	9	22	12	20	11	21	10	19	13	22
9	14	10	17	10	10	11	19	13	19	14	23	9	15	9	17	12	13
9	14	10	15	10	13	10	17	12	17	15	22	10	12	10	15	13	31
9	14	11	13	10	13	15	20	11	15	13	22	10	14	13	21	12	22
8	14	10	21	12	19	14	21	9	12	13	15	7	13	14	20	11	17
10	14	9	16	10	12	10	21	11	21	11	22	11	19	10	15	13	17
9	14	12	11	10	16	9	28	13	35	14	21	11	19	12	16	12	17
9	14	9	16	12	19	13	23	11	15	11	20	9	16	12	26	11	22
9	14	6	12	11	21	8	22	12	21	12	19	9	24	11	18	12	15
8	14	9	12	13	15	14	23	11	14	10	20	13	16	13	23	10	17
9	14	11	14	14	18	11	22	10	16	9	16	14	25	10	20	12	16
8	14	10	18	10	16	12	25	11	14	12	19	15	23	12	14	12	18
8.75	14	10.2	16.4	10.95	16	11.3	19.4	10.9	18.45	11.6	19	10.5	17.25	11.25	18.8	11.4	18.6

Average Number of Time Slots

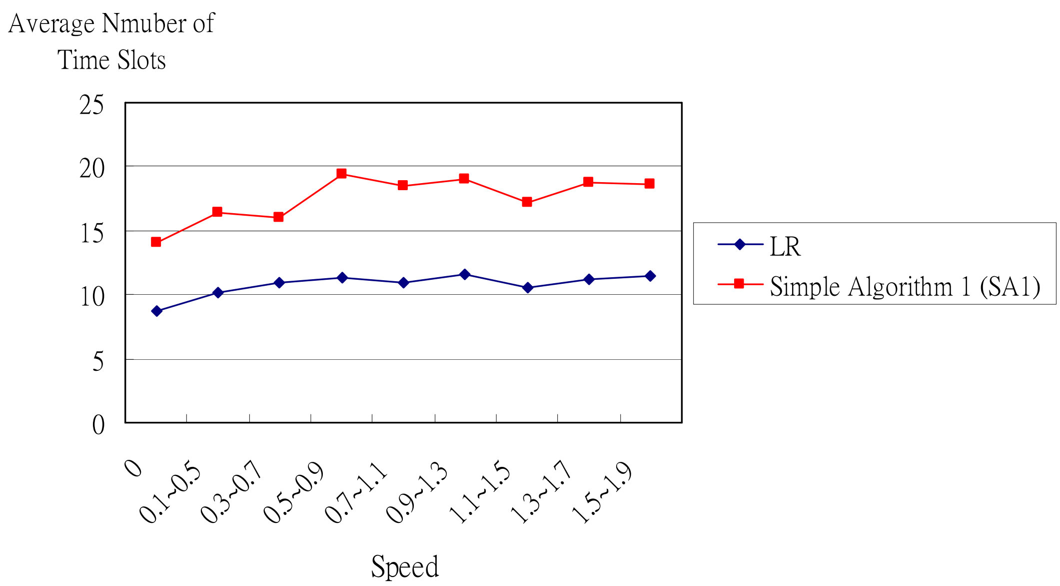


Fig 5-11 Experiment Result of Constant Speed during a Phase with Different Speed

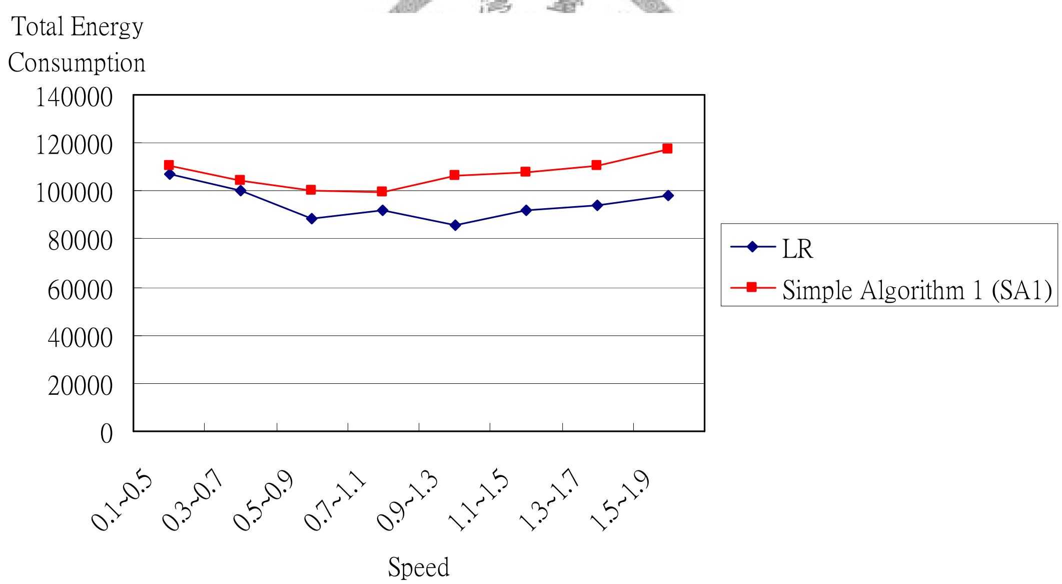


Fig 5-12 Energy Consumption of Constant Speed during a Phase with Different Speed

Each plot in the graph above represents the average value of solutions in 20 phases. Our solution is better than Shortest Path Tree algorithm in each case with different speed. The result has an important meaning. The experiment we conduct is a design problem in which our algorithm can run many iterations to explore the good solution. And the result shows our solution is better than the one of Shortest Path Tree algorithm. However, the important strength of Shortest Path Tree algorithm is efficiency since it only needs one run to compute the solution, which is superior to our algorithm. The simplicity makes it have higher practicability which is the weakness of our algorithm. Nevertheless, the experiment we conduct in this section is proved that, even in the environment close to reality, our algorithm can provide better solution than Shortest Path Tree algorithm by running limited iterations.

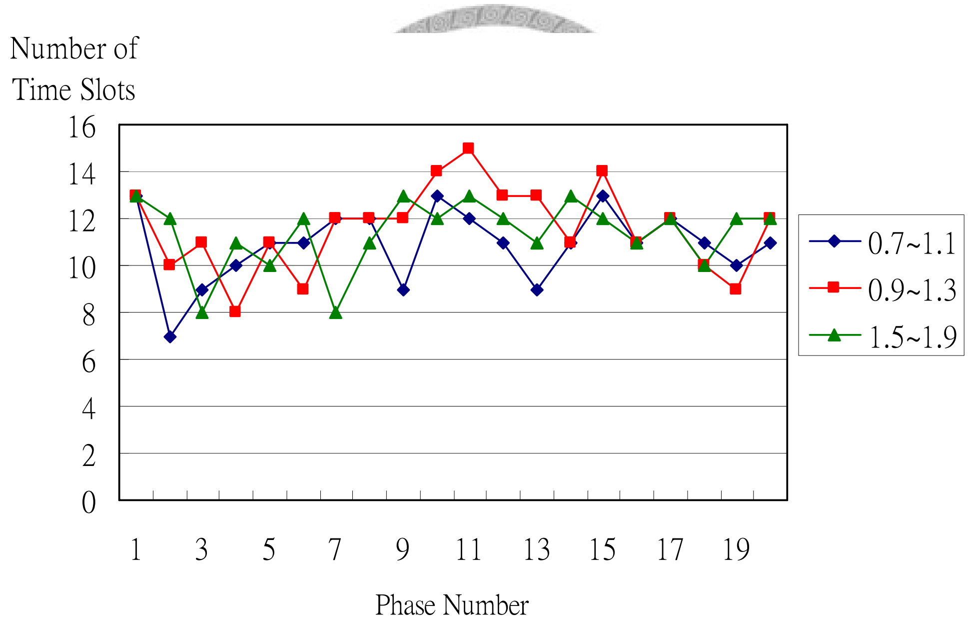


Fig 5-13 Experiment Result of Each Phase in the Scenario of Constant Speed during a Phase with Different Speed

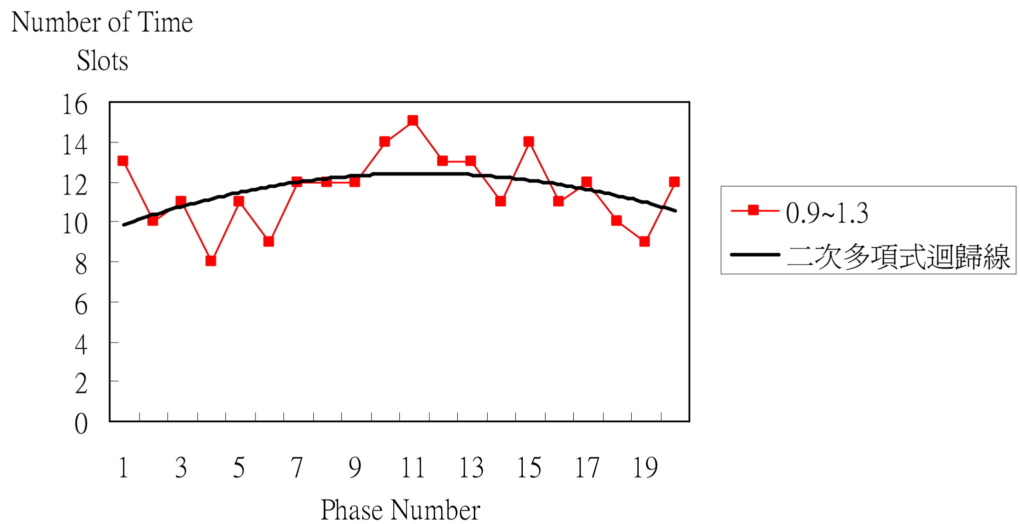


Fig 5-14 Experiment Result of Each Phase in the Scenario of Constant Speed during a Phase with Speed=0.9~1.3

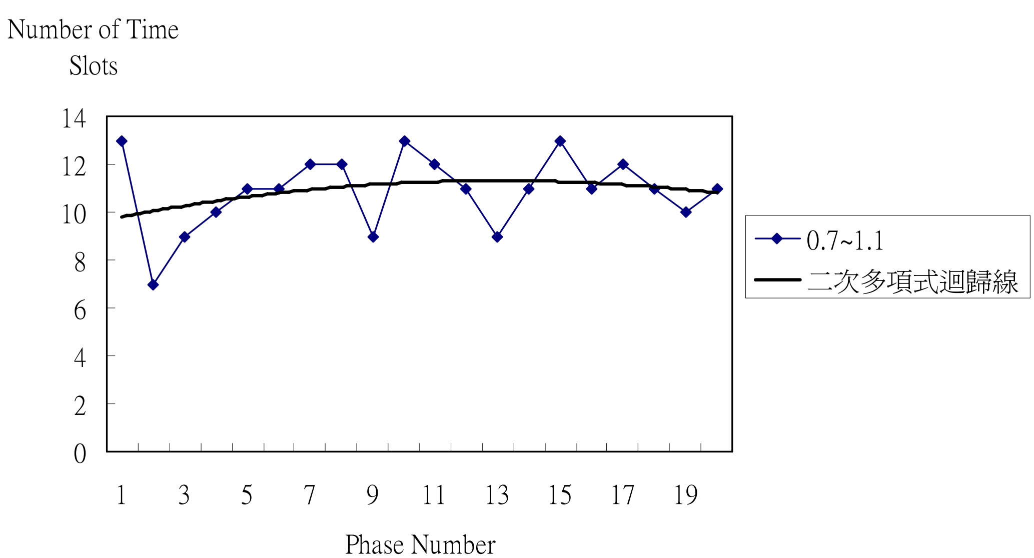


Fig 5-15 Experiment Result of Each Phase in the Scenario of Constant Speed during a Phase with Speed = 0.7~1.1

We show the solution of each iteration and the line of regression analysis to it. As mentioned before, the solution of each phase can be improved by the set of multipliers derived in previous phase. According to the result of this experiment, the trend in first ten iterations is increasing and then decreasing in last ten iterations. The possible reason is the statistical bias and we usually drop the first part of the result when we do the statistics analysis. However, we can not yet conclude that the solution is optimized phase by phase based on the multipliers.

- The Experiment Result of Different Number of Time Slots in each Phase

Table 5-18 Parameters of Constant Speed during a Phase with Different Number of Time Slots in Each Phase

Number of Nodes	**Number of Source Nodes**
30	5
Density of Destination Nodes	**Transmission Range**
100%	80~100
Speed	
0.7~1.1	

Table 5-19 Experiment Result of Constant Speed during a Phase with Different Number of Time Slots in Each Phase

10		**20**		**30**		**40**		**50**		**60**		**70**		**80**	
LR	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**	**LR**	**SA1**
8	10	8	10	8	10	8	10	8	10	8	10	8	10	8	10
7	11	7	11	7	13	7	14	8	15	9	17	8	14	8	17
8	11	10	14	14	12	14	13	8	13	8	9	7	11	8	11
10	11	9	15	11	15	8	17	7	13	7	9	7	8	6	9
9	11	10	16	7	14	9	18	8	9	5	8	5	10	5	7
10	14	8	16	7	13	7	11	7	11	8	17	7	7	7	11
11	14	8	17	8	11	9	9	8	9	7	7	6	11	7	16
9	15	10	17	9	15	8	11	6	7	9	12	7	13	8	11
10	11	14	17	11	14	8	11	6	10	7	14	9	19	6	12
7	9	8	16	11	15	8	15	6	7	10	11	5	14	5	14
7	11	9	18	8	17	7	15	9	14	6	8	7	8	6	11
9	12	9	16	8	8	6	7	9	18	6	10	5	10	7	12
8	7	9	14	7	9	7	10	9	12	8	11	9	15	5	12
7	9	7	12	6	11	7	7	7	16	7	12	6	10	6	17
8	9	10	15	7	11	8	10	5	10	5	12	4	6	8	14
10	11	8	13	7	9	6	14	8	15	6	9	8	12	8	13
9	11	9	10	7	11	6	13	9	14	5	9	8	15	7	18
9	10	7	13	7	11	7	16	8	11	5	12	7	14	8	10
7	10	6	9	8	7	7	17	7	12	6	6	9	14	9	11
8	10	6	8	5	13	8	12	7	14	6	11	9	12	7	12
8.55	10.85	8.6	13.85	8.15	11.95	7.75	12.5	7.5	12	6.9	10.7	7.05	11.65	6.95	12.4

Average Number of Time Slots

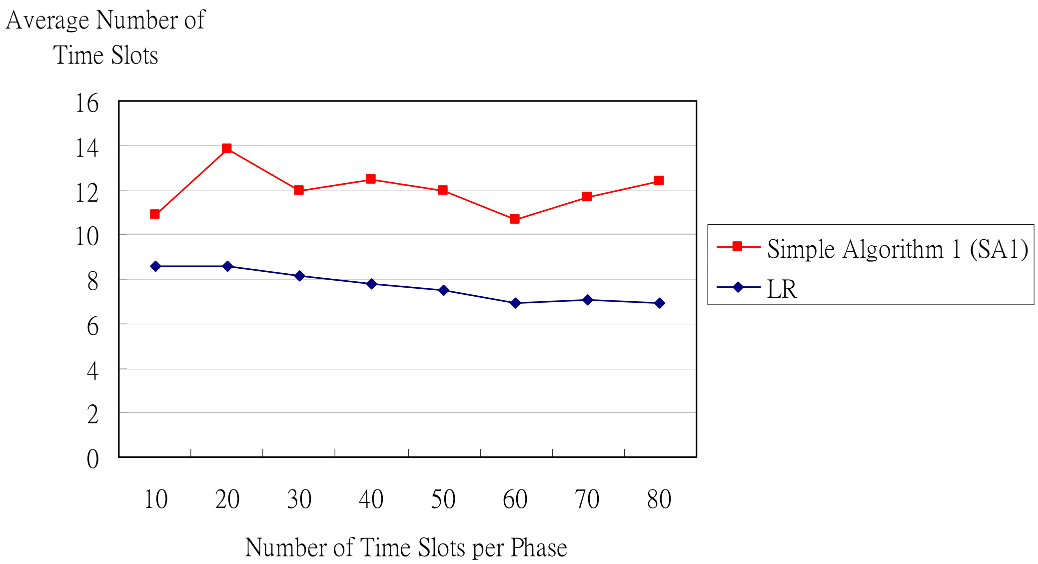


Fig 5-16 Experiment Result of Constant Speed during a Phase with Different Number of Time Slots in Each Phase

We extend the duration of each phase from 10 time slots to 80 time slots. The more time slots a phase occupies the more iteration our algorithm can executes, which would lead to better solution. However, another factor may influence the solution quality we find. The topology would have more difference between phases for the phases with longer duration, which may results in bigger variation of solution and higher probability of deriving worse solution.

According to the experiment result above, in the cases of 10 time slots ~ 60 time slots, we can derive the better solution as the duration of phase become longer. One may have doubt with this statement since Shortest Path Algorithm can also find better solution with just one run as the number of time slots a phase occupies increases. This trend represents that the topology in the scenario with longer duration can takes less time to finish all multicast requests. Therefore, the reason of deriving better solution is not the more iteration which makes us compute better solution. However, the result of 70 time slots and 80 time slots can overturn this inference. The delay computed by Shortest Path Algorithm increase while the delay derived by our algorithm remains stable. This result illustrates the longer duration of a phase can really help us explore better solution.

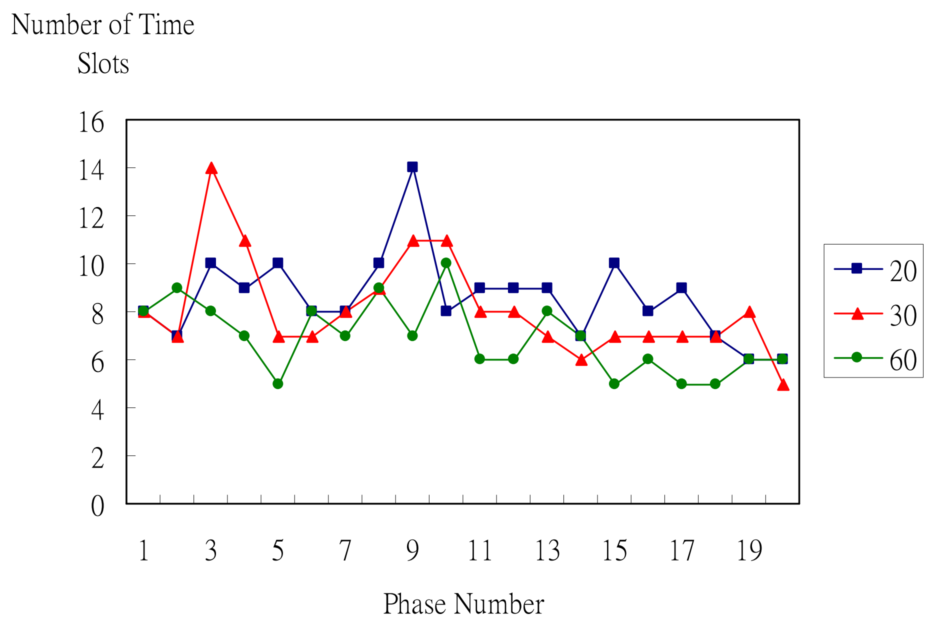


Fig 5-17 Experiment Result of Each Phase in the Scenario of Constant Speed during a Phase with Different Number of Time Slots in Each Phase

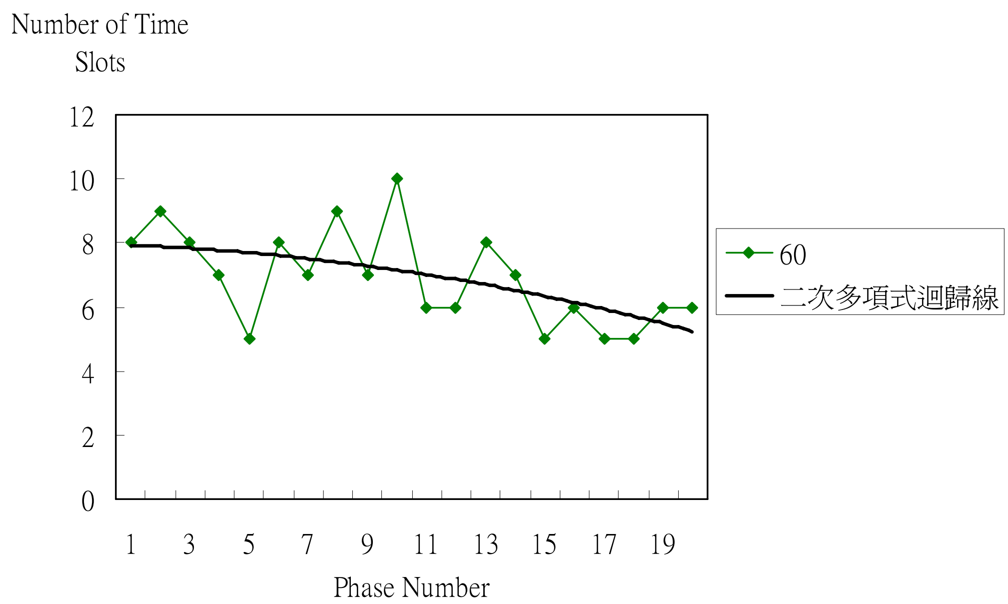


Fig 5-18 Experiment Result of Each Phase in the Scenario of Constant Speed during a Phase with Number of Time Slots = 60 in Each Phase

According to the experiment result of each phase of these cases, the solution of each phase is improved as we execute the algorithm for more phases. It proves that the multipliers derived in each phase can help us compute better solution in next phase and can be used as the initial value of multiplier used in the scenario with the same multicast requests in the future.

5.4.2 A Scenario of Variable Speed during a Phase

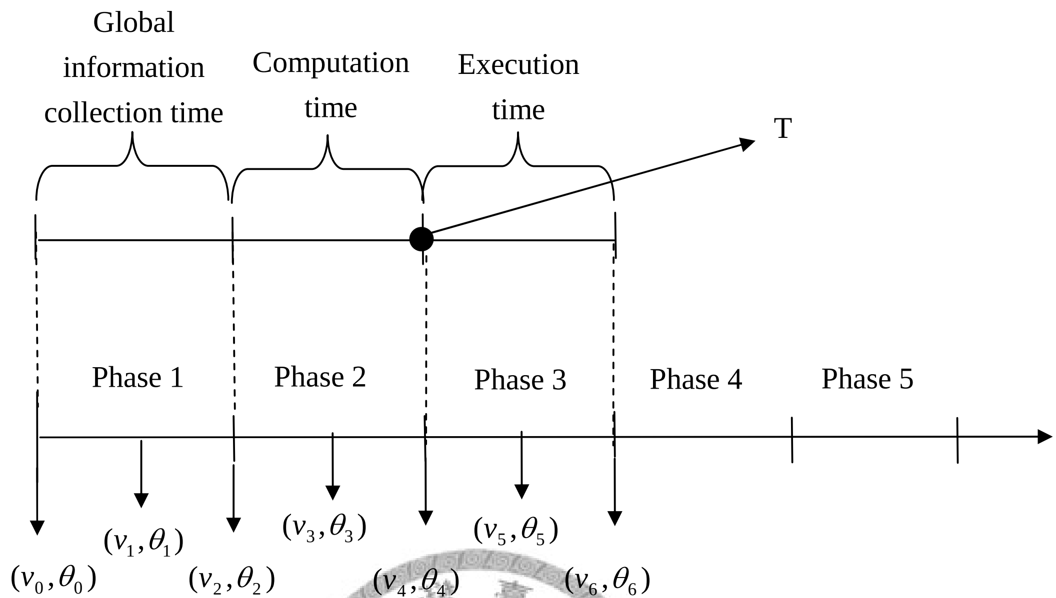


Fig 5-19 Scenario of Variable Speed during a Phase

In this experiment, the nodes can change their speed and direction during a phase. It means that the decision, which is computed in Phase 2 based on the mobility information at time T, may become invalid while implemented in Phase 3. In other words, the prediction of mobility information become inaccurate and the nodes may move outside the transmission range of its parent on the tree causing the loss of data. Therefore, the result we really concern about is how many destinations have received the data.

In order to study the performance of our algorithm in such scenario, we defined two metrics which are calculated by difference concept. The total number of destinations is donated by D and the number of destinations which have received the data successfully is donated by d. Delivery ratio 1 is calculated by $\frac{d}{D}$, which is destination-based. Next, we donate the total number of trees as T_r and the number of trees on which all destinations have received the data successfully as t_r. Delivery ratio 2 is calculated by $\frac{t_r}{T_r}$, which is tree based. We also vary speed and change frequency of speed and direction during a phase to test the robust of our algorithm.

- The Experiment Result of Different Speed

Table 5-20 Parameters of Variable Speed during a Phase with Different Speed

Number of Nodes	**Number of Source Nodes**
50	5
Density of Destination Nodes	**Transmission Range**
40%	50~90
Number of Time Slots / Phase	**Number of Iterations / Phase**
30	100
Change Frequency of Speed and Direction / Phase	
15	

Table 5-21 Experiment Result of Variable Speed during a Phase with Speed = 0.1~1.1

0.1~0.5		0.3~0.7		0.5~0.9		0.7~1.1	
Ratio 2	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**
0.4	0.96	0.2	0.9	0	0.89	0	0.89
0.2	0.95	0.6	0.93	0.2	0.95	0.4	0.92
0.2	0.93	0.4	0.94	0.2	0.91	0.2	0.88
0.8	0.99	0.4	0.97	0.6	0.97	0.6	0.96
0.2	0.96	0.8	0.99	0.4	0.86	0.6	0.98
0.4	0.91	0.4	0.92	0.2	0.86	0	0.92
0.6	0.98	0.4	0.8	0.4	0.97	0.4	0.94
0.6	0.9	0.6	0.97	0.4	0.96	0.4	0.93
0.8	0.99	0.4	0.96	0.4	0.95	0.4	0.93
0.4	0.94	0.8	0.99	0.6	0.98	0.2	0.95
0.8	0.99	0.6	0.95	0.6	0.95	1	1
0.6	0.98	1	1	0.6	0.98	0.8	0.99
0.6	0.96	1	1	0.6	0.95	0.8	0.99
1	1	0.2	0.82	0.6	0.98	0.6	0.97
0.6	0.97	0.4	0.97	0.4	0.97	0.4	0.97
0.4	0.93	0.8	0.99	1	1	0.8	0.99
0.8	0.99	0.6	0.97	0.6	0.98	0.8	0.99
0.6	0.98	0.6	0.98	0.8	0.99	0.6	0.98
0.8	0.99	0.8	0.99	0.8	0.99	0.6	0.98
1	1	0.6	0.98	0.8	0.99	0.6	0.97
0.59	0.965	0.58	0.951	0.51	0.954	0.51	0.9565

Average Delivery Ratio

Table 5-22 Experiment Result of Variable Speed during a Phase with Speed = 0.9~1.9

0.9~1.3		1.1~1.5		1.3~1.7		1.5~1.9	
Ratio 2	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**
0.2	0.88	0.4	0.9	0.4	0.9	0.4	0.9
0.4	0.89	0	0.94	0.2	0.91	0.6	0.96
0.2	0.93	0.2	0.94	0	0.8	0.6	0.95
0.2	0.83	0.4	0.95	0	0.91	0.2	0.92
0.6	0.97	0.4	0.96	0.6	0.95	0.4	0.95
0.4	0.96	0.4	0.94	0.6	0.98	0.2	0.95
0.2	0.95	0.6	0.98	0.8	0.99	0.4	0.95
0.2	0.94	0.2	0.81	0.6	0.98	0.2	0.96
0.4	0.83	0.8	0.96	0.8	0.98	0.2	0.95
0	0.94	0.6	0.9	0.4	0.95	0.2	0.91
0.8	0.99	0.4	0.95	0.6	0.96	0	0.88
0.6	0.91	0.2	0.93	0.6	0.98	0.2	0.93
0.6	0.97	0.6	0.98	0.2	0.92	0.6	0.98
0.2	0.95	0.6	0.97	0	0.94	0	0.89
0.4	0.95	0	0.92	0.6	0.91	0	0.92
0.6	0.9	0.4	0.97	0.2	0.96	0.2	0.93
0.6	0.98	0.2	0.89	0	0.94	0.6	0.98
0.6	0.98	0.4	0.95	0.6	0.91	0.2	0.89
0.2	0.95	0.2	0.91	0.2	0.94	0.4	0.95
0.2	0.93	0.2	0.9	0.6	0.97	0.4	0.93
0.38	0.9315	0.36	0.9325	0.4	0.939	0.3	0.934

Average Delivery Ratio

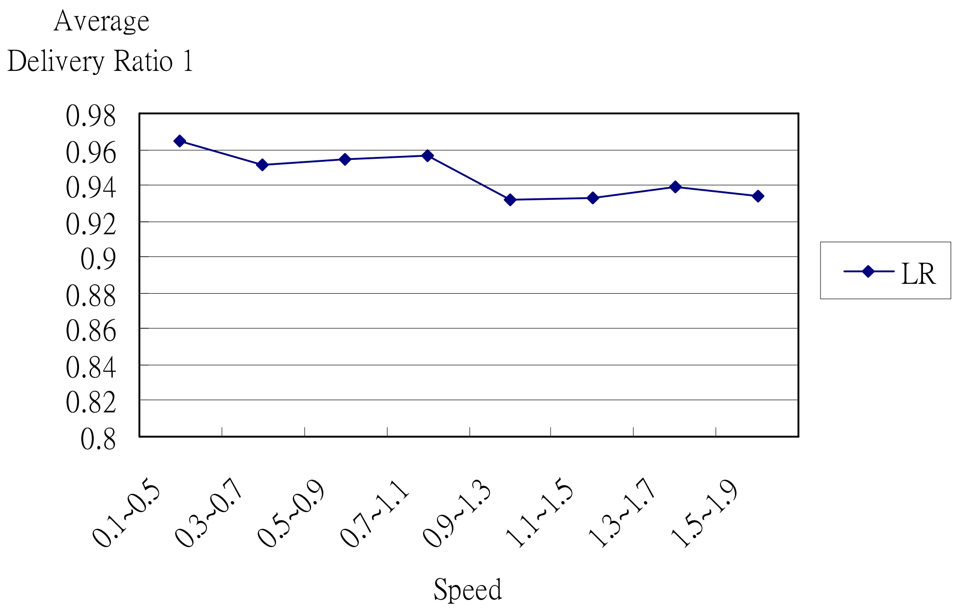


Fig 5-20 Average Delivery Ratio 1 of Variable Speed during a Phase with Different Speed

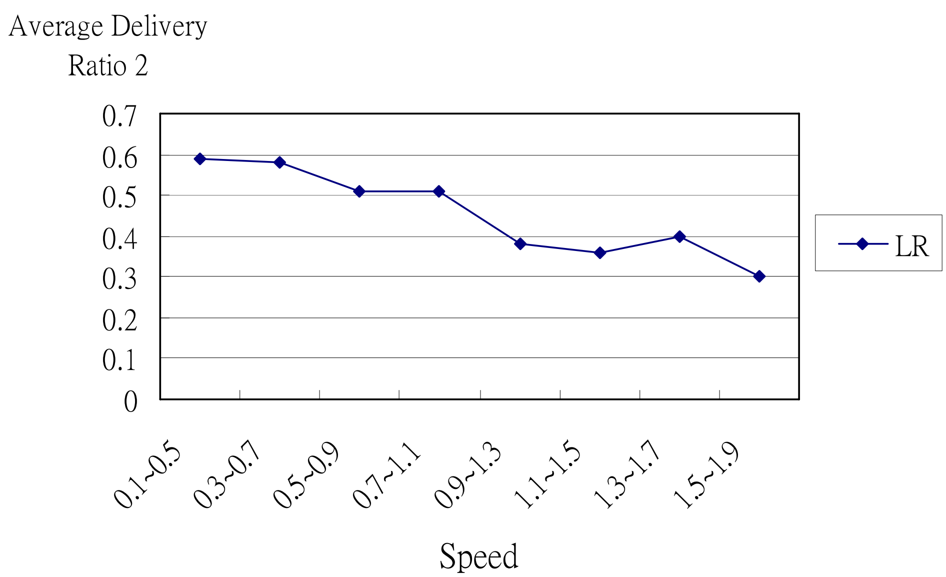


Fig 5-21 Average Delivery Ratio 2 of Variable Speed during a Phase with Different Speed

It is intuitive that the scheduling policy we make at the beginning of a phase may become invalid much faster as the moving speed of nodes increases. More nodes on the trees may loss the sent by their parents according to the transmission schedule since they are out of transmission range of their parents if they move with a fast speed. The trend is consistent with our experiment results that the delivery ratio is smaller when nodes moving faster. However, the delivery ratio of our algorithm still remains 0.93 calculated by delivery ratio 1 or 0.3 calculated by delivery ratio 2. The high delivery ratio implies our scheduling policy takes a short period of time to transmit data to destinations before the topology has rapid change. Therefore, delivery ratio can be considered as another significant metric to test the performance of our algorithm. Also, according to the following set of graphs, the delivery ratio can be improved and tend to converge as more phases passed by.

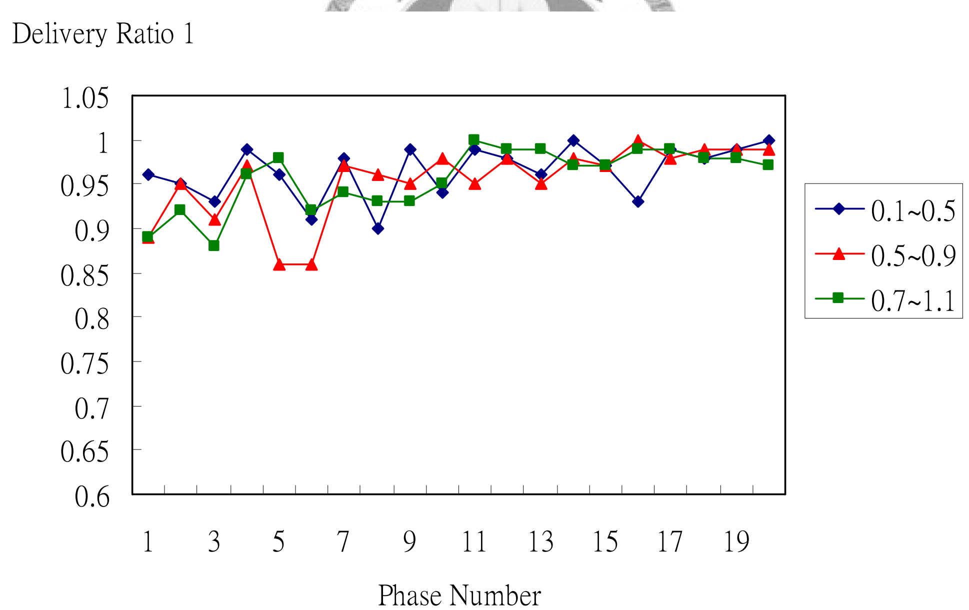


Fig 5-22 Delivery Ratio 1 of Each Phase in the Scenario of Variable Speed during a Phase with Different Speed

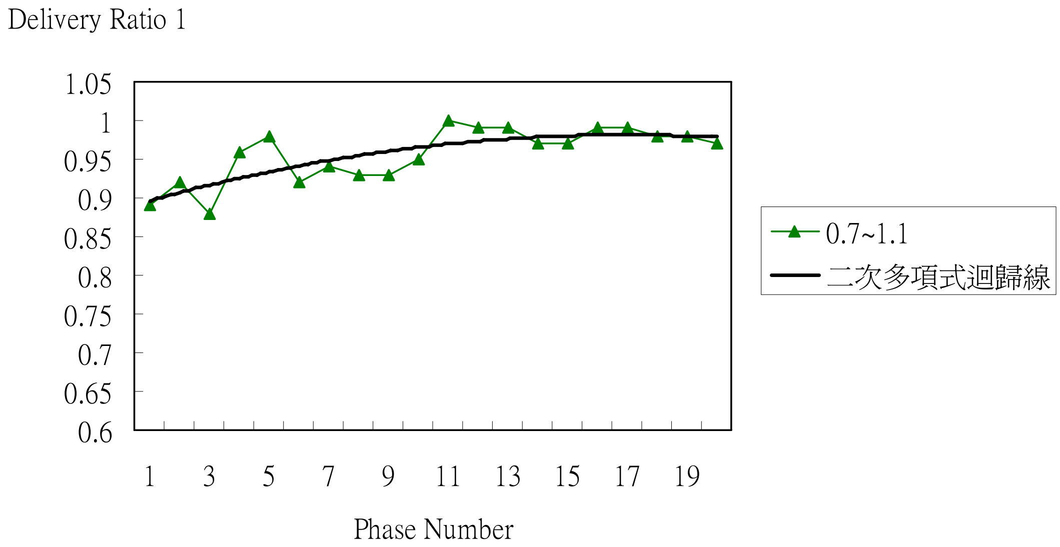


Fig 5-23 Delivery Ratio 1 of Each Phase in the Scenario of Variable Speed during a Phase with Speed = 0.7~1.1

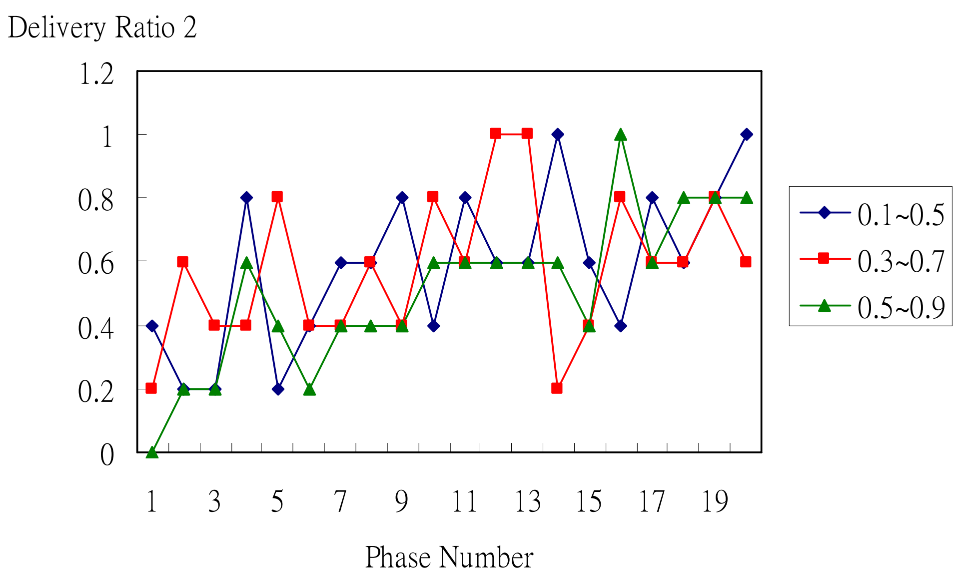


Fig 5-24 Delivery Ratio 2 of Each Phase in the Scenario of Variable Speed during a Phase with Different Speed

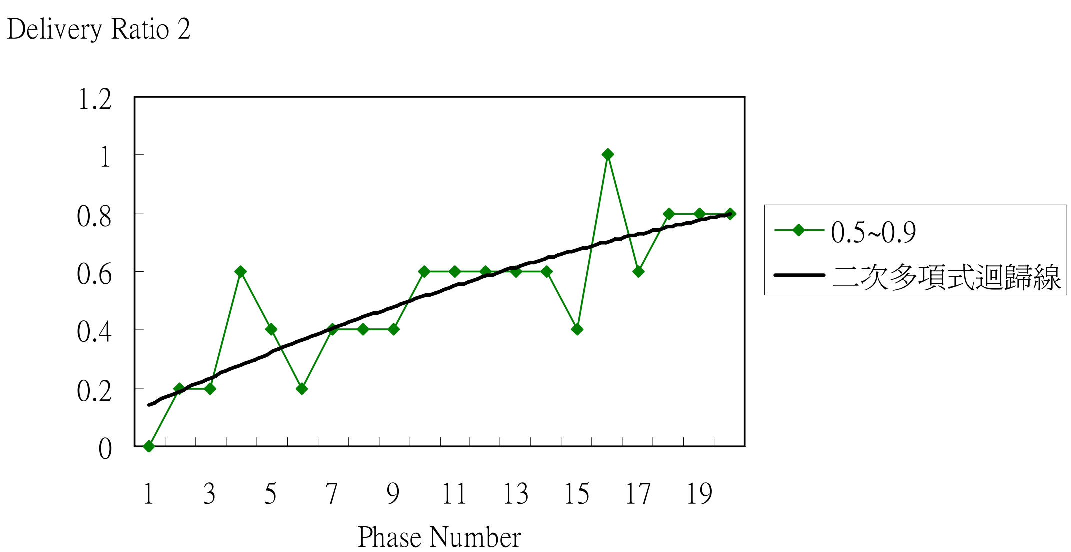


Fig 5-25 Delivery Ratio 2 of Each Phase in the Scenario of Variable Speed during a Phase with Speed = 0.5~0.9

- The Experiment Result of the Different Change Frequency of Speed and Direction

Table 5-23 Parameters of Variable Speed during a Phase with Different Change Frequency of Speed and Direction

Number of Nodes	**Number of Source Nodes**
50	3
Density of Destination Nodes	**Transmission Range**
100%	90~100
Number of Time Slots / Phase	**Number of Iterations / Phase**
30	100
Speed	
0.1~0.5	

Table 5-24 Experiment Result of Variable Speed during a Phase with Change Frequency = 2, 3, 4, 6

2		3		4		6	
Ratio 1	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**
0.986667	0.333333	0.986667	0.333333	0.98	0	0.973333	0
1	1	0.913333	0	0.973333	0	0.946667	0
0.966667	0	0.96	0.333333	1	1	0.98	0.333333
1	1	0.92	0	0.973333	0.333333	0.96	0
0.913333	0	1	1	0.946667	0	0.973333	0
0.993333	0.666667	1	1	0.926667	0	0.993333	0.666667
1	1	0.98	0.333333	0.966667	0	0.973333	0
0.946667	0	0.993333	0.666667	0.993333	0.666667	0.993333	0.666667
0.84	0	1	1	0.993333	0.666667	1	1
0.986667	0.666667	0.98	0	0.986667	0.333333	0.98	0.333333
1	1	1	1	0.973333	0.333333	0.973333	0
1	1	1	1	0.986667	0.666667	0.966667	0
1	1	1	1	1	1	0.98	0.333333
0.98	0.333333	1	1	1	1	1	1
1	1	0.993333	0.666667	0.993333	0.666667	0.986667	0.666667
1	1	1	1	1	1	0.993333	0.666667
1	1	1	1	1	1	1	1
1	1	1	1	1	1	0.986667	0.333333
1	1	1	1	0.986667	0.333333	1	1
1	1	1	1	1	1	0.986667	0.666667
0.980667	0.7	0.986333	0.716667	0.984	0.55	0.982333	0.433333

Average Delivery Ratio

Table 5-25 Experiment Result of Variable Speed during a Phase with Change Frequency =10, 15, 30

10		**15**		**30**	
Ratio 1	**Ratio 2**	**Ratio 1**	**Ratio 2**	**Ratio 1**	**Ratio 2**
0.96	0	0.953333	0	0.946667	0
0.973333	0	0.986667	0.333333	0.866667	0
0.953333	0	0.973333	0.333333	0.973333	0
0.98	0	0.953333	0.333333	0.933333	0
0.96	0	0.973333	0.333333	0.973333	0
0.946667	0	0.98	0	0.953333	0.333333
0.986667	0.333333	0.8	0	0.953333	0
0.973333	0.333333	0.966667	0	0.973333	0
0.96	0.333333	0.986667	0.333333	0.986667	0.333333
0.966667	0.333333	0.966667	0	0.973333	0.333333
0.953333	0.333333	0.98	0.333333	0.98	0.666667
0.953333	0.333333	0.953333	0.333333	0.94	0.333333
0.98	0.666667	0.966667	0.666667	0.986667	0.333333
0.986667	0.333333	0.96	0.333333	0.98	0
0.92	0	1	1	0.966667	0.333333
0.966667	0.333333	1	1	0.886667	0.333333
0.986667	0.666667	0.893333	0.333333	0.986667	0.666667
0.973333	0.333333	0.986667	0.333333	0.98	0.666667
0.98	0.333333	0.993333	0.666667	0.986667	0.333333
0.98	0.333333	0.986667	0.666667	0.946667	0
0.967	0.25	0.963	0.366667	0.958667	0.233333

Average Delivery Ratio

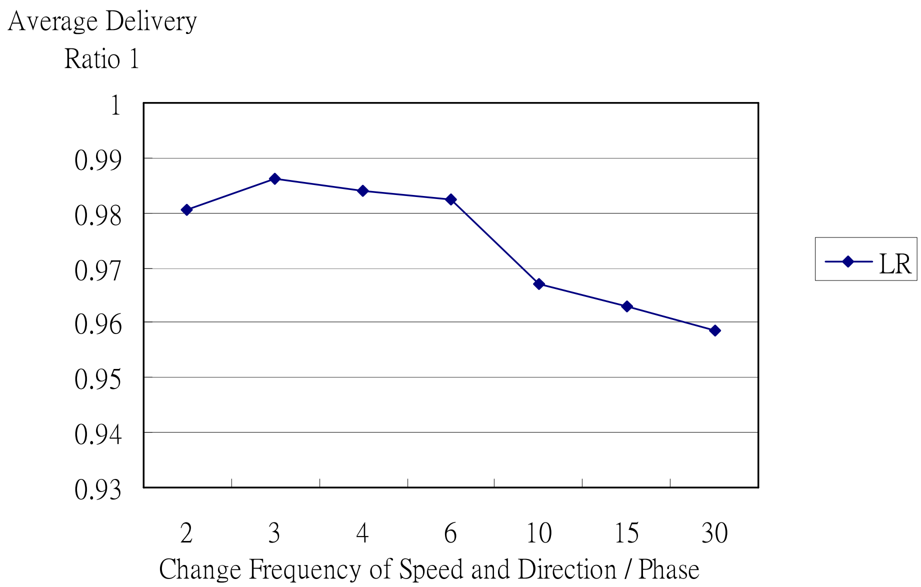


Fig 5-26 Average Delivery Ratio 1 of Variable Speed during a Phase with Different Change Frequency of Speed and Direction

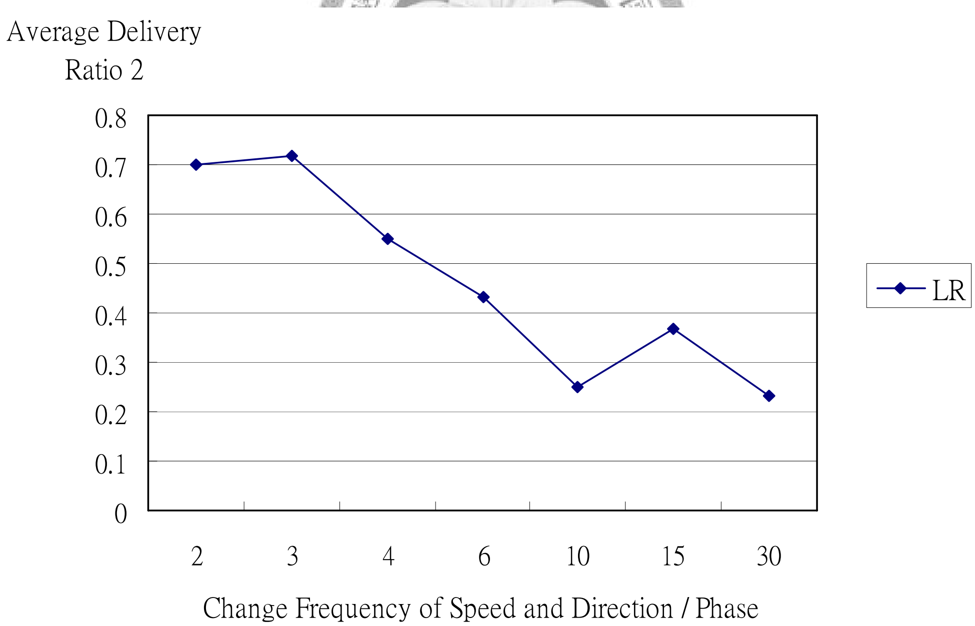


Fig 5-27 Average Delivery Ratio 2 of Variable Speed during a Phase with Different Change Frequency of Speed and Direction

In this experiment, we increase the change frequency of speed and direction in a phase to study the effect to delivery ratio. A phase occupies 30 time slots in this scenario and the change frequency =10 means that we change the speed and direction of all nodes once for each of three time slots, which has ten changes in a phase. The inference is similar with previous experiment that delivery ratio will drop down as the change of speed and direction is more frequent since the scheduling decision becomes invalid sooner. Note that delivery ratio may has a small increase as the value exceed some threshold, for example, change frequency =15 in the graph of delivery ratio 2. The possible cause is that some nodes gradually move away from each other and turn back after they reach the edge of the square. The positions of nodes may become closer, which will lead to the increase of delivery ratio. In addition, we also observe the result of the solution of each phase shown as follows, which can support our statement that the delivery ratio is improved phase by phase.

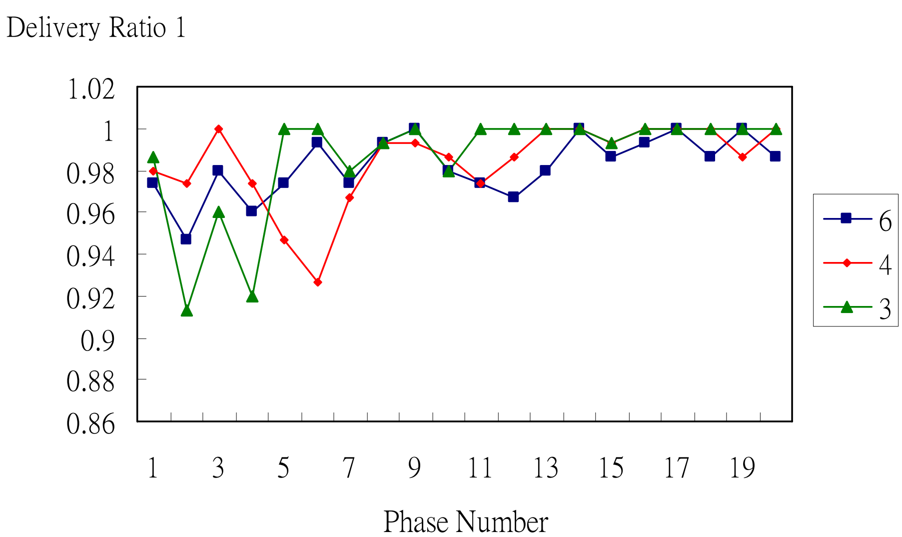


Fig 5-28 Delivery Ratio 1 of Each Phase in the Scenario of Variable Speed during a Phase with Different Change Frequency of Speed and Direction

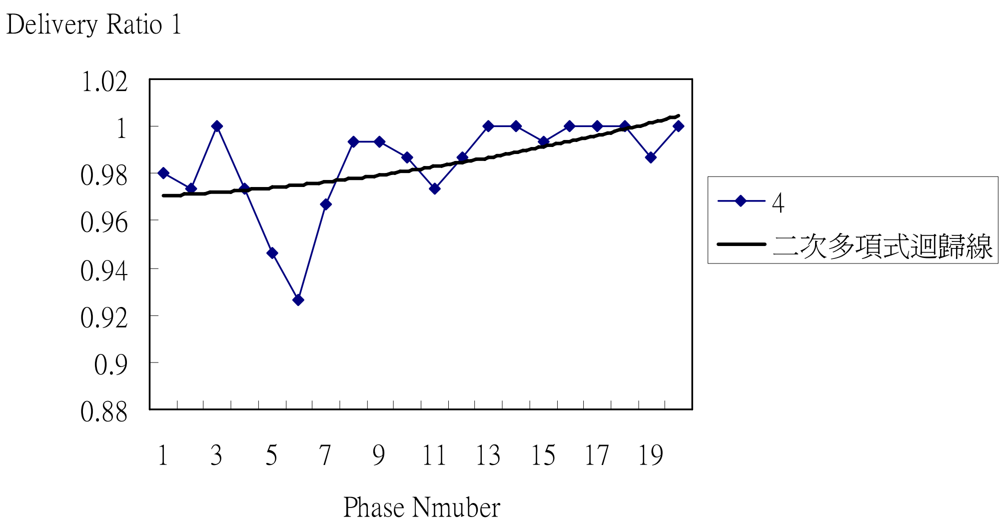


Fig 5-29 Delivery Ratio 1 of Each Phase in the Scenario of Variable Speed during a Phase with Change Frequency = 4

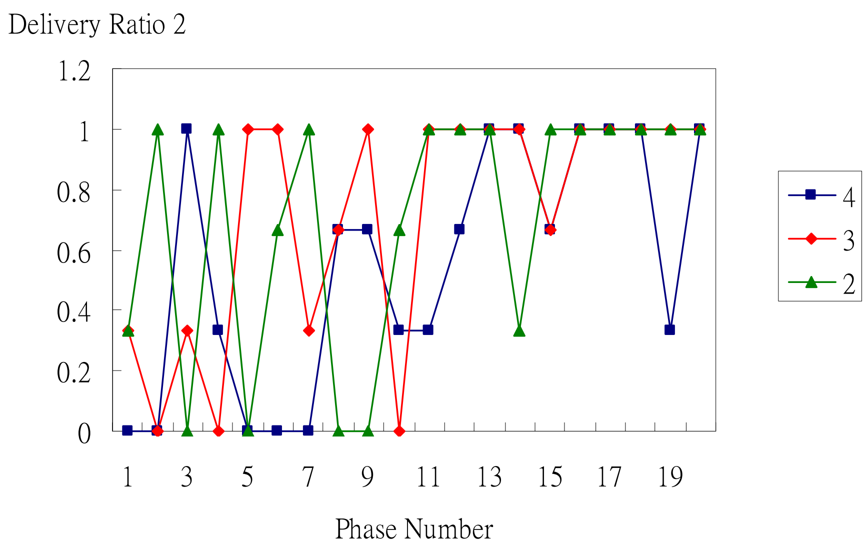


Fig 5-30 Delivery Ratio 2 of Each Phase in the Scenario of Variable Speed during a Phase with Change Frequency = 2, 3, 4

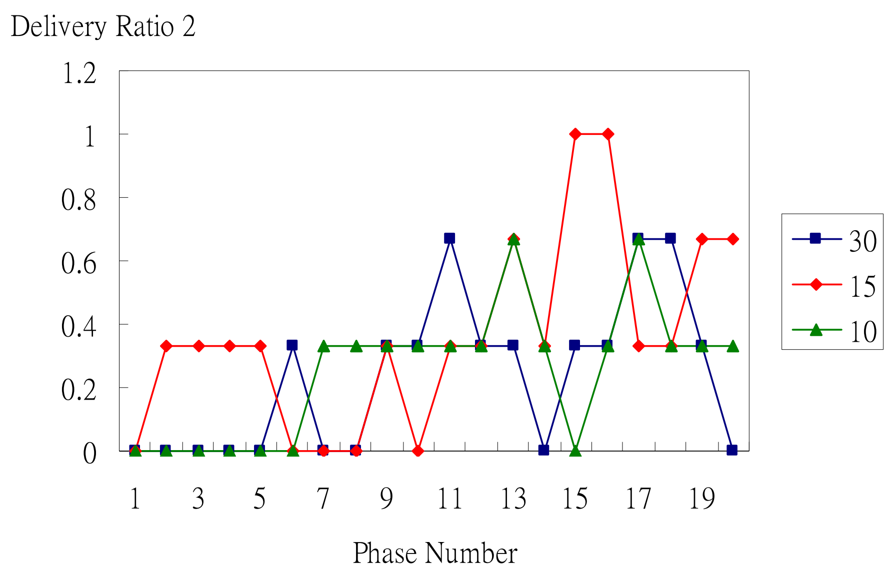


Fig 5-31 Delivery Ratio 2 of Each Phase in the Scenario of Variable Speed during a

Phase with Change Frequency = 30, 15, 10

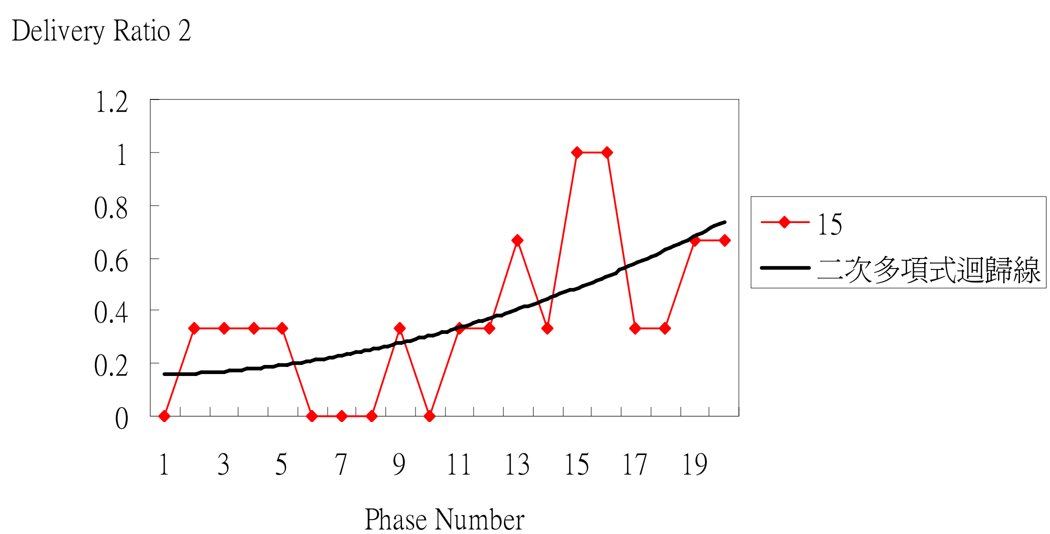


Fig 5-32 Delivery Ratio 2 of Each Phase in the Scenario of Variable Speed during a

Phase with Change Frequency = 15

Chapter 6 Conclusion

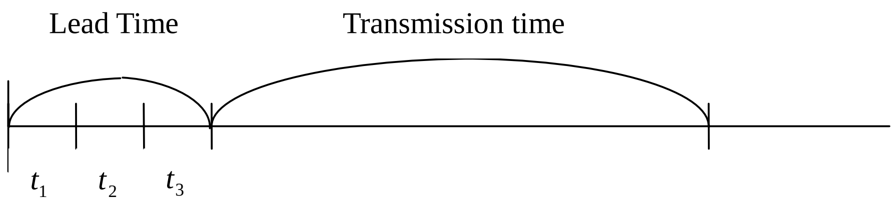
6.1 Summary

MANET is a challenging environment since the battery resource is limited and the rapid change of network topology due to the mobility of nodes. In our work, we focus on the multicasting function which imposes low latency in MANET with multiple multicast groups, and aim to design an efficient transmission schedule which can minimize the end-to-end delay for transmitting the data to all intentional destinations and consume low energy as well.

We formulate this problem as an integer programming problem and propose a four-stage heuristic based on the Lagrangean Relaxation technique to solve the complicated scheduling problem. We design a set of experiment with different scenarios to evaluate the solution quality of our LR based algorithm. The result of computational experiment indicates that our algorithm has better performance than other algorithms used for comparison.

We also conduct another experiment in which the scenario is a longer time interval divided into many small phase. In each phase, our algorithm is executed in order to satisfy the multicast requests in this phase. The goal of this experiment is to test the efficiency and practicability since the time our algorithm permitted to run is limited. It was proved by our experiment result that our algorithm still can derive good solution quality compared to Shortest Path Tree algorithm with limited computational iterations.

6.2 Future Work



In our work, according to the graph above, we compute a transmission schedule used in the period of transmission time based on the mobility information at the beginning of this period. The transmission schedule guarantees the delivery of data to all destinations before the links used for transmission break. And it is implemented until finishing all multicast requests. In other words, we will not change our scheduling policy as the time pass by. However, the network condition is different at each time slots due to the mobility of nodes. Some links may breaks and some nodes may move into the transmission range of other nodes. This dynamic condition may lead to the existence of better routing or scheduling policy which can provide lower latency. Therefore, one of the improvements of our work is that the schedule decision in each time slots can be adjustable dynamically according to the network condition we predict in each time slot, which can derive better solution. In addition, one of our assumptions is that the nodes move with constant speed during the period of transmission time. However, in the real environment, the movement of nodes is more variable. Hence, we can consider the acceleration or change of speed and direction in any time slots, which can be more close to the mobility pattern of nodes in ad hoc networks.

In addition, the nodes are distributed in MANET, there is no centralized node which can collect the information of entire network and calculate the scheduling decision. As a result, we can extend our model where the nodes in the network can be organized by the clustering method. Each cluster has a clusterhead which is responsible for computation, information collection, and decision dissemination along a spanning tree reaching to all nodes in the cluster. Also, we can consider the overhead and energy consumption of clustering in our model to improve this work.

References

[1] L. Chan and W. Heinzelman, "A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks," *IEEE Network*, Vol. 21, No. 6, pp. 30-38, November-December 2007.

[2] C. T. Chou, J. Qadir, and J. G. Lim, "Advances and Challenges with Data broadcasting in Wireless Mesh Networks," *IEEE Communication Magazine*, Vol. 45, No. 11, pp. 78-85, November 2007.

[3] X. Chen and J. Wu, "Multicasting Techniques in Mobile Ad Hoc Networks," in *The Handbook of Ad Hoc Wireless Networks*. Boca Raton, CRC Press, pp. 35-50, 2003.

[4] H. Y. Youn, C. Yu, B. Lee, and S. Moh, "Energy Efficient Multicast in Ad Hoc Networks," in *The Handbook of Ad Hoc Wireless Networks* . Boca Raton, CRC Press, pp. 374-385, 2003.

[5] J. E. Wieselthier, G.-D. Nguyen, and A. Ephremides, "Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Mobile Network and Applications*, Vol. 7, No. 6, pp. 481-492, December 2002.

[6] N. Rahnavard, B. N. Vellambi, F. Fekri, "Distributed Protocols for Finding Low-Cost Broadcast and Multicast Trees in Wireless Networks," *Proc. IEEE SECON'08*, pp. 551-559, June 2008.

[7] H. Liu, X. Jia, P.-J. Wan, X. Liu, and F. F. Yao, "A Distributed and Efficient Flooding Scheme Using 1-Hop Information in Mobile Ad Hoc Networks," *IEEE*

Transactions on Parallel and Distributed Systems, Vol. 18, No. 5, pp. 658-671, May 2007.

[8] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. 3rd ACM international symposium on Mobile ad hoc networking & computing*, pp. 194-205, 2002.

[9] J.-P. Sheu, P.-K. Hung, and C.-S. Hsu, "Scheduling of broadcasts in Multihop Wireless Networks," in *The Handbook of Ad Hoc Wireless Network.* Boca Raton, CRC Press, pp. 465-477, 2003.

[10] R. Gandhi, A. Mishra, and S. Parthasarathy, "Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks," *IEEE/ACM Transactions on Networking*, Vol. 16, No. 4, pp. 840-851, August 2008.

[11] A. A. Agashe and S. K. Bodhe, "Performance Evaluation of Mobility Models for Wireless Ad Hoc Networks," *Proc. ICETET'08*, pp. 172-175, July 2008.

[12] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing*, Vol. 2, No. 5, pp. 483-502, 2002.

[13] W. Lou and J. Wu, "Toward Broadcast Reliability in Mobile Ad Hoc Networks with Double Coverage," *IEEE Transactions on Mobile Computing*, Vol. 6, No. 2, February 2007.

[14] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in

Multihop Wireless Mobile Networks," *Mobile Network and Applications*, Vol. 7, No. 6, pp. 441-453, 2002.

[15] M. L. Fisher, "The Lagrangean Relaxation Method for Solving Integer Programming Problems," *Management Science*, Vol. 27, No. 1, pp. 10-21, January 1981.

[16] A. M. Geoffrion, "Lagrangean Relaxation and its Use in Integer Programming," *Mathematical Programming Study*, Vol. 2, pp. 82-114, 1974.

[17] M. Held, P. Wolfe, and H. P. Crowder, "Validation of Subgradient Optimization," *Mathematical Programming*, Vol. 6, pp. 62-88, 1974.

簡　歷

姓　名：李培維

出 生 地：台 灣　新竹市

生　日：中 華 民 國 七 十 四 年 一 月 二 十 七 日

學　歷：九 十 二 年 九 月 至 九 十 六 年 六 月

國 立 中 央 大 學　資 訊 管 理 學 系 學 士

九 十 六 年 九 月 至 九 十 八 年 七 月

國 立 台 灣 大 學　資 訊 管 理 學 研 究 所 碩 士