HIGH-PERFORMANCE MULTICAST IN MULTI-CHANNEL MULTI-RADIO WIRELESS MESH NETWORKS

HOANG LAN NGUYEN

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO
FEBRUARY 2012

Your file Votre référence
ISBN: 978-0-494-90341-4

Our file Notre référence
ISBN: 978-0-494-90341-4

HIGH-PERFORMANCE MULTICAST IN MULTI-CHANNEL MULTI-RADIO WIRELESS MESH NETWORKS

by **Hoang Lan Nguyen**

Examination Committee Members:

1. Professor Andrew Eckford (Dean's Representative)
2. Professor Muhammad Jaseemuddin (External Examiner)
3. Professor Regina Lee (Outside Member)
4. Professor Uyen Trang Nguyen (Supervisor)
5. Professor Jonathan Ostroff (Chair)
6. Professor Natalija Vlajic

Abstract

Wireless mesh networking (WMN) is an emerging technology that enables multi-hop wireless connectivity to areas where wiring or installing cables is difficult or expensive. Multicast is a form of communication that delivers information from a source to a group of destinations simultaneously in an efficient manner. In a single-channel WMN, all nodes share and communicate with each other via the same channel. In such a network, the throughput capacity of multicast degrades significantly as the network size increases. A critical factor that contributes to this rapid degradation is the co-channel interference in single-channel WMNs, worsened by the use of single, half-duplex radio per node. A node with a single half-duplex radio is restricted to access one channel at a time, and thus cannot transmit and receive simultaneously. One of the most effective approaches to achieve high throughput is to use systems with multiple channels and multiple radios (MCMR) per node. An MCMR node may transmit on one channel and receive on another at the same time using two different radios, and thus at least double the throughput. In this thesis,

we propose solutions to support high-performance multicast in MCMR WMNs, as follows.

1. We propose a novel channel assignment (CA) algorithm for multicast that minimizes interference among forwarding nodes, because existing CA algorithms for multicast suffer very low performance due to lack of interference-free solutions.

2. We propose routing algorithms that outperform traditional multicast routing schemes by taking into account the wireless broadcast advantage (WBA) and the underlying CA in order to minimize network bandwidth consumption. Traditional multicast routing algorithms such as shortest path tree and Steiner tree did not consider the WBA or the underlying CA in MCMR WMNs.

3. We develop analytical models for estimating the performance of network-coded multicast in MCMR WMNs, and validate the proposed models using realistic simulation-based scenarios. Network coding has been proven to be a promising technique for improving network throughput of WMNs. However, the performance of a multicast session in combination with network coding in the MCMR environment has not been studied prior to this research.

Acknowledgements

First and foremost, I would like to express my sincere thanks to my supervisor, *Prof. Uyen Trang Nguyen*. Without her support, guidance and encouragement, this thesis would not have been possible. She enlightened me on how to look research in its full depth and taught me how to produce and appreciate good scientific work that helps other researchers to build on it.

I would like to thank my entire committee, *Profs. Andrew Eckford, Muhammad Jaseemuddin, Regina Lee, Jonathan Ostroff* and *Natalija Vlajic*, whose insightful suggestions helped improve the readability and quality of this thesis. I am also grateful to the graduate program team for their excellent administrative supports, *Prof. Yves Lesperance, Prof. Richard Wildes* and *Ouma Jaipaul-Gill.*

I would also like to acknowledge many postgraduate fellows and friends who assisted me in so many ways during my time at York: Nariman Farsad, Hai Feng Huang, Celia Li, Marvin Pinto, Michael Portnoy, George Spanogiannopoulos, Dusan Stevanovic, David Xia, Xing Xiong, Frank Xu, Jin Xu, and Yi Zheng.

Last and most heartfeltly, I am forever indebted to my family, my dad Dr. Nhu Do Nguyen, my mom Tuyet Giang Dao, my brother Hoang Long Nguyen, my sister in-law Ngoc Linh Nguyen, my nephew Hoang Tuan Nguyen, and last but not least my wife Huong Ly Hoang for their unconditional love, companionship and patience throughout the years of my study. To them I dedicate this thesis.

Table of Contents

List of Tables

List of Figures

Publications from the Thesis

Refereed Journal Article

1. Hoang Lan Nguyen and Uyen Trang Nguyen, Channel Assignment for Multicast in Multi-Channel Multi-Radio Wireless Mesh Networks, Wiley Journal of Wireless Communications and Mobile Computing, Special Issue on Next Generation Wireless Communications and Mobile Computing/Networking Technologies, Vol. 9, No. 4, April 2009, pp. 557-571 (selected by IWCMC'08 TPC and extended for journal publication).

Refereed Conference Proceedings

2. Hoang Lan Nguyen and Uyen Trang Nguyen, Performance Evaluation of Network-Coded Multicast in Multi-Channel Multi-Radio Wireless Mesh Networks, IEEE International Conference on Communications (ICC 2012), Ot-

tawa, Canada, June 2012.

3. Hoang Lan Nguyen and Uyen Trang Nguyen, Performance Modeling of Multicast with Network Coding in Multi-Channel Multi-Radio Wireless Mesh Networks, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2012), San Francisco, USA, June 2012.

4. Hoang Lan Nguyen and Uyen Trang Nguyen, High-Performance Multicast Routing in Multi-Channel Multi-Radio Wireless Mesh Networks, IEEE Wireless Communications and Networking Conference (WCNC 2011), Cancun, Mexico, March 2011.

5. Hoang Lan Nguyen and Uyen Trang Nguyen, High-Performance Multicast Routing in Multi-Channel Multi-Radio Wireless Mesh Networks, IEEE Global Communications Conference (GLOBECOM 2010), Miami, USA, December 2010.

6. Hoang Lan Nguyen and Uyen Trang Nguyen, Bandwidth Efficient Multicast Routing in Multi-Channel Multi-Radio Wireless Mesh Networks, IEEE International Workshop on Scalable Ad Hoc and Sensor Networks (SASN 2009), Saint Petersburg, Russia, October 2009.

7. Hoang Lan Nguyen and Uyen Trang Nguyen, Minimum Interference Channel Assignment for Multicast in Multi-Radio Wireless Mesh Networks, IEEE

International Wireless Communications and Mobile Computing Conference (IWCMC 2008), Crete Island, Greece, August 2008.

Abbreviations

CA	Channel Assignment
HCP	Hidden Channel Problem
M4	Minimum-interference Multi-channel Multi-radio Multicast
MCM	Multi-Channel Multicast
MCMNT	Multi-Channel Minimum Number of Transmissions
MCMR	Multi-Channel Multi-Radio
MFT	Minimum number of Forwarders Tree
NetCoM	Network-Coded Multicast
PDR	Packet Delivery Ratio
ReM	Regular Multicast
SPT	Shortest Path Tree
ST	Steiner Tree
WBA	Wireless Broadcast Advantage
WMN	Wireless Mesh Network

Chapter 1

Introduction

Wireless mesh networking is an emerging technology that supports many important applications such as Internet access provisioning in rural areas, ad hoc networking for emergency and disaster recovery, security surveillance, and information services in public transportation systems. The technology enables networking capability where wiring or installing cables is difficult or expensive. In a wireless mesh network (WMN), mesh routers provide multi-hop wireless connectivity from a host to either other hosts in the same network or in the Internet (Figure 1.1). The mesh routers are often static and form a wireless mesh backbone. Our work in this thesis focuses on this mesh backbone, and we will use the terms "routers" and "nodes" interchangeably.

Multicast is a form of communication that delivers information from a source to a group of destinations simultaneously in an efficient manner. Important appli-

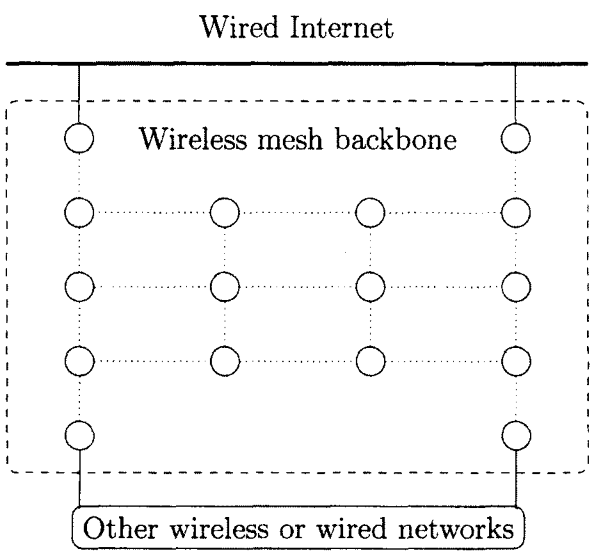


Figure 1.1: A sample wireless mesh network.

cations of multicast include distribution of financial data, billing records, software, and newspapers; audio/video conferencing; distance education; IP television; and distributed interactive games. Research on multicast in WMNs has considered mostly networks with a single channel, i.e., all nodes in the network share and communicate with each other via one single channel. The theoretical upper limit of per node throughput capacity in such networks is limited by $O(\frac{1}{\sqrt{n}})$, where n is the number of nodes in the network [1]. The theoretical *achievable* throughput is even lower, estimated as $\theta(\frac{1}{\sqrt{n \log n}})$ in a random ad hoc network with ideal global scheduling and routing [1]. It has also been shown through experiments that on a string topology using contention-based medium access control such as IEEE 802.11

[2], the throughput degrades approximately to $\frac{1}{n}$ of the raw channel bandwidth [3]. For multicast, the theoretical aggregate throughput is estimated as $O(\frac{1}{\sqrt{n^{\epsilon}\log n}})$, where $0 \leq \epsilon \leq 1$ [4]. The above results indicate that the throughput capacity of a single-channel WMN degrades significantly as the network size increases.

A critical factor that contributes to such rapid degradation is the co-channel interference in single-channel WMNs, worsened by the use of single, half-duplex radio per node. A node with a single half-duplex radio is restricted to access one channel at a time, and thus cannot transmit and receive simultaneously.

One of the most effective approaches to achieve higher throughput is to use systems with multiple channels and multiple radios (MCMR) per node. The network throughput in these MCMR systems can be increased multiplicatively at the cost of additional radio equipment. The tremendous popularity of wireless networking in recent years has led to the commoditization of wireless radios whose prices have fallen dramatically thanks to technology advances and mass production. Therefore, the idea of multi-radio multi-channel wireless networking is very promising, allowing us to use two or more radios on the same device [5, 6].

Figure 1.2 illustrates the MCMR model. The network has n channels, which may either overlap, such that a channel partially shares its frequency spectrum with the adjacent channels, or may be completely separated (non-overlapping or orthogonal). Orthogonal channels do not interfere with each other. For example, IEEE

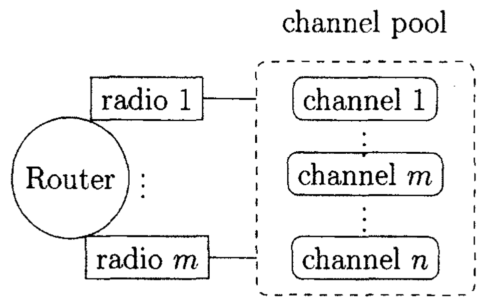
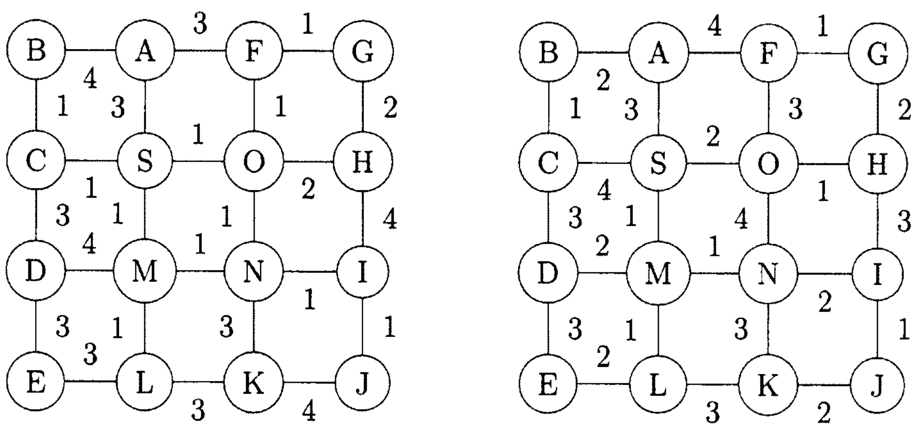


Figure 1.2: The Multi-Radio Multi-Channel (MCMR) model.

802.11b and 802.11g networks have 11 channels, numbered from 1 to 11, in which orthogonal channels are separated by at least four other channels, e.g., channels 2 and 7. A node in an MCMR network may have m radios (wireless interfaces). Typically, $1 < m < n$ (e.g., $m = 3$, $n = 11$). As a result, an MCMR node may transmit on one channel and receive on another at the same time using two different radios. On average, an MCMR wireless network at least doubles the throughput, since each node is now in full-duplex mode, being able to transmit and receive simultaneously. MCMR networks, in return, require efficient *channel assignment* (CA) and *routing* algorithms that can take advantage of multiple channels and multiple radios.

1.1 Motivations and Contributions of the Thesis

The task of CA is to decide which channel a node should use for data transmission in order to minimize interference. In early CA schemes, there is usually a constraint



(a) Network using $c \leq r$ approach

(b) Network using $c > r$ approach

Figure 1.3: Example networks using the two CA approaches. There is a total of four channels {1, 2, 3, 4} and each node has $r = 2$ radios. Each link is labeled with the assigned channel.

that the number channels c assigned to a node is less than or equal to the number of radios r the node possesses [7, 8, 9, 10, 11, 12, 13, 14]. Such a CA scheme is first applied to the entire network; then routing algorithms can be applied next to find optimal paths between a source and destination [7, 8, 9] or optimal multicast routing trees. Figure 1.3(a) shows an example network using the $c \leq r$ CA approach. The advantages of this CA approach are simple implementation, and no channel switching required because $c \leq r$.

Several other CA schemes do not impose the above constraint [15, 16, 17]. That is, a node may have more assigned channels than the number of radios the node possesses, i.e., $c > r$. Figure 1.3(b) shows an example network using the $c > r$ CA approach. The fact that $c > r$ leads to a need for channel switching, as illustrated by the following example.

Consider a link (A, B) in an MCMR network with $r = 2$ and $c = 4$, as shown in Figure 1.4. Assume that each node is equipped with two radios r_1 and r_2, and there are currently two unicast flows u_1 and u_2 between A and B that are assigned channels 1 and 2, respectively. Suppose that we use radio r_1 for flow u_1 on channel 1, and radio r_2 for flow u_2 on channel 2. Two multicast flow m_1 and m_2 then join the network. Suppose that link (A, B) is included in both multicast trees rooted at the two sources. It may happen that channel 3 is chosen as the best channel on link (A, B) for multicast flow m_1, and channel 4 is assigned to multicast flow m_2 on link

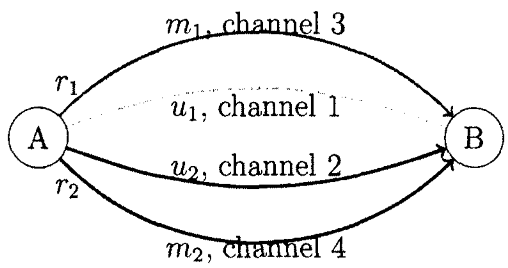


Figure 1.4: Channel switching example

(A, B). Node A now has to switch two radios among four channels (or more, if new flows later join the network). There are several ways to divide the radios among the flows. However, at least one radio has to switch among multiple flows, given that $r = 2$ and $c = 4$. Assume that radio r_1 serves flows u_1 and m_1 while radio r_2 serves flows u_2 and m_2. Before each transmission using radio r_1 (r_2), A has to tell B the channel on which B has to listen, either channel 1 or 3 (either channel 2 or 4). B then switches the radio to the specified channel waiting for the data packet.

The $c > r$ CA approach may be perceived as offering higher performance than the $c \leq r$ approach, because the former allows more channels to be used at a node than the latter, potentially lowering the amount of interference around the node. On the other hand, the $c > r$ CA approach requires efficient channel switching schemes and incurs channel switching overheads, which is not negligible [15, 18]. To the best our knowledge, there has not been any research that compares the network throughput offered by these two approaches. There has not been any consensus in the WMN research community either regarding the question of which approach is more favorable. Therefore, we consider both approaches in our work. Specifically, we propose the following solutions to support high-performance multicast in MCMR WMNs:

1. a novel CA algorithm for multicast named Minimum-interference Multi-channel Multi-radio Multicast (M4) using the $c > r$ approach;
2. multicast routing algorithms for MCMR WMNs that are based on the $c \leq r$ CA approach and minimize the total bandwidth consumption consumed by multicast trees;
3. analytical models and performance evaluations of multicast combined with network coding that enable us to understand how network coding can further enhance the performance of multicast in MCMR WMNs.

We now discuss the motivations and contributions of our work in detail.

1.1.1 Minimum-interference Multi-channel Multi-radio Multicast

Existing CA algorithms for multicast must rely on a metric called *interference factor* to determine how much interference a channel causes to another [19, 20, 21, 22, 23, 24]. A limitation of the interference factor is that it must be measured

Algorithms based on the $c > r$ CA approach	
Existing algorithms [19, 20, 21, 22, 23, 24]	**The M4 algorithm**
Inconvenient use of interference factor	Not rely on interference factor
Suffer from the hidden channel problem	No hidden channel problem
More interference in CA results	Less interference in CA results

Table 1.1: Advantages of the M4 algorithm.

before the CA can be applied, and may become inaccurate if the network topology and/or conditions later change. In addition, when computing the interference around a node v, previous works [19, 20, 21, 22, 23, 24] considered only the interference caused by directly connected (one-hop) neighbors of v, while neglecting the interference from nodes that are two or more hops away, and thus suffering low performance.

We propose a novel CA algorithm for multicast named Minimum-interference Multi-channel Multi-radio Multicast (M4) using the $c > r$ approach. The proposed M4 algorithm eliminates the inconvenient use of the interference factor, and still allows nodes to operate with minimum interference. Table 1.1 lists the advantages of the M4 algorithm over existing algorithms. Our experimental results show that M4 outperforms existing CA algorithms for multicast with respect to packet delivery ratio, end-to-end delay and throughput.

1.1.2 Bandwidth-Efficient Multicast Routing Algorithms

Given a $c \leq r$ CA scheme applied to an WMN, to support multicast routing, we need to construct a routing tree on top of the CA scheme and rooted at the source of a multicast group. Traditional multicast routing algorithms such as Shortest Path Tree (SPT) [25, 26] or Steiner Tree (ST) [27, 28, 29] do not consider the wireless broadcast advantage (WBA) or the underlying CA scheme in an MCMR WMN. The WBA refers to the fact that the transmission of a data packet from a given node to any number of its neighbors in a wireless broadcast medium can be done with a single data transmission. Consequently, these routing trees consume more network bandwidth and resources of routers than necessary to deliver data to their destinations.

We propose multicast routing algorithms for MCMR WMNs that take into account both the WBA and the channel diversity in order to minimize the amount of network bandwidth consumed by a multicast routing tree. The proposed routing algorithms operate on top of a CA scheme based on the $c \leq r$ approach, and thus do not require channel switching. Table 1.2 summarizes the differences between multicast algorithms of the two approaches.

Given an MCMR network and a CA scheme such as [7, 8, 9, 10, 11, 12, 13, 14], our objective is to construct a multicast routing tree that minimizes the total num-

	Multicast algorithms in MCMR WMNs	
	$c > r$ **CA approach**	$c \leq r$ **CA approach**
Operation	A multicast tree is built first. CA is done on top of the tree.	CA is done first. A tree is built next.
Objective	Focus on CA.	Focus on routing.
Channel switching	Yes.	No.
Network coding	Can be included.	Can be included.
Existing algorithms	MCM [19], Yin et al. [20], Chou et al. [21], Lim et al. [22], Cheng et al. [23], Chiu et al. [24], **M4**.	**MCMNT**.

Table 1.2: Multicast algorithms in MCMR WMNs (M4 and MCMNT are our contributions)

ber of transmissions required to deliver a data packet from the source to all multicast destinations. Such a tree is termed Multi-Channel Minimum Number of Transmissions (MCMNT) tree. By minimizing the number of transmissions, we enable more transmissions per time unit, and thus increasing the multicast throughput. At the same time, we minimize the amount of network bandwidth consumed by a multicast routing tree. We first show that this problem is NP-hard. We then propose heuristic algorithms to construct MCMNT trees, including a distributed algorithm for practical implementation in MCMR WMNs. Our work is the first that offers bandwidth-efficient multicast routing algorithms in networks with $c \leq r$ CA schemes. Experimental results show that our routing trees outperform traditional multicast trees such as SPTs, STs and minimum number of forwarders trees (MFTs) [30] in terms of number of transmissions, packet delivery ratio, throughput, end-to-end delay and delay jitter.

1.1.3 Network-coded Multicast

Regardless of the CA approach used, the throughput of an MCMR network can always be improved using network coding. Recently, network coding [43] has received much attention as a promising technique for improving network throughput. With network coding, a node can combine multiple packets within a single transmission, thus making more efficient use of network bandwidth. Previous studies on

	Single-channel networks	MCMR networks
Unicast	Le et al. [31], Chaporkar et al. [32], Scheuermann et al. [34], Lun et al. [35], Yazane et al. [37], Katti et al. [38], Koutsonikolas et al. [39]	Zhang et al. [33], Su et al. [36]
Multicast	Ho et al. [40], Cogill et al. [41], Eryilmaz et al. [42]	None

Table 1.3: Existing studies on network coding.

the benefits of network coding in wireless multi-hop networks have focused mostly on single-channel networks and provided only theoretical bounds [31, 32, 34], making assumptions about a particular coding structure [31] or unrealistic slotted MAC algorithms [32], or considering only simple traffic patterns or network topology [34] (see Table 1.3). In addition, the performance of network coding in the context of multicast communication in MCMR systems has not been studied. This suggests that the performance of network coding for multicast in MCMR WMNs under realistic network settings remains an important issue.

We propose analytical models to estimate the performance of Network-Coded Multicast (NetCoM) in MCMR WMNs. We then present a simulation-based performance evaluation of a multicast session with and without network coding in

MCMR WMNs, using realistic network scenarios and useful performance metrics such as throughput, packet end-to-end delay, and packet delivery ratio. The simulation results are also used to validate the proposed analytical models. Based on the analytical and simulation results, we analyze performance gains of network-coded multicast in MCMR WMNs. To the best of our knowledge, our work is the first that studies the performance of multicast with network coding in MCMR WMNs.

1.2 Thesis Organization

The remainder of the thesis is organized as follows. A literature review of related work is provided in Chapter 2. We present our novel channel assignment scheme for multicast and its advantages over existing CA algorithms in Chapter 3. In Chapter 4, we propose multicast routing algorithms that minimize the amount of network bandwidth consumption in MCMR WMNs. We describe our analytical models for estimating the performance of network-coded multicast in MCMR WMNs and use simulation-based experiments to validate the proposed models in Chapter 5. Chapter 6 concludes the thesis and provides directions for future research and open issues.

Chapter 2

Literature Review

Topics of this chapter begin with background information on wireless communication, followed by a survey of unicast and multicast routing in MCMR WMNs, and performance modeling and analysis of network coding and WMNs.

2.1 Background Information

In this section, we provide an overview of wireless communication, applications of wireless mesh networks, drawbacks of the single-channel single-radio technology, and finally introduce the concept of multi-channel multi-radio (MCMR).

2.1.1 Overview of Wireless Communication

A *radio*, also known as *wireless interface*, is a wireless network interface card (NIC) that is attached to the antenna of a wireless device. A *channel* represents a band

	802.11a	**802.11b**	**802.11g**
Frequency of operation	5.15 - 5.35 GHz, 5.725 - 5.825 GHz	2.4 - 2.4835 GHz	2.4 - 2.4835 GHz
Bandwidth	300 MHz	83.5 MHz	83.5 MHz
No. of orthogonal channels	12	3	3
Data rates	6, 9, 12, 18, 24, 36, 48, 54 Mbps	1, 2, 5.5, 11 Mbps	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54 Mbps
Compatibility		Wi-Fi	Wi-Fi, 802.11b

Table 2.1: The 802.11 family.

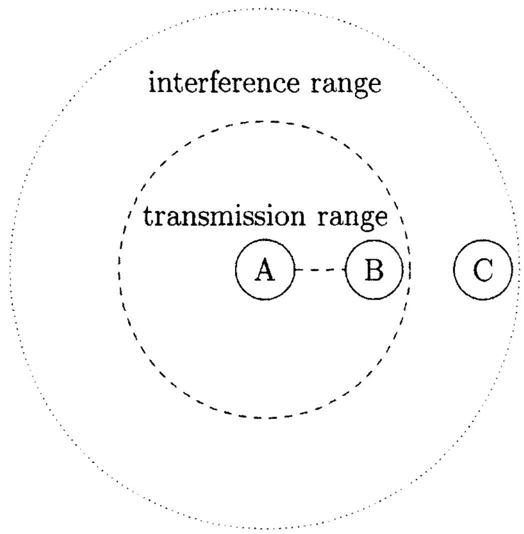


Figure 2.1: Transmission range and interference range.

of frequencies over which signals are carried from sources to destinations. Channels that are more than 25 MHz apart are non-overlapping (orthogonal), i.e. do not interfere with each other. On the other hand, channels that are less than 25 MHz apart are overlapping and hence interfere with each other. In the IEEE 802.11 family [2], the 802.11a standard operates at 5 GHz frequency band and has a maximum data rate of 54 Mbps, with 12 non-overlapping channels. Meanwhile, the 802.11b standard operates at 2.4 GHz and provides a maximum data rate of 11 Mbps, with 3 non-overlapping channels. The 802.11g standard operates at the same frequency band as 802.11b and offers the same maximum rate as 802.11a. Devices based on 802.11g are back-compatible with 802.11b. The 802.11 standard family is summarized in Table 2.1.

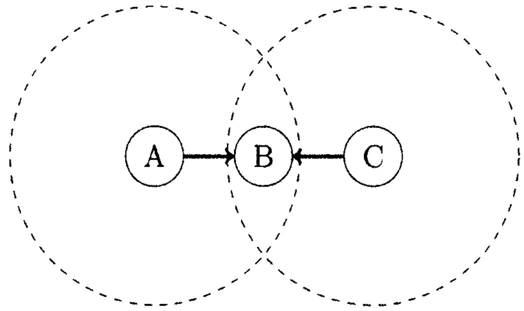


Figure 2.2: The hidden terminal problem.

The *transmission range* or communication range is the range in which a reliable communication between two nodes is possible. In other words, if a receiver B is within the transmission range of node A, it can receive data correctly from A, assuming no interference or noise (see Figure 2.1). Outside the transmission range, a receiver C is not able to receive data correctly from A because the signal from A is not strong enough. However, the signal from A is still strong enough to interfere with C's reception from another transmitter. The range in which transmissions from one node can detrimentally interfere with the transmissions from other nodes on the same or partially overlapping channels is called the *interference range.* It is important to note that the interference range is always larger than and covers the transmission range (Figure 2.1).

In wireless communication, concurrent packet transmissions on the same channel to a node B cause a packet collision at B, destroying the packets being received. To avoid such situation, the Carrier Sense Multiple Access with Collision Avoidance

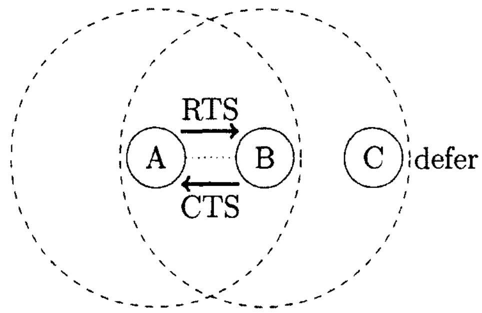


Figure 2.3: RTS/CTS exchange. Any node hearing RTS or CTS will defer channel access.

(CSMA/CA) mechanism [2] is used. Before transmitting on a channel, a node senses the channel to check whether it is idle, i.e., no other nodes are transmitting. However, the CSMA/CA protocol cannot prevent the *hidden terminal problem*, as described below.

A *hidden terminal* refers to a terminal C, which is outside the interference range of node A (Figure 2.2). The hidden terminal C is unable to sense an ongoing transmission from A to B, and hence may try to transmit a packet to B. Consequently, two concurrent transmissions from A and C interfere with each other and cause packet collision/error at the receiving node B.

To solve the hidden terminal problem, a signaling handshake using request-to-send (RTS) and clear-to-send (CTS) messages is exchanged between the transmitter and its intended receiver before the actual transmission takes place, as shown in Figure 2.3. The transmitter A first sends an RTS message to request a transmission

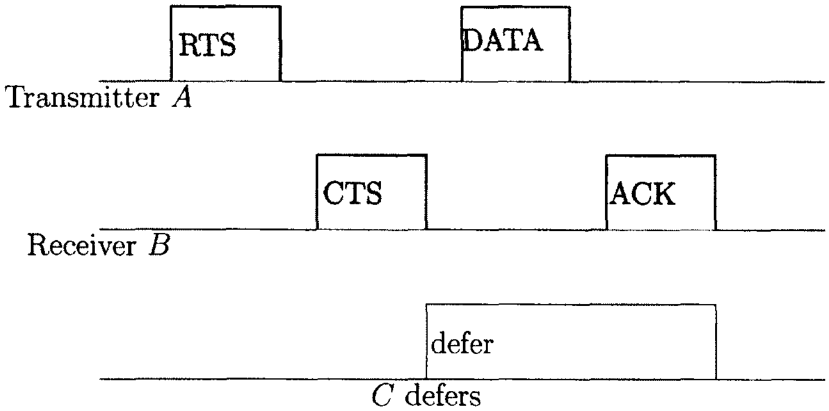


Figure 2.4: Time diagram of transmission with RTS/CTS/DATA/ACK.

and the receiver B replies with a CTS message if it is ready to receive. Upon overhearing the CTS from B, node C will then defer its transmission. For reliability, an acknowledgement (ACK) is added at the end of the data transmission. The time diagram of a complete transmission is illustrated in Figure 2.4.

Note that multicast communication does not make use of the RTS/CTS handshake mechanism, and thus still suffers from the hidden terminal problem. There currently does not exist an effective algorithm for implementing RTS/CTS exchanges at the branch points of a multicast tree, due to the following two reasons. First, CTS messages sent by the multicast receivers of a transmitter have a very high probability of colliding at the transmitter. More importantly, it may not be possible for all the multicast receivers to agree on a common time slot for the transmission of a packet, or the delay would be very long to reach such an agreement,

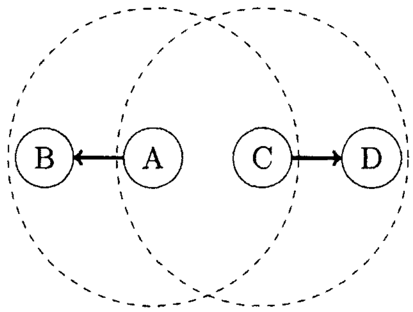


Figure 2.5: The exposed terminal problem.

e.g., polling each receiver one by one [44, 45], or require extensive modifications to the IEEE 802.11 standards [45, 46, 47, 48, 49].

Another issue is the *exposed terminal problem*, which is caused by the carrier sense operation. Figure 2.5 illustrates an *exposed terminal*, node C which is located within the transmission range of node A. In the exposed terminal scenario, when node A transmits a packet to node B which is located outside the transmission range of the exposed terminal C, node C will detect this transmission, and defer its own transmission (to node D) to avoid potential collision. However, this backoff would not be necessary since the transmission from A to B would not interfere with the transmission from C to D. C's deferral lowers its own throughput and degrades the overall throughput of the whole network. Note that the exposed terminal problem is not an operational problem, but instead a performance issue.

2.1.2 Interference Models

Finding interference relations between different nodes is an issue unique to wireless communications. Several interference models have been proposed in literature [50, 51, 52], which try to model real world interference characteristics. In this thesis, we consider two interference models: protocol and physical [50].

In the protocol model of interference, a transmission is considered a failure if there exist multiple transmitters within the sensing range of a receiver and they transmit concurrently on the same channel. The model assumes that interference is a "yes" or "no" phenomenon and is a function of distance between the transmitter and the receiver. In an algorithm that assumes only orthogonal channels, the interference is caused by signals transmitted on the same channel, and thus called co-channel interference. We consider the protocol interference model in the MCMNT algorithms (described in Chapter 4), which are designed for use with orthogonal channels in an MCMR network.

In the physical interference model, interference is measured based on the received signal strength at the receiver and the aggregate noise from all concurrent transmitters. In particular, the transmission is considered successful if the signal to noise ratio (SNR) of the received signal is greater than a pre-determined threshold. In an algorithm that uses overlapping channels, interference is usually measured using the physical model. All else being equal, the higher the degree of separation between two overlapping channels, the lower the interference among their signals. We assume the physical model of interference in the design of the M4 algorithm (in Chapter 3), which makes use of overlapping channels.

2.1.3 Wireless Mesh Networks (WMNs)

WMNs consist of a wireless backbone with mesh routers. One incentive to develop WMNs is to extend the coverage range of current wired or wireless networks in a quick and economical manner. A WMN has low deployment and configuration cost since a mesh router can self-configure and advertise itself to nearby routers to form a network. In addition, we can integrate WMNs with other wireless networks and provide services to end users of these networks.

Research and development of WMNs are motivated by several important applications. For example,

- in transportation systems, wireless mesh networking can effectively support convenient passenger information services, remote monitoring of in-vehicle security video, and driver communications;
- in a building, various electrical devices including power, lights, elevators, and air conditioners need to be controlled and monitored. Currently, this task is

accomplished through standard wired networks, which is very expensive due to the complexity in deployment and maintenance of wired systems. With WMNs, the deployment cost will be significantly reduced and the deployment process is also much simpler thanks to the auto-configuration of mesh connectivity among mesh routers;

- as security surveillance systems have become a necessity for enterprise buildings, shopping malls, and grocery stores in order to deploy such systems at locations as needed, WMNs are a more viable solution than wired networks to connect all devices;

- for emergency rescue teams who usually do not have in advance knowledge of where the network should be deployed, a WMN can be quickly established by simply placing mesh routers at desired locations;

- in a hospital or medical center, monitoring and diagnosis data need to be processed and transmitted from one room to another for various purposes. Traditional wired networks can only provide limited network access to certain fixed medical devices. Wi-Fi networks must rely on the availability of Ethernet connections, which may incur high costs and suffer from the presence of dead zones. Those issues do not exist in WMNs.

These above applications show that WMNs are a new emerging technology of future networking.

2.1.4 Single-Channel Single-Radio WMNs

Traditional WMNs use a single radio per node, operate on one single channel, and, as a result, face a significant limitation of limited network capacity. While the theoretical upper limit of the per node throughput capacity is asymptotically limited by $O(\frac{1}{\sqrt{n}})$, theoretically the achievable capacity of a node in a random static wireless ad hoc network, with ideal global scheduling and routing, is estimated as $O(\frac{1}{\sqrt{nlogn}})$ where n is the number of nodes in the network [50]. Therefore, with the increase of number of nodes in a network, the throughput capacity becomes unacceptably low. The use of a realistic medium access control (MAC), overheads of routing and transport protocols further make the achievable capacity in reality much lower than the theoretical upper limit. It has been found through experiments using carrier sense multiple access with collision avoidance (CSMA/CA) based MAC protocol such as IEEE 802.11 [2] that on a string topology, the throughput degrades approximately to $\frac{1}{n}$ of the raw channel bandwidth [3]. One of the reasons that contributes to such degradation in throughput is the exposed terminal problem (Section 2.1.1), combined with the use of single radio per node in single-channel networks.

2.1.5 Multi-Channel Single-Radio WMNs

Improving end-to-end throughput in multi-hop networks is naturally related to increasing the per-hop throughput, which in turn depends critically on the number of simultaneous transmissions that can be achieved in a given network area. This can be facilitated by the use of multiple channels.

Nevertheless, with only one radio per node, channel switching is required. This switching delay grows with the number of channels. For example, the switching delay for the present 802.11 hardware ranges from a few milliseconds to a few hundred milliseconds [18]. Such frequent channel switching adversely affects the end-to-end delay performance [15].

2.1.6 Multi-Channel Multi-Radio WMNs

Although there exist other factors such as the nature of routing protocols, greediness of some misbehaved nodes, or backoff mechanisms of MAC protocols, one of the most important factors that contribute to rapid throughput degradation is the use of half-duplex single-radio nodes, as explained in the previous section. As a result, the *multi-radio* concept is introduced as a key component in achieving both network scalability and high throughput in future wireless networks. The multi-channel multi-radio (MCMR) technology allows each wireless router to have two or more radio interfaces that operate independently on different channels. Therefore, each router is capable of transmitting and receiving data simultaneously, and thus theoretically doubling the average throughput. In return, MCMR networks require efficient channel assignment (CA) and routing protocols and algorithms that can take advantage of the MCMR technology.

In the following sections, we provide a review of unicast and multicast routing in MCMR WMNs, and prior work on performance modeling and theoretical analysis of network coding and MCMR WMNs.

2.2 Unicast Routing in MCMR WMNs

Routing in MCMR networks is closely related to channel assignment (CA) [53]. The problem of routing and CA in MCMR WMNs has been studied extensively for *unicast communications* [7, 8, 9, 10, 11, 12, 13, 14, 18, 54, 55, 56]. In the context of unicast communications, the routing and CA problem can be classified into three approaches:

- routing first, CA second [7, 10, 14, 18];
- CA first, routing second [9, 11, 12, 13]; and
- joint CA and routing [8, 54].

For instance, the protocol by Raniwala and Chiueh [7] performs routing first, followed by CA. The CA algorithm is called load-aware CA, because the traffic loads of the links are known at the time CA is performed. The protocol carries out the procedure of routing and CA periodically because link traffic loads may change over time.

Tang et al. [9] use the second approach in their algorithm: CA is done first, followed by routing. As a result of the CA outcome, the interference among links is known at the time routing is performed; routing based on this knowledge is thus called interference-aware routing.

Alicherry and Li [8], on the other hand, use linear programming to solve the problems of CA and routing simultaneously, i.e., joint CA and routing, taking into account the inter-dependence between routing and channel assignment to maximize the network throughput. A summary of the unicast CA-routing approaches is listed in Table 2.2.

To the best of our knowledge, there currently exist no work that quantitatively compares the performance of the three approaches, although our qualitative assessment is that the joint CA and routing approach would yield the best network throughput at the expense of high overheads. The joint approach adapts to network conditions and continuously optimizes the CA and routing using up-to-date network information.

Unicast approaches	Representative algorithms
Routing-first, CA-second	Raniwala et al. [7], Das et al. [10], Mohsenian et al. [14]
CA-first, routing-second	Tang et al. [9], Marina et al. [11], Ramachandran et al. [12], Subramanian et al. [13]
Joint CA-routing	Kodialam et al. [54], Alicherry et al. [8]

Table 2.2: Unicast approaches in MCMR WMNs.

2.3 Multicast Routing in MCMR WMNs

Recent work on multicast in WMNs has focused on multicast routing and performance study of routing approaches in single-channel networks [30, 57, 58, 59, 60, 61, 62]. Although the routing and CA problems have been studied extensively for unicast communication, the problem of multicast routing in MCMR WMNs has only been addressed recently. Similar to unicast routing, there currently exist two main approaches to solving the multicast problem in MCMR WMNs. The first approach is called "routing-first, CA-second" wherein a multicast tree is built first and then a CA scheme is applied on top of the constructed multicast tree. Existing algorithms/protocols based on this approach often assume that a multicast tree is available in advance and then focus on the CA problem [19, 20]. In this approach, a node may have more channels assigned to it than the number of radios it possesses, and thus requires channel switching. This is the $c > r$ CA approach introduced in Chapter 1.

The CA algorithm proposed by Yin et al. [20] depends on the use of the probability that a channel is being busy. However, the problem of how to compute this probability is not addressed. Furthermore, collecting and maintaining the probability information of link states for all links in the network would incur significantly high overhead. In [19], Zeng et al. propose a CA algorithm for multicast in MCMR WMNs called Multi-Channel Multicast (MCM). This algorithm suffers from the hidden channel problem (HCP), and must rely on the interference factor. The HCP occurs when two nodes that are two hops away from each other select a same channel to transmit, causing packet collision at a receiver who is in direct transmission range (i.e., one hop away) with both transmitters.

The interference factor between two channels indicates whether interference exists between them and by how much. It depends on various elements such as frequency separation of the channels, the distance and transmission rate of the nodes using the channels as well as environmental conditions like signal reflection or fading. Often, measuring the interference factor is difficult and may not be accurate due to changes in network conditions. The dependence of MCM on the interference factor makes the algorithm inconvenient and unreliable under dynamic network environment where network states change frequently. In this thesis, we propose a HCP-free channel assignment scheme for multicast called Minimum-interference Multi-channel Multi-radio Multicast (M4) that eliminates the use of the interference factor while achieving better performance than MCM. (See Table 1.1 for a qualitative comparison of M4 with existing algorithms in the "routing-first, CA-second" group.)

An alternative approach to addressing the multicast problem in MCMR WMNs is called "CA-first, routing-second". In an MCMR network, when a CA scheme has been applied to the network prior to routing, the design and operation of a new routing algorithm should take into account the underlying CA scheme. Given an MCMR network with a multicast group and pre-assigned channels, the goal of "CA-first, routing-second" based algorithms is to construct a multicast routing tree that optimizes some objective function. To the best of our knowledge, there is currently no algorithm/protocol, prior to our work that studies this routing problem. In this thesis, we propose routing algorithms for multicast named Multi-Channel Minimum Number of Transmissions (MCMNT) in MCMR WMNs that exploit the channel diversity and wireless broadcast advantage in order to minimize the number of multicast transmissions. (Table 1.2 illustrates the status of MCMNT algorithms in comparison with existing multicast CA and routing algorithms for MCMR WMNs.)

In the next sections, we review prior work on performance modeling and analysis of 802.11 networks and network coding.

2.4 Performance Modeling of 802.11 Networks

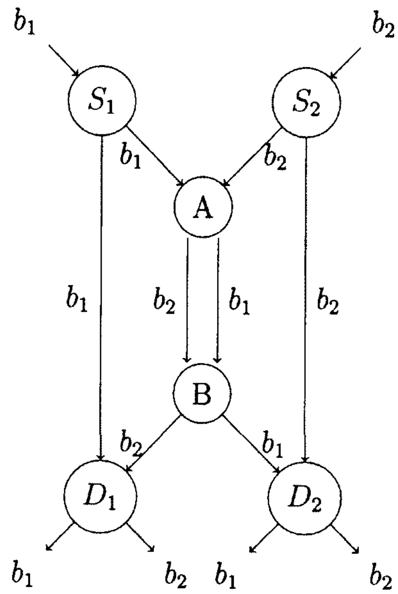
There has been quite a number of studies on the performance of both single-hop and multi-hop 802.11 networks [63, 64, 65, 66, 67, 68, 69, 70]. Bianchi [63] models the behaviour of the exponential backoff time at one unicast node as a discrete Markov chain. It determines the transmission probability and analyzes the saturation throughput under the assumption that in each transmission attempt, regardless of the number of retransmissions, a collision occurs with a constant and independent probability. Kumar et al. [64] present an analysis of Bianchi's model and provide explicit expressions for the collision probability, the aggregate attempt rate, and the aggregate throughput in systems with large numbers of nodes. These studies focus on single-hop wireless networks.

In [65], Medepalli et al. extend Bianchi's model to multi-hop wireless networks and study the effect of hidden and exposed nodes. Yang et al. [66] also extend Bianchi's model to multi-hop networks and investigate the impact of the transmission power and carrier sense threshold on network throughput. Garetto et al. [67] develop an analytical model to compute node throughputs of unicast flows and model flow starvation in a multi-hop wireless network with an arbitrary topology. Wang et al. [68] provide an analytical model to derive the saturation throughput of collision avoidance protocols in multi-hop ad-hoc networks, given the transmission probability at a node. Alizadeh-Shabdiz et al. [69] present models for the throughput performance of single-hop and multi-hop ad hoc networks by assuming a prior knowledge of the neighbors around each node. The authors in [70] use a two-dimensional Poisson distribution for node locations to analyze the throughput and delay of a multi-hop ad hoc network with random topology.

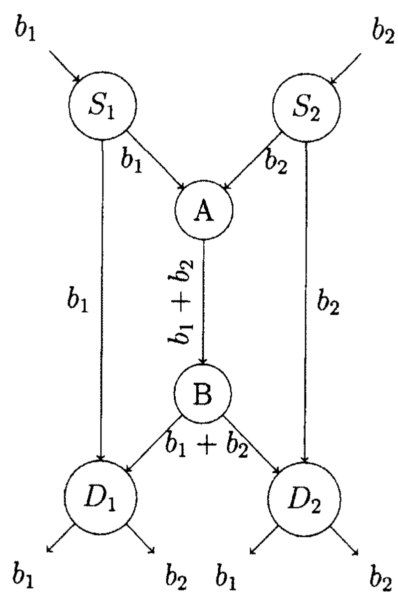
2.5 Performance Analysis and Modeling of Network Coding

Network coding was first introduced in [43]. The advantage of network coding can be seen from the butterfly network shown in Figure 2.6. There are two sources S_1 and S_2 (at the top), each having knowledge of some value b_1 and b_2, respectively. There are two destinations D_1 and D_2 at the bottom. The goal is to deliver b_1 and b_2 to both D_1 and D_2. To illustrate the benefit of network coding, in this example, we focus on the total number of transmissions required to achieve the specified goal.

Let us first consider the scenario in which there is no network coding (Figure 2.6(a)). First, S_1 transmits value b_1 to A and D_1. Similarly, S_2 transmits value b_2 to A and D_2. Next, A transmits value b_1 to B, and then value b_2 also to B. Then, B transmits value b_2 to D_1, and value b_1 to D_2. The total number of transmissions



(a) Without network coding



(b) With network coding

Figure 2.6: The butterfly network.

is therefore six.

Now, we consider the scenario where network coding is used (Figure 2.6(b)). S_1 transmits value b_1 to A and D_1, and S_2 transmits value b_2 to A and D_2. Node A, upon receiving b_1 (from S_1) and b_2 (from S_2), encodes and transmits value $(b_1 + b_2)$ to B, where "+" denotes the modulo 2 addition operation. B then transmits value $(b_1 + b_2)$ to D_1 and D_2. As a result, D_1 receives b_1 and $(b_1 + b_2)$, from which the value of b_2 can be computed. Similarly, D_2 receives b_2 and $(b_1 + b_2)$, from which the value of b_1 can be determined. The total number of transmissions in the case of network coding is four. This example shows that by using network coding, we can reduce the number of transmissions required to deliver a given set of packets to destinations by combining (encoding) multiple packets together.

Performance analyses of network coding in wireless networks [35, 37, 40, 41, 42] have gained much attention recently. Random linear coding of packets in a multicast flow was first introduced in [40], which provides a lower bound on the probability that all one-hop multicast receivers are able to successfully decode the data sent by the source and shows that this scheme can outperform a traditional store-and-forward routing mechanism. In [35], the authors consider a network with finite queue buffers for storing packets and propose a scheme for coding packets of a single unicast flow that arrive through a random process. They provide a framework that allows for the computations of the delay and queue blocking probability. The work in [41] provides analytical bounds on the completion time and stable throughput for random linear coding across multiple multicast flows. In [42], Eryilmaz et al. quantify the performance gains of network coding in terms of completion time from a single source to one-hop receivers with varying channel conditions, modeled as stochastic changes in ON/OFF state. Yazane et al. [37] analyze the throughput of a two-hop wireless network with network coding and model the middle node as a single-server queueing system with finite queue buffers using a continuous-time Markov chain.

In [38], Katti et al. propose COPE, a practical protocol that allow nodes to combine packets together by exploiting the wireless broadcast advantage through opportunistic listening, and show its coding gain over a non-coding scheme in a wireless testbed. Subsequent studies [31, 32, 34] indicate that the coding gain highly depends on network topology, traffic load and traffic pattern. In [39], Koutsonikolas et al. perform a simulation-based study of practical coding gains of unicast flows in single-channel WMNs.

All the above models consider wireless networks with a single channel. There have existed only a few analytical studies on the performance of *unicast* flows in MCMR wireless networks [36, 54, 71, 72]. Cho et al. [71] evaluate the delay and throughput performance of multi-channel wireless infrastructure networks with ring and grid topologies. Their analysis is based on the transmission range, interference range and sensing range. The authors of [54] study the capacity region of a multi-channel multi-radio wireless networks using linear programming. Su et al. [36] extend the work in [54] to model the throughput gain of network coding in two-way, star, and general network topologies. In [72], Li et al. study the queueing delay of a MCMR WMN under various channel fading conditions.

None of the work above, however, addressed the performance of multicast flows in combination with network coding in MCMR networks (see Table 1.3 for a summary of existing studies on network coding). In this thesis, we propose analytical models and provide a performance evaluation of network-coded multicast in MCMR WMNs. To the best of our knowledge, our work is the first that studies the performance of multicast with network coding in MCMR WMNs.

Chapter 3

Channel Assignment for Multicast in MCMR WMNs

In this section, we propose a novel CA algorithm for multicast named Minimum-interference Multi-channel Multi-radio Multicast (M4) that eliminates the inconvenient use of the interference factor, and still allows nodes to operate with minimum interference. This work has been published in [73, 74]. We first begin with a description of MCM, a state-of-the-art CA algorithm for multicast, and then compare our proposed M4 algorithm against it in our performance evaluation.

3.1 Multi-Channel Multicast (MCM)

In this section, we briefly describe the MCM algorithm [19] and analyze its drawbacks.

3.1.1 The MCM Algorithm

MCM uses the interference factor to measure the level of interference among one-hop neighboring multicast nodes. The interference factor is defined as the ratio of the interference range over the transmission range. The transmission and interference ranges depend on various dynamic factors such as transmission rate, transmission power, antenna gain, antenna height, physical layout of network surroundings, signal reflection and fading. As a result, accurately measuring the interference factor is usually difficult and complicated.

In [19], Zeng et al. describe a simple method for measuring the interference range as follows. Using four wireless routers, the authors established two wireless links (one link between each pair of routers) transmitting simultaneously at a specified transmission rate and then moved the two wireless links far away from each other gradually until they no longer interfered. The distance between the two links at this point is considered as the interference range between them and can be used to compute the interference factor defined above. Using this method, the values of the interference factor with respect to different channel separations, when the transmission rate at the physical layer is 2, 5.5 and 11 Mbits/s in 802.11b system, are listed in Table 3.1.

After obtaining the interference-factor values, the MCM channel assignment

Channel separation	2 Mbits/s	5.5 Mbits/s	11 Mbits/s
0	2.5	2.2	2.0
1	1.6	1.5	1.2
2	1.2	1.0	0.7
3	0.9	0.8	0.5
4	0.5	0.3	0.2
≥ 5	0.0	0.0	0.0

Table 3.1: Interference factors.

algorithm is applied upon a pre-constructed multicast tree, as follows. MCM starts with the source node (root) by assigning a channel to a radio/interface of the source. All multicast children of the source node listen to this channel for receiving multicast data from the source. The algorithm then traverses the rest of the multicast tree by following a breadth-first-search order [75]. MCM assigns a channel to a node such that the assignment minimizes interference between the node and its one-hop neighbors, who have been assigned a channel. Let $N(x)$ be the set of one-hop neighbors of node x that have been assigned a channel; c_x denotes the channel that is assigned to node x; and $\delta_{(c_x,c_u)}$ is the interference factor between two channels c_x and c_u. For each node x in the multicast tree (including the source), MCM selects a channel c_x for x so that it minimizes the following function:

$$\sum_{\forall u \in N(x)} \delta^2{}_{(c_x, c_u)} \tag{3.1}$$

If there is more than one channel that satisfies the optimization function, MCM will choose an arbitrary channel among possible solutions. The CA procedure repeats until it covers all nodes in the multicast tree.

3.1.2 Limitations of MCM

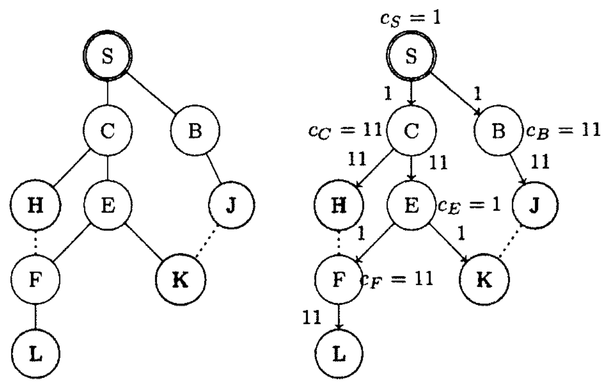
Although the MCM algorithm is simple and efficient, it suffers from the following limitations. First, MCM is exposed to the hidden channel problem (not to be confused with the hidden terminal problem described in Section 2.1.1). Second, its dependence on the interference factor makes MCM an inconvenient and inflexible solution. Third, when multiple choices of channels are available at a node, randomly selecting a channel may lead to non-optimal results at other nodes.

3.1.2.1 The Hidden Channel Problem

When computing the function in Equation (3.1) at a node, MCM only considers interference caused by one-hop neighbors of the node. For example, given a multicast tree in Figure 3.1(a) where node S is the multicast source and nodes H, J, K, L are multicast destinations, the corresponding CA outcome generated by the MCM algorithm is shown in Figure 3.1(b). With regard to node E, the set of neighbors $N(E)$ used in Equation (3.1) is $N(E) = \{C, F, K\}$. The hidden channel problem occurs to the MCM channel assignment as follows. Since node C receives data from node S on channel 1 and is within the transmission range of node E, which also transmits on channel 1, concurrent transmissions from nodes S and E will collide at C. This channel conflict is resulted from the CA computation at node E in which the channel transmitted by node S, a two-hop neighbor of E, is not taken into account. A similar problem exists between nodes C, H, and F on channel 11. Our proposed M4 algorithm eliminates this hidden channel problem by considering interference from both one-hop and two-hop neighboring nodes. Note that the hidden channel problem transforms to the hidden terminal problem (discussed in Section 2.1.1) when the same transmission channel is chosen for neighboring transmitters.

3.1.2.2 Interference Factor

As discussed earlier, the interference factor is dependent on various other factors such as transmission rates at the physical layer, distances between nodes, physical properties of the network. Before applying the MCM algorithm, we have to measure the interference factors in the given network area. However, the interference factors obtained in a network area may not be applicable to other areas because of



(a) A multicast tree (b) CA by MCM

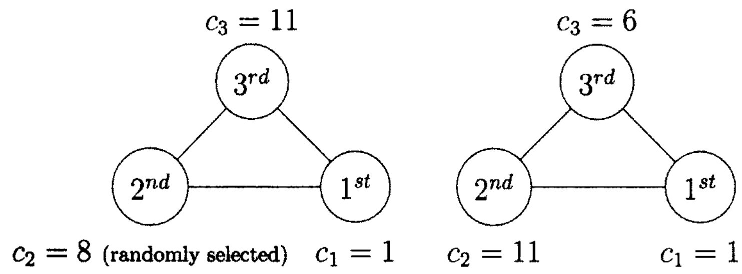
Figure 3.1: The hidden channel problem (HCP). (Dotted lines are not part of the multicast tree but shown to represent direct connectivity between nodes.)

different interference characteristics. Moreover, the interference factor values tend to fluctuate over time due to changing environmental conditions, leading to unstable CA results. To solve this problem, our proposed M4 algorithm avoids the use of the interference factor completely and instead uses a stable metric, channel number, in the CA computation function, yet still providing a lower level of interference compared to MCM.

3.1.2.3 Arbitrary Channel Selection

In some cases, there can exist multiple channels satisfying Equation 3.1. Consider the example in Figure 3.2(a) in which there are three nodes, each being in the transmission range of the other two. The nodes are labelled by the order they are picked for channel assignment. Initially, all three nodes have no channel assigned to them. The CA process begins with node "1^{st}" and assigns channel 1 to it. (In this example, without loss of generality, we assume nodes are running at a data rate of 2 Mbit/s in an 802.11b system where there is a total of 11 channels numbered from 1 to 11, and two non-overlapping channels are separated by at least four other channels. For instance, channels 1, 6, and 11 are non-overlapping.) As non-overlapping channels are not interfered with each other (i.e., the interference factor between them is zero), the objective is to select as many non-overlapping channels as possible to minimize interference, specifically the sum in Equation 3.1. Given that channel 1 has been used for node "1^{st}", all channels from 6 to 11 are good candidates for the second-picked node "2^{nd}". In this situation, MCM chooses a random channel in the pool of $\{6, 7, \ldots, 10, 11\}$ for node "2^{nd}". Suppose that random channel is 8. Finally, using Table 3.1 and Equation 3.1, the channel assigned to node "3^{rd}" is computed to be 11.

The CA solution computed by MCM is thus {1, 8, 11} (Figure 3.2(a)), having a pair of overlapping channels, 8 and 11. However, in this scenario, the optimal solution should be {1, 6, 11}, which contains no overlapping channels, as shown in Figure 3.2(b). The example shows that randomly selecting a channel when multiple channels are possible may not yield the best result. Therefore, we do not



(a) Non-optimal CA in MCM (b) Optimal CA

Figure 3.2: Non-optimal CA resulted from arbitrary channel selection.

use random selection in our M4 algorithm and propose an optimization function that provides optimal solutions in such scenario above.

3.2 The Proposed M4 Algorithm

In this section, we present our proposed CA algorithm for multicast named Minimum-interference Multi-channel Multi-radio Multicast (M4) that offers optimal CA solutions, operates without the need of the interference factor, and solves the hidden channel problem.

3.2.1 System Model and Assumptions

We consider wireless mesh networks with stationary wireless routers/nodes. Two nodes are directly connected if they are within the radio (transmission) range of each other, referred to as one-hop neighbors. Two nodes that communicate via one intermediate node are called two-hop neighbors.

We assume that a multicast tree has been constructed before the M4 algorithm is applied, as in MCM. The goal of the M4 algorithm is to minimize the interference among nodes in the given tree. There exist several approaches for building multicast trees, such as shortest path trees (SPTs) [25, 26, 76, 77, 78, 79, 80, 81], Steiner trees (STs) [27, 28, 29, 82, 83], minimum number-of-forwarders tree (MFT) [30]. The parent-child and sibling relationships between nodes in a multicast tree is the same as those defined for traditional rooted tree data structure [84], where the root of the tree is the multicast source. A multicast forwarder is an intermediate node that forwards multicast data packets from the multicast source towards multicast destinations. The multicast source is considered as a forwarder, and a multicast destination may also be a forwarder forwarding data to other multicast destinations.

When applying the M4 algorithm, each node in the multicast tree uses two radios: one for receiving multicast data from its parent (uplink interface), and the other for sending multicast data to its children (downlink interface). Other remaining radios, if any, can be used for other flows. Note that the multicast source has no uplink interface; and a multicast destination may or may not have a downlink interface, depending on whether it also acts as a forwarding node or not.

We assume that the network topology as well as the multicast membership are static. In practice, routers may be added to, removed from or moved inside the network; and multicast members may join or leave the multicast group freely at will. These events require reconstruction of the multicast tree and re-computation of the channel assignment. These issues are to be addressed in our future work.

3.2.2 The M4 Algorithm

In the M4 algorithm, when selecting a channel, a node considers the channels used by its one-hop and two-hop neighbors. For example, in the multicast tree shown in Figure 3.1, node E takes the channel used by its two-hop neighbor S into consideration, in addition to those used by its one-hop neighbors. This is to avoid the hidden channel problem. The question of how to collect the channel information of other nodes up to two hops away is discussed in Section 3.2.3.

We eliminate the use of the interference factor by developing an optimization function which consists of only channel numbers (e.g, 1, 2, ..., 11). Let $N^*(x)$ denote the set of one-hop *and* two-hop neighbors of node x that are multicast forwarders and have been assigned a channel; and c_x represents the channel assigned to node x. The channel separation between two channels c_x and c_y is defined as $|c_x - c_y|$. For instance, the channel separation between channel 1 and channel 5 is four. In IEEE 802.11b/g standards [2], two channels are orthogonal/non-overlapping (i.e., do not interfere with each other) if the separation between them is greater than or equal to five. For example, channels 1 and 6 are non-overlapping, whereas channels 3 and 5 are overlapping.

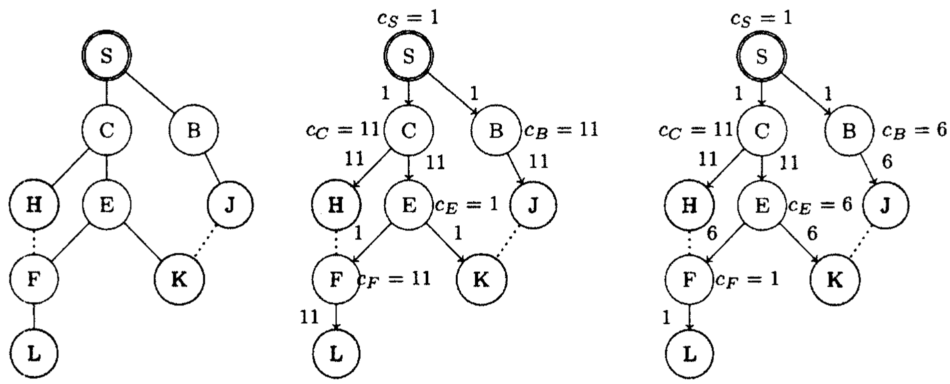
To minimize interference, most existing CA algorithms use only non-overlapping channels. The number of non-overlapping channels is typically scarce, compared with that of overlapping channels. Thus, using only non-overlapping channels limits network performance, while leaving other network resources such as overlapping channels under-utilized or unused. Our proposed M4 algorithm exploits both overlapping and non-overlapping channels, and still sustains the least amount of interference, compared to other CA algorithms.

We define the maximization function $F(c_x)$ at node x as follows:

$$F(c_x) = \frac{\prod_{\forall w \in N^*(x)} |c_x - c_w|}{\max_{\forall w \in N^*(x)} \{|c_x - c_w|\} \div \min_{\forall w \in N^*(x)} \{|c_x - c_w|\}} \tag{3.2}$$

For each forwarding node x in the multicast tree, given K as the set of all channels, M4 assigns to x a channel $c_x \in K$ such that the value $F(c_x)$ is maximal. The maximization function in Equation 3.2 is a ratio in which the numerator is the multiplication of the channel separations between x and w, where $w \in N^*(x)$. The goal is to maximize the channel separations between x and all of its neighbors. To do so, we need to maximize the channel separation between x and *each* of its neighbors, and then use the multiplication to maximize the combination of all the channel separations.

Considering all the channel separations between x and all the nodes in $N^*(x)$,



(a) A multicast tree (b) CA by MCM (c) CA by M4

Figure 3.3: Comparison between MCM and M4. (The hidden channel problem in MCM (Figure 3.1) no longer exists in M4.)

the denominator of the maximization function is the maximum channel separation divided by the minimum channel separation. The objective here is to balance all channel pairs by minimizing the gap between the maximum and the minimum. In other words, we favor a channel that leads to a solution in which the minimum channel separation is close to the maximum channel separation. This helps to avoid the problem present in MCM due to random channel selection wherein one channel pair is "over-separated" (channels 1 and 8), while the other pair is overlapping (8 and 11), as shown in Figure 3.2(a). Thanks to the channel balancing, M4 is able to produce optimal results suggested in Figure 3.2(b).

To further demonstrate the efficiency of M4, we use M4 to assign channels to nodes in the multicast tree shown in Figure 3.1(a). The CA results of M4 are obtained by applying Equation 3.2 to all forwarding nodes in the multicast tree. Compared to MCM, the CA results given by M4 show no hidden channels, no interference among multicast nodes, and are computed without the use of the interference factor. For easy comparison, the multicast tree and the CA results by MCM in Figure 3.1 are re-drawn in Figures 3.3(a) and 3.3(b), respectively, and the CA results by M4 are shown in Figure 3.3(c). The detailed $F(c)$ values of the M4 optimization function for each multicast forwarding node are listed in Table 3.2. In the example, without loss of generality, we assume that $c_S = 1$, i.e., the root node S, the first node to be assigned a channel, is given channel 1.

3.2.3 Implementation Details

In this section, we discuss implementation details for practical deployment of the M4 algorithm with regard to the channel collection of one-hop and two-hop neighbors. To be scalable in large networks, the channel information of one-hop and two-hop neighbors should be advertised and collected in a distributed manner. In M4, we use the following method. A node broadcasts an advertisement message containing its assigned channel to its one-hop neighbors. A neighbor, upon receiving the message, adds its channel information and then re-broadcasts the updated message to its own one-hop neighbors. Besides the channel information, these channel advertisement

Channel	$N^*(C) = \{S\}$	$N^*(B)$=\{S,C\}	$N^*(E)$=\{S,C\}	$N^*(F)$=\{C,E\}
c=1	$F(c) = 0$	$F(c) = 0$	$F(c) = 0$	$F(c) = 25$
c=2	$F(c) = 1$	$F(c) = 1$	$F(c) = 1$	$F(c) = 16$
c=3	$F(c) = 2$	$F(c) = 4$	$F(c) = 4$	$F(c) = 9$
c=4	$F(c) = 3$	$F(c) = 9$	$F(c) = 9$	$F(c) = 4$
c=5	$F(c) = 4$	$F(c) = 16$	$F(c) = 16$	$F(c) = 1$
c=6	$F(c) = 5$	$F(c) = 25$	$F(c) = 25$	$F(c) = 0$
c=7	$F(c) = 6$	$F(c) = 16$	$F(c) = 16$	$F(c) = 1$
c=8	$F(c) = 7$	$F(c) = 9$	$F(c) = 9$	$F(c) = 4$
c=9	$F(c) = 8$	$F(c) = 4$	$F(c) = 4$	$F(c) = 9$
c=10	$F(c) = 9$	$F(c) = 1$	$F(c) = 1$	$F(c) = 16$
c=11	$F(c) = 10$	$F(c) = 0$	$F(c) = 0$	$F(c) = 0$
$c_S = 1$	$c_C = 11$	$c_B = 6$	$c_E = 6$	$c_F = 1$

Table 3.2: The $F(c)$ values in M4.

messages also provide each node v with the set of one-hop and two-hop neighbors, $N^*(v)$.

We include an initial hop-count value of two in each channel entry so that the information is not propagated beyond two hops. The hop count associated with an advertisement message is decremented by one each time the message is forwarded. Once the hop count reaches zero, the message will not be forwarded further. Each message is transmitted three times to increase delivery reliability. To minimize the possibility of collision with transmissions from other nodes, when a node receives a message, it does not forward the message immediately, but waits for some random amount of time. The advertisement messages are exchanged before the actual data transmission starts, and thus do not interfere with the data traffic. If we consider dynamic multicast membership wherein nodes may join or leave during the network lifetime, the channel information will need to be updated based on the changes. In such cases, a more efficient algorithm for nodes to exchange their channel and neighbor information should be considered so that the updates would not over-consume other network resources. This will be addressed in our future work (see also Section 6.4.3).

With respect to the computation of function $F(c)$, when Equation 3.2 gives more than one solution with the same optimal value F_c, we break the tie as follows. We pick the solution with the most number of non-overlapping channels since non-

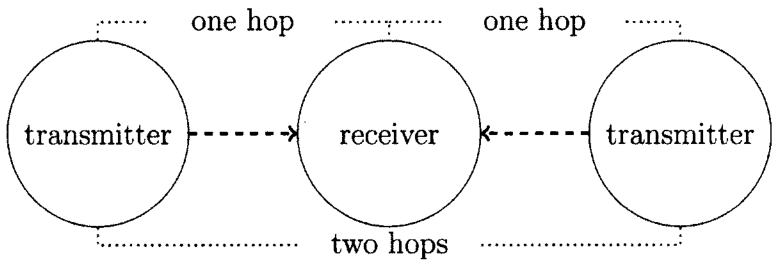


Figure 3.4: Collision at the receiving end.

overlapping channels are more favorable than overlapping channels. If the solutions have the same number of non-overlapping channels, the one with the node having the least number of one-hop and two-hop neighbors is chosen in order to further minimize the interference.

3.2.4 Discussion

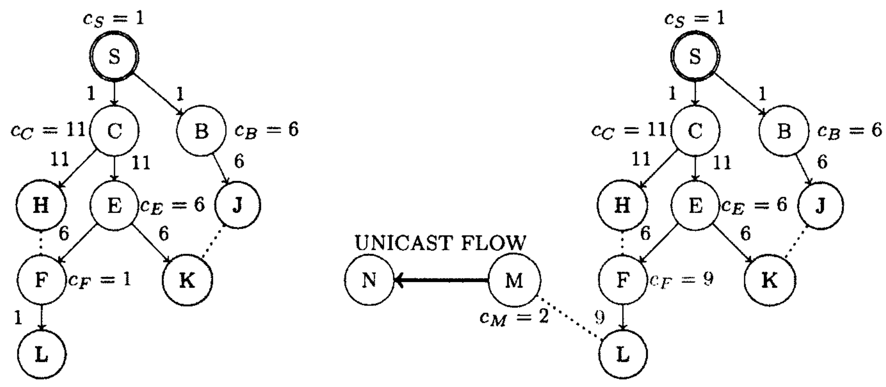
When designing the M4 algorithm, we recognized two types of interference among nodes: intra-flow and inter-flow interference. We discuss how the proposed M4 algorithm handles these types of interference.

3.2.4.1 Intra-flow interference

For a unicast flow, this is the interference among nodes on the path from the source to the destination. For a multicast group, the multicast tree connecting the group is considered as a flow and the multicast intra-flow interference is the interference among nodes in the multicast tree. When selecting a channel for a node, considering the interference from all other nodes in the rest of the tree would require excessive control and management overheads. It would not be necessary, however. In wireless communication, interference occurs at the receiving end when a receiver is within the transmission range (one-hop) of two or more senders transmitting concurrently on the same channel, as illustrated in Figure 3.4. Therefore, it is sufficient to avoid the interference by allocating different channels to the transmitters that are at most two hops apart, assuming the protocol interference model is used. In M4, when we include all one-hop and two-hop neighbors of a multicast node x in the computation, i.e., set $N^*(x)$ in Equation 3.2, we take into account all nodes within the two-hop distance of x, and hence preventing the intra-flow interference.

3.2.4.2 Inter-flow interference

This is the interference among nodes belonging to different flows. To account for the inter-flow interference, we can add the channel information of nodes from other flows



(a) M4 without unicast flow

(b) M4 with unicast flow

Figure 3.5: Inter-flow interference between multicast and unicast flows.

into Equation 3.2 when performing the CA computation for the multicast group. We can even use the M4 algorithm on unicast flows by applying Equation 3.2 to nodes in unicast source-to-destination paths.

Figure 3.5 illustrates an example which again borrows the multicast tree used in Figures 3.1 and 3.3. Suppose that a unicast flow with source M and destination N has been active on channel 2 when the multicast session starts. When we compute a channel c_F for the multicast node F, node M of the unicast flow is included in the set $N^*(F)$. This changes c_F to channel 9 (Figure 3.5(b)), instead of channel 1 when there is no (M, N) unicast flow (Figure 3.5(a)).

In inter-flow interference environment, when a new flow starts, the channel assignment should be recomputed to account for the interference caused by the new flow. Similarly, when a flow terminates, the channel assignment should be updated to exclude the interference of this flow. Furthermore, the traffic loads of different flows should also be taken into account. For instance, a flow with very light load imposes less interference than the ones with heavy loads. However, as network load conditions can be very dynamic, keeping track of every change in the network could be very expensive. A possible approach to this issue is to update the channel assignment periodically, regardless of the network conditions.

In the performance evaluation, we only simulate the intra-flow interference in order to focus on the performance of the multicast flow itself. Our future work will study the performance of the multicast flow in the presence of unicast flows and dynamic network traffic loads.

3.3 Performance Evaluation

We evaluate the performance of M4 and compare it with that of the MCM algorithm under various network scenarios using the QualNet simulation framework [85], one of the leading network simulators for wireless communications. In particular, we compare M4 against two versions of MCM:

- The original version of MCM as proposed by Zeng et al. in [19]. As described in Section 3.1, the CA computation in this version of MCM considers one-hop neighboring nodes only and thus suffers from the hidden channel problem

(HCP).

- Our modified version of MCM, denoted as Improved MCM (i-MCM). In the i-MCM version, we modify MCM to eliminate the HCP by taking into account one-hop and two-hop neighbors as in M4, but preserve the original MCM optimization form of Equation 3.1. By comparing M4 with i-MCM, we show that we can avoid the use of the interference factor without any loss of performance, and that our channel selection strategy still outperforms both MCM and i-MCM.

We use Multicast Ad-hoc On-demand Distance Vector (MAODV) [81], one of the most popular multicast routing protocols for WMNs, to build the underlying multicast routing trees. Following are our performance metrics, simulation parameters and results.

3.3.1 Performance Metrics

We use the following metrics to measure the performance of the M4 and MCM algorithms:

- **Average packet delivery ratio**. The packet delivery ratio (PDR) of a multicast destination is the number of data packets received by the destination divided by the number of data packets sent by the multicast source. The average PDR of a multicast group is the average of the PDRs of all the destinations in the group.

- **Average packet end-to-end delay**. The end-to-end delay (EED) of a packet is the latency from the time the packet is transmitted by the multicast source to the time the packet arrives at a multicast destination. The average over the EEDs of all the packets received at all multicast destinations is the average EED.

- **Average throughput**. The throughput of a multicast destination is defined as the total number of data packets the destination receives divided by the time interval between receiving the first and the last packets. The average taken over all multicast destinations is the average throughput of the multicast group.

3.3.2 Simulation Parameters

The transmission power and transmission range of each node were set constant at 20dBm and 315m, respectively. We simulated a small network of 50 nodes uniformly distributed over a 1000m x 1000m area, and a medium-size network of 100 nodes, over a 1700m x 1700m area. Since in wireless mesh networks, we have a control of where to place routers in a network terrain, we adapt the uniform distribution

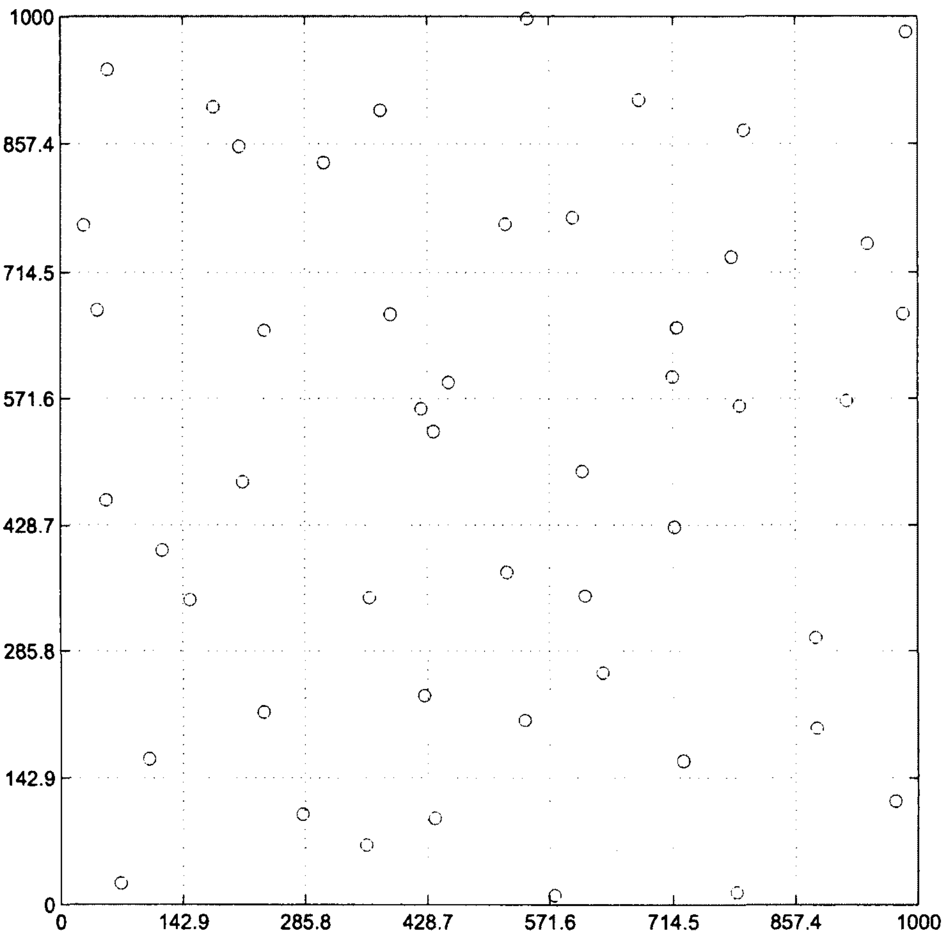


Figure 3.6: Node placement with 50 nodes in 1000m x 1000m.

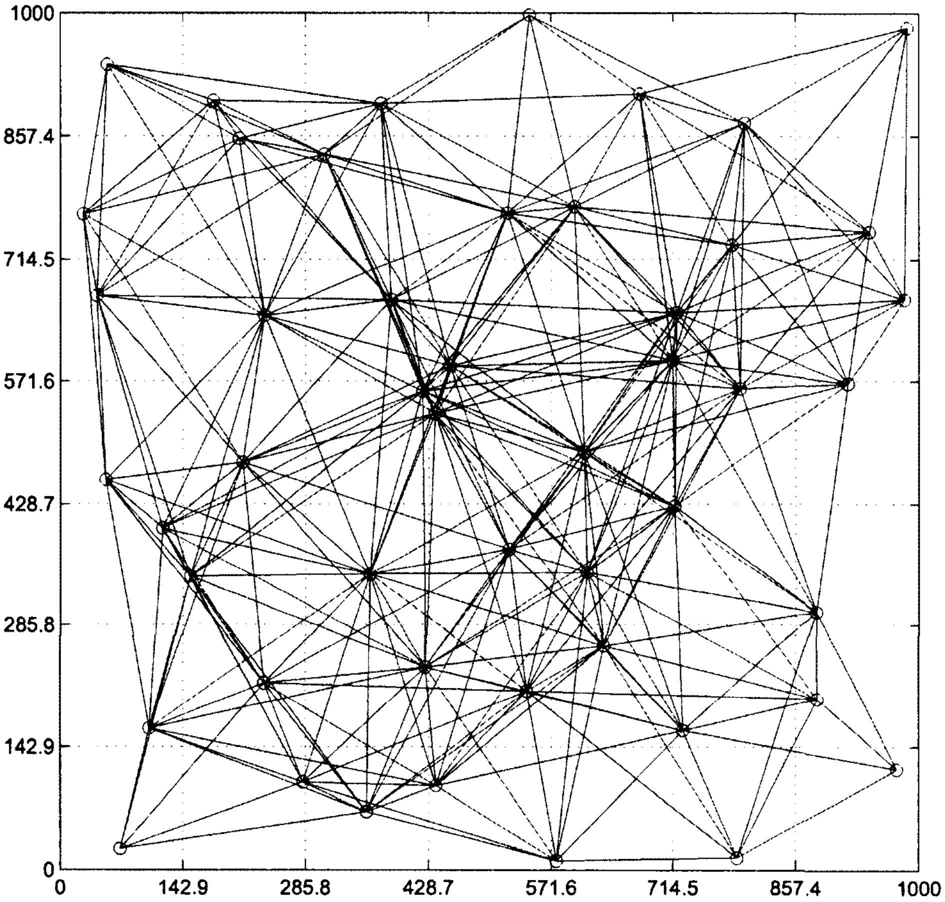


Figure 3.7: Network topology with 50 nodes in 1000m x 1000m.

to distribute loads evenly and to minimize interference among routers. In our simulation, we divided the network area into a grid of equal sub-areas (cells) and then placed each node randomly within a sub-area. The number of nodes and the corresponding network size were chosen in such a way that there were no disjoint nodes nor network partitions throughout the simulation. Furthermore, given the uniform distribution of nodes, the network size was computed such that any one-hop neighboring nodes were within the transmission range of each other, while any two-hop neighbors were outside their transmission ranges. Specifically, given y as the length of the diagonal of a cell in the grid network and R as the transmission range, the network size must satisfy the following conditions: $y < R$ and $2y > R$. Examples of such node placements and network topologies are shown in Figures 3.6 and 3.7.

We used the IEEE 802.11b standard [2] at the physical layer with a transmission rate of 11 Mbits/s. A two-ray propagation model [86] is used when the distance between two nodes is 250m or more; otherwise, a free space model is used to avoid the oscillation caused by the constructive and destructive combination of the two rays over short distances. The above distance threshold for switching between the two models is calculated by the QualNet software.

The IEEE 802.11 Carrier Sense Multiple Access with Collision Avoidance (CS-MA/CA) is chosen as the medium access control (MAC) for multicast transmissions.

We implemented only CSMA/CA without RTS (right to send), CTS (clear to send) or ACK (acknowledgment) for multicast medium access control. There currently does not exist an effective algorithm for implementing RTS/CTS/DATA/ACK exchanges at the branch points of a multicast tree for the following two reasons. First, CTS packets sent by the multicast neighbors of a transmitter have a very high probability of colliding at the transmitter. More importantly, it may not be possible for all the multicast neighbors to agree on a common time slot for the transmission of a packet, or the delay would be very long to reach such an agreement. Therefore, all multicast implementations in 802.11-based wireless networks so far have used only CSMA/CA without RTS/CTS/DATA/ACK exchanges.

The packet size excluding the header was 512 bytes. The size of the queue at every node is 50 Kbytes. Packets in a queue were scheduled for transmission in a first-in-first-out basis. At the transport layer, we did not implement any flow or congestion control mechanism in order to test network resilience under heavy loads.

Each multicast group has one source. The source of a multicast group transmits at a constant bit rate properly set for each experiment. The number of multicast destinations (the group size) is also specified for each scenario. We assume that each source or destination is connected to a different wireless router since our work focuses on the mesh backbone. (In practice, there can be many hosts communicating with a wireless router, e.g., to form a wireless local area network.) The source and the destinations of a multicast group were selected randomly, and the same source and destinations and the same network configuration were used for all CA algorithms in order to obtain a fair comparison. All destinations joined a multicast group at the beginning and stayed until the whole group terminated.

In each experiment, the source sent data for 300 seconds of simulated time, at a constant bit rate specified for each experiment. After the source finished sending, the simulation continued to run for 100 seconds of simulated time to give the last packets time to be processed and routed, for a total of 400 seconds. This 400-second duration does not include the time needed for constructing the routing tree at the beginning. Each data point in the graphs was obtained from 50 runs using different randomly generated seed numbers, and the collected data were averaged over the 50 runs.

The above parameters are summarized in Table 3.3.

3.3.3 Experiment Scenarios

We measured the average packet delivery ratio, end-to-end delay and throughput as functions of

- **multicast source rate** at the application layer. The source rate was varied from 10 to 100 packets/s. There were 20 and 35 multicast destinations in the 50-node and 100-node networks, respectively. The total number of channels,

Parameter	**Value**
Network size	50 nodes over a 1000m x 1000m area 100 nodes over a 1700m x 1700m area
Path loss model	free space for distances below 200m two-ray for distances of 200m or more
Physical layer protocol	IEEE 802.11b
Transmission power	20 dBm
Transmission rate at physical layer	11 Mbits/s
Medium access control	Carrier Sense Multiple Access
Packet size (excluding header size)	512 bytes
Packet queue size	50 Kbytes
Queuing policy	First-in-first-out
Traffic model	Constant bit rate
Duration of each experiment	300 seconds of simulated time
Number of runs per data point	50

Table 3.3: Common simulation parameters.

Function of	**Parameters**	**50-node**	**100-node**
multicast source rate	source rate	10 to 100 packets/s	
	number of channels	11	
	group size	20 nodes	35 nodes
multicast group size	source rate	60 packets/s	40 packets/s
	number of channels	11	
	group size	1 to 30 nodes	1 to 55 nodes
number of channels	source rate	60 packets/s	40 packets/s
	number of channels	1 to 20	
	group size	20 nodes	35 nodes

Table 3.4: Simulation scenarios.

including overlapping and non-overlapping channels, was 11.

- **multicast group size**. The number of multicast destinations in the multicast group ranged from 1 to 30 nodes in the 50-node network and from 1 to 55 nodes in the 100-node network. The total number of overlapping and non-overlapping channels was 11. The multicast source rates were set at 60 packets/s and 40 packets/s in the small and medium-size networks, respectively.

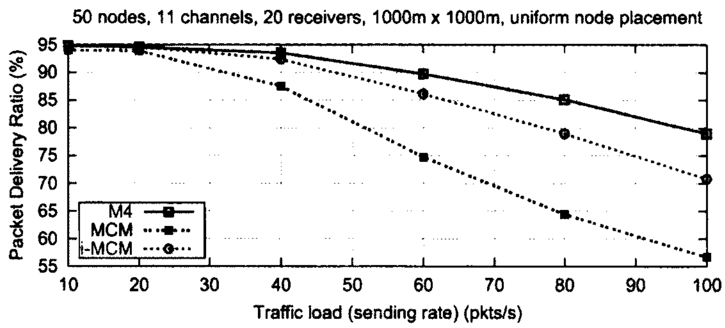
- **number of channels**. The total number of overlapping and non-overlapping channels was varied from 1 to 20. In the small network of 50 nodes, there were 20 multicast destinations, and the multicast source rate was set at 60 packets/s. In the medium-size network of 100 nodes, the multicast group contained 35 multicast destinations and the source rate was fixed at 40 packets/s.

A summary of these above scenarios is shown in Table 3.4.

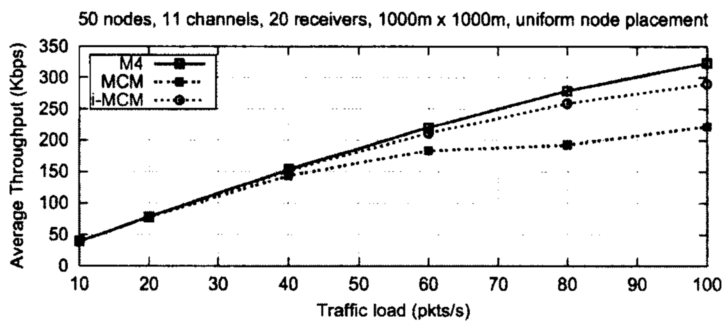
3.3.4 Function of Multicast Source Rate

In this set of experiments, the sender's rate varies from 10 to 100 packets/s. The multicast group in the 50-node network, has 20 receivers, and the results are given in Figure 3.8.

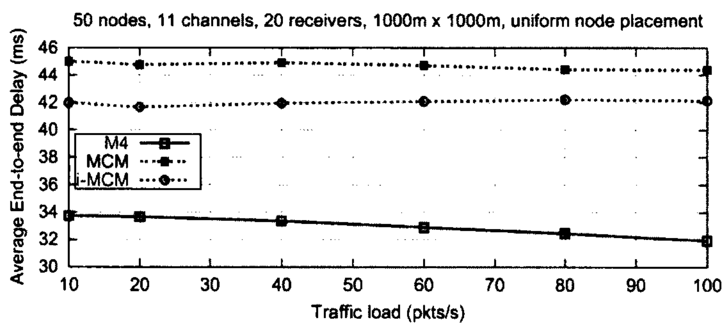
When the traffic load is light (10 - 20 packets/s), the three algorithms perform



(a) Average PDR



(b) Average throughput



(c) Average end-to-end delay

Figure 3.8: Functions of traffic load – 50-node network

similarly with respect to PDR and throughput. When the load is light, there is less medium contention and usage; the multicast group did not take advantage of MCMR. A single channel would have been adequate in this case. Therefore, the three algorithms perform similarly.

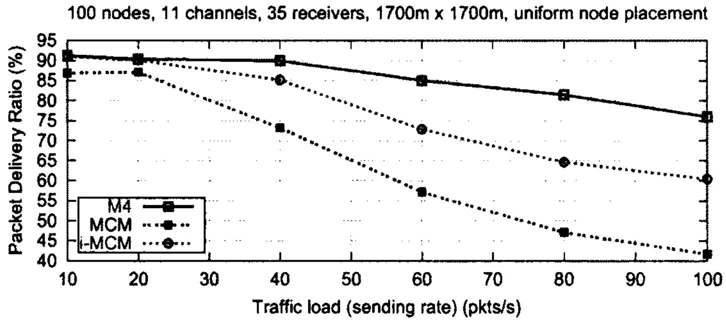
When the traffic load is moderate to heavy (above 40 packets/s), the advantage of MCMR clearly demonstrates, which leads to M4 outperforming i-MCM and MCM. For instance, under heavy load, 80 packets/s, the PDRs of M4, i-MCM and MCM are 85%, 79% and 64%, respectively, a difference of 21% between M4 and MCM.

The performance gap between M4 and MCM is larger than that between M4 and i-MCM. This indicates that the hidden channel problem is the main factor weighing down the performance of MCM.

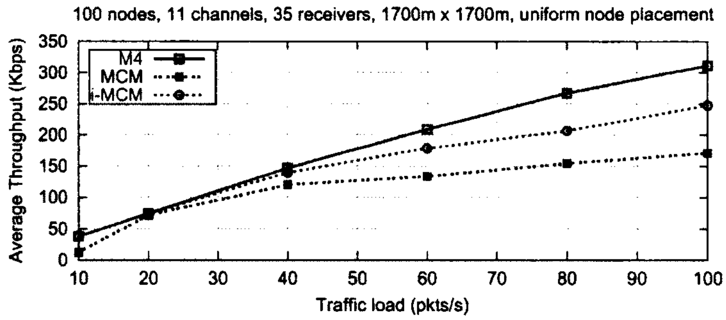
M4 performs better than i-MCM thanks to a channel selection strategy better than random selection when there exist multiple choices. The results also show that using simple channel numbers as a measure of channel separation in M4 is just as effective as using interference factors in MCM.

M4 offers the lowest average end-to-end delay, about 26% and 19% lower, than MCM and i-MCM, respectively. Better CA resulted in lower contention for medium, and thus lower end-to-end delay.

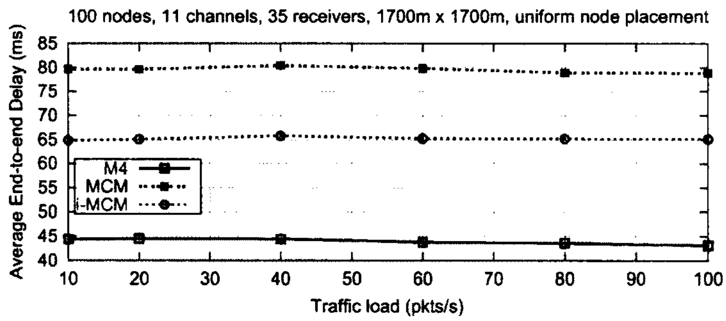
We now examine the performance of the three algorithm as function of traffic



(a) Average PDR



(b) Average throughput



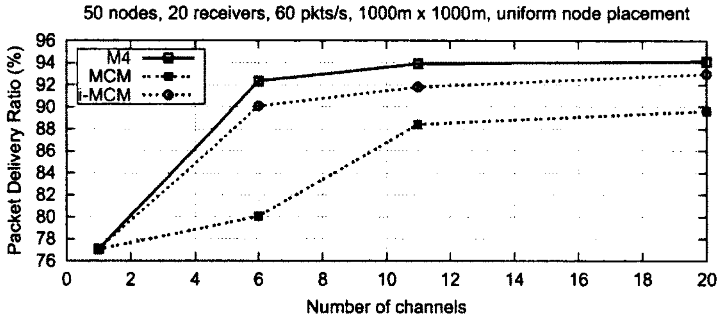
(c) Average end-to-end delay

Figure 3.9: Functions of traffic load – 100-node network

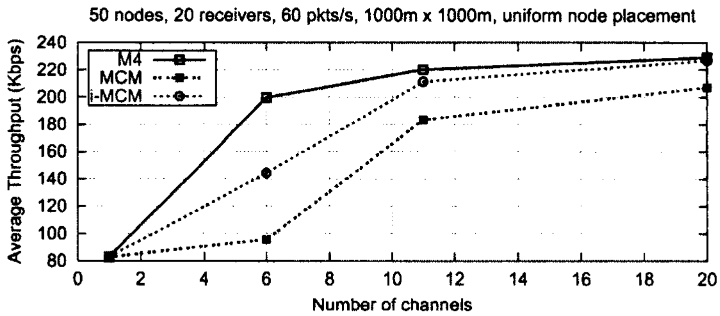
loads in the larger network of 100 nodes with 35 multicast receivers. The graphs are shown in Figure 3.9. As above, M4 performs similarly to MCM and i-MCM under light loads (10 - 20 packets/s), and significantly better under heavier loads with respect to all metrics.

The performance gap between M4 and i-MCM in the 100-node network is more pronounced than that in the smaller network. For instance, when the number of receivers is 20 and the traffic load is 60 packets/s, the PDRs of M4 and i-MCM in the network of 50 nodes are 89% and 86%, respectively (Figure 3.8(a)), while the PDRs in the network of 100 nodes are 85% and 73%, respectively (Figure 3.9(a)). In the same scenario, the average end-to-end delay given by i-MCM is about 19% higher than that of M4 in the smaller network (42 ms versus 33 ms in Figure 3.8(c)), and about 32% higher in the larger network (65 ms versus 44 ms in Figure 3.9(c)). The reason is that longer source-to-destination paths in the larger network take more advantage of the better channel selection algorithm of M4. Similarly, the performance gap between M4 and MCM also widens in the 100-node network. The HCP in MCM caused more collision and congestion when there were more nodes on a source-to-destination path.

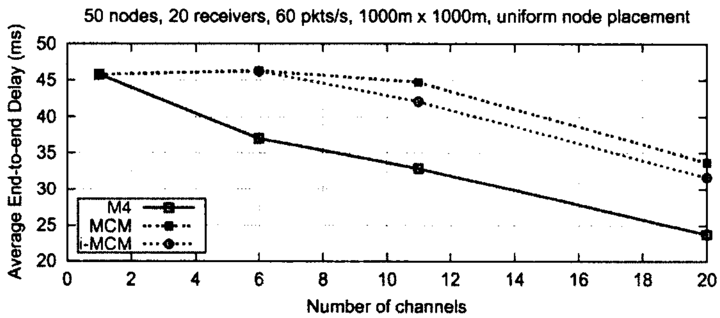
For all three algorithms in both networks, as the sender's rate increases, the throughput increases as expected; the PDR decreases because higher loads cause more congestion and collisions, resulting more packets dropped or damaged.



(a) Average PDR

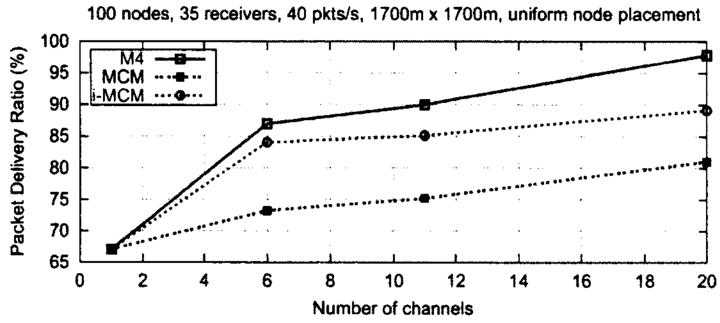


(b) Average throughput

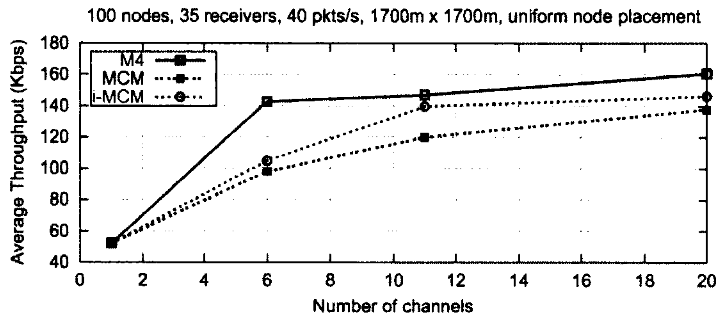


(c) Average end-to-end delay

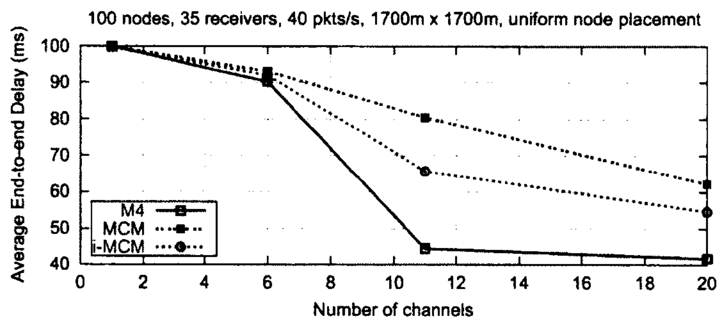
Figure 3.10: Functions of number of channels – 50-node network



(a) Average PDR



(b) Average throughput



(c) Average end-to-end delay

Figure 3.11: Functions of number of channels – 100-node network

3.3.5 Function of Number of Channels

The number of channels in this set of experiments is varied from 1 to 20. The multicast group in the 50-node network has 20 receivers, and its source sends at a rate of 60 pkts/s. This rate yields a moderate load for the given group size in this network.

The results in Figure 3.10 show that M4 and i-MCM outperform MCM in all cases, thanks to the elimination of the HCP. When the number of channels is 20, the PDRs of M4 and MCM are 94.1% and 89.6%, respectively. The average end-to-end delay of M4 is 25.8% lower than that of MCM. Note that in the network with only one channel the results from the three algorithms are almost the same as we would expect.

The performance of M4 is only slightly better than than of i-MCM in this set of experiments. Note, however, that our intention was to replace interference factors with a metric that is simpler, more convenient and more flexible. To that end, our optimization function using simple channel numbers to measure the degree of channel separation proves to be as effective as interference factors, because M4 performs similarly to or better than i-MCM in all cases.

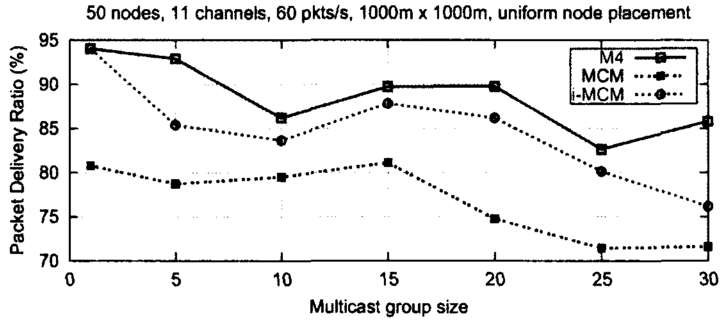
In the 100-node network, we simulated a multicast group having 35 receivers and a source rate of 40 pkts/s (Figure 3.11). Again, M4 performs better than MCM and i-MCM. The performance gap between M4 and MCM/i-MCM magnifies as the network size increases, for the same reason as explained above (longer source-to-destination paths).

For all three algorithms in both networks, as the number of channels increases, the PDR and throughput increase, and the average end-to-end delay decreases. The higher the number of channels, the less time spent contending for the medium. However, the performance of M4 increases at a faster rate than MCM thanks to the elimination of the HCP and better optimization function.

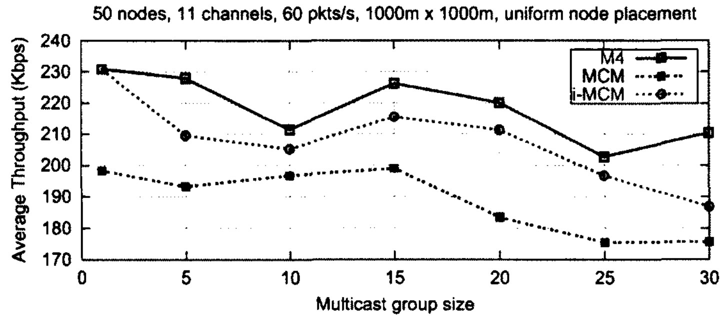
3.3.6 Function of Group Size

In the 50-node network, the multicast group size is increased from 1 to 30, while the number of channels and sender's rate are set to 11 and 60 pkts/s, respectively. The results are shown in Figure 3.12, which indicate that M4 performs better than MCM and i-MCM in all most all cases, especially when the group size is large. For instance, when the number of receivers is 30, the PDR of M4 is 85%, i.e., 9% and 14% higher than that of i-MCM and MCM, respectively (Figure 3.12(a)). Similarly, the average end-to-end delay of M4 is 12.9% lower than that of i-MCM (27 ms versus 31 ms) and 30.7% lower than that of MCM (27 ms versus 39 ms), as shown in Figure 3.12(c).

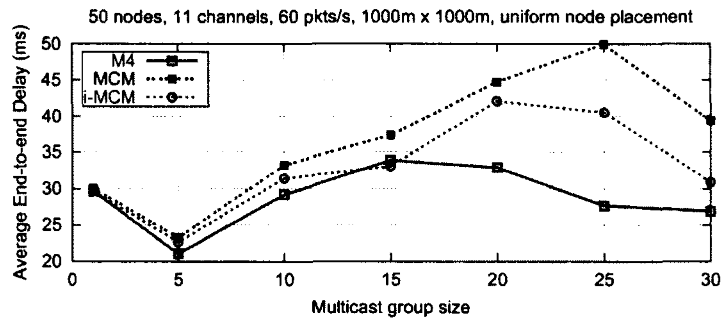
As the group size increases, the more traffic is created in the network. Therefore,



(a) Average PDR

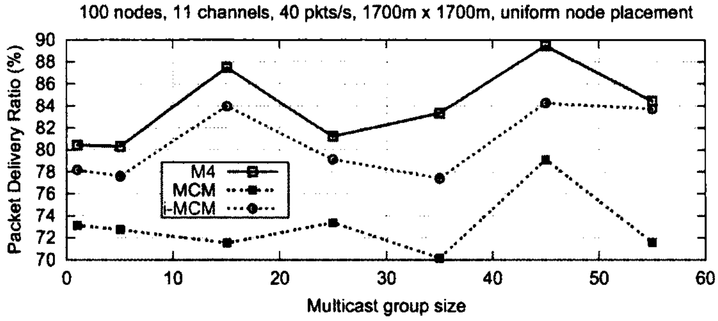


(b) Average throughput

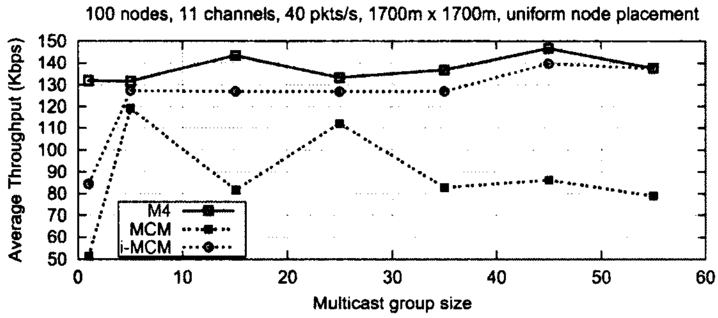


(c) Average end-to-end delay

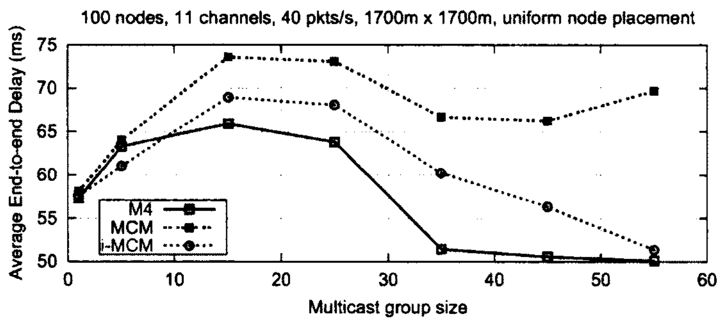
Figure 3.12: Functions of group size – 50-node network



(a) Average PDR



(b) Average throughput



(c) Average end-to-end delay

Figure 3.13: Functions of group size – 100-node network

the PDR and throughput of the three algorithms go down slightly. Similarly, the average end-to-end delays tend to go up as the group size increases.

We conducted the same experiment with 100-node network with the group size varying from 1 to 55. The results of this experiment, shown Figure 3.13, provide a similar comparison between M4, MCM and i-MCM. Specifically, when the multicast group size was 45, M4 achieved a PDR of 89%, while the PDRs of i-MCM and MCM were lower, at 84% and 79%, respectively (Figure 3.13(a)). The average end-to-end delay of M4 was 50 ms, 10.7% and 24% lower than that of i-MCM (56 ms) and MCM (66 ms), respectively (Figure 3.13(c)). Again, although i-MCM performs better than MCM, its results were not as good as M4 due to M4's better optimization function.

It may be noted that the curves in Figures 3.12 and 3.13 do not follow a consistent pattern as the group size increases. The reason is that for each data point, we chose randomly a different group of multicast members. A different multicast group resulted in different locations of multicast destinations, and thus different multicast trees. Therefore, the trends of different group sizes were not consistent. However, our purpose is to show that given any group distribution, M4 outperforms existing algorithms regardless of the locations of the multicast destinations.

3.4 Chapter Summary

In this chapter, we discussed the hidden channel problem, the inconvenient use of the interference factor and limitations of the MCM algorithm. We then proposed the M4 algorithm that does not suffer from the above problems. The optimization function of the M4 algorithm uses only channel numbers and thus does not rely on the computation of the interference factors. Advantages of our proposed algorithm include its simple implementation and high performance in both intra-flow and inter-flow interference environment. The effectiveness of the M4 algorithm is maximized in a network where the multicast group (tree) is dense. In such environment, the number of neighboring nodes around a node is high and thus, without a carefully designed CA algorithm like M4, the probability of channel conflicts among nodes would be very high. Our simulation results showed that the M4 algorithm outperforms MCM in terms of average PDR, throughput, and end-to-end delay under various traffic loads, group sizes and different number of channels.

Chapter 4

Minimum-Bandwidth Multicast Routing in MCMR WMNs

The M4 algorithm is based on the "routing-first, CA-second" approach in which a multicast tree is first established, and then the CA is applied on top of the multicast tree. An alternative approach to multicast support in MCMR WMNs is to apply a CA scheme to the network first, and then build a multicast tree on top of the underlying CA. This "CA-first, routing-second" approach allows quick and easy deployment of multicast services in an MCMR WMN as it can take advantage of existing CAs that are currently deployed in the network.

In this chapter, we propose multicast routing algorithms for MCMR WMNs that minimize network bandwidth consumption. To the best of our knowledge, our work is the first that considers the "CA-first, routing-second" approach for multicast support in MCMR WMNs. Given an MCMR network and its assigned CAs, the objective is to construct a multicast routing tree that minimizes the total number of transmissions required to deliver a data packet from the source to all multicast destinations. Such a tree is termed Multi-Channel Minimum Number of Transmissions (MCMNT) tree. We first show that this problem is NP-hard. We then propose heuristic algorithms to construct MCMNT trees, including a distributed algorithm for practical implementation in MCMR WMNs. Some of the results of this work have been published in [87, 88, 89, 90].

We begin with the problem statement of the MCMNT problem.

4.1 Problem Formulation

We assume that a CA scheme is independently applied to the network prior to the construction of the multicast tree. We also make the following assumptions, which are common constraints imposed by MCMR wireless mesh networking.

- The channels are orthogonal (non-overlapping).
- For any node, the number of distinct channels assigned to the node is less than or equal to the number of radios the node possesses. As a result, each radio is bound to a specific channel and no channel switching is needed. There exist many CA algorithms [7, 8, 9, 10, 11, 12, 13] that produce results satisfying

this condition.

The interference model we assume is the protocol model [91]. In the protocol interference model, a packet is received with errors if the recipient of the packet is located within the transmission range of two or more nodes concurrently transmitting on the same channel.

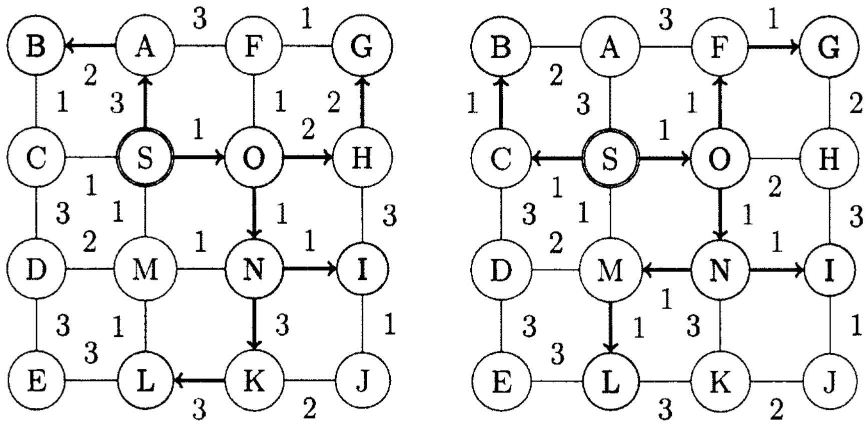
We now define the problem to be solved using an example followed by a formal definition.

4.1.1 Definition by Example

Consider the MCMR network shown in Figure 4.1. Assume that the network has three orthogonal channels and each node has two radios. The number associated with each link indicates the channel assigned to that link by a CA algorithm. (In this example, we use the CA algorithm by Das et al. [10], although the discussion is valid for any CA algorithms with the above assumptions.)

Given a multicast group with source S and six destinations B, G, I, L, N and O (shaded nodes) in Figure 4.1, tree nodes are connected by the thick arrows whose directions indicate the data flow in a routing tree. The originating node of each arrow is called a *forwarding node.* The source is considered a forwarding node. A destination can be a forwarding node, e.g., nodes N and O.

The problem focuses on the number of transmissions a forwarding node requires



(a) Multicast tree T_1 (b) Multicast tree T_2

Figure 4.1: A network with three channels and two radios per node. Each link is labeled with the assigned channel.

to multicast a packet to its one-hop neighbors in the routing tree. In single-channel systems, only one transmission is needed. However, in multi-channel systems, a forwarding node may need more than one transmission due to the channel diversity.

In Figure 4.1, we show two possible routing trees T_1 and T_2 for this multicast group. In tree T_1 (Figure 4.1(a)), N has to transmit two copies of every packet (i.e., two transmissions), one on channel 1 to I and the other on channel 3 to K, which will forward the packet to L. On the other hand, in tree T_2 (Figure 4.1(b)), N needs to perform only one transmission on channel 1 to reach both I and M, which will forward the packet to L. This example shows that the choice of route affects the number of transmissions a node has to perform to forward a data packet.

If we sum up the numbers of transmissions that all forwarding nodes in a routing tree need to perform to deliver a packet to their multicast neighbors, the result is the total number of transmissions the tree incurs to deliver a packet from the source to all the destinations, denoted by $S(T)$. For the example trees T_1 and T_2 in Figure 4.1, $S(T_1) = 9$ while $S(T_2) = 6$. Tree T_2 is therefore preferred because it requires less transmissions per packet and thus consumes less network bandwidth. Among all possible trees connecting the source to the destinations, our goal is to find a tree with the minimum $S(T)$.

Note that the term "transmission" refers to the original transmission of a copy of a packet. In the computation of the number of multicast transmissions, retransmissions caused by packet losses or errors are not counted due to the lack of an acknowledgement mechanism at the MAC layer in the current IEEE 802.11 standards [2]. IEEE 802.11 does not specify an acknowledgement or retransmission mechanism for multicast at the MAC layer due to the following reasons. First, multiple ACK packets sent by multicast receivers of a transmitter have a very high probability of colliding at the transmitter. Second, the delay of receiving ACKs from *all* the receivers would be very long. Although there exist several proposed MAC algorithms for multicast, they incur very long delay [44, 45] (e.g., by polling multicast receivers one by one), or require extensive modifications to the 802.11 MAC protocol [45, 46, 47, 48, 49]. Therefore, at the MAC layer, multicast packets are neither acknowledged nor retransmitted if being lost or damaged. (Recovery of lost or damaged packets can be performed by a reliable multicast protocol at the transport layer, which by itself is a separate and significant research topic, and out of the scope of this thesis.)

4.1.2 Formal Problem Definition

We consider multi-channel multi-radio wireless mesh networks with stationary wireless routers. Two nodes are directly connected and form a communication link if they are within the transmission radio range of each other and share a common channel. We model the network as a connected graph $G = (V, E)$, where V is the set of wireless mesh routers (nodes), and E is the set of communication links (edges). We assume that the connectivity and the channel assignment between any two adjacent nodes are symmetric. That is, if a node u is within the transmission range of a node v, then v is also within the transmission range of u, and if link (u, v) is assigned a channel c, then link (v, u) also uses channel c.

Given C as the set of available channels in the network, and $\mathcal{A}$ as a channel assignment algorithm that maps each link in E to a channel in C.

$$\begin{aligned} \mathcal{A}: \ E &\rightarrow C \\ e &\mapsto \mathcal{A}(e) = c \in C \end{aligned}$$

Let $E_{\mathcal{A}}$ be the set of corresponding links after $\mathcal{A}$ has been applied to E.

$$E_{\mathcal{A}} = \{(e, \mathcal{A}(e)) : \forall e \in E, \mathcal{A}(e) \in C\}$$

Given a channel-assigned graph $G_{\mathcal{A}} = (V, E_{\mathcal{A}})$, a multicast source s and a set of multicast destinations $\Delta \subset V$, the MCMNT problem is to find a multicast routing tree T connecting s to all destinations in Δ such that the total number of transmissions required to deliver one packet from s to the destinations is minimum.

A multicast tree T is defined as an acyclic directed subgraph of the graph $G_{\mathcal{A}}$:

$$T = (\Delta \cup F, \mathcal{E}_{\mathcal{A}}),$$

where F is the set of forwarding nodes in the tree, and $\mathcal{E}_{\mathcal{A}} \subset E_{\mathcal{A}}$ is the set of tree links. Note that $s \in F$, and a destination can be a forwarding node, i.e., it may be that $\Delta \cap F \neq \emptyset$.

Given a node $v \in F$, let $\mathcal{L}_v \subseteq \mathcal{E}_{\mathcal{A}}$ denote the set of outgoing links originating from node v in the tree. For example, the set $\mathcal{L}_N$ of node N in Figure 4.1(a) is $\mathcal{L}_N = \{((N, I), 1), ((N, K), 3)\}$. By projecting every element in set $\mathcal{L}_v$ onto the second attribute, the channel number, we obtain set χ_v, which is the channels v uses to multicast a packet to its neighbours in the multicast tree. We call $\tau_v = |\chi_v|$ the *multicast degree* of forwarding node v, which is effectively the number of transmissions v needs to multicast a packet to its neighbours. As such, the lower the multicast degree, the less bandwidth v consumes to multicast a packet. In the above example, $\chi_N = \{1, 3\}$ and $\tau_N = 2$.

Then, the total number of transmissions (per packet) $S(T)$ incurred by a multicast tree T is the sum of the multicast degrees of all forwarding nodes in T:

$$S(T) = \sum_{\forall v \in F} \tau_v$$

If we consider a single-channel network then $\tau_v = 1$ for $\forall v \in F$. As a result,

$$S(T) = \sum_{\forall v \in F} (1) = |F|,$$

which is equal to the number of forwarding nodes. (The MFT algorithm proposed by Ruiz et al. [30] aims at minimizing the number of forwarding nodes $|F|$ in single-channel networks.)

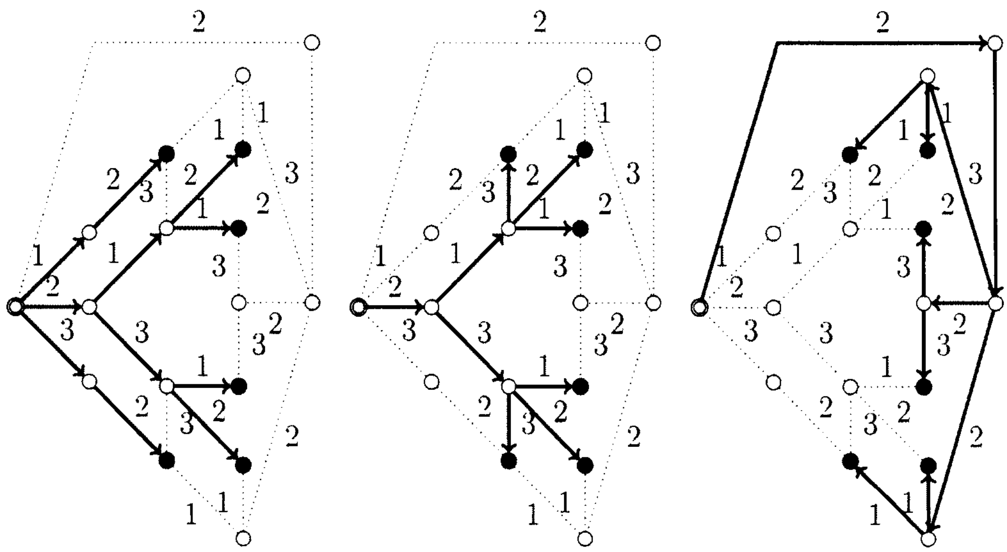
Let $\mathbf{\Psi}$ be the set of all possible multicast trees T connecting source s to all destinations in Δ. A minimum total number of transmissions tree T_m is defined as:

$$T_m = \min_{\forall T \in \mathbf{\Psi}} \{S(T)\}$$

The objective of the MCMNT problem is to find such a tree T_m given $G_{\mathcal{A}}$, s and Δ.

4.1.3 Comparison with Other Types of Trees

Given an MCMR network with channels assigned to its links as in Figure 4.2, we show routing trees computed by different multicast routing algorithms. In particular, we consider the following commonly seen types of trees: Shortest Path Tree



(a) SPT (b) MFT/ST (c) MCMNT

Figure 4.2: Multicast trees constructed by SPT, ST, MFT and MCMNT algorithms.

(SPT), Steiner Tree (ST), and Minimum number of Forwarders Tree (MFT) [30] in comparison with the MCMNT tree.

- SPT: For each source-destination pair, the path with the minimum hop count is computed.

- ST: The tree cost is defined as the sum of the costs of all the edges in the routing tree. An ST tree has the lowest tree cost. If we assume that the cost of every edge is one unit, an ST tree has the least number of edges.

- MFT: An MFT tree is a tree that has the least number of forwarding nodes.

	SPT	MST/MFT	MCMNT
Average path length	2.67	3	4
Number of forwarders	6	4	6
Number of edges (tree cost)	11	9	11
$S(T)$	11	9	6

Table 4.1: Various measures of the trees shown in Figure 4.2.

Assume that the network in Figure 4.2 has three channels and each node has three radios. The number associated with each link denotes the channel assigned to it. The doubly-lined circle represents the source of a multicast group, and the dark nodes indicate the destinations. The trees computed by the SPT, ST/MFT and MCMNT algorithms are given in Figures 4.2(a), 4.2(b), and 4.2(c), respectively. The tree edges are represented by thick arrows. Note that in this particular example, the MFT tree coincides with the ST tree, although this is not always the case in general.

The average source-to-destination path length, tree cost, number of forwarding nodes and number of transmissions $S(T)$ of each tree are summarized in Table 4.1. As we would expect, the SPT has the lowest average path length (2.67); the ST, the minimum number of edges or tree cost (9); the MFT, the minimum number of forwarding nodes (4); the MCMNT tree, the minimum number of transmissions

(6).

4.1.4 Complexity of the MCMNT Problem

Theorem: The MCMNT problem is NP-hard.

Proof: As shown in Section 4.1.2, the MFT problem [30] is a subset of the MCMNT problem in which all links are assigned the same channel. Given that the MFT problem is NP-hard as proved by Ruiz et al. [30], it follows that the MCMNT problem is also NP-hard. □

Since the MCMNT problem is NP-hard, in the next section we propose heuristic algorithms to find approximate solutions.

4.2 The Proposed Algorithms

We present the MCMNT algorithm in Section 4.2.2, followed by its distributed implementation in Section 4.2.3. We first define link and path costs.

4.2.1 Definitions of Link Cost and Path Cost

Given a channel-assigned connected graph $G_A = (V, E_A)$, for each node $u \in V$, $\mu_u(c)$ denotes the number of links that are incident on u and assigned channel c. For example, for node A in the graph shown in Figure 4.1, $\mu_A(1) = 0$, $\mu_A(2) = 1$ and $\mu_A(3) = 2$. Value $\mu_u(c)$ can be considered as the *channel utilization* of channel

c by node u: the higher the value, the more neighbors u can reach with a single transmission on channel c.

Since a *high* channel utilization value is desirable while the heuristic selects paths based on *minimum* costs, we convert channel utilization to a metric whose smaller values are more favorable than higher values in order to perform least cost path selection. In our heuristics, we take the inverse of the channel utilization value $\mu_u(c)$ and assign it to a new channel metric denoted by $\delta_u(c) = \frac{1}{\mu_u(c)}$. For any link $(u, v) \in E$ assigned with channel c, both $\mu_u(c)$ and $\mu_v(c)$ are greater or equal to 1, making the inverse function always defined.

Each directional link (u, v) is associated with a link cost $w(u, v)$ defined as:

$$w(u, v) = \frac{\delta_u(c)}{\delta_v(c)} \tag{4.1}$$

where c is the channel used by link (u, v). The originating node u of the directional link (u, v) is termed the *transmitter*, while the ending node v, the *receiver*. We favor a transmitter with a channel *highly utilized* so that the channel can be used for as many receivers as possible in the final tree in order to maximize the wireless broadcast advantage. This explains the term $\delta_u(c)$ in the numerator of the link cost.

If a link (u, v) using channel c has been added to the tree, the next-hop link (v, z) to be added should avoid using channel c so that transmissions from u and v do not interfere because u and v are one-hop neighbors of each other. This is to prevent the channel conflict between neighboring links. Therefore, given a transmitter u transmitting on a channel c, we should choose a receiver v whose channel c is *lowly utilized* so that node v will have less chance of being selected next as a transmitter on channel c. This minimizes interference among forwarding neighbors. This explains the term $\frac{1}{\delta_v(c)}$ in the link cost, i.e., higher values of $\delta_v(c)$ are more favorable than smaller values.

Finally, let $P(s,d)$ denote a path connecting a source s to a destination d. The path cost $W(P(s,d))$ of path $P(s,d)$ is the sum of the costs of the (directional) links on the path. Let $\Phi(s,d)$ be the set of all possible paths connecting s to d. The least cost path $P_{min}(s,d)$ is defined as the path whose cost is the lowest among all paths in set $\Phi(s,d)$.

4.2.2 The MCMNT Algorithm

Given a MCMR network with pre-assigned channels, the MCMNT algorithm operates by increasing the initial solution tree using least cost paths based on link costs. The heuristic works in a similar manner to the Dijkstra's [25] and Prim's algorithms [92] to some extent. The algorithm is summarized in **Algorithm 1**. Initially, the initial solution tree consists only the source, s. Multicast destinations are then added to the tree one by one using the least cost path from each destination to the current tree (the **while** loop on line 6). In particular, for each node v in the current

Algorithm 1 The MCMNT Algorithm

Input: $G_{\mathcal{A}} = (V, E_{\mathcal{A}})$; source $s \in V$; destination set $\Delta = \{d_1, ..., d_m\} \subset V$;

Output: tree T connecting s to Δ with minimized $S(T)$; set of forwarding nodes F.

Other global variables: current set of unconnected destinations Δ_{cur}; current set of forwarding nodes F_{cur}; current tree T_{cur}.

Initialization: $\Delta_{cur} = \Delta$; $F_{cur} = \{s\}$; $T_{cur} = \{s, \emptyset\}$; compute the costs w of all directional links in $E_{\mathcal{A}}$;

START

while $\Delta_{cur} \neq \emptyset$ **do**

  P_{min} = NULL; {least cost path (LCP) in this round}

  $W(P_{min}) = \infty$; {cost of this path}

  d_{min} = NULL; {destination of this LCP}

  {Find an unconnected destination that can be connected to the current tree with the minimum cost.}

  for all nodes $v \in T_{cur}$ **do**

    Compute the LCP connecting v to each node in Δ_{cur} using Dijkstra's algorithm.

    Among these LCPs, select the path $P(v, d)$ with the smallest cost, where d is some node in Δ_{cur}.

  {Keep $P(v, d)$ if it is better than current P_{min}}

if $W(P(v,d)) < W(P_{min})$ **then**

$P_{min} = P(v,d)$; $d_{min} = d$;

end if

end for

{d_{min} is connected to the current tree with the minimum cost among the unconnected destinations. Add d_{min} and P_{min} to tree.}

$T_{cur} = T_{cur} \cup$ {nodes and links on P_{min}};

$F_{cur} = F_{cur} \cup$ {intermediate nodes on P_{min}};

$\Delta_{cur} = \Delta_{cur} \backslash d_{min}$;

{Update applicable link costs to take advantage of the WBA in the next round.}

for all link (u,v) in P_{min} **do**

{N_u denotes the set of one-hop neighbors of a node u.}

for all $z \in N_u$ **do**

if $z \notin T_{cur}$ and $channel(u,z) = channel(u,v)$ **then**

$w(u,z) = 0$ {link cost set to zero}

end if

end for

end for

end while {terminates when all destinations are connected to T_{cur}}

$T = T_{cur}$; $F = F_{cur}$; **return** $[T, F]$;

END

tree T_{cur}, we find the least cost path connecting v to each node d in the current set of unconnected destinations Δ_{cur} using the Dijkstra's algorithm (line 11). We then consider all the computed least cost paths $P(v, d)$, $\forall d \in \Delta_{cur}$, $\forall v \in T_{cur}$, and select the path P_{min} with the minimum cost (lines 13-14). This path P_{min} and the corresponding destination d_{min} are then added to the solution tree (lines 17-19).

We then update the applicable link costs to take advantage of the WBA in the next round of inserting a new destination to the tree (lines 20-27). Specifically, for each directional link (u, v) on path P_{min} just selected, and for each one-hop neighbor z of u that currently resides *outside* the tree, if link (u, z) is assigned the same channel as link (u, v), we update the cost of link (u, z) to zero (lines 23-24). By doing this, we increase the chance of link (u, z) being selected in the next round. If (u, z) is later added to the solution tree, u will be able to reach z without requiring one more transmission. These link cost updates aim at exploiting the WBA, as suggested by Wieselthier et al. [93]. The above procedure is repeated until all the destinations are added to the solution tree.

4.2.2.1 Running Time Analysis

In the initialization step of the algorithm, every node in V calculates the μ, δ and w values of its outgoing links. Since the maximum number of outgoing links a node can have is O($|V|$) (a complete graph), the running time required to compute all the link costs is $O(|V|^2)$.

In each round of adding a new destination to the solution tree (lines 10-16), we run the Dijkstra's (or Bellman-Ford) algorithm once, which takes $O(|V|^2)$, and obtain the LCP from every node v to every other node in the network. This allows us to find the LCP connecting the current tree to the closest unconnected destination. Updating the applicable link costs after each round (lines 20-27) also takes $O(|V|^2)$ time, as each forwarder in F_{cur} has at most $|V|$ neighbors and there are at most $|V|$ forwarders in the set F_{cur} at any time. Since there are $|\Delta|$ destinations (i.e., rounds), the while loop (lines 6-28) runs in $O(|\Delta||V|^2)$ time.

The overall running time is thus $O(|\Delta||V|^2)$.

4.2.3 The Distributed MCMNT Algorithm

Similar to its centralized counterpart, the distributed MCMNT algorithm builds a multicast tree such that the source is connected to multicast destinations via the least cost paths (using link costs $w(u, v)$ defined by Equation (4.1)) and takes into account both the WBA and the underlying channel assignments. Unlike the centralized version, it uses only local information for routing decisions, i.e., a node uses only the information received from its neighbors. We assume that prior to the start of the algorithm, each node has computed the costs of its outgoing links using Equation (4.1) based on the channel information exchanged with its neighbors.

The algorithm works in two phases. In the first phase, a broadcast tree containing the least cost paths from s to all the other nodes in the network is constructed. The broadcast tree covering all nodes is constructed first in order to consider as many route possibilities as possible to optimize the multicast tree. In the second phase, the resulting broadcast tree is pruned to create a multicast tree.

4.2.3.1 Phase I

The broadcast tree is constructed in a similar manner to the least cost path algorithm by Chandy et al. [94], which is based on distance-vector routing [95]. It uses the link costs $w(u, v)$ defined by Equation (4.1), which considers both the WBA property and channel diversity in the computations of path costs. Phase I requires two types of messages: EXPLORE and REPLY. An EXPLORE message is a two-tuple (u, e_c), where u is the ID of the node sending the message and e_c is the cost of the current best path, as seen by node u, from the source to u's neighbor(s) that is (are) using channel c. A node sends REPLY messages in response to the EXPLORE messages it receives, as will be discussed shortly. The EXPLORE and REPLY messages, besides propagating path costs, ensure that all nodes in the network are covered in the process of constructing the broadcast tree. Following is the detailed description of Phase I.

Each node v maintains the following local variables:

Algorithm 2 The Distributed MCMNT Algorithm: Phase I

if v **receives an EXPLORE message** (u, e_c) **then**
  if $e_c < W_{best}$ **then**
    sends a REPLY_R to current parent p;
    $p = u$; $W_{best} = e_c$; {updates new p and W_{best};}
    {broadcasts EXPLORE messages to neighbors;}
    for all channel $c \in K_v$ **do**
      broadcasts EXPLORE $(v, W_{best} + \min_{\forall z \in N_v(c)} \{w(v, z)\})$;
    end for
  else
    sends a REPLY_R to u;
  end if
end if
if v **receives a** $\textbf{REPLY}_A$ **or** $\textbf{REPLY}_R$ **then**
  saves this reply message;
  if v has received reply messages from all neighbors **then**
    sends a REPLY_A to p;
  end if
end if

- K_v: the set of channels that are assigned to v. For example, in Figure 4.1(a), node M has $K_M = \{1, 2\}$.

- $N_v(c)$: the set of v's neighbors that communicate with v via channel c. Node M above has $N_M(1) = \{S, N, L\}$.

- W_{best}: the cost of the current best path P_{best} from the source to v as currently seen by v. Initially, $W_{best} = \infty$, except for the source where $W_{best} = 0$.

- p: the parent node of v on the best path from which v received the cost W_{best}.

Phase I is summarized in **Algorithm 2**. Phase I starts when the multicast source s broadcasts an EXPLORE message $(s, \min_{\forall u \in N_s(c)} \{w(s, u)\})$ on every channel $c \in K_s$. As an example, consider the network shown in Figure 4.3(a) in which each link is associated with two numbers: the first number (in the square brackets) indicates the channel assigned to the link; the second number denotes the link cost. For instance, link (S, A) is assigned channel 1 and has a cost of 3. Note that links and thus link costs are directional, and we show only the relevant directional links in the network diagram. Source S is connected to two nodes A and B, both links being assigned channel 1. Because on channel 1 $min\{w(S, A), w(S, B)\} = min\{3, 4\} = 3$, source S broadcasts message EXPLORE $(S, 3)$ on that channel. In this case, node S can reach both A and B at the lower cost of 3. The *min* function is used to exploit the WBA property, as illustrated by the above example. That is, if a node v has two neighbours z and z', both links (v,z) and (v,z') are assigned the same channel, and $w(v,z) < w(v,z')$, then it is sufficient for node v to reach both neighbors z and z' at the lower cost $w(v,z)$.

When a node v receives an EXPLORE message (u, e_c) from a node u via channel c, the path cost e_c is compared with the cost of the best path v has seen so far, W_{best}. If $e_c < W_{best}$, this indicates that the path $\{s, ..., u, v\}$ is better than the path v currently buffers$\{s, ..., x, v\}$, where x is another neighbor of v. Thus node v sends a REPLY$_R$ to its current parent x to reject the path $\{s, ..., x, v\}$ as the best path. (Subscript R stands for "Reject".) Node v then sets its parent to u and W_{best} to e_c. Path $\{s, ..., u, v\}$ becomes the current best path from s to v. Node v then broadcasts an EXPLORE message $(v, W_{best} + \min_{\forall z \in N_v(c)} \{w(v,z)\})$ on each channel $c \in K_v$ (lines 2-7). A node, except the source, will send an EXPLORE message to its neighbors every time it observes a path better than the one it currently buffers. This allows the neighbors to update their current best paths as well, if needed.

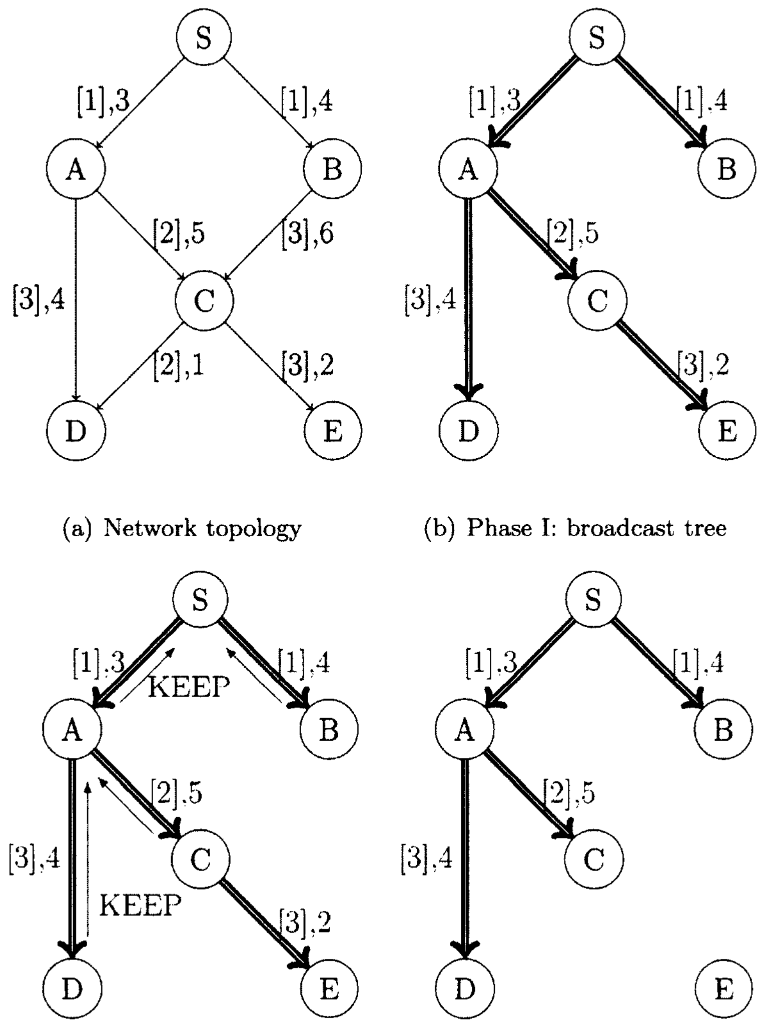
In the example in Figure 4.3(a), upon receiving EXPLORE (S,3) from S, node A sets its W_{best} to 3 and p to S (initially $W_{best} = \infty$ for all nodes). Since A is connected to C on channel 2, it computes $W_{best} + min\{w(A,C)\} = 3 + 5 = 8$, and then broadcasts EXPLORE $(A, 8)$ on channel 2. Node A is also connected to D on channel 3, and thus computes $W_{best} + min\{w(A,D)\} = 3 + 4 = 7$, and then broadcasts EXPLORE $(A, 7)$ on channel 3. A then waits for REPLY messages from all of its neighbors, except the neighbor currently recorded in variable p (its parent in the partially built broadcast tree).

If $e_c \geq W_{best}$, then v simply sends a REPLY$_R$ message to u to indicate that v has already got a path better than the path $\{s, ..., u, v\}$ (line 10). Node u should not wait for any more responses from v since v has got a better path that does not involve u. In our example, when C receives EXPLORE $(B, 9)$ from B on channel 3, it discards this EXPLORE message and immediately sends a REPLY$_R$ message to B since the current W_{best} of C is 8, which is the cost of the path (S, A, C).

When a node v has received replies from *all* of its neighbors, except the neighbor x currently recorded in variable p, this indicates that there is no path from s to v better than the path $\{s, ..., x, v\}$ that v currently buffers. Therefore, v sends a REPLY$_A$ to x, confirming that $\{s, ..., x, v\}$ is the best path from source s to v (lines 15-16). (Subscript A stands for "Accept".) For example, in Figure 4.3(a), after B receives a reply from C, it sends REPLY$_A$ to S, confirming that best path from S to B is (S, B).

In the above step, when counting the number of replies from its neighbors, a node does not distinguish between REPLY$_A$ or REPLY$_R$ messages. In general, if a node v receives K non-duplicate EXPLORE messages from its neighbors, then in the end, v will have sent back K reply messages, one of which is a REPLY$_A$ and $(K - 1)$ are REPLY$_R$ messages. The REPLY$_A$ is for the parent x of v on the best path from source s to v; that is, $p = x$ and W_{best} is the cost of the path $\{s, ..., x, v\}$. The $(K-1)$ REPLY$_R$ messages are for the neighbors whose paths were sub-optimal and rejected by v. In the example in Figure 4.3(a), C will send a REPLY$_A$ to A and a REPLY$_R$ to B, since path (S, A, C) has a lower cost than path (S, B, C), 8 versus 9. Note that before sending a REPLY$_A$ message to the parent node recorded in p, v must receive replies (either REPLY$_A$ or REPLY$_R$) from all of its neighbors, except parent p. Phase I terminates when the source s has received reply messages from all of its neighbors. Variable p of a node v then indicates the parent of v in the broadcast tree. For the example in Figure 4.3(a), the broadcast tree constructed after Phase I is shown in Figure 4.3(b).

Similar to the Internet multicast model, each multicast group is identified by the source ID s and group ID G. Every node in the broadcast tree maintains a routing entry of the form $[(s, G), f, L]$, where f is a boolean variable called *forwarding flag* initialized to false, and L is a list of channels used for multicasting (which will be computed in Phase II when the multicast tree is formed). In Phase II, the broadcast tree will be pruned to form a multicast tree. The nodes that belong to the forwarding group of the final multicast tree will have their forwarding flags set to true and forward data packets for multicast group (s, G). The nodes that are pruned from the broadcast tree keep their routing entries with the forwarding flags unchanged. Following is the detailed description of Phase II.



(a) Network topology (b) Phase I: broadcast tree

(c) Phase II: sending KEEP messages (d) Final multicast tree

Figure 4.3: Example of the distributed MCMNT algorithm

4.2.3.2 Phase II

To start the pruning phase, the source notifies all the nodes by broadcasting a PRUNE message using the broadcast tree constructed in Phase I, as follows. Upon receiving a PRUNE message, a node whose parent p matches with the sender of the message, re-broadcasts the message further. When a multicast destination router d receives a PRUNE message, it creates a KEEP message and sends it to its parent p. Once node p receives the KEEP message, if its forwarding flag in the multicast routing entry is false, it sets the forwarding flag f to true, indicating that it is now a forwarding node in the final multicast tree, adds to the list L the channel assigned to the link from which the KEEP message is received, and then forwards the KEEP message upstream to its own parent. If the forwarding flag of node p is already set to true upon receiving a KEEP message, p will not forward the message to the parent, but simply updates the channel list if needed. In the example in Figure 4.3(c), destination nodes B, C and D will send KEEP messages to their parents. Assume that node A first receives a KEEP message from C then another one from D some time later. Node A updates its multicast routing entry to $[(s, G), \text{true}, \{2, 3\}]$, and forwards the first KEEP message to its parent (but not the second one). The final multicast tree is shown in Figure 4.3(d).

At the end of Phase II, the nodes with their forwarding flags set to true belong to the forwarding group of the final multicast tree. However, we do not remove the multicast routing entries of the nodes with their forwarding flags set to false in order to facilitate future join operations, as described in Section 4.2.4.

In Section 4.2.4, we will also discuss several other issues when implementing the MCMNT algorithms such as reliable transmissions of control message, feedback consolidation, routing with multiple sources, dynamic membership handling and dynamic channel assignment.

4.2.3.3 Running Time Analysis

Let $|V|$ and $|C|$ denote the number of nodes and the number of orthogonal channels in the network, respectively. A node v broadcasts an EXPLORE message on k channels, where $k \leq |C|$ is the number of channels assigned to node v. Therefore, the number of broadcasts incurred by each EXPLORE message is $O(|C|)$. When a node v receives an EXPLORE message from a neighbor offering a path better than the path v currently buffers, it updates its best path information and broadcasts an EXPLORE message to its neighbors. If v has n neighbors, in the worst case, v will have done n updates and broadcasts n EXPLORE messages. A node can have a maximum of $(|V| - 1)$ neighbors, so the number of EXPLORE message broadcasts carried out by node v is $O(|C||V|)$. As there are $|V|$ nodes in the network, the total number of EXPLORE message broadcasts incurred by the algorithm is $O(|C||V|^2)$.

A node v will wait for REPLY messages from all of its neighbors. The number of replies received by node v is thus $O(|V|)$. The total number of REPLY messages in the whole network is therefore $O(|V|^2)$.

In phase II, each node v will broadcast PRUNE messages on the channels assigned to v. The number of PRUNE message broadcasts at each node is thus $O(|C|)$, or $O(|C||V|)$ for the whole broadcast tree. Finally, the number of KEEP messages is $O(|V|)$, since a node sends a KEEP message only once to its sole parent in the broadcast tree.

Overall, the total number of messages required by the distributed MCMNT algorithm is the sum of the numbers of EXPLORE, REPLY, PRUNE and KEEP messages, which is $O(|C||V|^2)$.

4.2.4 Implementation Issues

In this section, we discuss the following issues when implementing the MCMNT algorithms: reliable transmission of control messages, feedback consolidation, multiple sources, dynamic multicast membership, and dynamic channel assignment.

4.2.4.1 Reliable Transmissions of Control Messages

Because of the lossy nature of wireless links, each broadcast of an EXPLORE or PRUNE message is repeated multiple times for reliability. For REPLY or KEEP messages, the transmissions are unicast-based (i.e., one to one) and thus rely on the RTS/CTS/DATA/ACK exchange of the IEEE 802.11 medium access control protocol for reliability.

4.2.4.2 Feedback Consolidation

In Phase I of the distributed MCMNT algorithm, a node waits for reply messages from *all* of its neighbors before it can send a REPLY_A message to its parent. Similarly, Phase I terminates when the source s has received reply messages from all of its neighbors. Getting feedback from all neighbors may be infeasible (if one or more of them fail, or did not receive an EXPLORE message), or may incur very long delay (if one neighbor is overloaded). Therefore, a node will set a timer after sending out an EXPLORE message. If the timer expires before the node receives feedback from all the neighbors, it will go ahead and send a REPLY_A message to its parent (or terminate Phase I if it is the source).

4.2.4.3 Multiple Sources

In this section, we discuss how MCMNT builds its routing entries when there exist multiple multicast sources in the network.

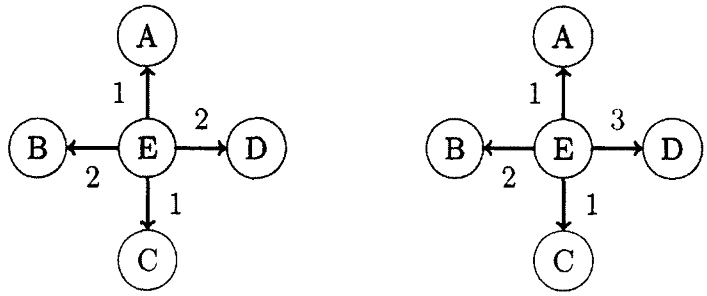
If each source s belongs to a different multicast group G, then s constructs its own multicast tree as specified by the above MCMNT algorithms. Each multicast tree and its associated routing entries are uniquely identified by the tuple (s, G).

In cases where multiple sources $\{s_1, \ldots, s_n\}$ send data to the same multicast group G, a unified multicast tree is built as follows. First, each source s_i, $i \in [1, n]$, builds its own multicast tree using an MCMNT algorithm, resulting in a routing entry of the form $[s_i, G, f_{i,v}, L_{i,v}]$ at each forwarding node v. (The computation of routing entries is described in Section 4.2.3.2.) After all n trees for n sources have been constructed, a forwarding node v may belong to k trees and have k routing entries belonging to group G. Node v consolidates these k routing entries into a single entry as follows. The node replaces the source ID with symbol $*$, indicating that it is forwarding data for a multiple-source group. It then applies the logical OR operation to all the forwarding flags in these k routing entries: $\mathrm{OR}\{f_{i_1,v}, \ldots, f_{i_k,v}\}$. If the result is true, this means that the node will forward data for at least one source of group G. Finally, the node takes the union of the channel lists: $\cup\{L_{i_1,v}, \ldots, L_{i_k,v}\}$, which specifies the channels on which it has to transmit data for this group. The final routing entry is thus the result of $[*, G, \mathrm{OR}\{f_{i_1,v}, \ldots, f_{i_k,v}\}, \cup\{L_{i_1,v}, \ldots, L_{i_k,v}\}]$.

4.2.4.4 Dynamic Membership Handling

Hosts can join (or leave) a multicast group using the Internet Group Management Protocol (IGMP) [96]. The IGMP tells the end-router to which a host is connected whether the host wishes to subscribe to a multicast group. If the end-router is currently not in the multicast tree, it can join the group by sending a KEEP message to its parent in the broadcast tree, as done in Phase II (Section 4.2.3.2). The KEEP message will be propagated upstream along a branch of the broadcast tree built in Phase I until the end-router is connected to the current multicast tree. (This is similar to the "graft" operation in DVMRP [78].) For this reason, we did not remove the physical routing entries of the non-multicast nodes from the broadcast tree when we pruned its branches in Phase II. In addition, we add one variable q to the routing entry of each node v in the broadcast tree, which records the number of child routers of v in the multicast routing tree, as follows $[(s, G), f, L, q]$. When v receives a KEEP message from a downstream node (in this "graft" procedure or in Phase II), it increments q by one.

When all hosts connected to an end-router r have left the multicast group (via IGMP) and r does not have any child routers to forward data to, r can remove itself from the group to save network bandwidth. This is done in a similar manner to the pruning process in Phase II. The end-router first sets its forwarding flag f to false, and then sends a LEAVE message to its parent. When a node v receives a LEAVE message, it decrements its variable q by one. If q becomes zero, v now has no children in the multicast tree, and can thus leave the group too if v has no subscribed hosts connected to itself. The LEAVE message is propagated upstream until the unused branch is "disconnected" from the current multicast tree (i.e., the



(a) Old CA (b) New CA

Figure 4.4: Dynamic channel assignment example

routers on the pruned branch set the forwarding flags of group (s, G) to false).

4.2.4.5 Dynamic Channel Assignment

To simplify the discussions above, we have so far assumed a static CA scheme, which is applied to the network before the MCMNT algorithms are executed. In practice, the MCMNT algorithms can be used in networks where the CA is dynamic. In some schemes [7, 8], which we term *global CA update*, the CA of the whole network is updated periodically (i.e., every two minutes, using a distributed algorithm) to reflect the current network conditions. In others, the CA is updated locally when a node detects significant changes in traffic or channel conditions around it; we call this approach *local CA update.*

After a global CA update is completed, this triggers the source of a multicast group to run the MCMNT algorithm to re-construct the routing tree accordingly.

There is some delay, albeit short, before a node completely updates its routing table since it may forward data for many flows. During this period, the node uses the new CA scheme but possibly some *old* routing entries, which may result in connectivity disruption as illustrated by the following example in which every node has three radios. In Figure 4.4(a), node E forwards multicast data of group (s, G) to shaded nodes A, B and D. The number associated with each link indicates the channel assigned to that link. The multicast routing entry at node E is thus $[s, G, \text{true}, \{1, 2\}]$. After a CA update, the channel between E and D is now changed to 3. (E would send a message on the old channel 2 to tell D to switch to channel 3.) However, the old routing entry tells E to forward data to its multicast neighbors only on channels 1 and 2. Node D thus will not receive the data. To prevent this connectivity disruption problem, when node E updates the CA, it must also add any new channel(s) to the MCMNT routing entry. In this example, the multicast routing entry becomes $[s, G, \text{true}, \{1, 2, 3\}]$ so that E will also send to D. (Note that a new channel may occur on a non-multicast link; for example, the channel between E and C is changed to 3, while the channel between E and D remains 2. Nevertheless, the new channel still has to be added to the channel list of the multicast routing entry of node E to prevent a potential connectivity disruption, because E does not know the IDs of its multicast neighbors.) The channel list at this point may not be efficient in terms of number of transmissions. However, this transition period is expected to last for a short period of time, until a new, optimized routing entry overwrites the old routing entry.

In local CA update schemes, after a multicast node performs a local channel update, it sends a request to the source asking it to initiate route establishment to update the tree. There are two issues to consider. First, many multicast nodes in a local region of the tree may send the same request, consuming network bandwidth unnecessarily. Thus there should be a mechanism to suppress redundant requests, e.g., using random timers in combination with overhearing to suppress one's own request if someone else has sent one. Second, frequent tree updates consume network resources. Therefore the source should collect requests and perform route establishment *periodically* and not in response to each request. Until the next route establishment round, a node with CA changes can update its channel list(s) as described above to prevent connectivity disruption. Again, these channel lists may not be efficient, but the inefficiency is limited to a local area of the tree.

It is worth mentioning that the trade-off between tree update frequency, update overheads, required resources, update latency, and service disruption is a challenging research issue in multicast routing in general, and not just multicast routing in MCMR WMNs.

4.3 Experiment Setting

We denote our proposed centralized and distributed algorithms by C-MCMNT and D-MCMNT, respectively, and refer to both C-MCMNT and D-MCMNT as MCMNT. We compare the performance of MCMNT trees with that of SPTs, Steiner trees [27] and MFTs. These three protocols are well designed and evaluated [57, 58, 97] and will likely be candidates for deployment in real-life WMNs. We did not consider the existing multicast routing protocols proposed for MCMR WMNs such as [19, 24, 74] in the comparison because of the lack of an effective channel switching algorithm required by these protocols, as discussed in Section 6.4.1.

We use QualNet [85], a software that provides scalable simulations of wireless networks and a commercial version of GloMoSim, for the performance evaluation.

4.3.1 Simulation Parameters

We simulated a medium-size network of 50 nodes and a large-size network of 100 nodes uniformly distributed in a 1200m $\times$ 1200m area and 1700m $\times$ 1700m area, respectively. The transmission power of the routers is set constant at 20dBm; the data transmission rate at the physical layer is 11 Mbits/s; the transmission range of the wireless routers is 315m, according to the specifications of wireless routers manufactured by Tropos [98]. The IEEE 802.11 CSMA/CA protocol without RT-

S/CTS exchange is chosen as the medium access control protocol for multicast transmissions.

At the transport layer, we do not use any flow or congestion control mechanisms in order to test the network performance under very high loads. The multicast group has one source placed at the center of the network, while the destinations are randomly selected. We assume that each source or destination is connected to a different wireless mesh router. That is, a multicast group with d destinations consists of d destination routers and one source router, since we are interested in the multicast performance of routers in the mesh backbone. The same source and destinations are used for all algorithms (SPT, ST, MFT, and MCMNT) to obtain a fair comparison. All destinations joined a multicast group at the beginning and stayed until the end of the simulation. In each experiment, the source transmits at a specified constant bit rate (CBR) for 600 seconds of simulated time. The simulator then continues to run for 100 seconds of simulated time to give the last packets time to be routed. Each data point in the graphs is averaged from 50 runs using different network topologies and random seeds and plotted with a confidence interval of 95%.

The parameters common to all experiments are summarized in Table 4.2.

Parameter	**Value**
Network size	50 nodes in 1200m x 1200m 100 nodes in 1700m x 1700m
Path loss model	free space for distances < 250m two-ray for distances ≥ 250m
Fading model	none
Transmission range	315m
Transmission rate at physical layer	11 Mbits/s
Physical layer protocol	PHY802.11b
Medium access control	MAC802.11 with DCF
MAC for multicast flows	CSMA/CA
Packet size (excluding header size)	512 bytes
Queue size at routers	50 Kbytes
Queuing policy at routers	first-in-first-out
Traffic model of sources	constant bit rate (CBR)
Duration of each experiment	600 seconds of simulated time
Number of runs per data point	50
Confidence interval	95%

Table 4.2: Common simulation parameters

4.3.2 Performance Metrics

We use the following performance metrics.

- **Total number of transmissions per packet**. This metric is defined in Section 4.1.2 and measures the bandwidth efficiency of a multicast tree.

- **Average packet delivery ratio**. The packet delivery ratio (PDR) of a destination in the multicast group is the number of data packets actually delivered to the destination over the number of data packets sent by the source. The average PDR of a multicast group is the average of the PDRs of all the destinations in the group. This measures the delivery efficiency and reliability of a multicast protocol.

- **Average end-to-end delay**. The end-to-end delay (EED) of every packet received at every destination is recorded; the average over all the packets received is then computed. This metric is important in real-time interactive applications such as on-line multi-player games and audio/video conferencing.

- **Average throughput**. The throughput is defined as the total number of data packets a destination actually receives divided by the time between receiving the first packet and the last packet. The average taken over all the destinations is the average throughput of the multicast group.

- **Average delay jitter**. Delay jitter is the variation (difference) of the inter-arrival intervals from one packet received to the next packet received. Each destination calculates the average delay jitter from the received packets. The average delay jitter is the average of the per-destination delay jitters taken over all the destinations. This is an important metric in audio/video applications.

For each type of tree, we measure the above performance metrics as functions of

- *multicast group size.* The number of multicast destinations varies from 10 to 40 in the 50-node network and from 10 to 70 in the 100-node network. The number of radios per node and the number of available channels are fixed at 3. The source transmits at a rate of 250 and 200 packets/s in the 50-node and 100-node networks, respectively. (The source rates were chosen so that we can observe the performance degradation as the group size increases, and the PDRs still stay at acceptable values (e.g., above 60%).)

- *multicast traffic load.* The multicast source rate at the application layer increases from 100 to 350 packets/s. The number of channels and the number of radios per node are set at 3. The multicast groups consist of 30 and 50 destinations in the 50-node and 100-node networks, respectively.

- *number of orthogonal (non-overlapping) channels.* In this scenario, we increase the number of channels from 1 to 7. The number of radios per node is 1 for one channel, 2 for two channels, and 3 for 3 to 7 channels. The multicast group size and source rate are 30 destinations and 250 packets/s in the 50-node network, respectively, and 50 destinations and 200 packets/s in the 100-node network.

- *number of multicast sources.* The number of multicast sources/groups varies from one to four. Each multicast source belongs to a different multicast group. A multicast group consists of 30 and 50 destinations in the 50-node and 100-node networks, respectively. Multicast destinations are chosen randomly and may belong to more than one multicast group. The number of channels is set at three. Each multicast source transmits at a rate of 250 and 200 packets/s in the networks of 50 and 100 nodes, respectively.

4.4 Experimental Results

The graphs in Figures 4.5 to 4.10 show that the C-MCMNT and D-MCMNT trees give the best results in most cases. Following is a detailed discussion of the experimental results.

4.4.1 Function of Multicast Group Size

The performance of the protocols as functions of the group size in the smaller network of 50 nodes is given in Figure 4.5. The graphs in Figure 4.5(a) show that in most cases the MCMNT trees require the least numbers of transmissions, followed by the MFTs, STs and SPTs. This shows the effectiveness of the proposed MCMNT algorithms. Low numbers of transmissions enable the MCMNT trees to outperform the other types of trees, in terms of PDR, throughput, end-to-end delay and delay jitter. Furthermore, the bigger the multicast group, the wider the performance gap between the MCMNT trees and the other trees. In particular, when the group size reaches 40, the PDR of the D-MCMNT tree is 6%, 8%, 14% higher than those of the SPT, MFT and ST, respectively; the PDR of the C-MCMNT tree is about 2% higher than that of the D-MCMNT tree (see Figure 4.5(b)). The average throughputs of the MCMNT trees are also higher than those of the other trees (Figure 4.5(c)). As for average end-to-end delays (Figure 4.5(d)), the C-MCMNT and D-MCMNT trees are capable of keeping the values stable and below 0.01 and 0.02 seconds, respectively. On the other hand, when the number of multicast destinations increases to 40, the end-to-end delays of the SPT, MFT and ST jump to 0.05, 0.06 and 0.14 seconds, which are 2.5, 3, and 7 times longer than that of the D-MCMNT tree. With regards to delay jitters (Figure 4.5(e)), the MCMNT trees also offer noticeably better performance than the other trees.

The superior performance of the MCMNT trees is thanks to their low numbers of transmissions required to multicast data packets to the group members. A low number of transmissions helps reduce the contention level among nodes, and the probability of packets colliding with others. This leads to higher PDRs and throughputs. Less transmissions and less contention also result in lower end-to-end delays and jitters.

In all types of trees, as the group size increases, the performance of all trees degrades, as expected. As more destinations are added, the multicast tree becomes more dense, leading to a higher traffic volume in the tree and thus more contention and packet collisions. However, the performance of the SPTs, STs and MFTs degrades at a higher rate than that of the MCMNT trees.

The above observations, comments and explanations also apply to the results obtained from the experiments with the 100-node network shown in Figure 4.6. Specifically, as the number of multicast destinations increases to 70, the PDRs and throughputs of the MCMNT trees are 5% higher than those of the MFT and ST, and about 15% higher than those of the SPT (Figures 4.6(b), 4.6(c)). The end-to-end delays of the MCMNT trees are kept below 0.05 seconds, while the respective values of the MFT, ST and SPT are two to four times higher (Figure 4.6(d)). Similarly, while the MCMNT trees keep their delay jitters below 0.02 seconds, the

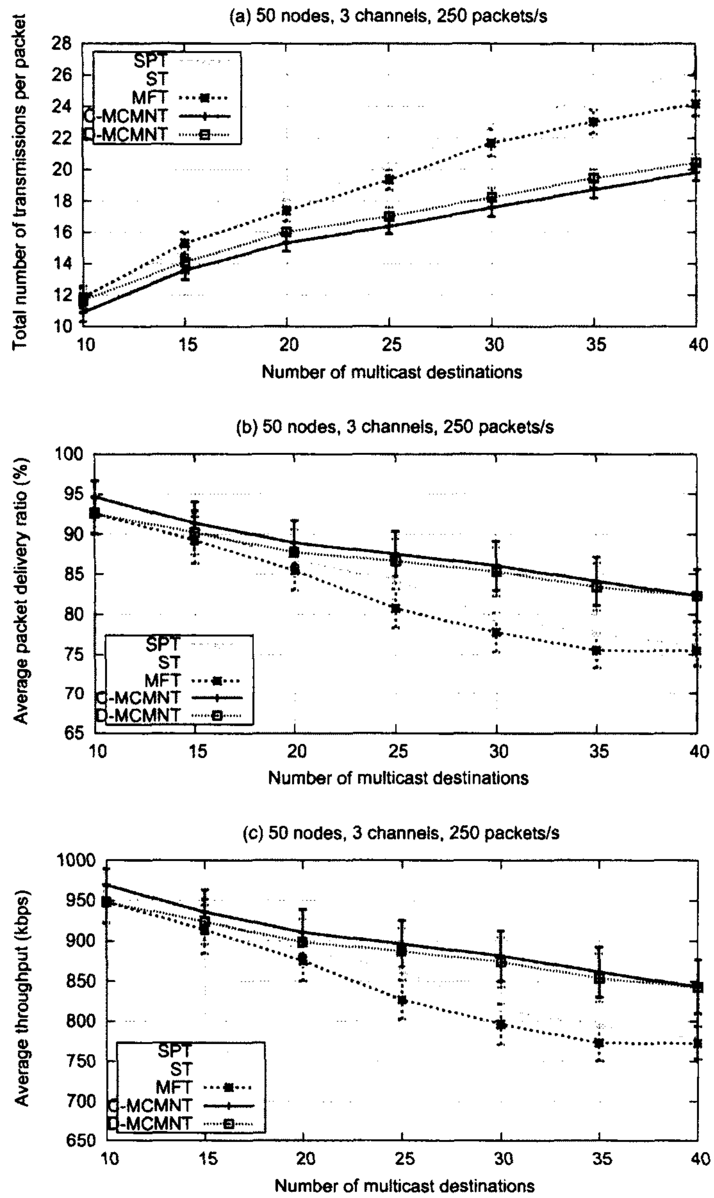


Figure 4.5: Functions of group size: 50 nodes

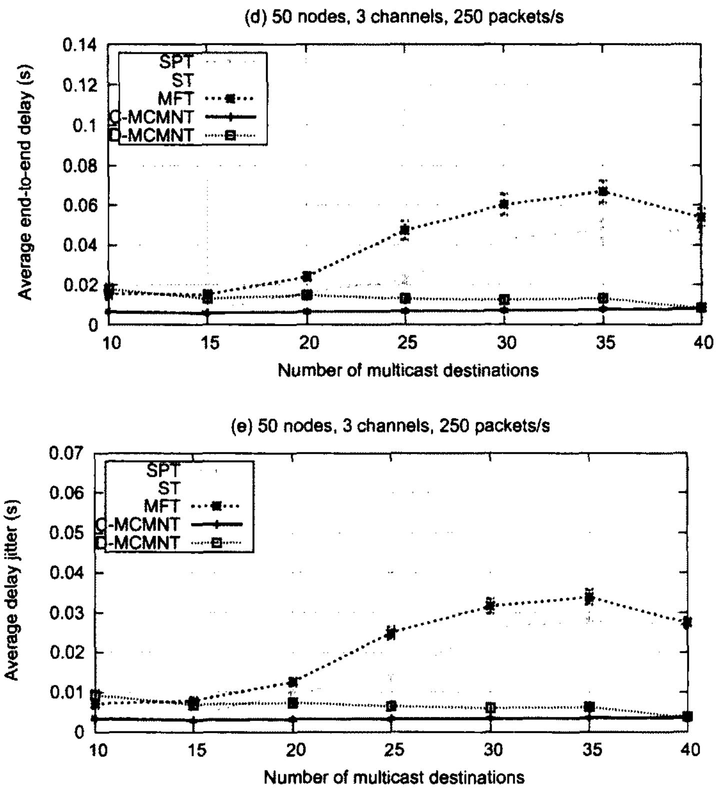


Figure 4.5: Functions of group size: 50 nodes (continued)

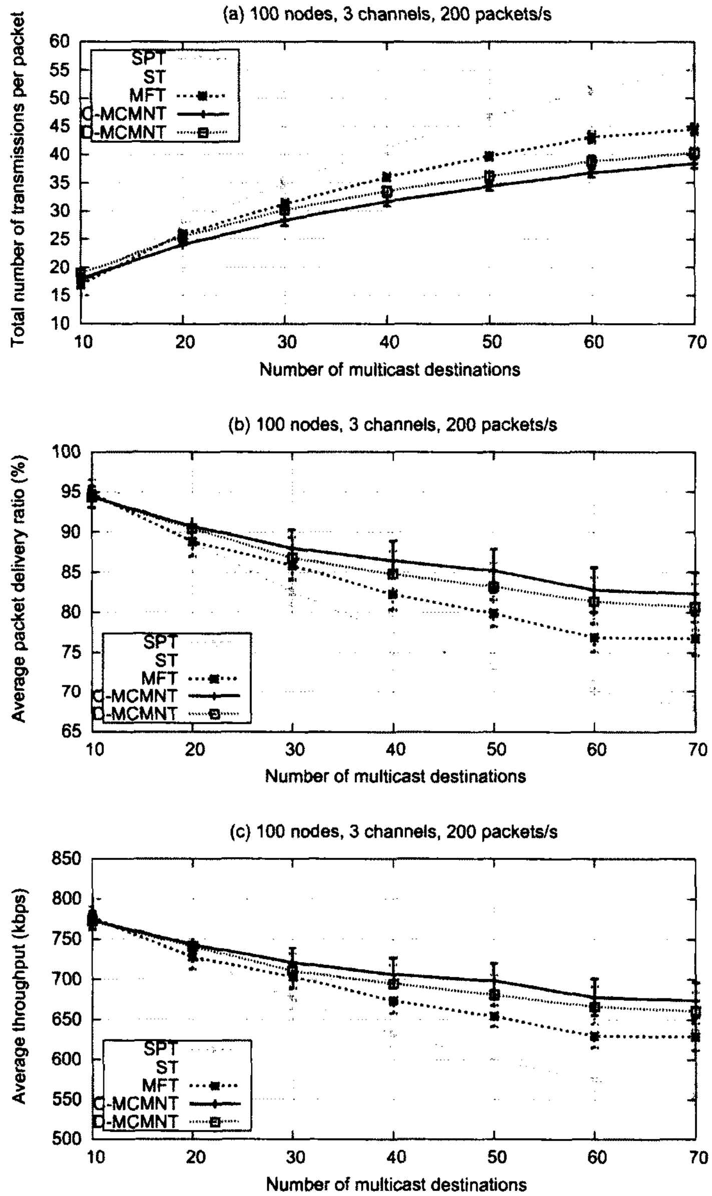


Figure 4.6: Functions of group size: 100 nodes

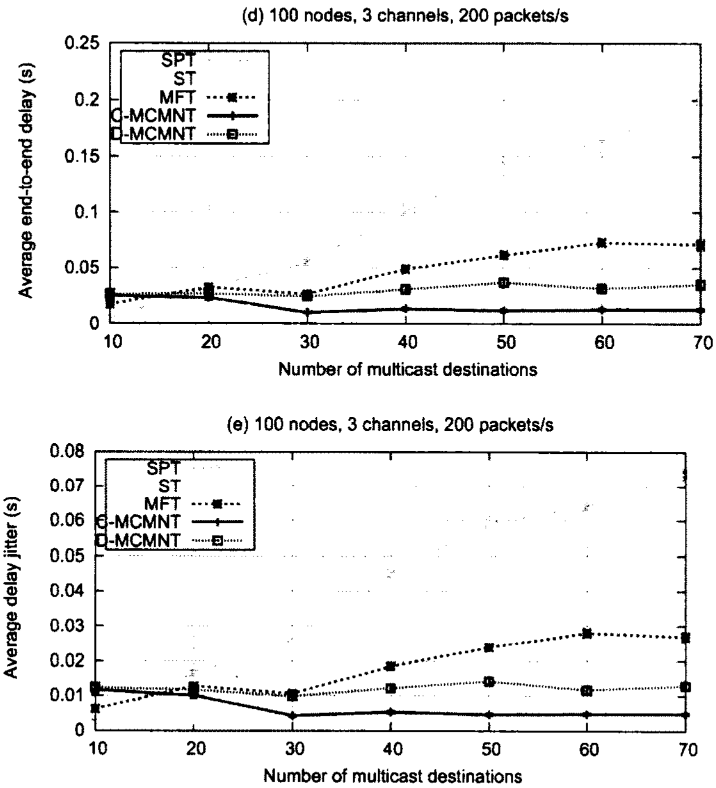


Figure 4.6: Functions of group size: 100 nodes (continued)

delay jitters of the MFT, ST, and SPT are relatively high, at 0.03, 0.035, and 0.07 seconds, respectively (Figure 4.6(e)).

Readers may note that the SPTs perform better than the STs in the 50-node network, but the situation is reversed in the larger 100-node network. Part of the reason is the source-to-destination path length, which also plays a role in the performance of a flow. In general, the longer the path length, the higher the loss rate and contention, leading to lower PDR, throughput, and longer delay [57]. In the 50-node, the SPTs and STs have similar numbers of transmissions (Figure 4.5(a)), but the SPTs, by their nature, have shorter average path lengths. Therefore, the SPTs give better performance than the STs. In the 100-node network, the SPTs require more transmissions per packet than the STs (see Figure 4.6(a)). Although the average path lengths of the SPTs are still shorter, this is not enough to make up for their higher numbers of transmissions. In the end, the STs offer better performance than the SPTs in the larger network. The above comparison emphasizes the importance of minimizing the number of multicast transmissions in MCMR networks, and that, in fact, is the objective of the proposed MCMNT algorithms.

4.4.2 Function of Multicast Traffic Load

Figures 4.7(a) and 4.8(a) show that in both networks, the C-MCMNT trees have the lowest numbers of transmissions per packet (NTP), followed by the D-MCMNT

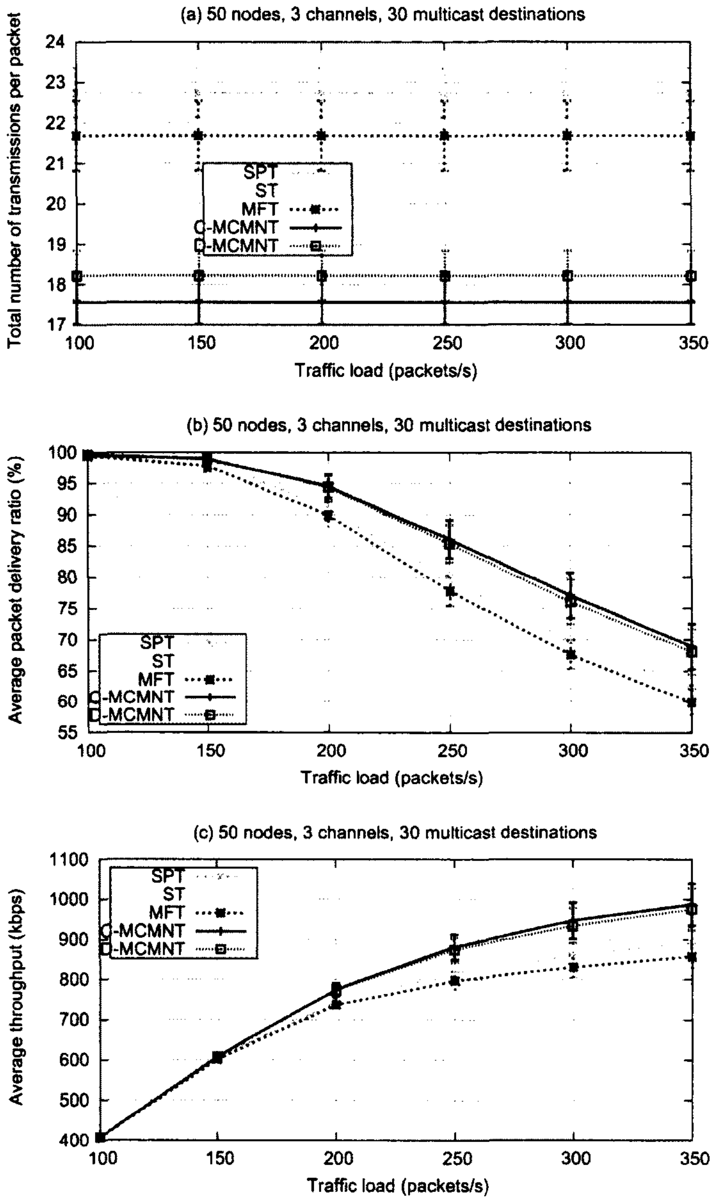
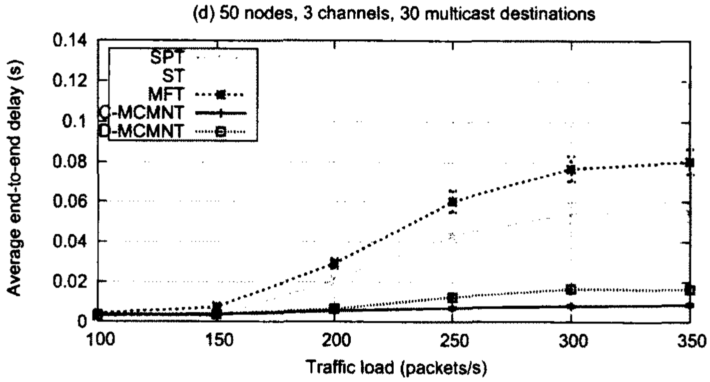


Figure 4.7: Functions of traffic load: 50 nodes



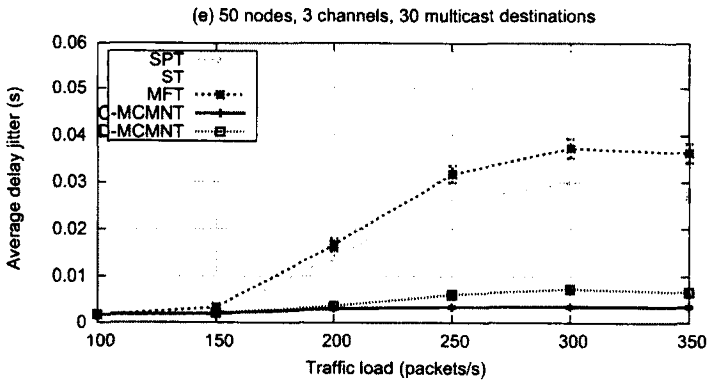


Figure 4.7: Functions of traffic load: 50 nodes (continued)

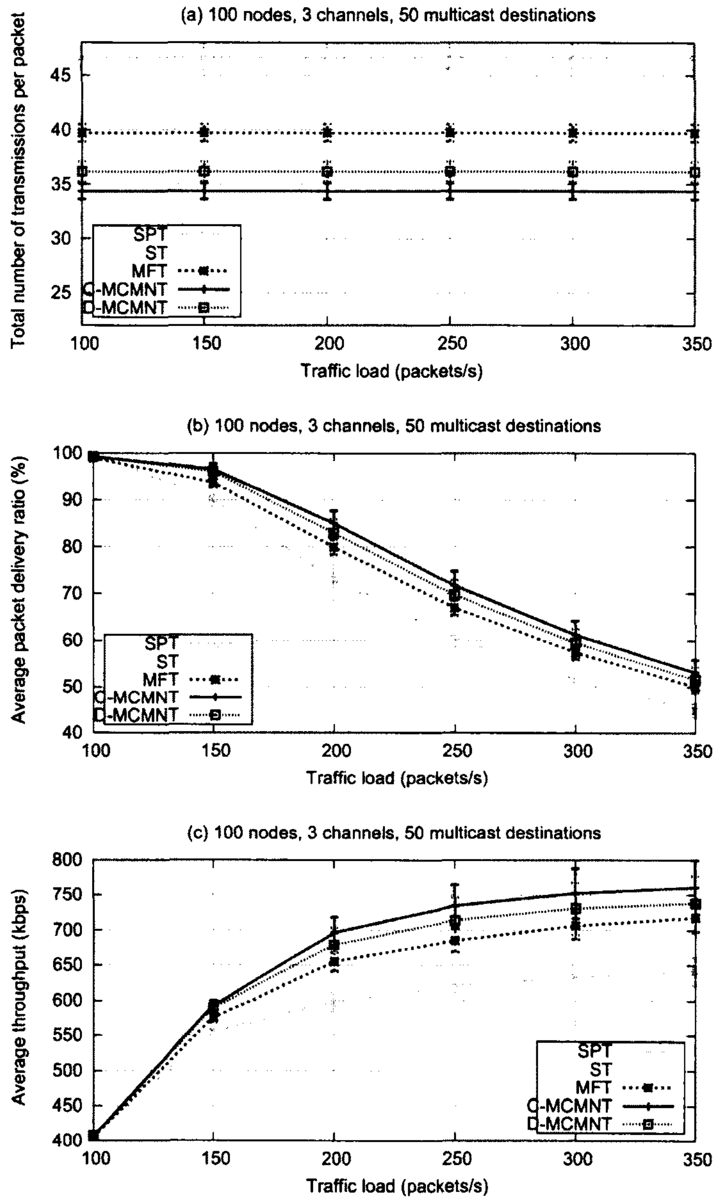


Figure 4.8: Functions of traffic load: 100 nodes

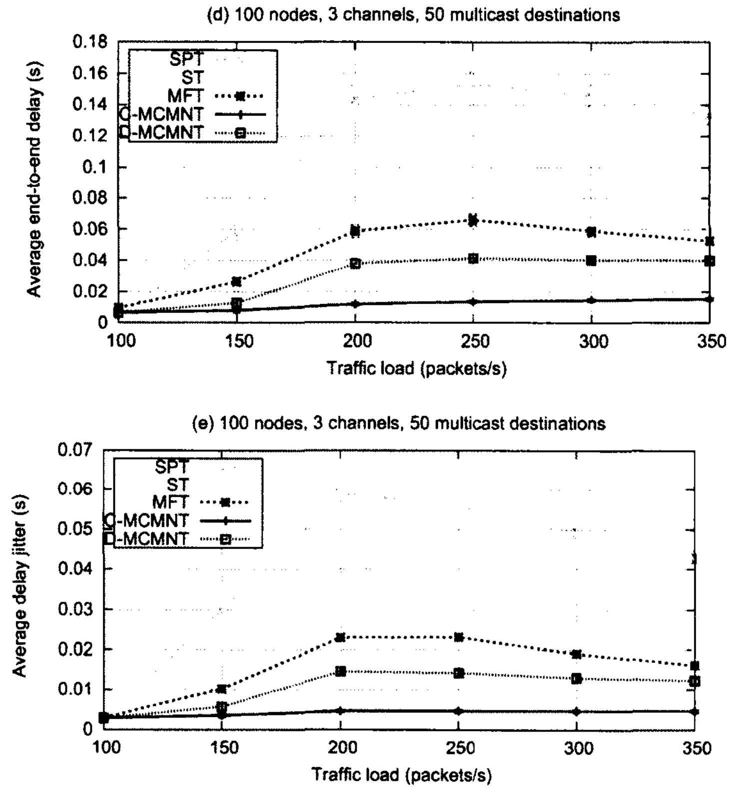


Figure 4.8: Functions of traffic load: 100 nodes (continued)

trees, then by the other trees. Note that the NTP values of a type of tree do not vary much as the traffic load increases, because the traffic load does not affect the multicast routing tree.

At low traffic loads of less than 150 packets/s, the PDRs of all the trees are high (above 90%) and similar (see Figures 4.7(b) and 4.8(b)). As the traffic load increases, the PDRs of all the trees start to decline, as expected. However, the MCMNT trees maintain the lead with PDRs about 5-10% higher than those of the MFTs, SPTs and STs. The PDRs of the D-MCMNT trees are very close to those of the C-MCMNT trees, with a difference of only about 1%, proving the effectiveness of the distributed algorithm. The MCMNT trees also achieve higher throughputs, about 10% to 25% higher than the other trees in both networks. In addition, the MCMNT trees outperform the SPTs, STs, and MFTs in terms of end-to-end delay and delay jitter, thanks to lower numbers of transmissions, which result in less contention among nodes in the routing trees. For example, in the 50-node network at a traffic load of 250 packets/s, the end-to-end delay of the C-MCMNT tree is four to 10 times lower than those of the SPT, MFT and ST (Figure 4.7(d)). In the 100-node network with a source rate of 200 packets/s, the delay jitter of the C-MCMNT tree is six to 14 times less than those of the MFT, SPT and ST (Figure 4.8(e)). The explanations for the better performance of the MCMNT trees in the previous section apply to this set of experiments as well.

Finally, we observe that the MCMNT trees begin to outperform the other trees when the traffic load is over 200 packets/s in the 50-node network (Figure 4.7), and 150 packets/s in the 100-node network (Figure 4.8). This implies that the MCMNT algorithms are particularly effective and useful in either highly loaded or large networks, or both.

4.4.3 Function of Number of Channels

In this set of experiments, we vary the number of orthogonal channels from one to seven. In general, increasing the number of channels improves the average PDRs, throughputs, end-to-end delays and delay jitters of all trees, as illustrated by the graphs in Figures 4.9 and 4.10. Note that as the number of channels increases, the number of transmissions also goes up because the single-channel broadcast advantage is reduced due to the increased channel diversity (Figures 4.9(a) and 4.10(a)). More transmissions *in this case*, however, do not necessarily imply performance degradation, because the loads are distributed over different channels and parallel transmissions can be used. That explains the improved performance as the number of channels increases.

When only one channel is available, the MFT has the least number of transmissions since the MFT algorithm is optimized for single-channel networks. As a result, the MFT provides the best PDR, throughput, end-to-end delay, and delay jitter

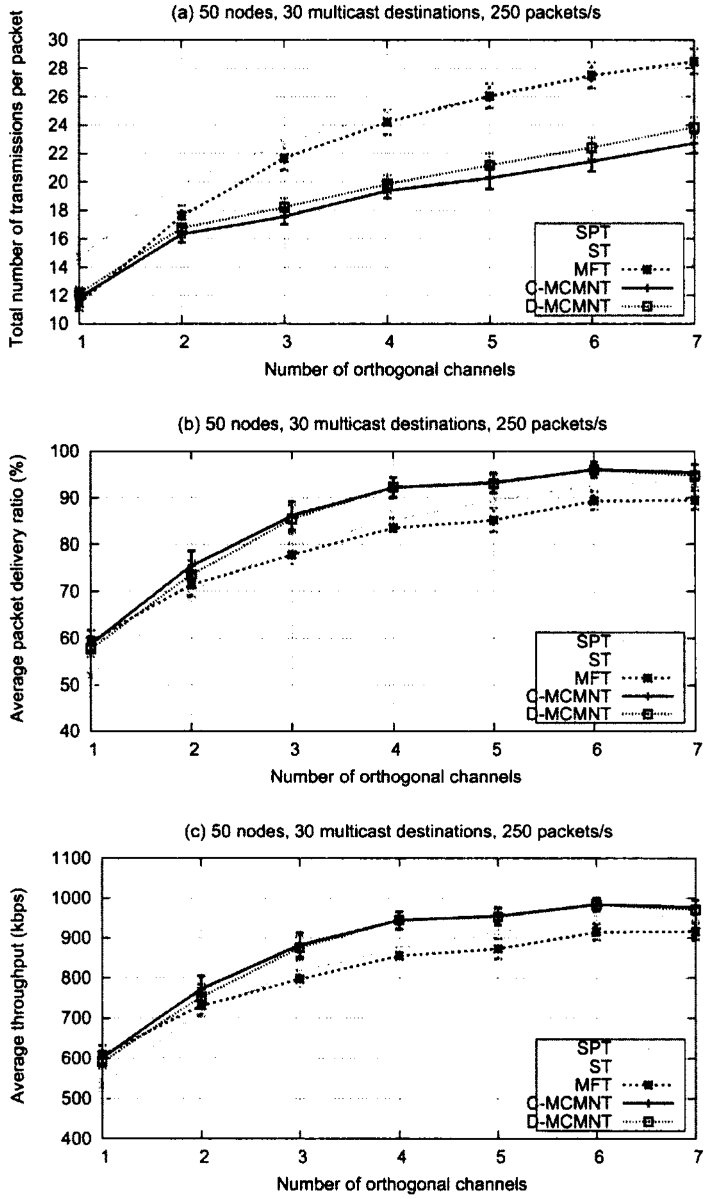


Figure 4.9: Functions of channels: 50 nodes

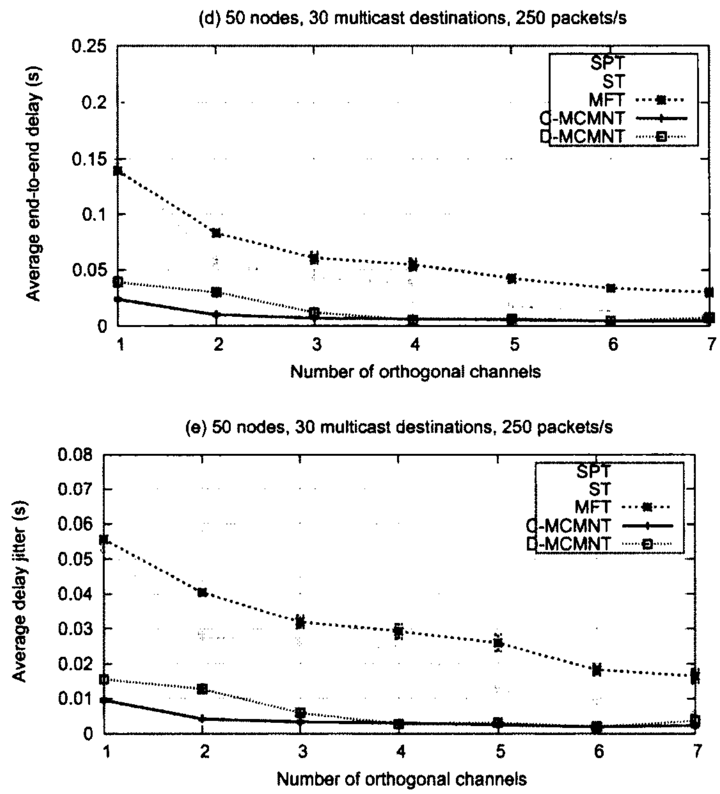
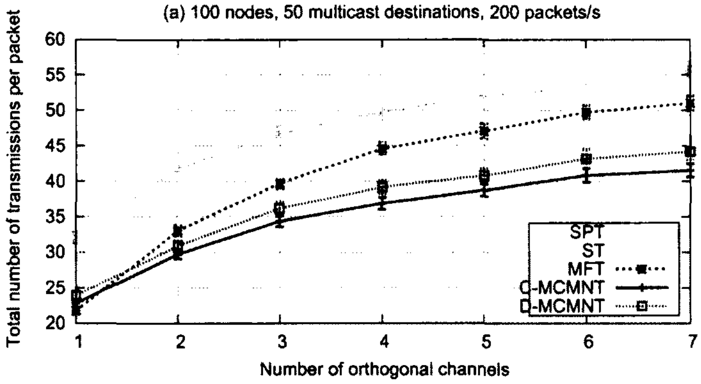
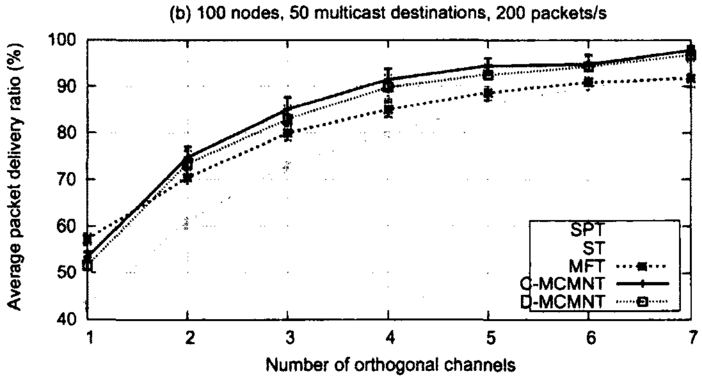


Figure 4.9: Functions of channels: 50 nodes (continued)





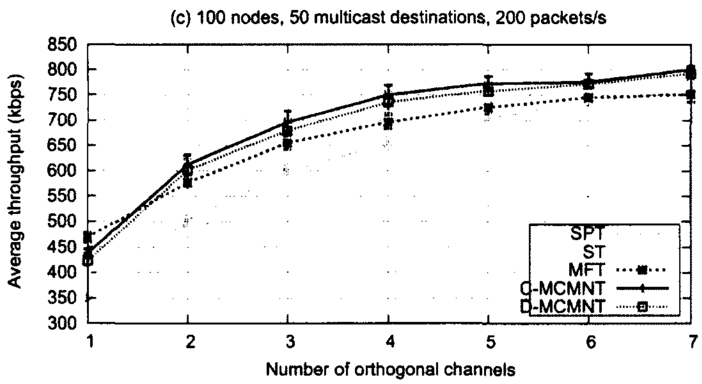


Figure 4.10: Functions of channels: 100 nodes

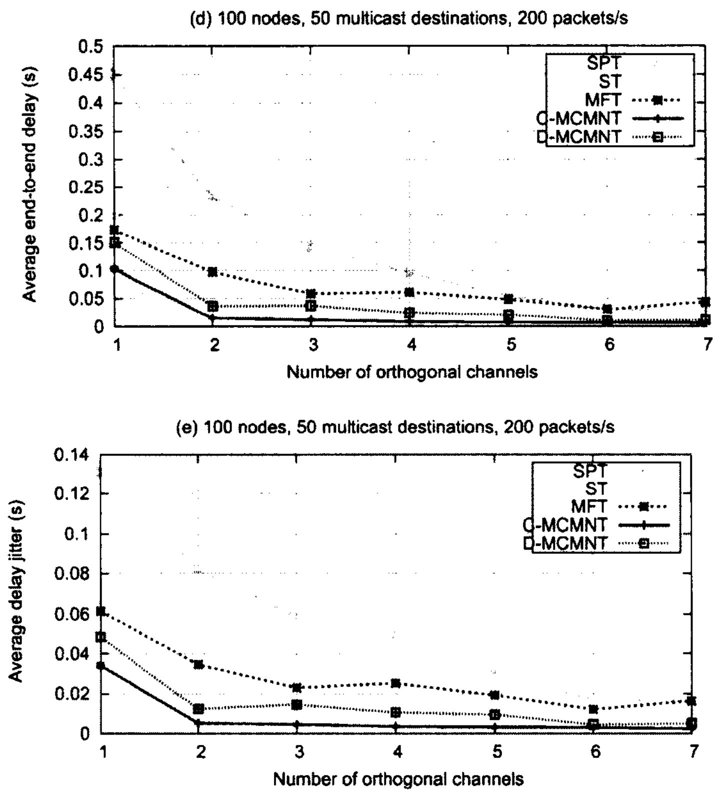


Figure 4.10: Functions of channels: 100 nodes (continued)

in this special case. The MCMNT algorithms are not optimized for single-channel networks as there is no channel diversity in such environments.

When multiple channels are used, the MCMNT algorithms produce trees with the least numbers of transmissions (Figures 4.9(a) and 4.10(a)) and, consequently, the highest PDRs and throughputs, as well as the lowest end-to-end delays, and delay jitters (Figures 4.9(b)-(e) and 4.10(b)-(e)). We also observe that the performance gap between the MCMNT trees and the other trees narrows as the number of channels is close to seven. Having such a high number of channels significantly reduces interference, transmission contention and packet collision in the whole network, making the performance of the MCMNT trees less dominant. Nevertheless, the MCMNT trees still offer noticeably better performance than the other trees.

4.4.4 Function of Number of Multicast Sources

When the number of multicast sources increases from one to four, the total number of multicast transmissions also increases (Figures 4.11(a) and 4.12(a)). Compared to other trees, multicast trees built by MCMNT have the least number of transmissions, as expected. Increasing the number of sources in the network also increases the traffic loads, leading to more interference, higher number of collisions and longer delays due to channel contention, and consequentially lowering the PDRs and throughputs of all the trees. However, the PDRs and throughputs of MCMNT trees are higher than those of SPT, ST and MFT trees, as shown in Figures 4.11(b), 4.11(c) and Figures 4.12(b), 4.12(c). With respect to average end-to-end delays and jitters, we observe similar behaviours, with the MCMNT trees having the lowest end-to-end delays and jitters in both small and medium-size networks (Figures 4.11(d), 4.11(e) and Figures 4.12(d), 4.12(e)).

4.5 Chapter Summary

In this chapter, we defined the MCMNT problem of minimizing the number of transmissions per packet consumed by a multicast routing tree in MCMR WMNs. We proved that the problem is NP-hard and then proposed approximate solutions. Our solutions include both centralized and distributed implementations with analyses of their running time complexities. We showed that multicast routing in MCMR WMNs should consider not only the wireless broadcast advantage but also the underlying channel assignment in order to minimize bandwidth consumption. Our experimental results showed that the proposed MCMNT algorithms perform significantly better than traditional multicast trees such as SPTs, STs and MFTs in terms of packet delivery ratio, throughput, end-to-end delay, and delay jitter.

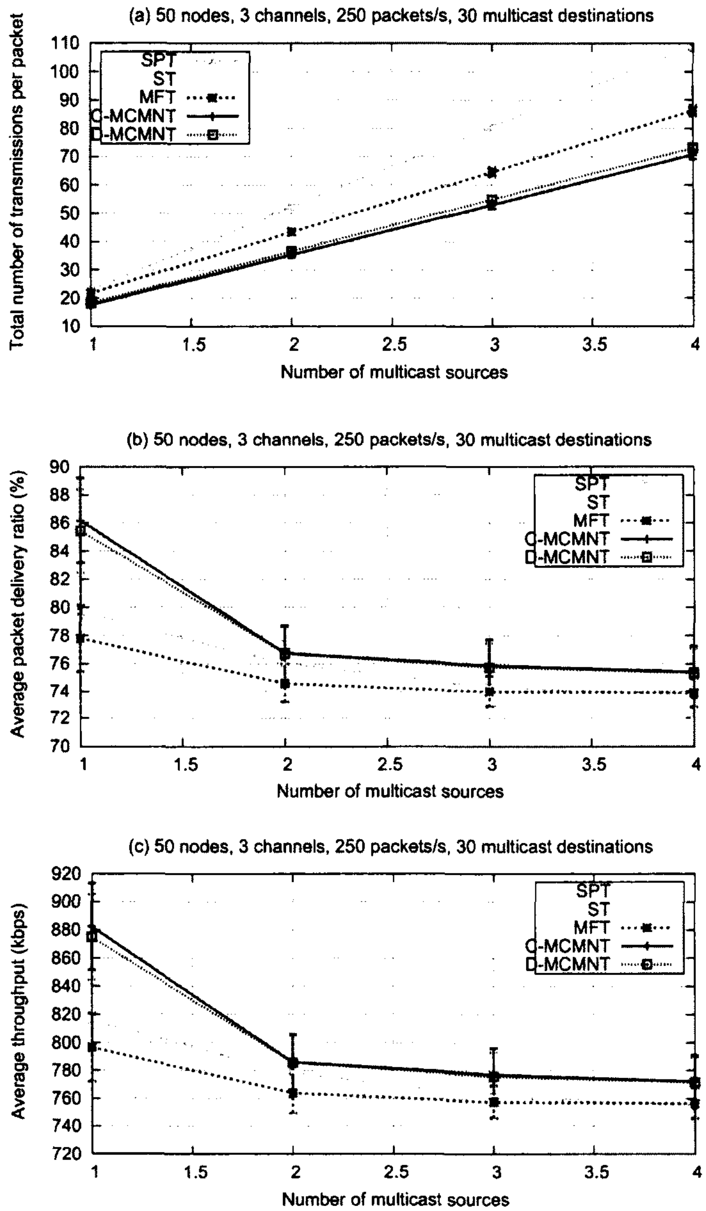
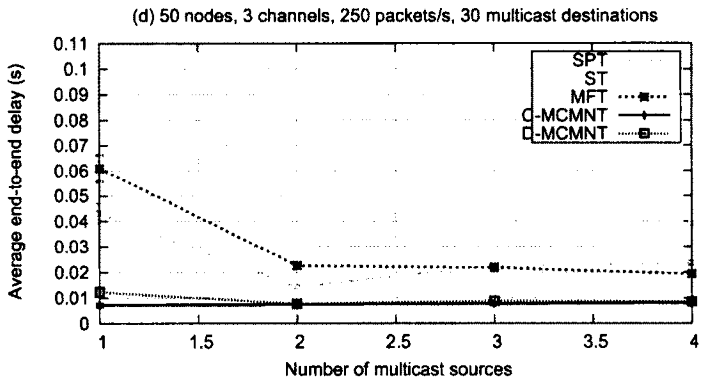


Figure 4.11: Functions of multicast sources: 50 nodes



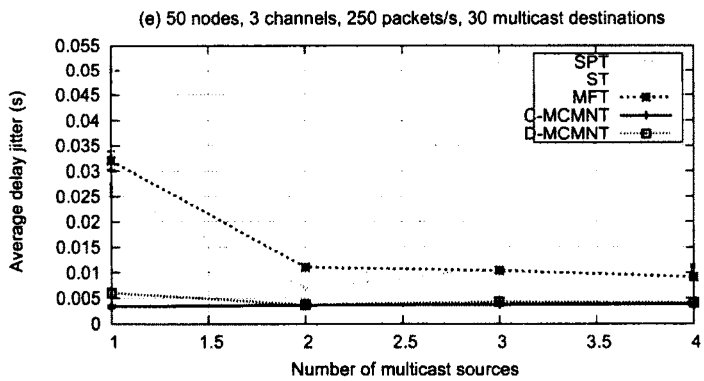


Figure 4.11: Functions of multicast sources: 50 nodes (continued)

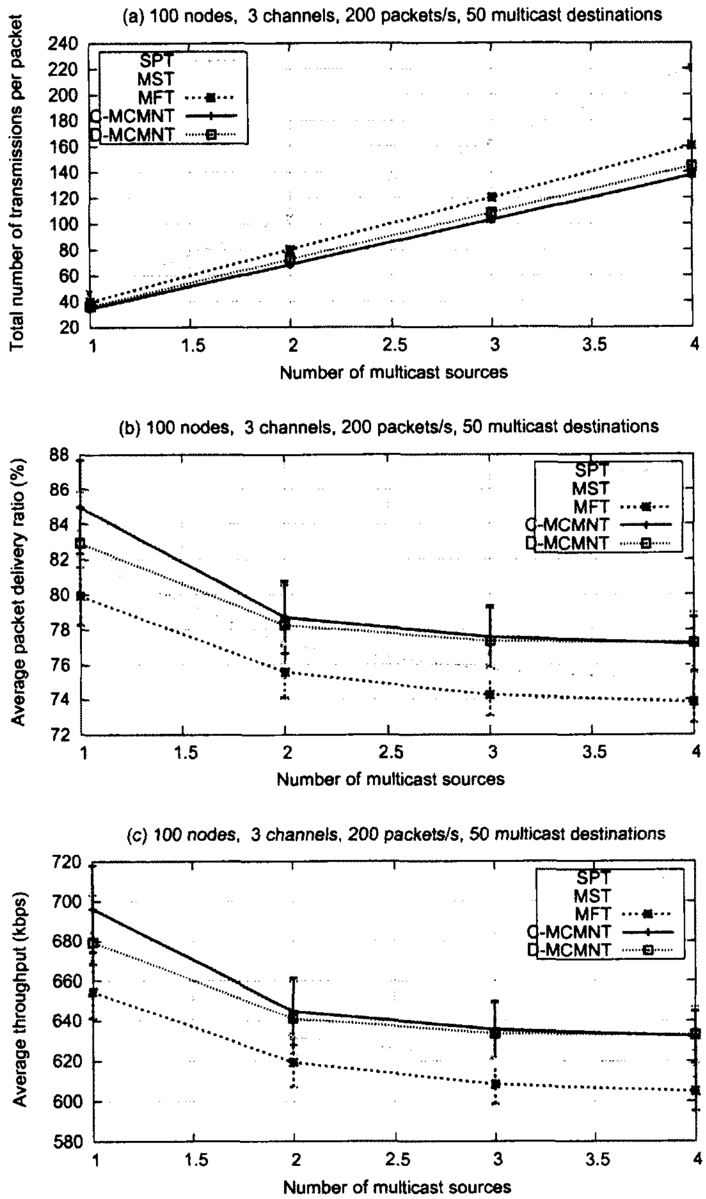
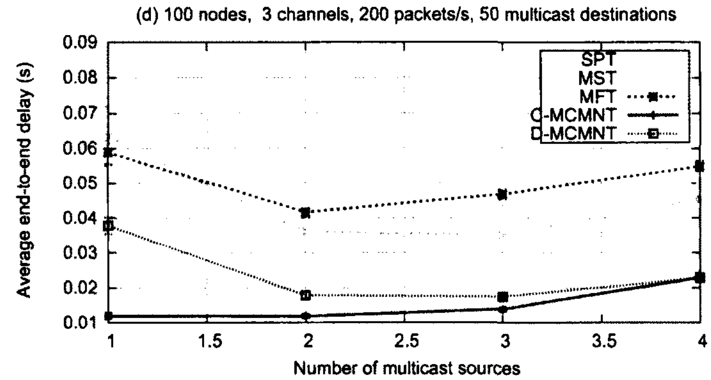


Figure 4.12: Functions of multicast sources: 100 nodes



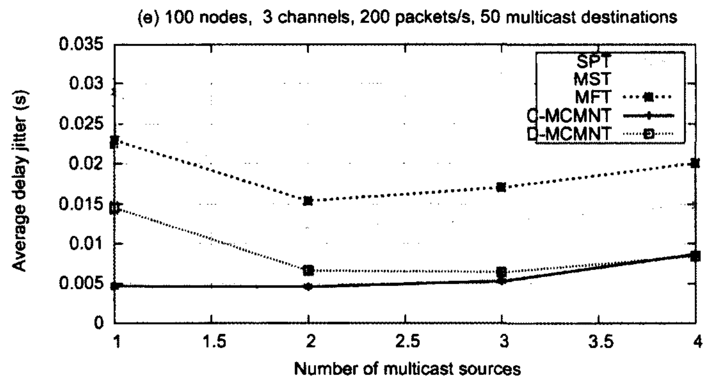


Figure 4.12: Functions of multicast sources: 100 nodes (continued)

Chapter 5

Network-Coded Multicast in MCMR WMNs

We propose analytical models to estimate the performance of a multicast session with and without network coding in MCMR WMNs. We then present a simulation-based performance evaluation of network-coded multicast in MCMR WMNs, using realistic network scenarios and useful performance metrics such as throughput, packet end-to-end delay, and packet delivery ratio. The simulation results are also used to validate the proposed analytical models. Based on the analytical and simulation results, we analyze performance gains of network-coded multicast in MCMR WMNs. To the best of our knowledge, our work is the first that studies the performance of multicast with network coding in MCMR WMNs.

We use the following system model in our analytical models and simulation experiments.

5.1 System Model

Unless otherwise stated, the following system model and assumptions are used in both the analytical modeling and simulations.

5.1.1 Multicast Communication Model

We assume that each multicast source or destination is associated with a different wireless mesh router. That is, a multicast group with j destinations consists of j distinct destination routers and one source router, since we are interested in the multicast performance of routers in the mesh backbone.

Multicast packets are delivered to multicast destinations in a multi-hop manner by following the multicast structure constructed by a multicast routing protocol [30, 80, 81]. Although the operations of different multicast protocols vary from one protocol to another, in general, the multicast structure of a multicast protocol is often defined by its *forwarding set*, a group of nodes designated to forward multicast packets. A multicast source by default is part of this set. Similar to the Internet multicast model, each multicast group is identified by the source ID s and group ID G. Every node maintains a routing entry of the form $[(s, G), \gamma]$, where γ is a boolean variable called *forwarding flag*. Nodes that belong to the forwarding set of the multicast structure have their forwarding flags set to true and, upon receiving non-duplicate multicast packets for the multicast group (s, G), re-broadcast them.

5.1.2 The 802.11 MAC Model for Multicast

The medium access control (MAC) for multicast uses the basic access procedure of the IEEE 802.11 distributed coordination function (DCF) with carrier sense multiple access and collision avoidance (CSMA/CA) without RTS (request-to-send), CTS (clear-to-send) and ACK (acknowledgment) [2]. At the MAC layer, multicast packets are neither acknowledged nor retransmitted if being lost. The 802.11 standard currently does not implement the RTS/CTS/ACK mechanism for multicast due to the following reasons. First, multiple CTS/ACK packets concurrently sent by multicast receivers of a transmitter have a very high probability of colliding at the transmitter. More importantly, it may not be possible for all the multicast receivers to agree on a common time slot for the transmission of a packet, and the delay would be very long to either reach a transmission time agreement or receive ACKs from all the receivers. Although there exist some works in the literature that propose RTS/CTS mechanisms for multicast, they either incur very long delay (e.g., by polling multicast receivers one by one) [44, 45] or require extensive modifications to the 802.11 MAC protocol [46, 48, 49].

According to the basic access procedure of the 802.11 DCF CSMA/CA protocol, a multicast node with a new packet to transmit monitors the channel activity until it is measured idle for an interval of distributed interframe space (DIFS). At this point, the node generates a random backoff timer before transmitting (this is the "Collision Avoidance" feature of the protocol), to minimize the probability of collision with packets being transmitted by other nodes. A backoff time is randomly selected in the range $[0, W-1]$, where W is the *minimum contention window*. The contention window in the backoff scheme for multicast does not increase due to the absence of the ACK packets[1]. The backoff time counter is decremented as long as the channel is sensed idle, paused when a transmission is detected on the channel, and resumed when the channel is sensed idle again for a DIFS interval. The node transmits when the backoff counter reaches zero. DCF employs a discrete-time backoff scale, meaning the backoff time following a DIFS interval is slotted. The contention window denotes the number of slots a node should wait before transmitting. The values of a *slot_time*, DIFS and W depend on the physical layer settings. For example, for the Direct Sequence Spread Spectrum (DSSS) used in the 802.11b standard, the *slot_time*, DIFS and W are set to 20 μs, 50 μs, and 32 (*slots*), respectively. We also assume the packet size does not exceed the maximum size allowed by the physical layer so that the packet can be transmitted in one

[1]In unicast communication, after each unsuccessful transmission, indicated by whether or not an ACK is received after the transmission, the contention window is doubled, up to a maximum value [2].

transmission with no fragmentation.

5.1.3 Intra-Flow Network Coding Model

With respect to network coding, we use the *intra-flow* random linear coding [99] to combine packets within a single multicast flow.

5.1.3.1 Multicast Source

When a multicast source with network coding function has a file to deliver to its multicast group, it breaks up the file into a set of ω batches, each having K packets. (In theory, the whole file could be treated as a batch, but the encoding/decoding time at the multicast forwarders would be very high if the file size is large.) To simplify the analysis, we assume $\omega = 1$ in our analytical models, i.e., the file consists of one batch; in other words, the file is simply divided into K packets. These K uncoded packets are called *native packets* and K is called the *batch size.* When the source is ready to send, it creates a random linear combination of the K native packets and broadcasts the coded packet. A coded packet x' is computed as:

$$x' = \sum_{i=1}^{K} c_i x_i,$$

where c_i values are random coefficients chosen from a finite field of size q, and x_i entities are native packets.

5.1.3.2 Multicast Forwarders

Nodes listen to all transmissions within its sensing range. When a node overhears a multicast packet, it checks whether it is a multicast forwarder (by looking up the routing entry of the multicast group for the forwarding flag value). If so, the node checks whether the packet is an *innovative packet.* A packet is innovative if it is linearly independent from the packets the node has previously received. Checking for linear independence can be done using Gaussian elimination [100]. The node keeps innovative packets and drops non-innovative ones.

Upon receiving an innovative packet, the forwarding node creates a new coded packet by generating a random linear combination of the innovative coded packets it has received, and broadcasts it. Note that a linear combination of coded packets is also a linear combination of the corresponding native packets. In particular, suppose that the forwarder has received m coded packets, each in the form of:

$$x'_j = \sum_{i=1}^{K} c_{ji} x_i,$$

where x_i is a native packet. The forwarder then linearly combines these coded packets to create a new coded packet as follows:

$$x'' = \sum_{j=1}^{m} a_j x'_j,$$

where a_j's are new random coefficients. The resulting coded packet x'' can be expressed in terms of the native packets as follows:

$$x'' = \sum_{j=1}^{m} a_j (\sum_{i=1}^{K} c_{ji} x_i) = \sum_{i=1}^{K} (\sum_{j=1}^{m} a_j c_{ji}) x_i,$$

thus, it is also a linear combination of the native packets.

5.1.3.3 Multicast Destinations

Upon receiving a packet, a multicast destination checks whether the packet is innovative and discards the packet if it is not. Once the destination receives K innovative packets, it can decode the batch and obtain the native packets using a simple matrix inversion:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{K1} & \cdots & c_{KK} \end{pmatrix}^{-1} \begin{pmatrix} x'_1 \\ \vdots \\ x'_K \end{pmatrix},$$

where x'_i is a coded packet whose coefficients are $c_{i1}, ..., c_{iK}$.

5.1.4 Multi-Channel Multi-Radio Systems

We consider multi-channel wireless mesh networks with multiple radios per node. Two nodes u and v are directly connected and form a communication link (u, v) if they are within the transmission range of each other and share a common channel. Each node is equipped the same number of radios r and the network has C

orthogonal (non-overlapping) channels. We assume that the network uses a channel assignment (CA) scheme that ensures that, at any point in time, the number of distinct channels assigned to any node is less than or equal to the number of radios the node possesses. As a result, each radio is bound to a specific distinct channel and no channel switching is needed. There exist many such CA algorithms [7, 8, 9, 10, 11, 12, 13] that produce results satisfying the above condition.

5.1.5 Queueing Model

We model a router with r radios and queue capacity of Q using the $M/M/r/Q$ queue [101] wherein the interarrival time and the service time are exponentially distributed. Each router is assumed to have r independent servers as each of the r radios can process and broadcast packets in parallel over different channels. All multicast packets waiting for service and those being served are kept in one multicast queue of capacity Q, meaning the queue can hold up to a maximum of Q packets (we assume that each router maintains a separate queue for multicast flows). When the queue is full, all packet arrivals are dropped. We assume that Q is greater than or equal to r; otherwise some radios would fail to operate due to lack of queueing buffers [101]. The service policy is first in first out (FIFO) and each server has a mean service rate of μ, which will be derived later for regular and network-coded systems in Sections 5.2.2.1 and 5.2.3.1, respectively. We assume that if the multicast source sends data packets at a constant bit rate of λ, then the average packet arrival rate at a multicast router is approximately λ.

Let p_m be the probability that there are m packets in a router, including packets being processed and those waiting to be processed. According to the $M/M/r/Q$ model, we have

$$p_0 = \left[1 + \frac{(1-\rho^{Q-r+1})(r\rho)^r}{r!(1-\rho)} + \sum_{i=1}^{r-1} \frac{(r\rho)^i}{i!}\right]^{-1}, \tag{5.1}$$

and

$$p_m = \begin{cases} \frac{\rho^m r^m p_0}{m!} & m \in [1, r-1] \\ \frac{\rho^m r^r p_0}{r!} & m \in [r, Q], \end{cases} \tag{5.2}$$

where $\rho = \frac{\lambda}{(r\mu)}$ is the traffic utilization, p_0 represents the probability that the router is empty, and p_Q represents the probability that the router is full. Using the expressions for p_m in (5.1) and (5.2), we can determine the expected number of packets $E[m]$ in a router as follows:

$$E[m] = \sum_{m=1}^{Q} m p_m = p_0 \left[\sum_{m=1}^{r-1} \frac{\rho^m r^m}{m!} + \sum_{m=r}^{Q} \frac{\rho^m r^r}{r!}\right] \tag{5.3}$$

5.2 The Proposed Performance Models

In this section, we present our analytical models and provide closed-form expressions for estimating the performance of a multicast session with and without network coding in multi-channel multi-radio wireless mesh networks. Our analysis is divided into two parts. First, we analyze the performance of a multicast session without network coding, which we refer to as **Re**gular **M**ulticast (ReM). Previous work on performance analysis for ReM has been mostly experiment-based, protocol-specific, and focused on systems with a single channel. Here, we provide a theoretical analysis, which is independent of any multicast routing protocol, in MCMR networks. We then present a performance modeling of **Net**work-**Co**ded **M**ulticast (NetCoM) in MCMR networks, a problem that has not been addressed prior to this work.

In the analyses, we consider the following performance metrics: average end-to-end delay, average throughput and average packet delivery ratio, which are defined as follows.

- *Average end-to-end delay.* The end-to-end delay of a packet received at a multicast destination is defined as the latency between the time the packet is transmitted from the multicast source and the time the packet is received at the destination. The average end-to-end delay is the average of the end-to-end delays of all the packets received at all multicast destinations.

- *Average packet delivery ratio.* The packet delivery ratio (PDR) of a multicast destination is the ratio of the number of packets received by the destination and the number of packets sent by the multicast source. The average end-to-end PDR of a multicast group is the average of the PDRs of all multicast destinations in the group.

- *Average throughput.* The throughput of a multicast destination is defined as the total number of packets the destination receives divided by the interval starting from the time the multicast source begins transmitting the first packet to the time the destination receives its last packet. The average taken over the throughputs of all multicast destinations is the average throughput of the multicast group.

Although all the above performance metrics are equally important, we will begin with the modeling of the end-to-end delay as it plays an critical role in deriving the other performance models. To model such delay of a multicast session, we need to consider the following time components incurred by a packet, from the time it arrives at a node to the time it is forwarded to a neighbouring node:

- *processing time*: time to react to an incoming event. Since the processing time is typically negligible compared to the other time components, we ignore this latency in our analyses.

- *queueing time*: time the packet waits in the queue (buffer) before being considered and forwarded.

- *backoff time*: time incurred by 802.11 CSMA/CA algorithm.

- *transmission time*: time for the transmitter to send out all bits of the frame.
- *propagation time*: time for a bit to travel from the transmitter to the receiver.
- *coding opportunity delay*: time the packet has to wait before it can be combined with other packets of the same flow for coding [102]. The waiting is caused by packets from different flows interleaving in the buffer. We do not consider the coding opportunity delay in this paper because we assume an *intra-flow* network model where all packets buffered at a node belong to the same flow, as in [99, 103]. We will extend the models to *inter-flow* network coding in our future work.

We begin the analysis by estimating the average backoff time of multicast nodes at the MAC layer. We then present the performance models of ReM and NetCoM in Sections 5.2.2 and 5.2.3, respectively.

5.2.1 Estimating the Average Multicast Backoff Time

In this section, we model the average backoff time a node waits before transmitting a multicast packet. The model follows the 802.11 CSMA/CA protocol described in Section 5.1.2.

We assume that a channel is busy, i.e., some node is transmitting on the channel, with a constant and independent probability α. Then, it is possible to model the

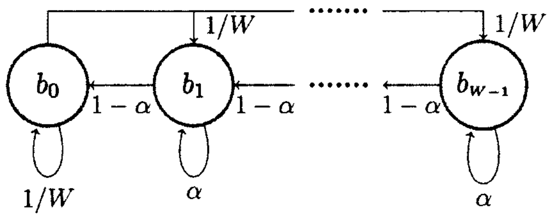


Figure 5.1: Markov chain model for backoff time at multicast node.

802.11 backoff scheme with a discrete-time Markov chain depicted in Figure 5.1. For a given multicast node, let b_k be the steady-state stationary distribution that the backoff time counter is equal to k, where $k \in [0, W-1]$, and $P\{i \rightarrow j\}$ be the one-step transition probability from state i to state j. In this Markov chain, the non-null one-step transition probabilities are as follows:

$$\begin{cases} P\{k \rightarrow k\} & = \alpha & k \in [1, W-1] \\ P\{k \rightarrow k-1\} & = 1-\alpha & k \in [1, W-1] \\ P\{0 \rightarrow k\} & = \frac{1}{W} & k \in [0, W-1] \end{cases} \tag{5.4}$$

The first equation in (5.4) models the fact that the backoff counter pauses when the node senses that the channel is busy. The second equation accounts for the fact that the backoff counter decrements when the node senses that the channel is idle. The third equation considers the fact that once the backoff counter reaches zero, the node can transmit its packet and starts a new backoff period for a new transmission. The new backoff time interval is again chosen randomly in the range

$[0, W-1]$, and thus having the probability $\frac{1}{W}$ of being in one of the W states, from 0 to $W-1$.

According to the chain regularities, we have

$$\begin{cases} b_1 & = \frac{b_0}{W} + \alpha b_1 + (1-\alpha) b_2 \\ b_2 & = \frac{b_0}{W} + \alpha b_2 + (1-\alpha) b_3 \\ \vdots & \vdots \\ b_{W-2} & = \frac{b_0}{W} + \alpha b_{W-2} + (1-\alpha) b_{W-1} \\ b_{W-1} & = \frac{b_0}{W} + \alpha b_{W-1} \end{cases} \tag{5.5}$$

If we rewrite (5.5) and express b_k values as functions of b_0, we get

$$b_k = \frac{W-k}{W} \times \frac{1}{1-\alpha} b_0, \qquad k \in [1, W-1] \tag{5.6}$$

The value b_0 is determined by imposing the normalization condition that the sum of all the state probabilities must be one and by substituting b_k values with $k \in [1, W-1]$ using (5.6):

$$1 = \sum_{k=0}^{W-1} b_k = b_0 + \sum_{k=1}^{W-1} \frac{W-k}{W} \times \frac{1}{1-\alpha} b_0 \tag{5.7}$$

From (5.7), we obtain:

$$b_0 = \frac{2(1-\alpha)}{W - 2\alpha + 1} \tag{5.8}$$

Since the equation for b_0 in (5.8) still has the variable α whose value is yet unknown, we now show how to obtain α and b_0. As a transmission occurs when the backoff counter equals zero, b_0 is also the probability that a node transmits at a randomly chosen time. Hence, the probability that a node does not transmit is $(1 - b_0)$. Let $n \geq 1$ denote the number of nodes in the network. To simplify the analysis, we assume that nodes are within the interference range of each other. In case of a single-channel network, n nodes contend for the sole channel; the probability that all n nodes do not transmit (i.e., the channel is idle) is thus $(1 - b_0)^n$. We are, however, interested in networks with multiple channels. Given C as the number of non-overlapping channels available in a multi-channel network, n nodes can be grouped into C autonomous regions, each of which does not interfere with the others and consists of $\frac{n}{C}$ nodes. Therefore, the probability that a channel is idle is $(1 - b_0)^{\frac{n}{C}}$. Assuming that all C channels are fully utilized, there should always be at least one node contending for a channel, even when $n < C$. (In practice, $n > C$.) We thus rewrite the probability that a channel is idle as $(1 - b_0)^{\max\{\frac{n}{C},1\}}$. (The single-channel equation is a special case of the multi-channel equation in which $C = 1$.) The probability α that a channel is busy in a multi-channel network is then:

$$\alpha = 1 - (1 - b_0)^{\max\{\frac{n}{C},1\}} \tag{5.9}$$

Using (5.8) and (5.9), we obtain the following polynomial equation of degree $(\max\{\frac{n}{C}, 1\}+1)$ with $(1 - b_0)$ being the unknown:

$$2(1 - b_0)^{\max\{\frac{n}{C},1\}+1} + (W - 1)(1 - b_0) - (W - 1) = 0 \tag{5.10}$$

Given known values for constants n, C and W, equation (5.10) can be solved for b_0 using numerical techniques or mathematical tools such as MATLAB, after $\frac{n}{C}$ is rounded to the nearest integer.

The average backoff time depends on the value of the backoff counter and the duration for which the counter pauses when the node detects transmissions from other nodes [104]. Let us first consider the average backoff time interval $E[k]$ in terms of number of slots, *without* taking into account the duration for which the counter is paused. This interval is given by: $E[k] = \sum_{k=1}^{W-1} kb_k$, i.e., when the backoff counter is at state b_k, a time interval of k slots is needed for the counter to reach zero. Substituting (5.6), (5.8), (5.9) into $E[k]$, we get

$$E[k] = \frac{(W+1)}{3}(1 - b_0) \tag{5.11}$$

Next, let us denote φ as the average duration for which the backoff counter remains paused. Since the mean number of consecutive idle slots the backoff counter decrements before it pauses due to the channel being busy is $\frac{1-\alpha}{\alpha}$, the average number of times the counter pauses during the $E[k]$ backoff period before it reaches zero is $(\frac{E[k]}{\max\{\frac{1-\alpha}{\alpha},1\}} - 1)$. The max function ensures that there should be at least one idle slot in every busy slot; otherwise the backoff counter would never reach zero. Given that once the counter pauses, it remains paused for the duration of a packet transmission, which is equal to $\frac{S}{B}$, where S is the packet size and B is the channel bandwidth at the physical layer, plus a period of DIFS, the total pause time is

$$\varphi = \left(\frac{E[k]}{\max\{\frac{1-\alpha}{\alpha}, 1\}} - 1 \right) \times \left(\frac{S}{B} + \text{DIFS}\right) \tag{5.12}$$

The average backoff time β is the sum of the initial DIFS period, the backoff time without pausing $E[k]$, and the pause duration φ.

$$\beta = \text{DIFS} + E[k] \times \mathit{slot\_time} + \varphi \tag{5.13}$$

Since $E[k]$ in (5.11) is expressed in terms of number of slots, to convert it to a proper time unit (e.g., microsecond), we multiply it by the *slot_time*.

5.2.2 Regular Multicast Performance Modeling

We now present our models for estimating the average end-to-end delay, packet delivery ratio and throughput of a regular multicast session without network coding.

5.2.2.1 Average End-to-End Delay

The average service time a node takes to serve a packet is the sum of the processing time, which is negligible; the average backoff time β given by (5.13); and the transmissions time $\frac{S}{B}$. The average service rate μ at a regular multicast router is thus

$$\mu = \frac{1}{(\beta + \frac{S}{B})} \tag{5.14}$$

Let L denote the average node latency experienced by a packet at a node u, which is the period from the time the packet enters u to the time u completely transmits the packet back into the network. More specifically, L is the sum of the service time $(\beta + \frac{S}{B})$ and the time waiting in the queue at u. L can be determined by Little's law [101] as follows:

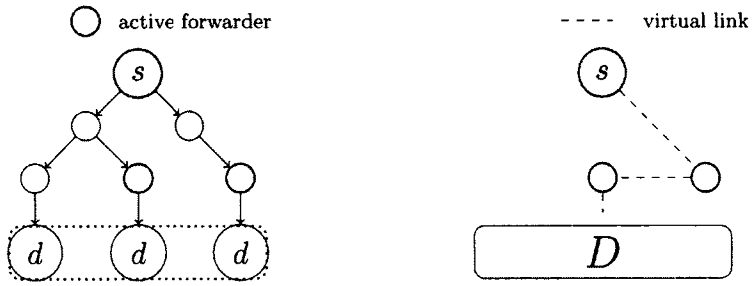
$$L = \frac{E[m]}{\lambda(1 - p_Q)}, \tag{5.15}$$

where λ is the packet arrival rate, p_Q is the probability that there are Q packets in u, and $E[m]$ is the mean number of packets in u, defined in (5.2) and (5.3), respectively. Since packets are dropped when the node is full, i.e., when there are Q packets, p_Q can be considered as the dropping probability and $(1 - p_Q)$ as the absorbing probability. The term $\lambda(1 - p_Q)$ is hence also known as the effective arrival rate of node u. From the value of μ given by (14), we compute the traffic utilization $\rho = \frac{\lambda}{(r\mu)}$, which then allows us to compute p_Q and $E[m]$ using (1), (2) and (3).

Consider a link (u, v) with u being the transmitter and v, the receiver. The sum of the average node latency L and the propagation time, which can be estimated as the transmission range R divided by the speed of light c, is the average point-to-point delay δ it takes for node u to deliver a packet to node v:

$$\delta = L + \frac{R}{c} \tag{5.16}$$

For a multicast session, let $|F|$ denote the number of forwarding nodes needed in the multicast structure (e.g., a tree or mesh) to forward data packets to the multicast destinations. We note that the shortest possible path length from the multicast source s to a multicast destination d, in terms of number of hops, is one. The shortest end-to-end delay would thus be the point-to-point delay δ. On the other hand, in the worst-case scenario where the multicast structure is a straight line, the longest possible source-to-destination path is $|F|$ hops long. The longest end-to-end delay would thus be $(|F| \times \delta)$. To derive the average end-to-end delay of a group of multicast destinations with different distances from the source, we approximate the average path length of the multicast group as follows. We model the multicast group at an instance in time as one *super-destination* D, a concept introduced in [105] and commonly seen in performance analysis for representing a group of entities [106, 107, 108]. Let ξ, $0 \leq \xi \leq |F|$, denote the number of active forwarders whose queues are not empty. At any given time t, only non-empty forwarders are active to forward packets to the multicast group or, more specifically, to the super-destination D. Therefore, at time t, the multicast structure can be viewed as a virtual path connecting ξ forwarders to D, and thus D can be said to be ξ hops away from the source, as illustrated in Figure 5.2. The average number of hops from the source to D can thus be approximated by the expected number of non-empty forwarders $E[\xi]$ at any moment in time. As a result, an average of



(a) Multicast tree at time t (b) A virtual path at time t

Figure 5.2: Virtual path from s to D.

$E[\xi]$ transmissions are needed to deliver a packet. In a single-channel network, this would result in an average end-to-end delay of $(E[\xi] \times \delta)$, as each transmission requires a point-to-point delay of δ. However, in a multi-channel network with C channels, $E[\xi]$ transmissions can be parallelized over the C channels, potentially reducing the average end-to-end delay Δ to:

$$\Delta = \max\{\frac{E[\xi]}{C}, 1\} \times \delta \tag{5.17}$$

The max function ensures that the average end-to-end delay Δ should never be less than the average point-to-point delay δ, as any packet arrived at a destination must have traversed over at least one link.

We now show how to derive $E[\xi]$. Given in (5.1) the probability p_0 that a multicast queue is empty, the probability that the queue of a multicast forwarder is non-empty is $(1 - p_0)$. Hence, among all $|F|$ forwarders, the expected number of forwarders with non-empty queues is

$$E[\xi] = (1 - p_0)|F| \tag{5.18}$$

In our model, $E[\xi]$ is considered as the *effective number of forwarders* of a multicast group and the set of effective forwarders is called the effective forwarding set. Using (5.3), (5.15), (5.16), (5.18), we can rewrite the average end-to-end delay Δ in (5.17) as follows:

$$\Delta = \max\left\{\frac{(1-p_0)|F|}{C}, 1\right\} \times \left[\frac{\sum_{m=1}^{Q} m p_m}{\lambda(1-p_Q)} + \frac{R}{c}\right], \tag{5.19}$$

where the p_m values are computed using (5.1), (5.2) and variable μ derived in (5.14). Other variables such as $|F|$, C, R, λ are known parameters whose values depend on the network settings. Note that the number of forwarding nodes $|F|$ can be obtained from the underlying routing multicast structure.

5.2.2.2 Average Packet Delivery Radio

We assume a packet transmission over a link (u, v) from node u to node v fails if either the transmission collides with other transmissions from other nodes, or the packet is dropped by v because the queue at v is full. Suppose that there are n nodes in a single-channel network. The probability that a transmitted packet from u encounters a collision would be approximately the probability that at least one of the $(n-1)$ remaining nodes transmits. Given a network with C channels, if we group n nodes into C autonomous, non-overlapping regions, each of which does not interfere with the others and consists of $\frac{n}{C}$ nodes, then the collision probability in the network can be approximated as the probability that at least one of $(\frac{n}{C}-1)$ nodes transmits. As the probability that a node transmits is b_0, the probability that collision occurs is

$$P[\text{collision}] = (1-(1-b_0)^{\max\{\frac{n}{C},1\}-1}) \tag{5.20}$$

The max function ensures that there should be at least one node per region regardless of how large C is.

Since p_Q is the probability that v is full, it is the packet dropping probability at v. The probability that v is not full and no collision occurs is

$$P[\text{not full, no collision}] = (1-p_Q)(1-P[\text{collision}]) \tag{5.21}$$

Then, the link error probability ε that a packet transmission over a link (u,v) fails due to either a collision or a full queue is:

$$\varepsilon = 1 - P[\text{not full, no collision}] \tag{5.22}$$

We again use the "super-destination" concept introduced in Section 5.2.2.1 to approximate the average packet delivery ratio Ω of a multicast group. In particular, since the super-destination D is $E[\xi]$ hops away from the source, and each hop experiences the link error probability ε, the probability that a packet is successfully delivered to the multicast group is

$$\Omega = (1-\varepsilon)^{E[\xi]} = (1-\varepsilon)^{(1-p_0)|F|} \tag{5.23}$$

5.2.2.3 Average Throughput

We begin by estimating the throughput Γ_d of an arbitrary multicast destination d. It is defined as the total amount of multicast packets d received divided by the time it takes d to receive all this data.

The total number of packets d received can be estimated as the number of packets ℓ the source sent multiplied by the average packet delivery ratio Ω given in (5.23). The time it takes d to receive all the packets is the time it takes the source to transmit all the packets, which $\frac{\ell}{\lambda}$, plus the time it takes the last packet to travel from the source to d, which can be approximated by the average end-to-end delay Δ determined in (5.19). The throughput of d is thus

$$\Gamma_d = \frac{\ell \times \Omega}{\frac{\ell}{\lambda} + \Delta} \tag{5.24}$$

Since Γ_d is computed using the average packet delivery ratio and average end-to-end delay of the multicast group, it can be considered as an approximation of the average throughput Γ of the multicast group as defined at the beginning of Section 5.2. We also substitute the average packet delivery ratio Ω using (5.23) and obtain the average throughput Γ as follows:

$$\Gamma = \frac{\ell(1-\varepsilon)^{(1-p_O)|F|}}{\frac{\ell}{\lambda} + \Delta} \tag{5.25}$$

5.2.3 Network-Coded Multicast Performance Modeling

In this section, we present our closed-form expressions for estimating the average end-to-end delay, packet delivery ratio and throughput of a network-coded multicast session in MCMR WMNs. In the following analysis, unless otherwise stated, we use the prime symbol (′) to indicate a variable in NetCoM. For example, if μ is the average service rate of a regular node, μ' is the average service rate of a network-coded node.

5.2.3.1 Average End-to-End Delay

Compared to ReM, besides the overall backoff time β and the transmission time $\frac{S}{B}$, the average service time at a multicast forwarder in NetCoM has an additional element, the *coding time*. The coding time ϕ depends on the batch size K, which is the number of native packets coded in one batch. By measuring the coding time as a function of K, our empirical results (Figure 5.3) show that ϕ can be estimated as a quadratic polynomial function of K: $\phi = \sigma_2 K^2 + \sigma_1 K$, where σ_1 and σ_2 are polynomial coefficients and depend on the processing power of routers. Note that $\phi = 0$ when $K = 0$. From (5.14), the average service rate of a NetCoM forwarder

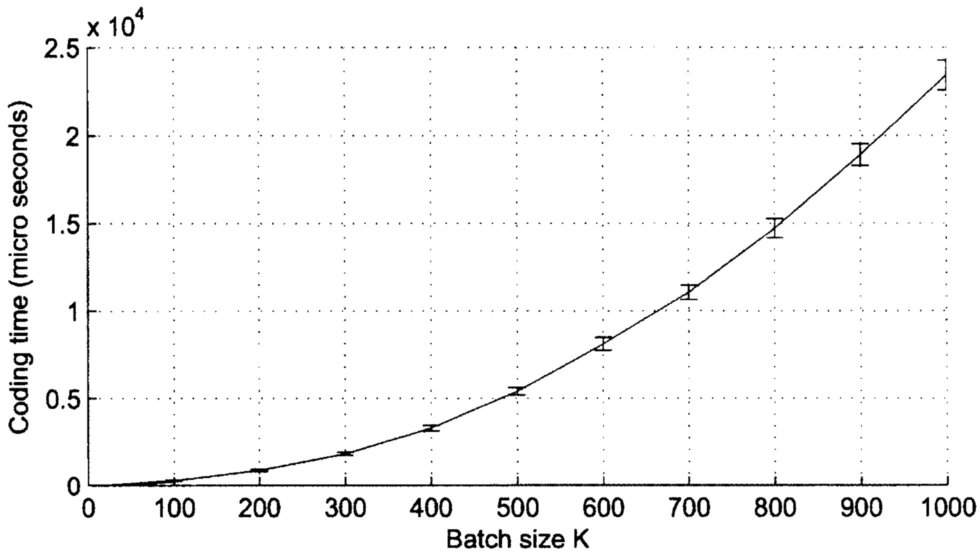


Figure 5.3: Coding time versus batch size K.

is then:

$$\mu' = \frac{1}{\beta + \frac{S}{B} + \phi} \tag{5.26}$$

Using the same analysis as in ReM, we derive the average point-to-point delay for a NetCoM forwarder as $\delta' = (L' + \frac{R}{c})$, where the average node latency L' of a NetCoM forwarder is computed using the NetCoM service rate μ' in (5.26).

We group the multicast destinations into a super-destination set D, as discussed in Sections 5.2.2.1 and 5.2.2.2 and, additionally, replace the set of forwarding nodes with a *super-forwarder* F [106, 107, 108] as shown in Figure 5.4. Although in practice, F and D may overlap (i.e., a multicast destination may act as a forwarding node) we assume that they are disjoint to simplify the analysis. Given that there

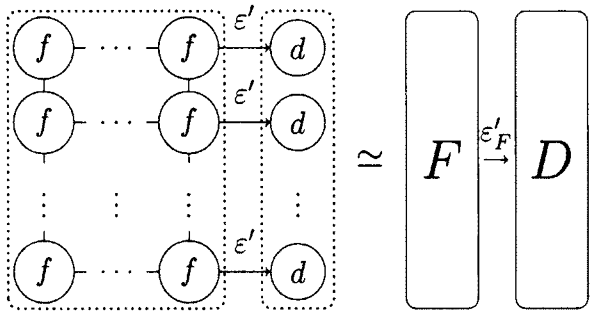


Figure 5.4: Super-forwarder F and super-destination D.

are effectively an average of $E[\xi']$ forwarders with non-empty queues in the super-forwarder F, and the average link error probability is ε', which is in the same form as (5.22) but computed using μ', the average error probability between any pair of forwarders within F is denoted by ε'_F and estimated to be

$$\varepsilon'_F = 1 - (1 - \varepsilon')^{E[\xi']} = 1 - (1 - \varepsilon')^{(1-p'_0)|F|} \tag{5.27}$$

By assuming that at any given time, the effective number of forwarders in F transmitting to D is also $E[\xi']$, each with an average link error probability of ε', the overall transmission error probability from F to D is estimated to be ε'_F.

A multicast destination d can decode to obtain K native packets when it receives K innovative coded packets. Since there may exist non-innovative packets among those received by d, given a finite field of size q from which coding coefficients are selected, the expected number of coded packets $\tilde{K}$ that d should receive before K

innovative packets are collected is given by [42]:

$$\bar{K} = \sum_{i=1}^{K} \frac{1}{1 - (\frac{1}{q})^i} \tag{5.28}$$

The expression for $\bar{K}$ in (5.28) is upper-bounded by $\frac{Kq}{q-1}$, which is close to K even with reasonably low values of q [42]. For instance, for a Galois field of size $q = 2^8$, on average it is sufficient for a multicast destination d to collect K innovative packets if the total number of coded packets it has received is K. Note that if d receives less than K packets, the decoding at d will fail.

Therefore, for the super-destination D to obtain K native packets, F must send to D at least K coded packets. Because in NetCoM, a multicast forwarder uses a new, random coding coefficient set for every transmission, any coded packet transmitted by any forwarder may possibly be an innovative packet. Each forwarder f in F can thus contribute a portion to the required K packets. Note that unlike NetCoM, such a contribution would not be feasible in non-coded ReM as without network coding or a special scheduling mechanism, packets transmitted by regular forwarders are duplicate; that is, in general, a ReM forwarder forwards the same packets it receives from its previous-hop forwarder.

We assume that each forwarder in F contributes an equal number of packets. Let π denote the number of packets each forwarder f contributes so that the total number of packets the super-forwarder F sends to D is K. A forwarder f will also overhear and forward (linear combinations of) the coded packets generated by the other forwarders to D. Since the average error probability between any pair of forwarders in set F is ε'_F, f will receive $\pi(1-\varepsilon'_F)$ packets from any other forwarder. As there are $E[\xi']$ effective forwarders in F, f will receive a total of $\pi(1-\varepsilon'_F)(E[\xi']-1)$ packets from the other effective forwarders. In total, f will have sent κ packets to d, where

$$\kappa = \pi + \pi(1-\varepsilon'_F)(E[\xi']-1) \tag{5.29}$$

We want κ to be at least K so that D may receive at least K coded packets. Replacing κ in (5.29) by K, we obtain π as follows:

$$\pi = \frac{K}{1+(1-\varepsilon'_F)(E[\xi']-1)} \tag{5.30}$$

In NetCoM, coded packets are transmitted in batches as linear combinations of native packets. We thus consider the average end-to-end delay Δ' of a *batch* instead of individual native packets. On the other hand, Δ' is also the average end-to-end delay of each individual native packet included in the batch (as if the native packets were encapsulated in a virtual data segment).

Using the same model as in ReM, Section 5.2.2.1, each coded packet travels an average distance of $E[\xi']$ hops to reach a multicast destination d. Given the average point-to-point delay δ' and the possibility of parallel transmissions over C channels, a coded packet will experience an average end-to-end delay of $(\delta' \times \max\{\frac{E[\xi']}{C}, 1\})$. In order for the destination d to receive a batch (having at least K coded packets),

f must send (contribute) π coded packets. Therefore, the average end-to-end delay Δ' is

$$\Delta' = \pi \times \delta' \times \max\left\{\frac{E[\xi']}{C}, 1\right\} \tag{5.31}$$

5.2.3.2 Average Packet Delivery Radio

Using the transformation model in Figure 5.4, the total effective number of packets N sent by $E[\xi']$ effective forwarders in F to D is $N = \lceil E[\xi']\pi \rceil$, because each effective forwarder contributes π coded packets to super-destination D. Substituting π using (5.30), we obtain

$$N = \left\lceil \frac{E[\xi']K}{[1 + (1 - \varepsilon'_F)(E[\xi'] - 1)]} \right\rceil \tag{5.32}$$

It can be seen that $N \geq K$ because the denominator in (5.32) is less than or equal to the term $E[\xi']$ in the numerator:

$$[1 + (1 - \varepsilon'_F)(E[\xi'] - 1)] \leq [1 + (E[\xi'] - 1)] = E[\xi']$$

Given the transmission error probability ε'_F from F to D, the average end-to-end packet delivery ratio Ω' of the super-destination D, or the multicast group, is the probability that D receives at least K out of N transmitted packets. This probability can be obtained using a binomial distribution:

$$\Omega' = \sum_{i=K}^{N} \binom{N}{i} (1 - \varepsilon'_F)^i (\varepsilon'_F)^{N-i} \tag{5.33}$$

5.2.3.3 Average Throughput

Similar to ReM, the throughput Γ'_d of an arbitrary multicast destination d is defined as the total number of native packets received by d, divided by the interval starting from the time the source begins transmitting to the time d receives the last native packet. In NetCoM, the average of this time interval is actually the average end-to-end delay Δ' of the batch, since all native packets are encapsulated in one batch, as explained in Section 5.2.3.1. Once d decodes the batch successfully, the number of native packets that d receives in the batch is K, and thus the throughput Γ'_d is equal to:

$$\Gamma'_d = \frac{K}{\Delta'} = \frac{K}{\pi \times \delta' \times \max\left\{\frac{E[\xi']}{C}, 1\right\}} \tag{5.34}$$

Since we use the average end-to-end delay Δ' in (5.34), the throughput Γ'_d of the arbitrary destination d can be considered as the average throughput Γ' of the multicast group.

In the following section, we present experimental results that validate the proposed models.

5.3 Numerical and Simulation Results

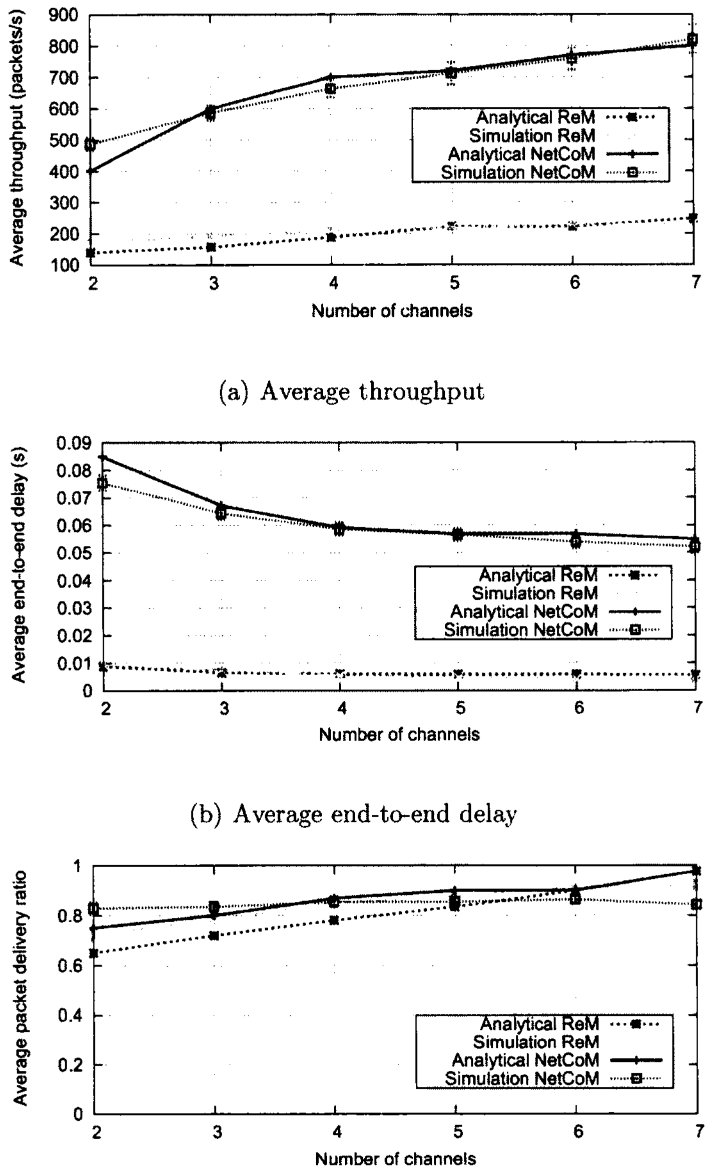
Using Qualnet [85], a software that provides scalable simulations of wireless networks, we simulate a network of $n = 50$ static nodes uniformly distributed in a

1200m $\times$ 1200m area. The channel bandwidth at the physical layer is $B = 11$ Mbits/s; the transmission range of the wireless routers is $R = 315$m, according to the specifications of the wireless routers manufactured by Tropos [98]. The IEEE 802.11 DCF CSMA/CA protocol without RTS/CTS/ACK exchange is chosen as the medium access control protocol for multicast transmissions, as explained in Section 5.1.2. The slot time, DIFS interval, and minimum backoff window size W are set at 20 μs, 50 μs, and 32, respectively, as dictated by the DSSS modulation scheme. The data packet size, excluding header size, is $S = 512$ bytes and the queue capacity at each router is $Q = 50$ packets. The queueing policy is FIFO. The path loss model is two-ray and there is no channel fading.

We use UDP at the transport layer in order to evaluate the network performance without any flow, congestion control or reliable mechanisms. The multicast group has one source placed at the center of the network, and 30 destinations randomly selected. The underlying routing algorithm is hop-count based, shortest path trees, built by applying the Dijkstra's algorithm [25] for each source-destination pair. The number of forwarding nodes $|F|$ is computed for each routing tree. The source transmits at a specified constant bit rate $\lambda = 250$ packets/s for 100 seconds of simulated time. The simulator then continues to run for 100 seconds of simulated time to give the last packets time to be routed. Each simulation data point is averaged from 50 runs using different network topologies and random seeds and plotted with a confidence interval of 95%.

We increase the number of channels C from 2 to 7 and measure the average end-to-end delays, packet delivery ratios and throughputs of both ReM and NetCoM as functions of C. The number of radios per node r is set at two for 2 channels and three for 3 to 7 channels. Channels are assigned to wireless links so that the number of distinct channels assigned to a node is not more than the number of radios of the node (as stated in Section 5.1.4). To obtain a fair comparison, the same number of packets $\ell = 25000$ is transmitted by the ReM and NetCoM sources with the ReM source sending uncoded original packets and the NetCoM source sending coded packets. For NetCoM, we select $K = 32$, a common batch size used in network coding experiments [99, 103]. Random coefficients for each linear combination are chosen from a Galois field of size $q = 2^8$, same as in [99, 103]. With this setting, the coding time ϕ is empirically found to be approximately 80 μs.

Using these same input parameters, the numerical results of the proposed models are computed by MATLAB and then plotted against the simulation results obtained from Qualnet, as shown in Figure 5.5. The graphs in Figure 5.5 show that the numerical results computed from the proposed analytical models are similar to the simulation results. Both the analytical and simulation results show that as the number of channels increases, the performance of ReM and NetCoM improves, as expected. Specifically, increasing the number of channels leads to higher throughput



(a) Average throughput

(b) Average end-to-end delay

(c) Average packet delivery ratio

Figure 5.5: Analytical results versus simulation results.

(Figure 5.5(a)), shorter end-to-end delay (Figure 5.5(b)), and higher packet delivery ratio (Figure 5.5(c)).

In addition, we observe that network coding does indeed help improve network throughput significantly (Figure 5.5(a)), in agreement with the objective of network coding. In particular, NetCoM average throughput is 2.5-3.5 times higher than ReM throughput. However, this gain comes at the expense of longer end-to-end delay (Figure 5.5(b)): NetCoM average end-to-end delay is 8-12 times longer than ReM end-to-end delay. There are two main factors that cause the longer end-to-end delay of NetCoM. First, NetCoM forwarding nodes require additional time for coding the received packets before transmitting. A ReM forwarder would simply forward the (native) packets it just received. Second, upon receiving a coded packet, a NetCoM destination may not be able to decode it right away. It has to wait to receive enough innovative packets (K of them as discussed in the above analysis, where K is the batch size) before decoding them to obtain the native packets enclosed in the batch. The delay to obtain the native packets is thus longer compared with ReM.

On the other hand, the packet delivery ratios of ReM and NetCoM are very close to each other (Figure 5.5(c)). This implies that network coding does not help or worsen the PDR of a multicast group, given the above network settings and simulation parameters.

5.4 Chapter Summary

In this chapter, we presented analytical models to estimate the average end-to-end delay, throughput and PDR of a multicast session without and with network coding in MCMR WMNs. Our models are based on the Markov chain, Poisson-based $M/M/r/Q$ queueing model, and the virtual super-node concept taking into account different timing components. The accuracy of the proposed models was validated via simulations, and realistic network settings. From the obtained results, we also showed the performance gains and the throughput-delay tradeoff of network-coded multicast in MCMR WMNs.

Chapter 6

Conclusion and Future Research Directions

In this thesis, we proposed the following solutions to provide high-performance multicast in MCMR WMNs.

6.1 Minimum-interference Multi-channel Multi-radio Multicast (M4)

We proposed the M4 algorithm for channel assignment in multicast trees in MCMR WMNs that minimizes interference without relying on the inconvenient use of interference factors. Advantages of the M4 algorithm include its simple implementation and high performance in both intra-flow and inter-flow interference environments.

Our simulation results showed that M4 outperforms existing CA algorithms in its group in terms of average PDR, throughput, and end-to-end delay under various network scenarios and conditions.

6.2 Bandwidth Efficient Multicast Routing (MCMNT)

We addressed the problem of building multicast routing trees with minimum numbers of transmissions in MCMR WMNs. We proved that the problem is NP-hard and then proposed approximate solutions. Our solutions include both centralized and distributed implementations with analyses of their running time complexities. We showed that multicast routing in MCMR WMNs should consider the wireless broadcast advantage and the underlying CAs in order to minimize network bandwidth consumption. The objective is to minimize the number of multicast transmissions as well as channel conflicts among multicast links for optimized performance. Experimental results showed that our proposed MCMNT routing algorithms perform significantly better than traditional multicast trees such as SPTs, STs and MFTs in terms of packet delivery ratio, throughput, end-to-end delay, and delay jitter.

6.3 Network-Coded Multicast Modeling

Network coding has received much attention as a promising technique for improving network throughput. Knowing the performance gain of network-coded multicast in MCMR WMNs not only guides the design of high-performance coding and multicast routing protocols, but also justifies the significant efforts being invested by the research community in exploring this new technology. To that end, we proposed analytical models to estimate the average end-to-end delay, throughput and PDR of a multicast session without and with network coding in MCMR WMNs. The accuracy of the proposed models was validated via simulations and realistic network settings. From the obtained results, we also showed the performance gains and the throughput-delay tradeoff of network-coded multicast in MCMR WMNs.

6.4 Open Issues and Directions for Future Research

In the following sections, we outline open issues and research directions for future work.

6.4.1 Channel Switching for Multicast in MCMR WMNs

When multiple flows (i.e., unicast, multicast and broadcast) coexist in the network, the traffic load of each flow should be considered in order to obtain the optimal solution. For instance, when a new flow starts, a CA or routing algorithm should be recomputed to account for the interference caused by the new flow. Similarly, when a flow terminates, the CA/routing must be updated to exclude the interference of this flow. Furthermore, a flow with very light load incurs less interference than one with heavy load. In other words, the traffic load of a flow determines its level of interference, and thus should be considered in a CA/routing algorithm. As a result, a node may switch its radio(s) from one channel to another on the fly during its network lifetime. Therefore, channel switching in MCMR WMNs plays a crucial role for a CA or routing algorithm to keep up-to-date with dynamic traffic changes. Although there exist several channel switching algorithms/protocols for unicast communication [109, 110, 111, 112, 113, 114], they are not applicable to multicast communication. The channel switching problem for multicast remains as a challenging task.

6.4.2 Comparison between the $c > r$ and $c \leq r$ Approaches

As discussed in Chapter 1, the $c \leq r$ CA approach has the advantages of simple implementation and not requiring channel switching. On the other hand, the $c > r$ CA approach may be perceived as offering a high performance because it allows more channel diversity at a node, potentially lowering interference around the node. However, the $c > r$ approach requires efficient channel switching schemes and carries additional channel switching overheads, which is not negligible [15, 18]. There has not been any work that compares the performance of these two approaches nor any consensus in the research community on the question of which approach is more favorable. A quantitative comparison will provide insight views of the pros and cons of each approach and guide towards an optimal solution.

Note that the lack of an efficient channel switching scheme for multicast prevents the $c > r$ approach from being deployed in an MCMR network. Therefore, we recommend the $c \leq r$ approach for a *quick deployment* of MCMR technologies and accommodation of various communication protocols in one single MCMR network.

6.4.3 Dynamic Multicast Membership and Traffic Loads

In a dynamic multicast group where members can join or leave at any time, multicast flows may start or terminate on the fly, creating a varying traffic load environment. When a new node joins the multicast group, CAs and routing paths should be updated to take into consideration the interference caused by the new node. On the other hand, when a node leaves, the interference from the node should be excluded. The traffic loads of new nodes or leaving nodes should also be considered since a node with a light load causes less interference than the ones with heavy loads. Nevertheless, as the multicast group and hence the traffic loads can be very dynamic, keeping track of all the activities in the network would be significantly expensive. Therefore, existing CA and routing schemes simply use periodic updates [7, 18, 115], regardless of network conditions. Our work in this thesis focused on network scenarios where multicast groups and traffic loads are static. A performance study of multicast flows under dynamic multicast membership and traffic loads remains an important open issue.

6.4.4 Network-Coded Multicast in MCMR WMNs

Existing work on the benefits of network coding focuses mostly on theoretical bounds [31, 32, 34] or analytical models [35, 37, 40, 41, 42]. A comprehensive performance evaluation of network-coded multicast in MCMR WMNs that investigates the practical gain of network coding using realistic network environment and useful performance metrics either under simulation or a real testbed remains an important research quest. Such study not only guides the design of high-performance coding and multicast routing protocols, but also justifies the significant efforts being invested by the research community in exploring this new technology.

Bibliography

[1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Inf. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[2] IEEE Std 802.11-2007, "Specific requirements: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2007.

[3] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *Proc. of ACM MOBICOM'01*, 2001, pp. 61–69.

[4] S. Shakkottai, X. Liu, and R. Srikant, "The Multicast Capacity of Large Multihop Wireless Networks," in *Proc. of ACM MOBIHOC'07*, Sep. 2007, pp. 247–255.

[5] P. Bahl, A. Adya, J. Padhye, and A. Walman, "Reconsidering Wireless Systems with Multiple Radios," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 5, pp. 39–46, Oct. 2004.

[6] W. Wang and X. Liu, "A Framework for Maximum Capacity in Multi-channel Multi-radio Wireless Networks," in *Proc. of IEEE CCNC'06*, 2006.

[7] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Networks," in *Proc. of IEEE INFOCOM'05*, 2005, pp. 2223–2234.

[8] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks," in *Proc. of ACM SIGMOBILE'05*, 2005, pp. 58–72.

[9] J. Tang, G. Xue, and W. Zhang, "Interference-aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks," in *Proc. of ACM SIGMOBILE'05*, 2005, pp. 68–77.

[10] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, "Optimization Models for Fixed Channel Assignment in Wireless Mesh Networks with Multiple Radios," in *Proc. of IEEE SECON'05*, 2005, pp. 463–474.

[11] M. Marina and S. Das, "A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks," in *Proc. of IEEE BroadNets'05*, 2005, pp. 381–390.

[12] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot,

"Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks," in *Proc. of IEEE INFOCOM'06*, 2006, pp. 1–12.

[13] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks," in *Proc. of IEEE SECON'07*, 2007, pp. 481–490.

[14] A. Mohsenian and V. Wong, "Joint Optimal Channel Assignment and Congestion Control for Multi-Channel Wireless Mesh Networks," in *Proc. of IEEE ICC'06*, 2006, pp. 1984–1989.

[15] P. Kyasanur and N. H. Vaidya, "Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'05)*, 2005, pp. 2051–2056.

[16] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, and Z. Zhang, "Distributed Channel Assignment and Routing in Multi-Radio Multi-Channel Multi-Hop Wireless Networks," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1972–1983, Nov. 2006.

[17] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks," in *Proc. of IEEE WCNC'07*, 2007.

[18] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks," in *Proc. of ACM SIGMOBILE'04*, 2004, pp. 50–65.

[19] G. W. Zeng, B. Ding, Y. L. Xiao, and M. Mutka, "Multicast Algorithms for Multi-Channel Wireless Mesh Networks," in *Proc. of IEEE ICNP'07*, 2007, pp. 1–10.

[20] Z. Yin, Z. Li, and M. Chen, "A Novel Channel Assignment Algorithm for Multicast in Multi-radio Wireless Mesh Networks," in *Proc. of IEEE ISCC'07*, 2007, pp. 283–288.

[21] C.-S. Chou, C. Chen, Y.-Y. Chen, R.-H. Jan, and C.-C. Hsieh, "Multicast Routing and Channel Assignment in Wireless Mesh Networks," in *Proc. of IEEE MASS'09*, Oct. 2009, pp. 676–681.

[22] S. H. Lim, C. Kim, Y. B. Ko, and N. H. Vaidya, "An Efficient Multicasting for Multi-Channel Multi-Interface Wireless Mesh Networks," in *Proc. of IEEE MILCOM'09*, 2009.

[23] H. Cheng and S. Yang, "Joint Multicast Routing and Channel Assignment in Multiradio Multichannel Wireless Mesh Networks Using Simulated Annealing," in *Springer LNCS*, 2008, pp. 370–380.

[24] H. S. Chiu, K. Yeung, and K. S. Lui, "Bandwidth-Guaranteed Multicast in Multi-Channel Multi-Interface Wireless Mesh Networks," in *Proc. of IEEE ICC'09*, 2009.

[25] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerical Mathematics*, vol. 1, pp. 269–271, Dec. 1959.

[26] R. Bellman, *Dynamic Programming.* Princeton University Press, 1957.

[27] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Mathematica Japonica*, vol. 24, pp. 573–577, 1980.

[28] L. Kou, G. Markowsky, and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica*, vol. 2, pp. 141–145, Jun. 1981.

[29] A. Zelikovsky, "An 11/6-approximation Algorithm for the Network Steiner Problem," *Algorithmica*, vol. 9, pp. 463–470, May 1993.

[30] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating Optimal Multicast Trees in Wireless Multihop Networks," in *Proc. of IEEE ISCC'05*, 2005, pp. 686–691.

[31] J. Le, J. C. S. Lui, and D. M. Chiu, "How Many Packets Can We Encode? - An Analysis of Practical Wireless Network Coding," in *Proc. of IEEE INFOCOM'08*, 2008.

[32] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. of ACM MOBICOM'07*, 2007.

[33] X. Zhang and B. Li, "On the Benefits of Network Coding in Multi-Channel Wireless Networks," in *Proc. of IEEE SECON'08*, 2008.

[34] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," in *Proc. of ACM CoNEXT'07*, 2007.

[35] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter, "An analysis of finite-memory random linear coding on packet streams," in *Proc. of WiOpt'06*, 2006.

[36] H. Su and X. Zhang, "Modeling Throughput Gain of Network Coding in Multi-Channel Multi-Radio Wireless Ad Hoc Networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 593–605, Jun. 2009.

[37] T. Yazane, H. Masuyama, S. Kasahara, and Y. Takahashi, "End-to-End Throughput Analysis of Multihop Wireless Networks with Network Coding," in *Proc. of IEEE ICC'10*, 2010.

[38] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft,

"XORs in the air: Practical wireless network coding," in *Proc. of ACM SIGCOMM'06*, 2006.

[39] D. Koutsonikolas, Y. C. Hu, and C.-C. Wang, "An Empirical Study of Performance Benefits of Network Coding in Multihop Wireless Networks," in *Proc. of IEEE INFOCOM'09*, 2009.

[40] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. of IEEE International Symposium on Information Theory*, 2003.

[41] R. Cogill, B. Shrader, and A. Ephremides, "Stability analysis of random linear coding across multicast sessions," in *Proc. of IEEE International Symposium on Information Theory*, 2008.

[42] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On Delay Performance Gains From Network Coding," in *Proc. of IEEE CISS'06*, 2006.

[43] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, Jul. 2000.

[44] K. Tang and M. Gerla, "Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks," in *Proc. of IEEE WCNC'00*, 2000.

[45] M.-T. Sun, L. Huang, A. Arora, and T.-H. Lai, "Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks," in *Proc. of International Conference on Parallel Processing*, 2002.

[46] A. Chen, D. Lee, G. Chandrasekaran, and P. Sinha, "HIMAC: High Throughput MAC Layer Multicasting in Wireless Networks," in *Proc. of IEEE MASS'06*, 2006.

[47] S.-H. Lee and J.-Y. Choi, "A Medium Access Control Scheme for Providing Reliability in Wireless Ad Hoc Networks," in *Proc. of WASA'06*, 2006.

[48] J. Xie, A. Das, and S. Nandi, "An Improvement to the Reliability of IEEE 802.11 Broadcast Scheme for Multicasting in Mobile Ad Hoc Networks," in *Proc. of IEEE SECON'04*, 2004.

[49] X. Wang, L. Wang, Y. Wang, Y. Zhang, and A. Yamada, "Supporting MAC Layer Multicast in IEEE 802.11n: Issues and Solutions," in *Proc. of IEEE WCNC'09*, 2009.

[50] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Inf. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[51] M. Kodialam and T. Nandagopal, "The Effect of Interference on the Capacity

of Multihop Wireless Networks," in *Proc. of IEEE International Symposium on Information Theory ISIT'04*, 2004.

[52] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Wireless Networks*, vol. 11, no. 4, pp. 471–487, 2005.

[53] Y. Zhang, J. Luo, and H. Hu, *Wireless Mesh Networking.* Auerbach Publications, 2007.

[54] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," in *Proc. of ACM MOBICOM'05*, 2005, pp. 73–87.

[55] H. Skalli, S. Ghosh, S. K. Das, L. Lenzini, and M. Conti, "Channel Assignment Strategies for Multiradio Wireless Mesh Networks: Issues and Solutions," *IEEE Commun. Mag.*, vol. 45, pp. 86–95, Nov. 2007.

[56] J. Crichigno, M. Y. Wu, and W. Shu, "Protocols and Architectures for Channel Assignment in Wireless Mesh Networks," *Ad Hoc Networks*, vol. 6, pp. 1051–1077, Sep. 2008.

[57] U. Nguyen and J. Xu, "Multicast Routing in Wireless Mesh Networks: Min-

imum Cost Trees or Shortest Path Trees?" *IEEE Commun. Mag.*, vol. 45, pp. 72–77, Nov. 2007.

[58] U. T. Nguyen, "On Multicast Routing in Wireless Mesh Networks," *Elsevier Computer Communications*, vol. 31, pp. 1385–1399, May 2008.

[59] S. Roy, D. Koutsonokolas, S. Das, and Y. C. Hu, "High-Throughput Multicast Routing Metrics in Wireless Mesh Networks," in *Proc. of IEEE ICDCS'06*, 2006, pp. 48–48.

[60] X. Zhao, C. T. Chou, J. Guo, and S. Jha, "Protecting Multicast Sessions in Wireless Mesh Networks," in *Proc. of IEEE Conference on Local Computer Networks (LCN'06)*, 2006, pp. 467–474.

[61] J. Yuan, Z. Li, W. Yu, and B. Li, "A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 2092–2103, Nov. 2006.

[62] X. Zhao, C. T. Chou, J. Guo, and S. Jha, "Protecting Multicast Sessions in Wireless Mesh Networks," in *Proc. of IEEE LCN'06*, 2006, pp. 467–474.

[63] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[64] A. Kimar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed point analysis of single cell IEEE 802.11 WLANs," in *Proc. of IEEE INFOCOM'05*, 2005.

[65] K. Medepalli and F. Tobagi, "Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications," in *Proc. of IEEE INFOCOM'06*, 2006.

[66] Y. Yang, J. C. Hou, and L.-C. Kung, "Modeling the Effect of Transmit Power and Physical Carrier Sense in Multi-hop Wireless Networks," in *Proc. of IEEE INFOCOM'07*, 2007.

[67] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling Per-Flow Throughput and Capturing Starvation in CSMA Multi-Hop Wireless Networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 864–877, Aug. 2008.

[68] Y. Wang and J. J. Garcia-Luna-Aceves, "Collision Avoidance in Multi-hop Ad Hoc Networks," in *Proc. of IEEE International Symposium on Modeling Analysis and Simulation of Computer and Telecommunications Systems*, 2002.

[69] F. Alizadeh-Shabdiz and S. Subramaniam, "Analytical models for single-hop and multi-hop ad hoc networks," in *Proc. of IEEE BroadNets*, 2004.

[70] A. A. Abdullah, F. Gebali, and L. Cai, "Modeling the Throughput and Delay in Wireless Multihop Ad Hoc Networks," in *Proc. of IEEE GLOBECOM'09*, 2009.

[71] S. Cho, S. Ramasubramanian, O. Turkcu, and S. Subramaniam, "Performance Analysis of Multi-Channel Wireless Infrastructure Networks," in *Proc. of IEEE LANMAN'10*, 2010.

[72] C. Li and W. Zhao, "Queueing Delay Analysis of Multi-Radio Multi-Channel Wireless Mesh Networks," in *Quality of Service Architectures for Wireless Networks: Performance Metrics and Management*, S. P. Sasan Adibi, Raj Jain and T. Tofigh, Eds. Information Science Publishing, 2010.

[73] H. L. Nguyen and U. T. Nguyen, "Minimum Interference Channel Assignment for Multicast in Multi-Radio Wireless Mesh Networks," in *Proc. of IEEE International Wireless Communications and Mobile Computing Conference (IWCMC'08)*, Aug. 2008, pp. 626–631.

[74] H. L. Nguyen and U. T. Nguyen, "Channel Assignment for Multicast in Multi-channel Multi-radio Wireless Mesh Networks," *Wiley Journal of Wireless Communications and Mobile Computing*, vol. 9, pp. 557–571, Apr. 2009.

[75] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.

[76] C. Cheng, R. Riley, S. Kumar, and J. Garcia-Luna-Aceves, "A Loop-free Extended Bellman-Ford Routing Protocol Without Bouncing Effect," *ACM SIGCOMM Computer Communications Review*, vol. 19, pp. 224–236, Sep. 1989.

[77] M. S. Sloman and X. Andriopoulos, "Routing Algorithm for Interconnected Local Area Networks," *Computer Networks and ISDN Systems*, vol. 9, pp. 109–130, 1985.

[78] D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol," RFC 1075, Nov. 1988.

[79] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Mar. 1994.

[80] S. J. Lee, M. Gerla, and C. C. Chiang, "On-Demand Multicast Routing Protocol," in *Proc. of IEEE WCNC'99*, 1999, pp. 1298–1302.

[81] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-demand Distance Vector Routing Protocol," in *Proc. of ACM/IEEE MOBICOM'99*, 1999, pp. 207–218.

[82] P. Winter, "Steiner Problem in Networks: A Survey," *Networks*, vol. 17, pp. 129–167, 1987.

[83] P. Winter and J. M. Smith, "Path-Distance Heuristics for the Steiner Problem in Undirected Networks," *Algorithmica*, vol. 7, pp. 309–327, Dec. 1992.

[84] M. A. Weiss, *Data Structures and Algorithm Analysis in Java.* Addison-Wesley, 2007.

[85] Qualnet Simulator. [Online]. Available: http://www.qualnet.com

[86] W. Schafer and E. Lutz, "Propagation Characteristics of Short-Range Radio Links at 60 GHz for Mobile Intervehicle Communication," in *Proc. of IEEE Telecommunications Symposium (ITS'90)*, 1990, pp. 212–216.

[87] H. L. Nguyen and U. T. Nguyen, "Bandwidth Efficient Multicast Routing in Multi-channel Multi-radio Wireless Mesh Networks," in *Proc. of IEEE SASN'09*, 2009.

[88] H. L. Nguyen and U. T. Nguyen, "High-Performance Multicast Routing in Multi-Channel Multi-Radio Wireless Mesh Networks," in *Proc. of IEEE GLOBECOM'10*, 2010.

[89] H. L. Nguyen and U. T. Nguyen, "Algorithms for Bandwidth Efficient Multicast Routing in Multi-channel Multi-radio Wireless Mesh Networks," in *Proc. of IEEE WCNC'11*, 2011.

[90] H. L. Nguyen and U. T. Nguyen, "Minimum Bandwidth Consumption Multicast Routing in Multi-Channel Multi-Radio Wireless Mesh Networks," 2011, submitted to the IEEE Transactions on Parallel and Distributed Systems (under review).

[91] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks," in *Proc. of ACM MobiHoc'09*, 2009.

[92] R. C. Prim, "Shortest Connection Networks and Some Generalisations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[93] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," in *Proc. of IEEE INFOCOM'00*, 2000, pp. 585–594.

[94] K. M. Chandy and J. Misra, "Distributed Computation on Graphs: Shortest Path Algorithms," *Communications of the ACM*, vol. 25, pp. 833–837, Nov. 1982.

[95] D. Bertsekas and R. Gallagher, *Data Networks*. Prentice Hall, 1991.

[96] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, v.3," RFC 3376, Oct. 2002.

[97] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks," *IEEE Communications Surveys and Tutorials*, vol. 11, p. 78, Mar. 2009.

[98] TROPOS Networks. [Online]. Available: http://www.tropos.com

[99] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. of ACM SIGCOMM'07*, 2007.

[100] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Trans. Netw.*, 2003.

[101] R. Jain, *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[102] Y. Yuan, K. Wu, and W. Jia, "On the Queueing Behaviour of Network Coding," in *Proc. of IEEE NetCod'10*, 2010.

[103] D. Koutsonikolas, Y. C. Hu, and C. C. Wang, "Pacifier: High-Throughput, Reliable Multicast without "Crying Babies" in Wireless Mesh Networks," in *Proc. of IEEE INFOCOM'09*, 2009.

[104] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traf-

fic conditions: throughput and delay analysis," *Elsevier Computer Communications*, vol. 25, pp. 313–321, 2002.

[105] R. Khalili, M. Ghaderi, J. Kurose, and D. Towsley, "On the Performance of Random Linear Network Coding in Relay Networks," in *Proc. of IEEE MILCOM'08*, 2008.

[106] R. Khalili and K. Salamatian, "A tighter Cut-Set bound for the multi-terminal erasure channel without side information," in *Proc. of IEEE International Symposium on Information Theory*, 2006.

[107] M. Chen, S. Serbetli, and A. Yener, "Distributed Power Allocation Strategies for Parallel Relay Networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 552–561, Feb. 2008.

[108] J. Liu, N. B. Shroff, and H. D. Sherali, "Understanding the Optimal Power Allocation Structure in AF-Based MIMO Cooperative Networks with Multiple Relays," Department of Electrical and Computer Engineering, The Ohio State University, Tech. Rep., Aug. 2010.

[109] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks," in *Proc. of IEEE INFOCOM'99*, 1999.

[110] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A Receiver-Initiated

Collision-Avoidance Protocol for Multi-Channel Networks," in *Proc. of IEEE INFOCOM'01*, 2001.

[111] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," in *Proc. of I-SPAN'00*, 2000.

[112] N. Jain and S. Das, "A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks," in *Proc. of IC3N'01*, 2001.

[113] A. Nasipuri, J. Zhuang, and S. R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," in *Proc. of IEEE WCNC'99*, 1999.

[114] A. Nasipuri and S. R. Das, "Multichannel CSMA with Signal Power-based Channel Selection for Multihop Wireless Networks," in *Proc. of IEEE VTC'00*, 2000.

[115] C.-J. Lin and C.-F. Chou, "Route-Aware Load-Balanced Resource Allocation for Wireless Mesh Networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'07)*, 2007, pp. 3093–3511.