# POWER-EFFICIENT MULTICASTING ALGORITHMS FOR WIRELESS AD HOC NETWORKS

## YE JIANYANG

NATIONAL UNIVERSITY OF SINGAPORE

2003

# POWER-EFFICIENT MULTICASTING ALGORITHMS FOR WIRELESS AD HOC NETWORKS

**YE JIANYANG**
*(B.Eng (Hons.) NJTU)*

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

*To my loving Dad, Mum, and Sister,*

*Darling Viola,*

*And all my friends*

# ACKNOWLEDGEMENTS

I would like to first express my sincere gratitude to my supervisors Professor Lawrence Wong, and Professor K. C. Chua. They took a chance by accepting me as their student, and patiently guided me throughout my research career at NUS. They always found time from their busy schedule whenever I need an intelligent discussion, and gave me helpful advices since my first day at NUS. Their knowledge and experiences in the field of mobile ad hoc networks are insurmountable. I felt so fortunate to have them as my supervisors. I also thank each one of my thesis examiners for their valuable comments and feedbacks.

I am very grateful to my family and friends, for their support, care, encouragement and understanding during the course of this research. They are always the strongest backings what keep me going on through the tough times. Particularly, I want to say “Thank You” to my darling girlfriend Viola. Without you, I cannot make it at all!

My gratitude also goes to all my colleagues at Mobile Multimedia Research Laboratory, for their kind help and their willingness to share their expertise without reservation.

Lastly, I thank National University of Singapore for providing me such a good research environment.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# PUBLICATIONS

J. Ye, W.C. Wong and K.C. Chua, "Power-Efficient Multicasting in Ad Hoc Networks", *Proceedings of the 11th IEEE International Conference on Networks (ICON 2003)*, Sydney, Australia, September/October 2003.

J. Ye, W.C. Wong and K.C. Chua, "Power Conservation and Path Efficiency for Multicast Ad Hoc Networks", *Proceedings of the 2004 IEEE International Conference on Communications (ICC 2004)*, Paris, France, June 2004.

# ABBREVIATIONS

| | |
|---|---|
| AMRIS | Adhoc Multicast Routing protocol utilizing Increasing id-numberS |
| AMRoute | Adhoc Multicast Routing Protocol |
| AODV | Ad Hoc On Demand Distance Vector |
| ATM | Asynchronous Transfer Mode |
| BS | Base Station |
| BTr | Receive Busy Tone |
| $BT_t$ | Transmit Busy Tone |
| CAMP | Core-Assisted Mesh Protocol |
| CBT | Core Based Trees |
| CDMA | Code Division Multiple Access |
| CTS | Clear To Send |
| DAG | Directed Acyclic Graph |
| DARPA | Defense Advanced Research Projects Agency |
| DBTMA | Dual Busy Tone Multiple Access |
| DVMRP | Distance Vector Multicast Routing Protocol |
| FG | Forwarding Group |
| GloMoSim | Global Mobile Information System Simulator |
| GPS | Global Positioning System |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |

| | |
|---|---|
| IPC | Ideal Power Controlled |
| LAM | Lightweight Adaptive Multicast |
| MAC | Medium Access Control |
| MACA | Multiple Access with Collision Avoidance |
| MACAW | Multiple Access with Collision Avoidance Wireless |
| MANET | Mobile Ad Hoc Network |
| MAODV | Multicast Ad hoc On Demand Distance Vector |
| Mbone | Multicast Backbone |
| MOSPF | Multicast Open Shortest Path First |
| MSC | Mobile Switching Center |
| ODMRP | On-Demand Multicast Routing Protocol |
| OSI | Open System Interconnect |
| PCMA | Power Controlled Multiple Access |
| PDR | Packet Delivery Ratio |
| PIM | Protocol Independent Multicast |
| PSTN | Public Switched Telephone Network |
| QoS | Quality of Service |
| RFC | Request for Comments |
| RPF | Reverse Path Forwarding |
| RREP | Route Reply |
| RREQ | Route Request |
| RTS | Request To Send |
| SIR | Signal-to-Interference Ratio |
| TCP/IP | Transfer Control Protocol / Internet Protocol |

| | |
|---|---|
| TORA | Temporally-Ordered Routing Algorithm |
| WLAN | Wireless Local Area Network |
| WNP | Wireless Nodal Propagation |

# SUMMARY

Wireless ad hoc networks have received renewed interests recently in the research community. However, there are also many challenges arising due to their infrastructureless nature. Specifically, wireless ad hoc networks often consist of battery-powered nodes with limited lifetime. For those power-constrained networks, a crucial issue in routing and multicasting is to conserve as much power as possible while still achieving good throughput performance. In this thesis, we look at the problem of constructing power-efficient multicasting trees for static wireless ad hoc networks.

After reviewing on current works of multicast protocols and power control within ad hoc networks, we first establish a new "node-based" wireless networking model for power-efficiency consideration. The unique wireless nodal propagation property is highlighted, which has great impact on the design of power-controlled multicasting trees. Then we propose a class of minimum-power multicasting algorithms using only transmission power as cost function. Simulation results yield much better performance in terms of power efficiency and interference control, compared with a baseline algorithm without power control. However, this improvement is achieved at the cost of increased multicast tree size and average hop count.

After investigating the potential path inefficiency behind minimum-power algorithms, we design a new cost function to jointly optimize power conservation and path efficiency. Performance of algorithms using this new cost function demonstrates great improvement of path efficiency in comparison with algorithms using pure power as cost function. The multicast tree size and average hop count are reduced significantly. Moreover, the new cost function still achieves similar or even better performance in terms of power conservation and interference minimization.

Finally, we extend our discussion to a full mobile environment. A distributed two-phase multicast tree maintenance algorithm is proposed. Simulation results show that it is able to effectively reduce the number of tree link breaks under mobility, and repair the broken tree link in a locally power-efficient manner.

# CHAPTER ONE

# Introduction

## 1.1 Background

Since their emergence in the early 1970s, wireless networks have become increasingly popular in the computing industry. This is particularly true within the past decade, which has seen rapid widespread development and deployment of various wireless communication devices and products, such as mobile cellular phones, portable notebook computers, palmtops, and so on. At the same time, the markets for wireless telephony and communication devices are experiencing incredible growth. By now it is well known, understood, and accepted that wireless communication, computation, and control is not only desirable but also unavoidable; the possibility of communicating at *anytime*, from *anywhere*, to *anybody* or *anything*, in the world and even beyond, is imminent. We can expect that, some day in the not-too-distant future, mobile computing and communication will become indispensable even at times when and at places where the necessary infrastructures are not available. Wireless computing devices should physically be able to communicate with each other, even when no routers or gateways or base stations can be found. In the absence of a network infrastructure, what is needed is that the wireless nodes themselves take on the missing functions and thereby form an *ad hoc network*.

### 1.1.1 Wireless Ad Hoc Networks

Ad hoc networks [1-4] have their roots in the 1960s as the DARPA packet radio networks [5]. Recently there has been a renewed interest in such areas due to the increasing advances in wireless technologies and mobile computing. A *Mobile Ad Hoc Networking* (MANET) working group [2] has also been formed within the Internet Engineering Task Force (IETF) to develop a routing framework for IP-based protocols in ad hoc networks.

An *ad hoc network*, or more recently termed as a *Mobile Ad Hoc Network* (MANET), is a collection of wireless mobile nodes that dynamically self-configure to form a network without the aid of any pre-established infrastructure or centralized administration, as shown in Figure 1.1. Without any inherent infrastructure, the mobiles have to handle all necessary networking and control tasks by themselves, generally through the distributed control algorithms. In such a network, *multihop* connections, whereby intermediate nodes forward the packets to their final destination, are required to allow for efficient wireless communication between nodes that are relatively far apart; each node must operate not only as a host but also a router. Therefore, wireless ad hoc networks are also known as *mobile multihop networks*.

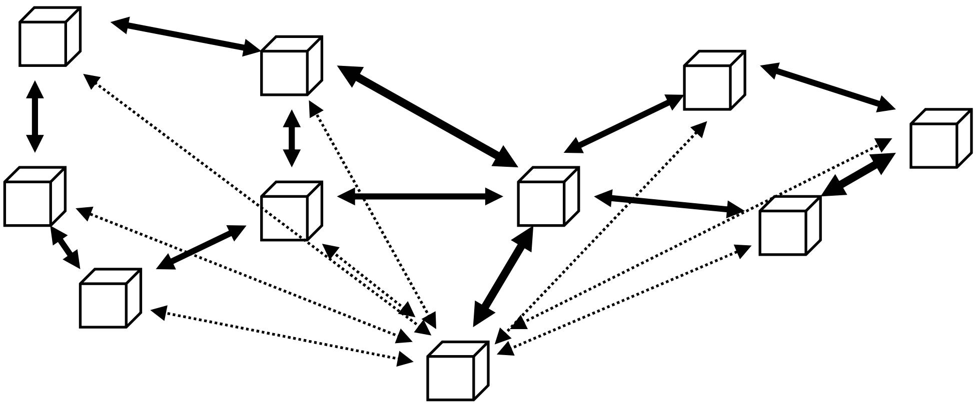


**Figure 1.1** An Example of Ad Hoc Networks

Ad hoc networks are highly appealing for many reasons. Due to their inherent autonomous nature, they can be rapidly deployed and reconfigured, which makes them particularly suitable for impromptu situations where no fixed wired infrastructure is available, either because it may not be physically possible or economically practical to provide the necessary infrastructure, or because the expediency of the situation does not permit its installation. As implied by the name of “ad hoc”, they can be easily tailored to specific applications. They are also highly robust because of their distributed nature, node redundancy, and the lack of single points of failure. These characteristics are especially important and attractive for military applications, such as battlefield communications and military sensor networks. On the other hand, typical commercial applications of ad hoc networks include emergency search and rescue missions and disaster recovery operations, conferences or meetings where participants need to share information dynamically and interactively, distributed collaborative computing and control, and so forth. In recent years, new ad hoc networking technologies, Bluetooth [6] for instance, have emerged and enabled personal area networks and home networks to be the new and promising application areas of wireless mobile ad hoc networks.

### 1.1.2 Unique Characteristics of Ad Hoc Networks

Although ad hoc networks enable many new and exciting applications, a set of unique characteristics make them greatly different from traditional wireless networks and pose significant technical challenges within this brand-new research area.

The lack of infrastructure inherent to ad hoc networks can be best illustrated in contrast with the prevalent wireless networks today: cellular networks and wireless local area networks (WLANs). In cellular telephone networks, a mobile terminal located in a given cell communicates directly with a base station (BS). There is no peer-to-peer communication between mobiles; all communication is via the BS through single-hop routing. Each BS is connected by a high-speed link (typically wired) to a mobile switching center (MSC) that in turn is connected to the public switched telephone network (PSTN). The BS and MSC perform all control and networking functions, including authentication, call routing, and handoff. The mobile units depend entirely on the BS/MSC/PSTN infrastructure for connectivity and centralized control. Most WLANs have a similar centralized single-hop architecture: mobile nodes communicate directly with a centralized access point that is connected to the backbone Internet, and the access point performs all networking and control functions for the mobile nodes. In sharp contrast, ad hoc networks have peer-to-peer communications, distributed networking and control functions among all mobiles, and multihop routing.

Another important characteristic of ad hoc networks is mobility in the mobile nodes. All mobiles are free to move arbitrarily; thus, the network topology may change randomly and rapidly at unpredictable patterns, and consist of both bidirectional and unidirectional links. Due to the possibly high-speed movement of the mobiles and fast changing wireless propagation conditions, network information, such as routing tables, for instance, may become quickly obsolete. This may lead to frequent network reconfigurations and excessive exchanges of large amount of control information over the wireless channels.

In a typical ad hoc environment, all communications are carried over the wireless medium, consisting of bandwidth-constrained, variable capacity links. Due to the radio communications being extremely vulnerable to propagation impairments, network connectivity may not be guaranteed. In fact, intermittent and sporadic connectivity may be quite common. In addition, wireless links will continue to have significantly lower capacity than their hardwired counterparts. The realized throughput of wireless channels, after accounting for the effects of noise, interference, fading, multiple access, and so on, is even much less than a radio's maximum transmission rate.

Energy constraints are not inherent to all wireless ad hoc networks. Mobile devices may be stationary and attached to a large energy source, or may be part of a large vehicle, such as a car or tank, which can generate significant amount of power over the long term. However, many ad hoc wireless network nodes will be powered by batteries with limited lifetime; some of them may not even be rechargeable. For these nodes, the most important system design criteria for optimization may be energy conservation.

In addition, some envisioned ad hoc networks, such as mobile military networks or highway networks, may be relatively large (e.g. up to tens of thousands of nodes). The need for scalability is crucial to these networks. Moreover, mobile wireless networks are generally more prone to physical security threats than wired networks. The increased possibility of eavesdropping, spoofing and denial-of-service attacks should be carefully considered for the design of secure wireless ad hoc networks.

In summary, those characteristics mentioned above, but by no means are limited to, make wireless mobile ad hoc networks a very unique and promising, but in the same time quite challenging research area.

### 1.1.3 Challenges in Routing and Multicasting

Routes in ad hoc networks are multihop because of their infrastructureless nature and limited propagation range of wireless radios. Since nodes in the network may move freely and randomly, routes often get disconnected. Routing protocols are thus responsible for maintaining and reconstructing the routes in a timely manner as well as establishing the durable routes. Additionally, routing protocols are required to perform all the above tasks without generating excessive control message overhead. Control packets must be utilized effectively and efficiently, and be generated only when necessary.

Multipoint communications [7, 8] have emerged as one of the most researched areas in the field of networking. As the technology and popularity of Internet grow, applications, such as video conferencing, which require multicast support, are becoming more and more widespread. In a typical ad hoc environment, network hosts often work in groups to carry out a given task. Therefore, multicast plays an important role in mobile ad hoc networks. However, multicast protocols used in static wired networks, such as Distance Vector Multicast Routing Protocol (DVMRP) [9], Multicast Open Shortest Path First (MOSPF) [10], Core Based Trees (CBT) [11], and Protocol Independent Multicast (PIM) [12], do not perform well in wireless ad hoc networks because multicast trees are fragile and must be readjusted or reconstructed as connectivity or multicast group membership changes. Furthermore, multicast trees usually require a global routing knowledge such as link state [13] or distance vector [14] stored in every node. The frequent exchange of routing information in mobile ad hoc networks, triggered by continuous topology changes, yields excessive routing and processing overhead and degrades greatly the performance of those traditional multicast protocols. Hence, the tree structures used in static networks must be modified, or a different topology among group members, mesh for example, needs to be deployed for efficient multicasting in mobile ad hoc networks.

### 1.1.4 Power Control Considerations within Ad Hoc Networks

Power control is of fundamental importance to the operation of wireless networks, especially cellular mobile Code Division Multiple Access (CDMA) systems, for a number of reasons. First of all, by adjusting transmitter power to maintain an acceptable receiving signal quality at each mobile using the least possible power, it is feasible and favorable to minimize power consumption and prolong the battery life of mobile nodes. Moreover, an efficient power control scheme can largely mitigate the near-in and far-out interferences, improve the spatial reuse of channel resources and increase the network capacity, which is particularly beneficial to CDMA systems [15-17]. Power control can also be used as a vehicle for performing several key dynamic network operations online, such as admission control, link Quality of Service (QoS) maintenance, channel probing, resource allocation, and handoffs [18].

With the explosion of cellular networks in the 1990s, power control receives more research attention around the world, and fairly mature commercial power control technologies for cellular mobile networks have been developed. However, in the new scenario of ad hoc networks, power control still remains as an unexplored area. A number of complications and challenges are introduced by the autonomy inherence of ad hoc networks.

Firstly, unlike cellular mobile networks, which have a fixed BS to centrally manage and facilitate power control of all mobile units inside each cell, there does not exist any centralized infrastructure in ad hoc networks. Efficient distributed power control algorithms are therefore required to be implemented at every mobile node.

Secondly, highly dynamic topology and arbitrary node movement make power measurement and interference estimation less accurate and more complex, which in turn causes power control design and implementation more difficult.

Thirdly, collision avoidance based contention mechanism is commonly used in current popular wireless medium access control (MAC) protocols in ad hoc networks, such as Multiple Access with Collision Avoidance (MACA) [19], Multiple Access with Collision Avoidance Wireless (MACAW) [20], Dual Busy Tone Multiple Access (DBTMA) [21], and the widespread IEEE 802.11 standard [22] MAC protocol. According to this approach, all mobile users contend for a single shared channel when they have packets to send, which means all communications are essentially only simplex, making transmission power control much more complicated. In addition, under the assumption of "commutativity of transmission", all these MAC protocols have typically used fixed power level to transmit, without taking into account the possibility of power control implementation. On the other hand, adaptively changing the communication area of a transmitter-receiver pair, according to their relatively physical distance, can lead to less interference and more efficient spatial channel reuse, and alleviate the well known hidden and exposed problems [23] inherent to shared channel wireless networks. Therefore, power control is highly desirable and crucial for improving the performance of current MAC protocols in ad hoc networks.

Fourthly, although power control is traditionally studied at the physical and link layer, it has a significant impact on the protocol stack above the link layer within the very particular paradigm of wireless ad hoc networks, which has mostly been overlooked in the past. For example, the level of transmitter power defines the "local neighborhood" — the collection of nodes that can be reached in a single hop — and thus in turn defines the context in which access, routing, and other higher-layer protocols operate. Moreover, power control introduces asymmetric links, where one station can transmit to another, but the latter cannot directly reply to the sender. Unlike cellular mobile networks, which are basically only the last-hop extensions of wired networks, ad hoc networks are essentially multihop. Therefore, varying network topology and asymmetric links induced by power control complicate the designs of higher-layer protocols, such as routing and multicasting, for mobile ad hoc network.

In short, although power control consideration is more necessary and desirable for wireless ad hoc networking, it introduces more complications and challenges in the meantime. Specifically, power control has a more prominent effect on protocols above the link layer; in other words, the independences among multiple layers induced by power control is so significant that we argue that traditional OSI and TCP/IP layered architectures may not be suitable or accurate any more for the proper description of mobile ad hoc networks. A holistic cross-layer protocol design that supports adaptivity and optimization across multiple layers may be required and would likely mark a milestone in the development of this field [24].

## 1.2 Accomplishments and Contributions

Our accomplishments and contributions, which are elaborated throughout this thesis, can be briefly listed as follows:

- Studied and compared in detail current multicast protocols for wireless ad hoc networks, including LAM, AMRoute, AMRIS, ODMRP, CAMP, and MAODV.

- Established a new “node-based” wireless networking model for power-efficiency design according to the observation of wireless nodal propagation property.

- Proposed a class of minimum-power multicasting heuristics for wireless ad hoc networks. By using transmission power as cost function, these heuristics globally minimize the power consumption and improve the overall multicasting performance at the expense of longer average hop count.

- Explored the tradeoff between power minimization and path efficiency, proposed a new cost function to jointly optimize power conservation and path efficiency, and applied the new cost function to develop some power-efficient algorithms to achieve power conservation and path efficiency together.

- Extended our discussion to a full mobile environment by proposing a distributed two-phase power-controlled multicast tree maintenance algorithm.

## 1.3 Organization of the Thesis

The rest of this thesis is organized as follows:

- Chapter 2 begins with an introduction to multicasting concept, followed by a brief summary of basic multicast algorithms used in wired networks. A review of current multicast protocols for ad hoc networks, including LAM, AMRoute, AMRIS, ODMRP, CAMP, and MAODV, is then detailed.
- In Chapter 3, some background of power control in cellular mobile networks is first introduced. The challenges and motivations of power control design in wireless ad hoc networks are then discussed, and a brief survey of current power control research is reported, which motivates our research interest on the designs of power-efficient multicasting algorithms and protocols. Finally a primary exploration on the possibility and potential methods to integrate power control with multicasting algorithm designs is investigated.
- Chapter 4 begins with some preparation knowledge introduction related to our algorithm design. A class of minimum-power multicasting heuristics for

wireless ad hoc networks are proposed and illustrated carefully after problem formulation. Careful simulations are conducted to investigate their performance. Then a new cost function is designed to jointly optimize power and path efficiency together; the performance of power-efficient algorithms using this new cost function is simulated and compared. Finally, a distributed two-phase power-controlled multicast tree maintenance algorithm is proposed.

- Chapter 5 concludes the thesis, and some suggestions about future works are discussed.

# CHAPTER TWO

# A Review of Current Multicast Protocols

As the technology and popularity of Internet grow, applications requiring multicast support, such as video conferencing, are becoming more widespread. Multipoint communications have emerged as one of the most researched areas in the field of networking. We begin this chapter with a background introduction of multicast communication and some basic multicast routing algorithms used in wired networks today. A detailed review of current multicast protocols for wireless mobile ad hoc networks, including LAM, AMRoute, AMRIS, ODMRP, CAMP, and MAODV, is then presented. The underlying methodologies of each protocol are expanded and compared; the relative strengths, weaknesses, and applicability of each are also discussed. Finally, a general comparison of those protocols and a brief classification of current approaches are summarized.

## 2.1 Background

### 2.1.1 The basics of Multicast Communication

The proliferation of multimedia applications associated with the explosion of Internet and other high-speed networks, such as ATM, is driving the need for *group communication*, which involve more than two users (these users define a "group") that wish to communicate with one another. Within the context of group communication, three communication types, unicast, broadcast and multicast, can be differentiated, depending on the number of senders and receivers involved [7]. *Unicast* is equivalent to traditional point-to-point communication. There is exactly one sender and one receiver. *Broadcast* is another type in which there also exists only one sender, but the receivers are the all nodes except that sender. Traditional televisions and radios are two everyday examples of broadcast communication. *Multicast* is the most diverse form of group communication, because it places no restriction on the number of senders and receivers that can communicate. From this point, both unicast and broadcast can be viewed as two special cases of multicast. Traditional one-to-many and many-to-many communications are two typical forms of multicast.

The main advantage of multicast is the efficiency achieved by reaching all members of a group at the same time. It is able to deliver identical information to an entire group simultaneously, instead of separately to each member of the group as in unicast; the latter often leads to large network resource waste and varying delays to the individual members. On the other hand, multicast can selectively deliver information only to those who are interested in, so it is more flexible and efficient than broadcast, especially when the number of interested receivers is small compared with that of the whole network nodes.

Typical applications of multicast communication include video conferences, distributed data bases, open distance learning or teleteaching, and, distributed cooperative teamwork.

### 2.1.2 Multicast Routing Algorithms in Wired Networks

Multicast routing differs from traditional point-to-point unicast routing in two ways. First, it has to deal with a group of receivers instead of a single one. In contrast to the single path between sender and receiver in unicast, some form of *distribution tree* is required for efficient multicasting. Second, the membership of the group can change dynamically, which makes the exchanges and updates of multicast routing information more difficult; and some multicast tree maintenance mechanism is needed.

Tree multicast is a well-established concept in wired networks, especially on Internet. There are three basic algorithms to construct multicast trees [7, 8].

- *Source-based Routing*: The source-based routing algorithm, also known as *Reverse Path Forwarding* (RPF), is due to Dalal and Metcalfe [26]. It has seen widespread use through IP multicast [27, 28] on the multicast test bed of the Internet, the MBone (Multicast Backbone). The RPF algorithm computes an implicit spanning tree per source which is minimal in terms of transit delay or hop count if the reverse path is calculated on a unicast metric of hop counts. It is optimized for dense receiver distribution and can be implemented in a distributed fashion with local recovery. The main advantage of it is that it does not require any resources in addition to the classic unicast routing tables. However, RPF routing may fail badly if the underlying unicast routing is asymmetric. Moreover, multiple per-source trees each rooted at a single sender are required if the group has more than one sender, which adds more routing overhead and makes source-based routing unscalable for large networks.
- *Steiner Trees*: The Steiner tree algorithm [29-30] is a *monolithic* algorithm that designs a tree spanning only the group of members with the minimal cost, according to a distance defined on the network edges. Since Steiner trees

optimize on a global basis, a recalculation is required each time a change in topology or group membership occurs. The Steiner problem is well-known NP-complete [30]. In other words, finding the minimum Steiner tree in a graph has an exponential cost for a result which is not necessarily optimal. It has been shown that the minimum cost of a Steiner tree algorithm is $O(n \log n)$, where $n$ is the number of nodes in the network and with all distances equal to one on the links [8]. Therefore, although Steiner tree represents a popular approach pursued in theory, its mathematical complexity and limited suitability for use in real-time environments make it less attractive for practical implementations. Furthermore, it is aimed at a centralized solution, which is often not practical in large real communication systems. Only heuristics can be implemented on a distributed manner. A variety of heuristics have been proposed for the construction of Steiner trees. In many cases, it shows that simple techniques can provide solutions that are as good [8].

- *Center-based Trees*: The center-based tree algorithm, also known as core-based tree (CBT) [11] algorithm, is aimed at multiple sender/multiple receiver scenario, in which one node is chosen as the center or core of the group, multicast packets from various senders are all first sent to the core node and then the core is responsible to forward packets to those intended receivers. It has the advantage over RPF of only requiring a state information item per group instead of per source per group. The centered approach does, however, suffer from suboptimal paths and traffic concentration, as the traffic from all senders of a given group will converge to the center. In addition, the core node represents a single point of failure, which means that fault tolerance is low. Furthermore, the selection of an optimal center node is an NP-complete problem [30]. Locating the optimal tree center requires the complete knowledge of the network topology and group membership.

The three basic techniques for multicasting described above are summarized and compared, along with some of their important characteristics, in Table 2.1.

**Table 2.1** Summary and Comparison of Basic Multicasting Algorithms.

| ***Algorithms*** / ***Criteria*** | ***Source-based Routing*** | ***Steiner Trees*** | ***Center-based Trees*** |
|---|---|---|---|
| Approach | Incremental | Monolithic | Incremental |
| Centralized/Distributed | Distributed | Centralized | Distributed |
| Routing Path | Reverse Shortest Paths | Globally Optimal | Suboptimal |
| Overhead | Large | Large | Medium |
| Scalability | Average | Low | High |
| Traffic Concentration | No | No | Yes |
| Single Point of Failure | No | No | Yes |
| Suitable Group Density | Dense | — | Sparse |
| Practical | Yes | No | Yes |
| Example Protocols | DVMRP, PIM Dense Mode | — | CBT, PIM Sparse Mode |

## 2.2 A Review of Current Multicast Protocols for Ad Hoc Networks

Although multicasting is a well researched area in the wired networks, it becomes much more complex in wireless mobile ad hoc networks, due to the infrastructureless nature, multihop routing, and arbitrary mobility. Popular multicast protocols used in static wired networks, such as Distance Vector Multicast Routing Protocol (DVMRP) [9], Multicast Open Shortest Path First (MOSPF) [10], Core Based Trees (CBT) [11], and Protocol Independent Multicast (PIM) [12], do not perform well in ad hoc networks. Multicast trees are fragile and must be readjusted or reconstructed as connectivity or multicast group membership changes, which is quite common in ad hoc environments. Furthermore, multicast trees usually require a global routing knowledge such as link state [13] or distance vector [14] stored in every node. The frequent exchange of routing information, triggered by continuous topology changes, yields excessive routing and processing overhead, and degrades greatly the performance of traditional multicast protocols. Hence, the tree structures used in static networks must be modified, or a different topology among group members, mesh for example, needs to be deployed for efficient multicasting in mobile ad hoc networks.

Until recently, a few multicast protocols have been proposed for wireless ad hoc networks. In this section, we present a detailed review about these protocols. Underlying methodologies are illustrated and compared; the relative strengths, weaknesses, and applicability of each are also discussed.

### 2.2.1 Lightweight Adaptive Multicast (LAM)

LAM [31] is a core-based shared tree multicast protocol which is built upon the Temporally-Ordered Routing Algorithm (TORA) [43, 44]. Conceptually, it can be thought of as an integration of the Core Based Tree (CBT) multicast routing and TORA unicast. TORA is a highly adaptive on-demand routing algorithm based on the concept of *link reversal*. All nodes involved in routing to a certain destination are totally ordered by labels called "*height*". A Directed Acyclic Graph (DAG) is established using height metric. Links in this DAG are assigned a *direction* (upstream or downstream) based on their relative heights. Thus, sequences of links with "down" directions form multiple routes towards the destination. Network control is localized using the process of link reversal when node mobility breaks a DAG route.

As with CBT, LAM builds a group-share multicast tree for each multicast group. This tree is centered at a pre-selected CORE node. When a node wants to join a particular multicast group, it sets the **member** flag indicating that it is interested in joining this group, and generates a **JOIN** message containing the group id and the target CORE id. By looking at the link status table in the version of TORA running for the target CORE, it picks the neighbor with the lowest height as the receiver of this **JOIN** message and sends out the message. The **JOIN** message is only supposed to travel along a "downwards" path in the TORA's DAG with respect to the target CORE. If a **JOIN** message is received over an upstream link, it is considered invalid and the node receiving it needs to send a **LEAVE** message back to the joiner as a rejection. Only the CORE itself or an on-tree node which receives the **JOIN** message via a valid non-downstream TORA link can accept this **JOIN** message and reply it by sending back an **ACK** message to the joining node. Reception of a valid **ACK** message changes the joining node's state from join-waiting to on-tree. Later if this node is no longer interested in this group and it has an empty **CHILD-LIST**, it can leave the tree by sending a **LEAVE** message to its parent, and then removing all information regarding this multicast group. If the parent's **CHILD-LIST** becomes empty and it is not interested in the group, it can also leave the tree by sending a **LEAVE** message to its parent.

LAM's tree maintenance phase does not utilize timers. Its tree maintenance control traffic is proportional to the rate of network changes. Therefore when tree links in the network are stable, no control messages are generated by LAM. As it is closely coupled with TORA, whenever there is any topological change, TORA will react accordingly to maintain its DAG and thereafter notify LAM about link status changes through their interface. Hence, LAM control operations are purely event-triggered, LAM does not need to check link status through periodic polling. Also, the portion of the tree that is affected during any failure is typically highly localized thanks to TORA's link reversal ability.

However, similar to other core-based protocols, LAM suffers from traffic concentration and the single point of failure at the CORE. Furthermore, although the tight coupling of LAM and TORA increases its reaction efficiency to topological changes and lowers maintenance control overhead, the great dependence upon TORA makes it less portable across domains supporting different unicast protocols.

### 2.2.2 Adhoc Multicast Routing (AMRoute)

AMRoute [32] creates a bidirectional shared *user-multicast tree* using only group senders and receivers as tree nodes. *Virtual unicast tunnels* or AMRoute *meshes* are used as tree links to connect group members. Each group has at least one *logical core* that is responsible for member and tree maintenance. Initially, each group member declares itself as a core for its own group of size one. Each core periodically floods **JOIN-REQ**s (using an expanding ring search) to discover other disjoint mesh segments for the group. When a member node receives a **JOIN-REQ** from a core of the same group but a different mesh segment, it replies with a **JOIN-ACK** and marks that node as a mesh neighbor. The node that receives a **JOIN-ACK** also marks the sender of the packet as its mesh neighbor. After the mesh creation, each core periodically transmits **TREE-CREATE** packets to mesh neighbors in order to build a shared tree. When a member node receives a non-duplicate **TREE-CREATE** from one of its mesh link, it forwards the packet to all other mesh links. If a duplicate **TREE-CREATE** is received, a

**TREE-CREATE-NAK** is sent back along the incoming link. The node receiving a **TREE-CREATE-NAK** marks the link as mesh link instead of tree link. Core nodes also use the reception of **TREE-CREATE** messages from other cores to decide whether to remain as a core by a distributed core resolution algorithm. The nodes wishing to leave the group send the **JOIN-NAK** to the neighbors and do not forward any data packets for the group.

The key characteristic of AMRoute is its usage of virtual mesh links to establish the user multicast tree. The data packets sent between *logically* neighboring members are *physically* sent on unicast tunnels through potentially many intermediate routers. Figure 2.1 is an example of AMRoute user multicast tree. Therefore, as long as routes between tree members exist via mesh links, the tree need not be readjusted when network topology changes due to node mobility, which reduces tree maintenance overhead. Non-members do not forward data packets and even need not maintain any multicast state or support any multicast protocol. Only the member nodes that form the tree incurs processing and storage overhead. AMRoute relies on an underlying unicast protocols to maintain network connectivity among members, and any general unicast protocol can be used.

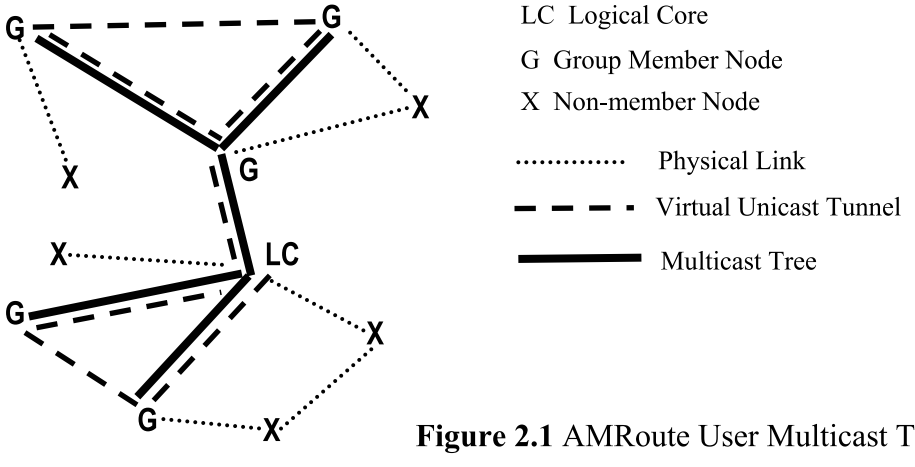


**Figure 2.1** AMRoute User Multicast Tree

Furthermore, certain tree nodes are designated by AMRoute as logical cores, and are responsible for initiating and managing the signaling component of AMRoute, such as detection of group members and tree setup. Logical cores differ significantly from those traditional cores in CBT, since they are not a central point for data distribution, and can migrate dynamically among member nodes.

The major disadvantage of AMRoute is that it suffers from transient loops and data loss during tree reconstruction stage, which may adversely affect its performance. In addition, AMRoute emphasizes robustness even with highly dynamic network topology; the penalty paid for increased robustness by using virtual unicast mesh tunnels is the reduced path efficiency, as non-member routers are not allowed to perform packet replication and forwarding.

### 2.2.3 Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)

AMRIS [33, 34] is an on-demand shared tree multicast protocol. The unique idea behind it is to dynamically assign every node in the network a *multicast session member id* (herein known as *msm-id*) number. The msm-id provides each node with an indication of its "logical height" in the multicast delivery tree. Each node except the root must have one parent that has a smaller msm-id than it. The ordering between msm-id numbers is used to direct the multicast flow, and the sparseness among them used for quick connectivity repair. A multicast delivery tree rooted at a special node called *Sid* who has the smallest msm-id joins up all members in the session. The relationship between the msm-id numbers (and the nodes that own them) and Sid is that the id numbers increase in numerical value as they radiate from Sid in the delivery tree. These id numbers help the nodes to dynamically join and leave a session, as well as adapt rapidly to link connectivity changes. Messages to repair a link breakage are confined to the very region where it occurs. Unlike LAM or AMRoute, AMRIS does not require any underlying unicast routing support.

AMRIS consists of two main mechanisms: *Tree Initialization (TI)* and *Tree Maintenance (TM)*. TI is the mechanism by which a multicast session is launched and advertised to nodes within the network, and a share multicast delivery tree is created for data forwarding. TM is the mechanism whereby nodes that become "detached" from the multicast tree rejoin the tree by executing a Branch Reconstruction (BR) routine. Before TI formally begins, it is necessary to determine which node will assume the role of Sid. In a single-sender, multiple-receiver session, Sid is normally that single sender. In a multiple-sender, multiple-receiver scenario, Sid may be elected from amongst the senders.

Initially, Sid broadcasts a **NEW-SESSION** packet, in which the Sid's msm-id is included. Neighboring nodes, upon receiving this packet, calculate their own msm-id number which are larger than the one specified by Sid. The nodes then rebroadcast the **NEW-SESSION** message with the msm-id replaced by their own. The msm-ids thus increase as they radiate from Sid. Each node is required to broadcast *beacons* to its neighbors. The beacon message contains the node id, its msm-id, membership status, registered parent and children's ids and their msm-ids. A node can join a multicast session by sending a **JOIN-REQ**. This **JOIN-REQ** is unicast to a potential parent node. The node receiving the **JOIN-REQ** sends back a **JOIN-ACK** if it is already a member of the multicast session. Otherwise, it sends a **JOIN-REQ** to its potential parent as well. This process is repeated until a node can satisfy the requirements of being a parent. That node then sends a **JOIN-ACK** which propagates back along the reverse path toward the original joining node. If a node fails to receive a **JOIN-ACK** or receives a

**JOIN-NAK** after sending a **JOIN-REQ**, it performs BR routine. The BR process is executed in an expanding ring search until the node succeeds in joining the multicast session.

AMRIS detects link breakage by a beaconing mechanism. If no beacons are heard for a predefined interval of time, the node assumes that this neighbor has moved out of its radio range. If the former neighbor is a parent, the node must rejoin the tree by sending a **JOIN-REQ** to a new potential parent.

With a single Sid as the root of the multicast delivery tree, however, AMRIS suffers the same “single point of failure” problem as traditional CBT. Furthermore, the periodic beacon flooding from each node greatly increases network overhead, which is more inefficient for large networks.

### 2.2.4 On-Demand Multicast Routing Protocol (ODMRP)

ODMRP [35-37] is an on-demand mesh-based protocol using the concept of *forwarding group* to build a forwarding *mesh* for each multicast group. A soft-state approach is taken to maintain multicast group membership. Member nodes are refreshed as needed and need not send explicit control messages when leaving the group.

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Similar to on-demand unicast routing protocols, a request phase and a reply phase comprise the protocol. When a multicast source has packets to send, it periodically broadcasts to the entire network a member advertising packet, called a **JOIN-QUERY**. This periodic flooding refreshes the membership information and updates the route as follows. When a node receives a non-duplicate **JOIN-QUERY**, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet.

When the **JOIN-QUERY** packet reaches a multicast receiver, the receiver creates or updates the source entry in its *Member Table*. If a valid entry exists in the *Member Table*, a **JOIN-REPLY** is broadcast periodically to the neighboring nodes. When a node receives a **JOIN-REPLY**, it checks if the next node ID of one of the entries matches its own. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the FG_Flag (Forwarding Group Flag) and broadcasts its own **JOIN-REPLY** built upon matched entries. The **JOIN-REPLY** is therefore propagated by each forwarding node until it reaches the source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the *forwarding group (FG)*.

The concept of FG is illustrated in Figure 2.2. The forwarding group is a set of nodes in charge of forwarding multicast data on shortest paths between any member pairs. Only a *forwarding node* (a node belongs to FG) can forward (broadcast) a multicast packet provided that it is not a duplicate. A multicast receiver can also be a forwarding node if it is on the path between a multicast source and another receiver. In Figure 2.2, all nodes inside the "bubble" form the forwarding group for multicast forwarding. The mesh provides richer connectivity among multicast members

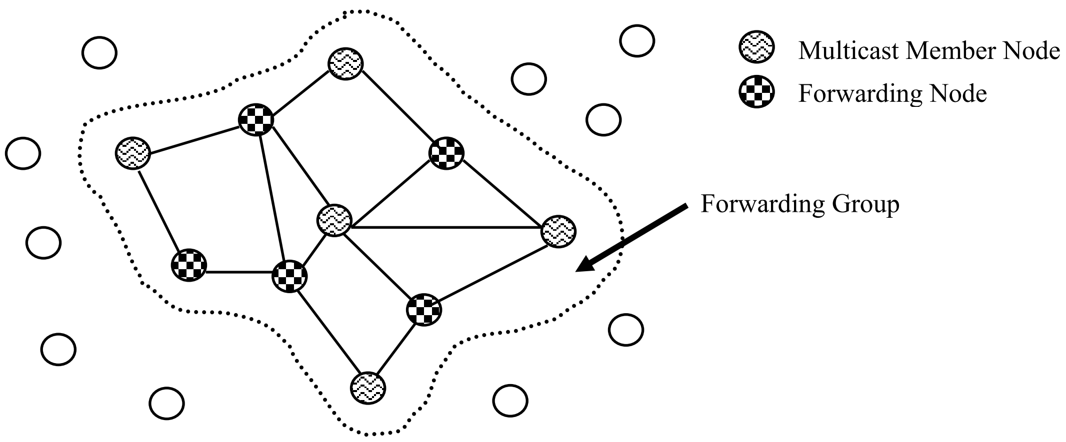


**Figure 2.2** ODMRP Forwarding Group

compared to trees. Flooding redundancy among forwarding group helps overcome node displacements and channel fading. Hence, unlike trees, frequent reconfigurations are not required.

Figure 2.3 is an example to show the robustness of a mesh configuration. Three sources ($S_1$, $S_2$, and $S_3$) send multicast data packets to three receivers ($R_1$, $R_2$, and $R_3$) via three forwarding group nodes (*A*, *B*, and *C*). The forwarding scheme can be viewed as "limited scope flooding". That is, flooding is confined within a properly selected forwarding set. Suppose the route from $S_1$ to $R_2$ is <$S_1$-*A*-*B*-$R_2$>. In a tree configuration, if the link between nodes *A* and *B* breaks or fails, $R_2$ cannot receive any packets from $S_1$ until the tree is reconstructed. ODMRP, on the other hand, already has a redundant route <$S_1$-*A*-*C*-*B*-$R_2$> to deliver packets without going through the broken link between nodes *A* and *B*.

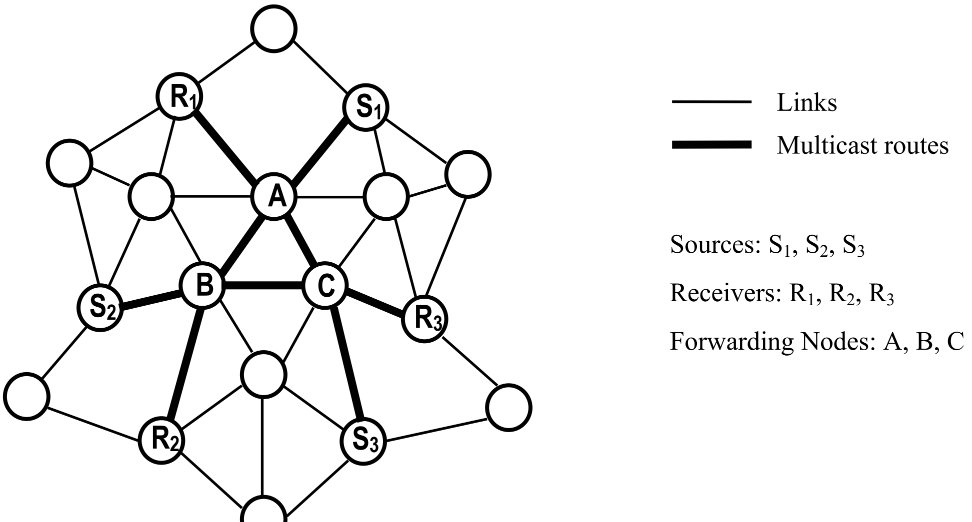


**Figure 2.3** Why A Mesh?

After the forwarding group establishment and route construction process, a multicast source can transmit packets to receivers via selected routes and forwarding nodes. Periodic control packets are sent only when outgoing data packets are still present. When receiving a multicast data, a node forwards it only if it is not a duplicate and the setting of the `FG_Flag` for the multicast group has not expired. This procedure minimizes traffic overhead and prevents sending packets through stale routes.

In ODMRP, no explicit control packets need to be sent to join or leave the group. A *soft-state* approach is applied to maintain multicast group membership. If a multicast source wants to leave the group, it simply stops sending **JOIN-QUERY** packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it removes the corresponding group entries from its *Member Table* and does not transmit **JOIN-REPLY** message for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no **JOIN-REPLY** packets received) before they timeout.

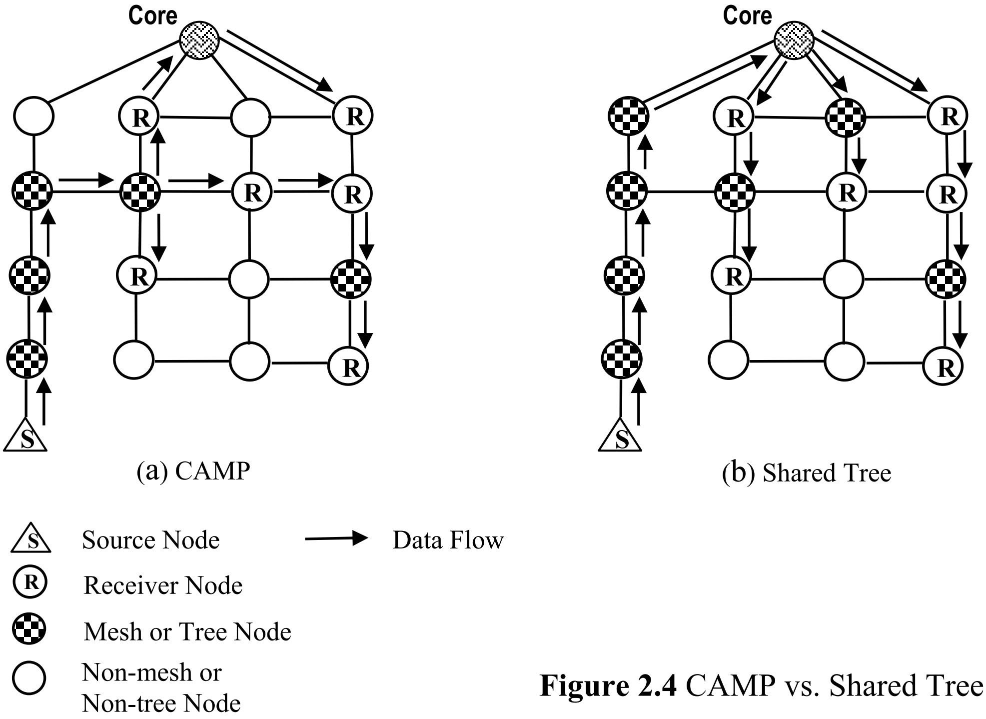
One of the major strengths of ODMRP is its unicast routing capability. Not only can it coexist with any unicast protocol, it can also operate efficiently as a unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol.

However, ODMRP requires network-wide periodic flooding of **JOIN-QUERY** packets to refresh routes and group membership, which is not cost-efficient especially for wireless channels. In addition, soft-state timer values for route refresh interval and forwarding group timeout interval have great impact on ODMRP performance.

### 2.2.5 Core-Assisted Mesh Protocol (CAMP)

CAMP [38, 39] supports multicasting by creating a shared *multicast mesh* structure for each multicast group. Within the mesh, packets from any source in the group are forwarded along the reverse shortest path to the source. CAMP need not first flood the entire network with either data packets (like DVMRP) or control packets (like ODMRP). The cores are used only to limit the traffic required for a router to join a multicast group; the failure of cores does not stop packet forwarding or the process of maintaining the multicast meshes. CAMP assumes the availability of routing information from an underlying unicast routing protocol which must guarantee correct distances to all destinations within a finite time.

CAMP builds and maintains a multicast mesh for information distribution within each multicast group. A *multicast mesh* is a subset of the network that provides at least one path from each source to each receiver in the group. CAMP ensures that the *reverse shortest paths* from receivers to sources are part of the group's mesh and packets are forwarded along these reverse shortest paths. The key difference between a mesh and a shared tree structure is how data packets are accepted to be processed, as illustrated in Figure 2.4. A mesh node is allowed to accept all packets coming from any neighbor in the mesh, as opposed to trees where a tree node can only take packets from its parent on the tree. In addition, CAMP tends to deliver data along shortest paths rather than always going through the core with longer paths.



(a) CAMP

(b) Shared Tree

**Figure 2.4** CAMP vs. Shared Tree.

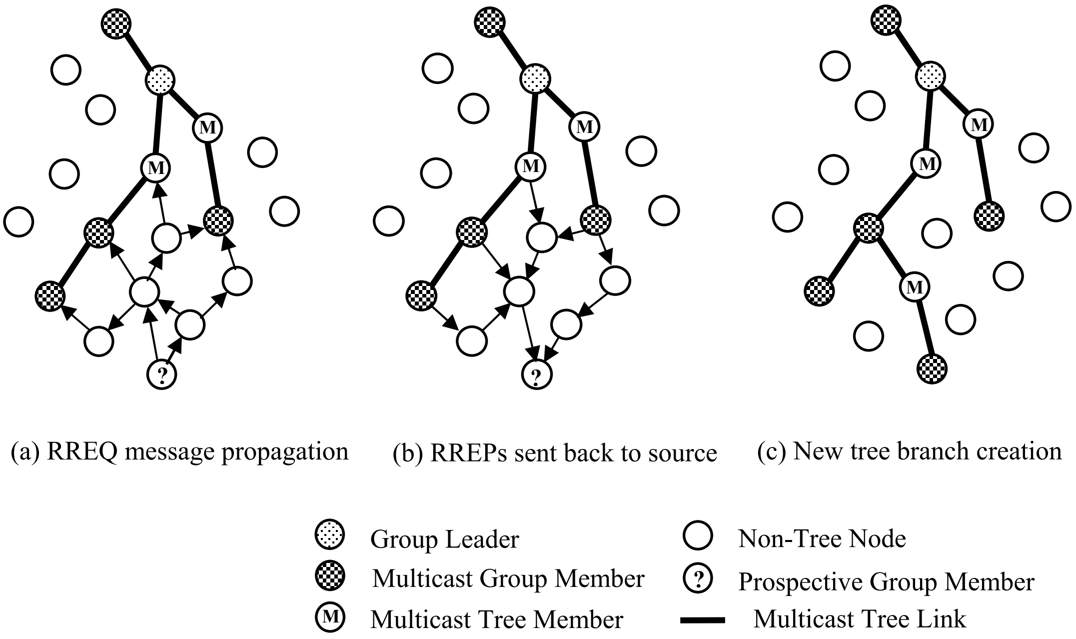
CAMP consists of mesh creation and mesh maintenance procedures. A node wishing to join a multicast group first consults its routing table to determine whether it has neighbors which are already members of the mesh. If so, the node announces its membership via an **UPDATE** message. Otherwise, the node either propagates a **JOIN REQUEST** towards one of the group cores, or attempts to reach a member router by an expanding ring search of broadcast requests. Any mesh member can respond with a **JOIN ACK**, which is propagated back to the source of the request.

Periodically, a receiver reviews its packet cache to determine if it is receiving data from those neighbors which are on the reverse shortest path to the source. If not, the node sends either a **HEARTBEAT** or a **PUSH JOIN** message towards the source along the reverse shortest path. This process ensures that the mesh contains all such reverse shortest paths from all receivers to all senders. The nodes also periodically choose and refresh their selected "anchors" to the multicast mesh by broadcasting **UPDATE**s. These anchors are neighbors which are required to re-broadcast any non-duplicate data packets they receive. A node is allowed to discontinue anchoring neighbors which are not refreshing their status. It can then leave the multicast mesh if it is not interested in the group and is not required as anchor for any neighboring node.

### 2.2.6 Multicast Ad hoc On Demand Distance Vector (MAODV)

MAODV [40, 41] is the multicast extension of Ad-hoc On-Demand Distance Vector Routing (AODV) [45, 46]. It builds a shared multicast tree on demand to connect multicast group members. Control of the multicast tree is distributed so that there is no single point of failure. The most remarkable feature about it is its ability to combine unicast and multicast together in one protocol, so that routing information obtained when searching for a unicast route can also increase multicast routing knowledge, and vice versa. A unique *group sequence number* is generated by a *multicast group leader* for each multicast group in order to prevent loops and detect stale routes.

Route discovery within MAODV is purely on-demand and follows a Route Request (RREQ) / Route Reply (RREP) cycle. A node sends a RREQ message when it wishes to join a multicast group, or when it has data to send to the group but does not have a route to it. If the node wishes to join the group, it sets the *J_flag* of the RREQ; otherwise, it leaves the flag unset. The RREQ may be either broadcast or unicast depending on the information available at the source node. If the source node has a record of the multicast group leader and a valid route to it, it just *unicasts* the RREQ along the known path to the group leader. Otherwise, it *broadcasts* the RREQ. Figure 2.5(a) illustrates the propagation of a broadcast RREQ.



(a) RREQ message propagation (b) RREPs sent back to source (c) New tree branch creation

**Figure 2.5** MAODV Join Operation.

Only a member of the desired multicast tree may respond to a *join* RREQ, while any node with a fresh enough route to the multicast group can respond to a non-join RREQ. If a node receives a RREQ which cannot be replied by it, it rebroadcasts this

RREQ to its neighbors. As the RREQ is broadcast across the network, nodes set up pointers to establish the *reverse path*.

If a node receives a join RREQ for a group, it may reply only if it is a router for that group's multicast tree and its recorded sequence number for the multicast group is *at least as great as* that contained in the RREQ. The responding node updates its route and multicast tables and then generates a RREP. This RREP is unicast along the reverse path towards the original source node. Figure 2.5(b) shows the paths of the RREPs back to the source. As nodes along the reverse path to the source node receive the RREP, they add both a route table and a multicast table entry for the node from which they receive the RREP, thereby creating the *forward path*. Finally, the source node often receives more than one RREP and only one must be selected to avoid loops. The source waits rte_discovery_timeout before selecting a route. During this period, it keeps the received route with the greatest sequence number and the shortest number of hops to the nearest multicast tree member; it disregards other routes. At the end of this period, it enables the selected next hop in its multicast table, and then *unicasts* a Multicast Activation (MACT) message along the forward path to activate the selected path, as illustrated in Figure 2.5(c). Nodes that had generated or forwarded RREPs delete the entry for the requesting node if they do not receive a MACT activating their routes after mtree_build time.

The first member of the multicast group becomes the leader for that group. The *multicast group leader* is responsible for maintaining the *multicast group sequence number* and for disseminating this number to the group. Periodically, the group leader broadcasts a Group Hello message across the entire network. The sequence number for the group is incremented for each Group Hello broadcast by the group leader. Nodes use the Group Hello message to update their request table of that group. Member nodes also use this message to update their current distance to the group leader.

However, MAODV utilizes only *symmetric* links due to the reverse shortest path selection when creating trees. It will fail badly if the underlying links are asymmetric. Moreover, the global broadcast of RREQ and Group Hello messages is not efficient or scalable even though only on demand.

## 2.3 Summary

Although multicasting is already one of the most researched areas in the wired networks, it still remains relatively unexplored in the wireless ad hoc networks due to the challenges of the infrastructureless nature and highly dynamic topology. Not until recently have a few multicast routing protocols been proposed specifically for ad hoc networks. In this chapter, a detailed review of current multicast protocols has been presented, as summarized in Table 2.2.

Generally, the current approaches can be categorized into two classes according to the multicast structure used. One is based on multicast trees, while the other uses mesh. The tree structure can be further grouped into per-source tree and shared tree. The per-source tree does not scale well and is only suitable for one-to-many multicasting. The main disadvantage of the shared tree is the existence of a "single point of failure" at the core and its traffic concentration. Some protocols are essentially hybrid of per-source tree and shared tree in which shared tree (with multiple cores) is only used for maintenance while multicast data is distributed along source tree. However, in the fast mobility scenario, mesh-based protocols still outperform tree-based protocols. The availability of redundant routes provides more resilience to mobility. The main problem with mesh-based algorithms is their global flooding for maintaining the mesh structures and inefficiency when forwarding multicast packets.

**Table 2.2** Summary of Current Multicast Protocols.

| *Protocols* / *Criteria* | *LAM* | *AMRoute* | *AMRIS* | *MAODV* | *ODMRP* | *CAMP* |
|---|---|---|---|---|---|---|
| Configuration | Tree | Tree | Tree | Tree | Mesh | Mesh |
| Dependency on Unicast Routing | Yes | Yes | No | No | No | Yes |
| Routing Metric | Ordered "Height" | Unicast Tunnels | Msm-id number | Reverse Shortest Path | Reverse Shortest Path | Reverse Shortest Path |
| Cores Used | Yes | Yes | Yes | Yes | No | Yes |
| Single Point of Failure | Yes | No | Yes | Yes | No | No |
| Loop-Free | Yes | No | Yes | Yes | Yes | Yes |
| Redundant Paths | Yes | Yes | No | No | Yes | Yes |
| Join Messaging | Unicast | Broadcast | Broadcast | Unicast/ Broadcast | Broadcast | Unicast/ Broadcast |
| Control Message Flooding | No | Yes | Yes | Yes | Yes | No |
| Unique Characteristics | Tightly coupled with TORA | Virtual unicast tunnels, core migration | Msm-id number | Group sequence number | Forwarding group, soft-state approach | No control flooding |

However, current protocols are mainly dealing with how to establish and maintain an efficient and robust multicast structure in face of node mobility. Other important concerns inherent to wireless mobile ad hoc networks, such as bandwidth and power constraints, are actually ignored. Based on this observation, we argue that a final selection of multicast protocol should take into account those specific considerations related to its unique situation and objective.

# CHAPTER THREE

# Power Control in Wireless Ad Hoc Networks

Communication link/path setup (and reconfiguration) and maintenance of the user-required quality of service (QoS) are key functions of network control in any communication network. In wireless networks, these functions are heavily dependent on *transmitter power control* (*power control* for short). In this chapter, we first introduce some background to power control in cellular mobile networks, followed by a discussion about the motivation of power control design in ad hoc networks. Then a brief survey of current power control research is reported, which is mainly concentrated on the MAC layer and network layer. We close this chapter with a primary exploration on the possibility and potential methods to integrate power control considerations with multicasting algorithm designs in order to improve energy efficiency and multicast performance simultaneously. The detailed description of our algorithms will be proposed in the next chapter.

## 3.1 Background and Motivation

### 3.1.1 Power Control Design in Cellular Mobile Networks

Power control design is always of fundamental importance to the proper operation of wireless networks. With efficient power control schemes, it is possible to reduce power consumption and prolong the battery life of the mobile nodes. Moreover, by adjusting transmission power, network capacity can be increased significantly as a result of mitigated network interference and improved channel spatial reuse [15-17]. Power control can also be used to implement various basic dynamic network operations online, such as admission control, link QoS maintenance, channel selection and switching, resource allocation, and handoff control, etc [18].

Along with the proliferation of cellular networks in 1990s, power control design, as an important means for providing good transmission quality and expanding wireless network capacity, receives more and more research attention. Numerous commercial power control technologies for cellular mobile networks have been developed in the last decade. Early work is focused on balancing the signal-to-interference ratios (SIRs) of all network users, globally lowering them as the network became congested. Recently, the interest has been on adjusting transmitter power to maintain a required SIR threshold for each network link using the least possible power.

In cellular networks, power control is needed for both forward channels/downlinks and reverse channels/uplinks. As there is a pilot channel on the forward link, power control is considerably less complicated to implement on forward links than on reverse links. The base station just generates the power control bit and inserts it on every forward channel to instruct individual mobile units to adjust their transmission power level. Power control for CDMA is more crucial but more sensitive and intractable on the reverse links in the sense that power levels received at the base station from different mobiles are nearly equal and no single mobile can dominate, so that a mobile unit close to the base station can mask the received signals from far-end mobiles and even prevent recovery of these signals at the base station.

Power control schemes in mobile cellular networks can be either *centralized* or *distributed*. A centralized mechanism is reportedly capable of achieving optimal performance, although it is costly and unscalable for large networks. A distributed mechanism, on the contrary, does not provide the best solution, but it is less costly to operate. According to the information measured to determine whether to raise or lower the transmitter power of a mobile unit, power control can also be divided into *strength-based* and *SIR-based*. In a strength-based scheme, the strength of a signal arriving at the base station from a mobile unit is measured to determine proper power control action, whereas in SIR-based design, the quantity measured at the base station is SIR value, with the interference consisting of both channel noise and multiuser interference. Obviously, the SIR-based scheme can measure the signal quality more accurately and therefore provides better performance than the strength-based scheme.

### 3.1.2 Motivation of Power Control Design in Ad Hoc Networks

Compared with cellular networks, power constraints are much stricter under the new scenario of ad hoc networks. Many mobile units will be powered by batteries with limited lifetime; some of them may not even be rechargeable. For networks consisting of power-constrained nodes, the most important system design criteria for optimization may be *power conservation*, which is mainly ignored currently and therefore becomes the objective of our study.

However, power control research within ad hoc networks still remains at the very infancy stage. A number of complications and challenges are introduced by the autonomy inherent in wireless ad hoc networks.

Firstly, unlike conventional cellular mobile networks, which have a fixed base station to centrally manage and facilitate power control of all mobile units inside each cell, there is no such centralized control entity in an ad hoc network to carry out the similar management functions. Efficient distributed power control algorithms are required to be run at each mobile node.

Secondly, highly dynamic topology and arbitrary node mobility make power measurements and interference estimations far less accurate and much more complex, which in turn causes power control designs and implementations to be more difficult in wireless ad hoc networks.

Thirdly, although power control is traditionally studied only at the physical and link layer, it now has a significant impact on higher level protocol design within the very particular paradigm of ad hoc networks, which has mostly been overlooked before. For instance, the level of transmitter power defines the "*local neighborhood*" — the collection of nodes that can be reached in a single hop — and thus in turn defines the context in which access, routing, and other higher-layer protocols operate. Moreover, power control introduces *asymmetric* links, where one station can transmit to another, but the latter cannot directly reply to the sender due to its coverage limitation. Unlike cellular mobile networks, which are basically only the last-hop extensions of wired networks, mobile ad hoc networks are essentially *multihop*. Therefore, varying network topology and asymmetric links induced by power control complicate the design of higher-layer protocols, such as routing and multicasting, for wireless ad hoc network.

On the other hand, power control is more favorable and crucial for ad hoc network designs. Besides the conventional benefits for wireless networks, such as reduced interference, better channel utilization and increased network capacity, power control can significantly enhance the performance of MAC and higher layer protocols.

For example, popular MAC protocols in mobile ad hoc networks, such as Multiple Access with Collision Avoidance (MACA) [19], Multiple Access with Collision Avoidance Wireless (MACAW) [20], Dual Busy Tone Multiple Access (DBTMA) [21], and the widespread IEEE 802.11 standard [22] MAC protocol, are all using a *contention-based* mechanism in which multiple users contend for a *single shared channel* when they have packets to send and some *collision avoidance* schemes are provided to reduce the possibility of packet collisions. Under the assumption of "*commutativity of transmission*", all these MAC protocols have typically used fixed power level to transmit, without taking into account the possibility of power control implementation. On the other hand, adaptively changing the coverage area of a transmitter-receiver pair, according to their relatively physical distance, can lead to less interference and more efficient spatial channel reuse, and greatly alleviate the well known hidden and exposed terminal problems [23] which are inherent in shared channel wireless medium and the main cause of low channel utilization in mobile ad hoc networks today. Figure 3.1 illustrates how power control can be applied to eliminate the exposed terminal problem. At first, nodes A, B, C, and D all use fixed maximum power level (dashed cycle coverage) to transmit, i.e., no power control scheme is used. If node B is transmitting to A, node C could not transmit to D at the same time since C would hear the RTS from Node B and defer transmission even though it would not have led to any collision at the intended receiver D. However, if B reduced its transmission power such that it would be just enough for A to capture its signal (solid cycle coverage), then C could also proceed with its transmission without any deferment. Clearly, power control is crucial and desirable for improving the performance of existing MAC protocols in ad hoc networks.

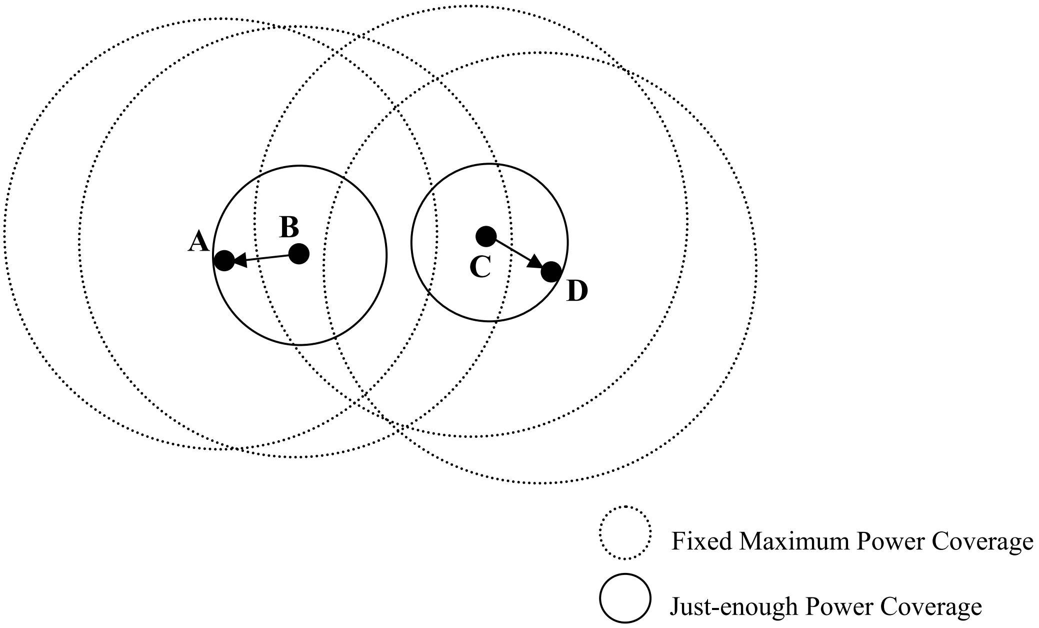


**Figure 3.1** Eliminating the Exposed Terminal Problem by Power Control.

Another interesting area is on the network layer. "*Neighbor discovery*" is one of the first steps in the initialization of a network of randomly distributed nodes. From the perspective of the individual node, this process involves determining the number and identity of network nodes with which *direct communication* can be established given some maximum transmission power level and link performance requirements. Clearly, the higher the allowed transmit power, the greater the number of nodes in a given neighborhood. Therefore, the level of transmit power defines the "*local neighborhood*" — the collection of nodes that can be reached *in a single hop*. The network topology now depends not only on those *uncontrollable* factors such as node mobility, interference and noise, etc., but also on a *controllable* parameter: transmitter power level, even without any node movement. This observation motivates the studies of some higher-layer optimization problems recently, such as *minimum-power routing problem* and *optimal power range assignment problem*, which will be elaborated in the next section. Generally, these joint considerations of power control and higher-layer protocol design can yield considerable performance improvement.

As a conclusion of this section, we believe that power control is more crucial and desirable for wireless ad hoc networks, although it is more challenging and sophisticated. We also want to point out specifically the fact that power control has significant impact on protocols above the link layer. The inter-dependencies among multiple layers induced by power control is so prominent that we argue that traditional OSI or TCP/IP layered architecture may not be suitable or accurate any more for the proper description of mobile ad hoc networks. A holistic cross-layer protocol design that supports adaptivity and optimization across multiple layers may be required and would likely mark a milestone in the development of this field [24].

## 3.2 Related Work

In this section, we present a brief survey on some power control related work for wireless mobile ad hoc networks. Basically, these works are mainly concentrated on MAC layer and network layer.

### 3.2.1 Power Control in MAC Layer

Currently, only a few researches [47-49] have gone into power control in MAC layer. In [47], Monks, et al. discuss some issues involved in implementing power control in MANETs. Two basic principles, power conserving principle and cooperation principle, are proposed. And an ideal power controlled (IPC) MAC protocol is defined assuming the availability of perfect (global) knowledge of the link gain between any two nodes, the noise at any potential destination, and the maximum transmission power that will not corrupt neighboring nodes' reception of their own incoming packets. The IPC protocol follows the power conserving principle by reducing the transmission power to be only slightly more than needed to reach the intended destinations and the cooperation principle by backing off transmissions if needed power is greater than the power upper bound. Simulations have demonstrated that IPC can allow for much more simultaneous senders than IEEE 802.11b by adjusting the transmission range to the minimum required to satisfy successful reception at the intended destinations. The authors then extend their IPC description to real implementation and propose a power controlled multiple access (PCMA) protocol [48]. PCMA uses signal strength of a received control message to limit the transmission power of the hidden and exposed stations. This control message is a "generalized version of CTS", which is a signal pulse in the "busy tone" channel. However, its reception by a hidden station does not preclude this station's transmission. Instead, each hidden station constrains its transmission power by a function of the received signal strength of the control packet.

Wu, et al. [49] explored the possibility and effects of combining the concept of intelligence power control with the RTS/CTS-based and busy-tone-based MAC protocols to further increase channel utilization of ad hoc networks. In their new protocol, a sender uses an appropriate power level to transmit so as to increase channel reuse. More specifically, data packet and transmit busy tone ($BT_t$) packet are transmitted with power control based on the power level of the received CTS packet, CTS and receive busy tone ($BT_r$) packet are transmitted at the normal (largest) power level, and RTS packet is transmitted at a power level determined by how strong the $\mathrm{BT_r}$ packets are around the requesting host. Analyses and simulations have proven the advantage of the new protocol over non-power-controlled DBTMA.

However, for these three protocols, IPC is not a real protocol as it assumes the availability of global knowledge of link gains and ambient noise; while the protocols in

[48, 49] require the single channel to be split into two channels — data channel and busy tone channel, which may not be bandwidth economical.

A new MAC protocol named Power-Aware Multiple Access protocol with Signaling (PAMAS) [50] has been recently proposed. The uniqueness of PAMAS is its ability to intelligently turn off radios when they need not transmit or receive packets. The simulation results yield up to 40-70% of battery power savings compared to MAC protocols with radios always on.

### 3.2.2 Power Control in Network Layer

Power control research in network layer is motivated by the observation that varying transmit power level is able to dynamically create or destroy network links and thereby change the total connectivity of wireless ad hoc networks. Much research attention has been attracted by this interesting observation and numerous proposals are reported recently. Generally, current research can be further categorized into two directions, one is optimal topology design problem with power control [51-53], and the other is power-aware routing problem [54-60].

Given a network of randomly distributed nodes and some maximum transmit power level and link performance requirements, the *power-optimal topology design problem*, or *minimum-power full-connectivity problem*, is aimed at maintaining *full connectivity* whereby each node in the network can reach every other node, often through multiple hops, using the *minimum transmit power* at each node. A distributed location-based network protocol optimized for minimum energy consumption in mobile ad hoc networks is reported in [51]. Equipped with a global positioning system (GPS), a simple local optimization scheme executed at each node can guarantee strong connectivity of the entire network and attains the global minimum energy solution for stationary wireless networks. Due to its localized nature, this protocol proves to be self-configuring and stays close to minimum solution when applied to mobile scenario. The *Min-Power Symmetric Connectivity problem* is proposed in [52, 53], where a link is established only if *both* nodes have transmission range at least as big as the distance between them, and the goal is to ensure that the network is fully connected.

There has been much recent work on power aware routing problem within ad hoc networks. Early works [54-57] mainly address *minimum-power routing problem*. Based on the assumption that nodes have the knowledge of location of their neighbors and can adjust their transmission power, the approach in those works is to *minimize the total consumed power* to reach the destination, by replacing the traditional constant routing metric (hop count) with a *power metric* that depends on distance between nodes. However, if all traffic is routed only through the minimum power path to the destination, the nodes in that path may be drained out of batteries quickly and make the whole network crash; while other nodes, which perhaps may be more power hungry if traffic is routed through them, will remain intact. Therefore, recent interests are on the problem of *maximizing the lifetime of the whole system* (the time till network partitions) or minimize the maximum power spent at each node. Five different power-aware metrics based on battery power consumption is presented in [58] to increase node and network lifetime. It is reported that using these power-aware routing metrics in a shortest-cost routing algorithm can reduce the cost of routing packets by 5-30% over general shortest-path routing. Chang and Tassiulas [59] formulated the routing problem with the objective of maximizing the system lifetime and present a class of flow augmentation algorithms and a flow redirection algorithm which balance the power consumption rates among the nodes in proportion to their power reserves. The proposed algorithms are localized and amenable to distributed implementation and shows close to the optimal performance most of the time, improving the system lifetime by as much as 60% on the average over the minimum-power routing. Ivan and Xu [60] define a new *power-cost metric* based on the combination of both node's lifetime and distance-based power metric. The combined power-cost localized (where each node makes routing decisions solely on the location of itself, its neighbors, and the destination) routing algorithm attempts to simultaneously minimize the total power consumed and extend battery's worst case lifetime at each node.

In summary, current research of power control in wireless ad hoc networks is not limited to physical and link layer only. On the contrary, some cross-layer experiments of power control with higher-layer protocol designs produce inspiring and promising results and attract much interest in the research community. Specifically, some interesting problems associated with network layer have been motivated recently. Generally, power control research on network layer often involves intractable combinational optimization problems for any sizable network; one has to search for simple and justifiable heuristics to obtain good practical solutions.

## 3.3 Multicasting with Power Control: A Primary Exploration

As reviewed in the last chapter, most up-to-date multicast routing protocols for wireless ad hoc networks are mainly dealing with the problem of establishing and maintaining efficient multicast structures in the face of node mobility. The performance metrics commonly used are packet delivery ratio (PDR), routing overhead, average end-to-end delay, etc. The implicit assumption under them is that node transmission power is fixed and no power control scheme is utilized.

On the other hand, in contrast to traditional cellular wireless networks, power capacity becomes a very constrained and precious resource for ad hoc networks. Many nodes may be powered by batteries with limited lifetime, some of which may not even be rechargeable. For those power-constrained networks, an important issue in routing and multicasting now is to conserve as much power as possible while still achieving good link quality. Power-aware unicast routing problem has been studied recently. Some power-related metrics are proposed as routing metric to either minimize the total consumed power on the path to the destination or maximize the lifetime of the whole network. Very promising results are reported in terms of power conservation and routing performance improvement. However, research on multicasting with power control is mostly overlooked. How a multicast routing algorithm can be designed to integrate with power control mechanism therefore becomes our research interest.

### 3.3.1 Wireless Nodal Propagation Property

The wireless communication channel is distinguished by its *broadcast* nature: when omni-directional antennas are used, every transmission by a node can be received by all nodes that lie within its covering range. We call it *wireless nodal propagation (WNP)* property. Consequently, as far as transmit power is concerned, the total power required by a node to reach a set of its neighboring nodes is simply the maximum required to reach any of them individually. Consider the example shown in Figure 3.2, in which a subset of the multicast tree involves Node $i$, which is transmitting to some of its neighbors, Nodes $j$, $k$, and $l$. The power required to reach Nodes $j$, $k$ and $l$ are $p_{ij}$, $p_{ik}$, and $p_{il}$, respectively ($p_{ik} \le p_{ij} \le p_{il}$). Under our assumption of WNP property, a single transmission at power level $p_{i,(j,k,l)} = \max(p_{ij}, p_{ik}, p_{il}) = p_{il}$ is sufficient for Node *i* to reach Node *j*, *k* and *l* simultaneously.

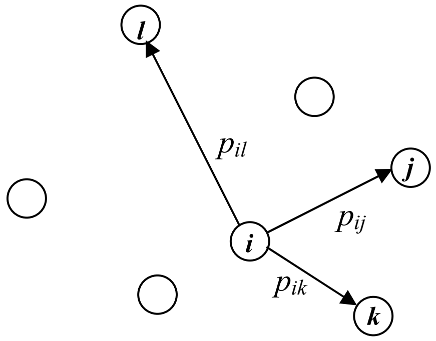


**Figure 3.2** Wireless Nodal Propagation Property: $p_{i,(j,k,l)} = \max(p_{ij}, p_{ik}, p_{il}) = p_{il}$

Therefore, when considering transmission power consumption under a broadcasting or multicasting scenario, wireless networks can be characterized as "*node-based*" environment [63]. In sharp contrast, in wired networks, as long as there is one wire link connecting two nodes, the transmission and reception is ensured only over that link and the power of Node *i*'s transmission to Nodes *j*, *k* and *l* would be the sum of the power to the individual node, i.e., $p_{i,(j,k,l)} = p_{ij} + p_{ik} + p_{il}$. Thus, wired networks can be viewed more accurately as "*link-based*" networks.

The unique WNP property makes multicasting in wireless ad hoc networks an excellent and interesting scenario to study the potential benefits of power-efficient algorithms and protocols. Since the wireless propagation loss varies *nonlinearly* with distance (at somewhere between the second and fourth power) [65], in *unicast* applications and from the perspective of *transmission power consumption*, it is best to transmit at the lowest possible power level, even though doing so requires multiple hops to reach the destination. However, in *multicast* applications, it is not prudent to draw such a conclusion a priori, because the use of higher power may permit simultaneous connectivity to a sufficiently large number of nodes, so that the total power required to reach all members of the multicast group may be actually reduced. Essentially, increasing power range has many benefits. The number of links on the multicast tree is reduced due to longer average path length, resulting in fewer tree link breaks. Since each multicast tree link repair requires control message overhead, reducing the number of repairs has the advantage of decreasing the amount of control overhead. However, a large transmission range also causes more network nodes to be affected by multicast data transmission, even when the nodes do not need to receive these packets, which is especially undesirable for sparse membership distribution case. A large transmission radius not only drains the batteries of the transmitting node, but also of all neighboring nodes within the source's coverage range. Worse, a large transmission radius reduces the effective bandwidth available to the individual nodes and increases the number of collisions seen throughout the network, as more nodes are contending for and utilizing the same network bandwidth. Obviously, there exists a tradeoff between reaching more nodes in a single hop by using higher transmission power but at a higher interference cost versus reaching fewer nodes by lower power but may lead to longer routes.

### 3.3.2 Related Work

As far as we know, only a few articles have recently addressed the problem of multicasting with power control for wireless ad hoc networks. In [61], the authors of AODV [45, 46] and MAODV [40, 41] study the effects of transmission power range on MAODV by examining the results achieved at varying transmission ranges and network configurations. Through intensive simulations, it is reported that transmission range is a key determinant of MAODV's performance. For unloaded networks, the packet delivery ratio (PDR) increases for larger power ranges due to the reduction in the number of hops between group members and the longer-lived tree links. However, increasing transmission range also leads to the increase in the number of collisions and interference seen by individual nodes, which causes a reduction in the PDR for loaded traffic patterns. Therefore, the existence of the tradeoff between being able to reach group members in a smaller number of hops, and keeping the set of nodes affected by multicast data transmission to a minimum, is verified. Finally it is concluded that the transmission range should be adjusted to meet the targeted throughput while minimizing battery power consumption. There are opportunities for power savings when nodes can get the same (or even better) performance by reducing the power drain caused by unnecessarily high transmission power ranges. Although this work does not give any hint on the potential methods to implement power-controllable multicast routing algorithms or protocols, it does provide some basic guideline to our designs of power-efficient multicasting algorithms, which will be elaborated in the next chapter.

The work in [62, 63] has been the first to address the problem of minimum-energy broadcasting in a static wireless ad hoc network, in which source-initiated, circuit-switched multicast sessions are investigated. The impact of the wireless medium on the broadcasting and multicasting and the fundamental tradeoffs are discussed. And a node-based network model is proposed. We base our work on this so-called “node-based” network model. Very recently, Cagalj, et al. have provided a formal proof of NP-hardness for the minimum-energy broadcast problem in all-wireless networks [66], based on the same node-based model. Three different integer programming models, which can be used for an optimal solution of the minimum power broadcast problem in wireless networks, have been presented by Das, et al. in [67]. Banerjee, et al. [68] have defined energy-efficient broadcast and multicast schemes for reliable communication in multihop wireless networks. Considering the error-prone nature of wireless channels, a retransmission-aware cost function has been designed based not only on the link distance, but also on the error rates associated with the link.

## 3.4 Summary

Power control is always of fundamental importance to the proper operation of wireless networks. In this chapter, we first introduce some background of power control in cellular mobile networks. The challenges and motivations of power control design in wireless ad hoc networks are discussed, and a brief survey of current power control research is reported, which is mainly focused on the MAC and network layer. The fact that some cross-layer investigations of power control with higher-layer, especially network layer, protocol designs yield promising results really motivates our research interest on the construction of power-efficient multicasting algorithms for ad hoc networks. We finally present a primary exploration on the possibility and potential methods to integrate power control considerations with multicast routing algorithm designs. Specifically, the unique wireless nodal propagation concept is highlighted, its impact on power-tunable multicasting problem is discussed, and a few related works are examined. In the next chapter, a class of power-efficient multicasting algorithms will be proposed and evaluated by simulations.

# CHAPTER FOUR

# On the Design of Power-Efficient Multicasting Algorithms for Wireless Ad Hoc Networks

Wireless ad hoc networks often consist of many battery-powered nodes with limited lifetime. For those power-constrained networks, a crucial issue in routing and multicasting is to conserve as much power as possible while still achieving good throughput performance. In this chapter, we first introduce some preparation knowledge related to our algorithm design. After problem formulation, a class of minimum-power multicasting heuristics are proposed and illustrated carefully, and their simulation results demonstrate much more power efficiency improvement compared to algorithm without power control. We further design a new cost function jointly optimizing power conservation and path efficiency and apply it to those heuristic algorithms to improve their path efficiency performance. Finally we extend our discussion to a full mobile environment with a detailed two-phase multicast tree maintenance algorithm.

## 4.1 Preparation Knowledge

### 4.1.1 Some Definitions [30, 64]

An *undirected graph* $G=(V,E)$ consists of a nonempty set $V$ of $v$ *vertices* and a set $E$, $E \subseteq V \times V$ of $e$ *edges* connecting pairs of vertices. An *undirected network* $G=(V,E,c)$ has $V$ and $E$ as in undirected graphs. In addition, it has a length or cost function $c$ associated with the edges.

An edge $e_l$ between a pair of vertices $v_i$ and $v_j$ is denoted by $(v_i, v_j)$. The vertices $v_i$ and $v_j$ are the *end-vertices* of $e_l$. The number of edges incident with a vertex $v_i$ in a network $G$ is called the *degree* of $v_i$, denoted by $deg_G\ (v_i)$ or $deg\ (v_i)$ if there is no danger of confusion.

A network is said to be *connected* if it has a path between every pair of vertices. Otherwise it is said to be *disconnected*. A network with every pair of vertices being adjacent is said to be *complete*.

For every edge $e_l=(v_i,v_j)$ in $G=(V,E,c)$, its *length* or *cost* $c(e_l)$ is also denoted by $c(v_i,v_j)$, $c_{v_i v_j}$ or $c_{ij}$. A network $H=(W,F,c_F)$ is called a *subnetwork* of $G=(V,E,c)$ if $W \subseteq V$, $F \subseteq E$ and $c_F(e_l)=c(e_l)$ for all $e_l \in F$. The *length* or *distance* of any subnetwork $H$ of $G$ is defined by $\sum_{e_l \in F} c(e_l)$ and denoted by $|H|$ or $c(H)$.

A *minimum length path* or *shortest path* $P_G(v_i,v_j)$ between two vertices $v_i$ and $v_j$ in $G$ is also denoted by $P(v_i,v_j)$, $P_{v_i v_j}$ or $P_{ij}$. Its total length $|P_G(v_i,v_j)|$ is also denoted by $d_G(v_i,v_j)$, $d_{v_i v_j}$ or $d_{ij}$. A complete network $D_G(W)$ with $W, W \subseteq V$ as its vertex set, and with $d_G(v_i,v_j)$ for each pair of vertices $v_i, v_j \in W$ as the length of the edge $(v_i, v_j)$, is called a *complete distance network* of $W$ in $G$.

Given an undirected connected network $G=(V,E,c)$, a *tree* is a subnetwork of $G$ that contains no cycles. *Network distance* is defined as follows. The *distance between two nodes* is the distance of the shortest path between them. Likewise, the *distance* *between a node and a tree* is the minimum among the shortest paths between the designated node and every node in the tree, and the *distance between two trees* is the minimum distance between a node in one tree and a node in the other.

### 4.1.2 Minimum Spanning Trees

A *spanning tree* of *G* is a subnetwork of *G* that is a tree and that includes all the vertices in *G*. A *minimum spanning tree* (MST) [64] is a spanning tree with the minimum sum of edge lengths. Any subtree of an MST is called a *fragment*. It can be formulated as follows:

- GIVEN: An undirected connected network $G = (V, E, c)$ where $c$ is an edge cost function.
- FIND: A subnetwork $T_G(V)$ of *G* such that:
  - $T_G(V)$ is a spanning tree of *G*,
  - Total length $|T_G(V)| = \sum_{e \in T_G(V)} c(e_l)$ is minimized.

There are mainly two methods to construct an MST. *Prim's algorithm* starts with an arbitrarily selected single vertex as a fragment and enlarges the fragment by successively adding a minimum length outgoing edge. While *Kruskal's algorithm* starts with each vertex being a single fragment. It then successively combines two fragments by using the edge that has the minimum length over all edges that when added to the current set of fragments do not form a cycle. Both of these algorithms terminate in $v-1$ iterations, where *v* is the number of vertices in *G*. And the worst-case time complexity of them is $O(v^2)$.

### 4.1.3 Steiner Tree Problem

The *Steiner problem* in networks [29, 30] can be formulated as follows:

- GIVEN: An undirected connected network $G = (V, E, c)$ where $c$ is an edge cost function, and a non-empty set $N, N \subseteq V$ of *terminals*.
- FIND: A subnetwork $T_G(N)$ of G such that:
  - There is a path between every pair of terminals,
  - Total length $|T_G(N)| = \sum_{e_l \in T_G(N)} c(e_l)$ is minimized.

The vertices in set $V - N$ are called *non-terminals*. Non-terminals that end up in $T_G(N)$ are called *Steiner vertices*. The subnetwork $T_G(N)$ is called a *Steiner minimal network* for N in G. If all edges in G have positive length, $T_G(N)$ must be a tree. The problem is therefore often referred in the literature as the *Steiner tree problem*, and $T_G(N)$ is called a *Steiner minimal tree* (SMT) or *Steiner tree* (ST) for N in G. In particular, $T_G(N)$ denotes a minimum spanning tree for G.

It is well known that ST problem is *NP-complete*. In other words, finding the minimum Steiner tree in a graph has an exponential cost for a result that is not necessarily optimal. It has been shown that the minimum cost of a ST algorithm is $O(v \log v)$, where v is the number of nodes in the network and with all distances equal to unity on the links. There exist various *exact algorithms* for the ST problem. While these approaches may produce good, even optimal Steiner trees, they often involve substantial computational effort. ST *heuristics*, in contrast, are relatively inexpensive and can also produce very good solutions. For this reason, a number of good, inexpensive heuristics exist for the ST problem [29, 30].

## 4.2 Problem Formulation

### 4.2.1 Network Modeling and Assumptions

We model our interested scenario as an *undirected connected network* $G = (V, E, c)$ as defined in section 4.1.1, where vertices set $V$ corresponds to the set of all nodes in the network, edge set $E$ corresponds to the set of links between pairs of nodes, each edge $e_l \in E$ has a cost function $c(e_l)$ representing the link cost, a non-empty set $N, N \subseteq V$ of terminals corresponds to the multicast group source and member destinations, in which there is a source node $S$ and others are all group members, denoted as a set $D$, and $V - N$ is the set of non-group-member nodes. We also use the following notations for further discussion: the symbols $v$, $n$ and $s$ represent the number of all network nodes, multicast members and non-members respectively. $P_{ij}$ is the shortest path between nodes $i$ and $j$, and $d_{ij}$ is the distance of the shortest path $P_{ij}$.

Because power consumption is our very concern, and motivated by the power-aware unicast routing problem discussed in the last chapter, we choose power level as the link cost function, i.e.,

$$c(e_l) = p(e_l),\ p(e_l) > 0 \text{ for all } e_l \in E \tag{4.1}$$

where $p(e_l)$ is the power level required to support the *bidirectional* link (therefore, *symmetric* power requirement). We only consider *transmission power* consumption, ignoring the receiving and signal processing part. For the purpose of power conservation, we assume each node can dynamically and independently adjust its transmission power, not exceeding a maximum value, based on the *distance* to the receiving node (we ignore the background noise and channel fading). In the most commonly used power-attenuation model [65], the signal power falls as $1/d^{\kappa}$, where $d$ is the *T-R separation distance* and $\kappa$ is a real *constant* dependent on the wireless environment, typically between 2 and 4. For simplicity, we take its value as 2, which corresponds to conventional *free space* propagation model [65]. Another common assumption is that all receivers have the same power threshold for signal detection, typically normalized to 1. With these assumptions, the transmission power required for supporting an undirected link $e_l$ separated by a distance $d$ is

$$p(e_l) = d^2 = c(e_l), \text{ for all } e_l \in E \tag{4.2}$$

It is also assumed that each node knows the relative distance to their neighbors, and thus the appropriate transmit power (according to eq. (4.2)) to reach each of them. The distance between neighboring nodes can be estimated on the basis of incoming signal strengths through some control message exchanges, such as RTS/CTS packets used in MAC protocols [49]. Alternatively, the location of nodes may be available by using GPS [60] if nodes are equipped with a small low power GPS receiver.

Moreover, the tunable power levels used at a mobile node have the effect of dynamically creating or destroying links, so that the network topology can be changed even without any node movement. To be more focused on the power-tunable multicasting algorithms, we assume therefore that there is no mobility for all nodes, i.e., a *static* wireless ad hoc network.

Finally, we assume *omni-directional antennas* are used, and the *wireless nodal propagation* property, as discussed in section 3.3.1, is applied. That is, a transmission by a source node can be received by all nodes within its transmission range. This feature is extremely useful and important for our power-efficient multicasting designs. More specifically, for a source Node *i* with a set *C* of neighboring nodes and a set *R* of recipient neighbors, $R \subseteq C$, the set $C - R$ are those neighbors who do not want to receive data from Node *i*, and the number of its neighbors and recipient neighbors are *c* and *r*, respectively, the transmission power or cost for one-hop multicasting from Node *i* to all nodes in recipient set *R* is

$$p_i = c_i = \max[c(e_{ij})] = \max[p(e_{ij})] \text{, for all } j \in R \tag{4.3}$$

### 4.2.2 Problem Formulation

Based on the above modeling and assumptions, now we are interested in constructing a minimum power one-source multicast tree for a static wireless ad hoc network. We formulate our problem in graph theoretical term as follows:

- GIVEN: An undirected connected network $G = (V, E, c)$ where $c$ is an edge cost function: $c(e_l) = p(e_l) = d^2$ for all $e_l \in E$, and a non-empty set *N*, $N \subseteq V$ of multicast group members, in which there is a source node *S* and the rest are all group destinations, denoted as a set *D*.
- FIND: A subnetwork $T_G(M)$ of *G*, consisting of a set *M* of nodes, $M \supseteq N$, such that:
  - $T_G(M)$ is a *multicast tree* rooted at *S* and spanning all nodes in set *D*,
  - Total cost $|T_G(M)| = \sum_{i \in M} c(e_i)$, (*i* is a transmitting node in *M*, and eq. (4.3) is applied to *i*) is minimized.

In other words, the total cost (power consumption) of the multicast tree is simply the sum of the transmission power expended at each of the *transmitting nodes* in the tree; *leaf nodes* which do not transmit do not contribute to this quantity.

## 4.3 Minimum-Power Multicasting Algorithms

If we deal with our problem by traditional wired "link-based" approach, it is just the standard Steiner tree problem. However, considering the WNP property, it differs a lot from Steiner tree problem. Currently, we do not know about any solution to this "node-based Steiner tree problem", though it appears to be at least as difficult as standard Steiner tree problem. Hence, heuristics are definitely needed. In this section, we propose a class of heuristics for constructing the Minimum-Power multicast tree. For better illustration, we describe our algorithms by presenting a simple example of tree construction for a 10-node network, as shown in Figure 4.1, where 10 nodes are randomly distributed in a $50m \times 50m$ square area, the maximum radio range $p_{\max}$ is 25m, Nodes {0, 3, 4, 6, 8, 9} are the set of a *multicast group*, in which Node 9 is the *source*, and Nodes {0, 3, 4, 6, 8} are the set of multicast *destinations*, the edges in the graph represent direct connections between pairs of nodes, and the values adjacent to edges (note that for the effect of illustration, the values have nothing to do with the exact length of each edge) represent the cost of the links, i.e., the symmetric power requirement to support that link, e.g., the edge between Node 0 and 1 represents that a bidirectional link can be established between them using transmit power 297, while Node 9 cannot communicate with Node 1 directly due to its coverage range limit.

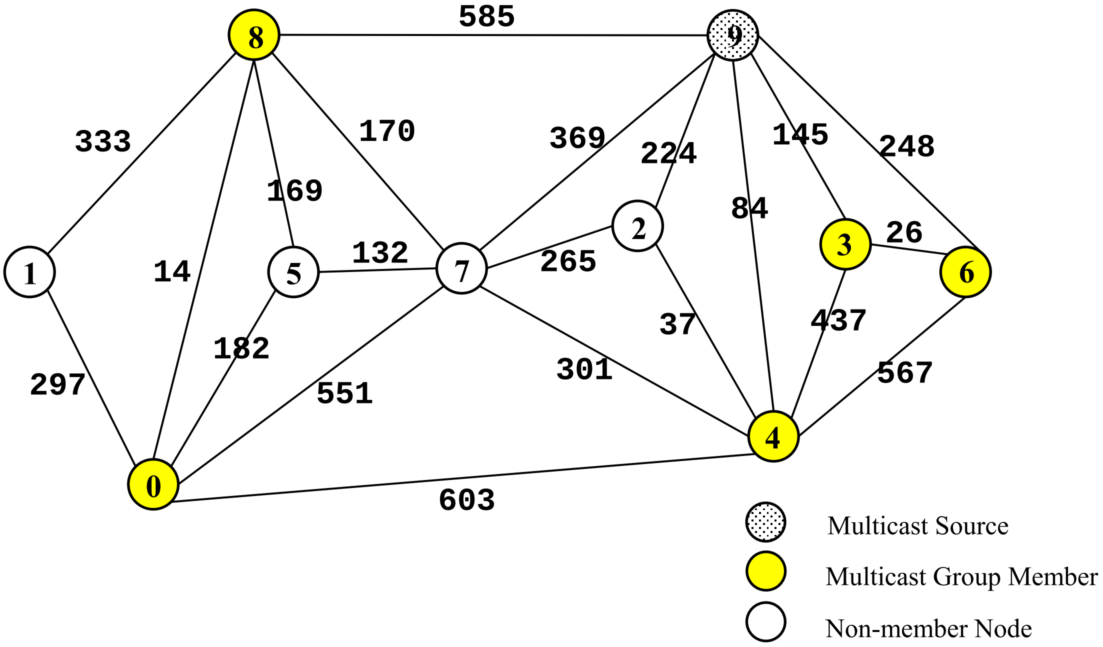
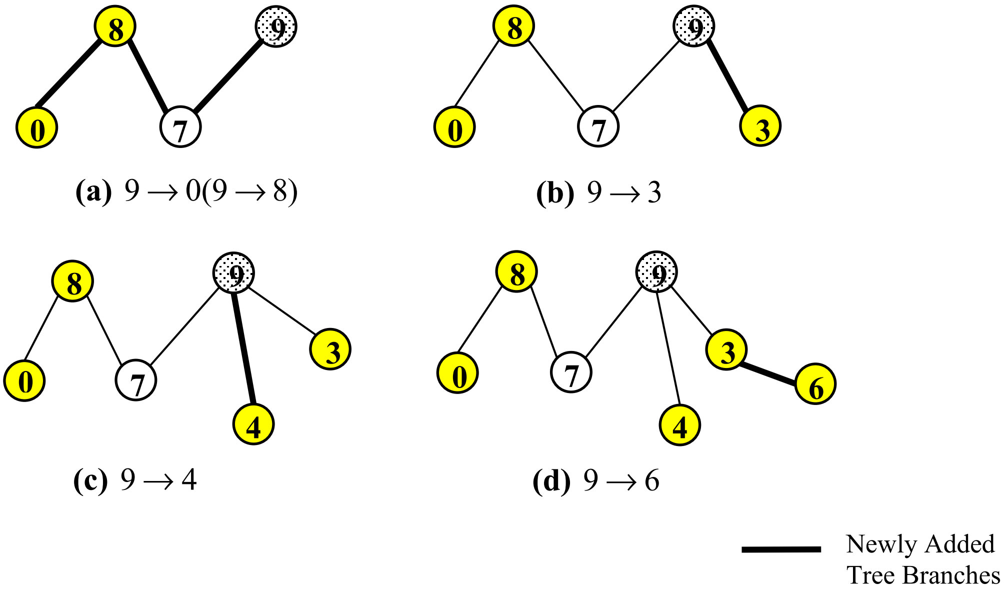


**Figure 4.1** The Example Network Used in Presenting the Heuristics.

### 4.3.1 Source Shortest Paths Multicasting (SSPM)

The first heuristic is a very simple algorithm that corresponds to the popular per-source tree multicast algorithm widely used in wired networks. Starting from a one-node tree consisting of only source as root, *shortest paths* from *source* to individual members in terms of power metric are added to the existing tree one at a time, as illustrated step by step in Figure 4.2. Therefore, SSPM is basically the *superposition of unicast min-power paths* from the source to each group member, and no wireless nodal propagation nature is considered.



**(a)** $9 \rightarrow 0(9 \rightarrow 8)$ **(b)** $9 \rightarrow 3$

**(c)** $9 \rightarrow 4$ **(d)** $9 \rightarrow 6$

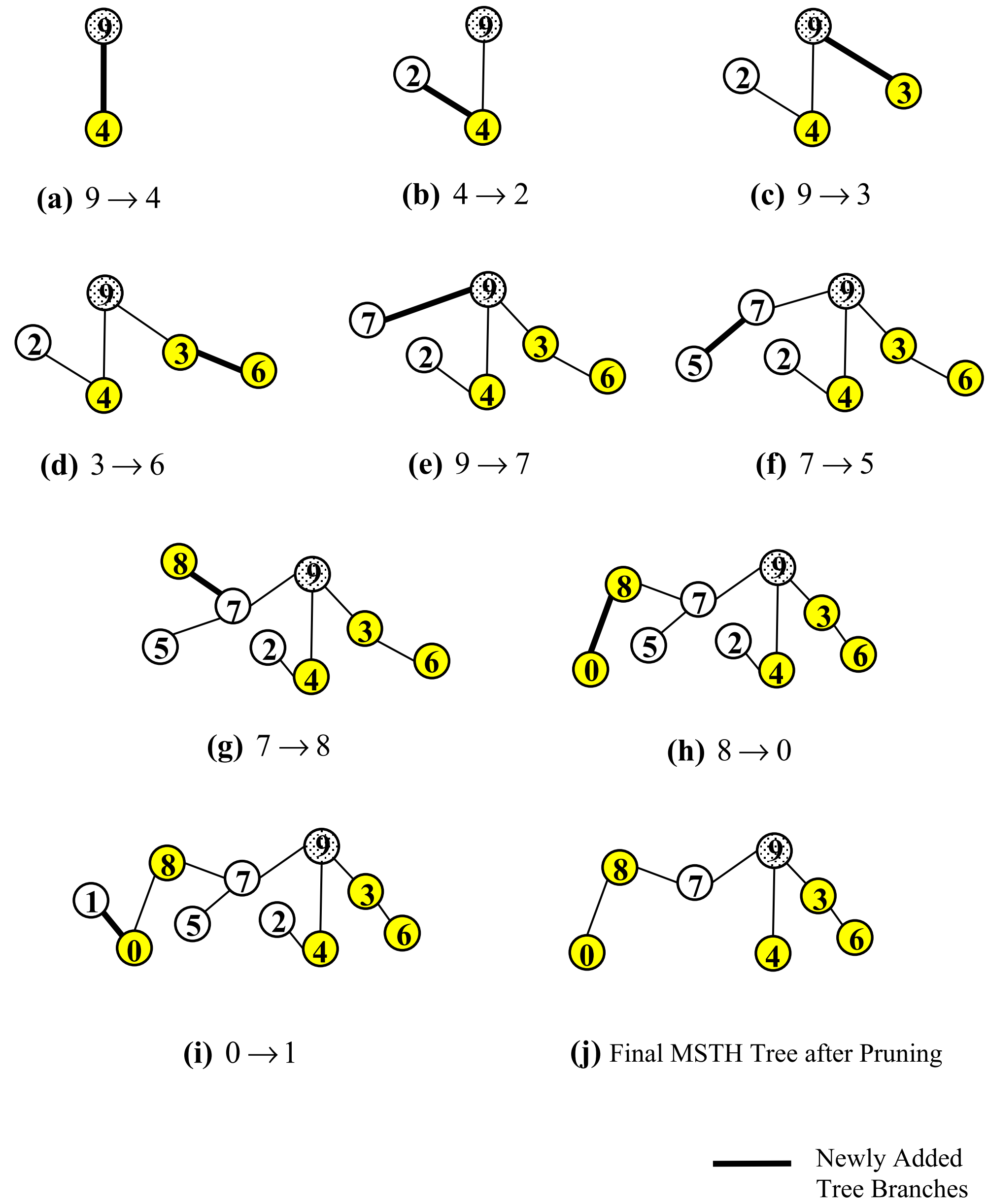
**Figure 4.2** Example of Constructing SSPM.

Since this algorithm needs to compute the shortest distance from the source to every group member, its run-time complexity is bounded by that of a shortest path algorithm such as Dijkstra's algorithm [64], $O(v^2)$, where $v$ is the total number of nodes in the network.

## 4.3.2 Minimum Spanning Tree Heuristic (MSTH)

### 4.3.2.1 The Original Algorithm

In MSTH, the approximate solution of min-power multicast tree $T_{MSTH}$ is obtained by pruning those non-member leaves and branches from the *MST* for *G*. However, our *MST* algorithm is specially tailored taking into account the wireless nodal propagation property. We describe the basic operation of MSTH using Figure 4.3.



**(a)** 9 → 4 **(b)** 4 → 2 **(c)** 9 → 3

**(d)** 3 → 6 **(e)** 9 → 7 **(f)** 7 → 5

**(g)** 7 → 8 **(h)** 8 → 0

**(i)** 0 → 1 **(j)** Final MSTH Tree after Pruning

**Figure 4.3** Example of Constructing MSTH.

- *Step 1*: Initially the MST only consists of the source node as root. We begin by determining which node is the first one can be added, which should be reached by the source at the minimum cost, i.e., the source's nearest neighbor. This node is Node 4, with a cost or power expenditure of 84. Thus, after this step, two nodes are included in the tree, Nodes 9 and 4 (Figure 4.3(a)). We use the notion of $9 \rightarrow 4$ to denote that the addition to the tree in this step is the transmission from Node 9 to Node 4.

- *Step 2*: Now we decide which node can be added to the tree next at minimum cost. There are two choices: either Node 9 increases its transmission power level to reach a second node, or Node 4 transmits to its nearest neighbor. If Node 9 chooses to increase its power, then Node 3 should be the next node added into the tree. Note that the cost associated with the addition of Node 3 is just the *incremental cost* in the sense that Node 9 increases its transmission power from a level already sufficient to reach Node 4 (84) to a level enough to reach Node 3 (145). Therefore, Node 9 only needs to increase its power level from 84 to 145 to reach both 4 and 3 *in a single transmission*. The *incremental cost* is $145 - 84 = 61$ ! Hence, we are able to exploit the wireless nodal propagation advantage to conserve power consumption. On the other hand, if Node 4 chooses to transmit, then Node 2 should be added and since Node 4 is a new *transmitting node*, the cost associated with the addition of Node 2 is just the power for 4 to reach 2, i.e., 37. Compare 61 and 37, the node added into the tree in this step is of course Node 2 (Figure 4.3(b)), which has a smaller cost. More generally, we can formulate the *node selection criterion* as this: every time when adding a new node into the partially established tree, for each node *i already in the tree* (it may be a *transmitting* node or just a *leaf* node then) and each node *j not yet in the tree*, the following *incremental cost* is evaluated:

$$c_{inc_i} = p_{inc_i} = p_{ij} - p_i , \tag{4.4}$$

where $p_{ij}$ is the *absolute cost* of a transmission from node *i* to node *j*, i.e., $c(e_{ij})$, and $p_i$ is the power level at which node *i* is *currently transmitting* (prior to the addition of node *j*). Obviously, $p_{ij} \geq p_i$, and if node *i* is a leaf node at that time in the tree, $p_i = 0$. The quantity $c_{inc_i}$ represents the incremental cost associated with adding node *j* to the set of nodes to which node *i* is currently transmitting. The pair *{i, j}* that results in the minimum $c_{inc_i}$ is selected. Let us reconsider *Step 2*, only two nodes, Nodes 9 and 4, are already in the tree, $p_9 = p_{9,4} = 84$, Node 4 is a leaf node at that time, therefore $p_4 = 0$. According to eq. (4.4), $c_{inc_9} = p_{9,3} - p_9 = 145 - 84 = 61$, $c_{inc_4} = p_{4,2} - p_4 = 37 - 0 = 37$, therefore, Node 2 is chosen. After addition, we need to update the current value of $p_4$ and $p_2$ to 37 and 0, respectively.

- *Step 3*: There are three nodes in the tree, namely, Nodes 9, 4 and 2, 9 and 4 are transmitting nodes and 2 is a leaf node. For each of them, we calculate the incremental cost to reach a new node. $c_{inc_9} = p_{9,3} - p_9 = 145 - 84 = 61$, $c_{inc_4} = p_{4,7} - p_4 = 301 - 37 = 264$, $c_{inc_2} = p_{2,7} - p_2 = 265 - 0 = 265$. Therefore, Node 3 is added (Figure 4.3(c)).
- *Repeat*: This procedure is repeated (Figure 4.3(d) – (i)) until all nodes are included and a minimum spanning tree is constructed (Figure 4.3(i)).

After the MST is constructed, we prune all non-member leaf nodes from the tree, if the pruning further makes other non-member nodes become leaves, repeat this pruning procedure until all leaf nodes are members (note that not all member nodes are necessarily leaf nodes, as some may act as tree branch nodes (relaying node) on the tree; moreover, some non-member nodes may also act as branch nodes) (Figure 4.3(j)).

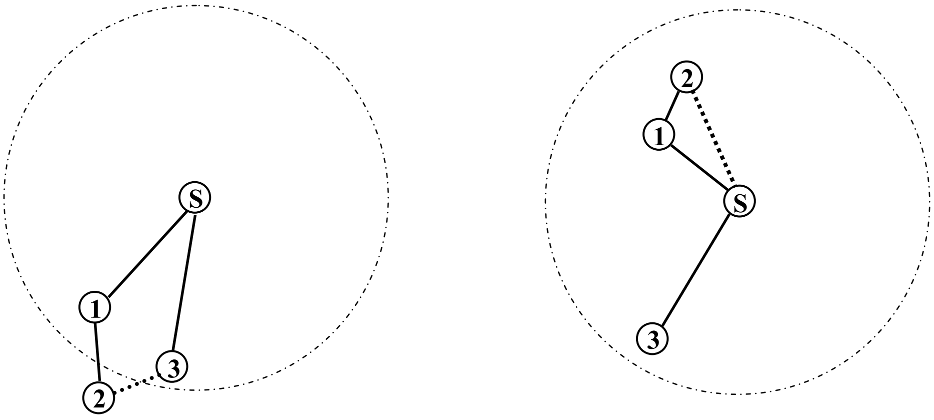
Our algorithm is similar in principle to Prim's MST algorithm [64], in the sense that new nodes are added one at a time on a minimum-cost (incremental cost) basis until all nodes are included in the tree. In fact, the implementation of the algorithm is based on the standard Prim's algorithm, with one fundamental difference. Whereas the inputs to Prim's algorithm are the link costs which remain *unchanged* throughout the execution of the algorithm, our MSTH must dynamically *update* the costs at each step, i.e., whenever a new node is added to the tree, to reflect the fact that the cost of adding a new node to a transmitting node's list of recipients is the incremental cost.

Unlike Prim's algorithm, which guarantees the formation of min-cost spanning trees for link-based costs as in wired networks, our MSTH does not necessarily provide genuine min-cost tree for wireless networks taking into account the wireless nodal propagation property. However, the performance results in sec. 4.4 demonstrate nonetheless that this algorithm does, in fact, provide satisfactory performance.

The computational complexity of MSTH, when implemented by means of modification of standard Prim's algorithm, is $O(v^3)$.

#### 4.3.2.2 Improved MSTH with ReRouting Operations (MSTH-RR)

Basically, according to our original MSTH algorithm, a node with minimum incremental cost is added into the tree one at a time until all nodes are in the tree. It is only a *localized cost*, not the *aggregate cost*, in the sense that this cost is only corresponding to an addition of a new branch into the tree. However, through careful simulations and investigation, we find some inaccuracy inherent to our original MSTH algorithm introduced by this localized calculation, which can be best illustrated by examples in Figure 4.4.



(a) (b)

**Figure 4.4** Inaccuracy of the Original MSTH.

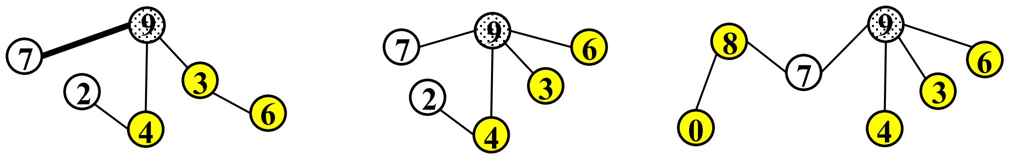
In Figure 4.4, the big dashed circle represents the maximum transmission range of source node *S*. Nodes 1, 2 and 3 are added into the tree one by one. Let us look at Figure 4.4(a) first. Node 2 is added into the tree earlier than 3 because the incremental cost for node 1 to add node 2 is smaller than that for node *S* to increase its power to add node 3. But after adding node 3, it is possible that the total cost is cheaper if node 2 switch to connect itself to node 3 (by the dashed link between 2 and 3) instead of from node 1. This problem is due of our localized calculation. In the original MSTH, when node 2 is added into the tree, only the incremental cost from 2 to *S* and node 1 are calculated, and the smaller one is chosen. After it is added into the tree, we label it as *permanent* tree node, and do not change the tree link between nodes 2 and 1, although later node 1 would increase its power range to reach a further node 3. When node 3 is added into the tree, we ignore the *rerouting possibility* for those *permanent tree nodes*. However, just as illustrated, after node 3 becomes a tree node, the total cost may be changed by rerouting node 2 to 3 so that we can get more accurate and efficient result.

Another problem is also due to the disregard of possible rerouting for those permanent tree nodes. In figure 4.4(b), node 3 is still the final one to be added into the tree, and it is still within the coverage of source. Now we can find the transmission from node 1 to 2 is essentially not necessary, as source *S* can obviously simultaneously reach 1 and 2 by a single transmission to 3! The more efficient MSTH tree is a one-hop tree connecting node 1, 2 and 3 to *S*.

Observing the potential inefficiency within our original MSTH, we propose the "rerouting" (RR) operations as follows to improve its performance.

- *RR Operation 1*: After adding a new node into the tree, calculate the incremental cost *from this node* to those previous tree nodes. If a cheaper route exists from this new node, reroute the previous tree node to this node by changing its parent as this new node.
- *RR Operation 2*: Every time when increasing a transmitting node's power to reach a further node, check if this increase can make another *neighbor* reachable (thus *its incremental cost is 0*) which is connected by another parent before with *incremental cost greater than 0*. If yes, change that node's parent to this transmitting node.

We apply the RR operations to the original MSTH example in Figure 4.3, and present the new tree in Figure 4.5. It is shown that the RR operation can improve the step 5 for original MSTH, i.e., after Node 7 is added, the transmission from Node 3 to 6 can be eliminated as Node 9 can reach Node 6 directly without any incremental cost after it increases its power to cover Node 7, as in Figure 4.3(e) and Figure 4.5(b). In the further discussion, we always use this improved version of MSTH-RR, unless otherwise stated explicitly.



**(a)** Figure 4.3(e) **(b)** After applying RR **(c)** Final MSTH-RR Tree

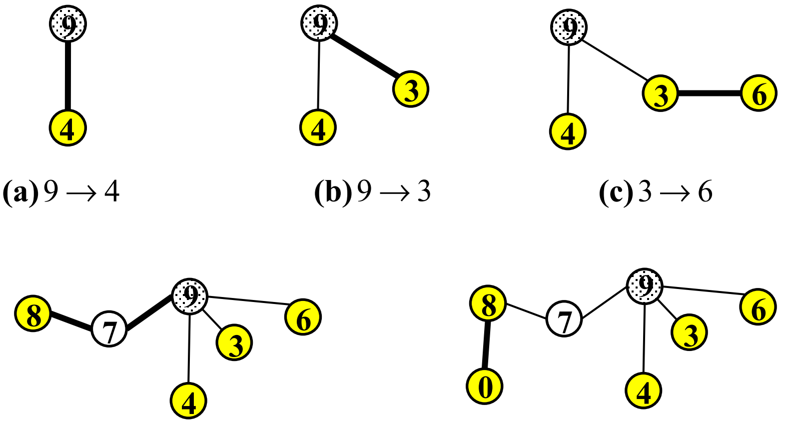
**Figure 4.5** Example of MSTH-RR.

### 4.3.3 Shortest Paths Heuristic (SPH)

SPH is also related to Prim's MST algorithm: when a partial tree containing a subset $N_k, N_k \subset N$ of multicast group members has been built up, an appropriately chosen member $s_{k+1} \notin N_k$ is connected to the tree by a shortest path in terms of incremental cost. More specifically, the SPH can be formulated as follows:

- *Step 1*: Begin with a subtree $T_{SPH}$ of $G$ consisting of a single node $s_1$, i.e., the source node as root, $k = 0$.
- *Step 2*: If $k = n$ ($n$ is the total number of multicast group members), then stop.
- *Step 3*: Determine a member node $s_{k+1} \notin T_{SPH}$ closest to $T_{SPH}$ (ties are broken arbitrarily). Add to $T_{SPH}$ a *shortest path* joining it with $s_{k+1}$, $k = k+1$, then go to *Step 2*.

SPH differs from MSTH in that each time a *multicast group member* is added into the tree one at a time in terms of *minimum incremental cost*, so that no spanning tree is required to be established first and then be pruned, and it is possible that *a path consisting of several hops* instead of a link is added into the tree at one time. For example, in Figure 4.6, let us look at the 4th step (step (d)) in which Node 8 is added into the tree. In this step, the incremental costs of adding Node 8 and Node 0 (they are the only two members not yet in the tree.) are evaluated, and Node 8 is chosen. The incremental cost of adding Node 8 into the tree is calculated as follows. Clearly, for the current tree nodes, Nodes 9, 4, 3 and 6, only 9 and 4 are possible candidates. The shortest path in terms of incremental cost from Node 9 to Node 8 is $9 \to 7 \to 8$, with $p_{inc_{9\to7\to8}} = p_{inc_{9\to7}} + p_{inc_{7\to8}} = p_{9,7} - p_9 + p_{7,8} - p_7 = 369 - 145 + 170 - 0 = 394$, and for Node 4, $p_{inc_{4\to7\to8}} = p_{inc_{4\to7}} + p_{inc_{7\to8}} = p_{4,7} - p_4 + p_{7,8} - p_7 = 301 - 0 + 170 - 0 = 471$. So finally Node 9 is chosen as transmitting node and the path $9 \to 7 \to 8$ is added into the tree. Moreover, the RR operations proposed for improving MSTH algorithm are also applicable to SPH, as illustrated in Figure 4.6(d).



**(a)** $9 \to 4$ **(b)** $9 \to 3$ **(c)** $3 \to 6$

**(d)** $9 \to 8$ (After RR Operations) **(e)** $8 \to 0$, **Final SPH Tree**

**Figure 4.6** Example of Constructing SPH.

The SPH can be implemented by a modification of Prim's algorithm, given shortest paths from every multicast member to all other nodes. Since shortest paths from each member can be determined by Dijkstra's algorithm in $O(v^2)$ time, the complexity of SPH algorithm is $O(nv^2)$, where $v$ is the total number of nodes, and $n$ is the number of group member nodes.

### 4.3.4 Distance Network Heuristic (DNH)

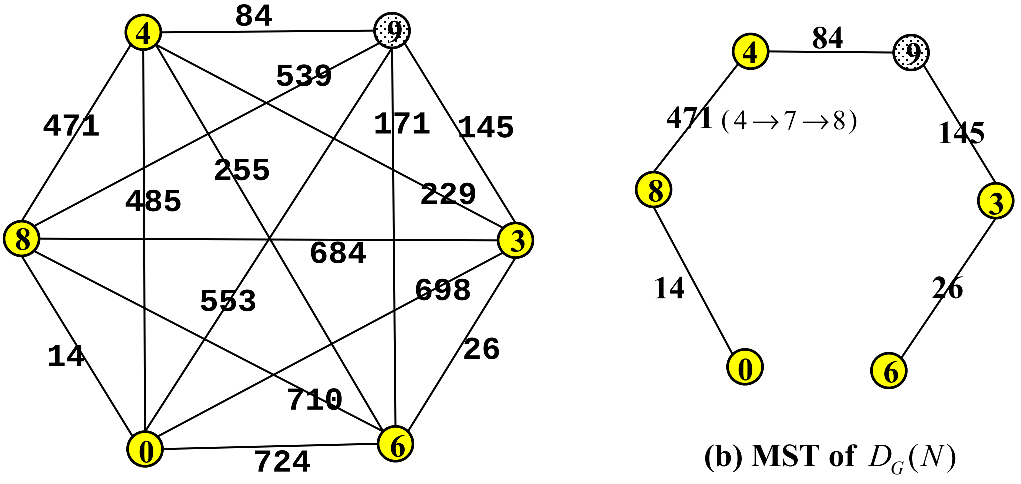
The DNH applies a minimum spanning tree algorithm to the *complete distance network* derived from the original network. It is based on the following lemma:

**Lemma**: Solving an instance of the Steiner tree problem for *N* in *G* is equivalent to solving an instance of the Steiner tree problem for *N* in the distance network $D = D_G(N)$. In other words, $|T_G(N)| = |T_D(N)|$.

The DNH algorithm therefore can be formulated as follows:

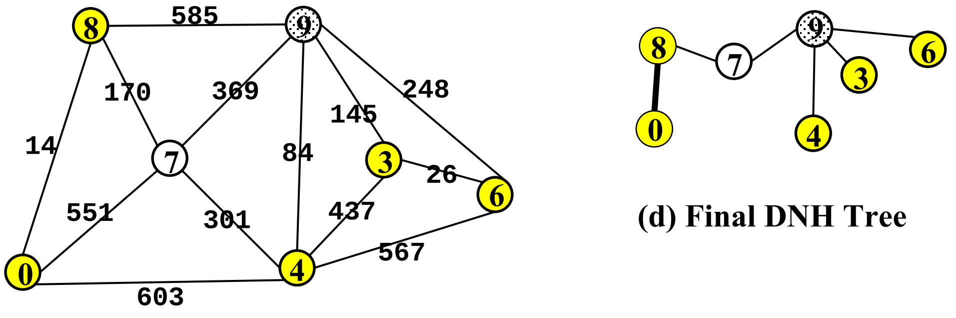
- *Step 1*: Construct the *complete distance network* $D_G(N)$ for *N* in *G*.
- *Step 2*: Determine a minimum spanning tree of $D_G(N)$.
- *Step 3*: Replace each edge in the MST by the corresponding *shortest path* in *G*. Let $T_D$ denote this network. Note that shortest paths can be selected in such a way that $T_D$ is a tree.
- *Step 4*: Determine a MST $T_{DNH}$ of the subnetwork of *G* induced by $T_D$.
- *Step 5*: Prune from $T_{DNH}$ those unnecessary edges so that all the leaves in $T_{DNH}$ are group members. Stop.

As defined in sec. 4.1.1, a *complete distance network* $D_G(N)$ for *N* in *G* is a network with *N*, the set of *multicast group members* as its vertex set, and with $d_G(v_i, v_j)$, the *shortest path* between $v_i$ and $v_j$, for each pair of vertices $v_i, v_j \in N$ as the *length of the edge* $(v_i, v_j)$. For a multicast group (including source and destinations) with *n* members, the complete distance network has $\frac{n(n-1)}{2}$ edges corresponding to the shortest paths in the original network among members. Figure 4.7(a) constructs such a complete distance network for the example network Figure 4.1, in which only group members are included and the edge costs correspond to the costs of shortest paths. After such a complete distance network $D_G(N)$ for $N$ in $G$ is constructed, a MST is determined for $D_G(N)$ (Figure 4.7(b)), and we replace each edge in the obtained MST by the corresponding shortest paths in G to get a tree $T_D$ (Figure 4.7(c)), in which only one edge between Nodes 4 and 8 is replaced by the shortest path $4 \to 7 \to 8$ and some unrelated nodes in the original network are removed, such as Nodes 1, 2, and 5. Finally we apply MSTH algorithm to $T_D$ to get the final solution of the multicast tree, as illustrated in Figure 4.7(d).



(a) **Complete Distance Network** $D_G(N)$

(b) **MST of** $D_G(N)$



(c) $T_D$ **From MST of** $D_G(N)$

(d) **Final DNH Tree**

**Figure 4.7** Example of Constructing DNH.

The complexity of DNH is $O(nv^2)$. It is *Step 1* that dominates all the remaining steps. It involves solving *n* shortest path problems.

### 4.3.5 Summary

In this section, we propose and illustrate a class of heuristics for constructing the minimum power multicasting tree taking into account the wireless nodal propagation property. They are summarized as in Table 4.1. SSPM is the simplest one which is, in fact, only the superposition of unicast shortest paths from individual members to source node, and the WNP property is neglected. MSTH is the most important design of our heuristics in the sense that the WNP property is properly reflected into the new definition of incremental-cost metric. In SPH, we extend this new metric to apply to a shortest path instead of just an edge, and for DNH, the MSTH algorithm is applied to a complete distance network of the multicast group in order to remove some unrelated nodes first. In the next section, we present simulation results for these heuristics.

**Table 4.1** Summary of Minimum Power Multicasting Algorithms.

| ***Algorithms*** / ***Criteria*** | ***SSPM*** | ***MSTH*** | ***SPH*** | ***DNH*** |
|---|---|---|---|---|
| Routing Metric | Unicast Power | Min Incremental-Cost Edge | Min Incremental-Cost Path | Min-Incremental-Cost Edge |
| Consideration of WNP Property | No | Yes | Yes | Yes |
| Complexity | $O(v^2)$ | $O(v^3)$ | $O(nv^2)$ | $O(nv^2)$ |
| Unique Characteristics | Superposition of Unicast Shortest Paths | Add One Node Each Time With Min Incremental-Cost | Add One Member Each time With Min Incremental-Cost | Apply MSTH to A Complete Distance Network |

## 4.4 Performance Evaluation of Min-Power Algorithms

In this section, we present a series of simulations to evaluate the performance of our minimum-power multicasting algorithms proposed in the last section. The performance metrics used for evaluation are first defined, and the simulation model and methodology are described. Then the performance results are analyzed and compared.

### 4.4.1 Performance Metrics

We have defined and used the following metrics for the performance evaluation of our algorithms. We use a multicast tree example in Figure 4.8 to illustrate, which corresponds to the MSTH tree obtained from Figure 4.3 for the example network in Figure 4.1. The values in parentheses adjacent to each node represent the number of neighboring nodes for that node in the original network, for instance, Node 9 has 6 neighbors, namely, Node 2, 3, 4, 6, 7 and 8 according to Figure 4.1.

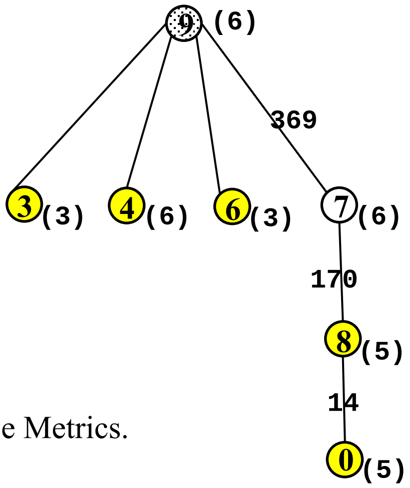


**Figure 4.8** Illustrations of Performance Metrics.

- **Total Power Consumption**: According to our cost definition, it is just the total costs to construct the multicast tree. Considering the WNP property, the total costs of the tree is simply the sum of the cost expended at each of the *transmitting nodes* in the tree; leaf nodes which do not transmit therefore do not contribute to this quantity. This metric represents the power efficiency of an algorithm. In the example tree of Figure 4.8, there are 3 transmitting nodes, namely, Nodes 9, 7 and 8. The total power consumption to establish this tree is $p_9 + p_7 + p_8 = 369 + 170 + 14 = 553$.

- **Multicast Tree Size**: The number of tree nodes (the sum of transmitting nodes and leaf nodes) in the multicast tree. This number represents the extent to which network nodes are involved in the multicasting tree. For example, in Figure 4.8, the multicast tree size is 7.
- **Number of Total Interfered Nodes**: For each transmission from a transmitting node, there may be some neighboring nodes that do not want to receive (hear) this transmission. This metric is the sum of these unwanted neighbors for every transmitting node in the tree. For instance, there are 3 transmitting nodes in Figure 4.8. For Node 9, by transmitting at power level 369, it can be actually heard by Nodes 2, 3, 4, 6 and 7 but not 8. Only Nodes 3, 4, 6, and 7 are in the tree, so Node 9's transmission is unwanted for Node 2, and the number of interfered nodes for Node 9 is thereby only 1. Likewise, Node 7's transmission at power 170 can only interfere with Node 5, and Node 8's transmission to Node 0 can affect no one else except its potential recipient Node 0. Therefore, the number of interfered nodes in that tree is $1 + 1 + 0 = 2$.
- **Average Interference Factor**: However, the above definition of the number of total interfered nodes is only an absolute value which does not consider the extent of interference incurred by individual transmitting nodes. For example, for two nodes both having 3 interfered nodes, one has only 1 desired recipient while the other has 7. Obviously, the interference extent of the former is much

stronger than the latter. Based on this investigation, we define the average interference factor for a transmitting node as the ratio of the interfered neighbor number to the number of total neighboring nodes reached by its transmission. This ratio is always smaller than or equal to 1. For the above two nodes with the same number of interfered neighbors, the average interference factor of the former node is $\frac{3}{3+1} = \frac{3}{4} = 0.75$, while that of the latter is only $\frac{3}{3+7} = \frac{3}{10} = 0.3$. Clearly, the interference extent or degree induced by the former is stronger than the latter. And the average interference factor for the whole tree is just the average for each transmitting node. We argue that this average interference ratio may be more accurate than the total number of interfered nodes to measure the interference degree incurred by a transmitting node and by the whole multicast tree. For the example tree in Figure 4.8, this factor is $\frac{\frac{1}{5}+\frac{1}{2}+\frac{0}{1}}{3} = \frac{0.7}{3} = 0.23$.

- **Average Hop Count**: Average number of hops traveled by data packets from the source node to individual group destinations. The metric reflects the routing efficiency of an algorithm.

### 4.4.2 Simulation Model and Methodology

The simulations are performed using the GloMoSim Network Simulator developed at UCLA [25]. Our simulation models a static wireless network of 50 nodes placed randomly within a $1500m \times 300m$ rectangular area. There are no network partitions throughout the simulation. In each scenario of one source and varied size of multicast group membership, source and member destination nodes are randomly picked.

Multiple runs with different seed numbers are conducted to simulate different network topology for each scenario and the collected data is averaged over those runs.

In order to fairly evaluate these heuristics, we also implement a baseline algorithm named Unicast Shortest Paths (USP) in which no power control is applied and all nodes transmit using a fixed $250m$ power range. USP is just the traditional per-source shortest paths algorithm and the *hop count* is used as the routing metric to calculate the shortest paths from the source to individual members by RPF algorithm. For all power-controlled heuristics proposed in the last section, all nodes can adjust their transmissions power range freely up to $500m$, the maximum range for transmission.

### 4.4.3 Simulation Results

#### 4.4.3.1 Total Power Consumption

Figure 4.9 gives the total power consumption performance of the five algorithms with respect to different sizes of multicast group membership.

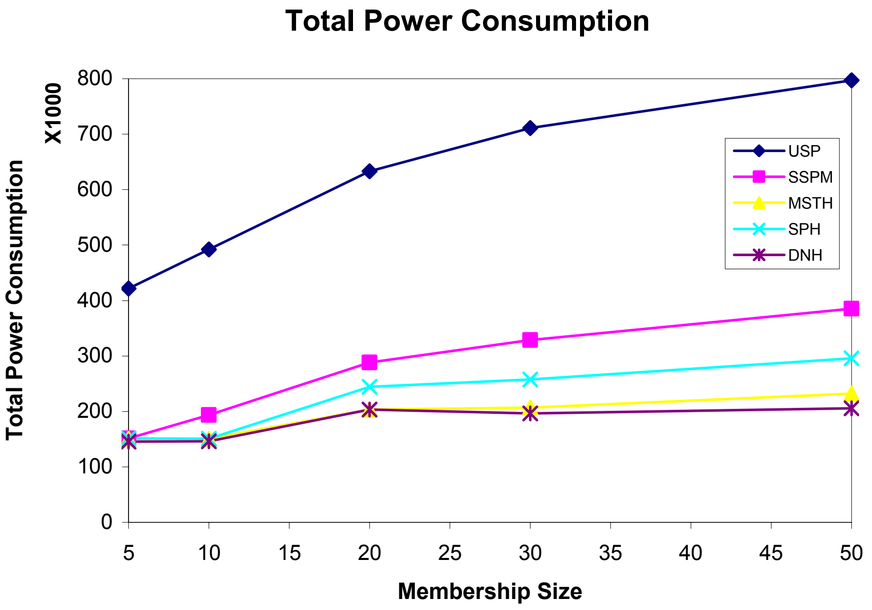


**Figure 4.9** Total Power Consumption of Min-Power Algorithms.

It is clear that all of our power-controlled algorithms are much more efficient in conserving power; the savings of power can be up to 80% (the difference between USP and DNH when group membership size is 50, which is actually the broadcasting situation). With the increase of membership size, power consumption of USP, which does not utilize power control, increases very fast, while this increase is relatively slow for all min-power algorithms, especially MSTH and DNH. This is due to the incorporation of WNP property into our algorithm designs. Therefore, although the membership size increases (obviously the tree size also increases), more transmitting nodes can take advantage of wireless broadcast propagation and reach more nodes in a single transmission so that the real power consumption does not increase too much.

Let us look at four min-power heuristics more carefully. Obviously, SSPM is the poorest one with respect to power efficiency. The reason is simple: though it uses power as routing metric, it does not take into consideration WNP property; it is only the superposition of unicast min-power paths from source to individual members. Hence, when the member number is sparse, it is as efficient as other algorithms because under this node density almost all unicast shortest paths are in the tree. However, when member nodes become dense, its power consumption rate is faster than the others since it fails to utilize wireless broadcast advantage.

MSTH and DNH have almost the same best power consumption performance except that when the member number is larger than 30, DNH outperforms MSTH marginally. This is also explainable, because in DNH, by the construction of a MST from the complete distance network of group members, some unrelated nodes are removed first so that the final multicasting tree of DNH is slightly more efficient than the MSTH tree. With the increase of membership size, we can also observe that the performance of SPH is poorer than MSTH and DNH. In SPH, a shortest path from a not-yet-on-tree member is added into the tree, instead of a minimum incremental-cost edge as in MSTH and SPH. Only the first hop in this shortest path has the possibility to take the advantage of WNP property; further hops are just the same as unicast shortest path. Therefore, the fact that SPH only partially takes into consideration of this advantage makes its performance better than SSPM, but worse than MSTH and DNH.

#### 4.4.3.2 Multicast Tree Size and Average Hop Count

The performance of multicast tree size and average hop count for each of algorithm with varied membership is illustrated in Figures 4.10 and 4.11, respectively.

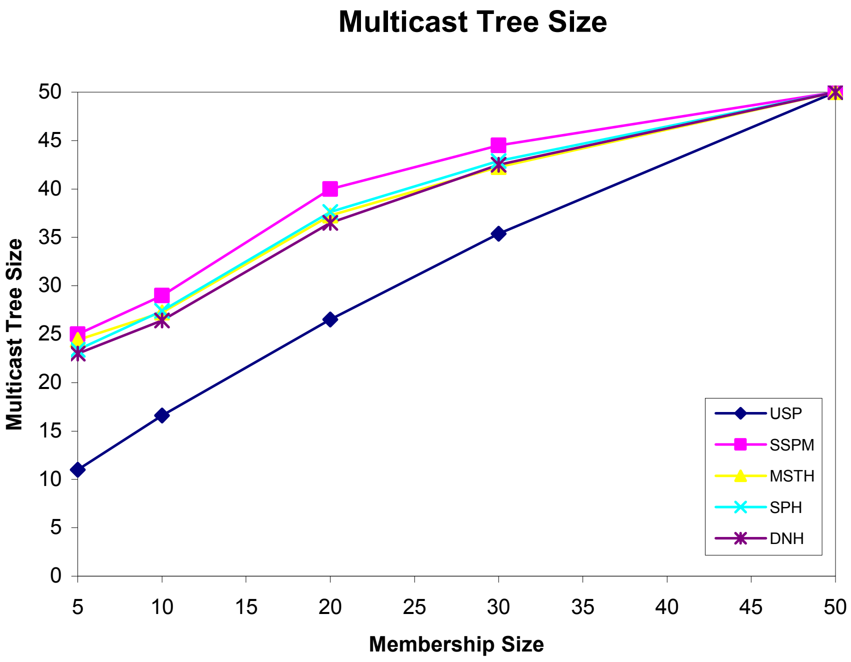


**Figure 4.10** Multicast Tree Size of Min-Power Algorithms.

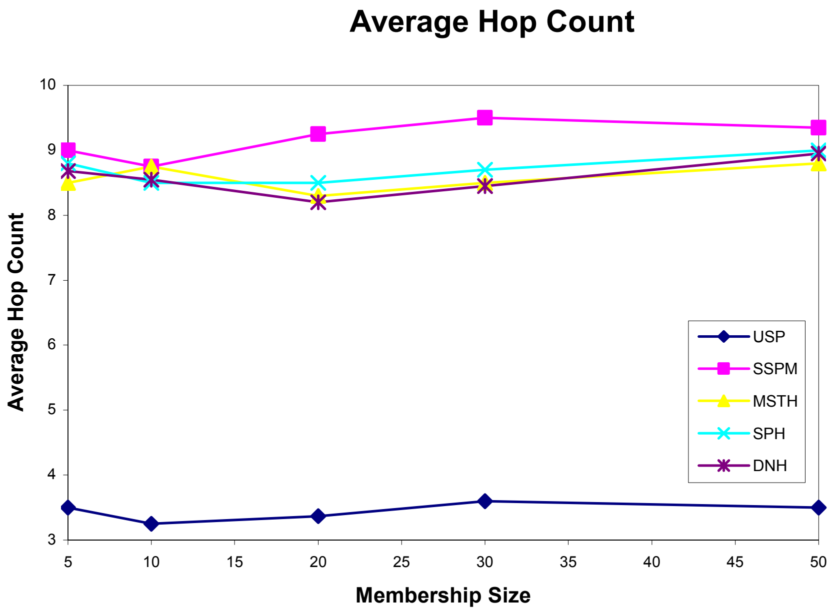


**Figure 4.11** Average Hop Count of Min-Power Algorithms.

It is shown that all algorithms using transmission power as routing metric have bigger tree size and average hop count compared with the baseline algorithm USP which uses hop count to calculate the shortest paths. Although in all these algorithms, nodes have the freedom to increase their power range up to 500m, the double of 250m fixed power used in USP, they still always attempt to route packets by as least power as possible to the next hop, in order to conserve energy. As a result, the multicast tree sizes and average hop count for these min-power algorithms are all larger than USP, with a large majority of paths consisting of more small-power hops. Among these algorithms, SSPM has the biggest tree size and average hop count, since it always routes packets using the least power. The rest of three heuristics, MSTH, SPH and DNH, yield similar tree size and average hop counts for various group sizes. Because they all utilize WNP property, their tree sizes and average hop counts are smaller than SSPM, but it is nevertheless evident that they are reluctant to use large power as this will result in greater power consumption. In summary, all these min-power algorithms achieve better power efficiency at the cost of increased tree size, and therefore higher average hop count. Another conclusion that can be drawn is that the average hop count is relatively unrelated with multicast group membership.

#### 4.4.3.3 Interference

Figures 4.12 and 4.13 present the number of total interfered nodes and the average interference factor of the multicast tree for all algorithms.

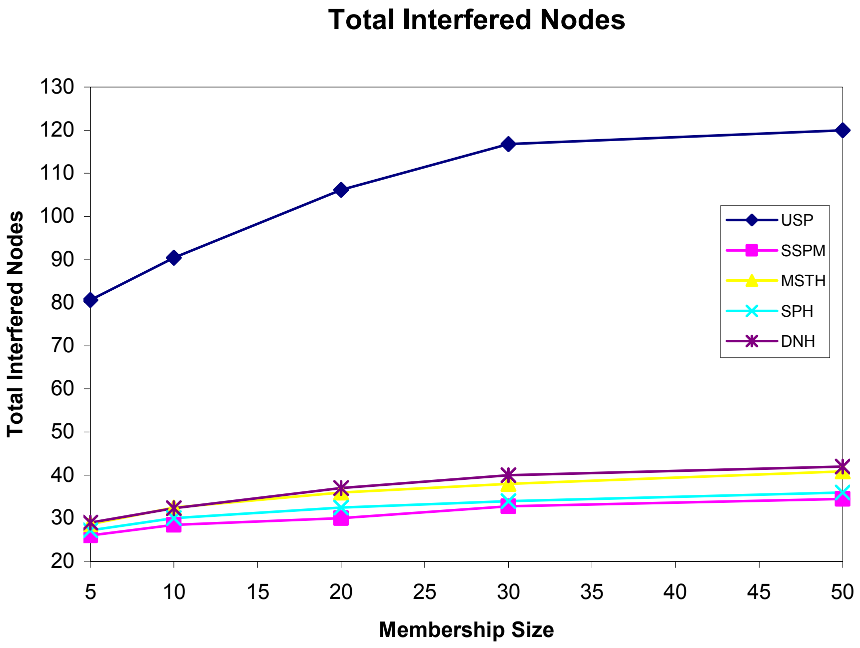


**Figure 4.12** Total Interfered Nodes of Min-Power Algorithms.

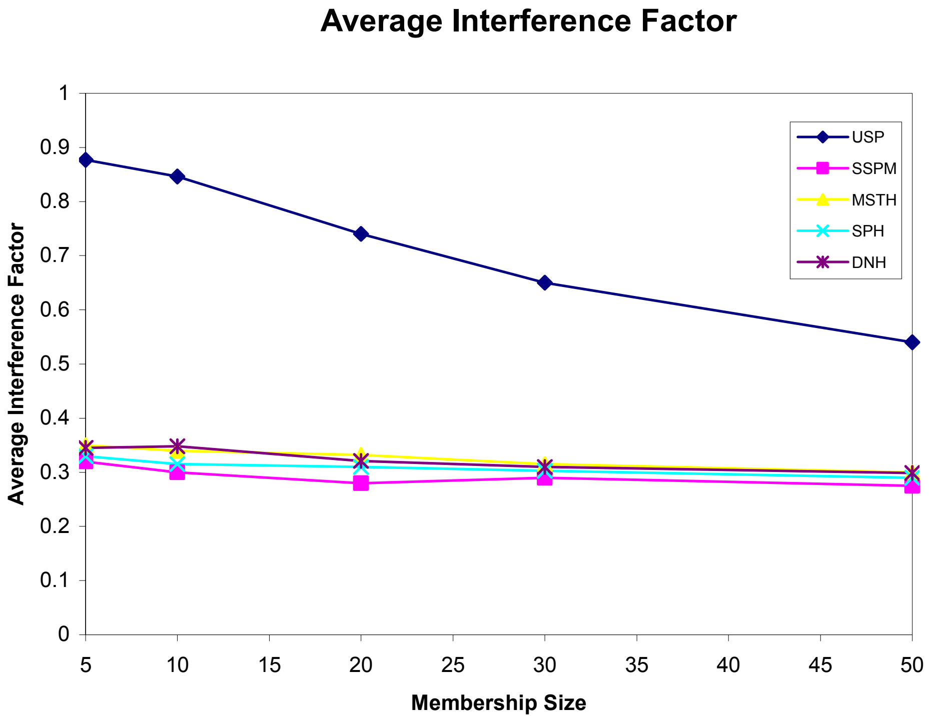


**Figure 4.13** Average Interference Factor of Min-Power Algorithms.

Again, all min-power heuristics demonstrate better performance in terms of interference control compared with USP, the only algorithm without power control scheme. With power conservation as their design objective, min-power algorithms are all biased to interference minimization. Actually we can also find that the interference incurred by these algorithms has a similar trend which is relatively stable for various sizes of membership. No matter how sparse or dense the group members are in the network, these algorithms always try to use minimum power for constructing the multicast tree and therefore produce similar interference results.

Another interesting finding is the apparent "contradiction" of the both performance metrics. Figure 4.12 shows a general increasing tendency of the absolute value of interfered nodes with the increase of group member nodes for all algorithms, while in Figure 4.13, the average interference factors are all dropping when increasing the membership. This is also explainable. Along with the increase of group members, the number of transmitting nodes in the tree increases, too. Therefore, the possibility of interfering with other nodes also increases, which results in more interfered nodes in the network. On the other hand, though the absolute number of interfered nodes increases, the number of reachable members in one hop transmission also increases and the average interference degree of the whole network may decrease. According to the definition of both interference metrics, our previous expectation that the average interference factor is more accurate than the total number of interfered nodes to measure the extent to which the network is interfered is therefore verified.

### 4.4.4 Summary

Through careful experiments, it is proved that our minimum-power multicasting algorithms are more efficient in terms of power conservation and interference control, compared with an algorithm without power control. This is an important improvement for power-constrained wireless ad hoc networks. More specifically, MSTH and DNH heuristics demonstrate the best performance of power-efficiency, thanks to their full incorporation of WNP into their algorithm design. SPH and SSPM algorithms are a little poorer than MSTH and DNH, while they still obtain much better power efficiency than the baseline algorithm without power control.

However, we also observe that these min-power heuristics achieve better power efficiency at the cost of increased multicast tree size and average hop count, which is particularly less desirable for applications requiring small average end-to-end delays, such as some multimedia applications. Moreover, longer paths may incur more storage and processing energy, and complicate the multicasting maintenance procedure. As we discussed before, there exists a tradeoff between short and simple paths and power conservation. In the next section, we design a new cost function trying to still obtain similar power savings with shorter paths.

## 4.5 Power-Efficient Multicasting Algorithms

In the last two sections, we propose a class of minimum-power multicasting algorithms for wireless ad hoc networks. Performance results show much better performance in terms of power efficiency. However, we also notice that this improvement is achieved at the cost of increasing multicast tree size and average path hop count. In this section, we will explore the possibility to improve path efficiency while still retaining power conservation for those min-power heuristics.

### 4.5.1 New Cost Function Definition

In the min-power multicasting algorithms proposed in the last two sections, we only use transmission power as the routing metric to define a pure power cost function. Although we incorporate WNP property into our algorithm design, simulation results still yield a trend to use low-power hops to relay packets in all min-power heuristics and therefore lead to increased multicast tree size and average hop count.

Now let us look at the MSTH in more detail. As we discussed before, to maintain a fixed receiving power threshold for all receivers, the transmission power required increases nonlinearly with distance, according to our assumption, the square of distance. As a result, even though we take into account WNP property, only those *near-zone* nodes with relatively small incremental cost are added into a transmitting node's children list in a single hop. *Far-zone* nodes can only be added into the tree by multiple low-power hops because the incremental power for that transmitting node to reach these far neighbors in one hop is too expensive. Figure 4.14 illustrates this situation. Node 1 is the nearest neighbor for source node *S*. Node 2 is also within the near-zone of *S*, and the incremental cost to add it is small, so it is added next. However, for the far-zone node 3, the incremental cost for *S* to reach it through the dashed link is higher than that for Node 2 to add its nearest neighbor Node 4. Therefore, it may not be possible to add that dashed link into the MST tree. The source node can only reach Node 3 through multiple small power hops, say, along path $S \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 3$ with 5 hops instead of a single large power hop, therefore reducing the path efficiency. Worse, the actual power required to establish this multiple-hop route may even be higher than power for a single-hop one.

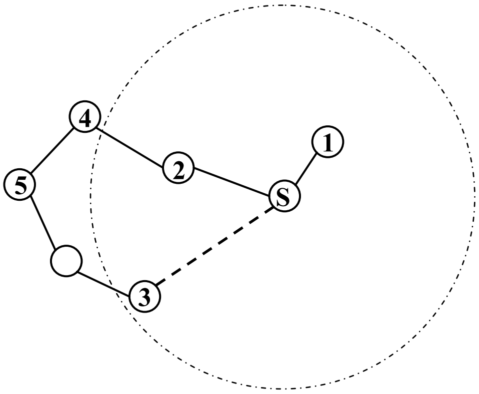


**Figure 4.14** Path Inefficiency in MSTH.

This path inefficiency is more significant when the number of member nodes is sparse. During the construction of the MST, we *do not consider the membership* and just establish a min-power spanning tree for the whole node set. After that, we prune the MST to obtain the multicast tree with all leaf nodes being only members. When the membership distribution is sparse, it is more likely for the source to establish long paths to reach those member nodes in the final multicast tree. This is verified by the multicast tree size results reported in Figure 4.10, where the gaps between USP (without power control) curve and others are more significant when membership size is small, for example, for 5-member and 10-member case.

Based on the above observation, we argue that the pure power metric cannot reflect the path efficiency. Some factors need to be added into the cost function to give preference to those high-power transmissions between connections of multicast group member nodes. Hence we design a new incremental cost function:

$$c_{inc} = \frac{p_{inc}}{p_{\max}} + \frac{N_{non-member}}{N_{covered}} \tag{4.5}$$

where $p_{inc}$ is the incremental power needed to support this new link as in eq. 4.4, $p_{\max}$ is the maximum transmission power for all nodes with power control. We normalize $p_{inc}$ by $p_{\max}$ in order to achieve the same order of magnitude as that of the second part. $N_{non-member}$ is the number of non-member nodes covered by this transmission, and $N_{covered}$ is the total number of nodes covered by this transmission. To explain the meaning of $N_{non-member}$ and $N_{covered}$, let us look at Figure 4.14 again. We assume only Nodes 1, 2 and 3 are group members. Now let us consider the step whether to add Node 3 or Node 4 into the tree. According to previous pure power cost function, it is Node 4 who should be added. However, based on our new cost function, although the incremental power part of Node 4 in eq. 4.5 is smaller than that of Node 3, the new transmission from Node 2 to Node 4 does not cover any member, its $N_{covered}$ and $N_{non-member}$ are both 1, i.e., only Node 4 itself, and the second part of $c_{inc}$ for it is $\frac{1}{1} = 1$. If source node S increases its transmission power level to reach Node 3 directly, all three covered nodes 1, 2 and 3 are members, so its $\frac{N_{non-member}}{N_{covered}}$ part is $\frac{3-3}{3} = 0$! The final value of $c_{inc}$ for Node 3 is smaller than Node 4 and Node 3 is added into the tree by a single transmission from the source. Essentially, we can regard the second part of eq. 4.5 as a path-efficiency factor in the sense it gives preference to increasing power level to reach more nodes in a single hop. However, not all large-power single-hop transmissions are preferred. Only transmissions covering more members are probably favored as long as their interfered nodes are not too many, i.e., $N_{non-member}$ is small. Therefore, we argue that our new cost function is more path-efficient, because a lot of small-power, multiple-hop paths, only connecting non-member nodes, are replaced by more efficient large-power hops. Multicast tree size and average hop count are expected to decrease compared with min-power algorithms. Moreover, the new cost function may still obtain similar or even better power-efficiency due to removals of many unnecessary small-power hops in the MST tree.

### 4.5.2 Simulation Results

The simulation results of our new cost function are presented in Figures 4.15 — 4.18. We apply this cost function to MSTH and DNH algorithms, which show best performance among all heuristics. We compare their performance with MSTH and DNH algorithms using only power as cost function, as reported in section 4.4. New MSTH and DNH algorithms are named as MSTH-II and DNH-II for differentiation. We also plot the performance curve of the baseline algorithm USP for comparison.

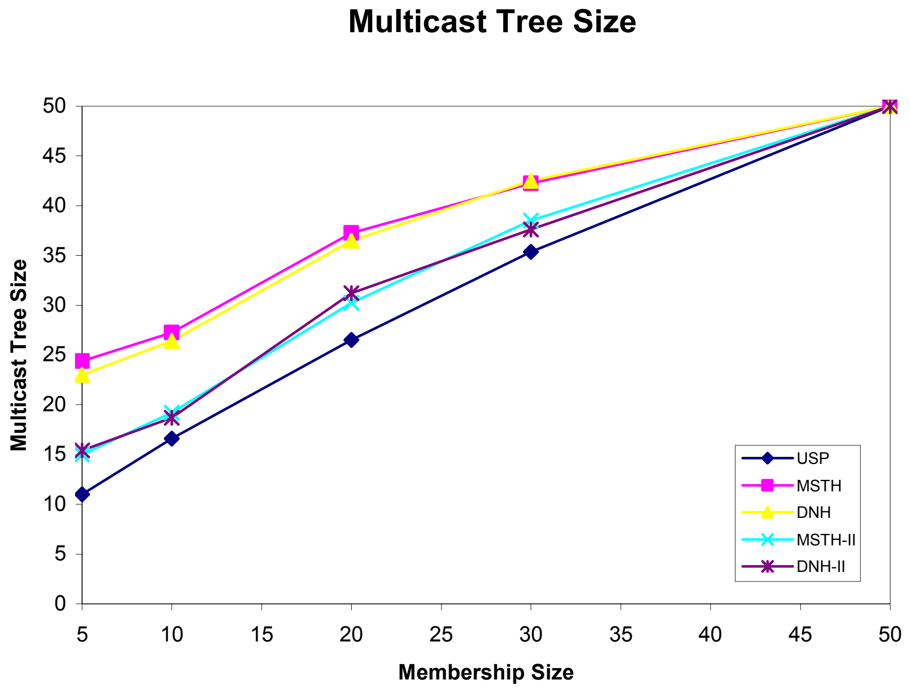


**Figure 4.15** Multicast Tree Size of Power-Efficient Algorithms.

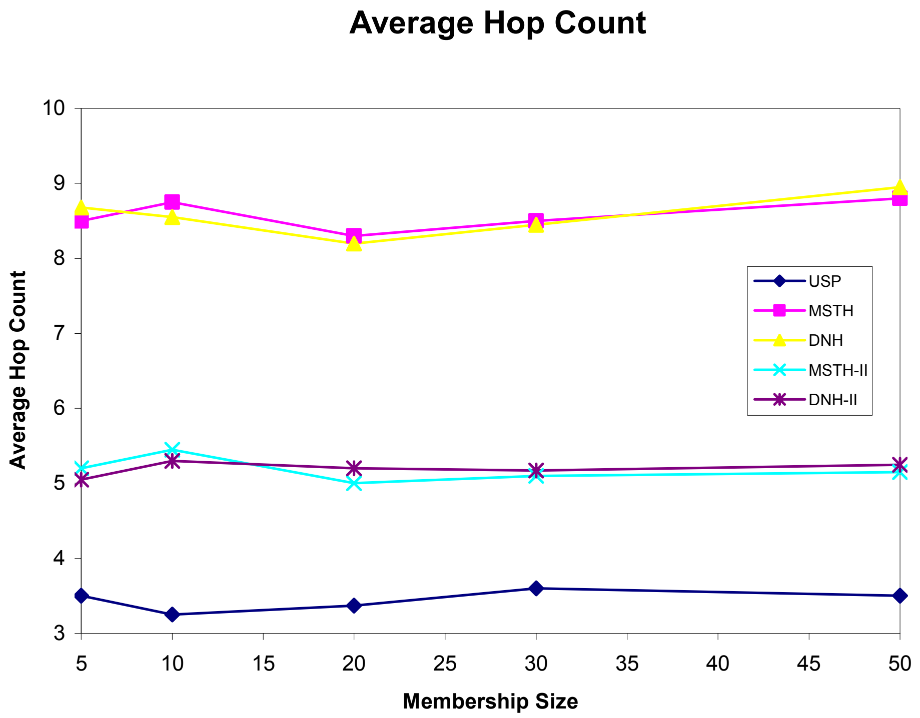


**Figure 4.16** Average Hop Count of Power-Efficient Algorithms.

As expected, MSTH-II and DNH-II achieve better path efficiency compared to original MST and DNH algorithms. Multicast tree size and average hop count are both reduced largely. And this improvement is even more significant for sparse membership distribution, such as 5-member and 10-member cases, for 5-member case, the multicast tree size shrinks about 40%! This is attributed to the incorporation of path efficiency factor into our new cost function. By giving preference to large-power hops connecting member nodes, the new power-efficient algorithms remove many unnecessary small-power hops which essentially results in path inefficiency over the original algorithms using only power as cost function. Then how about power performance?

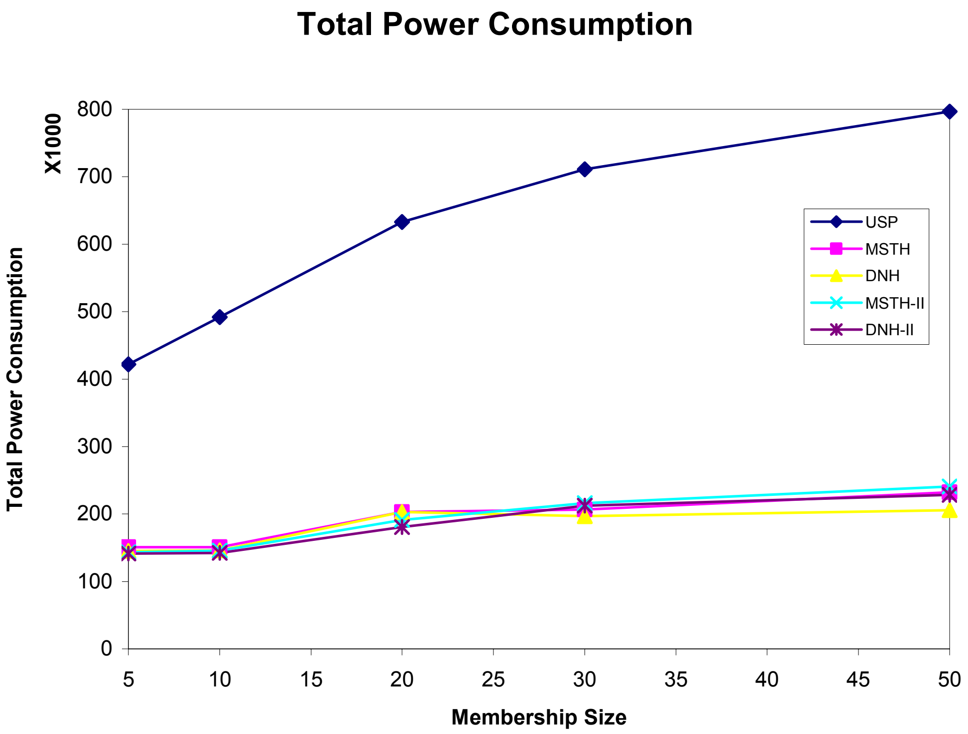


**Figure 4.17** Total Power Consumption of Power-Efficient Algorithms.

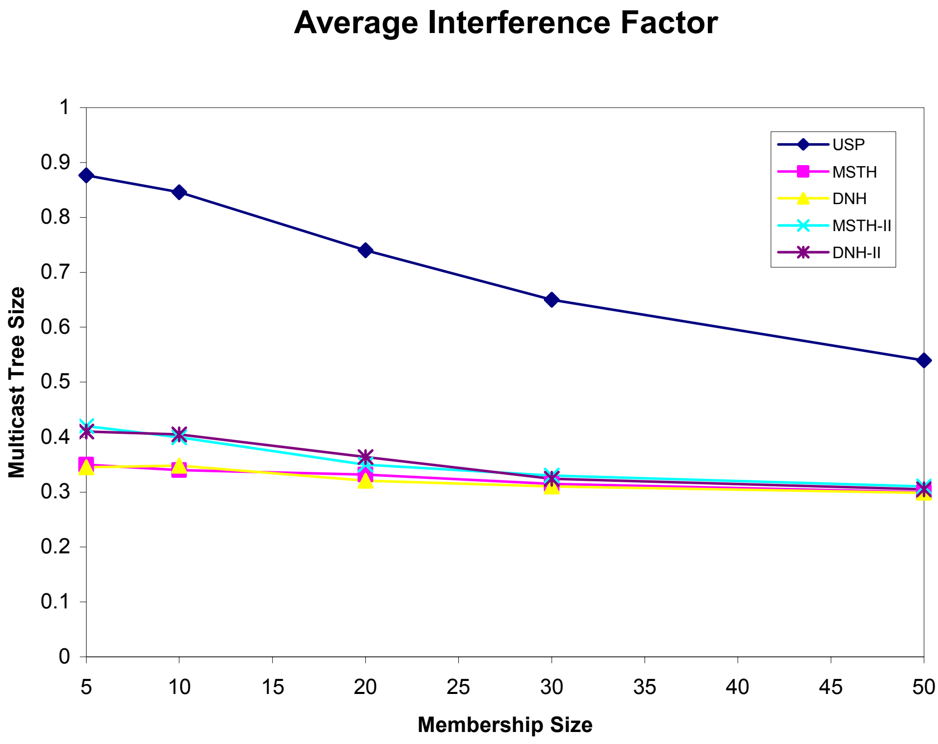


**Figure 4.18** Average Interference Factor of Power-Efficient Algorithms.

It is exciting to find that MSTH-II and DNH-II perform better than the original algorithms in terms of power minimization when member number is small, such as 5-member and 10-member cases. It seems that our new cost function conserves a little more power. This is also reasonable. According to the MST algorithm, a min-cost spanning tree is first constructed and then pruned to obtain the final multicast tree. When membership is sparse, although the MST tree is power-minimized, the pruned multicast tree may not be the same. Some unnecessary small-power hops and intermediate relay nodes are used, which may actually conserve more power. On the other hand, selective large-power hops are preferred according to our new cost function, and the final pruned tree consists of fewer nodes. Therefore, the total power consumed by fewer transmissions is less than the original version. With the increase of membership size, the new cost function is marginally less power-efficient compared with min-power cost function, because the MST construction is more accurate in forming the final multicast tree using only power as the cost function. In general, we can still conclude that our new cost function is at least as power-efficient as previous pure power cost function.

However, as for the interference performance, the new cost function incurs a little more interference for a sparse membership situation. This is also expected. By using larger power level, it inevitably interferes with more neighboring nodes in a single hop. But this interference is still much less than USP, because the interference factor is essentially taken into account in our new cost function, not only in the power calculation part, i.e. $\frac{p_{inc}}{p_{\max}}$ , but also in the path efficiency part, $\frac{N_{non-member}}{N_{\mathrm{cov}ered}}$. This calculation reflects interference induced by the transmission, except that it regards transmission to a single non-member node as full "interference" as well, by which large-power single-hop transmissions covering member nodes are therefore favored. When membership size gets bigger, the new cost function yields similar results as the previous version.

In summary, in this section, we investigate the potential path inefficiency behind min-power algorithms, which use only transmission power as cost function. Then we design a new cost function integrating power and path efficiency together. Simulation results demonstrate great path efficiency improvement compared to algorithms using power as cost function. The multicast tree size and average hop count are decreased significantly. Moreover, our new cost function can still achieve similar or even better performance in terms of power conservation and interference minimization.

## 4.6 Mobility Extensions

All algorithms proposed so far are based on the same assumption that there is no mobility for the whole network, i.e. a static ad hoc network. In this section, we extend our discussion to a full mobile environment. Taking mobility into account, a separate distributed tree maintenance algorithm is presented, which consists of two phases: power tuning phase and tree break repair phase. This tree maintenance algorithm can be combined with any of the power efficient tree-forming algorithms proposed before to reduce the number of link breaks effectively and repair the tree link breaks timely, while still conserving power consumption efficiently.

### 4.6.1 Network Modeling and Assumptions

We start our extension from the point that the multicast tree has been established using one of the power-efficient algorithms proposed previously. Then all nodes begin to move constantly with a predefined speed. Moving directions of each node are selected randomly, and when the node reaches the terrain boundary, it bounces back and continues to move. Since the network nodes are mobile now, links on the multicast tree are likely to break. Hence, there must be a way of maintaining the tree after topological changes in the network. We assume that the whole network is always connected, thus there is no network partition or tree merge after partition, and the only tree maintenance activity is to repair the broken tree links.

The source node is assumed to know the initial locations of each node in the network, so that all centralized algorithms proposed in the last several sections can work to form the tree. We also assume that each node knows the relative distances, not exceeding the maximum distance $d_{max}$, to each of its neighboring nodes, by periodic exchanges of beacon messages using the maximum power level. Each node maintains a Neighbor Table (NT) including the node ID of all neighbors, as well as the distances and required power levels to reach each of them.

All group members are assumed to join the multicast group at the beginning of simulation, and retain as members throughout the simulation. We do not consider the scenario when some nodes dynamically join or leave the group during the simulation.

Discrete power control is used to prevent continuous topology changes and frequent tree maintenances. More specifically, each node can only choose to transmit at several finite power levels $p_0, p_1, p_2, \ldots, p_n, 0 < p_0 < p_1 < p_2 < \ldots < p_n \le p_{\max}$, where $p_i = d_i^2$ (according to eq. (4.2)), $i = 0, 1, 2, \ldots, n$. When the distance to reach the receiving node is $d$, $d_i < d \le d_{i+1}$, $i = 0, 1, 2, \ldots, n-1$, it transmits at power level $p_{i+1}$. Compared with continuous power control scheme used in previous algorithms, discrete power control is easier to implement in practice, at the cost of more power consumption due to less accurate power calculation.

### 4.6.2 Multicast Tree Maintenance

Because the network nodes are mobile, links between nodes are likely to break. A multicast tree is maintained for the lifetime of the multicast group. Therefore, a tree maintenance algorithm is necessary to repair the broken tree links. The maintenance algorithm consists of two phases. In the power tuning phase, the transmitting node adaptively adjusts its power level to keep the tree link connected. In the break repair phase, the downstream node of the broken link runs a local repair algorithm to repair the break.

#### 4.6.2.1 Routing Tables

Each tree node must maintain a Multicast Routing Table (MRT). The MRT is used to maintain next hop information for the multicast tree. The detailed fields of the MRT are as follows:

- Multicast group ID,
- Source node ID,
- Current hop count to the source,
- Parent node on the tree, with the following sub-fields:
  - Parent node ID,
  - The distance to this parent,
  - The power level to this parent.
- Next hops (children) on the tree, with the following sub-fields per hop:
  - Next hop node ID,
  - The distance to this node,
  - The power level to this node,
  - Added flag.

Only a tree node, who is either a group member or a tree router, maintains such a MRT. Each MRT has a list of one or more next hops, or children on the multicast tree. The Added flag associated with each next hop is an indication of whether the link has been officially added to the multicast tree (see Section 4.6.2.3). According to WNP, the current transmission power of the node is the level required to reach the furthest child in its MRT.

Each node in the network, whether it is a tree node or not, must also maintain a Neighbor Table (NT) to record its neighbor information, with the following fields per neighbor:

- Neighbor node ID,
- The current distance to this neighbor,
- The current power level to this neighbor.

Each node is required to periodically broadcast a beacon message using the maximum power level. This message is prevented from being rebroadcast outside the neighborhood of the node by setting its time to live (TTL) value to 1. Neighbors receiving this message update their NT accordingly. The failure to receive such beacons in a predefined period of time (beacon_timeout) is an indication that the local link has broken, and the tree repair algorithm should be run to fix this break.

#### 4.6.2.2 Power Tuning

Power tuning is a phase that a transmitting node adaptively adjusts its power level to keep all next hop nodes on the tree connected, without running the tree maintenance algorithm. When a node senses that one or more children move out of its current power coverage $p_{current}$, while still within its maximum coverage $p_{max}$, it intelligently increases its power level to $p_{new}$, the level to cover its new furthest child, where $p_{current} < p_{new} \le p_{\max}$. The incremental power consumption $p_{inc} = p_{new} - p_{current} > 0$.

On the other hand, if this node senses that all children move closer to itself so that a lower power level $p_{new}$ now suffices for covering all of them, it tunes down from $p_{current}$ to $p_{new}$ to conserve power consumption, where $p_{new} < p_{current} \le p_{\max}$. The incremental power consumption $p_{inc} = p_{new} - p_{current} < 0$.

#### 4.6.2.3 Repairing Tree Link Breaks

Power tuning is a simple but effective method to reduce the link breaks. However, if a child moves beyond the maximum coverage of a transmitting node so that a tree link breaks, it is still necessary to have an efficient tree repair algorithm to fix the break.

It is assumed that each node learns of the distances to its neighbors by exchanging beacon messages periodically. A link break is detected if no such beacons are received from the particular neighbor in the time beacon_timeout.

When a link break on the multicast tree occurs, both the upstream node (node that is closer to the source) and the downstream node (node that is further from the source) of the link are able to detect the break. Only the downstream node is responsible for repairing the link. A node knows it is downstream or upstream of the link because it can check whether the node on the other side of the link is its parent or child in its MRT. Only the downstream node should initiate the link repair, since if nodes on both sides of the break try to repair the link, they might repair the link through different intermediate nodes, thus probably forming a loop. When an upstream node detects the link break, it just removes its next hop entry in its MRT associated with the downstream node of the broken link.

The downstream node initiates the repair by broadcasting a Tree Repair Request (TRREQ), which includes its own node ID, the broadcast ID of this request, the group ID, the source node ID, and its current hop count to the source. The broadcast ID is incremented for each TRREQ the node initiates. The broadcast ID, together with the initiating node ID, uniquely identifies each TRREQ. With the hop count included, only nodes that are no further to the source can respond to this TRREQ. This prevents the children of the requesting node, which are on the same side of the break with bigger hop counts to the source, from responding to the TRREQ and forming a loop eventually. As the two nodes are likely to still be close by, the downstream node can set the initial TTL of the TRREQ to be small, therefore allowing a quick local repair and preventing this TRREQ from being flooded across the entire network. The transmission power level of the TRREQ is also carefully chosen based on the following algorithm. Before the link break, let us denote the transmission power of downstream node and upstream node as $p_{down}$ and $p_{up}$, respectively. After the break, the downstream node initiates the TRREQ using a power level that is one level higher than the larger of $p_{down}$ and $p_{up}$ to reduce the interference incurred. If the initiating node does not receive a Tree Repair Reply (TRREP) before a predefined tree repair timeout, it broadcasts another TRREQ with its broadcast ID increased by one. As the network is assumed to be always connected, the initiating node is guaranteed to repair the break successfully after broadcasting finite TTREQs.

When a node receives a TRREQ, it first notes the node ID from which it receives the TRREQ, and creates a next hop entry in its MRT for that node. However, the Added flag for that next hop is set to FALSE, and only later is set to TRUE if the route is selected to be added to the multicast tree. The node then determines whether it is qualified to reply this TRREQ as described in the following. If it cannot respond, it just rebroadcasts this TRREQ. A node may receive the same TRREQ multiple times. If the node ID from which it receives the TRREQ is already added into its MRT, it just discards this TRREQ. Otherwise, this TRREQ must come from a new next hop, so it creates another next hop entry for this new node, also with Added flag unset.

Only a node currently on the multicast tree can respond to a TRREQ. When such a node receives a TRREQ, it first checks whether it is at least as close to the source as the requesting node, i.e. its hop count to the source should be at least the same as that of the requesting node. If so, it replies to the TRREQ by sending a Tree Repair Reply (TRREP) back to the initiating node along the reverse path or paths if it has more than one potential next hop entry (the next hop entry with Added flag unset) in its MRT. The power level is the bigger of its current power level and the power required to reach its furthest potential next hop. The TRREP contains its own node ID, its current hop count to the source, its current transmission power level, and the requesting node ID. The hop count field is incremented each time the TRREP is forwarded, so that when the initiating node receives the TRREP it may learn its hop count to the source node along this forwarding path. The power level field is also updated as follows. When a node receives a TRREP, it notes the power level $p_{current}$ in this TRREP, which indicates the current power level of its previous hop node. Then it consults its NT for its power level to reach that previous hop $p_{new}$, and updates the power field in TRREP as $p_{inc} = p_{new} - p_{current}$. If the previous hop is not a transmitting node on the tree currently, then update $p_{current} = 0$. If $p_{inc} \leq 0$, then update $p_{inc} = 0$. Therefore, the updated power field $p_{inc}$ indicates the incremental power required to add this new next hop. When eventually the initiating node receives this TRREP, it may know the incremental power required to add this forwarding branch into the tree.

After initiating a TRREQ, the requesting node waits the route discovery period before selecting a route. During this period, it keeps track of the best route, i.e. the smallest incremental power, to the multicast tree. At the end of the discovery time, the initiating node selects its best previous hop to the tree (the reverse path), and unicasts a Tree Branch Activation (TBA) message to this node. When the previous hop receives the TBA, it enables the next hop entry for the node from which it receives this TBA in its MRT by setting its Added flag as TRUE. If this node is already a multicast tree node, the addition of the new branch to the tree is finished. Otherwise, if this node is an intermediate node, it unicasts the TBA to its last hop, so that the last hop can accordingly activate its next hop entry for this node. Such processing continues until the node responding to the initiating node with the TRREP is reached and a new branch to the tree is added. Figure 4.19 illustrates an example of repairing a tree link break.

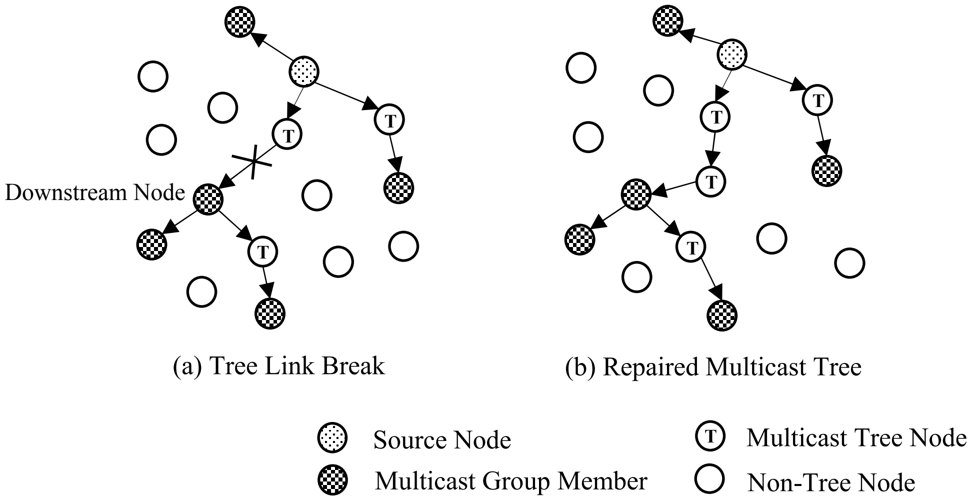


**Figure 4.19** Repair of Multicast Tree Branch.

### 4.6.3 Simulation Results

In this section, we perform a series of simulations to evaluate the tree maintenance algorithm. The simulations are performed using the GloMoSim Network Simulator [25]. This simulator is commonly used by ad hoc network researchers, which includes models for IP and UDP, and also allows for network node mobility. Each simulation simulates 300 seconds and models a wireless mobile ad hoc network of 50 nodes placed randomly within a $1500m \times 300m$ rectangular terrain. Since the tree maintenance algorithm is independent of the tree-establishing algorithm, we use the MSTH algorithm proposed in Section 4.3.2 to form a multicast tree at the beginning of the simulation. Then all nodes begin to move constantly with a predefined speed. Three different moving speeds are tested: 1m/s, 5m/s and 10m/s. Moving directions of each node are selected randomly, and when the node reaches the terrain boundary, it bounces back and continues to move. The tree maintenance algorithm is run at each node throughout the simulation. Table 4.2 shows the essential parameter values for the tree maintenance algorithm.

**Table 4.2** Simulation Parameter Values.

| Parameter Name | Meaning | Value |
|---|---|---|
| beacon_frequency | Frequency of beacon message broadcast | 1 sec |
| beacon_timeout | Max time allowed between beacon receptions | 3 sec |
| route_discovery_timeout | Max time to wait for a TRREP | 1 sec |

The MAC layer protocol used in the simulations is the modified IEEE standard 802.11 Distributed Coordination Function (DCF) [22], with discrete power control implemented. All nodes can adjust their transmission power level, from a minimum of 50*m* power range, to a maximum of 500*m* range, with an increase of 50*m* each level, i.e. 50*m*, 100*m*, 150*m*, 200*m*, 250*m*, 300*m*, 350*m*, 400*m*, 450*m*, and 500*m*.

Two different membership distribution patterns are experimented, a sparse group with only 5 group members and a dense group with 30 group members. In each scenario of one source and variable group membership, source and member nodes are randomly selected. The source node sends out data packets at a constant rate of eight packets per second throughout the simulation. Each data packet is 64 bytes. And the link bandwidth is 2 Mb/s. Multiple runs with different seed numbers are performed and the results are averaged over all the runs.

As a benchmark, we also implement a tree maintenance algorithm without power control. All nodes transmit at a fixed 250*m* power range. Hop count is used as the routing metric and the route selection criterion, instead of incremental power consumption used in the power controlled tree maintenance algorithm. In the legend of the following results, PC_5 denotes power-controlled algorithm with only 5 group members, NO_PC_30 denotes algorithm without power control and with 30 group members, and so on.

First, let us look at the normalized packet delivery ratio (PDR), as illustrated in Figure 4.20. In our simulation, the PDR is calculated by taking the number of data packets received by all group members, divided by the number of data packets transmitted from the source. Control packets are not counted. Since each data must be received by each of multicast group member, the PDR is then divided by the multicast group size to yield the normalized overall PDR. The power-controlled algorithm performs better than non-power-controlled one, especially in high mobility scenario. With the increase of mobility, the PDR of non-power-controlled algorithm drops very fast, while the dropping rate is relatively slow for power-controlled case. To explore why the PDR is different for these two algorithms, we investigate other aspects of the network, as discussed below.

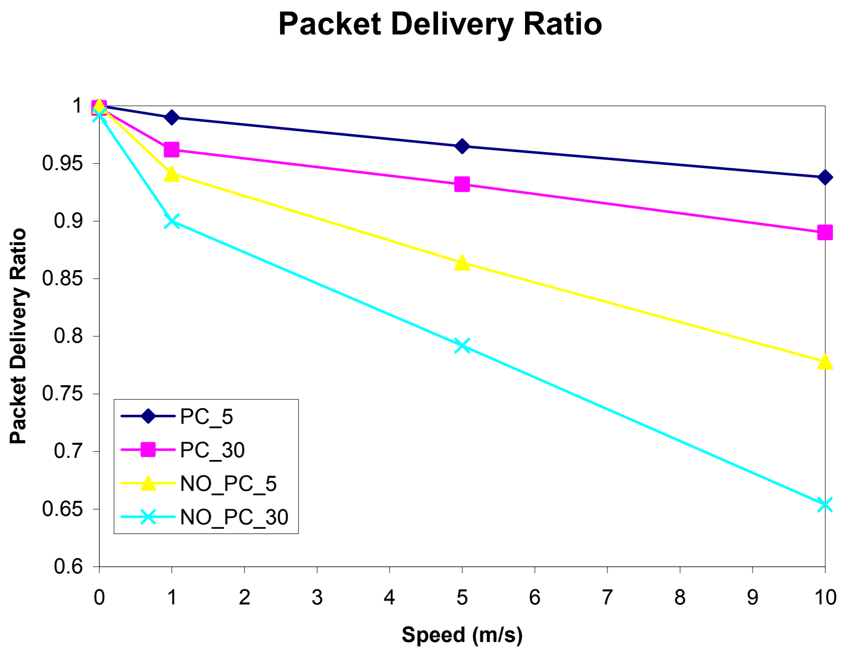


**Figure 4.20** Packet Delivery Ratio.

Figure 4.21 shows the average number of repairs needed to fix broken tree links during the simulation. For zero mobility, there are no repairs needed to the multicast tree. The number of the repairs increases for increasing speed, as more links are likely to break in high mobility situation. Also, the number of repairs increases with the increase of the membership size, because there are more tree nodes and links and therefore more possible breaks. It is very clear that power-controlled algorithm is efficient in reducing the number of the tree link breaks, thanks to the simple but effective power tuning algorithm. With the adaptivity to increase transmission power, a lot of potential link breaks are fixed without running the tree maintenance algorithm. On the contrary, the non-power-controlled algorithm has no means to reduce the link breaks. When mobility is low, as it uses a fixed 250*m* transmission range, the number of breaks are not too high, compared with power-controlled algorithm. However, after the moving speed increases, the number of tree breaks increases very sharply, especially in 10m/s case. A fixed medium power level now cannot prevent too many tree links breaking due to high mobility.

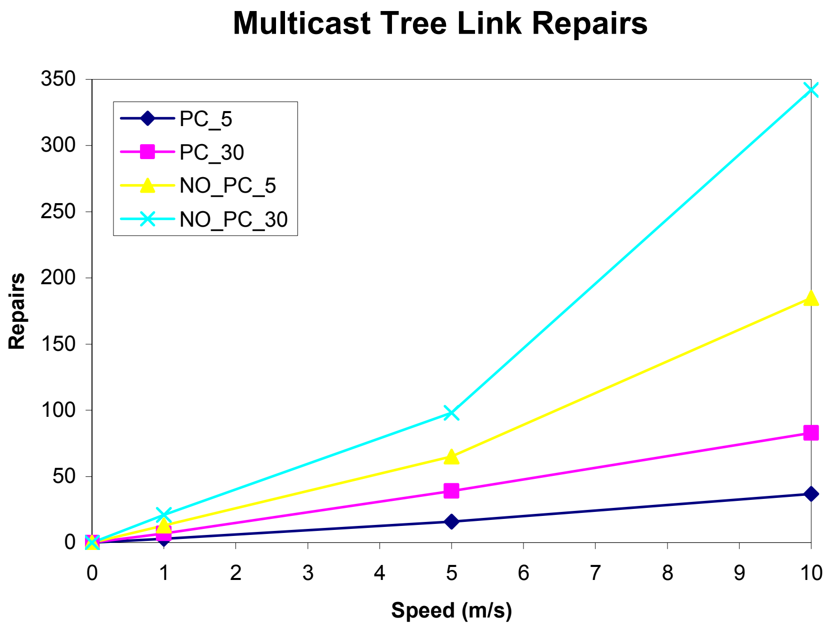


**Figure 4.21** Multicast Tree Link Repairs.

The amount of control overhead generated directly corresponds to the number of the repairs to the multicast tree, as illustrated by Figure 4.22. The number of control messages is calculated by summing the number of TRREQ, TRREP, and TBA initiated; the beacon messages are not counted in. As expected, the figure yields a similar curve as Figure 4.21. The difference in control overhead between power-controlled algorithm and non-power-controlled algorithm becomes more significant once the nodes move faster. As nodes travel more quickly, there are more breaks and repairs on the; tree, and hence more control messages generated for non-power-controlled case. Because there are fewer link breaks for power-controlled case, as shown in Figure 4.21, there are subsequently fewer control overhead generated.

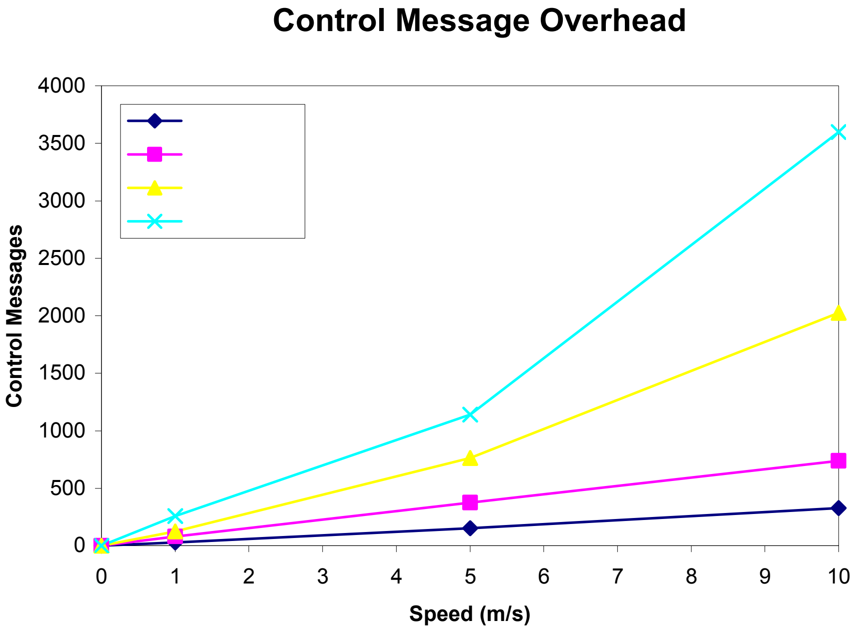


**Figure 4.22** Control Message Overhead.

The average total power consumption results are given in Figure 4.23. The power consumption is calculated each time a tree repair is finished and a complete multicast tree is maintained. This total power consumption consists of the transmission power for both data and control packets. It is exciting to find out that power-controlled algorithm outperforms non-power-controlled algorithm significantly, especially in high mobility scenario. When mobility is zero, both algorithms consume the same amount of transmission power, which is only the power for data transmission using the same MSTH algorithm. As the mobility increases, the increase of power consumption of power-controlled algorithm is relatively slow, because the number of tree link breaks does not increase very quickly. This is due to our two-phase tree maintenance algorithm. Power tuning reduces the number of breaks significantly, and the tree repair algorithm uses incremental power consumption as the branch-adding criterion to fix the breaks. The increase of power consumption for non-power-controlled algorithm, however, increases very fast with increase of mobility, because it suffers much more tree breaks and less power-efficient tree repairs in high mobility situations.

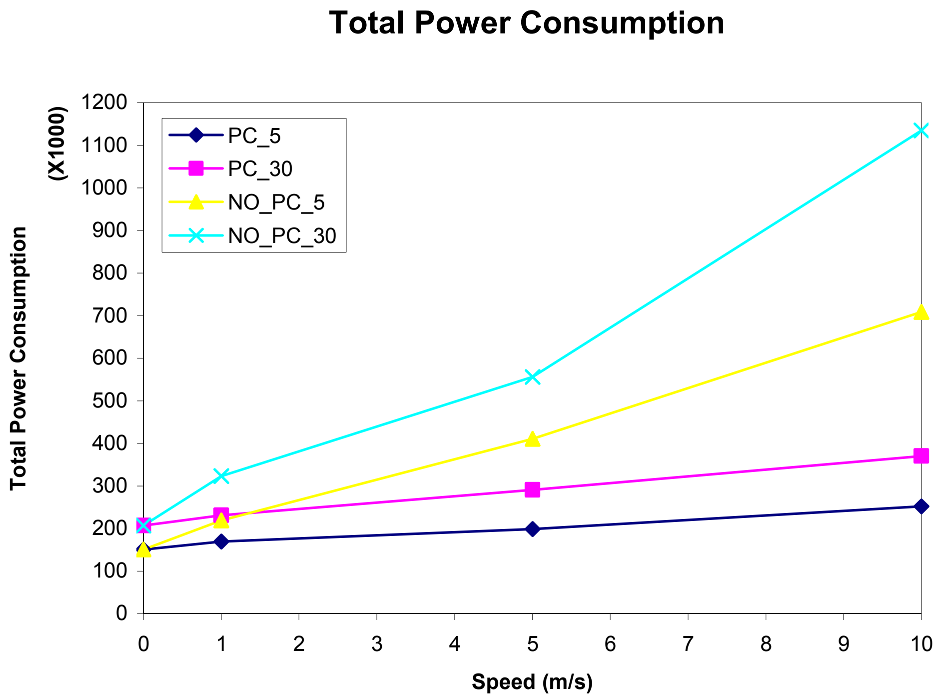


**Figure 4.23** Total Power Consumption.

How the power consumption for data and for control messages are affected by mobility is investigated and the results are shown in Figure 4.24 and 4.25, respectively.

It helps to understand why the total power consumptions are so different for power-controlled algorithm and non-power-controlled algorithm. For the data packets, their power consumption increases with increasing mobility. Even though we use incremental power as routing metric to add new branches after tree breaks, our tree maintenance algorithm is only a localized repair, due to the unavailability of global network information. Therefore, the repaired trees are not that globally power-efficient as the tree established using MSTH at the beginning of the simulation, when there is no mobility at all! Moreover, the discrete power control unavoidably incurs more power consumption. With the increase of moving speed, the power consumption increases for both power-controlled and non-power-controlled cases, because more tree links break and then are repaired locally. However, it is also obvious that the power-controlled algorithm conserves more transmission power for data packets than the non-power-controlled algorithm, thanks to the consideration of incremental power into its route selection criterion.

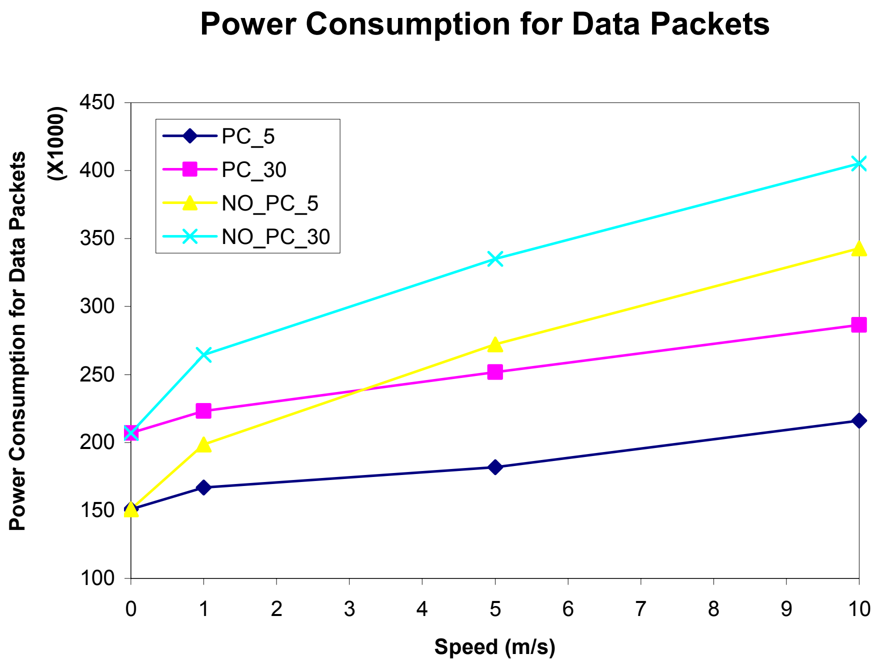


**Figure 4.24** Power Consumption for Data Packets.

The results of power consumption for control packet consolidate the findings we obtained from Figure 4.21 and 4.22. Because the number of tree breaks is reduced greatly for power-controlled algorithm, the power needed to exchange control packets is also reduced significantly, compared to the non-power-controlled algorithm. It is quite surprising to note that when mobility is high, for non-power-controlled algorithm, the power consumed by control packets is even higher than the power consumed by real data. Since the number of tree link breaks increases quickly, and only a fixed 250*m* power level is always used, a large amount of transmission power is wasted during the tree repair phase.

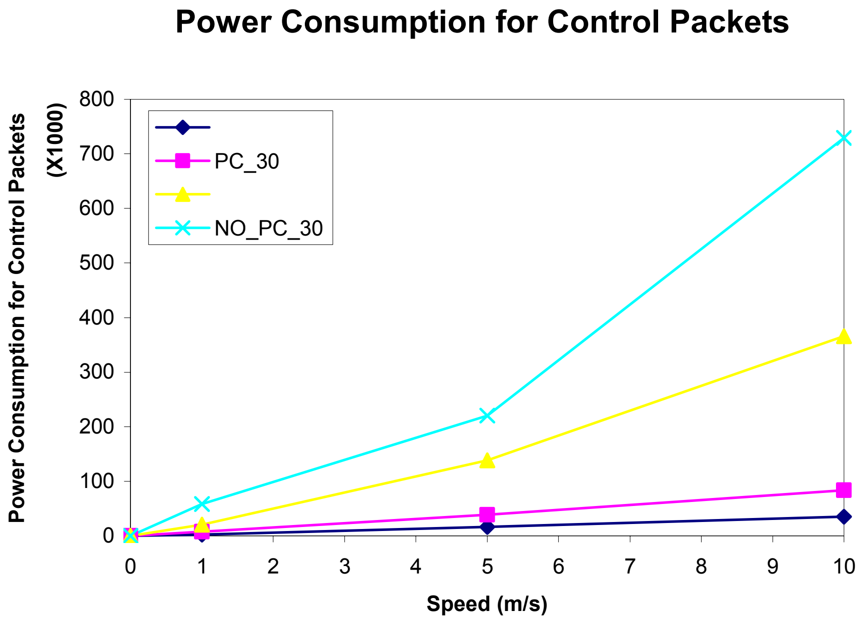


**Figure 4.25** Power Consumption for Control Packets.

Combining all the results obtained from above, we can conclude that our power-controlled tree maintenance algorithm is able to effectively reduce the number of tree link breaks, and repair the broken links in a locally power-efficient manner. As a result, it achieves higher PDR and consumes less transmission power than non-power-controlled algorithm.

## 4.7 Summary

The wireless nodal (broadcast) propagation property makes multicasting in wireless ad hoc networks an excellent scenario to study the potential benefits of power-efficient algorithms. In this chapter, we first propose a class of minimum-power multicasting algorithms using transmission power as cost function. Simulation results yield much better performance in terms of power efficiency and interference control, compared with algorithm without power control. However, we achieve this improvement at the cost of increased multicast tree size and average hop count. After careful investigation, we refine those algorithms with a new cost function, in which power and path efficiency are jointly optimized. Performance of algorithms using our new cost function demonstrates great improvement for path efficiency while still obtaining similar or even better power efficiency in comparison with algorithms using pure power as cost function. Finally, we extend our discussion to a full mobile environment. A distributed two-phase multicast tree maintenance algorithm is proposed. Simulation results demonstrate that it is able to effectively reduce the number of tree link breaks under mobility, and repair the broken tree link in a locally power-efficient manner.

# CHAPTER FIVE

# Conclusion

Wireless ad hoc networks have received renewed interests in the research community recently due to their promising popularity and potentials. However, there are also many challenges arising due to their infrastructureless nature. In this chapter, we conclude our thesis, which deals with the problem of power-efficient multicasting algorithm design, and suggest some future work about our research.

## 5.1 Summary and Contributions

We begin our thesis with some background study of multicasting concept. A review of current multicast protocols for wireless ad hoc networks, including LAM, AMRoute, AMRIS, ODMRP, CAMP, and MAODV, is presented and compared in detail.

Then some background of power control in cellular mobile networks is introduced. The challenges and motivations of power control design in wireless ad hoc networks are discussed, and a brief survey of current power control research is reported, which motivates our research interest on the designs of power-efficient multicasting algorithms.

After establishing a "node-based" wireless networking model and highlighting the wireless nodal propagation advantage for power efficiency consideration, we propose a class of minimum-power multicasting algorithms using transmission power as the cost function. The simulation results demonstrate much better power savings and interference control for our min-power heuristics compared with a baseline algorithm without power control, at the cost of increasing multicast tree size and average hop count. Based on this observation, we define a new cost function to jointly optimize power conservation and path efficiency. Performance of algorithms using this new cost function demonstrates significant improvement of path efficiency while still achieving similar or even better power efficiency in comparison with algorithms using pure power as the cost function. Finally, we extend our discussion to a full mobile environment. A distributed two-phase multicast tree maintenance algorithm is proposed. Simulation results show that it is able to effectively reduce the number of tree link breaks and repair the broken tree link in a locally power-efficient manner.

## 5.2 Future Work

The approaches in our studies aim to improve power efficiency of multicasting algorithms in general. We have not studied the possibility of implementing these algorithms in real network protocols, which can surely become the direction of future work. In addition, our tree-forming algorithms are all centralized in order to achieve near-optimal performance for the global power conservation. How to design the distributed version of those algorithms is also a challenging work worthy of further investigation.

# REFERENCES

[1] Z. J. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-Hoc Communications", *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Bedford, MA, October 1998, pp. 187-192.

[2] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter, http://www.ietf.org/html.charters/manet-charter.html.

[3] C. E. Perkins, "Mobile Ad Hoc Networking Terminology", *IETF Internet-Draft*, **draft-ietf-manet-term-01.txt**, November 1998, Work in Progress.

[4] C. E. Perkins (ed.), *Ad Hoc Networking*, Addison Wesley: Reading, MA, 2001.

[5] J. Jubin and J. D. Tornow, "The DARPA Packet Radio Network Protocols", *Proceeding of the IEEE*, Vol. 75, No. 1, January 1987, pp.21-32.

[6] J. Haartsen, "The Bluetooth Radio System", *IEEE Personal Communications*, Vol. 7, No. 1, February 2000, pp. 28-36.

[7] R. Wittmann and M. Zitterbart, *Multicast Communication: Protocols and Applications*, Morgan Kaufmann Publishers: San Francisco, CA, 2001.

[8] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms", *IEEE Journal on Selected Areas in Communications*, special issue on Multipoint Communications, Vol. 15, No. 3, April 1997, pp. 277-290.

[9] S. E. Deering and D. R. Cheriton, “Multicast Routing in Datagram Internetworks and Extended LANs”, *ACM Transactions on Computer Systems*, Vol. 8, No. 2, May 1990, pp. 85-110.

[10] J. Moy, “Multicast Routing Extensions for OSPF”, *Communications of the ACM*, Vol. 37, No. 8, August 1994, pp. 61-66, 114.

[11] T. Ballardie, P. Francis, and J. Crowcroft, "Core-based trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing," *Proceedings of the ACM SIGCOMM’93*, San Francisco, CA, October 1993, pp. 85-95.

[12] S. Deering, et al., “The PIM Architecture for Wide-Area Multicast Routing”, *IEEE/ACM Transactions on Networking*, Vol. 4, No. 2, April 1996, pp. 153-162.

[13] J. Moy, “Link-State Routing”, In *Routing in Communications Networks*, edited by M. E. Steenstrup, Prentice Hall, 1995, pp. 135-157.

[14] G. S. Malkin and M. E. Steenstrup, “Distance-Vector Routing”, In *Routing in Communications Networks*, edited by M. E. Steenstrup, Prentice Hall, 1995, pp.83-98.

[15] K. S. Gilhousen et al., “On the Capacity of a Cellular CDMA System”, *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 2, May 1991, pp. 303-312.

[16] W. C. Y. Lee, “Power Control in CDMA”, *Proceedings of IEEE Vehicular Technology Conference (VTC)*, St. Louis, MO, May 1991, pp. 77-80.

[17] Z. Rosberg and J. Zander, “Toward a Framework for Power Control in Cellular Systems”, *ACM/Baltzer Wireless Networks Journal*, Vol. 4, 1998, pp. 215-222.

[18] N. Bambos, “Toward Power-Sensitive Network Architectures in Wireless Communications: Concept, Issues, and Design Aspects”, *IEEE Personal Communications*, Vol. 5, No. 3, June 1998, pp. 50-59.

[19] P. Karn, "MACA - A New Channel Access Method for Packet Radio," *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990, pp. 134-140.

[20] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, “MACAW: A Media Access Protocol for Wireless LANs”, *Proceedings of the ACM SIGCOMM’94*, London, UK, August 1994, pp. 212-215.

[21] J. Deng and Z. J. Haas, “Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks”, *Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC)*, Florence, Italy, October 1998, pp. 973-977.

[22] *IEEE Std 802.11-1997: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, New York, 1997.

[23] F. A. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part II – The Hidden Terminal Problem in Carrier Sense Multiple Access Modes and the Busy Tone Solution”, *IEEE Transactions on Communications*, Vol. 23, No. 12, 1975, pp. 1417-1433.

[24] A. J. Goldsmith and S. B. Wicker, "Design Challenges for Energy-Constrained Ad Hoc Wireless Networks", *IEEE Wireless Communications*, Vol. 9, No. 4, August 2002, pp. 8-27.

[25] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems*, http://pcl.cs.ucla.edu/projects/domains/glomosim.html.

[26] Y. K. Dalal and R. M. Metcalfe, "Reverse Path Forwarding for Broadcast Packets", *ACM Transactions on Communications*, Vol. 21, No. 12, 1978, pp. 1040-1048.

[27] S. Deering, "Host Extensions for IP Multicasting", *RFC 1112*, August 1989.

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

[29] F. K. Hwang and D. S. Richards, "Steiner Tree Problems", *Networks*, Vol. 22, January 1992, pp. 55-89.

[30] F. K. Hwang, D. S, Richards and P. Winter, *The Steiner Tree Problem*, Elsevier Science Publishers B. V., North-Holland, 1992.

[31] L. Ji and M. S. Corson, "A Lightweight Adaptive Multicast Algorithm", *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Sydney, Australia, November 1998, pp. 1036-1042.

[32] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol", *IETF Internet-draft*, **draft-talpade-manet-amroute-00.txt**, August 1998, Work in Progress.

[33] C. W. Wu and Y. C. Tay, "AMRIS: A Multicast Protocol for Ad hoc Wireless Networks", *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Atlantic City, NJ, November 1999, pp. 25-29.

[34] C. W. Wu, Y. C. Tay and C-K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)", *IETF Internet-draft*, **draft-ietf-manet-amris-spec-00.txt**, November 1998, Work in Progress.

[35] S-H Bae, S-J Lee, W. Su, and M. Gerla, "The Design, Implementation, and Performance Evaluation of the On-Demand Multicast Routing Protocol in Multihop Wireless Networks", *IEEE Network*, Vol. 14, No. 1, January/February 2000, pp. 70-77.

[36] S-J. Lee, William Su and Mario Gerla, "Ad hoc Wireless Multicast with Mobility Prediction", *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 4-9.

[37] S-J. Lee, William Su and Mario Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks", *IETF Internet-draft*, **draft-ietf-manet-odmrp-02.txt**, January 2000, Work in Progress.

[38] J. J. Garcia-Luna-Aceves and E. L. Madruga, “A Multicast Routing Protocol for Ad-Hoc Networks”, *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, New York, NY, March 1999, pp. 784-792.

[39] J. J. Garcia-Luna-Aceves and E. L. Madruga, “The Core-Assisted Mesh Protocol”, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, August 1999, pp. 1380-1394.

[40] E. M. Royer and C. E. Perkins, “Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol”, *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Seattle, WA, August 1999, pp. 207-218.

[41] E. M. Royer and C. E. Perkins , “Multicast Ad hoc On Demand Distance Vector (MAODV) Routing”, *IETF Internet-draft*, **draft-ietf-manet-maodv-00.txt**, July 2000, Work in Progress.

[42] S-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, “A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols”, *Proceedings of IEEE INFOCOM 2000,* Tel Aviv, Israel, March 2000, pp. 565-574.

[43] V. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”, *Proceedings of IEEE INFOCOM’97*, Kobe, Japan, April 1997, pp. 1405-1413.

[44] V. Park and M. S. Corson, “Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification”, *IETF Internet-draft*, **draft-ietf-manet-tora-spec-00.txt**, December 1997, Work in Progress.

[45] C. E. Perkins and E. M. Royer, Ad-hoc On-Demand Distance Vector Routing, *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

[46] C. E. Perkins, E. M. Belding-Royer, and Samir Das, "Ad Hoc On Demand Distance Vector (AODV) Routing", *IETF Internet draft*, **draft-ietf-manet-aodv-12.txt**, November 2002, Work in Progress.

[47] J. P. Monks, V. Bharghavan, and W. Hwu, "Transmission Power Control for Multiple Access Wireless Packet Networks", *Proceedings of IEEE Conference on Local Computer Networks (LCN),* Tampa, FL, November 2000, pp. 12-21.

[48] J. P. Monks, V. Bharghavan, and W. Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks", *IEEE INFOCOM 2001*, April 2001, pp. 219-228.

[49] S. Wu, Y. Tseng, and J. Sheu, "Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control", *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 9, September 2000, pp. 1647-1657.

[50] S. Singh and C. S. Raghavendra, "PAMAS - Power-Aware Multiple Access protocol with Signaling for Ad Hoc Networks", *ACM SIGCOMM Computer Communication Review*, Vol. 28, No. 3, July 1998, pp. 5-26.

[51] V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, August 1999, pp. 1333-1344.

[52] D. M. Blough, M. Leoncini, G. Resta and P. Santi, “On the Symmetric Range Assignment Problem in Wireless Ad Hoc Network”, *Proceedings of 2nd IFIP International Conference on Theoretical Computer Science (TCS ’02)*, Montréal, Québec, Canada, August 2002, pp. 71-82 .

[53] G. Calinescu, I. I. Mandoiu, and A. Z. Zelikovsky, “Symmetric Connectivity with Minimum Power Consumption in Radio Networks”, *Proceedings of 2nd IFIP International Conference on Theoretical Computer Science (TCS ’02)*, Montréal, Québec, Canada, August 2002, pp. 119-130.

[54] D. J. Baker and A. Ephremides, “The Architectural Organization of A Mobile Radio Network via A distributed Algorithm”, *IEEE Transactions on Communications*, Vol. COM-29, No. 11, January 1981, pp. 56-73.

[55] A. Ephremides, J. E. Wieselthier, and D. J. Baker, “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping signaling”, *Proceedings of the IEEE*, Vol. 75, No. 1, January 1987, pp. 56-73.

[56] M. Ettus, “System Capacity, Latency, and Power Consumption in Multihop-routed SS-CDMA Wireless Networks”, *Proceedings of IEEE Radio and Wireless Conference (RAWCON ’98)*, Colorado Springs, CO, August 1998, pp. 55-58.

[57] T. H. Meng and V. Rodoplu, “Distributed Network Protocols for Wireless Communication”, *Proceedings of 3rd IEEE International Symposium on Circuits and Systems (ISCAS ’98)*, Monterey, CA, June 1998, pp. 600-603.

[58] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", *Proceedings of 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, Dallas, TX, October 1998, pp. 181-190.

[59] J. H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks", *Proceedings of ACM/IEEE INFOCOM '00*, Tel Aviv, Israel, March 2000, pp. 22-31.

[60] I. Stojmenovic and Xu Lin, "Power Aware Localized Routing in Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 11, November 2001, pp. 1122-1133.

[61] E. M. Royer and C. E. Perkins. "Transmission Range Effects on AODV Multicast Communication", To Appear in *ACM Mobile Networks and Applications*, 2002.

[62] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Multicasting in Energy-limited Ad-Hoc Wireless Networks", *Proceedings of the IEEE MILCOM'98*, Bedford, MA, October 1998, pp. 723-729.

[63] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for Energy-Efficient Multicasting in Ad Hoc Wireless Networks", *Proceedings of the IEEE MILCOM'99*, pp. 1414-1418.

[64] D. Bertsekas and R. Gallager, *Data Networks (2nd Edition)*, Prentice Hall, 1992.

[65] T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*, Prentice Hall, 2002.

[66] M. Cagalj, J-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues", *Proc. ACM MOBICOM'02*, Atlanta, GA, September 2002.

[67] A. Das, R.J. Marks II, M.A. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: integer programming formulations", *Proc. IEEE INFOCOM'03,* San Francisco, CA, March/April, 2003.

[68] S. Banerjee, A. Misra, J. Yeo, and A. Agrawala, "Energy-efficient broadcast and multicast trees for reliable wireless communication", *Proc. IEEE WCNC'03*, New Orleans, LA, March 2003.