

EFFICIENT MULTICAST DESIGN FOR WIRELESS MESH NETWORKS

By

Guokai Zeng

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Computer Science

2011

ABSTRACT

EFFICIENT MULTICAST DESIGN FOR WIRELESS MESH NETWORKS

By

Guokai Zeng

Wireless mesh networks (WMN) have emerged as an efficient means to expand the wireless reach of metro broadband deployments at a variety of locations or scenarios. It provides high quality service to end users as the last-mile of the Internet. Multicast provides an efficient mechanism for distributing data among a group of nodes, such as online games and video conferences. With the increasing popularity of content distribution and multimedia applications, efficient multicast communication becomes essential for the wide deployment of WMNs. In this dissertation, we discuss several research topics related with multicast communication in WMNs.

First, we investigate the problem of routing and channel assignment for multicast communication in link-homogeneous WMNs with the goal of maximizing throughput. In this dissertation, we consider WMNs equipped with multiple channels and multiple interfaces. Previous research work on multicast does not take the multi-channel characteristic into consideration, thus it cannot fully explore the network capacity of WMNs. By exploiting multi-channel and multi-interface, our proposed approach has two major steps: (i) it builds an efficient multicast tree that minimizes the number of relay nodes; and (ii) the dedicated channel assignment strategies are designed to

reduce the interference to improve the network capacity. We demonstrate that our proposed protocols not only improve the throughput, but also reduce the delay.

Second, we study the multicast problem in link-heterogeneous WMNs. Unlike previous work that focuses on link-homogeneous WMNs only, we consider one important form of link heterogeneity: different link loss ratios. Under this constraint, although minimizing relay nodes helps to decrease interference in the WMN, it is also important to choose high quality links to minimize the number of transmissions. This is because decreasing the number of transmissions helps to increase the throughput. Based on this consideration, we define a new graph theory problem: HW-SCDS to model link-heterogeneity. Maximizing WMN throughput is equivalent to computing a minimum HW-SCDS (MHW-SCDS) in the graph. We prove that computing an MHW-SCDS is NP-hard and devise a greedy algorithm for it. We show that our approach outperforms previous work in terms of throughput and delay.

Third, we investigate the problem of opportunistic multicast in WMNs. By exploiting the broadcast nature and spatial diversity of the wireless medium, opportunistic routing has emerged as a new routing paradigm to improve unicast throughput. However, its natural multicast extension does not build any efficient multicast structure, thus the explosion of unnecessary retransmission is unavoidable. To overcome this shortcoming, we propose a new opportunistic multicast protocol to improve throughput in WMNs. The key concept is a tree backbone in this protocol. Our tree backbone protocol represents a tradeoff between traditional structured multicast protocols and unstructured protocols. Therefore, our solution is able to improve throughput by both utilizing spatial diversity and reducing transmissions.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisors Dr. Li Xiao and Dr. Matt Mutka for their patient supervision. Without their scientific guidance and financial support, this dissertation could not have come into light.

I would like to thank Professor Abdol-Hossein Esfahanian, Professor Tien-Yien Li for serving on my advisory committee. Their insightful comments on my research are greatly appreciated. Special thanks go to Professor Eric Torng for his valuable advisory.

I am grateful to my colleagues in eLANs lab and CSE department: Dr. Chen Wang, Dr. Bo Wang, Dr. Yong Ding, Yuanteng Pei, Pei Huang, Dr. Xiaomei Liu, Dr. Zhiwei Cen, Dr. Haibing Cheng, Yi Huang, Dr. Moonseong Kim, Yang Yang, Jun Huang, and others. I also would like to thank friends at Michigan State who have offered me generous help: Dr. Yang Fang, Dr. Hua Deng, Dr. Chao Li, Hui Wang, Yan Gao, Dr. Fang Xie, Dr. Li Li, Dr. Feng Song, Dr. Wei Zou, Dr. Fei Zheng, Dr. Zongliang Cao, Qi Zhu, Dr. Yuping Huang, Dr. Deng Zhang, Dr. Lei Huang, and many others.

Finally I would sincerely thank my parents Guangcheng Zeng and Ruiying Pan, and Ms. Hanzi Li. It is their support and love that encouraged me to finish this dissertation successfully.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
1 Introduction	1
1.1 Background	2
1.2 Challenge and Motivation	5
1.2.1 Multicast in Multi-Channel Wireless Mesh Networks	5
1.2.2 Multicast in Link-Heterogeneous Wireless Mesh Networks	7
1.2.3 Opportunistic Multicast Routing in Wireless Mesh Networks	8
1.3 Structure of the Content	9
2 Related Work	11
2.1 Multicast in Wireless Mobile Ad Hoc Networks	11
2.2 Multicast in Wireless Mesh Networks	12
2.3 Channel Allocation in Wireless Mesh Networks	13
2.4 Opportunistic Routing in Wireless Networks	14
2.5 Multi-rate Routing	15
2.6 Other Related Work	16
3 Multicast Algorithms for Multi-Channel Wireless Mesh Networks	18
3.1 System Model	21
3.1.1 Basics	21
3.1.2 Measuring Partial Overlap	23
3.1.3 Design Consideration	25
3.2 Level Channel Assignment Algorithm	27
3.3 Multi-Channel Multicast Algorithm	31
3.3.1 Multicast Structure Construction	31
3.3.2 Channel Assignment	37
3.4 Further Discussion on MCM Algorithm	40
3.4.1 Repair Mechanism	40
3.4.2 Channel Assignment with More Interfaces	44
3.4.3 Multiple Gateways	45
3.5 Simulations	48
3.5.1 Impact of Network Size	49
3.5.2 Impact of Number of Channels	51
3.5.3 Impact of Transmission Rate	52
3.5.4 Delay Comparison	52

3.5.5	2 Interfaces vs 4 Interfaces	55
3.5.6	Partially Overlapping Channel Assignment Test	57
3.5.7	Multiple Gateways vs Single Gateway	59
3.6	Summary	60
4	Multicast Algorithms for Link-Heterogeneous Wireless Mesh Networks	61
4.1	Link Heterogeneity in Wireless Mesh Networks	63
4.1.1	Existence of Heterogeneous Links in Mesh Networks	63
4.1.2	Effects of Heterogeneous Links on Throughput	64
4.2	Minimum HW - SCDS	68
4.2.1	Basic Definitions	68
4.2.2	NP-Hardness Proof	71
4.2.3	Greedy Algorithm	73
4.3	Application of MHW-SCDS and Further Discussion	77
4.3.1	Different Link Loss Ratios	77
4.3.2	Channel Assignment	78
4.3.3	Repair Mechanism	80
4.3.4	Link Fluctuation	84
4.4	Performance Evaluation	87
4.4.1	Throughput	88
4.4.2	Delay Comparison	92
4.5	Summary	94
5	Efficient Opportunistic Multicast via Tree Backbone for Wireless Mesh Networks	95
5.1	Background and Motivation	98
5.1.1	System Model	98
5.1.2	Basic of Opportunistic Routing	99
5.1.3	Motivation Example	100
5.2	Opportunistic Multicast in Single-rate WMNs	101
5.2.1	Basic Idea	102
5.2.2	NP-Hardness Proof	104
5.2.3	Heuristic Algorithms	106
5.3	Euclidean Opportunistic Multicast in Multi-rate WMNs	108
5.3.1	Design Consideration	109
5.3.2	Euclidean Tree Backbone and Rate Selection	110
5.4	Companion Mechanism	114
5.4.1	Routing Information Distribution	114
5.4.2	Membership Update	115
5.5	Simulations	117

5.5.1	Impact of Network Size	118
5.5.2	Delay Comparison	120
5.5.3	Impact of Multiple Rates	120
5.6	Summary	123
6	Conclusion and Future Work	124
6.1	Conclusion	124
6.2	Future Work	127
6.2.1	Fair Rate Allocation for Opportunistic Routing in Wireless Mesh Networks	127
6.2.2	Multicast Application in Cognitive Wireless Mesh Networks .	131
	BIBLIOGRAPHY	136

LIST OF TABLES

4.1	Throughput Comparison	88
-----	---------------------------------	----

LIST OF FIGURES

1.1	Wireless Mesh Network (For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation)	3
2.1	Multicast in MANETs [1]	12
3.1	Experiment	24
3.2	Interference Factor vs Channel Separation	25
3.3	An Example for LCA and Tree Mesh	29
3.4	Relay Node Search Example	35
3.5	A Multicast Structure Example	37
3.6	Ascending Channel Allocation Example	39
3.7	Repair Mechanism Example	42
3.8	Example of 4-Interface Channel Allocation	44
3.9	Example of Multiple Gateways	46
3.10	Impact of Network Size	50
3.11	Impact of Number of Channels	51
3.12	Impact of Transmission Rate	53
3.13	Delay Comparison	54
3.14	2-interface vs 4-interface	56
3.15	Partially Overlapping Channel Test	58
3.16	Multiple Gateway Test	59
4.1	Sensitivity of Link Bandwidth to Node Location	65
4.2	Local Multicast Example	66
4.3	An Example for Two Different Multicast Trees	67
4.4	Greedy Algorithm Example	76
4.5	Redundant Relay Node	77

4.6	ETX Increment Example	81
4.7	Node Failure Repair Example	83
4.8	Impact of Number of Receivers	89
4.9	Impact of Number of Channels	91
4.10	Impact of Delivery Ratio	93
4.11	Delay Comparison	93
5.1	Motivation Example	99
5.2	ST-B Example	107
5.3	Relation between Distance and Transmissions	111
5.4	EOM Example	112
5.5	Impact of Network Size	119
5.6	Delay Comparison	121
5.7	Impact of Multiple Rates	122

CHAPTER 1

Introduction

Wireless mesh networks (WMNs) have emerged as a key technology for next-generation wireless networking. They are undergoing rapid development in various inspiring applications. Used as infrastructures to enhance wireless coverage and last-mile Internet access, WMNs are self-organized and self-configured. WMNs can be used at many different scenarios, one of which is to build a metropolitan network to gain internet access without wired option [2]. WMNs have to deal with the huge data traffic, because satisfying the requirement of end users is paramount for WMNs.

Multicast applications widely exist in wireless mesh networks. Some users want to retrieve the same data from the Internet. For example, people may watch FIFA world cup on the Internet at the same time. Efficient multicast technology plays an important role in WMNs, because it provides an efficient mechanism for distributing data among a group of nodes. In this dissertation, we investigate the multicast communication in WMNs, and design efficient multicast protocols for WMNs.

In this chapter, we first introduce the background of wireless mesh networks. We

then illustrate the motivation and research challenge for the multicast communication in WMNs.

1.1 Background

Wireless mesh networking (WMN) is a promising technology for building broadband wireless access networks [3, 4, 5, 6]. They provide a cost-efficient solution for the broadband Internet access of community or enterprise users. In such networks, most of the nodes are either stationary or minimally mobile. Because they usually have a permanent power supply, they do not have to worry about energy efficiency. Compared with their single-hop counterpart, wireless LANs, wireless mesh networks are self-organized with the nodes automatically establishing ad hoc networks and maintaining the connectivity. Thus, they are able to provide more reliability as well as larger coverage, and less equipment costs. Commercial deployments of WMNs have been used on the last mile for extending or enhancing Internet connectivity, such as MIT Roofnet [7], Seattle Wireless [2], and others[8, 9, 10].

Mesh networks are composed of three types of nodes: gateways, mesh routers and mesh clients. Gateways enable the integration of WMNs with various other networks including the Internet. As dedicated devices providing stable high throughput for mesh clients, mesh routers have minimal mobility and powerful computation ability. While mesh routers provide coverage in the neighborhood, they also connect with each other to form a mesh backbone. In order to further improve the flexibility and capacity of WMNs, one typical approach is to equip mesh routers with multiple

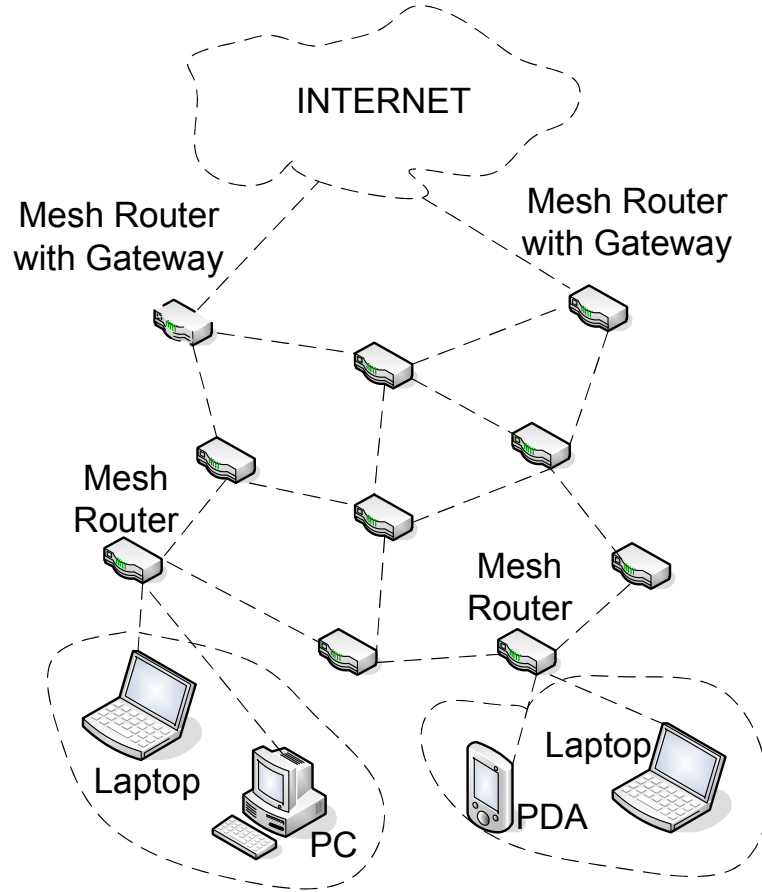


Figure 1.1. Wireless Mesh Network (For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation)

wireless interfaces and multiple channels. As we know, two adjacent transmissions can be scheduled at the same time if they are working on the non-overlapping channels. Thus, the multiple channel and multi-interface characteristic of WMNs creates the possibility of increasing network capacity. The mesh clients are usually end users, such as desktops, laptops and PDAs, which access the Internet through the mesh routers. Thus, the mesh clients are usually considered to be within one-hop of the mesh routers.

Wireless mesh networks have been used for some scenarios. Their typical appli-

cations include the following aspects:

- **Internet access:** The individual users of wireless mesh networks are usually interested in accessing the Internet to get important timely information and make life more convenient. Thus, wireless mesh networks provide a cost-efficient way for the communities to access Internet. A typical WMN deployment would be to mount mesh routers outside of windows or on roofs. Some mesh routers are connected to the Internet, and client devices are connected to mesh routers. So the whole community can share the Internet access. By replacing wired connection with wireless connection, not only can the coverage be enhanced, but the network also becomes more flexible.
- **Distributed Information Storage and Sharing:** The users of wireless mesh networks may be willing to store high-volume data in disks owned by other users, download files from other users or distributed database servers based on P2P mechanisms, or communicate with others on video phones. The information packets are relayed among the mesh backbone (that is formed by mesh routers) without through the Internet gateway. Thus, sharing network resource and interacting with each other among end users become more convenient.
- **Other Applications:** Wireless mesh networks are also used for health and medical systems, public transportation systems, and wilderness surveillance systems. All the applications clearly demonstrate the promising market of wireless mesh networks.

1.2 Challenge and Motivation

In this section, we will discuss the research challenge and motivation of multicast for wireless mesh networks. Firstly, the general motivation and challenge of multicast communication in WMNs is introduced. Secondly, we extend our work to link-heterogeneous wireless mesh networks. Thirdly, we illustrate the motivation and challenge when we apply opportunistic routing to multicast application in WMNs.

1.2.1 Multicast in Multi-Channel Wireless Mesh Networks

Wireless mesh networks share some features with ad hoc networks in the following aspects: (i) they are both multi-hop wireless networks, and (ii) they lack of a wired infrastructure. However, wireless mesh networks have their own specific features, which poses different challenges to the researchers for multicast in WMNs.

First, WMNs are designed for providing good service of broadband Internet access and sharing network resource for community and enterprise users, thus network capacity and throughput are the major concern of wireless mesh networks. At the same time, multicast communication is a resource-consuming application because it transmits the packets from the source to multiple destinations. The need to increase multicast throughput in WMNs to satisfy the requirement of end uses is paramount for WMNs. It is different from multicast in ad hoc networks that cares more about route discovery due to short-lived links caused by the node movement.

Second, traditional multicast protocols for wireless networks assume that each node is equipped with one interface, while WMNs can provide the nodes with multiple

interfaces that can be used to improve the throughput substantially. However, channel assignment is subject to the number of available channels and interfaces, the network topology, and the communication requests. An inappropriate channel assignment strategy will result in throughput reduction due to the multi-channel hidden terminal problem [11], disconnection of the topology [12], or unfair bandwidth allocation to various users [13]. Therefore, efficient channel assignment on the multicast structure proposes a major research challenge for multicast in WMNs.

Third, WMNs have relatively fixed wireless infrastructure. Mesh routers have limited mobility and provide stable, high throughput connectivity for mesh clients, thus they form an infrastructure for mesh clients. The infrastructure can be built by using various types of radio technologies, and the links are self-configuring and self-healing. Thus, WMN topologies are often deemed as static. On the contrary, ad hoc networks have no such kind of infrastructure, and the connectivity relies on individual client nodes to perform routing functions. Since the client nodes have high mobility, it is difficult to achieve high connectivity and throughput. However, wireless mesh networks have such advantage over ad hoc networks, they have more demanding network performance.

Fourth, the mesh routers are usually rechargeable, thus the energy efficiency is not a major concern in WMNs. In ad hoc networks, because the node cannot be recharged due to the movement, the goal of power saving greatly affects its network design.

In conclusion, the efficient multicast in WMNs should follow the general design consideration: (i) maximize the multicast throughput is the major design objective,

(ii) effective channel assignment is one important phrase of the multicast protocol design in WMNs, and (iii) the network topology is usually considered as static.

1.2.2 Multicast in Link-Heterogeneous Wireless Mesh Networks

The links in WMNs may have different quality due a variety of reasons, such as environment factors and diverse device abilities. Thus, taking link heterogeneity into consideration is one of the design consideration in this dissertation. As we know, one common approach to increase system throughput is to reduce unnecessary transmission in the network. Given the assumption of link homogeneity, minimizing the number of transmissions in a WMN is equivalent to minimizing the number of relay nodes in the multicast structure. However, individual WMN links may have different qualities in the real world for a variety of reasons. The link quality also determines the number of transmissions. This is because poor links may need more retransmission than good links. Although minimizing the number of relay nodes helps to decrease simultaneous interference in the WMN, it is also important to choose high quality links to minimize the total number of transmissions.

Therefore, by taking the link heterogeneity of WMNs into consideration, we need to build an efficient multicast structure with the minimum number of transmission.

1.2.3 Opportunistic Multicast Routing in Wireless Mesh Networks

Most traditional multicast protocols for WMNs discover the least cost or highest throughput paths to reach the destinations. Compared with their wired counterparts, they build efficient multicast structures based on the wireless communication facts: (i) the local broadcast enables multiple neighbors to receive the packet, (ii) the packet loss ratios cannot be ignored, and (iii) the packet delivery ratios are not the same on different links. Thus, they usually choose one or multiple next-hop destinations for each relay node, and the links between selected one-hop neighbors have good quality in the multicast structure.

However, these protocols do not fully take advantage of the spatial characteristics of wireless communication. It is unknown which neighbors can receive the packet in the current transmission. At the same time, some research work [14, 15] has demonstrated that utilizing the cooperative diversity to send packets through multiple relay nodes concurrently can further improve the system throughput, including the multicast throughput [16]. This routing strategy is known as opportunistic routing (OR).

The natural multicast extension of OR is discussed in [16], but it does not build any efficient multicast structure that is able to reduce the packet transmission. The increase of the number of transmissions not only consumes more bandwidth resource, but also leads to more local interference among nearby transmissions, which may result in lower throughput. Thus, in order to further improve the throughput, we

have to reduce the number of transmissions when we exploit opportunistic routing for multicast.

Furthermore, the recent trend in wireless communication is to enable devices with multiple transmission rates [17, 18, 19]. Generally speaking, low-rate communication provides a long transmission range, while high-rate has to occur at a short scope. The variance of transmission range implies the variance of the neighboring node set, so it leads to different spatial opportunities. The inherent rate-distance tradeoff for opportunistic unicast routing has had impact on performance [18]. It is intuitive to expect that the trade-off also affects opportunistic multicast.

Therefore, in order to achieve higher multicast throughput, the opportunistic multicast protocol in WMNs should have the following characteristics: (i) it builds upon opportunistic routing strategy to exploit spatial reuse, (ii) an efficient tree backbone structure should be built to minimize the number of transmissions, and (iii) it must take multi-rate into consideration.

1.3 Structure of the Content

The rest of this dissertation is organized as follows: Chapter 2 introduces the related work in wireless mesh networks. Chapter 3 presents the multi-channel multicast algorithms for wireless mesh networks, where link qualities are homogeneous. Chapter 4 discusses the link-heterogeneity in WMNs, and an efficient multicast algorithm is proposed to improve throughput. Chapter 5 applies the opportunistic routing paradigm to multicast applications in WMNs. Finally, Chapter 6 concludes the dissertation

and discusses possible future work.

CHAPTER 2

Related Work

Much research work has been proposed to address the multicast issues in wired networks and MANETs, but a few focus on wireless mesh networks. In this chapter, we first survey the multicast protocols proposed for MANETs, which is close to WMNs. We then briefly introduce the related work in WMNs, including multicast strategy, channel assignment, opportunistic routing and multi-rate routing. Afterwards, we also briefly introduce the other related work mentioned in this dissertation.

2.1 Multicast in Wireless Mobile Ad Hoc Networks

The multicast protocols in Mobile Ad Hoc Network (MANET) can be categorized into three groups: (i) tree-based, (ii) mesh-based, and (iii) stateless protocols. Tree-based protocols build an efficient multicast tree rooted at the source, and the packets are forwarded from the source to the destinations along tree paths [20, 21, 22]. This

method minimizes the total hop count distance and saves bandwidth, but it is not good at dealing with dynamic topology change. At the same time, mesh-based protocols build multiple trees among the multicast group members [23, 24, 25], such that the packets are delivered to each destination through multiple paths. This method makes multicast communication more robust. However, both tree-based and mesh-based protocols have to bear the overhead of creating and maintaining the routing information in the intermediate nodes. In order to address this drawback, stateless protocols are proposed that store the destination list in the packet header. The packets self route to the destinations based on geographical information [26, 27, 28].

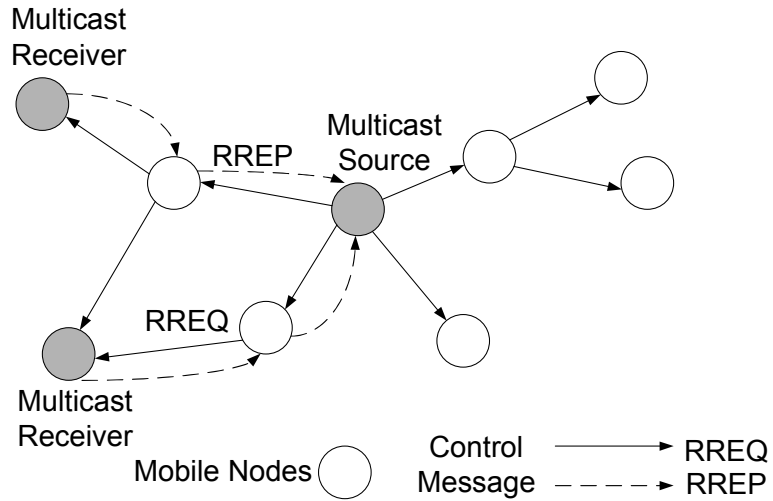


Figure 2.1. Multicast in MANETs [1]

2.2 Multicast in Wireless Mesh Networks

Compared with unicast routing that has been intensively studied in WMNs, the multicast communication draws less attention in the literature of mesh networks.

Efficient multicast, which cannot be readily achieved through combined unicast or simplified broadcast, is essential to wireless mesh networks and is worthy of thorough investigation.

Compared with traditional wireless multi-hop networks, multicast technology in WMNs focuses more on maximizing throughput [29]. Some research work emphasizes on the network coding technique for multicast [30, 16, 31, 32, 33, 34], and it develops game theoretic methods based on interference management [34, 35, 36]. Several efficient and distributed solutions are then derived and illustrated from the cross-layer framework. The research work in [37, 38] demonstrates that complex robustness protocols can be very expensive and may significantly lower the congestion point. At the same time, they provide simple mechanisms that can improve multicast performance. By doing extensive simulations and experiments, the authors in [39, 40, 41] design and prove that multicast protocols based on multi-channel and multi-interface can greatly increase throughput. Different multicast protocols are tested in experiments [42], and insights into the performance and recommendations for suitable routing approaches are provided.

2.3 Channel Allocation in Wireless Mesh Networks

Effective channel assignment is able to greatly decrease the nearby interference, and hence significantly improve the multicast throughput in wireless mesh networks.

Thus, multi-channel issues in wireless networks have drawn much attention in recent years.

Several link layer and MAC layer protocols have been proposed to improve the performance of wireless networks [11, 43, 44, 45, 46, 47]. These approaches are used to find the optimal channel for the current packet transmission for essentially avoiding interference. Such schemes have the key advantage that a single radio is required to support multiple channels. Some researchers aim to derive the lower bound or upper bound of the capacity in terms of achievable QoS in mesh networks [13, 48]. Many studies focus on how to assign channels to nodes in the network, either by static [12, 49] or dynamic methods [50, 51, 52, 53, 54]. They develop a set of centralized or distributed algorithms for channel assignments by taking the bandwidth cost, efficient routing, and load-balance into account. Most of them believe that static assignment outperforms dynamic assignment due to the channel switching cost and the delay. New metrics have also been proposed for multihop wireless networks with considering the impact of channel interference, which is used to find high throughput paths between sources and destinations [55, 37, 56].

2.4 Opportunistic Routing in Wireless Networks

In recent years, opportunistic routing has become an interesting topic that improves the throughput and the transmission reliability in the face of unreliable wireless links [14]. Some variants of opportunistic routing are proposed to improve throughput under different situations. The routing strategies in [30, 16, 31, 32, 57] aim to com-

bine network coding with opportunistic routing in a natural fashion, so that they can achieve the cooperative spatial diversity. The authors in [58, 59] extend OR to enable devices with only one interface to operate on multiple channels, which reduces interferences. The forwarding candidate set and relay priority are defined based on the nodes' geographic information [60, 61, 62]. For example, GeRaf [60] defines sets of regions with nodes closest to the destination with highest priority. In the ad hoc context, there are relevant works to consider, which takes advantage of diversity offered by multiple users [63].

2.5 Multi-rate Routing

One of current wireless technical trends is that modern wireless devices are able to utilize multiple transmission rates to accommodate a wide range of conditions. In [17, 19], the authors investigate the impacts of several factors on the carrier sensing threshold in the multi-rate wireless networks, and propose the bandwidth distance product as a routing metric to improve throughput. In order to utilize multiple channels in multi-rate networks, the data rate adaptive channel assignment algorithm is proposed in [64], which assigns links having the same data rates on the same channel to minimize the wastage of channel resources. The problem of achieving good scheduling for small delay in multi-rate multi-channel wireless networks is studied in [55]. The iterative resource allocation rule provides a very good delay performance to the users, in addition to network stability. A routing metric towards high throughput is proposed in [65]. The impact of multiple rates, interference, and candidate selection

in opportunistic routing is analyzed in [18, 66, 67], and rate selection schemes are also proposed.

2.6 Other Related Work

Most tree-based multicast protocols establish some form of connected distribution structures called *Virtual Multicast Backbone (VMB)*, which spans all multi-receivers and contains all the relay nodes. Steiner Connected Dominating Set is considered to be reasonable to model VMB. The Minimum Steiner Connected Dominating Set is proposed in [68] as the generalization of the well-know Minimum Connected Dominating Set (MCDS) problem [69]. Both MSCDS and MCDS are NP-hard problems, and some approximation algorithms have been presented [68, 69, 70, 71]. So far, the best approximation algorithm for MSCDS, which is based on the greedy scheme, achieves a performance ratio of $(c + 1)H(k) + c - 1$.

Recently, researchers have also paid attention to the heterogeneous characteristics of wireless networks. The theoretical paper [72] discussed the generalized connectivity problem in the directed Steiner network, which can be applied to a multicast problem by restricting some conditions. The multicast with QoS support in heterogeneous networks is studied in [73], which helps to reduce service time and packet loss rate. However, both of the above two papers do not consider the local broadcast characteristics we point out in this chapter. Two localized topology control algorithms for heterogeneous wireless multi-hop networks to keep the network connectivity have been proposed in [74]. Simunic, et al. [75] have developed integrated approaches for

the management of power and performance to mobile users in heterogeneous wireless environments.

CHAPTER 3

Multicast Algorithms for Multi-Channel Wireless Mesh Networks

Wireless mesh networking (WMN) is a promising technology for building broadband wireless access networks. In wireless mesh networks, most of the nodes are either stationary or minimally mobile and do not have power constraints. Compared with their single-hop counterpart, wireless LANs, WMNs are self-organized with the nodes automatically establishing ad hoc networks and maintaining their connectivity. This provides improved reliability as well as larger coverage and reduces equipment cost.

Wireless mesh networks considered in this dissertation are characterized by the use of multiple channels and multiple interfaces to improve system throughput. Recent research has focused on how unicast routing assigns channels to different wireless

interfaces to improve system throughput in WMNs. However, multicast communication, which intends to transmit the packets from the source to a set of nodes, draws less attention in the literature of mesh networks. We believe that efficient multicast, which cannot be readily achieved through combined unicast or simplified broadcast, is essential to WMNs and is worthy of thorough investigation. It is often necessary for a portion of end users to retrieve data packets from the Internet. For example, a large number of users may watch the FIFA World Cup on the Internet. The gateway that helps to connect the mesh network with the Internet can effectively multicast the data packets to those users.

Efficient multicast protocols in WMNs cannot be achieved by adopting or slightly modifying the multicast protocols for other types of multi-hop wireless networks. Unlike mobile ad hoc networks or WSNs, route recovery or energy efficiency is not the major concern for mesh networks due to the limited mobility and the rechargeable characteristic of mesh nodes. Moreover, supporting the potential major applications, such as Video On Demand, poses a significant challenge for the limited bandwidth of WMNs. Thus, it is necessary to design an effective multicast algorithm for mesh networks.

Traditional multicast protocols for wireless networks assume that each node is equipped with one interface. A mesh network we considered provides the nodes with multiple interfaces that can be used to improve the throughput substantially. However, channel assignment is subject to the number of available channels and interfaces, the network topology, the communication requests, and other factors. Interference cannot be completely eliminated due to the limited number of available channels. An

inappropriate channel assignment strategy will result in throughput reduction due to the multi-channel hidden terminal problem [11], disconnection of the topology [12], or unfair bandwidth allocation to various users [13].

In this chapter, we aim to design a multicast protocol for mesh networks that has the following characteristics: (i) it improves the system throughput by allowing simultaneous close-by transmissions with multi-channels and multi-interfaces, and (ii) it assigns all the available channels to the interfaces instead of just the non-overlapping channels.

We propose a Level Channel Assignment (LCA) algorithm and a Multi-Channel Multicast (MCM) algorithm to optimize throughput for multi-channel and multi-interface mesh networks. The algorithms first build a multicast structure by minimizing the number of relay nodes and hop count distances between the source and destinations, and use dedicated channel assignment strategies to improve the network capacity by reducing the interference.

Our design builds a new multicast backbone - “tree mesh”, which partitions the mesh routers into different levels based on the Breadth First Search (BFS), and then heuristically assigns channels to different interfaces. Tree-based multicast is well established in wireless networks for its data forwarding efficiency over other types of approaches at the expense of low robustness. However, unlike MANETs, WMNs are normally considered stationary and always put throughput maximization as the first priority. Thus, tree-based multicast is suitable for WMNs since the topology change is not a major concern in WMNs. Simulations show that our algorithms greatly outperform the single-channel multicast algorithm. We observe that MCM achieves

better throughput and shorter delay while LCA can be implemented in a distributed manner.

The rest of this chapter is organized as follows. Section 3.1 describes the system model and the design consideration. Section 3.2 proposes an intuitive algorithm, the LCA algorithm, which is easy to implement but has drawbacks. Section 3.3 introduces the MCM algorithm to build a more efficient multicast structure, which is followed by the description of how to assign channels on it. Several companion mechanisms for our protocols are presented in Section 3.4. Section 3.5 presents the simulation results, and the last section summarizes this chapter.

3.1 System Model

We start from the underlying network model by introducing some basic terminology and the partial channel conflict phenomena, which is followed by design considerations for multicast algorithms in WMNs.

3.1.1 Basics

Mesh networks are composed of three types of nodes: gateways (access points), mesh routers, and mesh clients. Gateways enable the integration of WMNs with various other networks, including the Internet. As dedicated devices providing stable high throughput for mesh clients, mesh routers have minimal mobility and form the mesh backbone. In order to further improve the flexibility and capacity of WMNs, one typical approach is to equip the mesh routers with multiple wireless interfaces. As a

result, two transmissions of two nearby pairs can be simultaneously scheduled if non-overlapping channels are assigned. Mesh clients are usually end users, such as laptops and PDAs, which access the Internet through the mesh routers, so that the mesh clients are usually within one-hop of the mesh routers. Since the multicast packets are always relayed among the mesh backbone, we only consider how to transmit the packets to multiple mesh routers; then packets will be forwarded one more hop to the corresponding mesh clients that desire to receive the packets.

To simplify the system model, we consider the network as a graph $G = (V, E)$, where V represents the set of gateways and mesh routers, and E represents the physical links among neighboring nodes (the node refers to the mesh router or the gateway in the subsequent sections). We assume that each node has the same fixed communication range, that is, if node u can transmit directly to node v (and vice versa), there is a link (u, v) in E .

The number of available channels is limited in the current network protocols. In addition, each node is able to be equipped with k ($k \geq 2$) Network Interface Cards (NICs), any of which can be tuned to any available channel. Multi-channel and multi-interface characteristics enable more concurrent transmissions. When one NIC is transmitting or receiving packets on one channel, another NIC on the same node is able to undertake transmission on another different channel at the same time. The value of k usually equals to 2, 3, or 4 due to economical reasons. In this chapter, we first consider the situation that each node has 2 interfaces, then we apply our algorithms to more interfaces.

3.1.2 Measuring Partial Overlap

To improve the throughput of WMNs, many studies have been conducted on how to assign orthogonal channels to adjacent wireless links to minimize interference. It is known that 802.11b/g and 802.11a provide 3 and 12 non-overlapping channels [76], respectively. Although 802.11a provides more non-overlapping channels than 802.11b/g, it has several drawbacks. Because 802.11a works on a higher frequency spectrum (5GHz) than 802.11b/g (2GHz), it is more difficult to penetrate walls and other obstructions, and thus 802.11a has a shorter range. In addition, the interfaces and access points for 802.11a are more costly. As a result, 802.11b/g is more commonly used.

Previous channel assignment algorithms for 802.11b/g only use three non-overlapping channels: 1, 6, and 11. In these studies, a binary interference model is usually assumed, that is, if two links are within interference range of each other, they will interfere with each other if they are on the same channel, and otherwise not. However, the interference can be further reduced by using the partially overlapping channels too, that is, by using any channel from 1 to 11 in the channel assignment.

Through experiments, we observe that the interference between two links depends on both their physical distance and channel separation [77]. Unlike the traditional interference model, the interference range is no longer a constant. Instead, it varies with the channel separation. Let I_c be the interference range of two links with channel separation c . That means, when the channel separation of two links is c , they will interfere with each other if their distance is less than I_c , and otherwise not. For

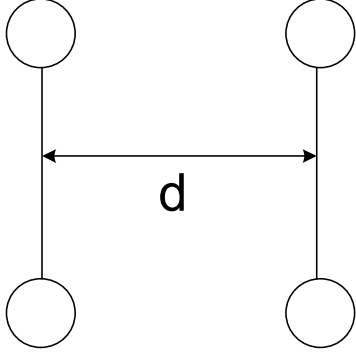


Figure 3.1. Experiment

example, $I_0 = 2R$, which means the same channel can be used on two links without any interference only when they are over twice the transmission range away. In [77, 78, 79, 80], experiments have been done to measure the interference between two wireless AP-Client links with different distances and channel separations.

Definition 1 *Interference Factor* *is defined as the ratio of the interference range to the transmission range. We use δ_t to represent the Interference Factor when the channel separation of two links is t .*

We perform experiments to measure the interference factor between two wireless links. We use 4 laptops with Netgear WAG511 PC Cards, each two of which construct a separate wireless link as shown in Fig. 3.1. We evaluate the interference between two links by comparing the total throughput when both links are active and the sum of each link's throughput when the other link is turned down. The length of each link is fixed at 5m. Linux kernel with Madwifi is used to drive the network cards. The two end nodes of the link work on the same channel. We configure the two links with different channels, and we vary the distance d between the two links to find out the

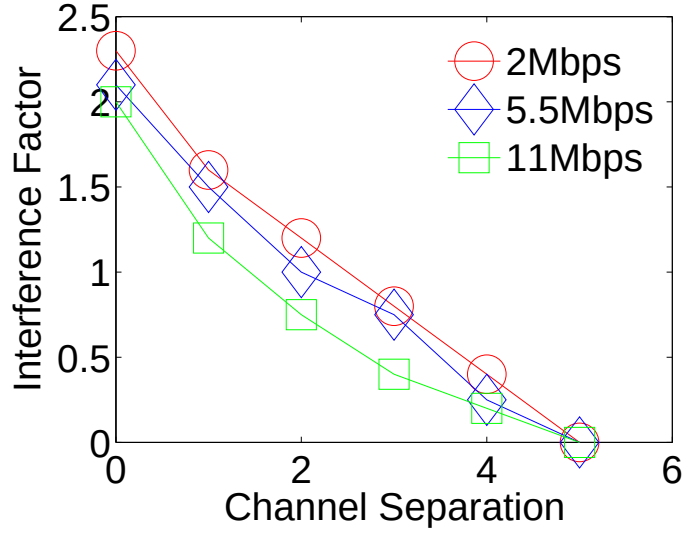


Figure 3.2. Interference Factor vs Channel Separation

interference range.

We found a similar trend in our experiments, that is, interference range decreases with the increase of channel separation. The results of these experiments are shown in Fig. 3.2. Therefore, if we can fully utilize these partially overlapping channels, we can further decrease the total interference in the network, and thus improve the network throughput.

3.1.3 Design Consideration

Routing protocols do exist to offer efficient multicasting service for conventional multi-hop wireless networks, such as MANETs and WSNs. Since the nodes become increasingly mobile or the networks only have scarce resources such as power constraints and limited computing ability, most previous work pays much attention to energy efficiency and how to build the multicast structure without knowing the global topol-

ogy. As a result, the multicast structure should be distributedly constructed, energy efficient, and should take care of the topology change as well as group member management, which may conflict with maximizing the throughput of the network to some extent.

However, since mesh networks are deployed to provide last-mile Internet access for enterprises or communities, the throughput and the network capacity are the major concerns. Deployed at the fixed locations, mesh routers have limited mobility. Furthermore, they are computationally powerful and do not rely on battery power compared with their counterparts in MANETs or sensor networks, which helps to achieve sufficient network capacity to meet the requirement of applications such as audio or video sharing among end users. Thus, we need to create a multicast structure that aims to deliver the packets rapidly to the multi-receivers (multi-receivers are defined as the multicast group members except for the source node) without worrying about the energy consumption and topology changes.

Moreover, equipping the mesh routers with more than one wireless interface could further improve the network capacity. The assignment of channels to interfaces on the multicast structure is also essential to throughput optimization. Inappropriate channel allocation will lead to topology disconnection and exacerbation of multi-channel hidden terminal problems, which reduces the system throughput. Therefore, both efficient multicast structure and effective channel assignment play important roles in mesh network multicast.

3.2 Level Channel Assignment Algorithm

A common method for multicast is to build a multicast tree, where the source node is usually the access point in this chapter. We first propose the Level Channel Assignment (LCA) algorithm, which can be achieved by the following steps.

First, the nodes obtain their level information. The breadth first search (BFS) is used to traverse the whole network. All the nodes are partitioned into different levels according to the hop count distances between the source and the nodes.

Definition 2 *if node a (in level i) and b (in level $i + 1$) are within each other's communication range, then a is called the **parent** of b , and b is called the **child** of a .*

Second, we build a multicast tree based on the node level information. Initially, the source and all the receivers are included in the tree. Then, for each multi-receiver v , if one of its parents is a tree node, then connect it with that parent, and stop. Otherwise, randomly choose one of its parents, say f_v , as relay node on the tree, and connect v and f_v . Afterwards, we try to find out the relay node for f_v recursively. This process repeats until all the multi-receivers are included in the multicast tree. Algorithm 1 gives the detail.

Next, the tree nodes decide their channel assignment with the level information.

1. The source node (level 0) only uses one interface, which is assigned channel 0.

This interface is responsible for sending packets to the tree nodes in level 1.

2. The internal tree node in level i ($i \geq 1$) uses two interfaces: one is assigned channel $i - 1$, which is used to receive packet from the upper level, the other is

assigned channel i , which is used to forward packets to tree nodes at level $i + 1$

3. The leaf in the level i ($i \geq 1$) uses two interfaces: one uses channel $i - 1$ to receive the packets from level $i - 1$, the other uses channel i to forward the packets to the mesh clients within its communication range who desire to receive the packets .

Algorithm 1: Multicast Tree Construction for LCA

Data: M : multi-receivers; s : source node;

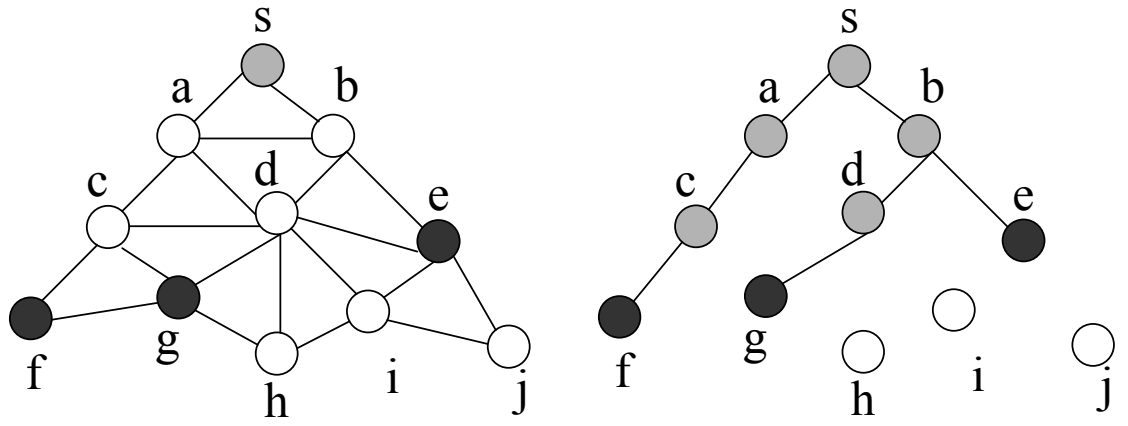
Result: T : multicast tree

```

1  $V(T) = M \cup \{s\}; E(T) = \emptyset;$ 
2 for  $\forall$  node  $v \in M$  do
3    $p = v;$ 
4   while none of  $p$ 's parents is included in  $V(T)$  do
5     Randomly select one of  $p$ 's parents, say  $f_p$ .
6      $V(T) = V(T) \cup \{f_p\};$ 
7      $E(T) = E(T) \cup \{(p, f_p)\};$ 
8      $p = f_p$ 
9    $E(T) = E(T) \cup \{(p, f'_p)\}$  ( $f'_p$  is the parent of  $p$ , and it is a tree node)

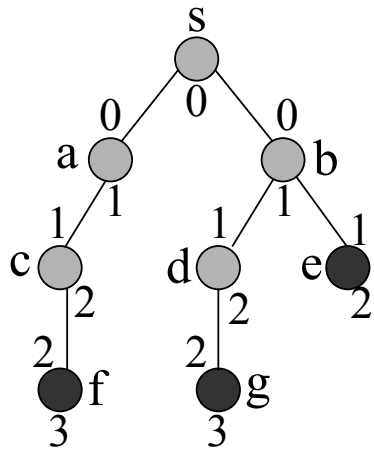
```

One example is shown in Fig. 3.3, where node s is the source, and e, f, g are the multi-receivers. Initially, $\{s, e, f, g\}$ are included in the multicast tree. At first, since none of g 's parents are tree nodes, randomly select one parent d as a tree node and connect g with d . We then choose d 's parent b as a tree node and connect d with b . Since b 's parent s is a tree node, we connect b with s and stop the process for including g in the multicast tree. Next, we start from the second multi-receiver e . Connect e with its parent b and stop since b is already a tree node. Similarly for the third multi-receiver f , we connect f with c , c with a , and then a with s . Now the tree construction is complete since all the receivers are connected to the tree. The constructed multicast tree is shown in Fig. 3.3(b).

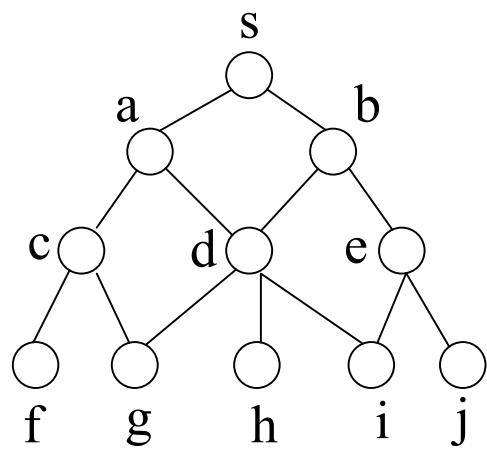


(a) Network Topology

(b) Multicast Tree



(c) Channel Assignment



(d) Tree Mesh

Figure 3.3. An Example for LCA and Tree Mesh

We can see that in the tree, level $0 = \{s\}$, level $1 = \{a, b\}$, level $2 = \{c, d, e\}$ and level $3 = \{f, g\}$. Thus, we get the channel assignment in Fig. 3.3(c), where the number above the node represents the channel for receiving, and the number below the node for sending.

The LCA algorithm has two advantages: simple implementation and throughput improvement. It only needs one BFS of the network at the beginning, and it creates the multicast tree by connecting the multi-receivers with the nearest tree nodes. The tree nodes then can decide the channels by themselves according to the level information, which can be realized distributedly. At the same time, the use of multiple channels reduces the close-by interference and allows more simultaneous transmissions.

However, there is still potential for the LCA algorithm to improve system throughput. Firstly, LCA cannot diminish the interference among the same levels since it uses the same channel at the same level. For example, in Fig. 3.3(c), since g is in the transmission range of both c and d , there will be interference when c and d use the same channel. Secondly, when the number of available channels is more than that of the levels, some channels will not be utilized, which is a waste of channel diversity. Thirdly, the channel assignment does not take the overlap property of the two adjacent channels into account. As we know, $\forall i$, channel i and channel $i + 1$ are adjacent in frequency, so they partially interfere with each other. Thus, the channel i for level i still has some inference effect with the channel $i + 1$ for level $i + 1$.

3.3 Multi-Channel Multicast Algorithm

To further improve the system throughput, we propose a Multi-Channel Multicast algorithm (MCM) to minimize the number of the relay nodes and the hop count distances between the source and the destinations, and further reduce the interference by exploiting all the partially overlapping channels instead of just the orthogonal channels.

3.3.1 Multicast Structure Construction

Following the design constraint of WMNs, we aim for a multicast protocol for WMNs, which includes two primary procedures. The first is to build an effective multicast structure, which is detailed in this subsection, and the second tries to allocate channels for minimizing interference in the next subsection.

Broadcast Structure. Some previous work treats broadcast and multicast in a different way. Actually, when all the nodes are multi-receivers, the multicast problem becomes the broadcast problem. We can say that broadcast is a special case of multicast. In order to focus on the basic idea of MCM, we first consider the situation that all the nodes are the multi-receivers. We then detail how to trim off those unnecessary branches based on the broadcast structure when the multi-receivers are only a portion of the nodes. The broadcast structure in the mesh network is built by the following steps.

The **first step** is realized by BFS, which is similar with the LCA algorithm. After the BFS traversal, all the nodes are divided into different levels. We then delete the

edges between any two nodes of the same level, with which we get the elementary communication structure — “tree mesh”. Fig. 3.3(a) and Fig. 3.3(d) give an example of the original network topology and its responding tree mesh. We use BFS to build the tree mesh with the the following reasons:

1. With the hop count distance increasing between the sender and the receiver, the intra-flow contention exacerbates. Moreover, shorter hop count distance means shorter transmission delay. Minimizing the delay is also important in WMNs, thus we build a shallow tree by BFS, which reduces the total hop count distances from the source to the receivers.
2. BFS guarantees that if two nodes are not at the same level or the adjacent levels, they are at least two hops away. Hence, when considering channel assignment, the two nodes may use the same channel since they are very likely not to interfere with each other.
3. The time complexity of BFS is $O(|V| + |E|)$, whose cost is much less than other broadcast or multicast tree construction algorithms.

In the **second step**, we identify the minimal number of relay nodes that form the broadcast tree. In the tree mesh, one node could have more than one parent. The purpose of this step is to identify the only parent (we call it a relay node here) for a node who has more than one parent so that the number of relay nodes is minimal.

A top-down approach, i.e., from level 0, level 1 to the lowest level, is used to identify the relay nodes. Suppose we have discovered the relay nodes in level 0, level

1... level $i - 1$, now we study how to find out the relay nodes in level i . We can see that the fewer relay nodes will result in less traffic flows in the network, which means less local interference. Thus, our objective is to identify the minimal number of relay nodes in level i that can communicate with all the nodes in level $i + 1$.

Definition 3 Given a tree mesh T_G , $T_G(i)$ is a subgraph of T_G and consists of only the nodes at level i and level $i + 1$ of T_G . This subgraph $T_G(i)$ is called $(i, i + 1)$ **subtree mesh**. In addition, the set S_i consisting of the nodes from level i is called the **upper node set** of $T_G(i)$, and the set S_j consisting of the nodes from level $i + 1$ is called the **lower node set**.

Algorithm 2: Relay Node Search in Level i Algorithm

Data: $T_G(i)$: $(i, i + 1)$ subtree mesh; S_i : nodes in level i ; S_j : nodes in level $i + 1$

Result: R : the set of the relay nodes in level i

```

1  $R = \emptyset$ ;
2 while  $S_j \neq \emptyset$  do
3   In  $T_G(i)$ , compute the number of parents of each node in  $S_j$ , and compute
   the number of children of each node in  $S_i$ ;
4   Find  $v_{i1}, v_{i2}, \dots$  in  $S_j$  with the minimal number of parents;
5   Among the parents of  $v_{i1}, v_{i2}, \dots$ , find  $t_f$  with the maximal number of
   children;
6    $R = R \cup \{t_f\}$ ;
7    $S_i = S_i - \{t_f\}$ ;
8   The children of  $t_f$  record  $t_f$  as their relay node;
9    $S_j = S_j - \{\text{the children of } t_f\}$ ;
```

We can see that identifying the minimal number of relay nodes at level i is equivalent to selecting the minimal number of nodes at upper node set of $T_G(i)$ that can cover all the nodes of lower node set. In fact, it is a variation of the set-cover problem,

which has been proved NP-complete. We devise an approximation algorithm, which is detailed in Algorithm 2.

1. Some parents are considered as relay candidates if one of their children has the minimal number of parents.
2. Among the relay candidates, we choose one node that has the maximal number of children. The reason is that given the fixed number of nodes at the level $i + 1$, the more children a relay node can forward packets to, the less number of relay nodes we will need at level i .
3. We remove the relay node and its children, and repeat the above process until all the nodes at level $i + 1$ are removed.

We use a simple example to further explain this algorithm. Fig. 3.4(a) gives a $(i, i + 1)$ subtree mesh, from which we can compute the number of parents of each node in level $i + 1$. The node 1, 5, and 7 have the minimal number of parents (1 parent), and their parents are nodes a , c , and d . The numbers of children of nodes a , c , and d are 3, 2, and 2 respectively. Since node a has the maximal number of children, a is chosen as a relay node.

We then remove a and its child nodes 1, 2, and 3 from the subtree mesh. In the new subtree mesh, which is shown in Fig. 3.4(b), the node 5 and 7 have the minimal number of parents, and their parents are nodes c and d . We randomly choose c as one relay node since c and d both have 2 children.

Afterwards, we remove c and its children, then get the subtree mesh shown in Fig. 3.4(c). Similarly, with the process above, we select node d as a relay node. After

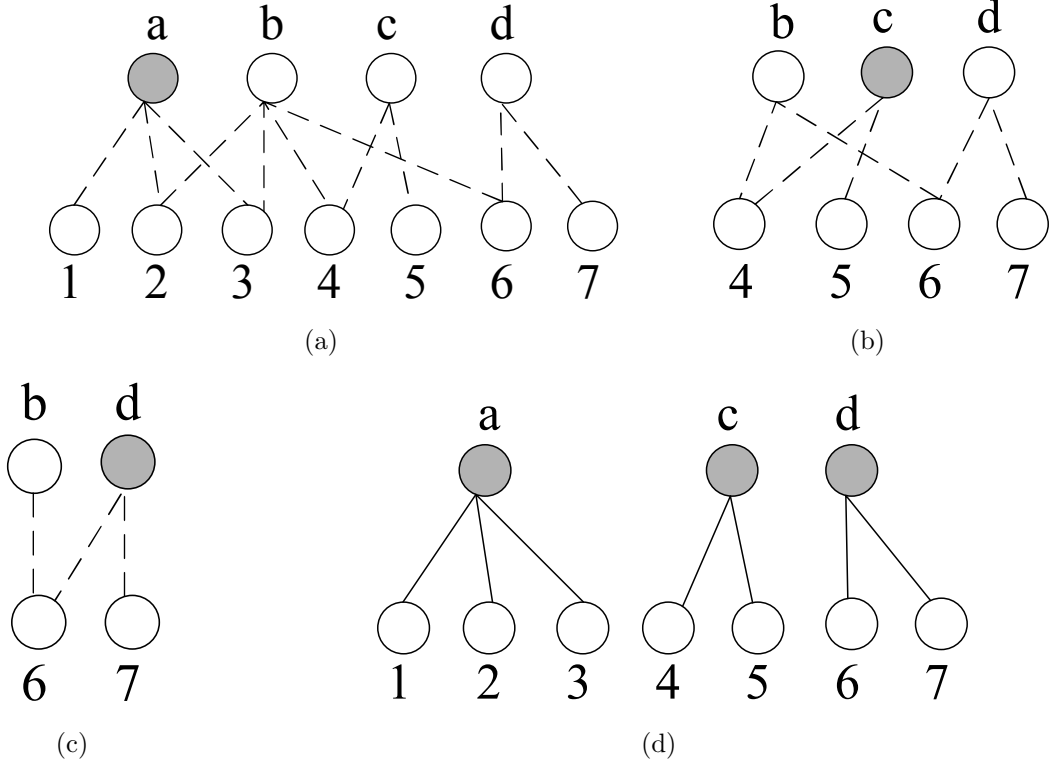


Figure 3.4. Relay Node Search Example

removal of node d and its children, the level $i + 1$ is empty, thus the algorithm stops. Finally, nodes a , c , and d are chosen as relay nodes at level i , which is shown in Fig. 3.4(d).

Algorithm 2 is superior to the Greedy Set Cover algorithm [81] by introducing step 1. We observe that if a node has just one parent, the parent has to be selected as a relay node, while that greedy algorithm recursively selects the node with the maximal number of children in the remained graph. For the above example, that greedy algorithm will select a , b , c and d as relay nodes.

Multicast Structure. The broadcast structure mentioned above contains some unnecessary branches if the destinations do not involve all the nodes. Instead, we propose to construct a “slim” structure by using the MCM Tree Construction algorithm

described in Algorithm 3.

Algorithm 3: MCM Tree Construction Algorithm

Data: T : tree mesh of the network
Result: T' : multicast tree

- 1 Use BFS to partition nodes into different levels;
- 2 **for** $\forall$ node $v \in V(T)$ **do**
- 3 $\lfloor$ $c[v]=\text{true}$ if and only if v is a multi-receiver or the source.
- 4 **for** $l = \text{LevelNum} - 1; l \geq 1; l = l - 1$ **do**
- 5 $S_i = \{\text{node } v_i | v_i \text{ belongs to level } l - 1\};$
- 6 $S_j = \{\text{node } v_j | v_j \text{ belongs to level } l \text{ and } c[v_j] = \text{true}\};$
- 7 **while** $S_j \neq \emptyset$ **do**
- 8 Find $v_{i1}, v_{i2}, \dots$ in S_j with the minimal number of parents;
- 9 Among the parents of $v_{i1}, v_{i2}, \dots$, find node t_f with the maximal number of children;
- 10 $c[t_f] = \text{true};$
- 11 $S_i = S_i - \{t_f\};$
- 12 The children of t_f record t_f as their relay node;
- 13 $S_j = S_j - \{\text{the children of } t_f\};$
- 14 $V(T') = \emptyset; E(T') = \emptyset;$
- 15 **for** $\forall$ node $v \in V(T)$ **do**
- 16 $V(T') = V(T') \cup \{v\}$ if and only if $c[v] = \text{true};$
- 17 edge $e = (v, v\text{'s relay parent});$
- 18 $E(T') = E(T') \cup \{e\};$

The goal of the algorithm is to discover the minimal number of relay nodes to construct a multicast tree. The search process starts from the bottom to the top. We use a boolean variable – “ $c[v]$ ” for any node v to represent that v is either a multi-receiver or a relay node if $c[v]$ is true. At each step, we intend to minimize the number of relay nodes at the upper level, which can cover all the multi-receivers and relay nodes at the lower level. The process is similar with the broadcast structure, except that we do not require that the relay nodes should cover those non-receiver and non-relay nodes of the lower level.

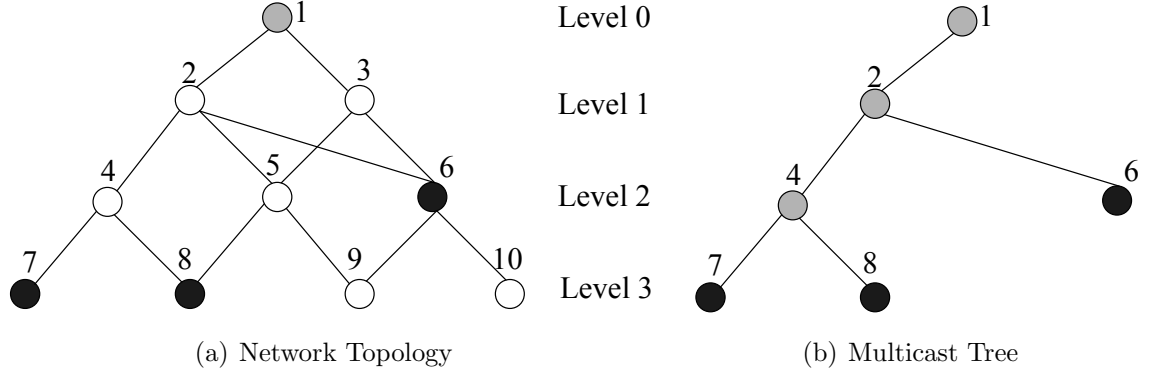


Figure 3.5. A Multicast Structure Example

We use a simple example to illustrate the process. There is a tree mesh in Fig. 3.5(a), where nodes 6, 7, and 8 are multi-receivers. First, we select node 4 at level 2 because it covers all the multi-receivers at level 3. Next, we select node 2 at level 1, which covers all the multi-receivers and the relay node at level 2. Finally, we get the multicast tree in Fig. 3.5(b).

3.3.2 Channel Assignment

The tree node discovery in the previous subsection allows each multi-receiver to connect with the gateway through minimal hop count distance. In this section, we discuss how to assign channels to the interfaces of the tree nodes.

As assumed in Section 3.1, each node has two interfaces. Specially, the interface that a node uses to receive packets from its relay node at the upper layer, termed Receive-Interface (RI), is disjoint from the interface the node uses to forward packets to its children, called Send-Interface (SI). In order to guarantee that the relay node can communicate with its children, each node's RI is associated with the SI of its relay node, i.e. they should be assigned the same channel. Ascending Channel Allocation

is proposed to assign channels and described in Algorithm 4.

Algorithm 4: Ascending Channel Allocation Algorithm

Data: $\{0, 1, \dots, C - 1\}$: available orthogonal channel set; T' : multicast tree;

Result: Channel assignment for interfaces

```

1 The source uses channel 0 for its SI;
2 Its children use channel 0 for their RIs;
3  $A = 0$ ;
4 for  $l = 1; l \leq LevelNum-2; l++$  do
5     for  $\forall$  relay node  $u$  at level  $l$  do
6          $A = (A + 1) \bmod C$ ;
7          $u$  uses channel  $A$  for its SI;
8          $u$ 's children use channel  $A$  for their RIs;
```

The basic idea of the algorithm is straightforward: from top to down in the tree, the channels are assigned to the interfaces in the ascending order until the maximum channel number is reached, then start from channel 0 again. Although simple, this approach avoids the situation that the same channel is assigned to two nearby links that interfere with each other. We use a simple case to illustrate this algorithm in Fig. 3.6, where the the number of the orthogonal channels are 3. Note that the number above the node represents the channel number used for its RI, while the number below the node represents the channel number for its SI.

In the algorithm we only use limited orthogonal channels. 802.11b provides 14 channels, 5 MHz apart in frequency. However, to be totally orthogonal, the frequency should be at least 30 MHz, so 802.11b can offer only 3 non-overlapping channels. Thus, although the Ascending Channel Allocation is easy to implement, its performance is still constrained by the limited number of orthogonal channels. Fortunately, as mentioned in Section 3.1, network throughput can be further improved by exploiting

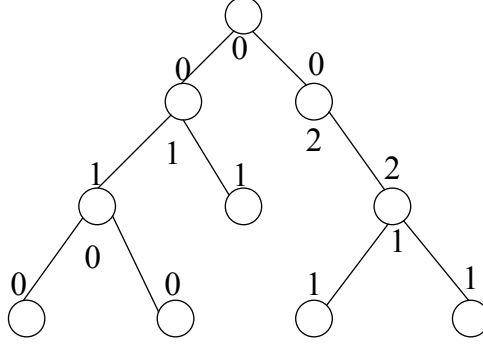


Figure 3.6. Ascending Channel Allocation Example

all the partially overlapping channels.

Heuristic Channel Assignment. In fact, we can utilize all the channels instead of just orthogonal channels. In Section 3.1, we observed that the interference range decreases with the increase of the channel separation. Intuitively, the channel assignment should make a large channel separation for two wireless links if the physical distance between them is short. We aim to minimize the sum of the interference area of all the transmissions.

We use $IR(u_v)$ to indicate the interference range of sender u of one link with respect to sender v of another link. According to the experiment we performed in Section 3.1, under the condition that all the nodes have the same transmission range R , $IR(u_v) = R * \delta_{|i_u - i_v|}$. Here, u and v use channel i_u and i_v for their SIs respectively, and δ_t is the Interference Factor. When allocating a channel for relay node u , the channel assignment should take a channel that minimizes the sum of the square of the IRs between u and u 's neighboring relay nodes, that is, minimize $\sum_{v \in N(u)} IR^2(u_v)$, where $N(u)$ represents the set of the neighboring relay nodes of u . This is because the bigger interference area means bigger chance two transmissions may interfere. In

addition, the interference area is approximated as a circle whose area is determined by $IR^2(u_v)$. Since $\sum_{v \in N(u)} IR^2(u_v) = \sum_{v \in N(u)} (R * \delta_{|i_u - i_v|})^2$, we just need to minimize $\sum_{v \in N(u)} \delta_{|i_u - i_v|}^2$. Based on this consideration, we propose the Heuristic Channel Assignment in Algorithm 5.

Algorithm 5: Heuristic Channel Assignment Algorithm

Data: CH : available channel set; T' : multicast tree;

Result: Channel assignment for interfaces

```

1 The source uses channel 0 for its SI;
2 Its children use channel 0 for their RIs;
3 for  $l = 1; l \leq LevelNum-2; l++$  do
4   for  $\forall$  relay node  $u$  at level  $l$  do
5      $S(u) = \{u\text{'s neighboring relay nodes that have been assigned channels}$ 
       $\text{for their SIs} \}$ 
6     Choose channel  $i \in CH$  that minimizes  $\sum_{v \in S(u)} \delta_{|i_u - i_v|}^2$ 
7      $u$  uses channel  $i$  for its SI;
8      $u$ 's children use channel  $i$  for their RIs;
```

3.4 Further Discussion on MCM Algorithm

In this section, we discuss some companion mechanisms to further improve the MCM algorithm.

3.4.1 Repair Mechanism

Here we discuss the failure recovery mechanism and node join mechanism.

Failure Recovery. Usually, the mesh routers work properly, but node failure can happen for various reasons. When a tree node fails, nodes in its subtree lose their

connectivity to the root. Our mechanism will reorganize the multicast tree to bypass the failed node and restore the connectivity.

At first, we can safely assume that any node is able to detect the failure of its neighbor quickly since the nodes periodically send “hello” messages to their neighbors. If a node does not receive a hello message from one neighbor for a period of time, it considers the neighbor to have failed. There are two cases that apply to the node failure: the collapsed node is a leaf or a relay node.

For the first case, if the collapsed node v is a leaf, we propose two approaches. One is just to leave it alone since the leaf is not responsible for forwarding packets to any other tree nodes. (The manner that the mesh clients within the communication range of the failed node restore the connectivity to the network is beyond the scope of this chapter.) This approach is simple, but the parent of the failure node will continue to receive packets even if the parent is not a multi-receiver.

The second choice is related with v 's parent u on the tree. If u has only one child on the tree, then it stops forwarding packets. Moreover, if u is not a multi-receiver, it sends out a message to its neighbors, announcing that it is no longer part of the tree. After that, the parent of u will do the same thing. The process continues until one ancestor of v has more than one child or it is a multi-receiver. The first choice is simple to implement, and v is able to join the multicast group again if v recovers from failure after a short period of time. The second choice helps to remove those unnecessary branches, which reduces the interference and saves bandwidth.

For the second case, when v is a relay node, all its children on the tree should check whether they are physical neighbors of some other relay nodes on the tree.

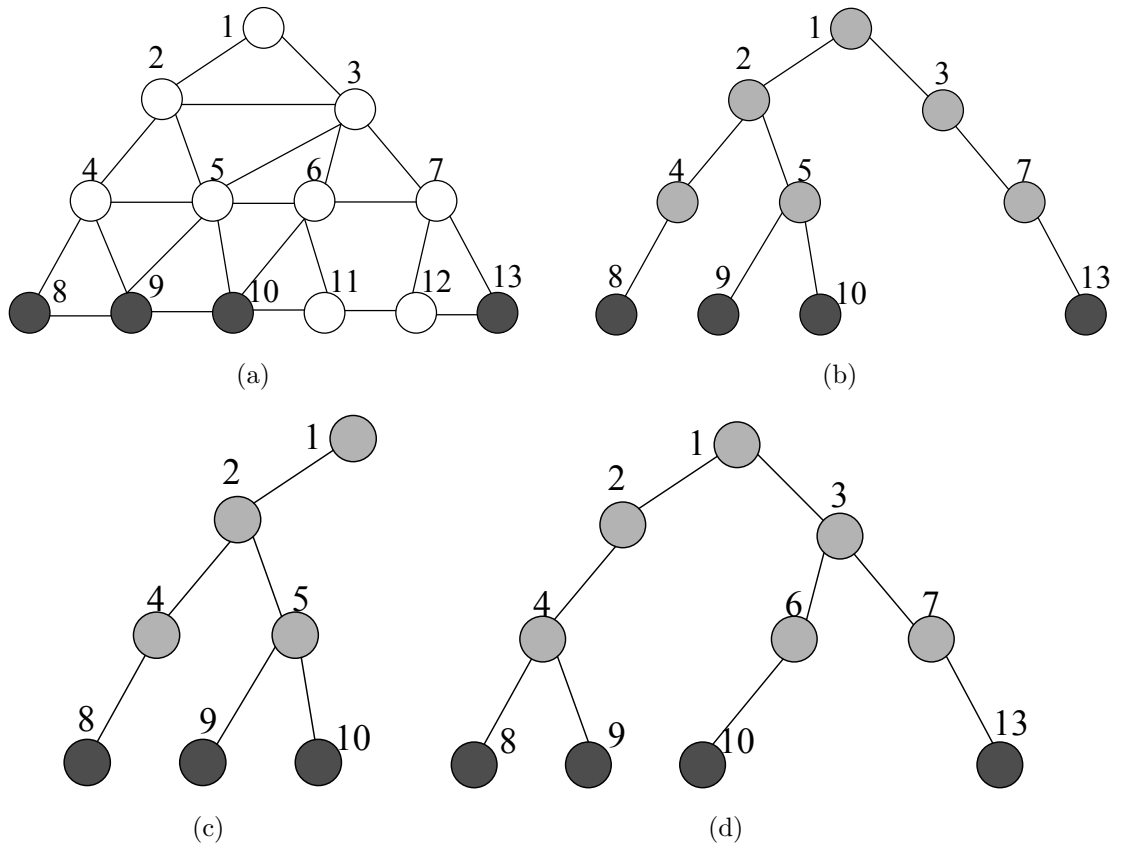


Figure 3.7. Repair Mechanism Example

(If two nodes are within each other's transmission range, they are called **physical neighbors** even if they are not using the same channel.) If they are, the channels of their RIs will be reassigned as that of the “backup” relay node's SI and re-establish the connectivity with the gateway. If they are not physical neighbors of any relay node, each node will randomly choose one neighbor t at the upper layer, requesting t to be its relay node. If t is the physical neighbor of one relay node, it connects with the relay node by using the same channel, otherwise, t will randomly choose one of its neighbors at the upper layer, asking that node to be a relay node. This process continues until the request arrives at a physical neighbor of any relay node.

We use an example in Fig. 3.7 to illustrate the repair mechanism. Fig. 3.7(a) and Fig. 3.7(b) give the network topology and the responding multicast tree respectively, where nodes 8, 9, 10, and 13 are multi-receivers. If node 13 breaks down, because it is a leaf, we can simply leave it alone. The other choice is that node 13 requires its parent 7 to stop forwarding packets. Node 7 realizes that it has one child on the tree, so it also asks its parent 3 to stop forwarding. The resulting multicast tree after failure of node 13 is shown in Fig. 3.7(c).

If node 5 breaks down, its children begin to look for other connections to the source. Node 9 finds that it can communicate with relay node 4, so it changes the channel on its RI for packet reception from node 4. Node 10 cannot communicate with any relay node on the tree, so it randomly selects one of its physical neighbors at the upper level, such as node 6, requesting node 6 to be its relay node. Node 6 then tries to communicate with any neighboring relay node, and it sets the channel of its RI the same as the SI of relay node 3. Node 6 then chooses a channel that is

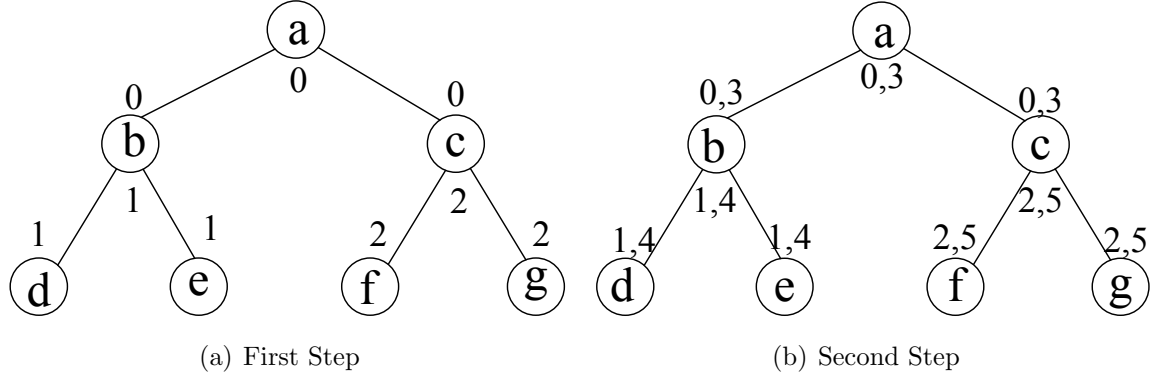


Figure 3.8. Example of 4-Interface Channel Allocation

not used by any of its neighbors for its SI, and node 10 accordingly sets the same channel for its RI for packet reception. The resulting multicast tree after the failure of node 5 is shown in Fig. 3.7(d).

Node Join. If a new node wants to join the multicast group, it sends out a request for connecting to a nearby relay node. The request is locally flooded, and the nodes on the path that reaches the nearest relay node will be absorbed into the multicast tree.

3.4.2 Channel Assignment with More Interfaces

The previous part of this chapter assumes that each node has only 2 interfaces. We now discuss the case when there are more interfaces for each node. In the multicast application in WMNs, we can utilize multiple interfaces (more than 3) to achieve parallel transmissions to further improve the throughput.

If each node has $2k$ interfaces, we divide the interfaces into two groups: the sending group $\{1, 2, 3 \dots k\}$ and the receiving group $\{k + 1, k + 2 \dots 2k\}$. The sending group is responsible for forwarding packets, while the receiving group is responsible

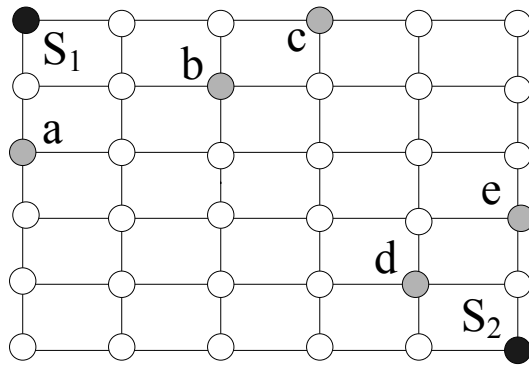
for receiving packets.

We still use the same multicast tree construction method described in Section 4, but we modify the channel assignment algorithm. Suppose there are C available channels. First, we assume the node has only 2 interfaces: RI and SI, and we use the channel assignment algorithms in Section 4 by restricting the number of available channels to $\frac{C}{k}$. After the process, RI is assigned channel i , and SI is assigned channel j respectively. We then apply this result to the $2k$ interfaces: interface p ($1 \leq p \leq k$) is assigned channel $i + \frac{C(p-1)}{k}$, and interface q ($k+1 \leq q \leq 2k$) is assigned channel $j + \frac{C(q-k-1)}{k}$.

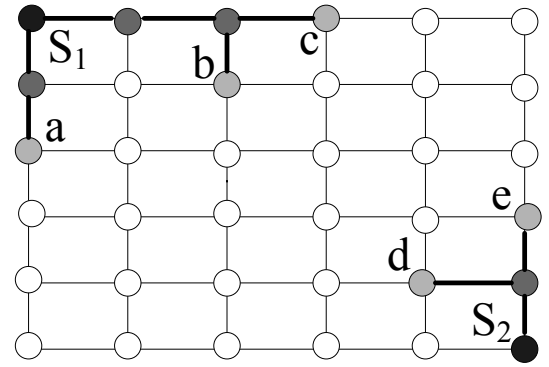
An example in Fig. 3.8 illustrates the mechanism. If $C = 6$ and $k = 2$, we allocate $C' = \frac{C}{k} = 3$ channels to the interfaces as if each node has only 2 interfaces, and the result is shown in Fig. 3.8(a). Next, we get the channel assignment for the 4-interface case, which is shown in Fig. 3.8(b), where the numbers above the node represent the channels for the receiving group and the numbers below the node represent the channels for the sending group. Now, we can see that node a can simultaneously transmit the packets to node b and c on 2 wireless links, which are on channel 0 and 3 respectively.

3.4.3 Multiple Gateways

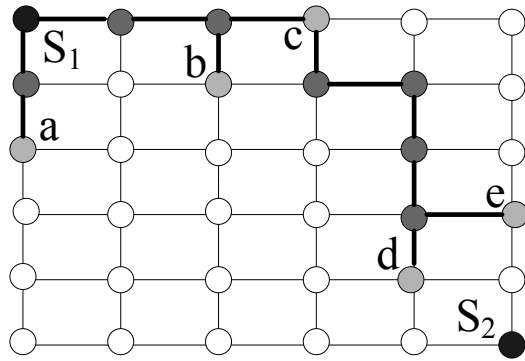
Multiple gateways may be available as the Internet access points for a wireless mesh network in order to provide more bandwidth and improve the system throughput. In the case of our multicast study with multiple gateways, the multi-receivers can



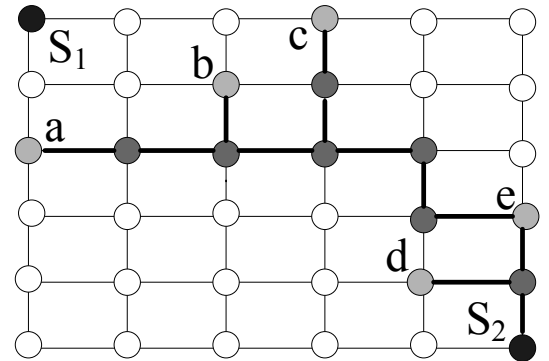
(a) Network Topology



(b) Multicast Forest



(c) Use Gateway S1



(d) Use Gateway S2

Figure 3.9. Example of Multiple Gateways

obtain the same information packets through different gateways. Therefore, one way to optimize the multicast performance is to build multiple multicast trees initialized from different gateways to shorten the hop count distance between each receiver to its root. We design the following algorithm to build multiple multicast trees rooted from different gateways.

First, each gateway starts a breadth first search so that each node in the network will get its own hop count distances to each gateway.

Second, each multi-receiver chooses the nearest gateway as its information root and notifies the gateway so that each gateway will know which multi-receivers belong to itself.

Third, for each gateway constructed by using Algorithm 3, a multicast tree rooted from this gateway is built for the multi-receivers who choose this gateway.

We use a simple example to demonstrate the whole process. Fig. 3.9(a) shows a network topology, where nodes s_1 and s_2 are two gateways, and nodes $a-e$ are the multi-receivers. At the beginning, node s_1 and node s_2 initialize a breadth first search respectively. After the breadth first search, each multi-receiver knows the hop count distances to both s_1 and s_2 . Each multi-receiver will select the gateway that is closer to it, so nodes a , b and c choose s_1 as their gateway, while nodes d and e choose s_2 as their gateway. Two multicast trees rooted from s_1 and s_2 will be built, which is shown in Fig. 3.9(b).

Instead, if we just use one gateway, the multicast tree becomes larger. For example, if only gateway s_1 is used, the multicast tree is built as shown in Fig. 3.9(c), where the total hop count distance is 12, and the number of relay nodes is 7. If

only gateway s_2 is used, the multicast tree is built as shown in Fig. 3.9(d), where the total hop count distance is also 12, and the number of relay nodes is 7. On the other hand, if we use both s_1 and s_2 , the total hop count distance of the multicast forest decreases to 9, and the number of relay nodes decreases to 4, so the multicast structure becomes “slim” and the system throughput will be improved as shown in next section.

3.5 Simulations

We evaluate the MCM algorithm by comparing it with the LCA algorithm and a single channel multicast algorithm through the following metrics.

- **Throughput:** the throughput is the average number of packets each multi-receiver receives during a time unit.
- **Delay:** the delay is the average time it takes for a packet to reach the destination after it leaves the source.

We use an NS2 simulator (version 2.29)[82] to simulate a flat area of 900 m by 900 m with varying number of randomly positioned wireless router nodes. By extending the NS2 simulator, we configure all nodes to use multiple interfaces/channels with a transmission range of 250 m and a carrier sensing range of 550 m. We use the default IEEE 802.11 MAC configuration in NS2, which supports multicasting using broadcasting at the base rate 1 Mbps.

We evaluate LCA and MCM algorithms in different scenarios. For each scenario, we randomly generate 100 different graphs where the source and the destinations are randomly selected. Traffic is generated by constant bit rate (CBR) sessions. We vary the session rate at some scenarios. The packet size for all traffic is set to be 512 bytes. Except for the last subsection, we use the orthogonal channels in the simulation.

3.5.1 Impact of Network Size

We evaluate the throughput in different network sizes by assigning the number of nodes with 30 and 60, and assigning the number of the available channels with 12. For each network size, we vary the number of multi-receivers from 5 to 55. We measure the throughput of the MCM algorithm, the LCA algorithm, and the single-channel algorithm in which only one single channel is used in the multicasting.

The results are shown in Fig. 3.10. We can see that using multi-channel and multi-interface significantly improves the throughput. The reason is that using different channels prevents the channel interference among close-by transmissions. Compared with LCA, MCM further improves throughput though they both take advantage of multiple channels and multiple interfaces. This is because that MCM builds a more efficient multicast tree and carefully assigns the channels on the tree; thus, it further reduces the interference.

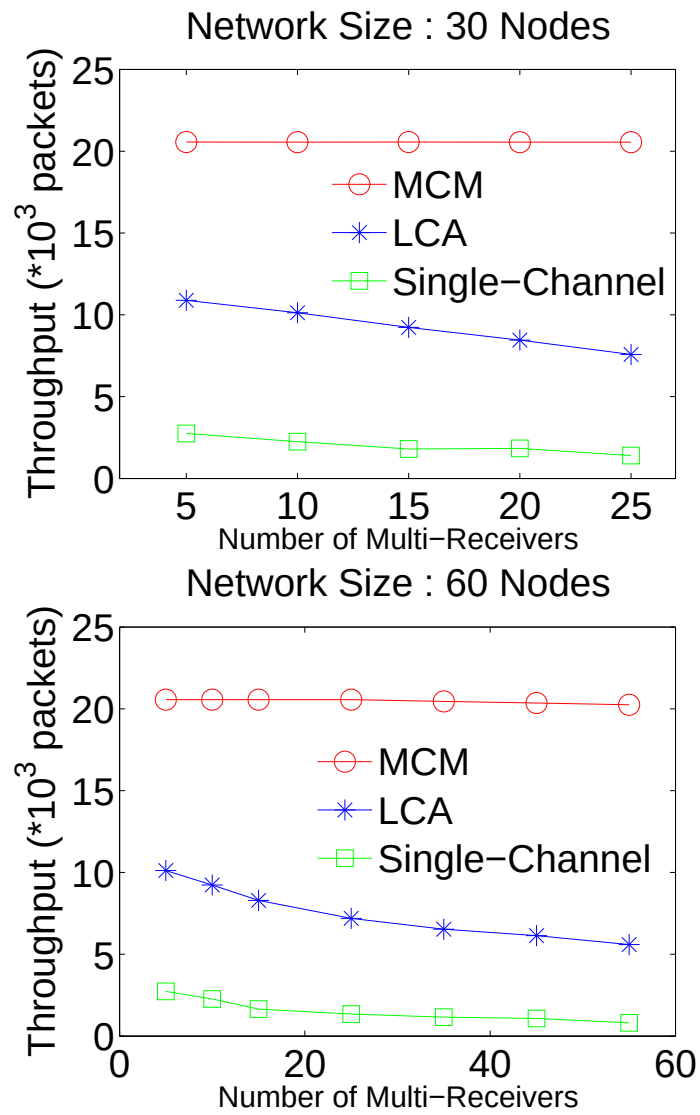


Figure 3.10. Impact of Network Size

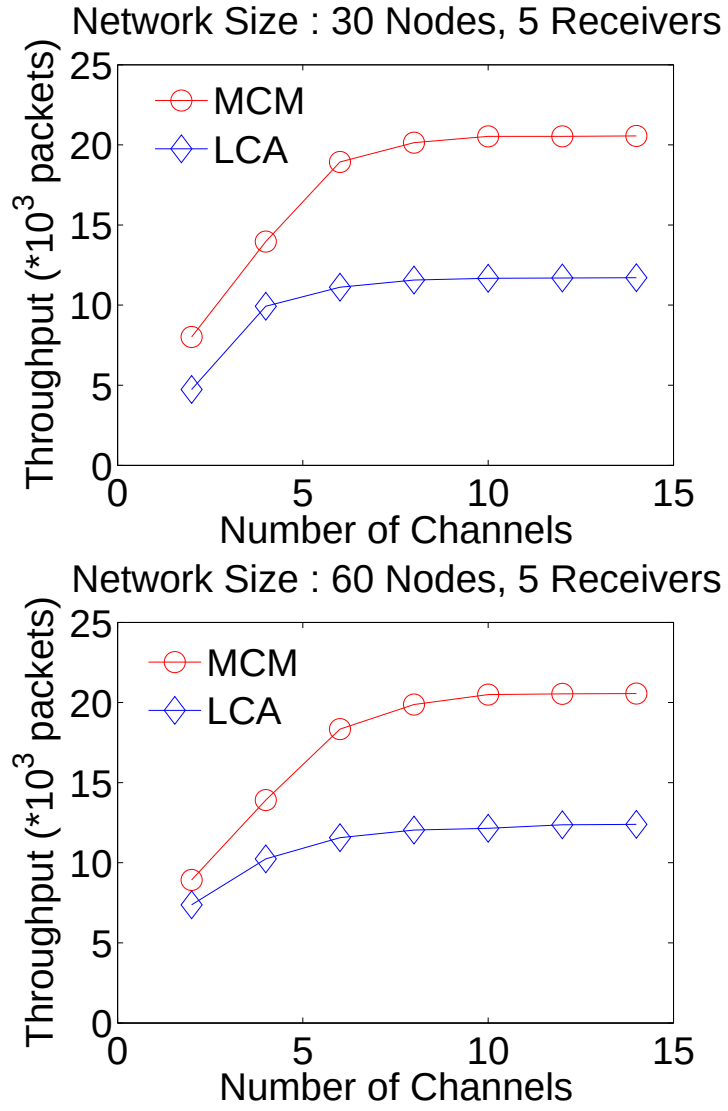


Figure 3.11. Impact of Number of Channels

3.5.2 Impact of Number of Channels

We vary the number of available channels from 2 to 14 and measure the throughput of MCM and LCA. Fig. 3.11 shows that MCM enhances throughput more than LCA with the increasing of the number of channels.

In addition, we notice that when the number of channels varies from 2 to 6, both MCM and LCA have great throughput improvement. However, when the number of

channels varies from 7 to 14, MCM has small throughput improvement while LCA almost has no improvement. The explanation of the phenomena on MCM is that when the number of channels increases to a certain extent, it is enough to eliminate almost all the interference in the network, thus using more channels cannot further improve throughput. At the same time, the number of channels that LCA uses is equal to the tree height, thus some channels are left unused when the number exceeds tree height although some interference still exist in the network.

3.5.3 Impact of Transmission Rate

We vary the transmission rate from 50 packet/s to 300 packet/s, and measure the throughput of MCM and LCA. Fig. 3.12 shows that MCM achieves much better throughput than LCA under different transmission rates. We also observe that the saturated transmission rates for MCM and LCA to achieve nearly the maximal throughput are 225 packet/s and 125 packet/s respectively. The transmission rate exceeding the saturated rate almost does not help to improve the throughput. Since MCM has the higher saturated transmission rate, this means that MCM can take greater advantage of the channel diversity than LCA.

3.5.4 Delay Comparison

In this simulation, we evaluate the delay of LCM and MCM by comparing the average time each packet takes to reach the destination. The transmission rate is set to 200 packet/s. Fig. 3.13 shows that MCM has a much shorter delay than LCA. We also

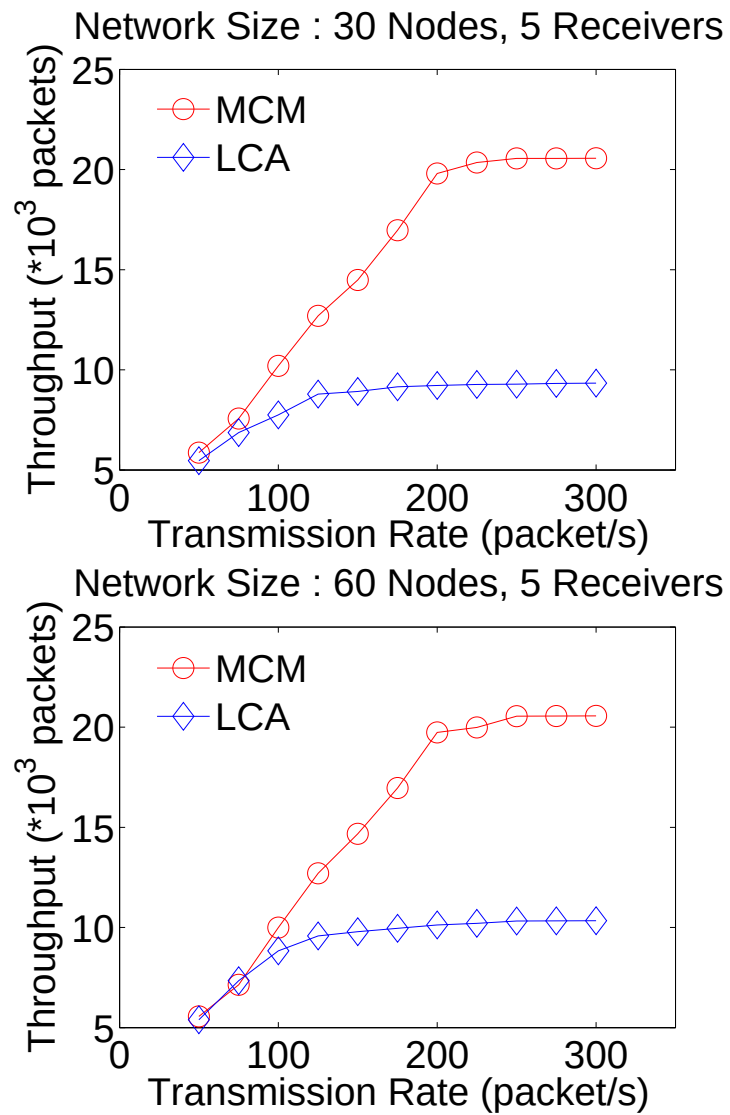


Figure 3.12. Impact of Transmission Rate

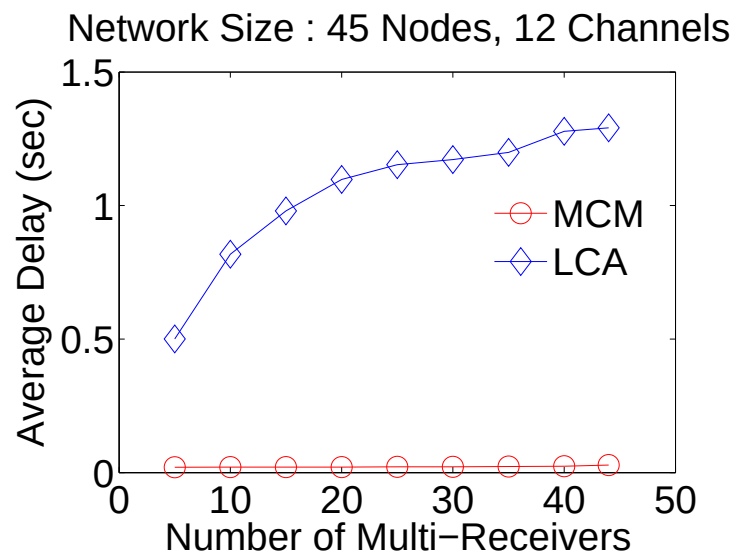
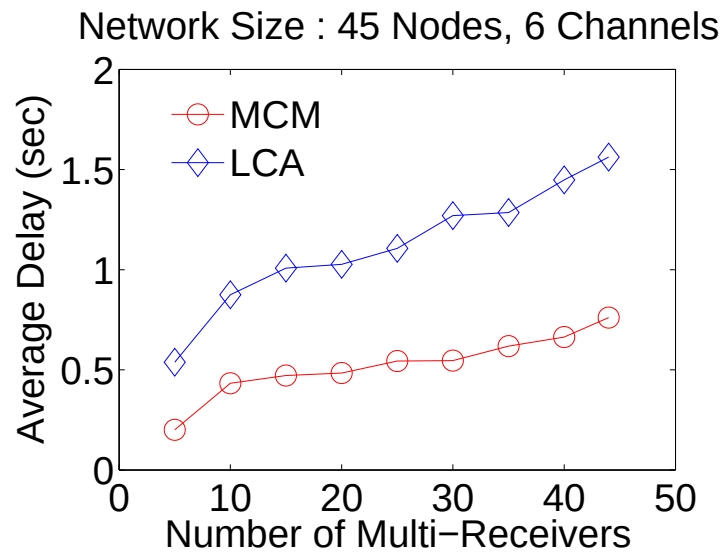


Figure 3.13. Delay Comparison

see that the delay of MCM decreases rapidly when the number of channels increases from 6 to 12, since interference is greatly reduced by using more channels.

3.5.5 2 Interfaces vs 4 Interfaces

In Section 3.4, we discussed how to allocate channels to more interfaces. Now we evaluate the throughput of MCM on a different number of interfaces by varying the channel numbers from 4 to 18, and the results are reported in Fig. 3.14.

We can see that using 4-interface usually can further improve the throughput, since utilizing more interfaces allows more simultaneous data transmissions. On the other hand, making use of more interfaces also leads to more local flow contentions. If there are not enough available channels, the extra contentions caused by extra interfaces degrade the network performance. Thus, when the number of available channels is small (i.e., under 6), using 2-interfaces is a better choice. Even if the number of channels exceeds 6, the network throughput of 4-interfaces does not achieve nearly 2 times of that of 2-interfaces as expected until the number reaches a threshold. Given the fixed network size, the threshold is related with the number of multi-receivers that decides the size of the multicast tree and the contention level, that is, more receivers bring more local contentions. The results show that the threshold for 5 receivers is 14, while it is 18 for 35 receivers.

As we know, the number of current available channels is limited to under 14, and the number of orthogonal channels is much less. Exploiting more interfaces does not achieve much better results subject to the current channel condition. In addition

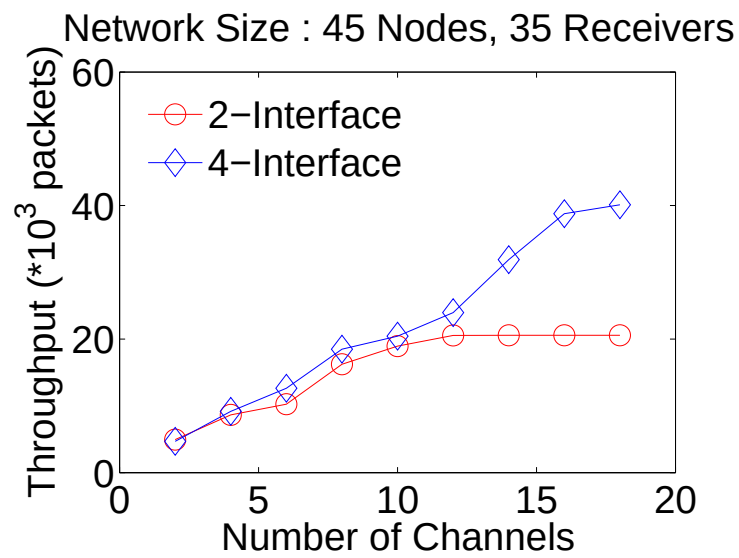
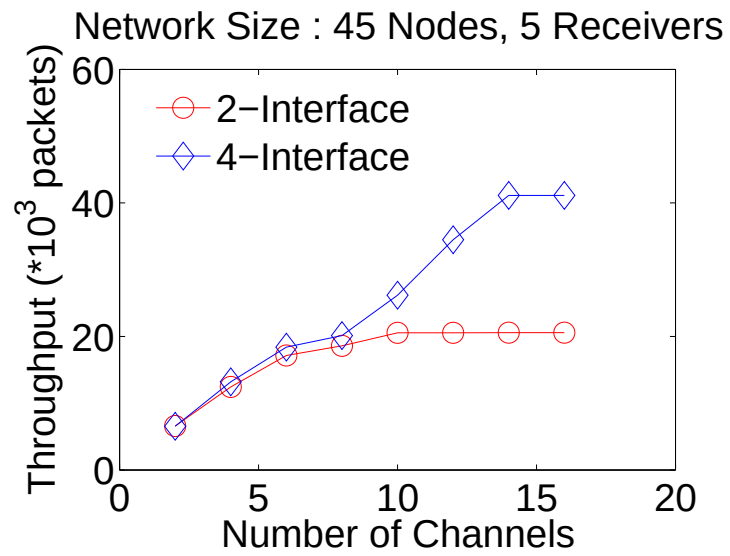


Figure 3.14. 2-interface vs 4-interface

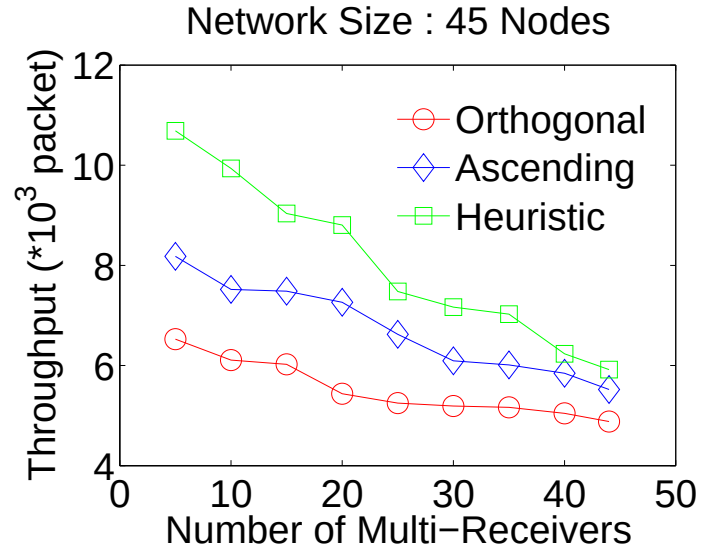
to the increased cost of equipping with more interfaces, using 2-interfaces currently seems to be a suitable choice. In the future, with exploiting more channels and more low-cost interfaces, making use of more interfaces may have an important application.

3.5.6 Partially Overlapping Channel Assignment Test

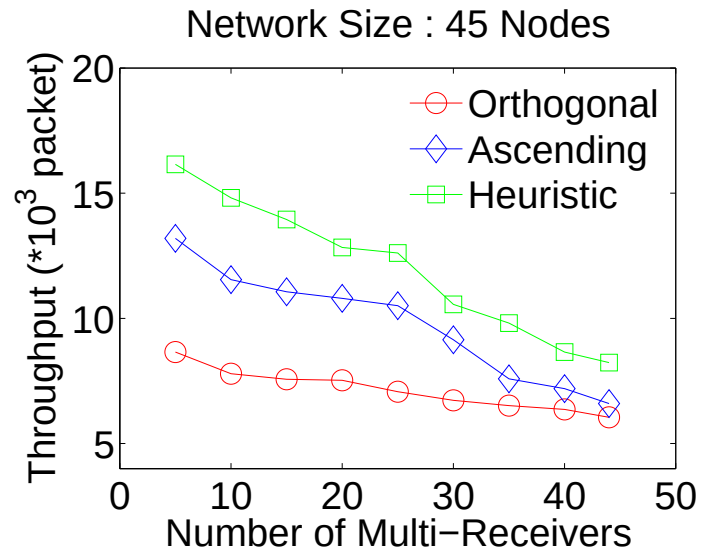
Usually, users are offered a range of frequency spectrum, where the numbers of orthogonal channels and partially overlapping channels are fixed. We choose two frequency spectrums: (i) 30MHz, which can offer 2 orthogonal channels and 6 partially overlapping channels, and (ii) 60MHz, which can offer 3 orthogonal channels and 12 partially overlapping channels. In the two spectrums, we compare the throughput of the MCM algorithm under the following different channel assignment methods.

1. We only use the orthogonal channels defined in the fixed range of frequency spectrum.
2. We use Ascending Channel Allocation to allocate all the partially overlapping channels in the frequency spectrum to the interfaces.
3. We use Heuristic Channel Assignment to allocate all the partially overlapping channels in the frequency spectrum to the interfaces.

The results in Fig. 3.15 show that using partially overlapping channels can achieve better throughput than using just orthogonal channels. This is because the orthogonal channels are so scarce that they can not eliminate all the interference, while the partially overlapping channels can further reduce interference. We also observe that



(a) Spectrum Range: 30 MHz



(b) Spectrum Range: 60 MHz

Figure 3.15. Partially Overlapping Channel Test

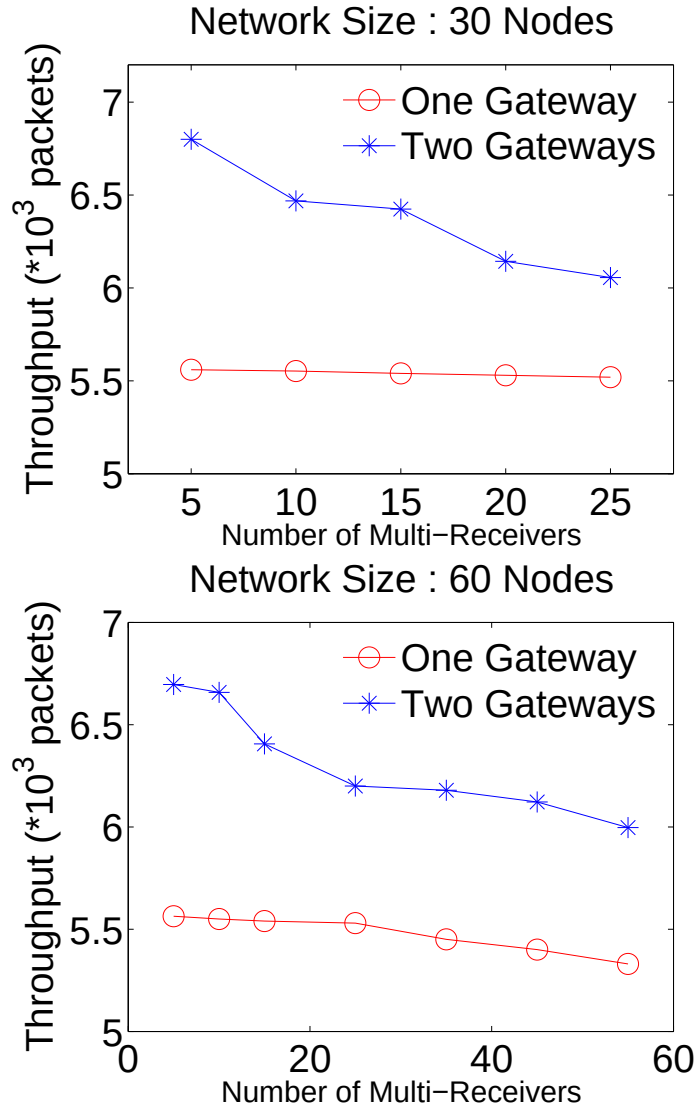


Figure 3.16. Multiple Gateway Test

Heuristic Channel Assignment is better than Ascending Channel Allocation, since it makes a large channel separation for the adjacent wireless links.

3.5.7 Multiple Gateways vs Single Gateway

There are usually multiple gateways in a wireless mesh network to improve the throughput of the data flows between the network to the Internet. In this subsection, we assume that the wireless mesh network has two gateways. We compare the

throughput of using both gateways with that of using only one gateway under different network sizes of 30 nodes and 60 nodes. The number of the available channels is 12, and the number of multi-receivers varies from 5 to 25 in a 30-node-sized network and from 5 to 55 in a 60-node-sized network. The packet transmission interval is set as 0.018 sec/packet. The results are shown in Fig. 3.16. We can see that using multiple gateways is able to significantly improve the network throughput because it can decrease the total hop count distance of the multicast trees.

3.6 Summary

In this chapter, we investigate the multicast algorithms in wireless mesh networks where the throughput and the delay have the paramount priorities. In order to achieve efficient multicast in WMNs, two multicast algorithms, LCA and MCM, are proposed by using multi-channels and multi-interfaces. An effective multicast structure is constructed to minimize the number of the relay nodes and the communication delay. The dedicated channel assignment helps to further minimize the interference as well. Compared with previous multicast approaches, our algorithms are based on the multi-channel and focus on the throughput maximization. The performance evaluation shows that our algorithms outperform the single-channel multicast in terms of throughput and delay, and more efficient multicast structure and subtle channel assignment can further improve throughput and reduce delay.

CHAPTER 4

Multicast Algorithms for Link-Heterogeneous Wireless Mesh Networks

As illustrated in Chapter 1, major issues for implementing multicast in WMNs are throughput and delay, as we consider potential multicast applications, such as Video on Demand. One common approach to increase throughput and minimize delay is to minimize the number of transmission (including retransmissions) in the network. This is because reducing transmissions not only saves bandwidth, but also reduces interference. In this chapter, we aim to improve the multicast throughput by minimizing the number of transmissions required in multicast.

Two WMN multicast algorithms (LCA and MCM) were proposed in Chapter 3 that increased throughput and minimized delay by taking advantage of multiple chan-

nels and multiple interfaces. These algorithms assumed that all WMN links are homogeneous. Given this assumption, minimizing the number of transmissions in a WMN is equivalent to minimizing the number of relay nodes in the multicast structure. However, individual WMN links may have different qualities for a variety of reasons, such as environmental factors and diverse device ability. The link quality also determines the number of transmissions. Therefore, although minimizing the number of relay nodes helps to decrease simultaneous interference in the WMN, it is also important to choose high quality links to minimize the total number of transmissions.

In this chapter, we point out that the local broadcast quality mainly relies on the worst involved link, which is mostly different from other related work. Based on this observation, this chapter proposes a new algorithm to minimize transmissions in link-heterogeneous WMNs with the ulterior goal of increasing throughput and minimizing delay. The key step is to model WMN link heterogeneity by assigning weights to each link that represent that link's quality. The node weight is set as the weight of the worst involved link. We consider one source of link heterogeneity: individual links have different link loss ratios, and thus different ETX.

In order to model link heterogeneity in multicasts, we define the concept of a Heterogeneous Weighted Steiner Connected Dominating Set (HW-SCDS), which consists of both nodes and arcs. The nodes indicate the multicast group members, including the relay nodes. The arcs indicate the packet flow direction. Minimizing the number of transmissions in a WMN is equivalent to computing the minimum HW-SCDS (MHW-SCDS) in the edge-weighted graph. We prove that computing a MHW-SCDS is NP-hard, and we devise a greedy algorithm that appears to work well for our

model of link heterogeneity. Simulations show that our algorithm significantly increases WMN throughput and significantly reduces WMN delay.

The rest of this chapter is organized as follows. In Section 4.1, we show the existence of link heterogeneity and its effect on WMN throughput. In Section 4.2, we define the HW-SCDS problem, provide its NP-hard proof, and present the greedy algorithm. We discuss the application of MHW-SCDS and provide companion mechanisms in Section 4.3. Performance evaluation is given in Section 4.4, and we give summary in the last section.

4.1 Link Heterogeneity in Wireless Mesh Networks

We first show experimentally that WMN links vary in quality, even with the same physical equipments. We then show the effects of link heterogeneity on WMN throughput.

4.1.1 Existence of Heterogeneous Links in Mesh Networks

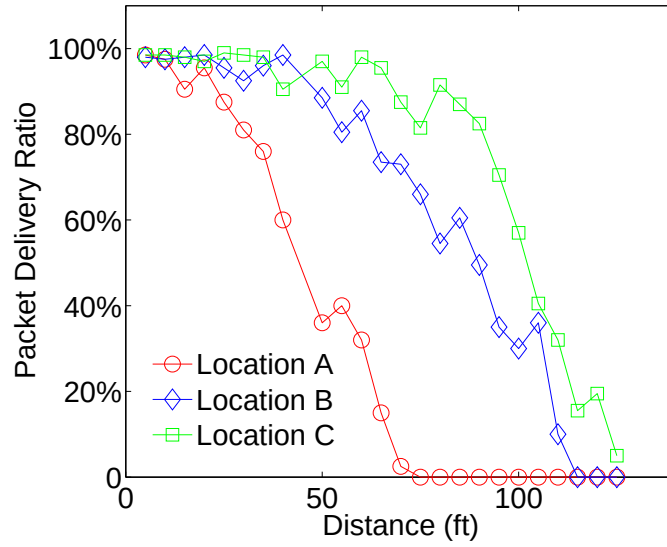
In practical applications, different mesh routers may have different communication and computation abilities that affect the quality of their transmission links. Even if all mesh routers are physically identical, the transmission links among them may still vary in quality due to environmental factors [83].

We conducted an experiment to determine how sensitive link delivery ratio is to

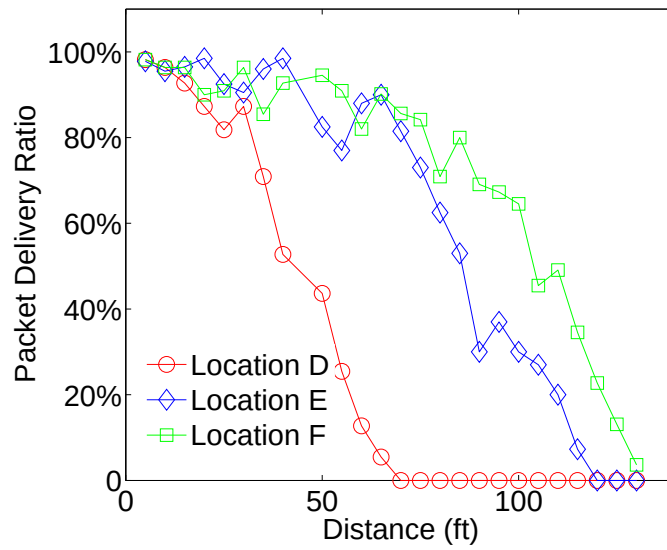
the location of mesh routers. We used two laptops with Netgear WAG511 PC cards to construct a wireless link. We tested the link bandwidth between these two laptops at 6 different pairs of locations. In all our tests, we kept the transmission power and channel constant. We used two transmission rates: 2 MBPS and 5.5 MBPS. In two of the 6 pairs of locations (A, D), the laptops are obstructed by a wall. The results shown in Fig. 4.1 demonstrate packet delivery ratio is very sensitive to the location of mesh routers. Thus, even with identical router nodes, transmission links can have very different qualities.

4.1.2 Effects of Heterogeneous Links on Throughput

In order to save bandwidth and decrease the interference for multicast communication in wireless networks, one local broadcast is preferable to multiple local unicasts from one sender. As shown in Fig. 4.2, sender s wants to forward packets to 3 receivers (a , b and c) in its neighborhood. Using local broadcast, all receivers get the information at approximately the same time. We use the lowest quality link in a local broadcast to approximate the quality of the local broadcast. For example, in Fig. 4.2, the successful delivery ratio of each link is 90%, 80% and 85%, respectively. Given that the expected transmission count $ETX = 1/(1 - p)$ where p is the link loss ratio [84], this implies that the expected transmission count for nodes a , b and c are 1.11, 1.25 and 1.18, respectively. We use the largest expected transmission count, in this case 1.25 for link (s, b) , to approximately model the expected transmission count for all receivers to successfully receive the packet.



(a) Transmission Rate: 2 MBPS



(b) Transmission Rate: 5.5 MBPS

Figure 4.1. Sensitivity of Link Bandwidth to Node Location

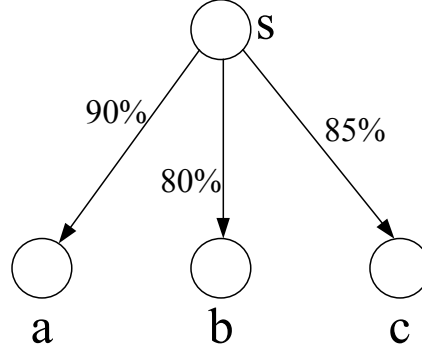
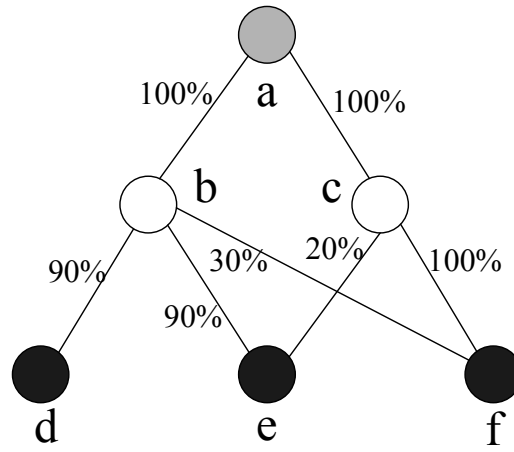


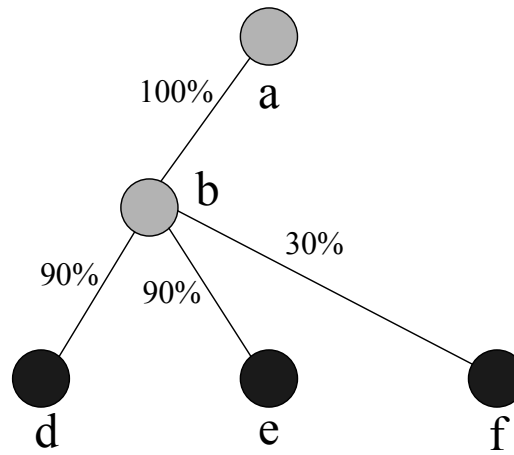
Figure 4.2. Local Multicast Example

All links are considered to be homogeneous in [85], so the algorithm aims to minimize the number of relay nodes in the multicast structure. Minimizing relay nodes does minimize the number of transmissions when links are homogeneous. This leads to reduced bandwidth consumption and less interference among adjacent flows. However, solely minimizing the number of relay nodes does not readily lead to minimizing transmissions if the links are heterogeneous. Link quality also influences the number of transmissions. That is, we want to ensure that packets transmit over high-quality links that have low link loss ratios.

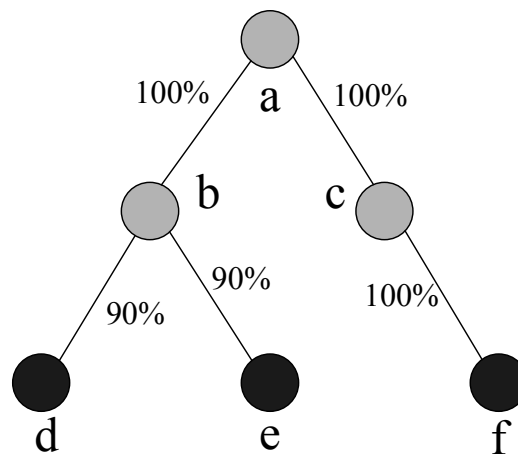
In the network topology in Fig. 4.3(a), node a is the source and nodes d , e and f are multi-receivers. The multicast tree built in [85] is shown in Fig. 4.3(b). Although the number of relay nodes is minimized, the total expected transmission count is approximately 4.33. The solution shown in Fig. 4.3(c) has a better total expected transmission count of roughly 3.11. Although minimizing the relay nodes as done in [85] decreases simultaneous interference, using slightly more relay nodes should not increase the local interference too much, particularly since mesh router nodes have multiple channels.



(a) Network Topology



(b) Multicast Tree 1



(c) Multicast Tree 2

Figure 4.3. An Example for Two Different Multicast Trees

4.2 Minimum HW - SCDS

Based on the above considerations, we model the multicast problem in link-heterogeneous mesh networks as a new graph theory problem – Minimum Heterogeneous Weighted SCDS (MHW-SCDS), which helps to minimize the number of transmissions in WMNs.

4.2.1 Basic Definitions

We consider two different categories of multicast communication: (i) group multicast to share information among a group of nodes where each node may take a turn as the source node, and (ii) general multicast where a specific source node transmits information to a specific set of destination nodes. The multicast structure (or multicast tree) needed to perform these two different types of multicast can vary. In our work, we focus on multicast structures to perform general multicast. However, we begin by considering the multicast structure for group multicast.

In group multicast, each node in the group must be able to send a multicast to all other nodes in the group. To support this symmetrical communication, we generally model the underlying network using an unweighted undirected graph $G = (V, E)$. This model implicitly assumes that all links are bidirectional and all links have the same quality. A single multicast structure that can facilitate group communication for a given set of nodes $D \subseteq V$ is a Steiner Connected Dominating Set (SCDS) of D ; that is, a set $S \subseteq V$ such that (i) each node in D is in S or has a neighbor in S and (ii) S is connected. An SCDS of D with minimum cardinality is called Minimum Steiner

Connected Dominating Set (MSCDS) of D . Computing an MSCDS is NP-hard [68], and its NP-complete decision problem, which we call SCDS-D, is defined below.

INSTANCE: A graph $G = (V, E)$, a subset $D \subseteq V$, and a positive integer $K \leq |V|$.

QUESTION: Is there an SCDS of D of size at most K ?

In general multicast, the multicast begins with a specified source node s . If we continue to model the network as an unweighted undirected graph $G = (V, E)$, an SCDS is still an appropriate multicast structure for general multicast. The only modification is that we require s to be included in the SCDS. This leads us to the following definition of a 1-SCDS for D and s : a set $S \subseteq V$ such that (i) each node in D is in S or has a neighbor in S and (ii) $s \in S$, and (iii) S is connected.

We will show that identifying a minimum cardinality 1-SCDS (1-MSCDS) is an NP-hard problem. Its corresponding decision problem 1-SCDS-D is defined as follows:

INSTANCE: A graph $G = (V, E)$, a node $s \in V$, a subset $D \subseteq V$, and a positive integer $K \leq |V|$

QUESTION: Is there a 1-SCDS for D and s of size at most K ?

Until now, we have assumed the underlying graph $G = (V, E)$ is unweighted and undirected. However, in general multicast, the packets are forwarded from the source node to the destination nodes. Furthermore, links in a heterogeneous mesh network have different qualities, and it is not obvious that for any two nodes u and v that the qualities of links (u, v) and (v, u) are identical. Thus, we propose modeling the underlying graph $G = (V, E)$ as a directed graph with a weight function $w : E \rightarrow \mathfrak{R}$. The weight on edge (a, b) is the ETX on link (a, b) .

We define a new multicast structure, Heterogeneous Weighted Steiner Connected

Dominating Set (HW-SCDS), that facilitates general multicast in a directed graph $G = (V, E)$ with a weight function w for a specified source node s and a set of destination nodes D . The key difference between HW-SCDS and SCDS or 1-SCDS is that HW-SCDS includes both the nodes and the arcs used in the multicast communication, whereas SCDS and 1-SCDS include only the nodes. The need for this is highlighted in Fig. 4.3. Consider the network topology in Fig. 4.3(a), and suppose we state that both nodes b and c will be used in the multicast structure to transmit a message from source node a to destination nodes d , e , and f . Because the links are heterogeneous, it makes a difference whether or not node f receives the message using link (b, f) or link (c, f) .

For a directed graph $G = (V, E)$ with a weight function w , an HW-SCDS for a source node s and a set of destination nodes $D \subseteq V$ is a set of nodes S and a set of edges A such that (i) for every arc $(u, v) \in A$, $u \in S$; (ii) for every node $v \in D$, there exists a directed path P from source node s to node v in A .

When working with a 1-SCDS S in an unweighted undirected graph $G = (V, E)$, we could use the number of nodes in S as the cost for this multicast structure as this number of nodes represents the total time required to perform all the transmissions. On the other hand, when working with an HW-SCDS (S, A) for a weighted directed graph $G = (V, E)$, we must take into consideration which arcs are present in A when computing the time required to perform all the transmissions. We formally define a cost metric to help calculate this total transmission time as follows.

Definition 4 In HW-SCDS (S, A) , if arc $(a, b) \in A$, node a is called the **multicast**

parent of b , and node b is called the **multicast child** of a .

For instance, in Fig. 4.2, s is the multicast parent of a , b , and c . And a , b , and c are the multicast children of s .

Definition 5 In HW-SCDS (S, A) , suppose node a has k multicast children $\{b_1, b_2 \dots b_k\}$. Then we set the **node weight** of a to be $\max(w(a, b_1), w(a, b_2), \dots, w(a, b_k))$.

Given an HW-SCDS (S, A) , the sum of all the node weights in S is called the **cost weight** of (S, A) . For instance, the cost weight of the HW-SCDS in Fig. 4.3(b) is 4.3, while the cost weight of the HW-SCDS in Fig. 4.3(c) is 3.1. In order to improve multicast performance, we need to compute an HW-SCDS with the minimum possible cost weight. We refer to this as a Minimum Heterogeneous Weighted SCDS or MHW-SCDS.

We prove that computing an MHW-SCDS is an NP-hard optimization problem in the next subsection by proving its corresponding decision problem, HW-SCDS-D defined below, is NP-hard.

INSTANCE: A weighted directed graph $G = (V, E)$, weight function w on E , a source node $s \in V$, a subset $D \subseteq V$, and a positive number K .

QUESTION: Is there an HW-SCDS (S, A) for s and D of cost weight at most K ?

4.2.2 NP-Hardness Proof

We now prove that computing an MHW-SCDS is NP-hard.

Lemma 1 *1-SCDS-D is NP-hard*

We now give a polynomial time Turing reduction from the SCDS-D problem to the 1-SCDS-D problem. That is, we give an algorithm $A'(G, D, K)$ that solves the SCDS-D decision problem in polynomial time by using a polynomial time algorithm $A(G, s, D, K)$ for solving the 1-SCDS-D problem as a subroutine.

The algorithm A' works as follows. Given an input instance $G = (V, E)$, s , and D , A' makes $|V|$ separate calls to subroutine A , one for each node $v \in V$. That is, for all $v \in V$, A' invokes $A(G, v, D, K)$. If any of these procedure calls returns “yes”, the A' returns “yes”, otherwise A' returns “no”.

If we can find a polynomial-time algorithm A that solves the 1-SCDS-D problem, then we have a polynomial time algorithm that solves the SCDS-D problem. However, SCDS-D is NP-hard, so 1-SCDS-D is also NP-hard.

Lemma 2 *HW-SCDS-D is NP-hard*

The proof is by restriction. Essentially, 1-SCDS-D is a restricted case of HW-SCDS-D. We show this by showing how every instance of 1-SCDS can be made into an equivalent instance of HW-SCDS-D. Basically, take every edge (u, v) in the unweighted undirected graph $G = (V, E)$ and replace it with two arcs (u, v) and (v, u) , each with weight 1. Since 1-SCDS-D is NP-hard, HW-SCDS-D is also NP-hard.

Corollary 1 *Computing an MHW-SCDS is NP-hard*

The decision problem HW-SCDS-D is NP-hard, so the optimization problem of computing an MHW-SCDS is also NP-hard.

4.2.3 Greedy Algorithm

We devise a greedy algorithm for choosing a good HW-SCDS that is similar to the algorithm used in [85]; the key difference is our use of edge weights to model link heterogeneity and the fact that we must identify which edges are chosen as well. The basic idea is to first partition the nodes into levels based on each node's hop count distance from the source and then select nodes in level i to cover the already selected nodes in level $i + 1$.

In the **first step**, we partition the nodes into levels by performing a breadth first search (BFS) from the source node. We remove any edges that connect two nodes on the same level to create a **tree mesh** T_G . Let S_i be the nodes in level i . Let E_i be the set of edges connecting nodes in level i to nodes in level $i + 1$. We now process the nodes from the bottom to the top. That is, we start with the nodes that are farthest from the source and proceed level by level choosing nodes and arcs to add to our HW-SCDS.

In the base case when dealing with the nodes in S_q where q is the maximum level in T_G , we simply choose the nodes in S_q that are in D . Now let's assume we have chosen nodes in S_{i+1} through S_q . Let the chosen nodes in S_{i+1} be denoted as C_{i+1} . We ignore all nodes in $S_{i+1} - C_{i+1}$. We now need to choose $C_i \subseteq S_i$ and $F_i \subseteq E_i$ to add to the HW-SCDS such that the cost weight of (C_i, F_i) is minimized and that every node $v \in C_{i+1}$ is covered by a node $u \in C_i$ and an arc $(u, v) \in F_i$.

In [85], the authors considered a problem that they named the Minimal Upstream Node (MUN) problem that is a special case of this problem of identifying an optimal

(C_i, F_i) where all the edge weights are 1. For consistency of notation, we name our weighted version of the MUN problem to be the **Minimal Weight Upstream Node (MWUN)** problem. Formally, MWUN is defined as follows:

INSTANCE: An edge-weighted bipartite graph $G = (U, V, E)$.

TASK: Find a set $U' \subseteq U$ along with a set of edges $E' \subseteq E$ such that each node in V is connected by an edge in E' to a node in U' and the cost weight of (U', E') is minimized.

The authors in [85] proved that MUN and its associated decision problem is NP-hard, so it follows that MWUN is NP-hard.

After partitioning the nodes into different levels in the first step, we devise a greedy algorithm for MWUN, which is detailed in Step 2.1 - Step 2.3. In this, we assume that we are identifying $C_i \subseteq S_i$ and $F_i \subseteq E_i$ to add to our HW-SCDS given already chosen nodes $C_{i+1} \subseteq S_{i+1}$. We also assume that E_i only contains arcs that end at nodes C_{i+1} in S_{i+1} .

Step 2.1: We transform an MWUN input instance $G = (S_i, C_{i+1}, E_i)$ and edge weight function w to a node-weighted bipartite set cover input instance $G' = (U, V, E)$ with node weight function w' as follows.

1. For each node $v \in C_{i+1}$, create node $v \in V$. For every node $u \in S_i$, create $d(u)$ nodes $\{u_1, u_2, \dots, u_{d(u)}\}$ in U where $d(u)$ is the degree of node u in G .
2. For each node u in S_i , label its neighboring nodes in C_{i+1} in the order $\{v_1, v_2, \dots, v_{d(u)}\}$ such that $w(u, v_1) \leq w(u, v_2) \leq \dots \leq w(u, v_{d(u)})$. For $1 \leq k \leq d(u)$, connect $u_k \in U$ to the k nodes in V that correspond to nodes

$v_1, v_2, \dots, v_k$ in C_{i+1} and set $w'(u_k) = w(u, v_k)$.

In Fig. 4.4(a), we give an MWUN input instance, and in Fig. 4.4(b), we give the resulting node-weighted bipartite set cover input instance.

Step 2.2: We use a greedy algorithm to discover a set cover $C' \subset U$ in $G' = (U, V, E)$.

1. For every $u_k \in U$, define its current value $v(u_k) = w'(u_k)/d(u_k)$, where $d(u_k)$ represents the degree of u_k in the current graph G' .
2. Add u_k with the smallest value in the current graph G' to C' . If there is a tie, choose the node with the most children. If there is still a tie, choose a node arbitrarily.
3. We modify G' as follows. $U = U - \{u_i\}$; $V = V - \{\text{the neighbors of } u_i\}$.
4. Repeat the above process until $V = \emptyset$.

Step 2.3: We use the set cover C' to construct C_i and F_i as follows.

1. For each node $u_k \in C'$, add u to C_i and arcs (u, v_l) to F_i for $1 \leq l \leq k$.

We use some examples to explain the greedy algorithm. Given the graph in Fig. 4.4(b), we compute each node's value: $v(a_1) = 2, v(a_2) = 2, v(a_3) = 2, v(b_1) = 1, v(b_2) = 3.5$. Since b_1 has the smallest value, b_1 is chosen.

We then remove b_1 and its neighbor e from G' . In the updated G' shown in Fig. 4.4(c), the value of each node is: $v(a_1) = 2, v(a_2) = 2, v(a_3) = 3, v(b_2) = 7$.

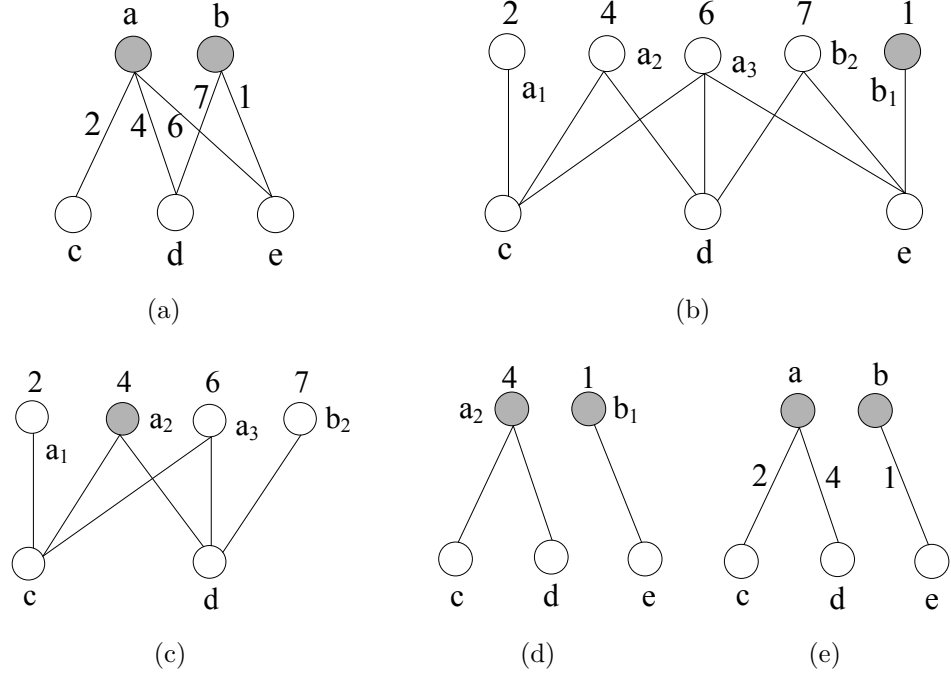


Figure 4.4. Greedy Algorithm Example

Nodes a_1 and a_2 both have the smallest value of 2, but a_2 has more children, so a_2 is chosen.

Afterwards, we remove a_2 and its children from G' . Since V is now empty, the algorithm stops. The chosen set cover C' and corresponding edges of G' are shown in Fig. 4.4(d). Since a_2 and b_1 are chosen for C' , $C_i = \{a, b\}$ and $F_i = \{(a, c), (a, d), (b, e)\}$ as shown in Fig. 4.4(e).

In some cases, it is possible that nodes u_j and u_k are selected for C' where $k > j$. For example, given an MWUN input instance in Fig. 4.5(a), we get the node-weighted bipartite set cover input instance in Fig. 4.5(b). By the above greedy algorithm, a_1 is chosen first, then a_2 is chosen. In such cases, u is added only once to C_i and the k smallest arcs connected to u are added once to F_i .

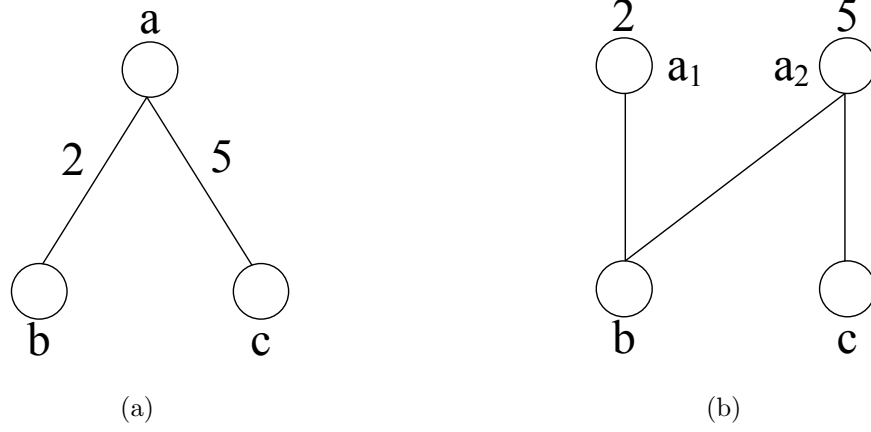


Figure 4.5. Redundant Relay Node

4.3 Application of MHW-SCDS and Further Discussion

In this section, we discuss the application of MHW-SCDS in mesh networks in terms of different link loss ratios. Although we propose one algorithm, we believe that the MHW-SCDS model can be applied to more application scenarios as long as setting the link weights appropriately. The description of the channel assignment strategy on the constructed multicast structure is also presented in this section. At last, we propose the tree repair mechanism to further improve our algorithm.

4.3.1 Different Link Loss Ratios

In this subsection, we discuss how we apply our greedy algorithm for solving MHW-SCDS in WMNs where links have different link loss ratios. In the previous work, it is simply assumed that the links either work well or do not work at all. Although this assumption is typically valid for wired networks, this is not a realistic assumption for

wireless networks where wireless links experience a wide range of intermediate packet loss ratios, and individual links experience different packet loss ratios due to a variety of environmental factors.

How can we cope with relatively large link loss ratios that require many retransmissions on this lossy link? One possible approach is to augment the current mechanism with a link loss rate threshold. However, links with a higher link loss ratio than the threshold may be the only way to reach a multi-receiver, and there might be significant loss ratio difference even among links that are below the threshold.

Instead of using a threshold, we propose the Minimum Loss Ratio Multicast algorithm (MLRM) that works as follows.

1. We create an edge-weighted graph by setting the weight of each link to be the reciprocal of that link's link loss ratio; that is, the expected transmission count on this link.
2. We apply our greedy algorithm for finding an HW-SCDS that dominates the source and the multi-receivers in the graph, and use this HW-SCDS as our multicast structure.

Although we propose one algorithm, we believe that the MHW-SCDS model can be applied to more application scenarios as long as we set the link weights appropriately.

4.3.2 Channel Assignment

The Ascending Channel Allocation algorithm is presented in [85], which assumes that each node has just two interfaces and all the channels are non-overlapping. The

interface that a node uses to receive packets, termed Receive-Interface (RI), is disjoint from the interface the node uses to forward packets, termed Send-Interface (SI). Each node's RI is associated with its relay node's SI, i.e., they should be assigned with the same channel. The basic idea of this algorithm is straightforward: from the top to the bottom in the multicast structure, the channels are assigned to the interfaces in the ascending order.

This algorithm works well since it avoids the situation that the same channel is assigned to two nearby links that may interfere with each other. However, if the number of available channels is small, it still cannot allocate each neighboring link with a different channel. In order to further diminish the interference, we propose the Weighted Channel Allocation algorithm (WCA) that is detailed in Algorithm 6.

At first, we construct a *Weighted Sender Contention Graph (WSCG)*, where nodes correspond to the relay nodes including the source. There is an edge between two nodes if they are physical neighbors. (If two nodes are within each other's transmission range, they are called **physical neighbors** even if they are not on the same channel). Each node has a weight that represents the number of its children in the multicast structure.

When a sender is interfered by other senders, all its children cannot receive the packet at that time. In order to maximize the throughput, a sender with more children should have higher priority not to be interfered. Intuitively, the channel assignment should guarantee that the number of the children of the interfered senders is minimized. Thus, when allocating a channel for relay node u , the channel assignment should take a channel that minimizes the children of u 's all adjacent relay nodes

in *WSCG*, that is, minimize $\sum_{v \in N(u)} C(v)$. Here, $N(u)$ represents the set of the neighboring relay nodes of u , and $C(v)$ represents the number of v 's children.

Algorithm 6: Weighted Channel Allocation Algorithm

Data: CH : the set of available channel; T' : multicast tree;

Result: Channel assignment for interfaces

```

1 Construct the WSCG;
2 The source uses channel 0 for its SI;
3 Its children use channel 0 for their RIs;
4 for  $l = 1; l \leq LevelNum-2; l++$  do
5     for  $\forall$  relay node  $u$  at level  $l$  do
6          $N(u) = \{u\text{'s adjacent relay nodes in WSCG that have been assigned}$ 
           channels for their SIs  $\}$ 
7         Choose channel  $i \in CH$  that minimizes  $\sum_{v \in N(u)} C(v)$ 
8          $u$  uses channel  $i$  for its SI;
9          $u$ 's children use channel  $i$  for their RIs;
```

4.3.3 Repair Mechanism

The mesh networks are able to properly work for a long time, but they may occasionally encounter some node failure during the network lifetime. On the other hand, some new nodes may want to join the multicast tree during the multicast session. Therefore, we need to discuss the node repair mechanism in this subsection.

Node Failure. Node failure may take place for various reasons. When a tree node collapses, the nodes in its subtree have to restore their connectivity to other tree nodes by bypassing this collapsed node and re-organizing the multicast tree. We assume that each node periodically send out “hello” messages to its neighbors. If one node has not received the message from one neighbor for a period, it considers this neighbor collapsed. The collapsed tree nodes may be relay nodes or leaf nodes. For

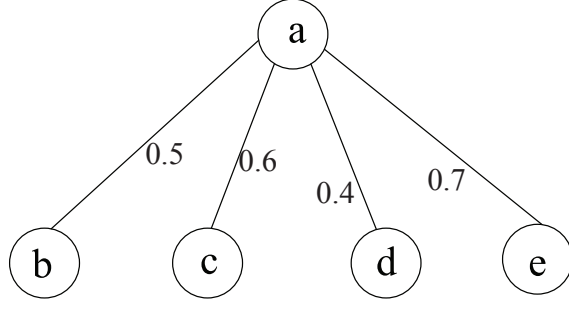


Figure 4.6. ETX Increment Example

each case, we design a strategy to deal with the node failure.

For the first case, when the collapsed node is a relay node, its subtree child (say c) should check whether it is a physical neighbor (defined in Section 4.2) of any tree node. The tree nodes that are physical neighbors of node c_i are marked as *recovery candidates*. Node c need to calculate the ETX increment of each recovery candidate. The node's ETX increment is defined as the difference between its new ETX after adding a new child and old ETX before adding a new child. For short, we use $EI(x)$ to represent the ETX increment of node x . For instance, in Fig. 4.6, node a originally only need to relay packets to nodes b and c , thus its old ETX is $1/0.5 = 2$. If d also becomes its child, the new ETX is $1/0.4 = 2.5$, so $EI(a) = 0.5$. Among all the recovery candidates, v_i chooses the one (say u) whose ETX increment is the minimum as its relay node, and connects with u .

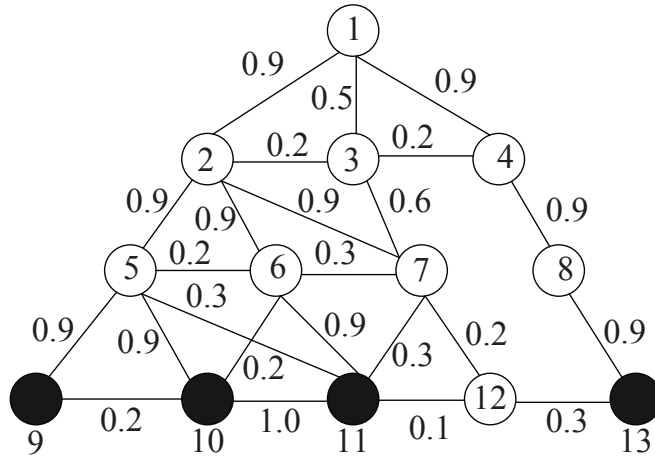
If unfortunately, c is not the physical neighbor of any tree node, it selects one physical neighbor (say v) as relay node such that the link (v, c) has the best link quality among node c 's one-hop neighborhood. Next, node v also need to search for a relay node. This process continues until the connection request arrives at a tree node.

For the second case, the collapsed node c is a leaf node. Its parent t will stop forwarding packets to v after t detects v collapsed. Furthermore, if t is not a multicast destination and it has only one child, it sends the “disconnection” message to its parent for disconnecting itself from the multicast tree. The process continues until one of c ’s ancestor is either a destination or it has multiple children.

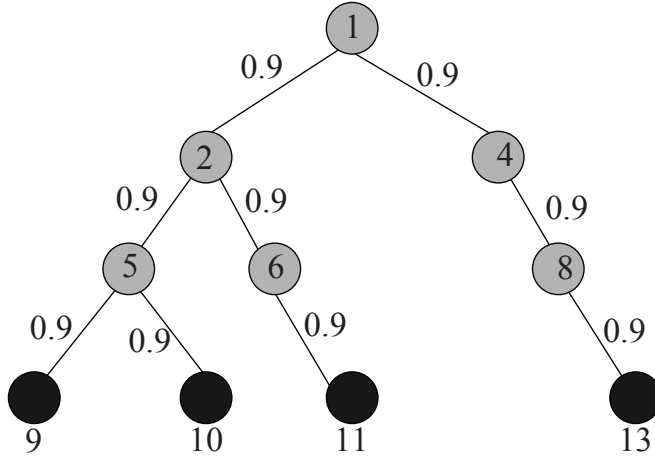
We use an example to illustrate the process. There is a network topology in Fig. 4.7(a), where node 1 is the source, and nodes 9, 10, 11 and 13 are destinations. The number on each edge indicates the packet delivery ratio of this link. The original multicast tree is shown in Fig. 4.7(b). If node 6 breaks down, its child 11 begins to look for its relay node. Node 11 has two recovery candidates: 5 and 10. $EI(10) = 1$, while $EI(5) = 1/0.3 - 1/0.9 = 2.2$. Thus, node 11 selects node 10 as the relay node. The resulting multicast tree after node 10 breaks down is shown in Fig. 4.7(c).

If node 13 breaks down, because it is a leaf node, its parent 8 stops forwarding packets to 13. Because node 8 is not a destination, and it has only one child, it informs its parent 4 to stop forwarding packets. The resulting tree is shown in Fig. 4.7(d).

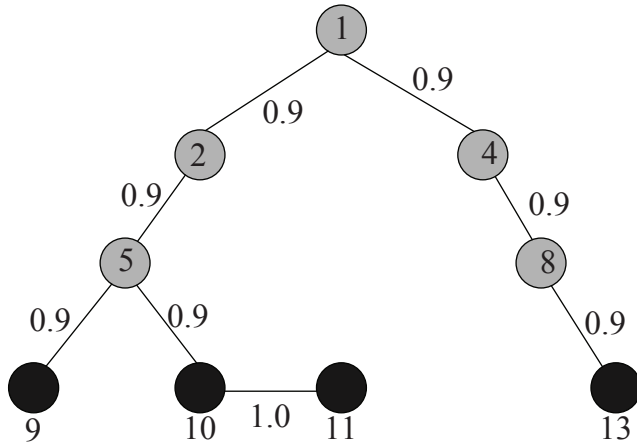
Node Join. If a new node z wants to join the multicast group, it needs to send “connection” request to nearby nodes. This request is locally flooded, until it reaches some tree nodes. The response from a tree node is traveled back along the sending path. The ETX of each link on the path is accumulated until the response arrives at node z . Among all the returned responses, node z chooses the path whose ETX sum is the minimum. The nodes along the path are then absorbed into the multicast tree.



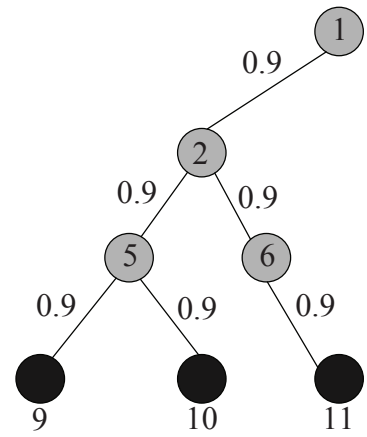
(a)



(b)



(c)



(d)

Figure 4.7. Node Failure Repair Example

4.3.4 Link Fluctuation

In wireless network, the link quality is not always the same all the time. Due to diverse environmental factors, the packet delivery ratio may fluctuate in different time period. That means the ETX on each relay node may fluctuate. The link fluctuation problem caused by fluctuating link quality may degrade the network throughput. In this subsection, we will discuss two basic mechanism to deal with dynamic link fluctuation: (i) dynamically adjust tree based on fluctuating link quality, and (ii) rebuild the tree after a period.

Dynamically Adjust. Each tree node (at level i) is able to detect the link fluctuation between itself and the tree node at the upper layer (level $i - 1$) in its one-hop neighborhood. For simplicity, the tree node at level i is called v_i and the tree node at level $i - 1$ is called v_{i-1} . During a period T_1 (say 5 minutes), node t_i adds the special sequence number (0, 1, 2, ...99) to some of the packets it wants to transmit. The number of special packets v_i would send out during T_1 period is predefined. Those packets with special sequence numbers are called special packets. If node v_i happens to be the receiver of node v_{i-1} , it can calculate the packet delivery ratio on link (v_i, v_{i-1}) by collecting how many special packets its gets during T_1 period. Even If node v_i is not the receiver of node v_{i-1} , but it is within one-hop of v_{i-1} , it can still detect how many special packets it is able to listen during T_1 period.

At the end of T_1 period, node v_i compares the packet delivery ratio between itself and the tree nodes on level $i - 1$, chooses the best link and connects itself with the corresponding tree node. If the old parent of node v_i is no longer needed to send

packets to v_i and it does not have any other children, the tree repair mechanism mentioned in Section 4.3 is activated afterwards.

Tree Re-Construction. Although the first mechanism discussed above is able to adjust the multicast tree locally, but it may not satisfy the network condition variance. If the network condition changes a lot, we need to rebuild the multicast tree instead of just locally changing some tree paths. We propose two cases that we need to rebuild the multicast tree. The details are illustrated as follows.

Case 1: At first, we set the source node as the multicast coordinator. The multicast coordinator decides when to reconstruct the multicast tree. The source node is usually the gateway [85, 39]. Because the gateway is the access point to the Internet, it is more powerful than any other mesh node. Thus, gateway is assumed to be more stable and always on during the multicast session. If unfortunately the gateway is down, that is, the source is down, we do not need to worry about whether to reconstruct the multicast structure. Thus, the gateway can function as the multicast coordinator.

When the multicast tree is built initially, each relay node records its ETX, and sends its ETX back to the multicast coordinator along with the tree path. The multicast coordinator records the sum of all the ETX, which is called the *original ETX sum*. Each time period T_2 (for example, 10 minutes), no matter the tree is repaired or the link quality is changed, the related tree nodes need to send their ETX to the multicast coordinator. The multicast coordinator calculates the sum and compares it with the original ETX sum. If the ratio of new sum to the original ETX sum exceeds a threshold (e.g. 130%), the multicast coordinator begin to reconstruct the multicast tree. If the tree node's ETX is not changed during the time period, it

does not need to inform its ETX to the multicast coordinator, so that we can decrease the unnecessary network traffic.

We consider using the ETX sum as a standard to evaluate whether we need to reconstruct the multicast structure. This is because the ETX sum is directly related with our optimization objective – minimize the number of transmission. The ETX sum varies with the change of the network condition. If the ETX sum decreases, we do not need to rebuild the tree. On the contrary, if the ETX sum increases to a threshold, it means the current multicast structure may not be suitable to the current network condition.

Case 2: Although multicast coordinator can gather the ETX variance from all the tree nodes, it only gets the network condition within the tree. The network condition outside of the tree would be also important for the multicast application, because the outside-tree area may provide better paths after network condition is changed. Thus, for a fixed period T_3 (say 30 minutes), we force the multicast coordinator to rebuild the tree no matter the ETX variance within the tree exceeds the threshold or not. Because T_3 is not a small period, the network is able to afford the overhead of rebuilding the multicast.

By setting the appropriate time period and threshold in case 1 and 2, the new multicast tree can adapt to the network situation in time and avoid frequent tree construction. How to set the time period and threshold is our future research work.

4.4 Performance Evaluation

We study the performance gains of the proposed MLRM algorithm compared with the MCM algorithm [85] (the best multicast algorithm for the multi-channel mesh networks so far) using the following metrics:

- **Throughput:** the average number of packets each multi-receiver receives during a time unit (10 seconds).
- **Delay:** the average time it takes for a packet to reach the destination after it leaves the source.

We modify the NS2 simulator to support multiple network interfaces and multiple channel assignment on each node with a transmission range of 250m and a carrier sensing range of 550m. We evaluate the MLRM and MCM algorithms in a variety of different settings. For each setting, 50 different graphs are randomly generated in a flat area of 1000m by 1000m. The source and the destinations are randomly selected, and traffic is generated by constant bit rate (CBR) sessions. The default IEEE 802.11 MAC configuration is used in NS2, which supports multicasting using different broadcasting rates in different scenarios. The packet size for all traffic is set as 512 bytes. The link capacity is 11 MPBS, and the transmission rate is set to be 1000 packet/s.

Table 4.1. Throughput Comparison

	MLRM	MCM
30 nodes	2368	1512
60 nodes	1890	973

4.4.1 Throughput

In this subsection, the network throughput is evaluated on the MLRM and MCM algorithms.

Base Configurations

We evaluate the throughput of the MLRM algorithm and the MCM using two standard configurations. In one configuration, we have a 30-node network; in the second configuration, we have a 60-node network. In both configurations, the number of multi-receivers is 5, the number of available channels is 6, and the link loss ratio on each link is a randomly generated value between 10% and 90%.

The resulting throughputs of the two configurations are shown in Table 4.1. We can see that MLRM achieves much higher throughput than MCM in both configurations. This is because MLRM is based on the HW-SCDS model that explicitly represents link heterogeneity, whereas MCM assumes all links are the same even though they are not. Thus, MLRM chooses better links with higher delivery ratios whereas MCM only minimizes the number of relay nodes but may choose poor quality links as a result.

MLRM’s advantage over MCM is greater in the denser network with 60 nodes. The reason is that in this denser network, there are also more links. Thus, MLRM

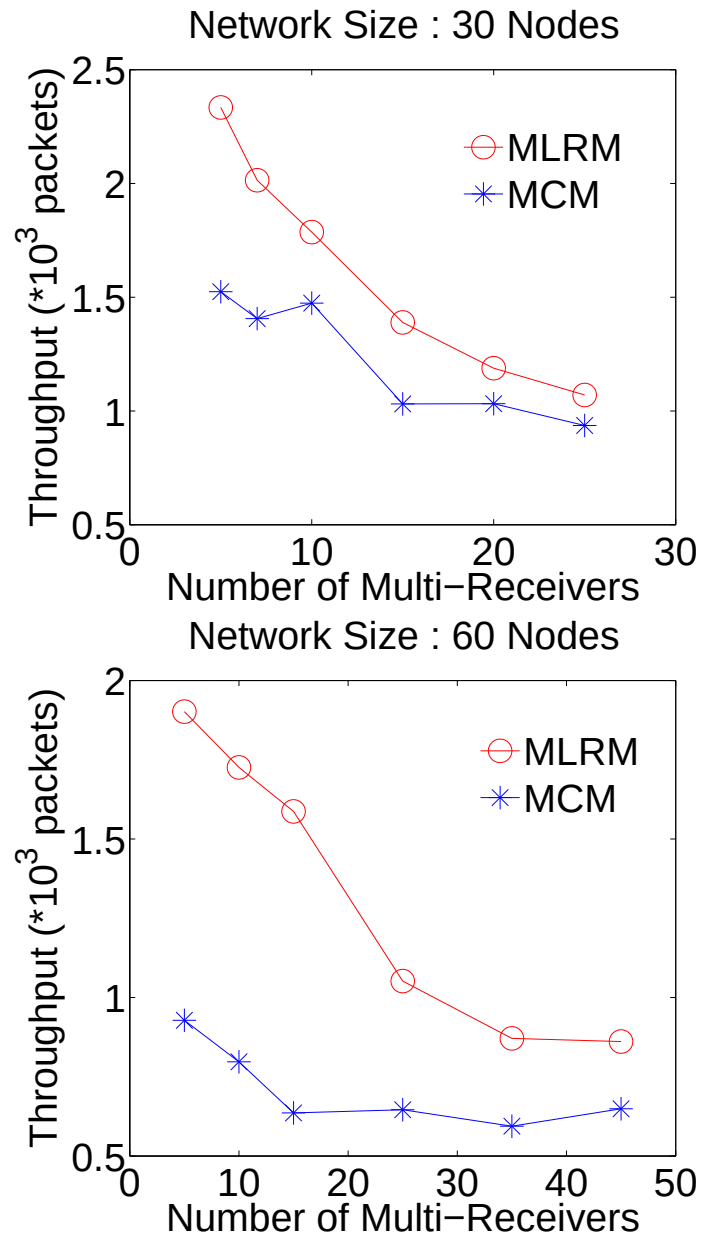


Figure 4.8. Impact of Number of Receivers

has more opportunities to choose good links, and this leads to a bigger difference in throughput.

Impact of Number of Receivers

We now assess the impact of the number of receivers on the throughput of the MLRM and MCM algorithms. We use our two base configurations but vary the number of receivers from 5 to 25 in the 30 node network and 5 to 55 in the 60-node network. The throughput results for both configurations are shown in Fig. 4.8.

The results indicate that MLRM significantly improves throughput for all numbers of receivers tested. However, MLRM's improvement decreases as the number of receivers increases. For the small network, MLRM's improvement is 13% given 25 receivers, and for the large network, MLRM's improvement is 32% given 55 receivers. One reason for this is that the interference increases as the multicast structure increases in size. In such cases, minimizing the number of relay nodes to minimize interference becomes more important. However, we see in all cases that MLRM still significantly outperforms MCM.

Impact of Number of Channels

We now assess the impact of the number of channels on the throughput of the MLRM and MCM algorithms. We use our two base configurations but vary the number of available channels from 2 to 12. The throughput results for both configurations are shown in Fig. 4.9. The results indicate that MRLM significantly and consistently improves throughput for all numbers of channels. One observation is that since MLRM

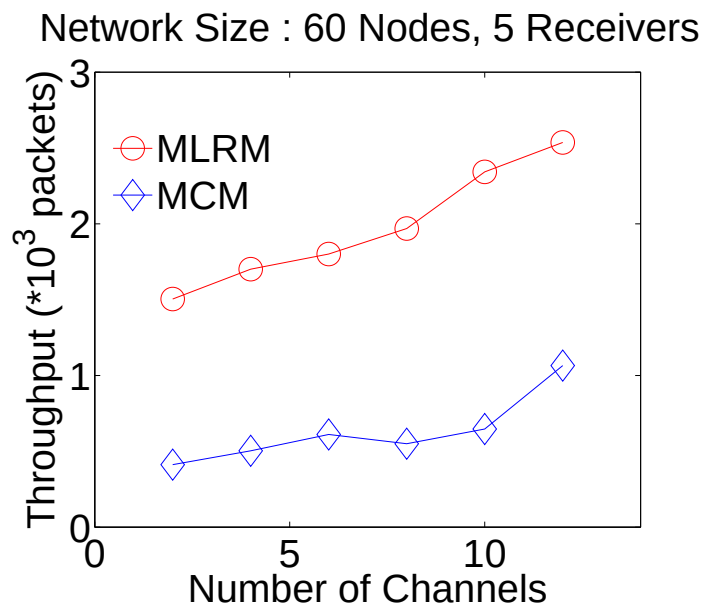
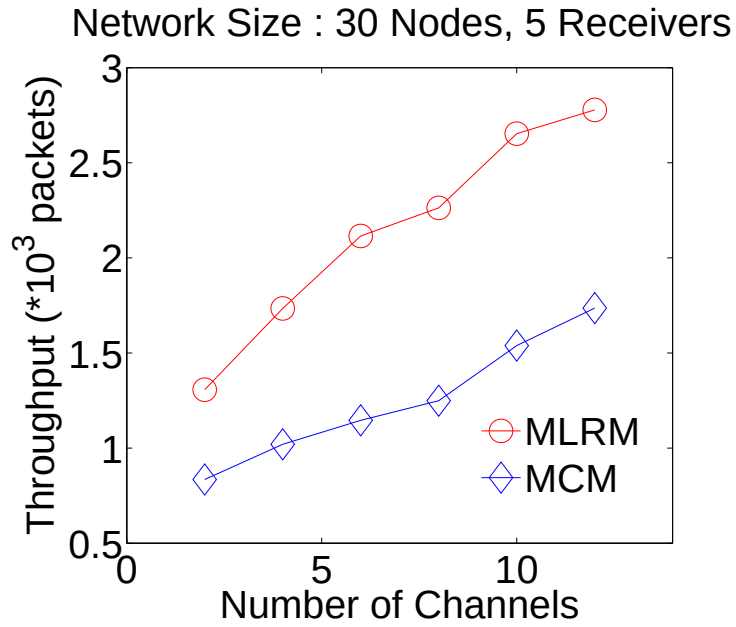


Figure 4.9. Impact of Number of Channels

may use more relay nodes than MCM, MLRM may benefit from increasing the number of channels more than MCM.

Impact of Link Quality

We now assess the impact of the varying link quality on the throughput of the MLRM and MCM algorithms. We use our two base configurations but we vary the link quality as follows. We maintain our upper bound of 90% on the packet delivery ratio (which is $1 - \text{link loss rate}$), but we vary the lower bound L on the packet delivery ratio from 10% to 60%. That is, the packet delivery ratio of each link is randomly generated between L and 100% where L varies from 10% to 60%. The throughput results for both configurations are shown in Fig. 4.10.

In all cases, MLRM outperforms MCM. However, as L increases, the throughput of MCM gets closer to the throughput of MLRM. The reason is that increasing L means that the variance in link quality shrinks. Thus, MLRM gains less and less benefit from its consideration of link quality. This shows that MLRM is better in networks with significant link heterogeneity.

4.4.2 Delay Comparison

In this subsection, we evaluate the delays incurred by MLRM and MCM by measuring the average time each packet takes to reach its destination. For these experiments, we used 50 nodes and 10 channels, and we varied the number of multi-receivers from 5 to 40. The packet delivery ratio of each link is randomly generated between 10% and 90%. The results are shown in Fig. 4.10.

Network Size : 60 Nodes, 5 Receivers

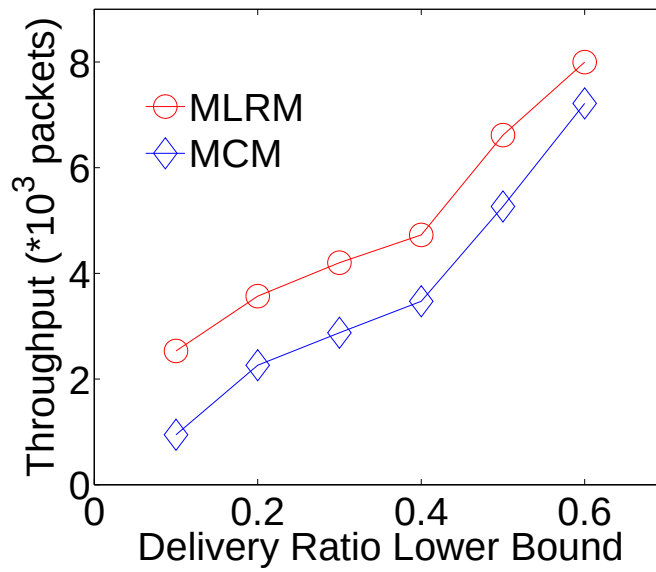


Figure 4.10. Impact of Delivery Ratio

Network Size : 50 Nodes, 10 Channels

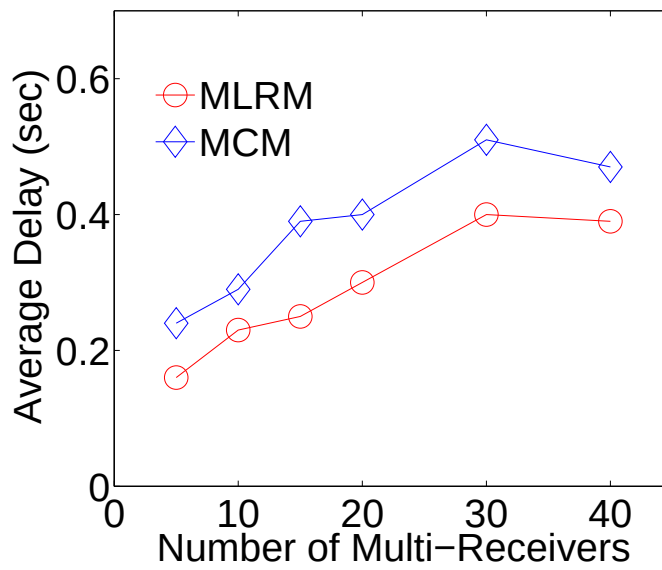


Figure 4.11. Delay Comparison

MLRM has much shorter delays than MCM for all numbers of multi-receivers. The main reason is that since MLRM chooses more reliable links, it experiences fewer retransmissions. Thus, each packet arrives at its destination much more quickly. This is particularly important for real-time multicast applications.

4.5 Summary

This chapter investigates multicast communication in the link-heterogeneous wireless mesh networks where throughput maximization is most important. Compared with previous techniques, our work more accurately models link heterogeneity in WMNs. We model the mesh network as a weighted directed graph, where a link's weight represents its quality. An optimal multicast structure in this edge-weighted directed graph is an MHW-SCDS. We prove finding an MHW-SCDS is NP-hard. We propose a greedy algorithm MLRM for this problem. Simulation results show that our MLRM algorithm significantly outperforms the current best multicast algorithm in WMNs in terms of throughput and delay.

CHAPTER 5

Efficient Opportunistic Multicast via Tree Backbone for Wireless Mesh Networks

Our previous multicast research work in Chapter 3 and Chapter 4 were based on traditional multicast routing. Most traditional multicast protocols for wireless multihop networks discover the least cost or highest throughput paths to reach the destinations. Compared with their wired counterparts, they build efficient multicast structures based on the wireless communication facts: (i) the local broadcast enables multiple neighbors to receive the packet, (ii) the packet loss ratios cannot be ignored, and (iii) the packet delivery ratios are not the same on different links. Thus, they usually choose one or multiple next-hop destinations for each relay node, and the links between selected one-hop neighbors have good quality in the multicast structure.

However, these protocols do not fully take advantage of the spatial characteristics of wireless communication. It is unpredicted that which neighbors can receive the packet in the current transmission. That is, when a packet is transmitted, it is possible that some neighboring nodes receive the packet while the designated next-hop destination does not. Therefore, some research work [14, 15] has demonstrated that utilizing the cooperative diversity to send packets through multiple relay nodes concurrently can further improve the system throughput, including the multicast throughput [16]. This new routing strategy is known as opportunistic routing (OR).

In opportunistic routing, any node that overhears the packet transmission is encouraged to forward the packet if it is closer to the destination. More concurrent relay nodes give each transmission more chance to make progress. The multicast extension of OR is discussed in [16], but it does not build any efficient multicast structure that is able to reduce the packet transmission. As we know, OR brings an increase of the number of transmissions because more neighbors participate in forwarding the packets. The increase of transmissions not only consumes more bandwidth resource, but also brings more local interference among nearby transmissions. It may result in lower throughput. However, increasing throughput is paramount for multicast communication in wireless mesh networks. We have to reduce the number of transmissions when we exploit opportunistic routing for multicast, which helps to improve the system throughput. On the contrary, traditional multicast protocols design the efficient multicast structure that reduces packet transmissions, but they lack of the spatial reuse of wireless communication that also contributes to increasing throughput.

In order to achieve higher multicast throughput, we propose the opportunistic

multicast protocol by adopting the OR strategy with an efficient multicast structure. To utilize the spatial diversity, the key step of this protocol is to design a multicast tree backbone that is different from a traditional multicast tree. Tree backbone specifies the packet transmission direction instead of designating the exact next-hop destinations for the relay nodes. The “adjacent” backbone nodes may be multi-hop away in the network. An efficient backbone structure must minimize the number of transmissions.

In this chapter, we prove that computing a backbone that minimizes the expected number of transmission is NP-hard, and we devise two heuristic algorithms that appear to work well for building an efficient backbone in single-rate WMNs. Based on the tree backbone, packets then self route from the upstream backbone node to the downstream backbone nodes by OR until they arrive at the destinations. Therefore, our protocol not only takes advantage of spatial diversity of wireless communication by utilizing OR, but also reduces the unnecessary packet transmissions by building an efficient tree backbone.

The recent trend in wireless communication is to enable devices with multiple transmission rates [17, 18, 19]. Generally speaking, low-rate communication provides a long transmission range, while high-rate has to occur at a short scope. The variance of transmission range implies the variance of the neighboring node set, which leads to different spatial opportunities. The inherent rate-distance tradeoff for opportunistic unicast routing has had impact on performance [18]. It is intuitive to expect that this trade-off also affects opportunistic multicast. In this chapter, we investigate this trade-off and further propose a Euclidean opportunistic multicast protocol. This

protocol minimizes the number of transmissions for multi-rate WMNs by devising a Euclidean backbone structure as well as an efficient rate selection scheme.

The rest of this chapter is organized as follows. Section 5.1 describes the background and the motivation. Section 5.2 proposes the opportunistic multicast protocol in single-rate WMNs. We apply the basic idea of opportunistic multicast to multi-rate WMNs in Section 5.3. Several companion mechanisms for our protocols are introduced in Section 5.4. Section 5.5 presents simulation results, and the last section summarizes this chapter.

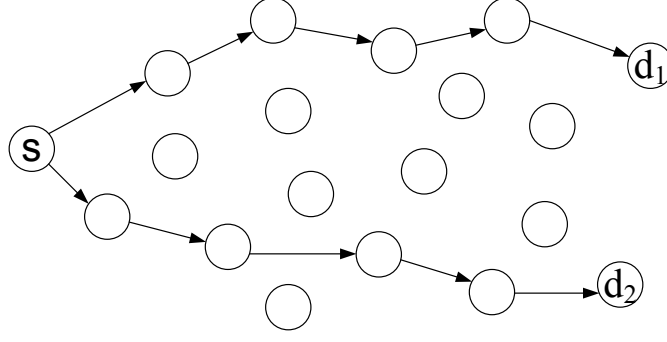
5.1 Background and Motivation

We start from the underlying network model by introducing some terminology and the basic idea of opportunistic routing, which is followed by the motivation.

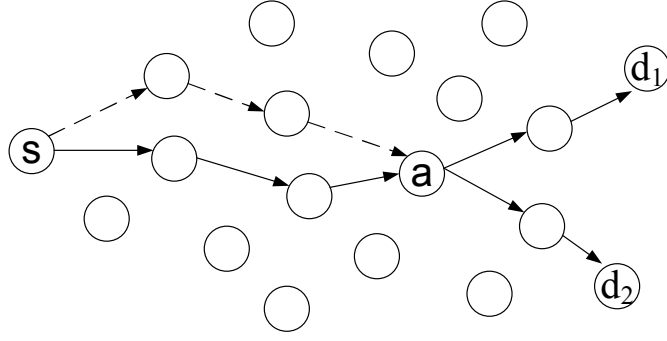
5.1.1 System Model

To simplify the system model, we consider the network as a weighted graph $G = (V, E)$ with function w , where V represents the set of gateways and mesh routers, and E represents the physical links among neighboring nodes (the node refers to the mesh router or the gateway in this chapter). Individual WMN links may have different qualities for a variety of reasons, thus we use the weight on each edge to denote the packet delivery ratio.

Each node n_i ($1 \leq i \leq |V|$) can transmit a packet at K different rates $R^1, R^2, \dots, R^K$. We assume that each node has the same fixed communication range



(a) Natural Extension



(b) Efficient Multicast Path

Figure 5.1. Motivation Example

under the same rate. That is, if nodes u and v use the same rate, and if u can transmit packets directly to node v (and vice versa), there is a link (u, v) in E . In this chapter, we first consider the multicast issues in single-rate WMNs, then we apply our idea to multi-rate WMNs.

5.1.2 Basic of Opportunistic Routing

There are different variations of OR. In the following, we describe the basic details common to all OR schemes.

A crucial component of OR is a forwarding set F_{ij} for sending a message from node n_i to node n_j . This set F_{ij} is carefully selected to minimize the number of forwarding

nodes while maximizing the throughput improvement. Furthermore, F_{ij} is an ordered set where typically nodes which are closer to destination node n_j have higher priority. In this chapter, we define closeness to n_j as the number of transmissions to move a packet along the best traditional route to n_j [14, 16, 84].

OR begins with the sender n_i broadcasting a batch of packets. Its forwarding candidates continue the forwarding based on their relay priority. That is, higher priority forwarding nodes are given the first chance to forward packets. When receiving packets, each forwarding node also determines if the newly received packet it receives should be forwarded, either by explicit coordination or by exploiting network coding properties. The above process repeats until the destination informs the source that enough packets have been received and the transmission can stop.

5.1.3 Motivation Example

Like traditional unicast routing, traditional multicast protocols discover the least cost or highest throughput paths to the destinations. This multicast strategy is effective in wired networks, but not efficient in wireless networks, since it does not exploit the spatial characteristic of wireless communication. For example, building a shared tree is a common way in traditional multicast protocols, where transmissions to different destinations may share some hops in the tree to minimize the bandwidth cost. However, the shared tree designates the next-hop destination for each relay node. As a consequence, there are no spatial opportunities for each transmission. We can safely conclude that the exact shared tree is not suitable for opportunistic

multicast.

However, if we utilize OR to realize the multicast application without any structure, we have to face another drawback. For instance, a natural multicast extension from OR is briefly introduced in [16], which requires the packet self route to all the destinations by OR. Although utilizing the spatial diversities, this extension also brings unnecessary transmissions and results in more interferences.

There is an example in Fig. 5.1, where s is the source, and nodes d_1 and d_2 are destinations. Under the natural extension, to deliver a packet, two copies may travel along different paths to d_1 and d_2 by OR. Fig. 5.1(a) shows a case of natural extension, which requires totally 10 hops. However, if we allow the transmissions to distinct destinations share some hops, it can decrease the total number of transmissions. For example, the packet is first desired to self route to node a , then it is split into two copies that would be forwarded to the two destinations by OR respectively. Fig. 5.1(b) shows one case (solid line) of this improvement strategy, which only needs 7 hops. Thus, we need to devise an efficient opportunistic multicast protocol that achieves high throughput by building an efficient backbone to reduce transmissions.

5.2 Opportunistic Multicast in Single-rate WMNs

In this section, we first introduce the definition of tree backbone (TB), and explain the basic idea of our Opportunistic Multicast (OM) protocol. In order to save bandwidth and decrease interference, the tree backbone should minimize the number of transmissions along it. We then describe how to calculate the number of transmissions on

a TB, and prove that computing a TB with the minimum number of transmissions in single-rate WMNs is NP-hard. Afterwards, we present two heuristic algorithms to construct an efficient TB.

5.2.1 Basic Idea

Opportunistic multicast builds the tree backbone instead of a multicast tree, which both allows the spatial opportunities given by OR and minimizes the number of transmissions by letting packets transmitted to different destinations share some hops.

Definition 6 *Including a source s and a set of destinations $D \subseteq V(G)$, a set of nodes $T \subseteq V(G)$ is selected to be the backbone of multicast structure. The nodes in T are called **tree backbone nodes**.*

Definition 7 *Among tree backbone nodes, if we designate that packets need to be delivered from backbone node a to backbone node b by OR, we say that there is a **direction** $a \rightarrow b$. Node a is called the **upstream backbone node** of b , and b is called the **downstream backbone node** of a ,*

Definition 8 *In graph G , given a source node s , a set of destinations D , a set of tree backbone nodes T , and a set of directions R , the multicast backbone $TB(s, D, T, R)$ is called **tree backbone**.*

After the tree backbone (TB) is built, starting from the source node, the packets self route to the source's downstream backbone nodes by opportunistic routing. After the packet arrives at one tree backbone node, say t , it continues routing to t 's downstream backbone node, until it reaches the destination. At each tree backbone node,

if the backbone node has multiple downstream backbone nodes, the packet is split into multiple copies and routed to the corresponding downstream backbone nodes with different random paths by OR.

For example, in Fig. 5.1, $T = \{s, a, d_1, d_2\}$ and $R = \{s \rightarrow a, a \rightarrow d_1, a \rightarrow d_2\}$. So, the packets are desired to be delivered from s to a , then from a to d_1 and d_2 respectively. Node a is the downstream backbone node of s , while d_1 and d_2 are the downstream backbone nodes of a . Note that, since routing from the upstream node to the downstream node uses OR, different packets may travel along different paths. For instance, from s to a , the first packet may route along the solid line, while the second packet may route along the dotted line.

A good metric to evaluate the effectiveness of a TB is the expected number of transmissions for one packet to reach all the destinations through the TB. This is because the less transmissions means less bandwidth cost and less interference, which results in higher throughput and smaller delay. For short, we call this metric the **cost weight** of TB. In order to improve multicast performance, we need to compute a TB with the minimum cost weight. We refer to this as a **Minimum Tree Backbone** or MTB. We prove that computing an MTB is NP-hard by proving its corresponding decision problem, TB-D defined below, is NP-hard.

INSTANCE: A weighted graph $G = (V, E)$, a weight function w on E , a source node $s \in V$, a subset $D \subseteq V$, and a positive number K .

QUESTION: Is there a $TB(s, D, T, R)$ for s and D of cost weight K ?

5.2.2 NP-Hardness Proof

Lemma 3 *TB-D is NP-hard*

First, we explain how to calculate the expected number of transmissions Z^{sd} for one packet to be transmitted from source s to destination d in opportunistic routing.

For any two nodes i and j , let $i < j$ represent that i is “closer” to d than j , that is, i has smaller ETX to d than j . Let ϵ_{ij} denote the packet loss ratio from i to j . Let z_j^{sd} be the expected number of transmissions that forwarder j must take to route one packet from s to d . The expected number of packets that j must forward, denoted by L_j^{sd} , is [16]

$$L_j^{sd} = \sum_{i>j} (z_i^{sd}(1 - \epsilon_{ij}) \prod_{k<j} \epsilon_{ik}) \quad (5.1)$$

Note that L_s^{sd} is 1 since source s generates the packet. From Eq. 5.1, the authors in [16] deduce that the expected number of transmissions that j must make is:

$$z_j^{sd} = \frac{L_j^{sd}}{1 - \prod_{k<j} \epsilon_{jk}} \quad (5.2)$$

Suppose there are N nodes in the network. The calculation of z_j^{sd} can be achieved in $O(N^2)$ [16]. Furthermore, we compute the total expected number of transmissions Z^{sd} in the network by summing up all the nodes’ expected number of transmissions, that is,

$$Z^{sd} = \sum_{j \in V(G)} z_j^{sd} \quad (5.3)$$

Second, based on the above result, given a TB(s, D, T, R), for any direction $i \rightarrow j \in R$, the expected number of transmissions for delivering a packet from i to j is Z^{ij} . Hence the cost weight of the TB is:

$$\lambda_D^s = \sum_{\forall i \rightarrow j \in R} Z^{ij} \quad (5.4)$$

Third, we construct a complete graph G_c with a weight function w , where $V(G_c) = V(G)$. For any edge (u, v) on G_c , the weight $w(u, v) = Z^{uv}$, which denotes the total expected number of transmissions occurring in the network for one packet transmitted from u to v by OR. It takes $O(N^2)$ operations to calculate z_j^{sd} for each node j , thus it takes $O(N^3)$ operations to compute Z^{sd} , and $O(N^5)$ to get graph G_c . Graph G_c is constructed at the beginning of network, and it will not change as long as the network topology does not change, thus computing G_c is a one-time activity regardless how many multicast sessions are generated in the network.

For a given TB, we can find a corresponding Steiner tree S on G_c , where $V(S) = T$. An edge (u, v) appears on S if and only if there is a direction $u \rightarrow v$ in R . The cost weight of the TB is equal to the weight sum of S . It is well known that computing a Steiner tree with weight sum K in a weighted graph is NP-hard [86], thus TB-D is also NP-hard.

Corollary 2 *Computing an MTB is NP-hard*

The decision problem TB-D is NP-hard, so the optimization problem of computing an MTB is also NP-hard.

5.2.3 Heuristic Algorithms

We propose two heuristic algorithms for MTB: one is minimum spanning tree (MST)-based, and the other is Steiner tree-based. For short, we call them MST-B and ST-B, respectively. For each algorithm, we suppose that the above complete graph G_c is constructed at the beginning of the network.

MST-B algorithm. In this algorithm, we first construct a complete graph G'_c induced from G_c with $V(G'_c) = \{s\} \cup D$, where the weight on each edge $w(u, v) = Z^{uv}$ ($u, v \in V(G'_c)$).

Afterwards, we use Prim's algorithm [81] to compute an MST on G'_c . Prim's algorithm starts from s , and grows until the tree spans all the destinations. At each step, an edge e with the minimum possible weight is added to the partially constructed tree A that connects A to an isolated destination. During the process of computing an MST, we spontaneously get the corresponding $TB(s, D, T, R)$ as follows:

1. $T = \{s\} \cup D$.
2. For any edge (u, v) in the MST, if u is added to the tree before v , there is a direction $u \rightarrow v \in R$.

Given G_c , we can build G'_c in $O(M^2)$ time (M is the multicast group size). In addition, to build a MST on G'_c takes $O(M^2)$ operations. Therefore, the time complexity of MST-B is $O(M^2)$.

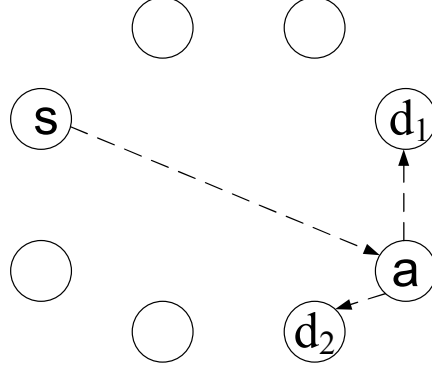


Figure 5.2. ST-B Example

ST-B algorithm. In this algorithm, under the well-known Takahashi-Matsuyama (T-M) heuristic [87], a Steiner tree S is built by an incremental approach to span over source s and destination set D in G_c . Initially, the tree contains only s . At each iteration, the nearest unconnected destination to the partially constructed tree S is found and the least-cost path between them is added to the tree. Here, the least-cost path $P(u, v)$ refers to a path that connects u and v , and the weight sum on $P(u, v)$ is the smallest among all paths between u and v .

During constructing S , we can get the backbone nodes of the corresponding TB as follows:

1. Initially, $T = \{s\} \cup D$.
2. At each iteration, when a least-cost path $P(u, v)$ is added to S , $T = T \cup \{u\} \cup \{v\}$.

After the backbone nodes are determined, the set of directions for TB is also discovered by this rule: for any two nodes $u, v \in T$, if there is not any other backbone node on path $P(u, v)$ in S , and u is on path $P(s, v)$ in S , there is a direction $u \rightarrow$

$v \in R$.

We use a simple example to illustrate the process. There is a complete graph G_c in Fig. 5.2 (For ease of reading, we do not draw the edges on G_c), where s is the source and d_1, d_2 are destinations. Initially, s is included in the tree. Next, since d_1 is “closer” to s , the least-cost path (s, a, d_1) is added to the tree. Afterwards, since a is the closest tree node to d_2 , the least-cost path (a, d_2) is added. Based on the Steiner tree, we can build the tree backbone TB with $T = \{s, a, d_1, d_2\}$ and $R = \{s \rightarrow a, a \rightarrow d_1, a \rightarrow d_2\}$

The time complexity of T-M heuristic is $O(N^2)$, and searching for R also takes $O(N^2)$ operations. Therefore, the time complexity of ST-B is $O(N^2)$.

The major difference between MST-B and ST-B is that ST-B allows some non-multicast group members (Steiner points) to be tree backbone nodes while MST-B only takes group members as backbone nodes. The time complexity of MST-B is much less than ST-B when the group size M is much less than N . However, adding Steiner points to the tree backbone can further decrease the cost weight at the cost of computing complexity.

5.3 Euclidean Opportunistic Multicast in Multi-rate WMNs

Multi-rate capacity is a common feature of wireless communication. On one side, a higher data rate can be used to increase throughput, but it also has shorter trans-

mission range and hence more hops to reach the destination. Besides, there are few spatial opportunities due to the low neighbor diversity in one hop. On the other side, lower data rate often has a longer transmission range and hence less hops in the selected path. The higher neighbor diversity brings more spatial opportunities, but the low rate disadvantage may counteract the above benefit. The inherent tradeoff between rate and distance is hereby worthy of a careful study.

5.3.1 Design Consideration

A local metric, called Expected Advancement Rate (EAR) [18], has been proposed to find a best rate for each node:

$$EAR_{i,d}^{F_{i,d}} = R_i \sum_{q=1}^r a_{ii_q} p_{ii_q} \prod_{k=0}^{q-1} (1 - p_{ii_k}) \quad (5.5)$$

Here, a_{ii_q} is defined as the packet advancement, which is the distance between transmitter n_i and destination n_d subtracting the distance between candidate n_{i_q} and the destination. $F_{i,d}$ is the forwarding candidate set of n_i , and the node order in $F_{i,d}$ is based on the relay priority.

The metric addresses the rate-distance tradeoff in opportunistic unicast, hence it achieves good results in simulations [18]. However, this metric requires modification in multicast. When a node selects a rate, it should consider the expected bit advancement per second toward each destination, not just a specific one. Thus, we propose the Multicast Expected Advancement Rate (MEAR), a naive extension of EAR, by summing up the EAR to each destination:

$$MEAR_{i,D} = \sum_{h=1}^{|D|} EAR_{i,d_h}^{F_{i,d_h}} \quad (5.6)$$

For each node, at each transmission rate R_j , we calculate the largest MEAR based on Eq. 5.6, and then we select the best rate that yields the largest MEAR. However, this naive metric is not suitable for opportunistic multicast. In a tree backbone, the packets have to travel from the source to a series of intermediate backbone nodes by OR until they arrive at the destinations. When a forwarder receives packets, it only cares how to route the packets to their next downstream backbone nodes rather than the destinations, so the routing metric should take the intermediate backbone nodes into considerations. Unfortunately, MEAR ignores the tree backbone nodes, so it does not serve well for opportunistic multicast.

On the other hand, we also need to propose a new tree backbone construction for multi-rate WMNs. Previous TB construction is based on the number of transmissions Z^{sd} on each pair of source s and destination d . Calculating Z^{sd} requires the information of link delivery ratios. However, for each link, the delivery ratio is different under different rates [18]. Before the rate allocation is finished, it is impossible to calculate the number of transmissions and determine the tree backbone.

5.3.2 Euclidean Tree Backbone and Rate Selection

We assume that a longer geometric distance requires more transmissions to reach the destination. This assumption is straightforward, but it does not apply to any case, since the number of transmissions depends on both the network topology and the

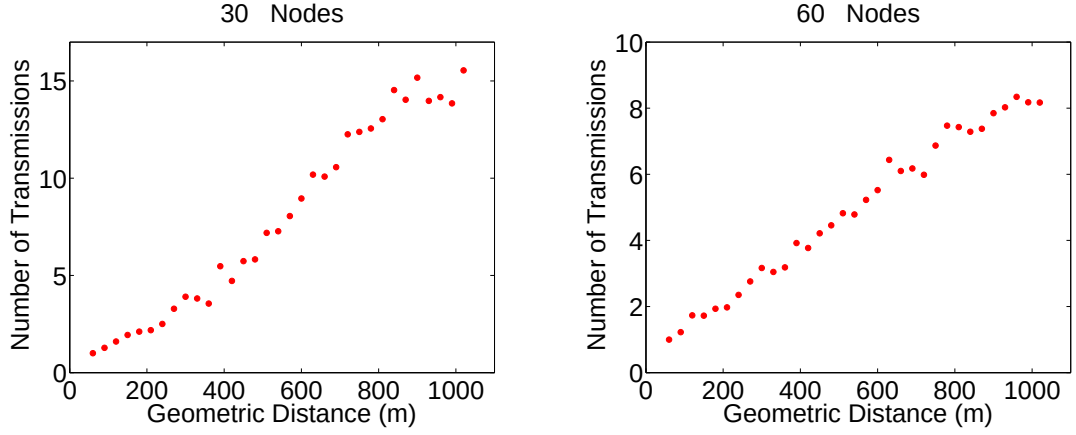


Figure 5.3. Relation between Distance and Transmissions

link quality. Fortunately, on average, the trend of number of transmissions increases monotonically with distance, which is demonstrated by the following simulations.

Our simulation area is $1000\text{m} \times 1000\text{m}$, where the transmission range of each node is 250m. In the first simulation, 30 nodes are randomly generated, while 60 nodes are generated in the second. For each simulation, we randomly choose the source-destination pairs, and we obtain their geometric distance as well as the number of transmissions on the shortest path. We group the collected data into brackets of distance $[30n, 30(n+1)]$, where $n = 2, 3, \dots, 34$. For each bracket, we collect 50 samples and compute the average number of transmissions. From the results in Fig. 5.3, we observe that the trend of average number of transmissions almost monotonically increases with the geometric distance.

Based on the above consideration, we propose the Euclidean Opportunistic Multicast (EOM) in multi-rate WMNs. To distinguish the tree backbone in EOM from the tree backbone we proposed in the previous section, we call it **Euclidean tree backbone** (ETB). After the ETB is built, we also give a new metric to help the

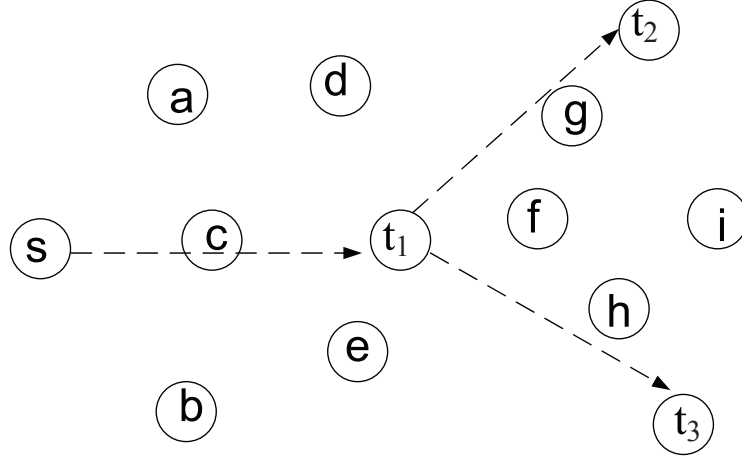


Figure 5.4. EOM Example

transmitter to choose the rate.

The **first step** is to build a Euclidean Steiner tree that is rooted at the source and spans over all the destinations. Generating an optimal Euclidean Steiner tree is NP-hard [88], and there are many heuristics [89, 90, 91] to address this problem. We use a simple heuristic: Initially, the tree contains only the source node. At each iteration, the geometrically nearest destination is added to the partially constructed tree with a corresponding edge. This process is repeated until all the destinations join the tree.

The **second step** is to determine the direction set R . For any edge (u, v) in the Euclidean Steiner tree, if u is added to the tree before v , there is a direction $u \rightarrow v \in R$.

As an illustration, there is a network in Fig. 5.4, where s is the source, and t_1 , t_2 , and t_3 are destinations. For simplicity, we do not label the network topology. Initially, the ETB only contains s . Within the remaining set of destinations, node

t_1 is geometrically closest to s , then t_1 is added to the tree as well as the direction $s \rightarrow t_1$, since s is added to the tree before t_1 . In the following steps, node t_2 and direction $t_1 \rightarrow t_2$ are added to the ETB, then t_3 and direction $t_1 \rightarrow t_3$ are added.

The major difference between ETB and TB is that we use geometric distance instead of ETX as measurement of closeness in ETB, so that ETB can be built independent of rate allocation.

The **third** step is to allocate rates to nodes. As discussed before, the forwarder cares how to quickly route the packet to its next downstream backbone node. Different packets received may aim to different next downstream backbone nodes, thus the forwarder needs to consider all those downstream backbone nodes, which are defined as the **targets** of the forwarder.

The target set of different nodes is different due to their different locations. For instance, in Fig. 5.4, t_1 should not be the target of f , while it should be the target of c , since the packet that f usually gets is delivered towards t_2 or t_3 , not t_1 .

We propose a heuristic approach to search the target set of each node n_i : for a ETB node t_k , if n_i is closer to source s than t_k ($|n_i - s| < |t_k - s|$), and n_i is closer to t_k than s ($|n_i - t_k| < |s - t_k|$), t_k is n_i 's target candidate. After we determine all the target candidates of n_i , we check the directions among the candidates. For any candidate t_k , if it is the downstream backbone node of another candidate, t_k is removed from the candidate set. Finally, the remaining nodes in the candidate set are n_i 's targets. There is an example in Fig. 5.4. For node c , the target candidate set is $\{t_1, t_2, t_3\}$. Since t_2 and t_3 are downstream backbone nodes of t_1 , the final target set of c is $\{t_1\}$.

Afterwards, we modify MEAR to a new metric: Improved-MEAR (I-MEAR) based on the target set:

$$I - MEAR_{i,Q_i} = \sum_{h=1}^{|Q_i|} EAR_{i,d_h}^{F_i,d_h} \quad (5.7)$$

Here, Q_i is the target set of n_i . For each rate R_j , we calculate the largest I-MEAR based on Eq. 5.7, and then we select the best rate that yields the largest I-MEAR.

5.4 Companion Mechanism

To augment and enhance opportunistic multicast, we introduce two companion mechanisms: (i) routing information distribution, and (ii) membership update.

5.4.1 Routing Information Distribution

WMNs usually have static topology, thus at the beginning of each multicast session, the tree backbone construction process can be performed in the gateways, which are usually the sources of multicast sessions and may have more powerful computation ability than other mesh routers.

The opportunistic routing from one upstream backbone node to its downstream backbone node does not need an underlying unicast mechanism, which is similar to the stateless protocols [92, 93], but good at the spatial opportunities. However, traditional stateless multicast protocols for multi-hop wireless networks, such as DDM [26] and GMP [94], do not build any efficient structure to minimize transmissions.

Moreover, they store the destinations in the packet header, which results in large overhead when the group size is not small.

In order to avoid this drawback, opportunistic multicast distributes the routing information to the backbone nodes. That is, after the gateway builds the tree backbone, the set of directions R is multicast to the backbone nodes. For each backbone node, it only records its downstream backbone nodes. When a backbone node receives a data packet, it clears the address list in the packet header, and puts its downstream node in the list as the packet's next stage destination. For example, in Fig. 5.1, node s attaches a in the packets' header. And node a attaches d_1 or d_2 when it receives packets from s , and then forwards the packets.

In addition, if the destinations do not change, the backbone will not change either. So, distributing the route information also avoids recomputing the tree backbone at each backbone node, which reduces the processing overhead in forwarding packets.

5.4.2 Membership Update

Our multicast protocols have to be augmented by a membership update mechanism, since some destinations may join or quit the group during the multicast session. A straightforward way is that when a node wants to join the group or a destination decides to quit, this node sends a "join" or "quit" request to the source that is usually a gateway through a best route with traditional routing. As soon as the source receives the request, it re-computes the tree backbone and distributes the routing information to all the backbone nodes. Although simple, this method leads

to frequent computations in the source and frequent information dissemination in the network.

In order to address the drawback, we design a distributed mechanism. If a destination d wants to leave the group, there are two cases: d has no downstream backbone nodes or d has downstream backbone nodes.

For the first case, we present an approach to remove unnecessary directions in the tree backbone. Node d sends a “quit” request to its upstream backbone node u with traditional routing. If u has only one downstream backbone node, then it stops forwarding packets to d . Moreover, if u is not a destination, it sends out a message to its upstream backbone node with traditional routing, indicating that it is no longer a tree backbone node. Afterwards, the upstream backbone node of u will do the same thing as u does. The process continues until one ancestor of d has more than one downstream backbone node or it is a destination.

For the second case, if d has downstream backbone nodes $t_1, t_2 \dots t_x$, except for notifying d 's ancestors in the backbone, we need to search for an alternative upstream backbone nodes for $t_1, t_2 \dots t_x$. Node t_i ($1 \leq i \leq x$) requires an active nearby backbone node that is able to forward a packet to t_i with the minimum transmissions. t_i performs this discovery process by searching in a table that records Z^{ij} for any pair of nodes i and j in the network. This table is created at the source and delivered to each backbone node at the beginning of the network. After node t_i decides its new upstream backbone node, say node g , it sends out a request to g . From then on, g will forward packets to t_i .

If node v wants to become a member of an existing group, it can also search the

table to find out which one will be its best upstream backbone node. Afterwards, it sends out a request for connecting to the nearby backbone node.

Note that if the number of group members joining or quitting the group exceeds a threshold, the new tree backbone generated by the distributed mechanism may not be efficient, thus the gateway should re-compute the tree backbone based on the new group membership.

5.5 Simulations

We evaluate our opportunistic multicast algorithms by comparing them with the natural multicast extension [16] and a traditional multicast algorithm through the following metrics.

- **Throughput:** the average number of packets each destination receives during a time unit.
- **Delay:** the average time it takes for a packet to reach the destination after it leaves the source.

We use an NS2 simulator to simulate a flat area of 1300m by 1300m with varying number of randomly positioned wireless router nodes. We use the default IEEE 802.11 MAC configuration in NS2 that supports multicasting using broadcasting at different base rates. Except for the last subsection, the simulations take place in single-rate WMNs, where the transmission range of each node is 250m. For each scenario, we randomly generate 30 different graphs, where the source and the destinations are also

randomly selected. Traffic is generated by constant bit rate (CBR) sessions, and the packet size for all traffic is set to be 512 bytes.

5.5.1 Impact of Network Size

We evaluate the throughput of our proposed algorithms (ST-B and MST-B), the natural multicast extension and the tree-based algorithm at different network sizes by assigning the number of nodes with 20, 40, and 60. For each network size, we vary the number of receivers from 5 to 55. The results are shown in Fig. 5.5. We can see that building an efficient tree backbone can improve the throughput, since it not only takes advantage of spatial diversity, but also minimizes the number of transmissions.

Compared with MST-B, ST-B further improves the throughput when the receivers are a small portion. This is because ST-B builds a more efficient structure by allowing some non group members to become Steiner points to further reduce transmissions. However, when the number of receivers increases, there is not much difference between the throughput of MST-B and ST-B, since the tree backbone generated by MST-B becomes similar to ST-B. Another interesting result is that, when the number of receivers is not large, the throughput of the large-size network is much more than the small-size because the large-size network has large node density resulting in more spatial opportunities.

The tree-based algorithm shows a low throughput, since it does not take any spatial opportunities. This once more demonstrates that the OR technique can virtually take place concurrently on multiple outgoing links of the same transmitter to achieve

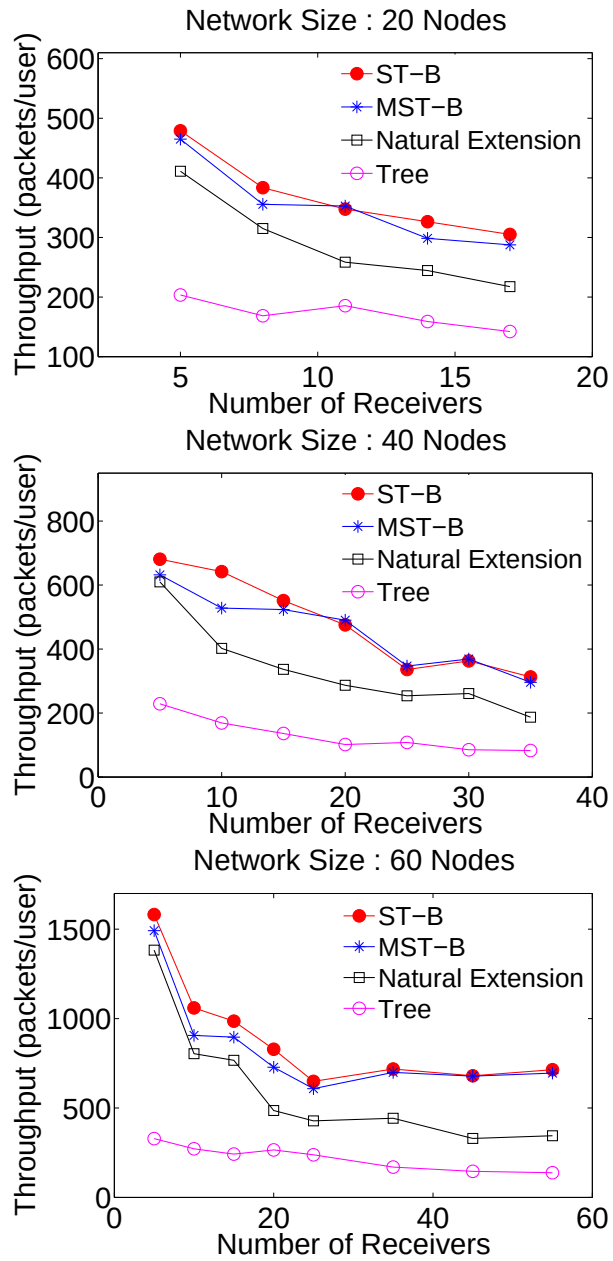


Figure 5.5. Impact of Network Size

higher throughput. We do not simulate the tree-based algorithm any more in the following subsections.

5.5.2 Delay Comparison

In this simulation, we evaluate the delay of our proposed algorithms (ST-B and MST-B) and the natural extension by comparing the average time each packet takes to reach the destination. Fig. 5.6 shows that our algorithms have a much shorter delay than the natural extension. Our efficient tree backbone is able to minimize the number of transmissions as well as interferences, which greatly speeds up the packet delivery. In addition, when the number of receivers increases, the multicast structure becomes bigger, thus the delay of all algorithms increases due to the increasing number of transmissions.

5.5.3 Impact of Multiple Rates

In this simulation, we compare our proposed Euclidean opportunistic multicast (EOM) algorithm with the single-rate opportunistic multicast (OM) algorithm under different rates in multi-rate WNNs. The rates 18, 11, and 6 Mbps are studied, and their corresponding transmission radii are 183, 304 and 395m [18, 95], respectively. Different scenarios take place in a 1000m * 1000m flat area, where 20, 40, or 60 nodes are randomly generated. The results are shown in Fig. 5.7, where OM (X Mbps) denotes the single-rate OM algorithm under rate X Mbps ($X = 18, 11, 6$).

We observe that EOM can achieve higher throughput than the others due to its

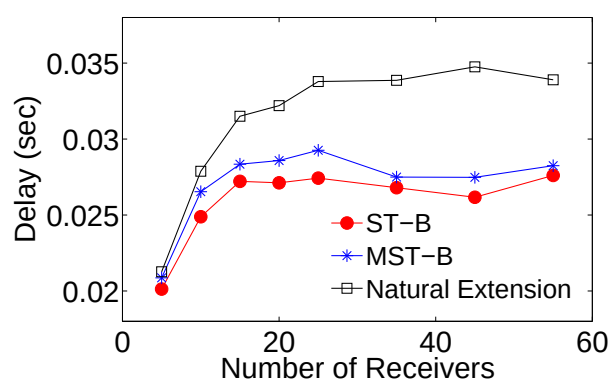
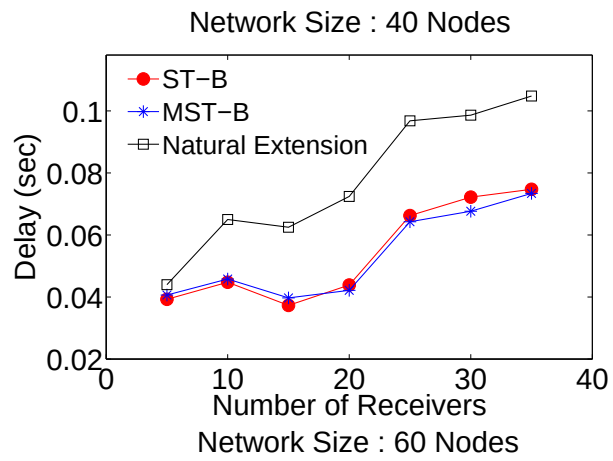
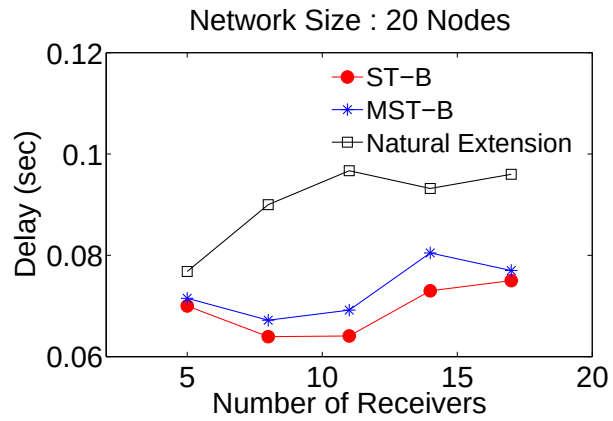


Figure 5.6. Delay Comparison

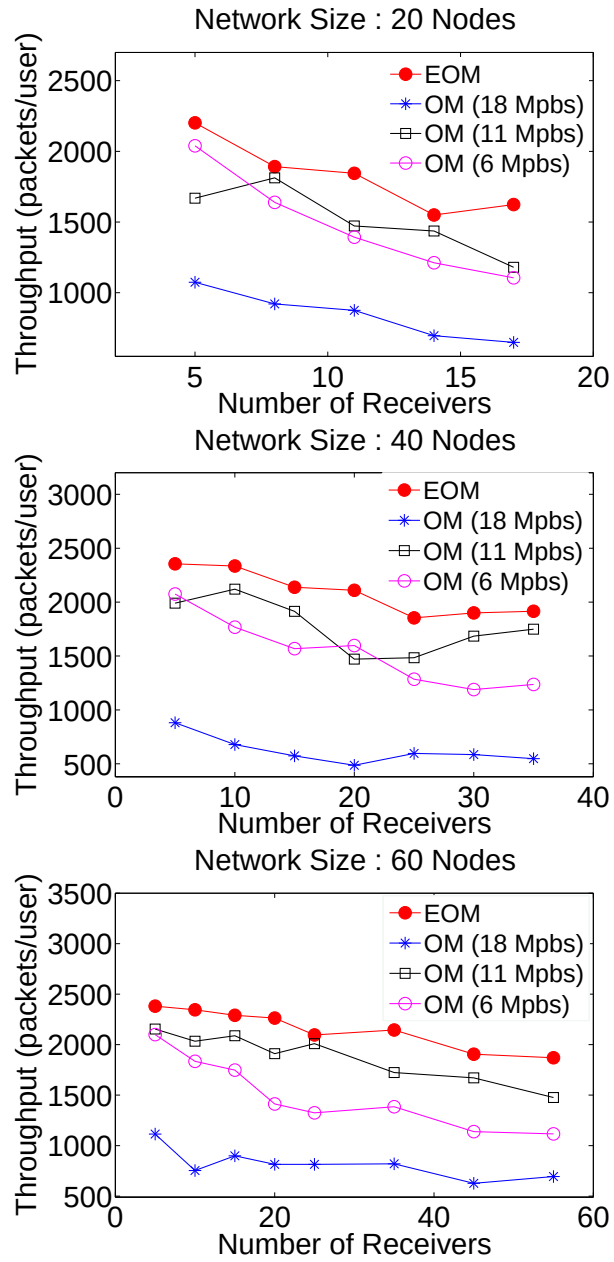


Figure 5.7. Impact of Multiple Rates

efficient backbone and operating on multiple rates. Another phenomenon we need to mention is that, the OM algorithm with 18 Mbps shows very low throughput. It indicates that 18 Mbps has a short transmission range that greatly decreases the spatial opportunities. The lack of spatial diversity overwhelms the benefit of its high transmission rates and dominates the results.

5.6 Summary

In this chapter, we apply opportunistic routing to multicast applications in WMNs. We propose the opportunistic multicast protocol in single-rate WMNs, where a tree backbone is build to help the packets self route to destinations by OR. We show that an efficient tree backbone should minimize transmissions to increase the throughput. We prove that computing a tree backbone with the minimum transmissions is NP-hard, and devise two heuristic algorithms for it. We also investigate the inherent rate-distance tradeoff in opportunistic multicast. We propose the Euclidean opportunistic multicast protocol in multi-rate WMNs. We build a Euclidean tree backbone to minimize the number of transmissions, and present an effective rate selection scheme. Simulations show that our opportunistic multicast can achieve higher throughput and shorter delay than the natural multicast extension and the traditional multicast.

CHAPTER 6

Conclusion and Future Work

In this chapter, we conclude this dissertation and discuss some possible future research work.

6.1 Conclusion

With the advances of wireless technology, wireless mesh networks draw much attention from researchers. They have a variety of applications. For instance, wireless mesh networks have become popular for supporting last-mile broadband access. Multicast communication that provides a fundamental service for WMNs is a challenging problem, because throughput maximization conflicting with scarce bandwidth has the paramount priority. In previous discussion, we have demonstrated that the efficient multicast in WMNs should consider the following facts:

1. Maximizing throughput is the major concern for multicast applications in WMNs, because WMNs have a demanding network performance requirement.

2. To allow simultaneous nearby transmissions, utilizing multi-channel is able to further improve the throughput.
3. Building an efficient multicast structure can reduce interference and save network bandwidth.
4. Network topology is fixed, and energy efficient is not a concern.

In this dissertation, we discuss several research topics related to multicast communication in WMNs. First, we investigate the multicast algorithms for multi-channel WMNs where the throughput and the delay have paramount priorities. Traditional multicast algorithms assume that each node has one channel and one interface, but it is not the same case for WMNs. The mesh node may be equipped with multiple channels and multiple interfaces. Thus, unlike previous research work that is based on the single-channel multicast, two multicast algorithms (LCA and MCM) are proposed to exploit multiple channels and multiple interfaces in Chapter 3. An effective multicast structure is constructed to minimize the number of relay nodes and hop count distances. We also design a dedicated channel assignment to further reduce the interference. The performance evaluation shows that our algorithms outperform the single-channel multicast in terms of throughput and delay.

Although LCA and MCM are able to significantly increase the multicast throughput in multi-channel WMNs, they do not consider the link-heterogeneity phenomena in WMNs. Thus, secondly, we extend our multicast work to link-heterogeneous wireless mesh networks in Chapter 4. As we know, increasing throughput in WMNs can be achieved by reducing the interference. One common approach to decrease interference

is to minimize the number of transmission. So, we should minimize transmissions to increase throughput. In link-homogeneous WMNs, minimizing transmissions is equivalent to minimizing the number of relay nodes, but it is not true for link-heterogeneous WMNs. Thus, we model the link-heterogeneous WMNs as a weighted directed graph, where a link's weight represents its quality, that is, the link loss ratio. We define a new graph theory problem, HW-SCDS, on an edge-weighted directed graph. An optimal multicast structure in this edge-weighted directed graph is an MHW-SCDS. We prove that finding an MHW-SCDS is NP-hard. A greedy algorithm MLRM for this problem is presented. Simulations show that our algorithm outperforms the current best WMN multicast algorithm by both increasing throughput and reducing delay.

Thirdly, our previous research work is based on traditional multicast routing paradigm. That is, we designate one or several next-hop destinations for each forwarder. This routing paradigm does not fully exploit the spatial reuse of wireless communications. At the same time, a new routing paradigm called opportunistic routing has been proposed in recent years. In OR, any node that overhears the packet transmission is encouraged to forward the packet if it is closer to the destination. Thus, more spatial reuse is explored to increase the throughput. Based on this consideration, we propose a new opportunistic multicast protocol to improve multicast throughput for WMNs in Chapter 5. Our opportunistic multicast builds upon opportunistic routing strategy. The key concept is a tree backbone. Our tree backbone protocol represents a tradeoff between traditional structured multicast protocols where a complete multicast tree is constructed and unstructured protocols where multicast is treated as a collection of unicasts. For single-rate WMNs, we show

that constructing an efficient tree backbone that minimizes the number of transmissions is NP-hard, and we devise two effective heuristic algorithms. For multi-rate WMNs, we investigate the inherent rate-distance tradeoff and propose a Euclidean opportunistic multicast protocol by devising a Euclidean tree backbone as well as an efficient rate selection scheme. In our simulations, our tree backbone multicast protocols outperform both the completely structured traditional multicast protocols and the completely unstructured unicast-based protocols augmented with OR in both throughput and delay.

6.2 Future Work

In this section, we discuss some future work. First, we introduce the future work on fair rate allocation for opportunistic routing in wireless mesh networks. Second, we propose to devise efficient multicast algorithms for cognitive wireless mesh networks.

6.2.1 Fair Rate Allocation for Opportunistic Routing in Wireless Mesh Networks

Most traditional routing protocols for wireless networks follow the concept of routing in wired networks by discovering the least cost or highest throughput paths to reach the destinations. However, when the link quality is poor, the delivery ratio to the next-hop destination is low. In contrast, opportunistic routing (OR) [14] encourages any node overhearing the transmission and closer to the destination to participate in packet forwarding. It takes advantage of the spatial characteristic of wireless

communication and achieves higher throughput than traditional routing [14, 16].

Some variants of opportunistic routing are proposed to improve throughput under different situations. The routing strategies in [30, 31, 16, 57] aim to combine network coding with opportunistic routing in a natural fashion, so that they can achieve the cooperative spatial diversity. The authors in [58, 59] extend OR to enable devices with only one interface to operate on multiple channels, which reduces interferences. The forwarding candidate set and relay priority are defined based on the nodes' geographic information [60, 61, 62]. For example, GeRaf [60] defines sets of regions with nodes closest to the destination with highest priority. In the ad hoc context, there are relevant works to consider, which takes advantage of diversity offered by multiple users [63].

All the above research on opportunistic routing is to improve the system throughput by exploiting more spatial opportunities in wireless networks. However, opportunistic routing also brings the unfairness problem to the wireless network. This is because opportunistic routing employs the CSMA protocols for MAC layer transmission. Different nodes have different observable views of the system state based on the random nature of CSMA/CA MAC protocols and the multihop network topology [96]. Unfortunately, unfairness problem may degrade the network performance. So, in wireless networks, fair allocation of bandwidth among different nodes is one critical problem that affects the serviceability of the entire network system [97, 98, 99, 100]. When the network resources cannot satisfy the demand, they should be divided among the clients of the network. In the future, one of our research objective is to provide proportional fairness among the sessions in opportunistic routing.

There have been some research work addressing the fairness issues for traditional routing by fairly assigning the rates to each session [101, 102], but they cannot be directly applied to opportunistic routing. This is because those research work about fair rate allocation relies on the flow contention graph that is based on the link contention relationship. In traditional routing, all the packets travel along the predefined routing paths. The interference relationship among links is fixed when the routing paths are determined. The flow contention graph is easy to build for traditional routing. However, it is unpredictable which nodes would forward the specified packet in OR. That is, there is no designated routing path for each transmitted packet. Thus, the flow contention graph cannot be built with the traditional way.

To address the unfairness problem in opportunistic routing, one research work has been proposed. In [103], the authors use adaptive rate control to determine an appropriate sending rate based on the current network condition. The destination sends an end-to-end ACK to the source node after receiving a fixed number of packets or when a timer expires. The sender adjusts its sending rate according to the ACKs it receives in the previous time interval. This strategy prevents the starvation of flows and provides better opportunities for equal medium access across different flows. However, this strategy does not guarantee that the proportional fairness is maximized in the network. This is because the source of each session only gets the feedback from its destination. It does not get to know the status of the other sessions. That is, the decision of rate control is within the session without sensing the global network condition. Thus, the rate control on each source node may not guarantee that the whole network can achieve the maximum proportional fairness. In [104], the authors

discuss whether there exists a tradeoff relation between fairness and efficiency in opportunistic forwarding. They use an offline approach to construct forwarding paths while ensuring both fairness and efficiency. The fairness in this paper refers to a relative equality in the distribution of resource usage among neighboring nodes in the network. However, in our future work, we consider the fairness among all the sessions in the network. Thus, our future research objective is different with [104].

In the future, we want to study the fairness problem in opportunistic routing. From the above discussion, the research issues for fairness in opportunistic routing are the following: (1) model the contention relationship among transmissions based on opportunistic routing paradigm and CSMA/CA random nature, and (2) provide maximum proportional fairness to all the sessions in the network by calculating the appropriate rate at each sender. We might consider one prospective solution: at first, we can build a transmission contention graph to model the contention relationship among different opportunistic transmission in the network. A transmission consists of the following attributes: (1) a forwarder, (2) a list of recipients, (3) the session carried by this transmission, and (4) the weight to describe the possibility of forwarding this packet. This transmission contention graph can capture the random nature of opportunistic routing. With the transmission contention graph, some optimization based method may be applied to design a framework. This framework uses utility function to control the rate at each sender. In this way, our prospective solution could be used to maximize the proportional fairness by fairly assigning the rates.

6.2.2 Multicast Application in Cognitive Wireless Mesh Networks

The cognitive radio has become a promising technology that is able to use dynamic spectrum access for the next generation wireless networks [105, 106, 107]. Because access to spectrum has been chronically regulated with static spectrum allocation policies, today's wireless networks operate within a fixed spectrum, such as cell phone network, Wi-Fi, Bluetooth, and so on. However, with the fast developing spectrum-based devices, the remaining available spectrum is being exhausted, which is known as the spectrum scarcity problem. At the same time, a considerable portion of the licensed spectrum is under-utilized with the variation of time and geographic locations. Thus, cognitive radio, which is able to sense the spectrum holes across the licensed bands and switch to any free frequency band, is proposed to solve this inefficient spectrum use problem.

Multicast is a key technology in all types of wireless multihop networks, including cognitive wireless mesh networks. There have been some research work addressing multicast application in cognitive radio networks (CRN). The authors in [108] propose an on-demand multicast routing and channel allocation algorithm that takes channel heterogeneity and switching latency into consideration. It aims at reducing the end-to-end delay by using a dynamic programming approach. A low complexity approximation algorithm with bounded performance guarantee for constructing the minimum-energy multicast tree is proposed in [109], which transforms the multicast problem into a directed Steiner tree problem. In order to minimize the required

network-wide resource to support a set of multicast sessions with a given bit rate requirement, some research work [110, 111, 112] formulates this problem via cross-layer approaches by taking consideration of scheduling and routing jointly. However, none of the multicast work on cognitive radio networks consider QoS requirement. One of our future work is to consider the QoS support for multicast in cognitive radio networks. That is, we will investigate on joint routing and spectrum allocation to support as many as possible QoS multicast sessions in cognitive WMNs.

QoS requirement is enforced in some multicast applications. For example, video in demand needs to satisfy a given throughput requirement. QoS routing is a major challenge in wireless networks because of the interference among adjacent transmissions. If adjacent links potentially interfere with each other, it is unlikely to predict the throughput on them due to the random nature of CSMA/CA MAC mechanism. QoS connection request usually comes with a bandwidth requirement. To guarantee the QoS routing, one common method is to find interference-free paths for all the QoS connections, and assign enough bandwidth on each routing path.

As we know, cognitive radio uses Orthogonal Frequency Division Multiplexing (OFDM) to achieve flexible spectrum assignment [113, 114]. The spectrum can be divided into subbands that are modulated with orthogonal subcarriers, so that the data can be divided into parallel data streams. The terms “channel” and “subcarrier” are interchangeably used in this chapter. A new technology called Discontiguous Orthogonal Frequency Division Multiplexing (DOFDM) [115] has also come into being. A radio is able to access multiple non-contiguous spectrum fragments at the same time with the DOFDM technology. It is well known that the bandwidth capacity of a

spectrum is linear in its frequency width [116]. Thus, to satisfy the bandwidth requirement for each link on the multicast path, each link on the path must be allocated with enough frequency bandwidth (or equally enough number of subcarriers).

From the above description, there are three major research issues for QoS-aware multicast communication in cognitive radio networks. (i) How to compute the interference-free multicast path for each multicast session. Here, we must ensure that there is no intra-flow or inter-flow interference among links. (ii) We must minimize the channel consumption when we allocate spectrum to each multicast session. The channel consumption of selecting channel c is defined as the number of nodes in this neighborhood if they have available channel c . This is because when we assign a channel c to a link e , channel c cannot be used in e 's neighborhood. If there is less channel consumption, the more channels are reserved for future multicast sessions. (iii) Spectrum allocation needs to consider the spectrum mobility issue. To address the above research issues, one prospective way is to model the network as an weighted graph. In this graph, both edges and nodes have weights. The edge weight represents the common channel set between the adjacent nodes. And the node weight indicates the available channel set the node has. In order to minimize the channel consumption, we are going to define the concept of a Multicast Structure with Minimum Structure Resource Cost (MS-MSRC) in a graph. We will try to demonstrate that minimizing channel consumption is equivalent to finding an MS-MSRC in the weighted graph. The MS-MSRC not only shows the multicast path, but also indicates the spectrum allocation. In addition, one important action of our approach is to let cognitive radios work on channels with lower probability of being reclaimed by primary users, so that

the chance of spectrum mobility would be reduced.

Based on the above discussion, it is a potential research topic to study the joint routing and spectrum allocation scheme to optimize the multicast performance in the cognitive WMNs.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] M.K. Marina and S.R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of the International Conference for Network Protocols 01*, 2001.
- [2] <http://www.seattlewireless.net>.
- [3] IAN F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Radio Communications*, September 2005.
- [4] IAN F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *IEEE Computer Networks*, 2005.
- [5] A. Dhananjay, Jingyang Li, L. Subramanian, and H. Zhang. Design and implementation of high performance dual-radio mesh networks. In *Sigcomm 09*, 2009.
- [6] B. Radunovic, D. Gkantsidis, C. Gunawardena, and P. Key. Horizon: Balancing tcp over multiple paths in wireless mesh network. In *Mobicom 08*, 2008.
- [7] <http://pdos.csail.mit.edu/roofnet/doku.php>.
- [8] <http://www.research.microsoft.com/mesh/>.
- [9] <http://www.radiantnetworks.com/>.
- [10] Community wireless networking in western australia
<http://www.wafreenet.org/>.
- [11] Jungmin So and Nitin Vaidya. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc '04*, 2004.
- [12] Krishna Ramachandran, Elizabeth M. Belding, Kevin Almeroth, and Milind Buddhiko. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Infocom 06*, 2006.
- [13] Jian Tang, Guoliang Xue, and Weiyi Zhang. Maximum throughput and fair bandwidth allocation in multi-channel wireless mesh networks. In *INFOCOM 06*, 2006.
- [14] Sanjit Biswas and Robert Morris. Exor: Opportunistic multi-hop routing for wireless networks. In *SIGCOMM 05*, 2005.

- [15] R.U. Nabar, H. Bolcskei, and F.W. Kneubuhler. Fading relay channels: Performance limits and space-time signal design. *IEEE Journal on Selected Areas in Communications*, June 2004.
- [16] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM 07*, 2007.
- [17] H Zhai and Y. Fang. Impact of routing metrics on path capacity in multirate and multihop wireless ad hoc networks. In *ICNP 06*, 2006.
- [18] Kai Zeng, Wenjing Lou, and Hongqiang Zhai. On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks. In *Infocom 08*, 2008.
- [19] H Zhai and Y. Fang. Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks. In *Infocom 06*, 2006.
- [20] C. Wu, Y. Tay, and C. Toh. Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) functional specification. In *Internet Draft*, Nov. 1998.
- [21] E. Royer and C. Perkins. Multicast operation of the ad-hoc on demand distance vector routing protocol. In *MOBICOM 99*, 1999.
- [22] Jorjeta Jetcheva and David B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *MobiHoc 01*, 2001.
- [23] S. Lee, M. Gerla, and C. Chiang. On demand multicast routing protocol. In *IEEE WCNC 99*, pages 1313–1317, Aug. 1999.
- [24] J. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17(8), Aug. 1999.
- [25] S. Das, B. Manoj, and C. Murthy. A dynamic core based multicast routing protocol. In *MobiHoc 02*. 2002.
- [26] Lusheng Ji and M. Scott Corson. Differential destination multicast-a manet multicast routing protocol for small groups. In *INFOCOM 01*, 2001.
- [27] K. Chen and K. Nahrstedt. Effective location-guided overlay multicast in mobile ad hoc networks. *International Journal of Wireless and Mobile Computing(IJWMC)*, 3, 2005.
- [28] M. Mauve, H. Fuler, J. Widmer, and T. Lang. Position-based multicast routing for mobile ad hoc networks. In *MobiHoc 03*, June 2003.

- [29] A. Kumar and K. Hegde. Multicasting in wireless mesh networks: Challenges and opportunities. In *ICIME 09*, 2004.
- [30] X. Fang, D. Yang, and G. Xue. Consort: Node-constrained opportunistic routing in wireless mesh networks. In *INFOCOM 11*, 2011.
- [31] Y. Lin, B. Liang, and B. Li. Slideor: Online opportunistic network coding in wireless mesh networks. In *INFOCOM 10*, 2010.
- [32] Y. Lin, , B. Li, and B. Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *ICNP 08*, 2008.
- [33] Sachin Katti, Dina Databi, Hari Balakrishnan, and Muriel Medard. Symbol-level network coding for wireless mesh networks. In *SIGCOMM 08*, 2008.
- [34] J. Yuan, Z. Li, W. Yu, and B. Li. A cross-layer optimization framework for multihop multicast in wireless mesh networks. *IEEE Journal of Selected Areas in Communications*, 24:2092 – 2103, Nov. 2006.
- [35] Bo Wang, Matt Mutka, and Eric Torng. Optimization based rate allocation and scheduling in tdma based wireless mesh networks. In *ICNP 08*, 2008.
- [36] D. Koutsonikolas, Y. Hu, and C. Wang. High-throughput, reliable multicast without crying babies in wireless mesh networks. In *INFOCOM 09*, 2009.
- [37] Sabyasachi Roy, Dimitrios Koutsonikolas, Saumitra Das, and Y. Charlie Hu. Highthroughput multicast routing metrics in wireless mesh networks. In *ICDCS 06*, 2006.
- [38] I. Chakeres, C. Koundinya, and P. Aggarwal. Fast, efficient, and robust multicast in wireless mesh networks. In *the 5th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, 2008.
- [39] Guokai Zeng, Bo Wang, Yong Ding, Li Xiao, and Matt Mutka. Efficient multicast algorithms for multi-channel wireless mesh networks. *IEEE Transactions on Parallel and Distributed Systems*, 2010.
- [40] S. Lim, C. Kim, Y. Ko, and N. Vaidya. An efficient multicasting for multi-channel multi-interface wireless mesh networks. In *MILCOM 09*, 2009.
- [41] H. Cheng and S. Yang. Joint multicast routing and channel assignment in multiradio multichannel wireless mesh networks using simulated annealing. In *LNCS 08*, 2008.
- [42] U. Nguyen. On multicast routing in wireless mesh networks. *Computer Communications*, 31:1385 – 1399, May 2008.

- [43] P. Bahl, R. Chandra, and J. Dunagan. Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *ACM MobiCom 2004*, 2004.
- [44] Y. Liu and E. Knightly. Opportunistic fair scheduling over multiple wireless channels. In *IEEE INFOCOM 03*, 2003.
- [45] A. Tzamaloukas and J. J. Garcia-Luna-Aceves. A receiver-initiated collision-avoidance protocol for multi-channel networks. In *IEEE INFOCOM 2001*, 2001.
- [46] W. Tam and Y. Tseng. Joint multi-channel link layer and multi-path routing design for wireless mesh networks. In *INFOCOM 07*, 2007.
- [47] A. Subramaniam, H. Gupta, and S. Das. Minimum-interference channel assignment in multi-radio wireless mesh networks. In *SECON 07*, 2007.
- [48] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom 05*, 2005.
- [49] Bhaskaran Raman. Channel allocation in 802.11-based mesh networks. In *INFOCOM 06*, 2006.
- [50] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian. Practical, distributed channel assignment and routing in dual-radio mesh networks. In *SIGCOMM 09*, 2009.
- [51] S. Pediaditaki, P. Arrieta, and M. Marina. A learning-based approach for distributed multi-radio channel allocation in wireless mesh networks. In *ICNP 09*, 2009.
- [52] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks (mc2r). *ACM Mobile Computing and Communications Review*, 8(2):50–65, 2004.
- [53] Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *IEEE INFOCOM 2005*, 2005.
- [54] A. Hamed Mohsenian Rad and Vincent W.S. Wong. Joint channel allocation, interface assignment and mac design for multi-channel wireless mesh networks. In *Infocom 2007*, 2007.
- [55] S. Bodas, S. Shakkottai, and R. Srikant. Scheduling for small delay in multi-rate multi-channel wireless networks. In *INFOCOM 11*, 2011.

- [56] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom 04*, 2004.
- [57] Sachin Katti, Dina Katabi, Hari Balakrishnan, and Muriel Medard. Symbol-level network coding for wireless mesh networks. In *SIGCOMM 08*, 2008.
- [58] K. Zeng, Z. Yang, and W. Lou. Opportunistic routing in multi-radio multi-channel multi-hop wireless networks. In *INFOCOM 10*, 2010.
- [59] Anatolij Zubow, Mathias Kurth, and Jens-Peter Redlich. Multi-channel opportunistic routing. In *IEEE European Wireless Conference*, 2007.
- [60] M. Zorzi and R.R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4), 2003.
- [61] R.C Shah, A. Bonivento, D. Petrovic, E. Lin, J. van Greunen, and J. Rabaey. Joint optimization of a protocol stack for sensor networks. In *IEEE MILCOM 04*, 2004.
- [62] K. Zeng, W. Lou, W. Yang, and D.R. Brown. On throughput efficiency of geographic opportunistic routing in multihop wireless networks. In *QShine'07*, 2007.
- [63] Cedric Westphal. Opportunistic routing in dynamic ad hoc networks: the oprah protocol. In *MASS 06*, 2006.
- [64] Niranjana, Sugam Pandey, and Aura Ganz. Design and evaluation of multichannel multirate wireless networks. *Mobile Networks and Applications*, 11:697–709, 2006.
- [65] X. Zhao, J. Guo, C. Chou, A. Misra, and S. Jha. A high-throughput routing metric for reliable multicast in multi-rate wireless mesh networks. In *INFOCOM 11*, 2011.
- [66] R. Kumar, S. Tati, F. Mello, S. Krishnamurthy, and T. Porta. Network coding aware rate selection in multi-rate ieee 802.11. In *ICNP 10*, 2010.
- [67] T. Kim, S. Vural, I. Broustis, D. Syrivelis, S. Krishnamurthy, and T. Porta. A framework for joint network coding and transmission rate control in wireless networks. In *INFOCOM 10*, 2010.
- [68] S Guha and S Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4), 1998.

- [69] Jeremy Blum, Min Ding, Andrew Thaeler, and Xiuzhen Cheng. Connected dominating set in sensor networks and manets. *Handbook of Combinatorial Optimization*, pages 329–369, 2004.
- [70] Yafeng Wu, Yinlong Xu, Guoliang Chen, and Kun Wang. On the construction of virtual multicast backbone for wireless ad hoc networks. In *MASS 05*, 2005.
- [71] Khaled M Alzoubi, Peng-Jun Wan, and Ophir Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.
- [72] Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. In *SODA 08*, 2008.
- [73] Li-Der Chou, Huang-Ju Hsieh, and Jui-Ming Chen. Multicast with qos support in heterogeneous wireless networks. In *EUC 04*, 2004.
- [74] Ning Li and Jennifer C. Hou. Topology control in heterogeneous wireless networks: Problems and solutions. In *Infocom 04*, 2004.
- [75] T. Simunic, W. Qadeer, and De G. Micheli. Managing heterogeneous wireless environments via hotspot servers. In *MMCN’05*, 2005.
- [76] IEEE 802.11 Standard.
- [77] Arunesh Mishra, Eric Rozner, Suman Banerjee, and William Arbaugh. Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage. In *ACM/USENIX Internet Measurement Conference*, 2005.
- [78] Arunesh Mishra, Vivek Shrivastava, and Suman Banerjee. Partially overlapped channels not considered harmful. In *SIGMetrics/Performance 06*, 2006.
- [79] P. Li, N. Scalabrino, Y. Fang, E. Gregori, and I. Chlamtac. Channel interference in ieee 802.11b systems. In *GLOBECOM 07*, 2007.
- [80] Y. Ding, Y. Huang, G. Zeng, and L. Xiao. Channel assignment with partially overlapping channels in wireless mesh networks. In *WICON 08*, 2008.
- [81] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction To Algorithms*. The MIT Press, 2001.
- [82] <http://www.isi.edu/nsnam/ns/index.html>.
- [83] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh networks. In *Sigcomm 04*, 2004.

- [84] S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multihop wireless routing. In *MOBICOM 03*, 2003.
- [85] Guokai Zeng, Bo Wang, Yong Ding, Li Xiao, and Matt Mutka. Multicast algorithms for multi-channel wireless mesh networks. In *ICNP 07*, 2007.
- [86] F. Hwang, D. Richards, and P. Winter. Steiner tree problem. *The Netherlands: North-Holland*, 1992.
- [87] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathmatica Japonica*, pages 573–577, 1998.
- [88] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.
- [89] M. Zachariasen and P. Winter. Concatenation-based greedy heuristics for the euclidean steiner problem. *Algorithmica*, 25:418–437, 1999.
- [90] M. Minoux. Efficient greedy heuristics for steiner tree problems using reoptimization and supermodularity. *INFOR*, 28(221-233), 1990.
- [91] J.E. Beasley. A heuristic for euclidean and rectilinear steiner problems. *EJOR*, 58(284-292), 1992.
- [92] Brad Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *MOBICOM 00*, 2000.
- [93] H. Frey. Scalable geographic routing algorithms for wireless ad hoc networks. *IEEE Network Magazine*, pages 18–22, 2004.
- [94] S. Wu and K. S. Candan. Gmp:distributed geographic multicast routing in wireless sensor networks. In *IEEE ICDCS 06*, 2006.
- [95] Cisco aironet 802.11a/b/g wireless lan client adapters (cb21ag and pi21ag) installation and configuration guide. Technical report, Cisco Systems, 2004.
- [96] V. Gambiroza and E. Knightly. Congestion control in csma-based networks with inconsistent channel state. In *WICON 06*, 2006.
- [97] H. Luo, J. Cheng, and S. Lu. Self-coordinating localized fair queueing in wireless ad-hoc networks. *IEEE Transactions on Mobile Computing*, 3:86–98, 2004.
- [98] J. Jaramillo. Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic. In *INFOCOM 10*, 2010.
- [99] L. Chen, S. Low, and J. Doyle. Joint congestion control and media access control design for wireless ad hoc networks. In *INFOCOM 05*, 2005.

- [100] B. Li. End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks. In *ICDCS 05*, 2005.
- [101] Z.Y. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *Infocom 04*, 2004.
- [102] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematic theory of network architectures. *the IEEE*, 95(1):255–312, 2007.
- [103] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks. In *IEEE Transactions on Mobile Computing*, 2009.
- [104] A. Mtibaa and K. Harras. Fog: Fairness in mobile opportunistic networking. In *IEEE SECON 11*, 2011.
- [105] I.F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, pages 2127–2159, 2006.
- [106] T. Chen, H. Zhang, G.M. Maggio, and I. Chlamtac. Topology management in cogmesh: A cluster-based cognitive radio mesh network. In *ICC 07*, 2007.
- [107] S. Jafar and S. Srinivasa. Capacity limits of cognitive radio with distributed and dynamic spectral activity. *IEEE J. Sel. Areas Commun.*, 25(3):529–537, 2007.
- [108] H. Almasaeid, T. Jawadwala, and A. Kamal. On-demand multicast routing in cognitive radio mesh networks. In *GLOBECOM 10*, 2010.
- [109] W. Ren, X. Xiao, and Q. Zhao. Minimum-energy multicast tree in cognitive radio networks. In *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2009.
- [110] C. Cao, Y. Shi, Y. Hou, H. Sherali, and H. Zhou. Multicast communications in multi-hop cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 29:784–793, 2011.
- [111] J. Jin, H. Xu, and B. Li. Multicast scheduling with cooperation and network coding in cognitive radio networks. In *INFOCOM 10*, 2010.
- [112] D. Hu, Y. Mao, S. and Hou, and J. Reed. Scalable video multicast in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 28:334–344, 2010.

- [113] Anibal Luis Intni. Orthogonal frequency division multiplexing for wireless networks. Technical report, UNIVERSITY OF CALIFORNIA SANTA BARBARA, 2000.
- [114] R. Rajbanshi. *Ofdm-based cognitive radio for dsa networks*. PhD thesis, The University of Kansas, 2007.
- [115] J. Poston and W. Horne. Discontiguous ofdm considerations for dynamic spectrum access in idle tv channels. In *DySPAN 05*, 2005.
- [116] Y. Yuan, P. Bahl, T. Moscibroda, and Y. Wu. Allocating dynamic time-spectrum blocks in cognitive radio networks. In *MOBIHOC*, 2007.